Document Number: MM9Z1_638D1
Rev. 4.0, 10/2016

Intelligent battery sensor with CAN and LIN

The MM9Z1_638 is a fully integrated intelligent battery monitoring system. The device supports precise current measurement via an external shunt resistor. It features four voltage measurements via internal calibrated resistor dividers or external dividers. It includes an internal temperature sensor, allowing close proximity battery temperature measurements, plus four external temperature sensor inputs.

The MM9Z1_638 features the LIN 2.2 protocol and physical interface, as well as an msCAN protocol controller, for interfacing to automotive buses. The MM9Z1_638 is able to supply and control external CAN interfaces.

This device is powered by SMARTMOS technology.

MM9Z1_638

BATTERY MONITORING SYSTEM

EP SUFFIX (WF-TYPE)
98ASA00343D
48-PIN QFN

Features

- Wide range battery current measurement; On-chip temperature measurement
- Four battery voltage measurements with internal resistor dividers, and up to five direct voltage measurements for use with an external resistor divider
- Measurement synchronization between voltage channels and current channels
- Five external temperature sensor inputs with internal supply for external sensors
- Low-power modes with low-current operation
- Multiple wake-up sources: LIN, timer, high-voltage input, external CAN interface, and current threshold and integration
- Precision internal oscillator and connections for external crystal
- LIN 2.2/ 2.1/ 2.0 protocol and physical interface
- msCAN protocol controller, and supply capability for 8 and 14 pin CAN interfaces
- MM9Z1_638: S12Z microcontroller with 96/128 kByte Flash, 8.0 kByte RAM, 4.0 kByte EEPROM

Applications

- 12 V lead-acid battery monitoring
- Multi-cell battery (Li-ion) monitoring
- HV battery pack sensor
- Truck battery monitoring

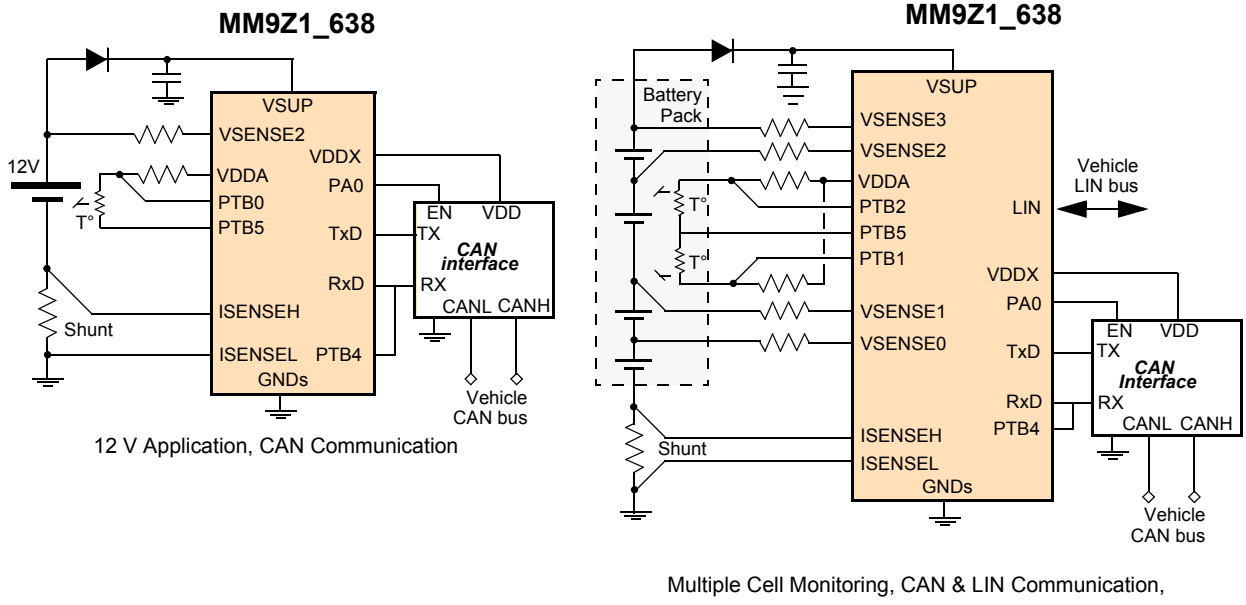


Figure 1. 12 V simplified application diagrams

© 2016 NXP B.V.

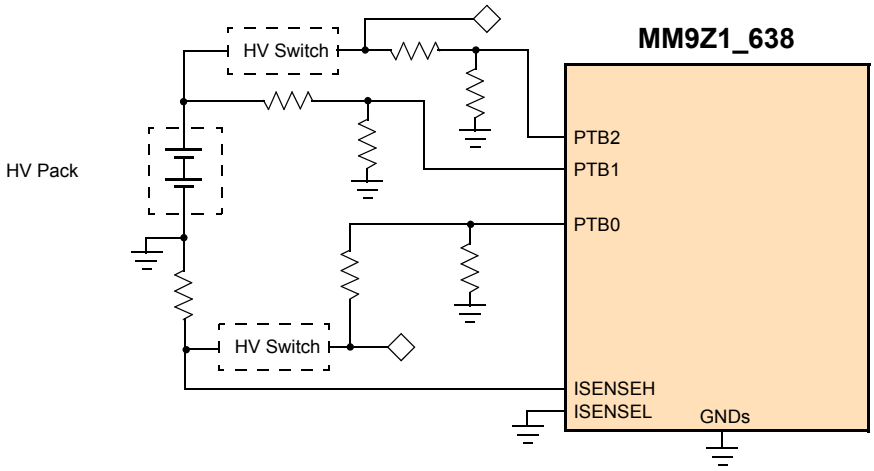


Figure 2. Simplified application diagram for HV pack monitoring - Use of an external divider principle schematic

Table of contents

1 Ordering information . . . 6
3 Pin assignment . . . 8
3.1 MM9Z1_638 pin description . . . 8
3.2 Typical applications . . . 10
4 Electrical characteristics . . . 13
4.1 General . . . 13
4.2 Absolute maximum ratings . . . 13
4.3 Operating conditions . . . 14
4.4 Supply currents . . . 14
4.4.1 Measurement conditions . . . 14
4.5 Analog die electrical characteristics . . . 17
4.5.1 Static electrical characteristics . . . 17
4.5.2 Dynamic electrical characteristics . . . 22
4.6 S12ZI128 electrical characteristics . . . 25
4.6.1 Electrical Characteristics . . . 25
4.6.2 NVM electrical parameters . . . 28
4.6.3 Electrical specification for voltage regulator . . . 30
4.6.4 OSCLCPcr electrical specifications . . . 31
4.6.5 PLL electrical specifications . . . 32
4.6.6 IRC electrical specifications . . . 33
4.6.7 SPI electrical specifications . . . 33
4.7 Thermal protection characteristics . . . 37
4.8 Electromagnetic compatibility (EMC) . . . 37
5 MM9Z1_638 overview . . . 39
5.1 MM9Z1_638 analog die overview . . . 41
5.1.1 Introduction . . . 41
5.1.2 Features . . . 42
5.2 MC9S12ZI128 overview . . . 43
5.2.1 Introduction . . . 43
5.2.2 Features . . . 43
5.2.3 Module features . . . 44
5.2.4 Block diagram . . . 47
5.2.5 Device memory map . . . 47
5.2.6 Modes of operation . . . 52
5.2.7 Security . . . 53
5.2.8 Resets and interrupts . . . 55
5.2.9 BDC clock source connectivity . . . 57
5.2.10 COP configuration . . . 57
6 Functional description and application information . . . 58
6.1 Device register map . . . 58
6.1.1 Detailed module register map . . . 58
6.2 Clock, reset, and power management unit (S12ZCPMU + PCR) . . . 71
6.2.1 S12Z clock, reset and power management unit (S12ZCPMU) . . . 71
6.2.2 Analog die - power, clock and resets - PCR . . . 109
6.2.3 Window watchdog . . . 129
6.3 Channel acquisition . . . 132
6.3.1 Features . . . 132
6.3.2 Block diagrams . . . 133
6.3.3 Channel acquisition . . . 135
6.4 General purpose I/O - GPIO . . . 173
6.4.1 Introduction . . . 173
6.4.2 Features . . . 173
6.4.3 Block diagram . . . 174
6.4.4 Modes of operation . . . 175
6.4.5 Memory map and registers . . . 175

6.5 Port integration module (S12ZIPIMV1) . . . 183
6.5.1 Introduction . . . 183
6.5.2 External signal description . . . 184
6.3.3 Channel acquisition . . . 135
6.6 Die to die interface . . . 192
6.6.1 Die-to-die initiator (D2DIV2) . . . 192
6.6.2 Die to die interface - target . . . 204
6.7 Interrupt module (S12ZINTV0 + IRQ) . . . 206
6.7.1 Interrupt (S12ZINTV0) . . . 206
6.7.2 Interrupt module - IRQ . . . 216
6.8 ECC generation module (SRAM_ECCV1) . . . 223
6.8.1 Introduction . . . 223
6.8.2 Memory map and register definition . . . 223
6.8.3 Functional description . . . 228
6.9 Memory mapping control (S12ZMMCV1) . . . 230
6.9.1 Introduction . . . 230
6.9.2 External signal description . . . 232
6.9.3 Memory map and register definition . . . 232
6.9.4 Functional description . . . 236
6.10 S12Z debug module (S12ZDBGV2) . . . 239
6.10.1 Introduction . . . 239
6.10.2 External signal description . . . 241
6.10.3 Memory map and registers . . . 241
6.10.4 Functional description . . . 262
6.10.5 Application information . . . 278
6.11 Background debug controller (S12ZBDCV1) . . . 279
6.11.1 Introduction . . . 279
6.11.2 External signal description . . . 282
6.11.3 Memory map and register definition . . . 282
6.11.4 Functional description . . . 285
6.12 Serial peripheral interface (S12SPIV5) . . . 304
6.12.1 Introduction . . . 304
6.12.2 External signal description . . . 306
6.12.3 Memory map and register definition . . . 307
6.12.4 Functional description . . . 314
6.13 NXP's scalable controller area network (S12MSCANV3) . . . 324
6.13.1 Introduction . . . 324
6.13.2 External signal description . . . 326
6.13.3 Memory map and register definition . . . 327
6.13.4 Functional Description . . . 352
6.13.5 Initialization/application information . . . 363
6.14 128 KB flash module (S12ZFTMRZ128K4KV1) . . . 364
6.14.1 Introduction . . . 364
6.14.2 External signal description . . . 366
6.14.3 Memory map and registers . . . 366
6.14.4 Functional description . . . 384
6.14.5 Security . . . 402
6.14.6 Initialization . . . 404
6.15 Basic timer module - TIM (TIM16B4C) . . . 404
6.15.1 Introduction . . . 404
6.15.2 Signal description . . . 405
6.15.3 Memory map and registers . . . 405
6.15.4 Functional description . . . 416
6.15.5 Resets . . . 417
6.15.6 Interrupts . . . 417
6.16 Life time counter (LTC) . . . 418
6.16.1 Introduction . . . 418
6.16.2 Memory map and registers . . . 418

6.17 Serial communication interface (S08SCIV4) 421
6.17.1 Introduction 421
6.17.2 Memory map and registers 424
6.17.3 Functional description 431
6.18 LIN 435
6.18.1 Introduction 435
6.18.2 Overview 435
6.18.3 Memory map and registers 437
7 Packaging 443
7.1 Package dimensions 443
8 Revision history 449

1 Ordering information

Table 1. Ordering information

Device (Add an R2 suffix for tape and reel orders)	Temperature range (T_A)	Package	Flash (kB)	RAM (kB)	EEPROM (kB)
MM9Z1J638BM2EP	-40 °C to 125 °C	48 QFN-EP	128	8.0	4.0
MM9Z1I638BM2EP			96		

2 Internal block diagram

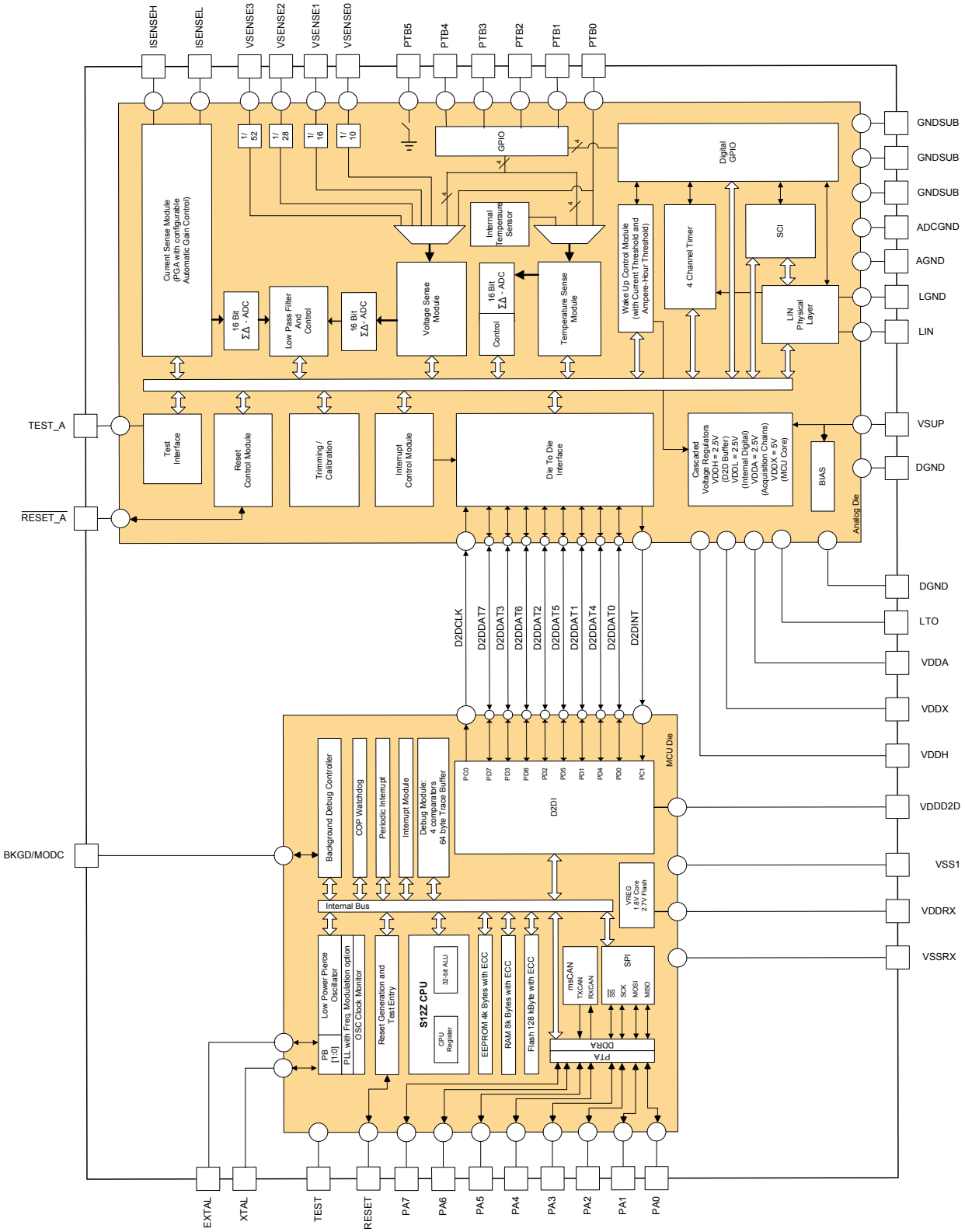


Figure 3. Simplified internal block diagram

3 Pin assignment

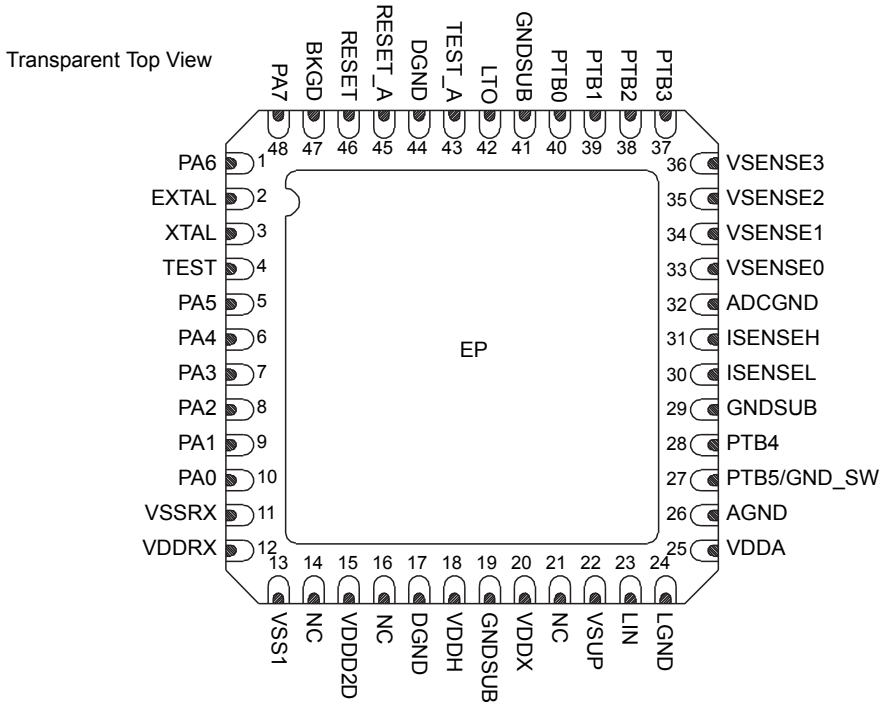


Figure 4. MM9Z1_638 pin connections

3.1 MM9Z1_638 pin description

The following table gives a brief description of all available pins on the MM9Z1_638 device. Refer to the highlighted chapter for detailed information

Table 2. MM9Z1_638 pin description

Pin #	Pin name	Formal name	Description
1	PA6	MCU PA6	General purpose port A input or output pin 6. See Port integration module (S12ZIPIMV1).
2	EXTAL	MCU Oscillator	EXTAL is one of the optional crystal/resonator drivers and external clock pins, and the PB0 port may be used as a general purpose I/O. On reset, all the device clocks are derived from the internal reference clock. See S12Z clock, reset and power management unit (S12ZCPMU).
3	XTAL	MCU Oscillator	XTAL is one of the optional crystal/resonator drivers and external clock pins, and the PB1 port may be used as a general purpose I/O. On reset all the device clocks are derived from the internal reference clock. See S12Z clock, reset and power management unit (S12ZCPMU).
4	TEST	MCU Test	This input only pin is reserved for test. This pin has a pull-down device. The TEST pin must be tied to VSSRX in user mode.
5	PA5	MCU PA5/CAN TXD	General purpose port A input or output pin 5. This pin is shared with the TXDCAN of the integrated msCAN module. See Port integration module (S12ZIPIMV1).
6	PA4	MCU PA4/CAN RXD	General purpose port A input or output pin 4. This pin is shared with the RXDCAN of the integrated msCAN module. See Port integration module (S12ZIPIMV1).
7	PA3	MCU PA3 / $\overline{SS}$	General purpose port A input or output pin 3, shared with the $\overline{SS}$ signal of the integrated SPI interface. See Port integration module (S12ZIPIMV1).
8	PA2	MCU PA2 / SCK	General purpose port A input or output pin 2, shared with the SCLK signal of the integrated SPI interface. See Port integration module (S12ZIPIMV1).

Table 2. MM9Z1_638 pin description (continued)

Pin #	Pin name	Formal name	Description
9	PA1	MCU PA1 / MOSI	General purpose port A input or output pin 1, shared with the MOSI signal of the integrated SPI interface. See Port integration module (S12ZIPIMV1).
10	PA0	MCU PA0 / MISO	General purpose port A input or output pin 0, shared with the MISO signal of the integrated SPI interface. See Port integration module (S12ZIPIMV1).
11	VSSRX	MCU 5.0 V Ground	External ground for the MCU - VDDRX return path.
12	VDDRX	MCU 5.0 V Supply	5.0 V MCU input power supply. This pin must be connected to VDDX.
13	VSS1	MCU 2.5 V Ground	External ground for the MCU - VDDD2D return path.
14	NC	Not connected	This pin must be grounded in the application.
15	VDDD2D	MCU 2.5 V Supply	2.5 V MCU input power supply for the die to die interface. This pin must be connected to VDDH.
16	NC	Not connected	This pin must be grounded in the application.
17	DGND	Digital Ground	This pin is the device digital ground connection.
18	VDDH	Voltage Regulator Output 2.5 V	2.5 V output voltage. This pin must be connected to the VDDD2D. An external capacitor (C_{VDDH}) is needed.
19	GNDSUB	Substrate Ground	Substrate ground connection.
20	VDDX	MCU 5.0 V and External CAN Supply	5.0 V output power supply for the internal MCU die and an external 8 or 14 pin CAN interface. This pin must be connected to VDDRX. An external capacitor (C_{VDDX}) is needed.
21	NC	Not connected	This pin is not connected to the die. Within the application, it could be connected or left open.
22	VSUP	Power Supply	This pin is the device power supply pin. A reverse battery protection diode is required. External capacitor(s) needed.
23	LIN	LIN Bus	This pin is the single-wire LIN bus.
24	LGND	LIN Ground	This pin is the device LIN ground connection.
25	VDDA	Analog Voltage Regulator Output	Voltage regulator output pin, to supply the analog front end, and the external temperature sensor circuitry. An external capacitor (C_{VDDA}) is needed.
26	AGND	Analog Ground	This pin is the device analog voltage regulator and LP oscillator ground connection.
27	PTB5/ GND_SW	GND Switch Temp	This pin is a switch to GND for connection of the external temperature sensors circuitry.
28	PTB4	Analog Input & General Purpose Input 4	General purpose 5.0 V Input (connection to timer and selectable pull-down). This pin shares several functions: Temperature sensor input. Direct voltage sense, used with an external resistor divider Wake-up input. General purpose 5.0 V input
29	GNDSUB	Substrate Ground	Substrate ground connection.
30	ISENSEL	Current Sense L	Current sense input “Low”. This pin is used in combination with ISENSEH to measure the voltage drop across a shunt resistor.
31	ISENSEH	Current Sense H	Current sense input “high”. This pin is used in combination with ISENSEL to measure the voltage drop across a shunt resistor.
32	ADCGND	Analog Digital Converter Ground	Analog digital converter ground connection.
33	VSENSE0	Voltage Sense 0	Precision battery voltage measurement input. This pin can be connected directly to the battery line for voltage measurements. The voltage preset at this input is scaled down by an internal voltage divider. The pin is self protected against reverse battery connections. An external resistor (R_{VSENSE}) is needed for protection.
34	VSENSE1	Voltage Sense 1	Precision battery voltage measurement input. This pin can be connected directly to the battery line for voltage measurements. The voltage preset at this input is scaled down by an internal voltage divider. The pin is self-protected against reverse battery connections. An external resistor (R_{VSENSE}) is needed for protection.

Table 2. MM9Z1_638 pin description (continued)

Pin #	Pin name	Formal name	Description
35	VSENSE2	Voltage Sense 2	Precision battery voltage measurement input. This pin can be connected directly to the battery line for voltage measurements. The voltage preset at this input is scaled down by an internal voltage divider. The pin is self-protected against reverse battery connections. An external resistor (R_{VSENSE}) is needed for protection.
36	VSENSE3	Voltage Sense 3	Precision battery voltage measurement input. This pin can be connected directly to the battery line for voltage measurements. The voltage preset at this input is scaled down by an internal voltage divider. The pin is self-protected against reverse battery connections. An external resistor (R_{VSENSE}) is needed for protection.
37	PTB3	Analog Input & General Purpose I/O 3	This pin shares several functions: Temperature sensor input. Direct voltage sense, used with an external resistor divider General purpose 5.0 V I/O (connection to timer, LIN SCI, and selectable pull-up to VDDX).
38	PTB2	Analog Input & General Purpose I/O 2	This pin shares several functions: Temperature sensor input. Direct voltage sense, used with an external resistor divider General purpose 5.0 V I/O (connection to timer, LIN SCI, and selectable pull-up to VDDX).
39	PTB1	Analog Input & General Purpose I/O 1	This pin shares several functions: Temperature sensor input. Direct voltage sense, used with an external resistor divider General purpose 5.0 V I/O (connection to timer, LIN SCI, and selectable pull-up to VDDX).
40	PTB0	Analog Input 0	This pin shares several functions: Temperature sensor input. Direct voltage sense, used with an external resistor divider
41	GNDSUB	Substrate Ground	Substrate ground connection.
42	LTO	Logic Test Output	Reserved pin for logic test output signal. Typical voltage is 2.5 V. Should be left open during device operation.
43	TEST_A	Test Mode	Test mode pin. This pin must be grounded in applications.
44	DGND	Digital Ground	This pin is the device digital ground connection.
45	RESET_A	Reset I/O	Reset output pin of the analog die in Normal mode. Bidirectional reset I/O of the analog die in Stop mode. Active low signal with internal pull-up to VDDX. This pin must be connected to RESET.
46	RESET	MCU Reset	Bidirectional reset I/O pin of the MCU die. Active low signal with internal pull-up to VDDRX. This pin must be connected to RESETA.
47	BKGD	MCU Background Debug and Mode	The BKGD/MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as an MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The BKGD pin has a pull-up device. See Background debug controller (BDC).
48	PA7	MCU PA7	General purpose port A input or output pin 7. See Port integration module (S12ZIPIMV1).

3.2 Typical applications

Figure 5 and Table 3 show a typical application to illustrate some of the device capabilities.

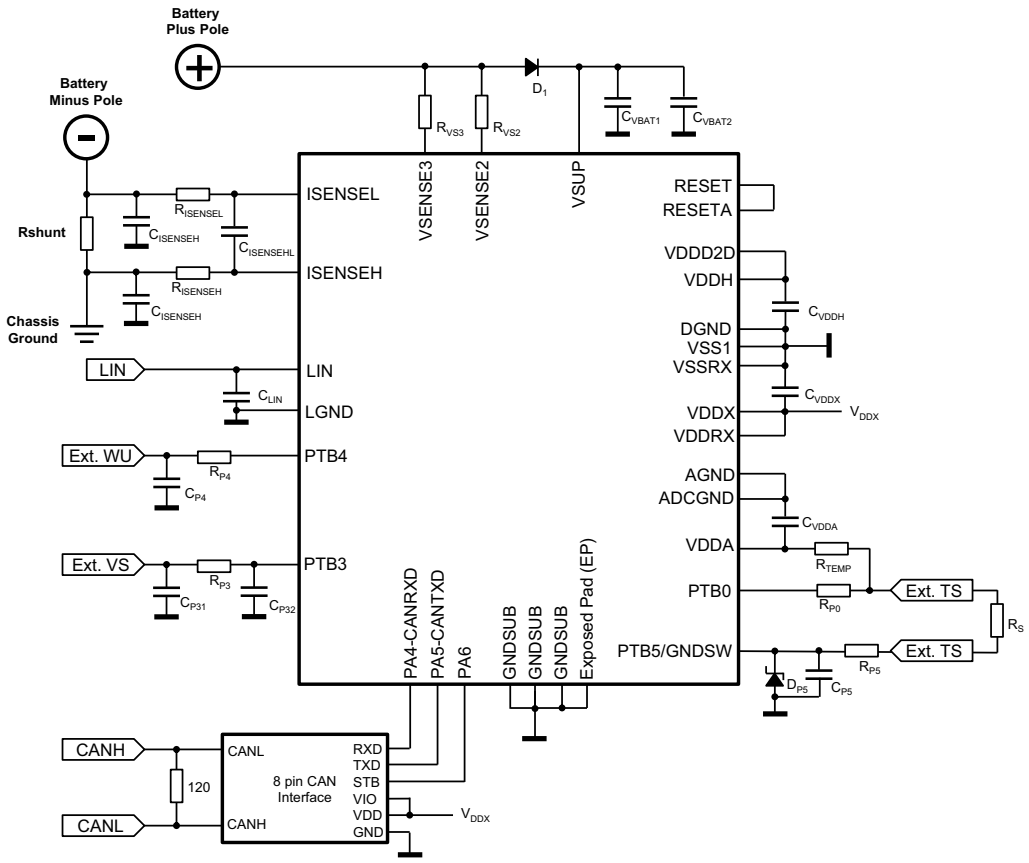
Both CAN and LIN communications are described.

VSENSE_2 and VSENSE_3 inputs are used to show possible redundancy.

External wake-up via PTB4 is used.

Single external voltage sense is used utilizing PTB3.

Single external temperature sense is used utilizing PTB0.



Note: Module GND connected to Battery Minus or Chassis Ground – based on configuration

Figure 5. Typical application example

Table 3. External components

Name	Description	value	Comment
D1	Reverse Battery Diode	N/A	
C_{VBAT1}	Battery Decoupling Capacitor	4.7 µF	
C_{VBAT2}	Battery Decoupling Capacitor	100 nF	
R_{VS2}	Vsense Current Limitation	2.2 kΩ	
R_{VS3}	Vsense Current Limitation	2.2 kΩ	
R_{SHUNT}	Current Measurement	50 to 200 µΩ	
$R_{ISENSEL}$	EMC resistor	max 500 Ω	Selection for best EMC perfomance
$R_{ISENSEH}$	EMC resistor	max 500 Ω	
$C_{ISENSEL}$	EMC capacitor	2.2 nF (typ)	
$C_{ISENSEH}$	EMC capacitor	2.2 nF (typ)	
$C_{ISENSEHL}$	EMC capacitor	2.2 nF (typ)	
C_{LIN}	LIN bus filter	Optional 220 pF	Selection per OEM requirement, for EMC and ESD performance.
R_{P4}	PTB4 Protection Resistor	10 kΩ	Minimum value to meet max rating
C_{P4}	PTB4 Protection Capacitor	10 nF	Minimum value to meet max rating
C_{VDDX}	VDDX Decoupling Capacitor	470 nF	
C_{VDDH}	VDDH Decoupling Capacitor	470 nF	An additionnal 4.7 nF capacitor might be required for specific EMC test conditions.

Table 3. External components (continued)

Name	Description	value	Comment
C_{VDDA}	VDDA Decoupling Capacitor	47 nF	
R_{TEMP}	Temp Sense Serial Resistor	100 kΩ	
R_S	Temperature Sensor	N/A	ex: NTC temperature Sensor
R_{P0}	PTB0 Protection Resistor	2.2 kΩ	To meet maximum rating. Higher or different value might be required. These components are optional and required to sustain EMC and ESD requirements when the pins go outside of the module.
R_{P5}	PTB5 Protection Resistor	1.0 kΩ	
C_{P5}	PTB5 Protection Capacitor	2.2 nF	
D_{P5}	PTB5 Protection Zener Diode	5.1 V	
C_{P31}	VSENSE_EXT Protection Capacitor	2.2 nF	To meet maximum rating. Higher or different value might be required. These components are optional and required to sustain EMC and ESD requirements when the pins go outside of the module.
C_{P32}	VSENSE_EXT Protection Capacitor	22 nF	
R_{P3}	VSENSE_EXT Protection Resistor	10 kΩ	

4 Electrical characteristics

4.1 General

This section contains electrical information for the microcontroller and the analog die.

4.2 Absolute maximum ratings

Absolute maximum ratings are stress ratings only. A functional operation under or outside these maximums is not guaranteed. Stress beyond these limits may affect the reliability, or cause permanent damage of the device.

This device contains circuitry protecting against damage due to high static voltage or electrical fields. However, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate voltage level. All voltages are with respect to ground, unless otherwise noted.

Table 4. Absolute maximum electrical ratings - analog die

Ratings	Symbol	Value	Unit
VSUP pin voltage	V_{VSUP}	-0.3 to 42	V
VSENSE0, VSENSE1, VSENSE2 pin voltage with 2.2 k resistor in serial (1)	$V_{VSENSE012}$	-40 to 42	V
VSENSE3 pin voltage with 2.2k resistor in serial (1)	$V_{VSENSE3}$	-40 to 62	V
PTB0, PTB1, PTB2, and PTB3 Voltage	VPTB0-3	-0.3 to VDDX+0.3	V
PTB4 direct voltage PTB4 with external 47 k serial resistor	VPTB4	-0.3 to 7.0 -27 to 42	V
PTB5 GND Switch current	IPTB5	1.0	mA
ISENSEH and ISENSEL pin voltage	V_{ISENSE}	-0.5 to V_{DDA}+0.25	V
ISENSEH and ISENSEL pin current	I_{ISENSE}	-1.0 to 1.0	mA
LIN pin voltage	V_{BUS}	-27 to 42	V
LIN pin current (internally limited)	I_{BUSLIM}	on page 18	mA
Voltage at VDDX	V_{DDX}	-0.3 to 5.55	V
Voltage at VDDH	V_{DDH}	-0.3 to 3.0	V
VDDH output current	I_{VDDH}	internally limited	mA
VDDX output current	I_{VDDX}	internally limited	mA
$\overline{\text{RESET_A}}$ pin voltage	V_{IN}	-0.3 to V_{DDX}+0.3	V

Notes:
1.It has to be assured by the application circuit that these limits will not be exceeded, e.g. by ISO pulse 1.

Table 5. Maximum thermal ratings

Ratings	Symbol	Value	Unit
Storage temperature	T_{STG}	-55 to 150	°C
Package thermal resistance (2)	$R_{\theta JA}$	32 typ.	°C/W

Notes:
2.$R_{\theta JA}$ value is derived using a JEDEC 2s2p test board

4.3 Operating conditions

This section describes the operating conditions of the device. Conditions apply to all the following data, unless otherwise noted.

Table 6. Operating Conditions [3]

Ratings	Symbol	Value	Unit
Functional operating supply voltage - Device is fully functional. All features are operating.	V_{SUP}	3.5 to 28	V
Extended range for RAM Content is guaranteed. Other device functionality is limited. With Cranking mode enabled (see Cranking mode).	V_{SUPL}	2.5 to 3.5	V
Functional operating VSENSE0 voltage[4]	V_{SENSE0}	0.0 to 10	V
Functional operating VSENSE1 voltage[4]	V_{SENSE1}	0.0 to 16	V
Functional operating VSENSE2 voltage[4]	V_{SENSE2}	0.0 to 28	V
Functional operating VSENSE3 voltage[4]	V_{SENSE3}	0.0 to 52	V
External voltage and temperature sense input - PTB0-4	V_{PTB0-4}	0.0 to 1.0	V
ISENSEH, ISENSEL voltage	VISH/L	-0.3 to 0.3	V
LIN output voltage range	V_{VSUP_LIN}	7.0 to 18	V
MCU oscillator	f_{OSC}	4.0 to 16.384	MHz
MCU bus frequency	f_{BUS}	51.2	MHz
Operating ambient temperature	T_A	-40 to 125	°C
Operating junction temperature - analog die	T_{J_A}	-40 to 150	°C
Operating junction temperature - MCU die	T_{J_M}	-40 to 150	°C

Notes:
3. The parametric data are guaranteed while the pins are within Operating Conditions. Other conditions are presented at top of parametric tables or noted into parameters.
4. Values for $V_{SENSE-x}$ > Max. Specified Value are flagged in the VTH bit.

4.4 Supply currents

This section describes the current consumption characteristics of the device, as well as the conditions for the measurements.

4.4.1 Measurement conditions

All measurements are without output loads. Unless otherwise noted the currents are measured in special single chip mode and the CPU code is executed from RAM. For Run and Wait current measurements PLL is on and the reference clock is the IRC1M trimmed to 1.024 MHz. The bus frequency is 51.2 MHz and the CPU frequency is 102.4 MHz. Table 7 and Table 8 show the configuration of the CPMU module for Run, Wait and Stop current measurement. Table 9 shows the configuration of the peripherals for run current measurement

Table 7. CPMU configuration for pseudo stop current measurement

CPMU REGISTER	Bit settings/conditions
CPMUCLKS	PLLSEL = 0, PSTP = 1, CSAD = 0, PRE = PCE = RTIOSCSEL = COPOSCSEL = 1
CPMUOSC	OSCE = 1, External Square wave on EXTAL f_{EXTAL} = 4.0 MHz, V_{IH} = 1.8 V, V_{IL} = 0 V
CPMURTI	RTDEC = 0, RTR[6:4] = 111, RTR[3:0] = 1111;
CPMUCOP	WCOP = 1, CR[2:0] = 111

Table 8. CPUM Configuration for Run/Wait and Full Stop Current Measurement

CPMU REGISTER	Bit settings/conditions
CPMUSYNR	VCOFRQ[1:0] = 3,SYNDIV[5:0] = 49
CPMUPOSTDIV	POSTDIV[4:0] = 0,
CPMUCLKS	PLLSEL = 1, CSAD = 0
CPMUOSC	OSCE = 0, Reference clock for PLL is f_{REF} = f_{IRC1M} trimmed to 1.024 MHz
API settings for stop current measurement	
CPMUAPICTL	APIEA = 0, APIFE = 1, APIE = 0
CPMUACLKTR	trimmed to ≥ 10 kHz
CPMUAPIRH/RL	set to 0xFFFF

Table 9. Peripheral configurations for run supply current measurements

Peripheral	Configuration
SPI	configured to master mode, continuously transmit data (0x55 or 0xAA) at 1.0 Mbit/s
msCAN	continuously transmit data (0x55 and 0xAA) at 1.0 MBaud/s
D2DI	continuously read data
COP	COP Warchdog Rate 2^{24}
RTI	enabled, RTI Control Register (RTICTL) set to $FF
DBG	the module is disabled

Table 10. Analog die configurations for normal mode supply current measurements

Peripheral	Configuration
D2D	maximum frequency (25.600 MHz)
LIN	enabled, 50% dominant, 50% recessive
TIMER	enabled, all channels active in output compare mode with minimum timeout
LTC	enabled, maximum timeout
SCI	continuously transmitting data (0x55 or 0xAA) with 19.2 kBit/s
Channels	Current/voltage: highest sampling rate (8.0 kHz), LPF enabled, chopper ON for Current and Temperature channels, OFF for Voltage channels and compensation enabled, automatic gain adjustment enabled temperature: internal temperature measurement enabled, 1.0 kHz sampling rate

Table 11. Supply currents

Parameter	Symbol	Min.	Typ.[5]	Max.	Unit
MM9Z1_638 combined consumption					
Normal mode current measured at V_{SUP} excluding external load current, (3.5 V ≤ V_{SUP} ≤ 28 V; -40 °C ≤ T_A ≤ 125 °C) parameter tested up to T_A = 85 °C	I_{RUN}	–	35	40	mA
Normal mode current measured at V_{SUP} - analog die contribution - excluding mcu and external load current, (3.5 V ≤ V_{SUP} ≤ 28 V; -40 °C ≤ T_A ≤ 125 °C) parameter tested up to T_A = 85 °C	I_{NORMAL}	–	1.5	4.0	mA
Stop mode current measured at V_{SUP}	I_{STOP}				µA
• Continuous base current [6]					
T = -40 °C		–	105	125	
T = 85 °C		–	110	195	
T = 125 °C [7]		–	210	450	
• Stop current during Cranking mode [6]					
T = -40 °C		–	110	135	
T = 85 °C		–	130	235	
T = 125 °C [7]		–	235	500	

Table 11. Supply currents

Parameter	Symbol	Min.	Typ.[5]	Max.	Unit
Sleep mode measured at V_{SUP} • Continuous base current [6] T = -40 °C T = 85 °C T = 125 °C [7]	I_{SLEEP}	 – – –	 65 65 85	 85 135 145	µA
Pseudo stop current [6] T = -40 °C T = 85 °C T = 125 °C [7]		 – – –	 205 210 310	 225 305 550	µA
Current adder during current trigger event in stop or sleep modes- (typ. 10 ms duration [8], temperature measurement = OFF)		–	1500	1750	µA

Notes:

5.Typical values noted reflect the approximate parameter mean at T_A = 25 °C.

6.From V_{SUP} 6.0 to 28 V

7.Guaranteed by design and characterization

8.Duration based on channel configuration. 10 ms typical for Decimation Factor = 512, Chopper = ON.

4.5 Analog die electrical characteristics

4.5.1 Static electrical characteristics

All characteristics noted under conditions 3.5 V ≤ V_{SUP} ≤ 28 V, -40 °C ≤ T_A ≤ 125 °C, unless otherwise noted. Parameters tested up to T_A = 85 °C unless otherwise noted.Typical values noted reflect the approximate parameter mean at T_A = 25 °C under nominal conditions, unless otherwise noted.

Table 12. Static electrical characteristics - power supply

Parameter	Symbol	Min.	Typ.	Max.	Unit
Low Voltage Reset L (POR) Assert (measured on LTO) • Cranking Mode Disabled	V_{PORL}	1.75	1.9	2.12	V
Low Voltage Reset L (POR) Deassert (measured on LTO) • Cranking Mode Disabled	V_{PORH}	1.85	2.1	2.35	V
Low Voltage Reset L (POR) Assert (measured on LTO) • Cranking Mode Enabled(9)	V_{PORCL}	1.0	1.3	1.7	V
Low Voltage Reset A (LVRA) Assert (measured on VDDA)	V_{LVRAL}	1.9	2.05	2.2	V
Low Voltage Reset A (LVRA) Deassert (measured on VDDA)	V_{LVRAH}	2.0	2.15	2.3	V
Low Voltage Reset X (LVRX) Assert (measured on VDDX)	V_{LVRXL}	2.5	2.75	3.0	V
Low Voltage Reset X (LVRX) Deassert (measured on VDDX)	V_{LVRXH}	2.7	2.95	3.25	V
Low Voltage Reset H (LVRH) Assert (measured on VDDH)	V_{LVRHL}	1.9	2.075	2.2	V
Low Voltage Reset H (LVRH) Deassert (measured on VDDH)	V_{LVRHH}	2.05	2.175	2.3	V
Undervoltage Interrupt (UVI) Assert (measured on VSUP), Cranking Mode Disabled	V_{UVIL}	4.65	5.2	6.15	V
Undervoltage Interrupt (UVI) Deassert (measured on VSUP), Cranking Mode Disabled	V_{UVIH}	4.9	5.4	6.3	V
Undervoltage Cranking Interrupt (UVI) Assert (measured on VSUP) Cranking Mode Enabled	V_{UVCIL}	3.4	3.6	4.0	V
Undervoltage Cranking Interrupt (UVI) Deassert (measured on VSUP) Cranking Mode Enabled	V_{UVCIH}	3.5	3.8	4.15	V
VSENSE2 High Voltage Warning Threshold Assert(10)	V_{TH}		28		V

Notes:
9.Deassert with Cranking off = V_{PORH}

10.5.0 V < V_{SUP} < 28 V, Digital Threshold at the end of channel chain (incl. compensation)

Table 13. Static electrical characteristics - resets

Parameter	Symbol	Min.	Typ.	Max.	Unit
Low-state Output Voltage I_{OUT} = 2.0 mA	V_{OL}	–	–	0.8	V
Pull-up Resistor	R_{RPU}	25	–	50	kΩ
Low-state Input Voltage	V_{IL}	–	–	$0.3V_{DDX}$	V
High-state Input Voltage	V_{IH}	$0.7V_{DDX}$	–	–	V
Reset Release Voltage (VDDX)	V_{RSTRV}	0.0	0.02	1.0	V
RESET_A pin Current Limitation	I_{LIMRST}	–	–	10	mA

Table 14. Static electrical characteristics - voltage regulator outputs

Parameter	Symbol	Min.	Typ.	Max.	Unit
Analog Voltage Regulator - VDDA(11)					
Output Voltage I_{VDDA} ≤ 1.0 mA	V_{DDA}	2.25	2.5	2.75	V
Output Current Limitation (Max. value occurs under VDDA short to GND condition)	I_{VDDA}	–	–	30	mA
Load current available for external sensor supply(i.e Temp sensor)	$I_{VDDA\ EXT}$	–	–	1.0	mA
Line regulation, V_{SUP} 3.5 to 28 V, I_Load 1.0 mA	LINE REGA	-30	–	30	mV

Table 14. Static electrical characteristics - voltage regulator outputs (continued)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Load regulation, I_Load 2.0 µA - 3.0 mA[12]	LOAD REGA	-50	–	50	mV
Voltage in Low-power modes (sleep and stop)	$V_{DDA\ LP}$	–	0.0	100	mV
Logic Test Output - LTO[11]					
Output Voltage	V_{LTO}	2.25	2.5	2.75	V
Output current limitation (for pin FMEA purpose only)	I_{LTO}	1.0	–	30	mA
High Power Digital Voltage Regulator - VDDH[13]					
Output Voltage 1.0 mA ≤ I_{VDDH} ≤ 18 mA	V_{DDH}	2.4	2.5	2.75	V
Output Current Limitation	I_{VDDH}	–	–	65	mA
5.0 V Voltage Regulator - VDDX[13]					
Output Voltage in Normal Mode, 1.0 mA ≤ I_{VDDX} ≤ 150 mA, V_{SUP} > 5.5 V	V_{DDX}	4.75	5.0	5.25	V
Output Current Limitation	I_{VDDX}	150	–	300	mA
Load current available for external supply - V_{SUP}> 5.5 V, for all external loads like: CAN transceiver, MCU I/Os, and PTBx for external temperature sensors.	$I_{VDDX\ EXT}$	–	–	100	mA
Output Voltage in Low-power Stop mode I_{VDDX} ≤ 2.0 mA • V_{SUP}> 5.5 V • V_{SUP} > 3.5 V	V_{DDXSTP}	4.75 3.2	5.0 –	5.25 –	V
Line regulation in normal mode, V_{SUP} 6.0 to 18 V, no load and I_{LOAD} = 150 mA	Line Reg Vx	-50	–	50	mV
Load regulation in normal mode, V_{SUP} 6.0 V, I_{LOAD} < 150 mA	Load Reg Vx	–	–	300	mV
Line regulation in Low-power Stop mode V_{SUP} 5.5 to 18 V	Line Reg VxLP	-50	–	50	mV
Load regulation in Low-power Stop mode	Load Reg VxLP	–	–	100	mV
Output current limitation in Low-power Stop mode	$I_{VDDXSTP}$	3.0	–	10	mA
Drop voltage, I_{LOAD} < I_{LIM} (I_{VDDX})	$VDDX_{DRP}$	–	–	300	mV
External decoupling capacitor	C at V_{DDX}	–	470	–	nF

Notes:
11.No additional current must be taken from those outputs.
12.Total VDDA regulator current, including internal device consumption
13.The specified current ranges does include the current for the MCU die. No external loads recommended.

Table 15. Static electrical characteristics - LIN physical layer interface - LIN

Parameter	Symbol	Min.	Typ.	Max.	Unit
Current Limitation for Driver dominant state. V_{BUS} = 18 V	I_{BUSLIM}	40	120	200	mA
Input Leakage Current at the Receiver incl. Pull-up Resistor R_{SLAVE}; Driver OFF; V_{BUS} = 0 V; V_{BAT} = 12 V	$I_{BUS_PAS_DOM}$	-1.0	–	–	mA
Input Leakage Current at the Receiver incl. Pull-up Resistor R_{SLAVE}; Driver OFF; 8.0 V < V_{BAT} < 18 V; 8.0 V < V_{BUS} < 18 V; V_{BUS} ≥ V_{BAT}	$I_{BUS_PAS_REC}$	–	–	20	µA
Input Leakage Current; GND Disconnected; GND_{DEVICE} = V_{SUP}; 0 < V_{BUS} < 18 V; V_{BAT} = 12 V	$I_{BUS_NO_GND}$	-1.0	–	1.0	mA
Input Leakage Current; V_{BAT} disconnected; V_{SUP_DEVICE} = GND; 0 < V_{BUS} < 18 V	$I_{BUS_NO_BAT}$	–	–	10	µA
Receiver Input Voltage; Receiver Dominant State	V_{BUSDOM}	–	–	0.4	V_{SUP}
Receiver Input Voltage; Receiver Recessive State	V_{BUSREC}	0.6	–	–	V_{SUP}
Receiver Threshold Center (V_{TH_DOM} + V_{TH_REC})/2	V_{BUS_CNT}	0.475	0.5	0.525	V_{SUP}
Receiver Threshold Hysteresis (V_{TH_REC} - V_{TH_DOM})	V_{BUS_HYS}	–	–	0.175	V_{SUP}
Voltage Drop at the serial Diode	D_{SER_INT}	0.3	0.7	1.0	V
LIN Pull-up Resistor	R_{SLAVE}	20	30	60	kΩ
LIN Internal Capacitor [14]	C_{LIN}	–	5.0	30	pF

Table 15. Static electrical characteristics - LIN physical layer interface - LIN (continued)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Low Level Output Voltage, I_{BUS}=40 mA	V_{DOM}	–	–	0.3	V_{SUP}
High Level Output Voltage, I_{BUS}=-10 µA, R_L=33 kΩ	V_{REC}	VSUP-1	–	–	V
J2602 Detection Deassert Threshold for VSUP level	V_{J2602H}	5.9	6.3	6.7	V
J2602 Detection Assert Threshold for VSUP level	V_{J2602L}	5.8	6.2	6.6	V
J2602 Detection Hysteresis	$V_{J2602HYS}$	70	190	250	mV
BUS Wake-up Threshold	V_{LINWUP}	4.0	5.25	6.0	V

Notes:
14.This parameter is guaranteed by process monitoring but not production tested.

Table 16. Static electrical characteristics - high voltage input - PTB4

Parameter	Symbol	Min.	Typ.	Max.	Unit
Wake-up Threshold - Rising Edge	V_{WTHR}	1.3	2.6	3.4	V
Input High-voltage (digital Input)	V_{IH}	$0.7V_{DDX}$	–	V_{DDX}+0.3	V
Input Low-voltage (digital Input)	V_{IL}	V_{SS}-0.3	–	$0.35V_{DDX}$	V
Input Hysteresis	V_{HYS}	–	140	–	mV
Internal Positive Clamp Voltage	$V_{PTB4CLMP}$	–	9.8	–	V
Input Current PTB4, V_{IN} = 8V	I_{IN}	-10	0	10	uA
Internal Pull-down Resistance[15]	R_{PD}	50	100	200	kΩ
External Series Resistor	R_{PTB4}	42.3	47	51.7	kΩ
External Capacitor	C_{PTB4}	–	2.2	–	nF

Notes:
15.Disabled by default.

Table 17. Static electrical characteristics - general purpose I/O - PTB[1…3]

Parameter	Symbol	Min.	Typ.	Max.	Unit
Input High Voltage	V_{IH}	$0.7V_{DDX}$	–	V_{DDX}+0.3	V
Input Low Voltage	V_{IL}	V_{SS}-0.3	–	$0.35V_{DDX}$	V
Input Hysteresis	V_{HYS}	–	140	–	mV
Input Leakage Current (pins in high-impedance input mode) (V_{IN} = V_{DDX} or V_{SSX})	I_{IN}	-1.0	–	1.0	µA
PTB1, 2, 3. Output High Voltage (pins in output mode), I_{OH} = –5.0 mA	V_{OH}	V_{DDX}-0.8	–	–	V
PTB1, 2, 3. Output Low Voltage (pins in output mode), I_{OL} = 5.0 mA	V_{OL}	–	–	0.8	V
Internal Pull-up Resistance for PTB1, 2, 3 only (V_{IH} min. > Input voltage > V_{IL} max)[16]	R_{PUL}	25	37.5	50	kΩ
Input Capacitance	C_{IN}	–	6.0	–	pF
Output Drive strength at 10 MHz	C_{OUT}	–	–	100	pF

Notes:
16.Disabled by default.

Table 18. Static electrical characteristics - general purpose I/O - PTB5

Parameter	Symbol	Min.	Typ.	Max.	Unit
PTB5 switch to GND: On resistance	$PTB5_{RON}$	20	50	100	Ω

Table 19. Static electrical characteristics - current sense module

Parameter	Symbol	Min.	Typ.	Max.	Unit
Gain Error [17], [18]					
• without common mode from -40 °C to 85 °C [20]					
without life time drift	$I_{GAINERR}$	-0.3	–	0.3	%
including life time drift		-0.5	–	0.5	
• without common mode from 85 °C to 125 °C [20]					
without life time drift		-0.5	–	0.5	
including life time drift		-0.7	–	0.7	
• additional common mode error for gain 64 and 256 [20]		-0.2	–	0.2	
Offset Error [17],[18],[19]	$I_{OFFSETERR}$	–	–	0.5	µV
Resolution (LSB)					
• COMP_TF.IRSEL = 000 (50 µΩ)	I_{RES}	–	0.05	–	µV
• COMP_TF.IRSEL = 001 (75 µΩ)		–	0.75	–	
• COMP_TF.IRSEL = 010 (100 µΩ)		–	0.1	–	
• COMP_TF.IRSEL = 011 (150 µΩ)		–	0.15	–	
• COMP_TF.IRSEL = 100 (200 µΩ)		–	0.2	–	
ISENSEH, ISENSEL [20]					
• terminal voltage range for gain 4 and 16 for specified gain error without additional common mode error	V_{INC}	-300	–	300	mV
• terminal voltage range for gain 64 for specified gain error without additional common mode error		-100	–	100	
• terminal voltage range for gain 256 for specified gain error without additional common mode error		-50	–	50	
• differential signal voltage range	V_{IND}	-150	–	150	
PGA Gains					
• gain 4	PGA_{GAIN}	–	4	–	
• gain 16		–	16	–	
• gain 64		–	64	–	
• gain 256		–	256	–	
Differential Leakage Current: differential voltage between ISENSEH/ ISENSEL, with IsenseH or IsenseL connected to GND	I_{SENSE_DLC}	-2.0	–	2.0	nA
Wake-up Current Threshold Resolution	$I_{RESWAKE}$	–	0.2	–	µV
Resistor Threshold for OPEN Detection	R_{OPEN}	0.8	1.25	1.8	MΩ

Notes:
17.Device trimmed and after system calibration. Automatic gain compensation over temperature.
18.Chopper Mode = ON, Gain with automatic gain control enabled
19.Parameter not tested. Guaranteed by design and characterization
20.Voltage level referred to AGND.

Table 20. Static electrical characteristics - voltage sense module [22]

Parameter	Symbol	Min.	Typ.	Max.	Unit
VSENSE0 internal resistor divider ratio	VS0_div	–	10	–	
VSENSE1 internal resistor divider ratio	VS1_div	–	16	–	
VSENSE2 internal resistor divider ratio	VS2_div	–	28	–	
VSENSE3 internal resistor divider ratio	VS3_div	–	52	–	
VSENSE0, accuracy (accuracy includes both gain and offset errors) [21], [23]					
• from -40 to 85 °C, input range 1.8 V to 10 V	$VS0_{ACC}$	-0.25	–	0.25	%
• from -40 to 85 °C, input range 1.25 V to 1.8 V		-0.5	–	0.5	
• from 85 to 125 °C, input range 1.8 V to 10 V		-0.5	–	0.5	
• from 85 to 125 °C, input range 1.25 V to 1.8 V		-0.6	–	0.6	
VSENSE1, accuracy (accuracy includes both gain and offset errors) [21], [23]					
• from -40 to 85 °C, input range 2.8 V to 16 V	$VS1_{ACC}$	-0.15	–	0.15	%
• from -40 to 85 °C, input range 2.0 V to 2.8 V		-0.5	–	0.5	
• from 85 °C to 125 °C, input range 2.8 V to 16 V		-0.25	–	0.25	
• from 85 °C to 125 °C, input range 2.0 V to 2.8 V		-0.5	–	0.5	

Table 20. Static electrical characteristics - voltage sense module [22] (continued)

Parameter	Symbol	Min.	Typ.	Max.	Unit
VSENSE2, accuracy (accuracy includes both gain and offset errors) [21], [23] • from -40 to 85 °C, input range 5.0 V to 28 V • from -40 to 85 °C, input range 3.5 V to 5.0 V • from 85 °C to 125 °C, input range 5.0 V to 28 V • from 85 °C to 125 °C, input range 3.5 V to 5.0 V	$VS2_{ACC}$	 -0.15 -0.5 -0.25 -0.5	 – – – –	 0.15 0.5 0.25 0.5	%
VSENSE3, accuracy (accuracy includes both gain and offset errors) [21], [23] • from -40 to 85 °C, input range 9.4 V to 50 V • from -40 to 85 °C, input range 6.5 V to 9.4 V or > 50 V • from 85 °C to 125 °C, input range 9.4 V to 50 V • from 85 °C to 125 °C, input range 6.5 V to 9.4 V and > 50 V	$VS3_{ACC}$	 -0.15 -0.5 -0.25 -0.5	 – – – –	 0.15 0.5 0.25 0.5	%
VSENSE[0..3] Drift • drift due to MSL3 (according JEDEC J-STD-020) reflow • drift due to Life Time Drift	VS_{DRIFT}	 – –	 0.03 -6.0	 – –	 % ppb/h
VSENSE0,resolution and offset [21] • voltage measurement resolution • extrapolated offset error	$VS0_{RO}$	 – –	 0.25 ±3.7	 – –	mV
VSENSE1,resolution and offset [21] • voltage measurement resolution • extrapolated offset error	$VS1_{RO}$	 – –	 0.25 ±3.7	 – –	mV
VSENSE2,resolution and offset [21] • voltage measurement resolution • extrapolated offset error	$VS2_{RO}$	 – –	 0.5 ±6.2	 – –	mV
VSENSE3,resolution and offset [21] • voltage measurement resolution • extrapolated offset error	$VS3_{RO}$	 – –	 1.0 ±19.4	 – –	mV

Notes:
21.Device trimmed and after system calibration. Automatic gain compensation over temperature.
22.The data are valid for both chop modes (off and on). The dies are delivered with chop off compensation values.
23.Including 2.2 kΩ perfect resistor into measurement path.

Table 21. Static electrical characteristics - internal temperature sense module

Parameter	Symbol	Min.	Typ.	Max.	Unit
Measurement Range	T_{RANGE}	-40	–	150	°C
Accuracy • -40 °C ≤ T_J ≤ 85 °C • 85 °C < T_J ≤ 150 °C [24]	T_{ACC}	 -2.0 -3.0	 – –	 2.0 3.0	K
Resolution	T_{RES}	–	8.0	–	mK
Max. Calibration Request Interrupt Temperature Step	$T_{CALSTEP}$	-25	–	25	K

Notes:
24.Temperature not tested in production. Guaranteed by design and characterization.

Table 22. Static electrical characteristics - PTB0 to 4 voltage and temperature measurements

Parameter	Symbol	Min.	Typ.	Max.	Unit
PTB0, 1, 2, 3, and 4: measurement resolution for voltage measurement (PTBx routed to Voltage Analog to Digital Converter)	$PTB0\text{-}4_{VRES}$	–	25	–	µV
PTB0, 1, 2, 3 and 4: voltage sense gain error from -40 °C to 85 °C • input range 0.13 V to 1.0 V [25] PTB0, 1, 2, 3 and 4: voltage sense gain error from 85 °C to 125 °C • input range from 0.13V to 1.0V [25]	$PTB0\text{-}4_{ACC}$	 -0.15 -0.25	 – –	 0.15 0.25	%
PTB0, 1, 2, 3 and 4: extrapolated voltage sense offset error, obtained by linear regression at 25 °C.	$PTB0\text{-}4_{OFF}$	-0.5	–	0.5	mV

Table 22. Static electrical characteristics - PTB0 to 4 voltage and temperature measurements (continued)

Parameter	Symbol	Min.	Typ.	Max.	Unit
PTB0, 1, 2, 3, and 4: measurement resolution for temperature measurement (PTBx routed to Temperature Analog to Digital Converter)	$PTB0\text{-}4_{TRES}$	–	19	–	µV
PTB0, 1, 2, 3 and 4: temperature sense gain error (25)	$PTB0\text{-}4_{TSG}$	-0.15	–	0.15	%
Offset temperature coefficient		-12	-7.0	0.0	µV/°C

Notes:
25.Device trimmed and after system calibration. Automatic gain compensation over temperature.

4.5.2 Dynamic electrical characteristics

Dynamic characteristics noted under conditions 3.5 V ≤ V_{SUP} ≤ 28 V, -40 °C ≤ T_A ≤ 125 °C, unless otherwise noted. Parameters tested up to T_A = +85° C, unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25 °C under nominal conditions, unless otherwise noted.

Table 23. Dynamic electrical characteristics - modes of operation

Parameter	Symbol	Min.	Typ.	Max.	Unit
Low Power Oscillator Frequency	f_{OSCL}	–	512	–	kHz
Low Power Oscillator Tolerance overtemperature range (26) • -40 to 85 °C • after life time -40 to 85 °C • 85 to 125 °C	f_{TOL_A}	 -3.0 -3.5 -8.0	 – – –	 3.0 3.5 3.5	%
Low Power Oscillator Tolerance - synchronized ALFCLK(27) • ALF clock cycle = 1.0 ms • ALF clock cycle = 2.0 ms • ALF clock cycle = 4.0 ms • ALF clock cycle = 8.0 ms	f_{TOLC_A}	 f_{TOL}-0.2 f_{TOL}-0.1 f_{TOL}-0.05 f_{TOL}-0.025	f_{TOL}	 f_{TOL}+0.2 f_{TOL}+0.1 f_{TOL}+0.05 f_{TOL}+0.025	%

Notes:
26.At T = 125 °C: min = -8.0%, max = -1.0%.
27.Parameter not tested. Guaranteed by design and characterization.

Table 24. Dynamic electrical characteristics - die to die interface - D2D

Parameter	Symbol	Min.	Typ.	Max.	Unit
Operating Frequency (D2DCLK, D2D[0:3])	f_{D2D}	–	–	25.600	MHz

Table 25. Dynamic electrical characteristics - resets

Parameter	Symbol	Min.	Typ.	Max.	Unit
Reset Deglitch Filter Time	t_{RSTDF}	1.0	2.0	3.2	µs
Reset Release Time for WDR and HWR	t_{RSTRT}	–	32	–	µs

Table 26. Dynamic electrical characteristics - wake-up / cyclic sense

Parameter	Symbol	Min.	Typ.	Max.	Unit
Cyclic Wake-up Time(28)	t_{WAKEUP}	ALFCLK	–	TIM4CH	ms
Cyclic Current Measurement Step Width(29)	t_{STEP}	ALFCLK	–	16Bit	ms

Notes:
28.Cyclic wake-up on ALFCLK clock based 16 Bit TIMER with maximum 128x prescaler (min 1x)
29.Cyclic wake-up on ALFCLK clock with 16 Bit programmable counter

Table 27. Dynamic electrical characteristics - window watchdog

Parameter	Symbol	Min.	Typ.	Max.	Unit
Initial Non-window Watchdog Timeout	t_{IWDTO}	256 ms. see Section 6.2.3			ms

Table 28. Dynamic electrical characteristics - LIN physical layer interface - LIN

Parameter	Symbol	Min.	Typ.	Max.	Unit
Bus Wake-up Deglitcher (Sleep and Stop Mode)	t_{PROPWL}	60	80	100	µs
Fast Bit Rate (Programming Mode)	BR_{FAST}	–	–	100	kBit/s
Propagation delay of receiver	t_{RX_PD}	–	–	6.0	µs
Symmetry of receiver propagation delay rising edge w.r.t. falling edge	t_{RX_SYM}	-2.0	–	2.0	µs
TxD Dominant Timeout	t_{TxDDOM}	4	–	8	ms
LIN Driver - 20.0 kBit/s; bus load conditions (C_{BUS}; R_{BUS}): 1.0 nF; 1.0 kΩ / 6,8 nF;660 Ω / 10 nF;500 Ω					
Duty Cycle 1: $TH_{REC(MAX)} = 0.744 \times V_{SUP}$ $TH_{DOM(MAX)} = 0.581 \times V_{SUP}$ $7.0\ V \le V_{SUP} \le 18\ V$; $t_{BIT} = 50\ µs$; $D1 = t_{BUS_REC(MIN)}/(2 \times t_{BIT})$	D1	0.396	–	–	
Duty Cycle 2: $TH_{REC(MIN)} = 0.422 \times V_{SUP}$ $TH_{DOM(MIN)} = 0.284 \times V_{SUP}$ $7.6\ V \le V_{SUP} \le 18\ V$; $t_{Bit} = 50\ µs$ $D2 = t_{BUS_REC(MAX)}/(2 \times t_{BIT})$	D2	–	–	0.581	
LIN Driver - 10.0 kBit/s; bus load conditions (C_{BUS}; R_{BUS}): 1.0 nF; 1.0 kΩ / 6,8 nF;660 Ω / 10 nF;500 Ω					
Duty Cycle 3: $TH_{REC(MAX)} = 0.778 \times V_{SUP}$ $TH_{DOM(MAX)} = 0.616 \times V_{SUP}$ $7.0\ V \le V_{SUP} \le 18\ V$; $t_{Bit} = 96\ µs$ $D3 = t_{BUS_REC(MIN)}/(2 \times t_{BIT})$	D3	0.417	–	–	
Duty Cycle 4: $TH_{REC(MIN)} = 0.389 \times V_{SUP}$ $TH_{DOM(MIN)} = 0.251 \times V_{SUP}$ $7.6\ V \le V_{SUP} \le 18\ V$; $t_{BIT} = 96\ µs$ $D4 = t_{BUS_REC(MAX)}/(2 \times t_{BIT})$	D4	–	–	0.590	
LIN Transmitter Timing, (V_{SUP} from 7.0 to 18 V) - See Figure 6					
Transmitter Symmetry $t_{TRAN_SYM} < MAX(ttran_sym60\%, t_{TRAN_SYM}40\%)$ $t_{TRAN_SYM}60\% = t_{TRAN_PDF}60\% - t_{TRAN_PDR}60\%$ $t_{TRAN_SYM}40\% = t_{TRAN_PDF}40\% - t_{TRAN_PDR}40\%$	t_{TRAN_SYM}	-7.25	0.0	7.25	µs

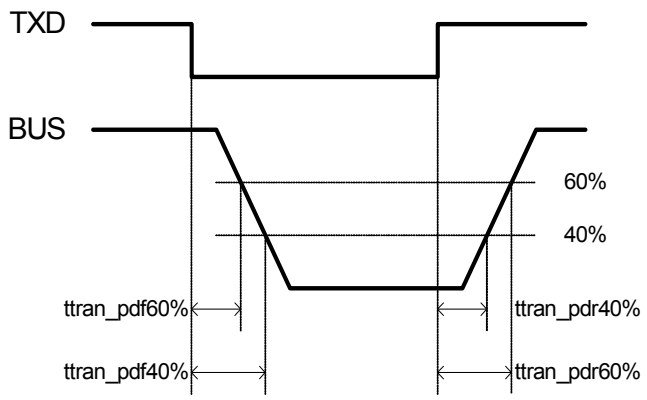


Figure 6. LIN Transmitter Timing

Table 29. Dynamic electrical characteristics - general purpose I/O - PTB4

Ratings	Symbol	Min.	Typ.	Max.	Unit
Wake-up filter time, positive edge	$t_{PTB4-PF}$	–	20	–	µs
Wake up filter time 1, negative edge	$t_{PTB4-NF1}$	350	700	1000	ns
Wake up filter time 2, negative edge	$t_{PTB4-NF2}$	0.8	1.6	2.2	us

Table 30. Dynamic electrical characteristics - general purpose I/O - PTB[1…3]

Parameter	Symbol	Min.	Typ.	Max.	Unit
GPIO Digital Frequency	f_{PTB}	–	–	5.0	MHz
Propagation Delay - Rising Edge[30]	t_{PDr}	–	–	20	ns
Rise Time - Rising Edge[30]	t_{RISE}	–	–	17.5	ns
Propagation Delay - Falling Edge[30]	t_{PDf}	–	–	20	ns
Rise Time - Falling Edge[30]	t_{FALL}	–	–	17.5	ns

Notes:
30.Load PTBx = 100 pF

Table 31. Dynamic electrical characteristics - current sense module

Parameter	Symbol	Min.	Typ.	Max.	Unit
Frequency Attenuation[31],[32] • <100 Hz (f_{PASS}) • >500 Hz (f_{STOP})		 – 40	 – –	 3.0 –	dB
Signal Update Rate[33]	$f_{IUPDATE}$	0.5	–	8.0	kHz
Signal Path Match with Voltage Channel [34]	$f_{IVMATCH}$	–	2.0	–	µs
Gain Change Duration (Automatic GCB active)[34]	t_{GC}	–	–	14	µs

Notes:
31.Characteristics identical to Voltage Sense Module
32.With default LPF coefficients
33.After passing decimation filter
34.Parameter not tested. Guaranteed by design and characterization.

Table 32. Dynamic electrical characteristics - voltage sense module

Parameter	Symbol	Min.	Typ.	Max.	Unit
Frequency attenuation[35],[36] • 95 to 105 Hz (f_{PASS}) • >500 Hz (f_{STOP})		 – 40	 – –	 3.0 –	dB
Signal update rate[37]	$f_{VUPDATE}$	0.5	–	8.0	kHz
Signal path match with Current Channel[38]	$f_{IVMATCH}$	–	2.0	–	µs

Notes:
35.Characteristics identical to Voltage Sense Module
36.With default LPF coefficients
37.After passing decimation filter
38.Parameter not tested. Guaranteed by design and characterization.

Table 33. Dynamic electrical characteristics - temperature sense module

Parameter	Symbol	Min.	Typ.	Max.	Unit
Signal Update Rate[39]	$f_{TUPDATE}$	1.0	–	4.0	kHz

Notes:
39.1.0 kHz with Chopper Enabled, 4.0 kHz with Chopper Disabled (fixed decimeter = 128)

4.6 S12ZI128 electrical characteristics

4.6.1 Electrical Characteristics

4.6.1.1 General

This supplement contains the most accurate electrical information for the MC9S12ZI-Family microcontroller available at the time of publication.

This introduction is intended to give an overview on several common topics like power supply, current injection etc.

4.6.1.2 Parameter classification

The electrical parameters shown in this supplement are guaranteed by various methods. To give the customer a better understanding the following classification is used and the parameters are tagged accordingly in the tables where appropriate.

Note

This classification is shown in the column labeled “C” in the parameter tables where appropriate.

- P: Those parameters are guaranteed during production testing on each individual device.
- C: Those parameters are achieved by the design characterization by measuring a statistically relevant sample size across process variations.
- T: Those parameters are achieved by design characterization on a small sample size from typical devices under typical conditions unless otherwise noted. All values shown in the typical column are within this category.
- D: Those parameters are derived mainly from simulations.

4.6.1.3 Power supply

The VDDD2D, VSSD2D/VSSD2D1 pin pair supplies the D2DI interface.

The VDDRX, VSSRX pin pair supplies the I/O pins.

VSSD2D & VSSD2D1 pins are internally connected by metal.

4.6.1.4 Pins

There are four groups of functional pins.

4.6.1.4.1 I/O Pins

The I/O pins have a level in the range of 3.13 V to 5.5 V. This class of pins is comprised of all port I/O pins, the analog inputs, BKGD and the $\overline{\text{RESET}}$ pins. Some functionality may be disabled.

4.6.1.4.2 D2DI interface

The pins D2DI interface pins have a level in the range of 2.5 V +/-5%.

4.6.1.4.3 Oscillator

The pins EXTAL, XTAL have a nominal 1.8 V level. Whenever the MCU is running on the internal clock, the oscillator pins may be used as I/O pins at a voltage level of 3.13 V to 5.5 V.

4.6.1.4.4 TEST

This pin is used for production testing only. The TEST pin must be tied to ground in all applications.

4.6.1.5 Current injection

Power supply must maintain regulation within V_{DDRX} operating range during instantaneous and operating maximum current conditions. If positive injection current ($V_{in} > V_{DDRX}$) is greater than I_{DDX}, the injection current may flow out of V_{DDRX} and could result in external power supply going out of regulation. Ensure external V_{DDRX} load will shunt current greater than maximum injection current. This will be the greatest risk when the MCU is not consuming power; e.g., if no system clock is present, or if clock rate is very low which would reduce overall power consumption.

4.6.1.6 Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only. A functional operation under or outside those maxima is not guaranteed. Stress beyond those limits may affect the reliability or cause permanent damage of the device.

This device contains circuitry protecting against damage due to high static voltage or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either V_{SSRX} or V_{DDRX}).

Table 34. Absolute maximum ratings[40]

Rating	Symbol	Min.	Max.	Unit
Digital logic and I/O supply voltage	V_{DDRX}	–0.3	5.55	V
D2DI interface supply voltage	V_{DDD2D}	–0.3	3.0	V
Digital I/O input voltage (PA0….PA7, PB0, PB1)	V_{IN}	–0.3	6.0	V
EXTAL, XTAL	V_{ILV}	–0.3	2.16	V
Instantaneous maximum current, Single pin limit for all digital I/O pins[41]	I_D	–25	+25	mA
Instantaneous maximum current, Single pin limit for EXTAL, XTAL	I_{DL}	–25	+25	mA
Storage temperature range	T_{stg}	–65	150	°C

Notes:
40.Beyond absolute maximum ratings device might be damaged.
41.All digital I/O pins are internally clamped to V_{SSRX} and V_{DDRX}.

4.6.1.7 ESD protection and latch-up immunity

All ESD testing is in conformity with CDF-AEC-Q100 stress test qualification for automotive grade integrated circuits. During the device qualification ESD stresses were performed for the Human Body Model (HBM) and the Charge Device Model.

A device will be defined as a failure if after exposure to ESD pulses the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature followed by hot temperature, unless specified otherwise in the device specification.

Table 35. ESD and latch-up test conditions

Model	Description	Symbol	Value	Unit
Human Body	Series Resistance	R1	1500	Ω
	Storage Capacitance	C	100	pF
	Number of Pulse per pin positive negative	-	- 3 3	
Machine	Series Resistance	R1	0	Ω
	Storage Capacitance	C	200	pF
	Number of Pulse per pin positive negative	-	- 3 3	
Latch-up	Minimum input voltage limit		-2.5	V
	Maximum input voltage limit		7.5	V

Table 36. ESD and latch-up protection characteristics

C	Rating	Symbol	Min.	Max.	Unit
C	Human Body Model (HBM)	V_{HBM}	2000	-	V
C	Machine Model (MM)	V_{MM}	200	-	V
C	Charge Device Model (CDM)	V_{CDM}	500	-	V
C	Charge Device Model (CDM) (Corner Pins)	V_{CDM}	750	-	V
C	Latch-up Current at 125 °C positive negative	I_{LAT}	 +100 -100	 - -	mA
C	Latch-up Current at 27 °C positive negative	I_{LAT}	 +200 -200	 - -	mA

4.6.1.8 Operating conditions

This section describes the operating conditions of the device. Unless otherwise noted those conditions apply to all the following data.

Table 37. Operating conditions

Rating	Symbol	Min.	Typ.	Max.	Unit
I/O and supply voltage	V_{DDRX}	3.13	5	5.5	V
D2DI interface supply voltage	V_{DDD2D}	2.375	2.5	2.625	V
Digital logic supply voltage	V_{DD}	1.72	1.8	1.98	V
Oscillator	f_{osc}	4	—	16.384	MHz
Bus frequency	f_{bus}	1	—	51.2	MHz
D2DI frequency[42]	f_{d2di}	—	—	25.6	MHz
Temperature Option M Operating junction temperature range Operating ambient temperature range	 T_J T_A	 –40 –40	 — 27	 150 125	°C

Notes:
42.V_{DDD2D} = 2.5 V +/-5%

Note

Operation is guaranteed when powering down until low voltage reset assertion.

4.6.1.9 I/O characteristics

This section describes the characteristics of all I/O pins except EXTAL, XTAL, TEST, D2DI, and supply pins.

Table 38. I/O characteristics

Conditions are 3.13 V < V_{DDRX}< 5.5 V junction temperature from –40 °C to +150 °C, unless otherwise noted. I/O Characteristics for all I/O pins except EXTAL, XTAL,TEST, and supply pins.

C	Rating	Symbol	Min.	Typ.	Max.	Unit
P	Input high voltage	V_{IH}	0.65*V_{DDRX}	—	—	V
T	Input high voltage	V_{IH}	—	—	V_{DDRX}+0.3	V
P	Input low voltage	V_{IL}	—	—	0.35*V_{DDRX}	V
T	Input low voltage	V_{IL}	V_{SSRX}–0.3	—	—	V
C	Input hysteresis	V_{HYS}	—	250	—	mV
P	Input leakage current (pins in high-impedance input mode)[43] V_{in} = V_{DDRX} or V_{SSRX} M temperature range –40 °C to +150 °C	I_{in}	 –1.00	 —	 1.00	µA

Table 38. I/O characteristics (continued).

Conditions are 3.13 V < V_{DDRX}< 5.5 V junction temperature from –40 °C to +150 °C, unless otherwise noted. I/O Characteristics for all I/O pins except EXTAL, XTAL,TEST, and supply pins.

C	Rating	Symbol	Min.	Typ.	Max.	Unit
P	Output high voltage (pins in output mode) I_{OH} = –4 mA	V_{OH}	V_{DDRX} – 0.8	—	—	V
C	Output low voltage (pins in output mode) I_{OL} = +4mA	V_{OL}	—	—	0.8	V
D	Input capacitance	C_{IN}	—	7.0	—	pF
T	Injection current(44) Single pin limit Total device Limit, sum of all injected currents	 I_{ICS} I_{ICP}	 –2.5 –25	 — —	 2.5 25	mA

Notes:

43.Maximum leakage current occurs at maximum operating temperature. Current decreases by approximately one-half for each 8.0 °C to 12 °C in the temperature range from 50 °C to 125 °C.

44.Refer to Current injection" for more details

4.6.2 NVM electrical parameters

4.6.2.1 NVM timing parameters

The time base for all NVM program or erase operations is derived from the bus clock using the FCLKDIV register. The frequency of this derived clock must be set within the limits specified as f_{NVMOP}. The NVM module does not have any means to monitor the frequency and will not prevent program or erase operation at frequencies above or below the specified minimum. When attempting to program or erase the NVM module at a lower frequency, a full program or erase transition is not assured.

Table 39 provides the time required to execute specific flash commands. All timing parameters are a function of the bus clock frequency, f_{NVMBUS}. All program and erase times are also a function of the NVM operating frequency, f_{NVMOP}.

Table 39. NVM timing characteristics

Num	Command	f_{NVMOP} cycle	f_{NVMBUS} cycle	Symbol	Min.(45)	Typ.(46)	Max.(47)	Lfmax (48)	Unit
1	Bus frequency	1		f_{NVMBUS}	1	50	51.2		MHz
2	Operating frequency		1	f_{NVMOP}	0.8	1.0	1.05		MHz
3	Erase Verify All Blocks (49),(50)	0	35384	t_{RD1ALL}	0.71	0.71	1.42	70.77	ms
4	Erase Verify Block (Pflash) (49)	0	33337	t_{RD1BLK_P}	0.67	0.67	1.33	66.67	ms
5	Erase Verify Block (EEPROM) (50)	0	2573	t_{RD1BLK_D}	0.05	0.05	0.10	5.15	ms
6	Erase Verify P-Flash Section	0	505	t_{RD1SEC}	0.01	0.01	0.02	1.01	ms
7	Read Once	0	481	t_{RDONCE}	9.62	9.62	9.62	481.00	us
8	Program P-Flash (4 Word)	164	3077	t_{PGM_4}	0.22	0.23	0.41	12.51	ms
9	Program Once	164	3054	$t_{PGMONCE}$	0.22	0.23	0.23	3.26	ms
10	Erase All Blocks (49),(50)	100066	35773	t_{ERSALL}	96.02	100.78	101.50	196.63	ms
11	Erase Flash Block (Pflash) (49)	100060	33596	t_{ERSBLK_P}	95.97	100.73	101.40	192.27	ms
12	Erase Flash Block (EEPROM) (50)	100060	2895	t_{ERSBLK_D}	95.35	100.12	100.18	130.87	ms
13	Erase P-Flash Sector	20015	914	t_{ERSPG}	19.08	20.03	20.05	26.85	ms
14	Unsecure Flash	100066	35838	t_{UNSECU}	96.02	100.78	101.50	196.76	ms
15	Verify Backdoor Access Key	0	515	t_{VFYKEY}	10.30	10.30	10.30	515.00	us
16	Set User Margin Level	0	427	t_{MLOADU}	8.54	8.54	8.54	427.00	us
17	Set Factory Margin Level	0	436	t_{MLOADF}	8.72	8.72	8.72	436.00	us
18	Erase Verify EEPROM Section	0	583	$t_{DRD1SEC}$	0.01	0.01	0.02	1.17	ms
19	Program EEPROM (1 Word)	68	1657	t_{DPGM_1}	0.10	0.10	0.20	6.71	ms

Table 39. NVM timing characteristics

Num	Command	f_{NVMOP} cycle	f_{NVMBUS} cycle	Symbol	Min.(45)	Typ.(46)	Max.(47)	Lfmax (48)	Unit
20	Program EEPROM (2 Word)	136	2660	t_{DPGM_2}	0.18	0.19	0.35	10.81	ms
21	Program EEPROM (3 Word)	204	3663	t_{DPGM_3}	0.27	0.28	0.50	14.91	ms
22	Program EEPROM (4 Word)	272	4666	t_{DPGM_4}	0.35	0.37	0.65	19.00	ms
23	Erase EEPROM Sector	5015	810	t_{DERSPG}	4.79	5.03	20.34	38.85	ms
24	Protection Override	0	475	$t_{PRTOVRD}$	9.50	9.50	9.50	475.00	us

Notes:
45.Minimum times are based on maximum f_{NVMOP} and maximum f_{NVMBUS}
46.Typical times are based on typical f_{NVMOP} and typical f_{NVMBUS}
47.Maximum times are based on typical f_{NVMOP} and typical f_{NVMBUS} plus aging
48.Lowest-frequency max times are based on minimum f_{NVMOP} and minimum f_{NVMBUS} plus aging
49.Affected by Pflash size
50.Affected by EEPROM size

4.6.2.2 NVM reliability parameters

The reliability of the NVM blocks is guaranteed by stress test during qualification, constant process monitors and burn-in to screen early life failures.

The data retention and program/erase cycling failure rates are specified at the operating conditions noted. The program/erase cycle count on the sector is incremented every time a sector or mass erase event is executed.

Note

All values shown in Table 40 are preliminary and subject to further characterization.

Table 40. NVM reliability characteristics

NUM	C	Rating	Symbol	Min.	Typ.	Max.	Unit
Program Flash Arrays							
	C	Data retention at an average junction temperature of T_{JAVG} = 85 °C (51) after up to 10,000 program/erase cycles	t_{NVMRET}	20	100(52)	—	Years
	C	Program Flash number of program/erase cycles (-40 °C ≤ T_J ≤ 150 °C)	n_{FLPE}	10K	100K(53)	—	Cycles
EEPROM Array							
	C	Data retention at an average junction temperature of T_{JAVG} = 85 °C (51) after up to 100,000 program/erase cycles	t_{NVMRET}	5	100(52)	—	Years
	C	Data retention at an average junction temperature of T_{JAVG} = 85 °C (51) after up to 10,000 program/erase cycles	t_{NVMRET}	10	100(52)	—	Years
	C	Data retention at an average junction temperature of T_{JAVG} = 85 °C (51) after less than 100 program/erase cycles	t_{NVMRET}	20	100(52)	—	Years
	C	EEPROM number of program/erase cycles (-40 °C ≤ T_J ≤ 150 °C)	n_{FLPE}	100 k	500 k(53)	—	Cycles

Notes:
51.T_{JAVG} does not exceed 85 °C in a typical temperature profile over the lifetime of a consumer, industrial or automotive application.
52.Typical data retention values are based on intrinsic capability of the technology measured at high temperature and de-rated to 25 °C using the Arrhenius equation. For additional information on how NXP defines Typical Data Retention, refer to Engineering Bulletin EB618
53.Spec table quotes typical endurance evaluated at 25 °C for this product family. For additional information on how NXP defines Typical Endurance, refer to Engineering Bulletin EB619.

4.6.3 Electrical specification for voltage regulator

Table 41. ivregcrz_ll18 characteristics

Num	C	Characteristic	Symbol	Min.	Typ.	Max.	Unit
	P	Input Voltages	V_{VDDRX}	3.13	—	5.5	V
	P	V_{DDRX} Low Voltage Interrupt Assert Level (54) V_{DDRX} Low Voltage Interrupt Deassert Level	V_{LVIA} V_{LVID}	4.04 4.19	4.23 4.38	4.47 4.60	V V
	P	V_{DDRX} Low Voltage Reset Deassert (55), (56)	V_{LVRXD}	—	—	3.25	V
	T	CPMU ACLK frequency (CPMUACLKTR[5:0] = %000000)	f_{ACLK}	—	20	—	kHz
	C	Trimmed ACLK internal clock(57) $\Delta f / f_{NOMINAl}$	df_{ACLK}	- 5.0	—	+ 5.0	%
	D	The first period after enabling the counter by APIFE might be reduced by ACLK start up delay	t_{sdel}	—	—	100	us
	D	The first period after enabling the COP might be reduced by ACLK start up delay	t_{SDEL}	—	—	100	us
		VDDRX Low Voltage Reset(58) Assert Level Deassert Level	 V_{LVRXA} V_{LVRXD}	 2.906 —	 3.037 3.054	 — 3.25	 V V

Notes:
54.Monitors VDDA, active only in Full Performance mode. Indicates I/O & ADC performance degradation due to low supply voltage.
55.Device functionality is guaranteed on power down to the LVR assert level
56.Monitors VDDRX, active only in Full Performance mode. MCU is monitored by the POR in RPM (see Figure 7)
57.The ACLK Trimming CPMUACLKTR[5:0] bits must be set so that f_{ACLK}=10KHz.
58.Monitors VDDRX, active only in Full Performance mode. V_{LVRA} and V_{PORD} must overlap (see Figure 7)

Note

The LVR monitors the voltages V_{DD}, V_{DDF}, and V_{DDRX}. As soon as voltage drops on these supplies which would prohibit the correct function of the microcontroller, the LVR is triggering a reset.

4.6.3.1 Chip power-up and voltage drops

LVI (low voltage interrupt), POR (power-on reset) and LVRs (low voltage reset) handle chip power-up or drops of the supply voltage.

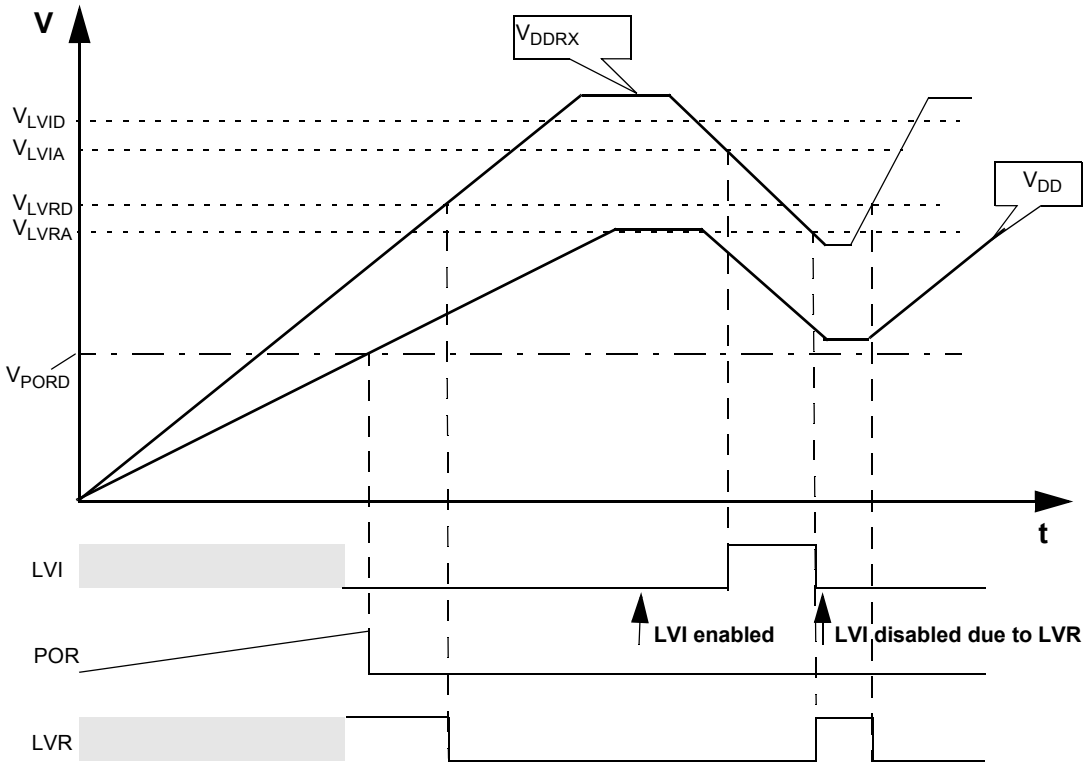


Figure 7. Chip power-up and voltage drops OSCLCPcr electrical specifications

4.6.4 OSCLCPcr electrical specifications

Table 42. XOSCLCP characteristics

Conditions are shown in Table 37 unless otherwise noted

Num	C	Rating	Symbol	Min.	Typ.	Max.	Unit
1	C	Nominal crystal or resonator frequency	f_{OSC}	4.0	—	16.384	MHz
2	P	Startup Current	i_{OSC}	100	—	—	µA
3a	C	Oscillator start-up time (4.0 MHz) (59)	t_{UPOSC}	—	2.0	10	ms
3b	C	Oscillator start-up time (8.0 MHz) (59)	t_{UPOSC}	—	1.6	8.0	ms
3c	C	Oscillator start-up time (16 MHz)(59)	t_{UPOSC}	—	1.0	5	ms
4	P	Clock Monitor Failure Assert Frequency	f_{CMFA}	200	450	1200	KHz
5	D	Input Capacitance (EXTAL, XTAL pins)	C_{IN}	—	7.0	—	pF
6	C	EXTAL Pin Input Hysteresis	$V_{HYS,EXTAL}$	—	120	—	mV
7	C	EXTAL Pin oscillation amplitude (loop controlled Pierce)	$V_{PP,EXTAL}$	—	1.0	—	V
7	C	EXTAL Pin oscillation amplitude (full swing Pierce)	$V_{PP,EXTAL}$	—	1.8	—	V
8	D	EXTAL Pin oscillation required amplitude (60)	$V_{PP,EXTAL}$	0.8	—	—	V

Notes:
59.These values apply for carefully designed PCB layouts with capacitors that match the crystal/resonator requirements.
60.Needs to be measured at room temperature on the application board using a probe with very low (≤5.0 pF) input capacitance.

4.6.5 PLL electrical specifications

4.6.5.1 Reset, oscillator and PLL

4.6.5.2 Phase locked loop

4.6.5.2.1 Jitter information

With each transition of the feedback clock, the deviation from the reference clock is measured and the input voltage to the VCO is adjusted accordingly.The adjustment is done continuously with no abrupt changes in the VCOCLK frequency. Noise, voltage, temperature and other factors cause slight variations in the control loop resulting in a clock jitter. This jitter affects the real minimum and maximum clock periods as illustrated in Figure 8.

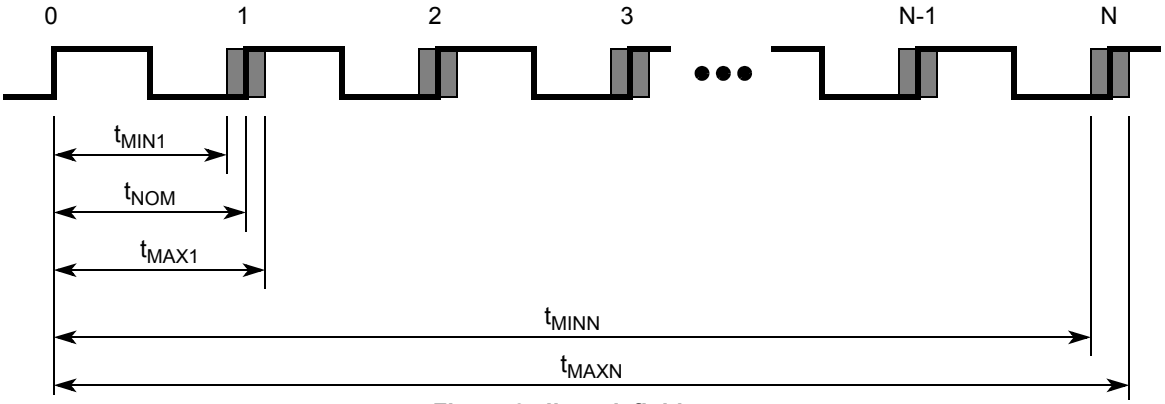


Figure 8. Jitter definitions

The relative deviation of t_{NOM} is at its maximum for one clock period, and decreases towards zero for larger number of clock periods (N).

Defining the jitter as:

The following equation is a good fit for the maximum jitter:

$$J(N) = max\left(\left|1 - \frac{t_{max}(N)}{N \cdot t_{nom}}\right|, \left|1 - \frac{t_{min}(N)}{N \cdot t_{nom}}\right|\right)$$

$$J(N) = \frac{j_1}{\sqrt{N}}$$

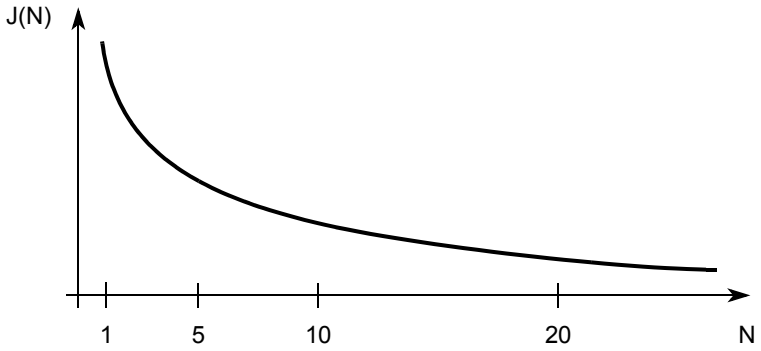


Figure 9. Maximum bus clock jitter approximation

Note

On timers and serial modules a prescaler will eliminate the effect of the jitter to a large extent.

Table 43. ipll_1vdd_ll18 characteristics

Conditions are 4.5 V < V_{DDRX} < 5.5 V junction temperature from –40 °C to +150 °C, unless otherwise noted

Num	C	Rating	Symbol	Min.	Typ.	Max.	Unit
1	D	VCO frequency during system reset	f_{VCORST}	8.0	—	32	MHz
2	C	VCO locking range	f_{VCO}	32	—	100	MHz
3	C	Reference clock	f_{REF}	1.0	—	—	MHz
4	D	Lock detection	$\|\Delta_{LOCK}\|$	0.0	—	1.5	%(61)
5	D	Unlock detection	$\|\Delta_{UNL}\|$	0.5	—	2.5	%(61)
7	C	Time to lock	t_{LOCK}	—	—	150 + 256/f_{REF}	µs
8	C	Jitter fit parameter 1(62)	j_1	—	—	2.0	%

Notes:
61.% deviation from target frequency
62.f_{REF} = 1.0 MHz, f_{BUS} = 50 MHz

4.6.6 IRC electrical specifications

Table 44. IRC electrical characteristics

Num	C	Rating	Symbol	Min.	Typ.	Max.	Unit
1	P	Junction Temperature - 40 to 150 Celsius Internal Reference Frequency, factory trimmed	f_{IRC1M_TRIM}	1.010	1.024	1.038	MHz
2	P	Internal Clock Frequency Tolerance -40 °C ≤ T_A ≤ 85 °C (63)	f_{TOL}	-1.0	–	1.0	%
3	C	Internal Clock Frequency Tolerance 85 °C< T_A ≤ 125 °C (63)	f_{TOL}	-1.3	–	1.3	%
4		Clock Frequency Tolerance with External Oscillator (64)	f_{TOLEXT}	-0.5	–	0.5	%

Notes:
63.Target is 1.024 MHz
64.Dependent on the external OSC

4.6.7 SPI electrical specifications

This section provides electrical parameters and ratings for the SPI.

Measurement conditions are listed in Table 45.

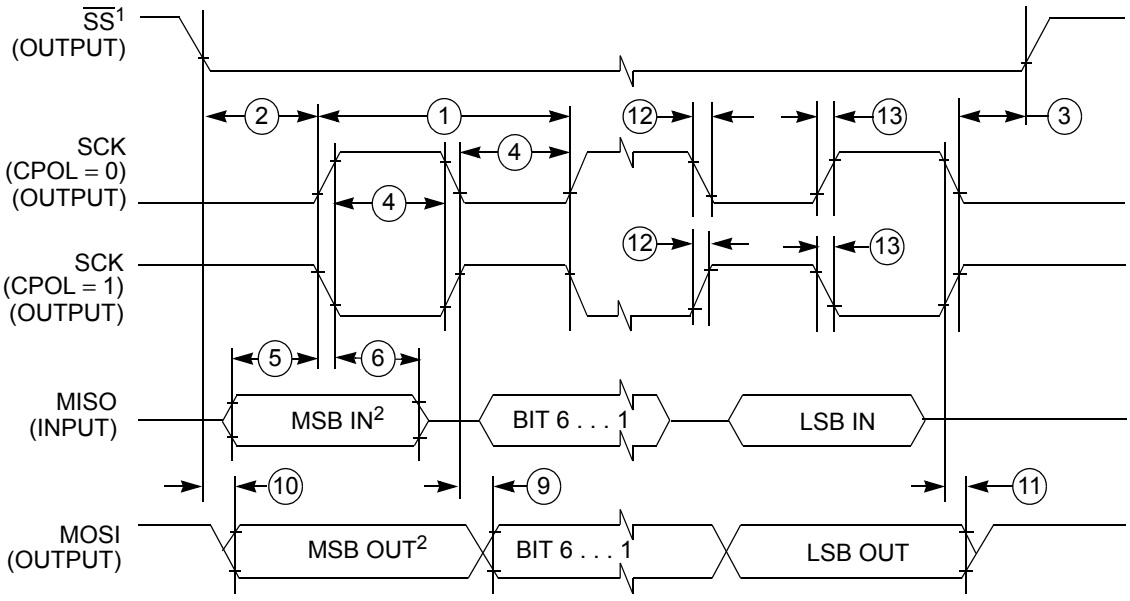
Table 45. Measurement conditions

Description	Value	Unit
Drive mode	full drive mode	—
Load capacitance C_{LOAD}(65) on all outputs	50	pF
Thresholds for delay measurement points	(20% / 80%) VDDRX	V

Notes:
65.Timing specified for equal load on all SPI output pins. Avoid asymmetric load.

4.6.7.1 Master mode

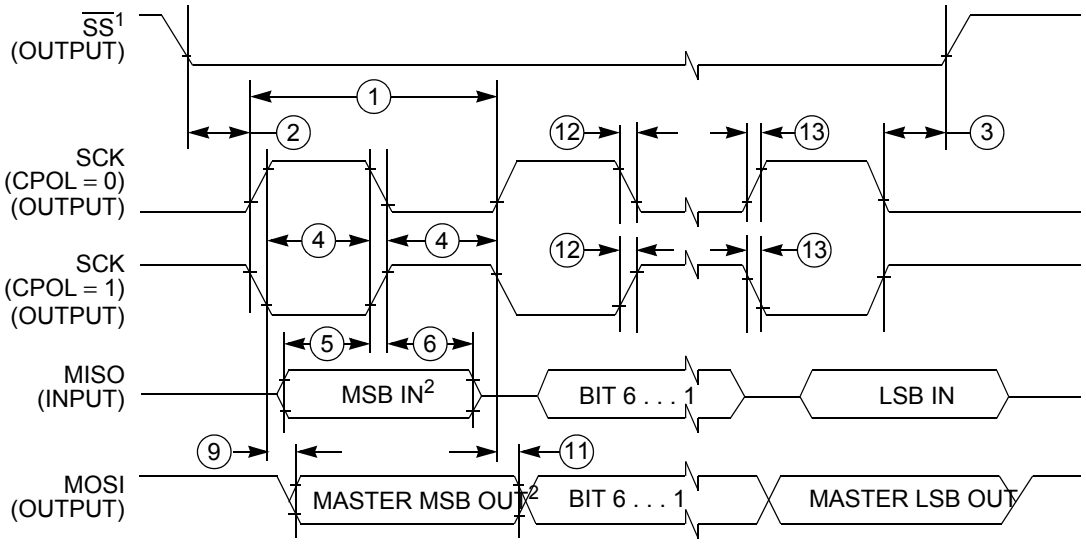
The timing diagram for master mode with transmission format CPHA=0 is depicted in Figure 10.



1. If enabled.
2. LSBFE = 0. For LSBFE = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure 10. SPI master timing (CPHA=0)

The timing diagram for master mode with transmission format CPHA=1 is depicted in Figure 11.



1. If enabled.
2. LSBFE = 0. For LSBFE = 1, bit order is LSB, bit 1, ..., bit 6, MSB.

Figure 11. SPI master timing (CPHA=1)

Timing characteristics for master mode are listed in Table 46.

Table 46. SPI master mode timing characteristics

Num	Characteristic	Symbol	Min.	Typ.	Max.	Unit
1	SCK Frequency	f_{SCK}	1/2048	—	1/2 (66)	f_{BUS}
1	SCK Period	t_{SCK}	2.0 (66)	—	2048	t_{BUS}
2	Enable Lead Time	t_{LEAD}	—	1/2	—	t_{SCK}
3	Enable Lag Time	t_{LAG}	—	1/2	—	t_{SCK}
4	Clock (SCK) High or Low Time	t_{WSCK}	—	1/2	—	t_{SCK}
5	Data Setup Time (Inputs)	t_{SU}	8.0	—	—	ns
6	Data Hold Time (Inputs)	t_{HI}	8.0	—	—	ns
9	Data Valid after SCK Edge	t_{VSCK}	—	—	15	ns
10	Data Valid after $\overline{SS}$ fall (CPHA=0)	t_{VSS}	—	—	15	ns
11	Data Hold Time (Outputs)	t_{HO}	0.0	—	—	ns
12	Rise and Fall Time Inputs	t_{RFI}	—	—	8.0	ns
13	Rise and Fall Time Outputs	t_{RFO}	—	—	8.0	ns

Notes:
66.See Figure 12

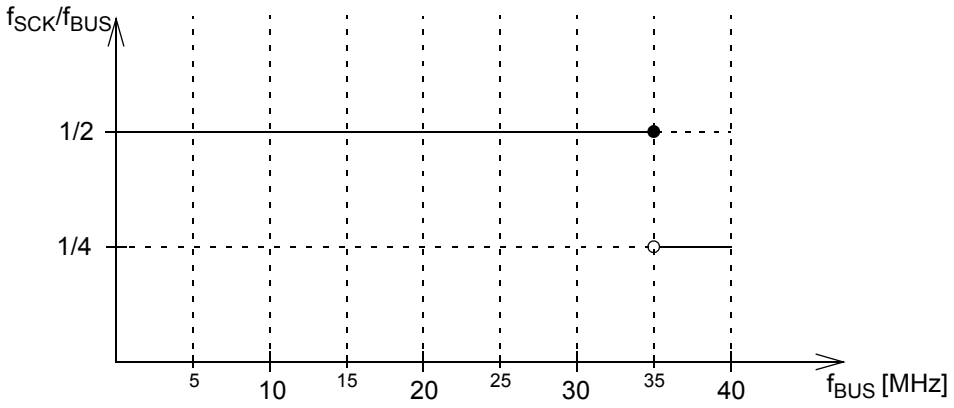


Figure 12. Derating of maximum f_{SCK} to f_{BUS} ratio in master mode

In Master Mode the allowed maximum f_{SCK} to f_{BUS} ratio (= minimum Baud Rate Divisor. See SPI Block Guide) derates with increasing f_{BUS}, see Figure 12.

4.6.7.2 Slave mode

The timing diagram for slave mode with transmission format CPHA=0 is depicted in Figure 13.

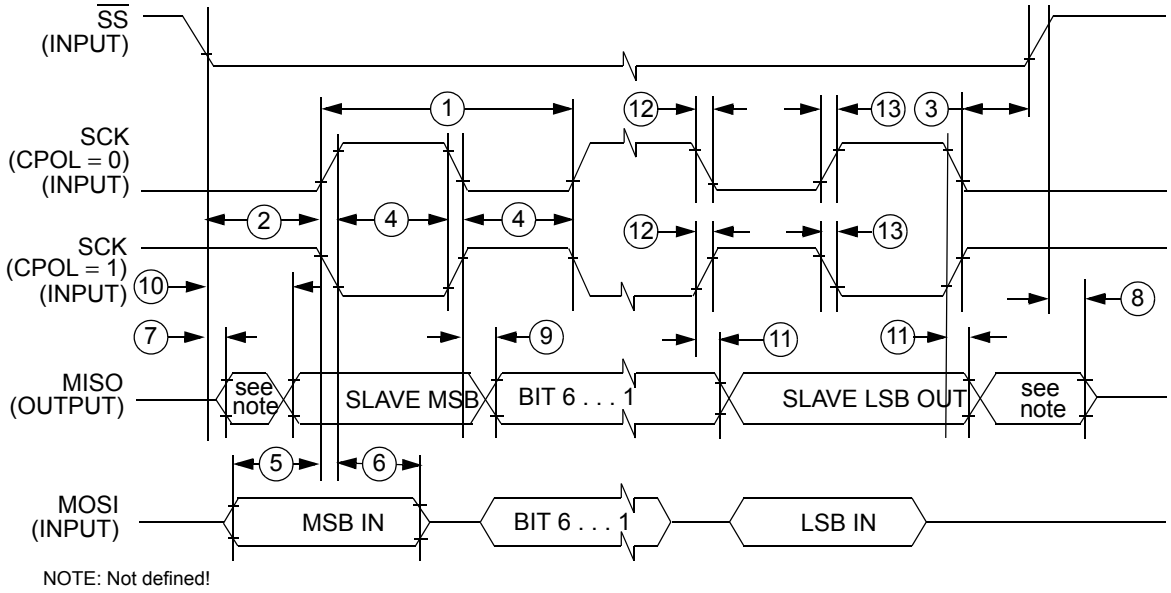


Figure 13. SPI slave timing (CPHA=0)

The timing diagram for slave mode with transmission format CPHA=1 is depicted in Figure 14.

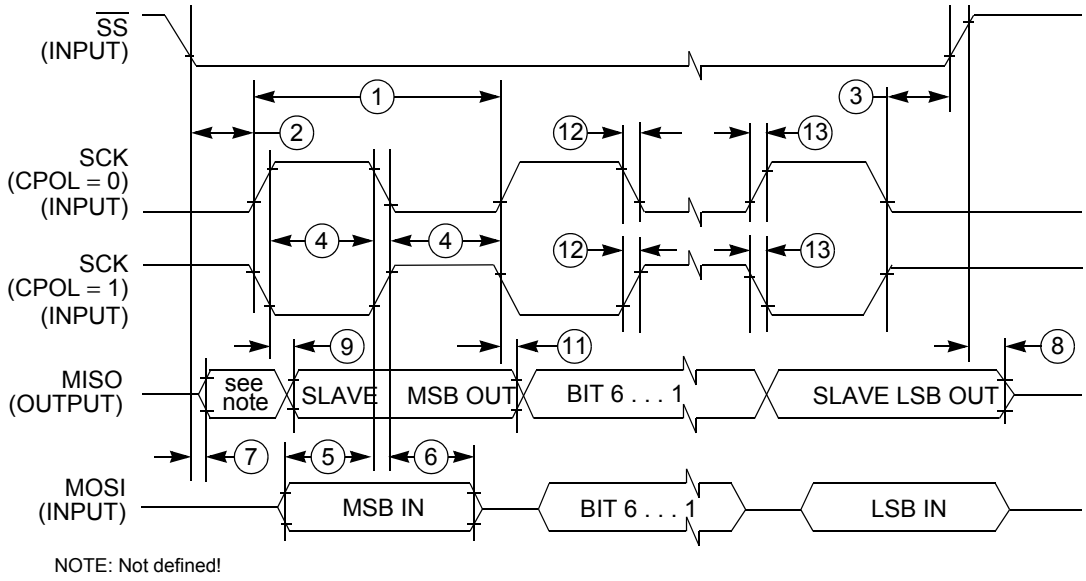


Figure 14. SPI slave timing (CPHA=1)

The timing characteristics for slave mode are listed In Table 47.

Table 47. SPI slave mode timing characteristics

Num	Characteristic	Symbol	Min.	Typ.	Max.	Unit
1	SCK Frequency	f_{SCK}	DC	—	1/4	f_{BUS}
1	SCK Period	t_{SCK}	4.0	—	∞	t_{BUS}
2	Enable Lead Time	t_{LEAD}	4.0	—	—	t_{BUS}
3	Enable Lag Time	t_{LAG}	4.0	—	—	t_{BUS}
4	Clock (SCK) High or Low Time	t_{WSCK}	4.0	—	—	t_{BUS}
5	Data Setup Time (Inputs)	t_{SU}	8.0	—	—	ns
6	Data Hold Time (Inputs)	t_{HI}	8.0	—	—	ns
7	Slave Access Time (time to data active)	t_A	—	—	20	ns
8	Slave MISO Disable Time	t_{DIS}	—	—	22	ns
9	Data Valid after SCK Edge	t_{VSCK}	—	—	28 + 0.5 * t_{BUS} (67)	ns
10	Data Valid after $\overline{SS}$ fall	t_{VSS}	—	—	28 + 0.5 * t_{BUS} (67)	ns
11	Data Hold Time (Outputs)	t_{HO}	20	—	—	ns
12	Rise and Fall Time Inputs	t_{RFI}	—	—	8.0	ns
13	Rise and Fall Time Outputs	t_{RFO}	—	—	8.0	ns

Notes:
67.0.5t_{BUS} added due to internal synchronization delay

4.7 Thermal protection characteristics

Characteristics noted under conditions 3.5 V ≤ V_{SUP} ≤ 28 V, -40 °C ≤ T_A ≤ 125 °C, unless otherwise noted. Typical values noted reflect the approximate parameter mean at T_A = 25 °C under nominal conditions, unless otherwise noted.

Table 48. Thermal Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
VDDH/VDDA/VDDX High Temperature Warning (HTI) • Threshold • Hysteresis	 T_{HTI} T_{HTI_H}	 125 –	 – 10	 – –	°C
VDDH/VDDA/VDDX Overtemperature Shutdown • Threshold • Hysteresis	 T_{SD} T_{SD_H}	 160 –	 – 10	 –	°C
LIN Overtemperature Shutdown	T_{LINSD}	150	165	180	°C
LIN Overtemperature Shutdown Hysteresis	T_{LINSD_HYS}	–	20		°C

4.8 Electromagnetic compatibility (EMC)

All ESD testing is in conformity with the CDF-AEC-Q100 stress test qualification for automotive grade integrated circuits. During the device qualification, ESD stresses are performed for the Human Body Model (HBM), Machine Model (MM), Charge Device Model (CDM), as well as LIN transceiver specific specifications.

A device will be defined as a failure, if after exposure to ESD pulses, the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature, followed by hot temperature, unless specified otherwise in the device specification.

The immunity against transients for the LIN, PTB4, VSENSEX, ISENSEH, ISENSEL, and VSUP, is specified according to the LIN Conformance Test Specification - Section LIN EMC Test Specification (ISO7637-2), refer to the LIN Conformance Test Certification Report - available as separate document.

Table 49. Electromagnetic compatibility

Ratings	Symbol	Value / limit	Unit
ESD - Human Body Model (HBM) following AEC-Q100 / JESD22-A114 (C_{ZAP} = 100 pF, R_{ZAP} = 1500 Ω) • LIN (all GNDs shorted) • All other Pins	V_{HBM}	 ±8.0 ±2.0	kV
ESD - Charged Device Model (CDM) following AEC-Q100 • Corner Pins • All other Pins	V_{CDM}	±750 ±500	V
ESD - Machine Model (MM) following AEC-Q100 (C_{ZAP} = 200 pF, R_{ZAP} = 0 Ω), All Pins	V_{MM}	±200	V
Latch-up current at T_A = 125 °C[68]	I_{LAT}	±100	mA
ESD GUN - LIN Conformance Test Specification[69], unpowered, contact discharge. (C_{ZAP}= 150 pF, R_{ZAP} = 330 Ω); LIN (no bus filter C_{BUS}) Vsup with Cvsup capacitors (4.7 µF and 100 nF) Vsense[3..0] with serial 2.2 kΩ PTB4 with serial 47 kΩ and 2.2 nF (external wake-up configuration) PTB[4..0] with serial 2.2 kΩ and 2.2 nF to gnd (external Tsense configuration) VDDA with 100 kΩ and 47 nF to gnd PTB5 with serial 1.0 kΩ, 2.2 nF and 5.1 V zener diode		± 8.0	kV
ESD GUN - IEC 61000-4-2 Test Specification[70], unpowered, contact discharge. (C_{ZAP}= 150 pF, R_{ZAP} = 330 Ω); LIN (no bus filter C_{BUS}) Vsup with Cvsup capacitors (4.7 µF and 100 nF) Vsense[3..0] with serial 2.2 kΩ PTB4 with serial 47 kΩ and 2.2 nF (external wake-up configuration) PTB[4..0] with serial 2.2 kΩ and 2.2 nF to gnd (external Tsense configuration) VDDA with 100 kΩ and 47 nF to gnd PTB5 with serial 1.0 kΩ, 2.2 nF and 5.1 V zener diode		± 8.0	kV
ESD GUN - ISO10605[70], unpowered, contact discharge, C_{ZAP}= 150 pF, R_{ZAP} = 2.0 kΩ; LIN (no bus filter C_{BUS}) Vsup with Cvsup capacitors (4.7 µF and 100 nF) Vsense[3..0] with serial 2.2 kΩ PTB4 with serial 47 kΩ and 2.2 nF (external wake-up configuration) PTB[4..0] with serial 2.2 kΩ and 2.2 nF to gnd (external Tsense configuration) VDDA with 100 kΩ and 47 nF to gnd PTB5 with serial 1.0 kΩ, 2.2 nF and 5.1 V zener diode		± 8.0	kV
ESD GUN - ISO10605[70], powered, contact discharge, C_{ZAP}= 330 pF, R_{ZAP} = 2.0 kΩ; LIN (with bus filter C_{BUS} 220 pF) Vsup with Cvsup capacitors (4.7 µF and 100 nF) Vsense[3..0] with serial 2.2 kΩ PTB4 with serial 47 kΩ and 2.2 nF (external wake-up configuration) PTB[4..0] with serial 2.2 kΩ and 2.2 nF to gnd (external Tsense configuration) VDDA with 100 kΩ and 47 nF to gnd PTB5 with serial 1k.0 Ω, 2.2 nF and 5.1 V zener diode		± 8.0	kV

Notes:
68.Input Voltage Limit = -2.5 to 7.5 V
69.Tested internally. Certification pending.
70.Tested internally only, following the reference document test procedure.

5 MM9Z1_638 overview

The MM9Z1_638 (as shown in Figure 15) contains the following features:

Battery Voltage Measurement

- Dedicated 16-bit second order ΣΔ ADC
- Simultaneous sampling with current channel
- Programmable signal filtering shared with current measurement
- Four battery voltage measurement w/ internal divider, full device measurement range 3.5 to 52 V
- Five voltage sensor inputs routable to both voltage and temperature channel

Differential battery current measurement

- Dedicated 16-bit second order ΣΔ ADC, with a programmable gain amplifier with four programmable gain factors (24-bit result)
- Gain control block for automatic gain adjustment
- Measurement range up to ±2000 A with 100 μΩ shunt resistor

Temperature Measurement

- Internal, on-chip temperature sensor
- Dedicated 16-bit ADC with anti-aliasing filter
- Five single ended sensor inputs routable to both voltage and temperature channels
- Internal supply for external sensors

Normal and Low Power Mode

- Current integration via 32-bit accumulator during low-power mode
- Programmable current threshold detection during low-power mode
- Programmable wake-up timer, triggered wake-up from LIN

Advanced system level management

- Internal oscillator
- Communication via LIN 2.1, LIN 2.0 interface with fast mode for Flash programming over LIN
- An msCAN protocol controller, with TxD and RxD pins, and bus wake-up detection
- S12Z micro controller with 96/128 kByte Flash, 8.0 kByte RAM, 4.0 kByte EEPROM all with ECC
- Enhanced VDDX capability to supply MCU and external components
- Fast, die-to-die bus interface with transparent integration of analog IC registers into the MCU register map, automatic synchronization, and error detection
- Automotive EMC and ESD performance

Operating conditions

- Ambient operating temperature: -40 °C < T_A < 125 °C
- Junction operating temperature: -40 °C < T_J < 150 °C

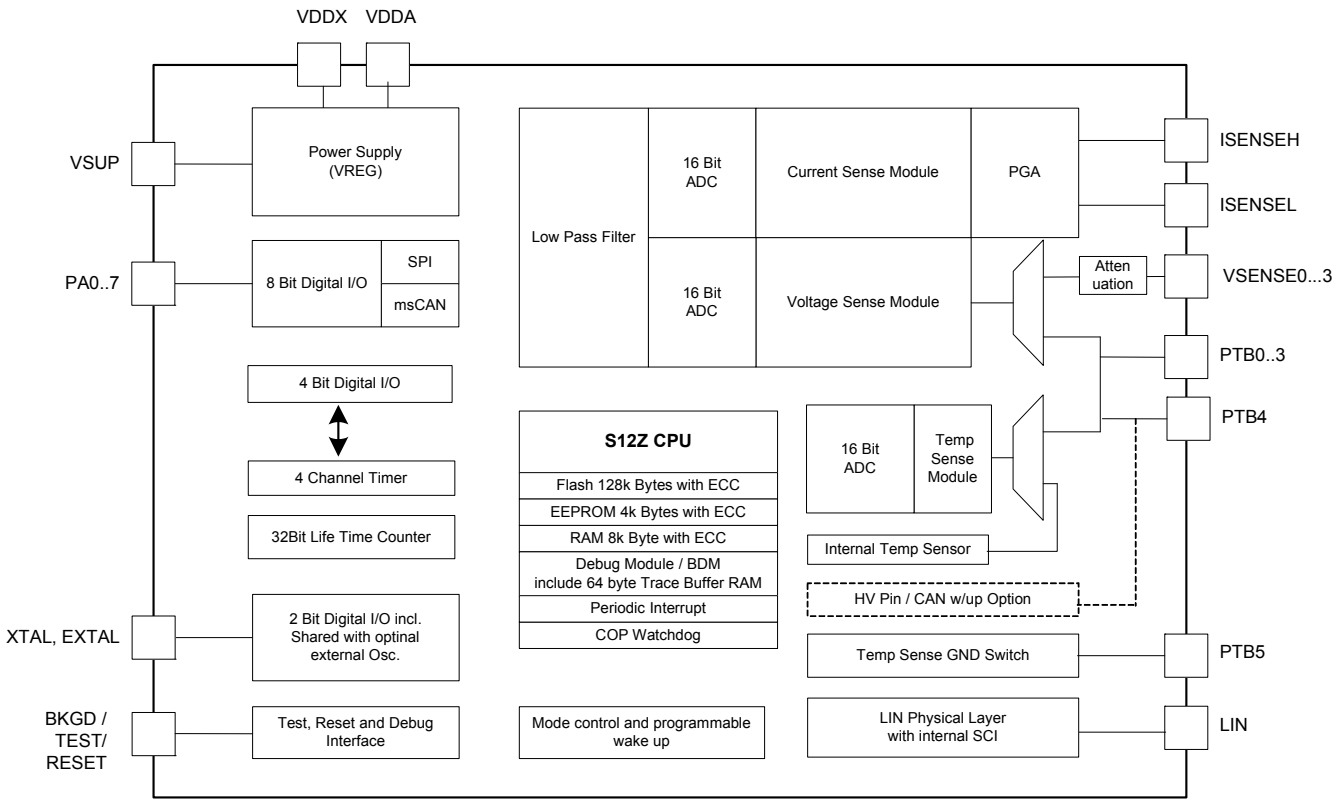


Figure 15. Simplified block diagram

5.1 MM9Z1_638 analog die overview

5.1.1 Introduction

The MM9Z1_638 analog die implements all system base functions to operate the integrated micro controller, and delivers applications specific input capturing.

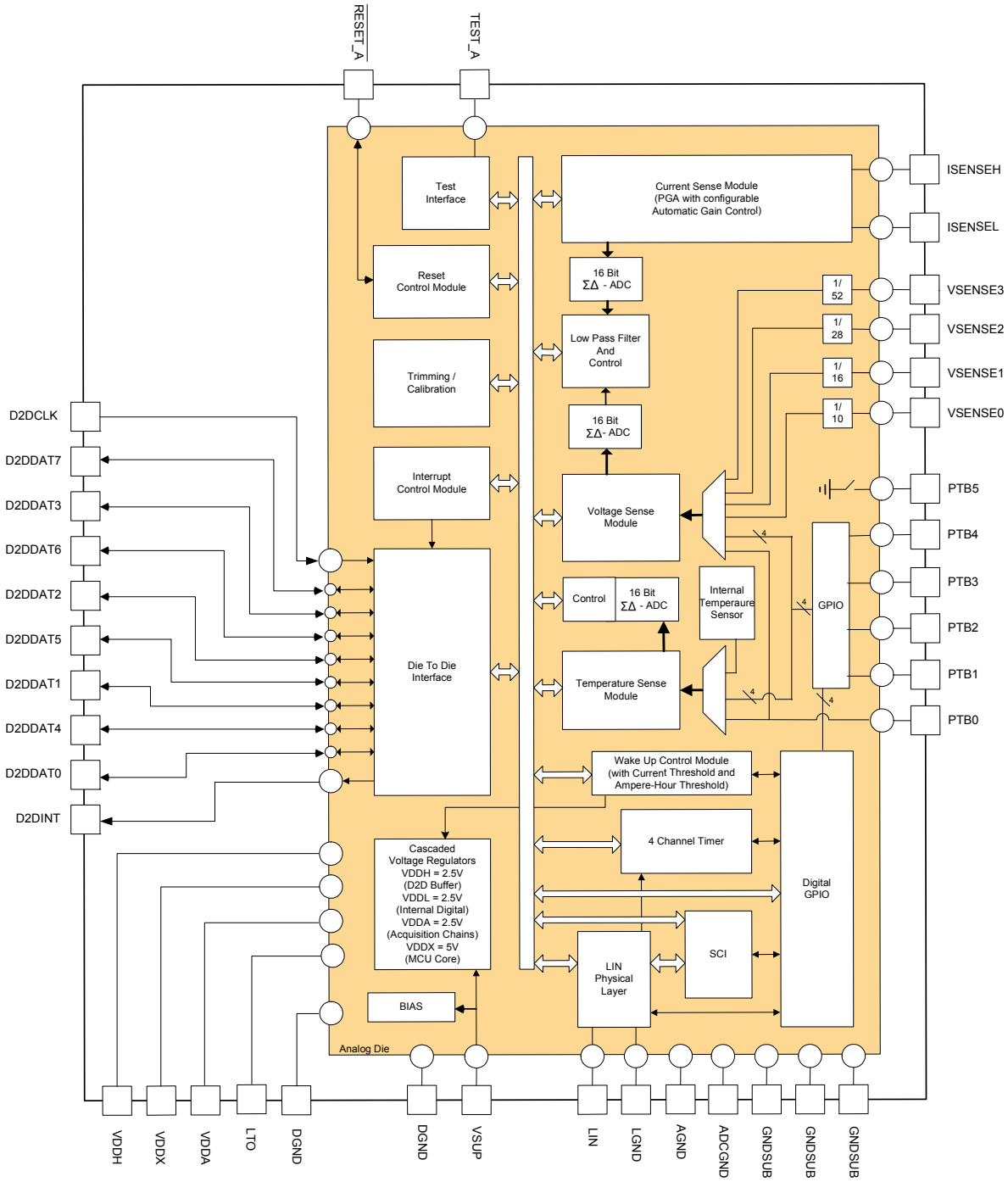


Figure 16. Analog die block overview

5.1.2 Features

The analog die contains the following features:

5.1.2.1 Three sigma delta analog to digital converters

5.1.2.1.1 Current sensing

One 16-bit SD ADC for current sense. The input of this SD ADC is connected to the output of a PGA (Programmable Gain Amplifier) with 4 different gains.This SD converter operates in Normal mode, but can also operate in Low Power mode. The PGA gain selection can be either user selectable or automatically selected. The user can choose which gains will be used by the automatic selection. The converted value available in the output register has a fixed LSB value, independent of the selected gain. The output value is a signed value available on a 24-bit range register. The external shunt value can be selected from one of the following values, 50, 75, 100, 150, and 200 μΩ.

5.1.2.1.2 Voltage sensing

One 16-bit SD ADC for voltage measurements. The input of this SD ADC is connected to the output of a multiplexer, and allows selection of voltage sense with an internal resistor divider, VSENSE0 to VSENSE3, or direct voltage sense, PTB0 to PTB4. PTB inputs allow usage of a user defined resistor divider, and a negative measurement range. Each VSENSE has a different resistor divider ratio, allowing a large range of voltage sense.

- Synchronization between the voltage and current SD ADCs
- Configurable internal hardware filters for the current and voltage SD ADC.

5.1.2.1.3 Temperature sensing

One 16-bit SD ADC for temperature measurement. The input of this SD ADC is connected to the output of a multiplexer, and allows selection of internal temperature sensor or external temperature sensors via the direct voltage sense, PTB0 to PTB4. SD ADC measures the voltage between PTB5 and one of the following PTB0 to PTB4.

- Fixed hardware filter for the temperature SD

Each of the three SD ADCs has its own set of registers for offset and gain compensations. The user can access and use these to enhance system performance, taking into account external components.

5.1.2.2 5 GPIO pins, named PTB0 to PTB5

The PTB0 to PTB4 pins can be used as direct input voltage measurements, in conjunction with the voltage SD ADC, or for external temperature sense, in conjunction with the temperature SD ADC. PTB5 is a dedicated switch to GND, used for external temperature sensors. PTB4 has a configurable wake-up capability, and battery voltage rating performances. PTB1 to PTB3 can be used as General Purpose 5.0 V I/Os. PTB4 can be used as a General Purpose 5.0 V Input.

5.1.2.3 Interface with the embedded microcontroller

This is achieved via a dedicated Die to Die interface. The die to die operating frequency and operating mode is configurable.

5.1.2.4 Power supply and voltage regulators

VDDX is the 5.0 V regulator to supply the MCU die. Current capability is up to 150 mA. VDDA is the 2.5 V regulator, to supply the SD ADCs. This pin is also the supply for the external temperature sensors. VDDH is for the die to die interface supply. LTO is the internal logic device supply.

5.1.2.5 Operating modes and wake-up

The device features four main operating modes:

- Normal mode, with all functions operating: three SD ADCS, microcontroller, SCI and LIN interface, timer, GPIO, Watchdog, reset, and INT.
- Cranking mode allows operation at a very low supply.
- Low Power Sleep mode, with VDDX OFF.
- Low Power Stop mode, with VDDX ON.

From Low Power modes, the device can wake up from one of the below events:

- LIN
- Timer
- Current threshold
- Ampere/Hour counter
- Lifetime counter
- Events on PTB4
- Temperature calibration request

5.1.2.6 SCI and LIN:

- SCI interface and LIN physical interface, compliant with LIN 2.1, including TxD permanent dominant protection.

5.1.2.7 Watchdog

- Configurable window watchdog for the MCU operation and software monitoring

5.1.2.8 Timer

- Four input capture / output compare channels 16-bit timer, with clock prescaler.
- It can be used as the regular time base, or routed to PTB1 to PTB4, or to the LIN physical interface.

5.2 MC9S12ZI128 overview

5.2.1 Introduction

The MC9S12ZI128 device are designed as counter parts to an analog die and are not being offered as stand-alone MCUs.

The MC9S12ZI128 device contains a S12Z Central Processing Unit (CPU), offers 128 kB of Flash memory and 8.0 kB of system SRAM, up to eight general purpose I/Os, an on-chip oscillator and clock multiplier, one CAN module (msCAN), one Serial Peripheral Interface (SPI), an interrupt module and debug capabilities via the on-chip debug module (DBG) in combination with the Background Debug Mode (BDM) interface. Additionally there is a die-to-die initiator (D2DI) which represents the communication interface to the companion (analog) die.

5.2.2 Features

This section describes the key features of the MC9S12ZI128 micro controller unit.

5.2.2.1 Chip-level features

On-chip modules available within the family include the following features:

- S12Z CPU core (S12ZCPU)
- 128 kbyte on-chip flash with ECC
- 4.0 kbyte on-chip EEPROM with ECC
- 8.0 kbyte on-chip SRAM with ECC
- Phase locked loop (IPLL) frequency multiplier with internal filter
- 4.0 to 16.384 MHz amplitude controlled Pierce oscillator
- 1.024 MHz internal RC oscillator
- 51.2 MHz bus frequency
- One CAN module (msCAN)
- One serial peripheral interface (SPI) module
- On-chip voltage regulator (VREG) for regulation of input supply and all internal voltages
- Die to Die Initiator (D2DI)

5.2.3 Module features

The following sections provide more details of the modules implemented on the MC9S12ZI128.

5.2.3.1 S12Z central processor unit (CPU)

The S12Z CPU is a revolutionary high-speed core, with code size and execution efficiencies over the S12X CPU. The S12Z CPU also provides a linear memory map eliminating the inconvenience and performance impact of page swapping.

- Harvard Architecture - parallel data and code access
- Three stage pipeline
- 32-Bit wide instruction and databus
- 32-Bit ALU
- 24-bit addressing, of 16 MByte linear address space
- Instructions and Addressing modes optimized for C-Programming & Compiler
- Optimized address path so it is capable to run at 51.2 MHz without Flash wait states
  - MAC unit 32 bit += 32 bit*32 bit
  - Hardware divider
  - Single cycle multi-bit shifts (Barrel shifter)
  - Special instructions for fixed point math
- Unimplemented opcode traps
- Unprogrammed byte value (0xFF) defaults to SWI instruction

5.2.3.1.1 Background debug controller (BDC)

- Background debug controller (BDC) with single-wire interface
  - Non-intrusive memory access commands
  - Supports in-circuit programming of on-chip nonvolatile memory

5.2.3.1.2 Debugger (DBG)

- Enhanced DBG module including;
  - Four comparators (A, B, C, and D) each configurable to monitor PC addresses or address of data access
  - A and C compare full address bus and full 32-bit data bus with data bus mask register
  - B and D compare full address bus only
  - Three modes: simple address/data match, inside address range, or outside address range
  - Tag-type or force-type hardware breakpoint requests
- State sequencer control
- 64 x 64-bit circular trace buffer to capture change-of-flow addresses or address and data of every cycle
  - Begin, End and Mid alignment of tracing to trigger
- Profiling mode for external visibility of internal program flow

5.2.3.2 Embedded memory

5.2.3.2.1 Memory access integrity

- Illegal address detection
- ECC support on embedded NVM and system RAM

5.2.3.2.2 Flash

On-chip flash memory on the MC9S12ZI128 features the following:

- Up to 128 kbytes of program flash memory
  - Automated program and erase algorithm
  - Protection scheme to prevent accidental program or erase

5.2.3.2.3 EEPROM

- Up to 4.0 kbytes EEPROM
  - 16 data bits plus six syndrome ECC (error correction code) bits allow single bit error correction and double fault detection
  - Erase sector size 4.0 bytes
  - Automated program and erase algorithm
  - User margin level setting for reads

5.2.3.2.4 SRAM

- Up to 8.0 kbytes of general purpose RAM with ECC
  - Single bit error correction and double bit error detection

5.2.3.3 Clocks, reset & power management unit (CPMU)

- Real time interrupt (RTI)
- Clock monitor, supervising the correct function of the oscillator (CM)
- Computer operating properly (COP) watchdog
  - Configurable as window COP for enhanced failure detection
  - Can be initialized out of reset using option bits located in flash memory
- System reset generation
- Autonomous periodic interrupt (API) (combination with cyclic, watchdog)
- Low Power Operation
  - RUN mode is the main full performance operating mode with the entire device clocked.
  - WAIT mode when the internal CPU clock is switched off, so the CPU does not execute instructions.
  - Pseudo STOP - system clocks are stopped but the oscillator the RTI, the COP, and API modules can be enabled
  - STOP - the oscillator is stopped in this mode, all clocks are switched off and all counters and dividers remain frozen, with the exception of the COP and API which can optionally run from ACLK.

5.2.3.3.1 Internal phase-locked loop (IPLL)

- Phase-locked-loop clock frequency multiplier
  - No external components required
  - Reference divider and multiplier allow large variety of clock rates
  - Automatic bandwidth control mode for low-jitter operation
  - Automatic frequency lock detector
  - Configurable option to spread spectrum for reduced EMC radiation (frequency modulation)
  - Reference clock sources:
    - Internal 1.024 MHz RC oscillator (IRC)
    - External 4.0 -16.384 MHz crystal oscillator/resonator

5.2.3.3.2 Internal RC oscillator (IRC)

- Trimmable internal 1.024 MHz reference clock.
  - Trimmed accuracy over -40 to 150 °C junction temperature range: ±1.3%max.

5.2.3.4 Main external oscillator (XOSCLCP)

- Amplitude controlled Pierce oscillator using 4.0 MHz to 16.384 MHz crystal
  - Current gain control on amplitude output
  - Signal with low harmonic distortion
  - Low power
  - Good noise immunity
  - Eliminates need for external current limiting resistor
  - Transconductance sized for optimum start-up margin for typical crystals
  - Oscillator pins shared with GPIO functionality

5.2.3.5 System integrity support

- Power-on reset (POR)
- System reset generation
- Illegal address detection with reset
- Low-voltage detection with interrupt or reset
- Real time interrupt (RTI)
- Computer operating properly (COP) watchdog
  - Configurable as window COP for enhanced failure detection
  - Initialized out of reset using option bits located in flash memory
- Clock monitor supervising the correct function of the oscillator

5.2.3.6 Multi-scalable controller area network (MSCAN)

- Implementation of the CAN protocol — Version 2.0A/B
- Five receive buffers with FIFO storage scheme
- Three transmit buffers with internal prioritization using a “local priority” concept
- Flexible maskable identifier filter supports two full-size (32-bit) extended identifier filters, or four 16-bit filters, or either 8-bit filters
- Programmable wake-up functionality with integrated low-pass filter

5.2.3.7 Serial peripheral interface module (SPI)

- Configurable 8 or 16-bit data size
- Full-duplex or single-wire bidirectional
- Double-buffered transmit and receive
- Master or slave mode
- MSB-first or LSB-first shifting
- Serial clock phase and polarity options

5.2.3.8 On-chip voltage regulator (VREG)

- Linear voltage regulator with bandgap reference
- Low-voltage detect (LVD) with low-voltage interrupt (LVI)
- Power-on reset (POR) circuit
- Low-voltage reset (LVR)

5.2.3.9 Die to die initiator (D2DI)

- Up to 2.0 Mbyte data rate
- Configurable 4-bit or 8-bit wide data path

5.2.4 Block diagram

Figure 17 shows a block diagram of the MC9S12ZI128 device

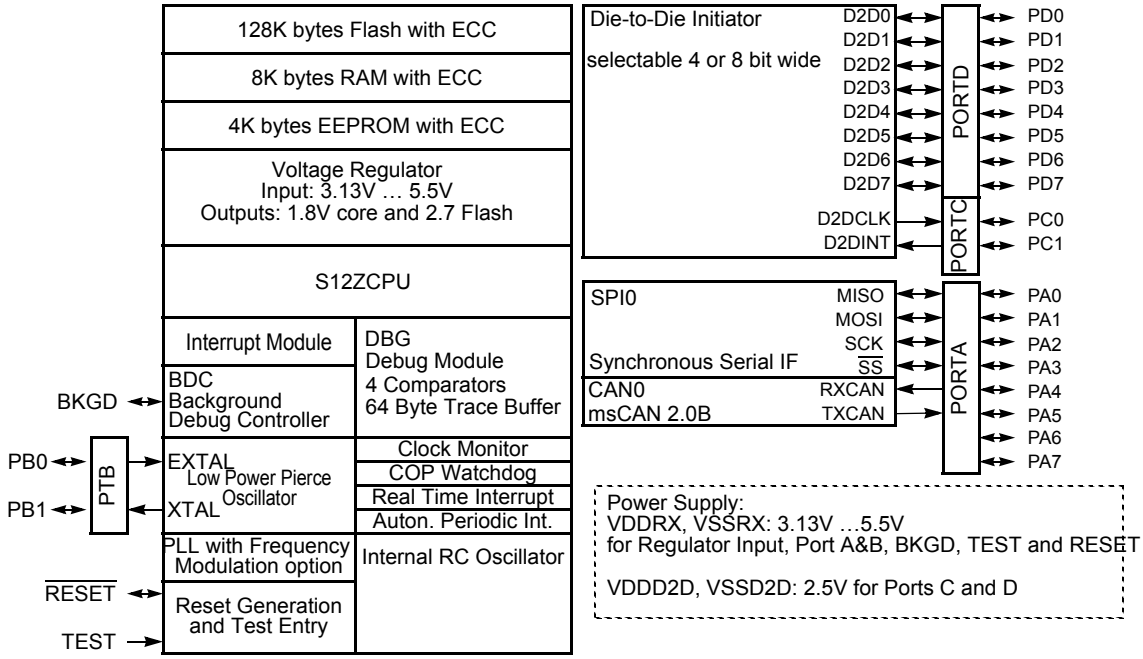


Figure 17. MC9S12ZI128 block diagram

5.2.5 Device memory map

Table 50 shows the device register memory map.

Table 50. Module register address ranges

Address	Module	Size (bytes)
0x0000–0x0003	Part ID Register Part ID assignments	4
0x0004–0x000F	Reserved	12
0x0010–0x001F	INT	16
0x0020–0x006F	Reserved	80
0x0070–0x007F	MMC	16
0x0080–0x00FF	Reserved	128
0x0100–0x017F	DBG	128
0x0180–0x01FF	Reserved	128
0x0200–0x02FF	PIM	256
0x0300–0x037F	Reserved	128
0x0380–0x039F	FTMRZ	32
0x03A0–0x03BF	Reserved	32
0x03C0–0x03CF	RAM ECC	16
0x03D0–0x06BF	Reserved	752
0x06C0–0x06DF	CPMU	32
0x06E0–0x077F	Reserved	160
0x0780–0x0787	SPI0	8
0x0788–0x07EF	Reserved	104

Table 50. Module register address ranges

Address	Module	Size (bytes)
0x07F0–0x07F7	D2DI (Die-to-Die Initiator)	8
0x07F8–0x07FF	Reserved	8
0x0800–0x083F	CAN0	64
0x0840–0x0DFF	Reserved	1472
0x0E00-0x0EFF	D2DI (Die-to-Die Initiator, blocking access window) (71)	256
0x0F00–0x0FFF	D2DI (Die-to-Die Initiator, non-blocking write window) (71)	256

Notes:

71. Refer to Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI).

Note

Reserved register space shown in Table 50 is not allocated to any module. This register space is reserved for future use. Writing to these locations have no effect. Read access to these locations returns zero.

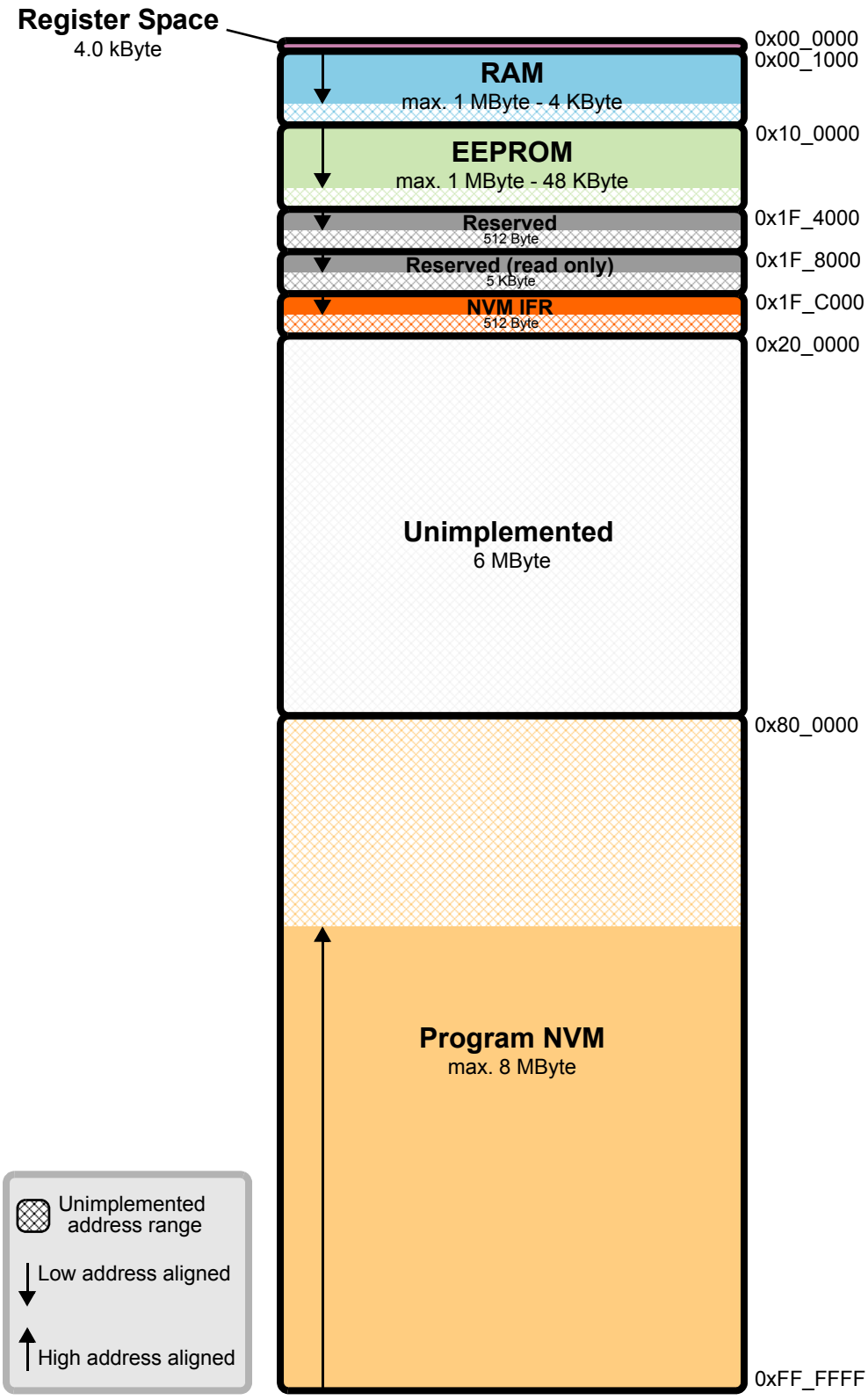


Figure 18. MC9S12ZI128 global memory map. (See Table 51 for individual device details)

5.2.5.1 Part ID assignments

The part ID is located in four 8-bit registers at addresses 0x0000-0x0003. The read-only value is a unique part ID for each revision of the chip.

5.2.5.2 Detailed signal descriptions

Table 51. Signal properties summary

Pin name function 1	Pin name function 2	Pin name function 3	Power supply	Internal pull resistor		Description
				CTRL	Reset state	
PB0	EXTAL	—	VDDRX	PERB/PPSB	DOWN	Port B I/O, Oscillator pin
PB1	XTAL	—	VDDRX	PERB/PPSB	DOWN	Port B I/O, Oscillator pin
RESET	—	—	VDDRX	PULLUP		External reset
TEST	—	—	N.A.	$\overline{\text{RESET}}$ pin	DOWN	Test input
BKGD	MODC	—	VDDRX	PULLUP		Background debug
PA7	—	PDO	VDDRX	PERA/PPSA	Disabled	Port A I/O, DBG
PA6	—	PDOCLK	VDDRX	PERA/PPSA	Disabled	Port A I/O, DBG
PA5	TXCAN0	—	VDDRX	PERA/PPSA	Disabled	Port A I/O, MSCAN
PA4	RXCAN0	—	VDDRX	PERA/PPSA	Disabled	Port A I/O, MSCAN
PA3	SS0	—	VDDRX	PERA/PPSA	Disabled	Port A I/O, SPI
PA2	SCK0	—	VDDRX	PERA/PPSA	Disabled	Port A I/O, SPI
PA1	MOSI0	—	VDDRX	PERA/PPSA	Disabled	Port A I/O, SPI
PA0	MISO0	—	VDDRX	PERA/PPSA	Disabled	Port A I/O, SPI
PC1	D2DINT	—	VDDD2D	D2DEN	Disabled	D2DI
PC0	D2DCLK	—	VDDD2D	NA	NA	D2DI
PD7-0	D2DDAT7-0	—	VDDD2D	D2DEN	Disabled	D2DI

5.2.5.2.1 PB0 / EXTAL, PB1 / XTAL — oscillator pins

EXTAL and XTAL are the crystal driver and external clock pins. XTAL is the oscillator output. On reset all the device clocks are derived from the internal reference clock and port B may be used for general purpose I/O.

5.2.5.2.2 $\overline{\text{RESET}}$ — external reset pin

The $\overline{\text{RESET}}$ pin is an active low bidirectional control signal. It acts as an input to initialize the MCU to a known start-up state, and an output when an internal MCU function causes a reset. The $\overline{\text{RESET}}$ pin has an internal pull-up device.

5.2.5.2.3 TEST — test pin

This input only pin is reserved for factory test. This pin has an internal pull-down device.

Note

The TEST pin must be tied to VSSRX in all applications.

5.2.5.2.4 BKGD / MODC — background debug and mode pin

The BKGD/MODC pin is used as a pseudo-open-drain pin for the background debug communication. It is used as a MCU operating mode select pin during reset. The state of this pin is latched to the MODC bit at the rising edge of $\overline{\text{RESET}}$. The BKGD pin has an internal pull-up device.

5.2.5.2.5 PA7 / PDO — port A I/O Pin 7

PA7 is a general purpose input or output pin. PDO is the profiling data output signal used when the DBG module profiling feature is enabled. This signal is output only and provides a serial, encoded data stream that can be used by external development tools to reconstruct the internal CPU code flow.

5.2.5.2.6 PA6 / PDOCLK — port A I/O Pin 6

PA6 is a general purpose input or output PDOCLK is the clock output signal for the profiling data output.

5.2.5.2.7 PA5 / TXCAN0 — port A I/O Pin 5

PA5 is a general purpose input or output pin. TXCAN0 is the transmit output of the scalable controller area network controller (MSCAN0)

5.2.5.2.8 PA4 / RXCAN0 — port A I/O Pin 4

PA4 is a general purpose input or output pin. RXCAN0 is the receive input of the scalable controller area network controller (MSCAN0).

5.2.5.2.9 PA3 / $\overline{SS0}$ — port A I/O Pin 3

PA3 is a general purpose input or output pin. It can be configured as the slave select pin $\overline{SS0}$ of the serial peripheral interface (SPI0).

5.2.5.2.10 PA2 / SCK0 — port A I/O Pin 2

PA2 is a general purpose input or output pin. It can be configured as the serial clock pin SCK0 of the serial peripheral interface (SPI0).

5.2.5.2.11 PA1 / MOSI0 — port A I/O Pin 1

PA1 is a general purpose input or output pin. It can be configured as the master output (during master mode) or slave input pin (during slave mode) MOSI0 for the serial peripheral interface (SPI0).

5.2.5.2.12 PA0 / MISO0— port A I/O Pin 0

PA0 is a general purpose input or output pin. It can be configured as the master input (during master mode) or slave output pin (during slave mode) MISO0 for the serial peripheral interface (SPI0).

5.2.5.2.13 PC1 / D2DINT — port C I/O Pin 1

PC1 can be configured as the interrupt input of the D2DI. PC1 has an internal pull down device if D2DI is enabled.

5.2.5.2.14 PC0 / D2DCLK— port C I/O Pin 0

PC0 can be configured as the clock output of the D2DI.

5.2.5.2.15 PD[7:0] — port D I/O Pins 7-0

PD[7:0] can be configured as data pins 7:0 of the D2DI. Ports PD[7:0] have internal pull down devices if D2DI is enabled.

5.2.5.3 Power supply pins

MC9S12ZI128 power and ground pins are described below.

Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high frequency characteristics and place them as close to the MCU as possible.

Note

All V_{SS} pins must be connected together in the application.

5.2.5.3.1 VDDRX, VSSRX— power and ground pins for I/O drivers and internal voltage regulator

External power and ground for I/O drivers and the internal voltage generator. Bypass requirements depend on how heavily the MCU pins are loaded.

5.2.5.3.2 VDDD2D, VSSD2D/VSSD2DI— power and ground pins for D2D interface

External power and ground for the Die-to-Die Interface. VSSD2D and VSSD2D1 are connected together internally.

5.2.5.3.3 VSS

The voltage supply of nominally 1.8 V is derived from the internal voltage regulator. The return current path is through the VSS pin. No static external loading of these pins is permitted.

5.2.5.3.4 Power and ground connection summary

Table 52. Power and ground connection summary

Mnemonic	Nominal voltage	Description
VDDRX	3.13 to 5.5 V	External power and ground, supply to I/O pin drivers and Internal Voltage Regulator
VSSRX	0.0 V	
VDDD2D	2.25 to 3.6 V	External power and ground supply to D2D Interface
VSSD2D	0.0V	

5.2.5.3.5 BDC clock source connectivity

The BDC clock, BDCCLK, is mapped to the IRCCLK generated in the CPMU module.

5.2.6 Modes of operation

The MCU can operate in different modes. These are described in Chip configuration modes.

The MCU can operate in different power modes to facilitate power saving when full system performance is not required. These are described in Low power modes.

Some modules feature a software programmable option to freeze the module status while the background debug module is active to facilitate debugging. This is referred to as freeze mode at module level.

5.2.6.1 Chip configuration modes

The different modes and the security state of the MCU affect the debug features (enabled or disabled).

The operating mode out of reset is determined by the state of the MODC signal during reset (Table 53). The MODC bit in the MODE register shows the current operating mode and provides limited mode switching during operation. The state of the MODC signal is latched into this bit on the rising edge of $\overline{\text{RESET}}$.

Table 53. Chip modes

Chip Modes	MODC
Normal single chip	1
Special single chip	0

5.2.6.1.1 Normal single chip mode

This mode is intended for normal device operation. The opcode from the on-chip memory is being executed after reset (requires the reset vector to be programmed correctly). The processor program is executed from internal memory

5.2.6.1.2 Special single chip mode

This mode is used for debugging operation, boot-strapping, or security related operations. The background debug mode BDM is active on leaving reset in this mode.

5.2.6.2 Low power modes

The device has two dynamic power modes (run and wait) and two static low-power modes stop and pseudo stop). For a detailed description refer to the CPMU section.

- Dynamic power mode: Run
  - Run mode is the main full performance operating mode with the entire device clocked. The user can configure the device operating speed through selection of the clock source and the phase locked loop (PLL) frequency. To save power, unused peripherals must not be enabled.
- Dynamic power mode: Wait
  - This mode is entered when the CPU executes the WAI instruction. In this mode, the CPU does not execute instructions. The internal CPU clock is switched off. All peripherals can be active in system wait mode. For further power consumption the peripherals can individually turn off their local clocks. Asserting RESET, XIRQ, IRQ, or any other interrupt that is not masked, either locally or globally by a CCR bit, ends system wait mode.
- Static power mode: Pseudo-stop
  - In this mode the system clocks are stopped, but the oscillator is still running and the real time interrupt (RTI), watchdog (COP) and autonomous periodic interrupt (API) may be enabled. Other peripherals are turned off. This mode consumes more current than system STOP mode but, as the oscillator continues to run, the full speed wake-up time from this mode is significantly shorter.
- Static power mode: Stop
  - In this mode, the oscillator is stopped and clocks are switched off. The counters and dividers remain frozen. The autonomous periodic interrupt (API) may remain active but has a very low power consumption. The MSCAN transceiver modules can be configured to wake the device, whereby current consumption is negligible.

5.2.7 Security

The MCU security mechanism prevents unauthorized access to the flash memory. It must be emphasized that part of the security must lie with the application code. An extreme example would be application code that dumps the contents of the internal memory. This would defeat the purpose of security. Also, if an application has the capability of downloading code through a serial port and then executing that code (e.g. an application containing bootloader code), then this capability could potentially be used to read the EEPROM and Flash memory contents even when the microcontroller is in the secure state. In this example, the security of the application could be enhanced by requiring a response authentication before any code can be downloaded.

Device security details are also described in the flash block description.

5.2.7.1 Features

The security features of the S12Z chip family are:

- Prevent external access of the non-volatile memories (Flash, EEPROM) content
- Restrict execution of NVM commands
- Prevent BDC access of internal resources

5.2.7.2 Securing the microcontroller

The chip can be secured by programming the security bits located in the options/security byte in the Flash memory array. These non-volatile bits keep the device secured through reset and power-down.

This byte can be erased and programmed like any other Flash location. Two bits of this byte are used for security (SEC[1:0]). The contents of this byte are copied into the Flash security register (FSEC) during a reset sequence.

The meaning of the security bits SEC[1:0] is shown in Table 54. For security reasons, the state of device security is controlled by two bits. To put the device in unsecured mode, these bits must be programmed to SEC[1:0] = ‘10’. All other combinations put the device in a secured mode. The recommended value to put the device in secured state is the inverse of the unsecured state, i.e. SEC[1:0] = ‘01’.

Table 54. Security bits

SEC[1:0]	Security state
00	1 (secured)
01	1 (secured)
10	0 (unsecured)
11	1 (secured)

Note

Refer to the Flash block description for more security byte details.

5.2.7.3 Operation of the secured microcontroller

By securing the device, unauthorized access to the EEPROM and Flash memory contents is prevented. Secured operation has the following effects on the microcontroller:

5.2.7.3.1 Normal single chip mode (NS)

- Background debug controller (BDC) operation is completely disabled.
- Execution of Flash and EEPROM commands is restricted (described in flash block description).

5.2.7.3.2 Special single chip mode (SS)

- Background debug controller (BDC) commands are restricted
- Execution of Flash and EEPROM commands is restricted (described in flash block description).

In special single chip mode, the device is in active BDM after reset. In special single chip mode on a secure device, only the BDC mass erase and BDC control and status register commands are possible. BDC access to memory mapped resources is disabled. The BDC can only be used to erase the EEPROM and Flash memory without giving access to their contents.

5.2.7.4 Unsecuring the microcontroller

Unsecuring the microcontroller can be done using three different methods:

1. Back door key access
2. Reprogramming the security bits
3. Complete memory erase

5.2.7.4.1 Unsecuring the MCU using the back door key access

In normal single chip mode, security can be temporarily disabled using the back door key access method. This method requires that:

- The back door key has been programmed to a valid value
- The KEYEN[1:0] bits within the Flash options/security byte select ‘enabled’
- The application program programmed into the microcontroller has the capability to write to the back door key locations

The back door key values themselves would not normally be stored within the application data, which means the application program would have to be designed to receive the back door key values from an external source (e.g. through a serial port).

The back door key access method allows debugging of a secured microcontroller without having to erase the Flash. This is particularly useful for failure analysis.

Note

No back door key word is allowed to have the value 0x0000 or 0xFFFF.

5.2.7.5 Reprogramming the security bits

In normal single chip mode, security can also be disabled by erasing and reprogramming the security bits within Flash options/security byte to the unsecured value. Because the erase operation will erase the entire sector from (0x7F_FE00–0x7F_FFFF), the back door key and the interrupt vectors will also be erased; this method is not recommended for normal single chip mode. The application software can

only erase and program the Flash options/security byte if the Flash sector containing the Flash options/security byte is not protected (see Flash protection). Thus Flash protection is a useful means of preventing this method. The microcontroller will enter the unsecured state after the next reset following the programming of the security bits to the unsecured value.

This method requires that:

- The application software previously programmed into the microcontroller has been designed to have the capability to erase and program the Flash options/security byte.
- The Flash sector containing the Flash options/security byte is not protected.

5.2.7.6 Complete memory erase

The microcontroller can be unsecured by erasing the entire EEPROM and Flash memory contents. If ERASE_FLASH is successfully completed, then the Flash unsecures the device and programs the security byte automatically.

5.2.8 Resets and interrupts

5.2.8.1 Resets

Table 55 lists all reset sources and the vector locations. Resets are explained in detail in the Packaging".

Table 55. Reset sources and vector locations

Vector address	Reset source	CCR mask	Local enable
0xFFFFFC	Power-On Reset (POR)	None	None
	Low Voltage Reset (LVR)	None	None
	External pin $\overline{\text{RESET}}$	None	None
	Clock monitor reset	None	OSCE Bit in CPMUOSC register
	COP watchdog reset	None	CR[2:0] in CPMUCOP register

5.2.8.2 Interrupt vectors

Table 56 lists all interrupt sources and vectors in the default order of priority. The interrupt module description provides an interrupt vector base register (IVBR) to relocate the vectors.

Table 56. Interrupt Vector Locations (Sheet 1 of 2)

Vector address [(72)]	Interrupt source	CCR mask	Local enable	Wake-up from STOP	Wake-up from WAIT
Vector base + 0x1F8	Unimplemented page1 op-code trap (SPARE)	None	None	-	-
Vector base + 0x1F4	Unimplemented page2 op-code trap (TRAP)	None	None	-	-
Vector base + 0x1F0	Software interrupt instruction (SWI)	None	None	-	-
Vector base + 0x1EC	System call interrupt instruction (SYS)	None	None	-	-
Vector base + 0x1E8	Machine exception	None	None	-	-
Vector base + 0x1E4	Reserved				
Vector base + 0x1E0	Reserved				
Vector base + 0x1DC	Spurious interrupt	—	None	-	-
Vector base + 0x1D8	D2DI Error Interrupt	X Bit	None	No	No
Vector base + 0x1D4	D2DI External Interrupt	I bit	D2DCTL (D2DIE)	Yes	Yes
Vector base + 0x1D0	RTI timeout interrupt	I bit	CPMUINT (RTIE)	See CPMU section	Yes
Vector base + 0x1CC to Vector base + 0x1A4	Reserved				
Vector base + 0x1A0	SPI0	I bit	SPICR1 (SPIE, SPTIE)	No	Yes

Table 56. Interrupt Vector Locations (Sheet 2 of 2) (continued)

Vector address (72)	Interrupt source	CCR mask	Local enable	Wake-up from STOP	Wake-up from WAIT
Vector base + 0x19C to Vector base + 0x184	Reserved				
Vector base + 0x180	Oscillator status interrupt	I bit	CPMUINT (OSCIE)	No	Yes
Vector base + 0x17C	PLL lock interrupt	I bit	CPMUINT (LOCKIE)	No	Yes
Vector base + 0x178 to Vector base + 0x174	Reserved				
Vector base + 0x170	RAM error	I bit	ECCIE (SBEEIE)	No	Yes
Vector base + 0x16C to Vector base + 0x168	Reserved				
Vector base + 0x164	FLASH error	I bit	FERCNFG (SFDIE)	No	Yes
Vector base + 0x160	FLASH command	I bit	FCNFG (CCIE)	No	Yes
Vector base + 0x15C	CAN0 wake-up	I bit	CANRIER (WUPIE)	Yes	Yes
Vector base + 0x158	CAN0 errors	I bit	CANRIER (CSCIE, OVRIE)	No	Yes
Vector base + 0x154	CAN0 receive	I bit	CANRIER (RXFIE)	No	Yes
Vector base + 0x150	CAN0 transmit	I bit	CANRIER (TXEIE[2:0])	No	Yes
Vector base + 0x14C to Vector base + 0x108	Reserved				
Vector base + 0x104	Low-voltage interrupt (LVI)	I bit	CPMUCTRL (LVIE)	No	Yes
Vector base + 0x100	Autonomous periodical interrupt (API)	I bit	CPMUAPICTRL (APIE)	Yes	Yes
Vector base + 0xFC to Vector base + 0x10	Reserved				

Notes:
72.15 bits vector address based

5.2.8.3 Effects of reset

When a reset occurs, MCU registers and control bits are initialized. Refer to the respective block sections for register reset states.

On each reset, the Flash module executes a reset sequence to load Flash configuration registers.

5.2.8.3.1 Flash configuration reset sequence phase

On each reset, the Flash module holds CPU activity while loading Flash module registers from the Flash memory. If double faults are detected in the reset phase, Flash module protection and security may be active when leaving reset. This is explained in more detail in the Flash module description.

5.2.8.3.2 Reset while flash command active

If a reset occurs while any Flash command is in progress, that command is immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

5.2.8.3.3 I/O pins

Refer to the PIM section for reset configurations of all peripheral module ports.

5.2.8.3.4 RAM

The system RAM arrays, including their ECC syndromes, are initialized following a power on reset. All other RAM arrays are not initialized out of any type of reset.

With the exception of resets resulting from low voltage conditions, the RAM content is unaltered by a reset occurrence.

5.2.9 BDC clock source connectivity

The BDC clock, BDCCLK, is mapped to the IRCCLK generated in the CPMU module.

The BDC clock, BDCFCLK, is mapped the device bus clock generated in the CPMU module.

5.2.10 COP configuration

The COP timeout rate bits CR[2:0] and the WCOP bit in the CPMUCOP register are loaded from the Flash configuration field byte at global address 0xFF_FE0E during the reset sequence. See Table 57 and Table 58 for coding.

Table 57. Initial COP rate configuration

NV[2:0] in FOPT Register	CR[2:0] in COPCTL Register
000	111
001	110
010	101
011	100
100	011
101	010
110	001
111	000

Table 58. Initial WCOP configuration

NV[3] in FOPT Register	WCOP in COPCTL Register
1	0
0	1

6 Functional description and application information

This chapter describes the MM9Z1_638 dual die device functions on a block by block base.

6.1 Device register map

Table 59 shows the device register memory map overview.

Table 59. Device register memory map overview

Address	Module	Size (Bytes)
0x0000–0x0DFF	Ref to Module register address ranges	3584
0x0E00–0x0EFF	D2DI (die 2 die initiator, blocking access window)	256
0x0F00–0x0FFF	D2DI (die 2 die initiator, non-blocking write window)	256

Note

The reserved register space shown in Table 59 is not allocated to any module. This register space is reserved for future use. Writing to these locations has no effect. Read access to these locations returns a zero.

6.1.1 Detailed module register map

Bit Legend

= Unimplemented, Reserved

= Implemented (do not alter)

X = Indeterminate

0 = Always read zero

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x00	PCR_CTL (hi)	R	0	0	0	0	0	0	0	0
	PCR Control Register	W	HTIEM	UVIEM	HWRM		PFM		OPMM	
	PCR_CTL (lo)	R	HTIE	UVIE	0		PF		OPM	
	PCR Control Register	W			HWR					
0x02	PCR_SR (hi)	R	HTF	UVF	HWRF	WDRF	HVRF	LVRF	WULTCF	WLPMF
	PCR Status Register	W	Write 1 will clear the flags							
0x03	PCR_SR (lo)	R	WUAHTHF	WUCTHF	WUCALF	WULINF	WUPTB4F	WUPTB3F	WUPTB2F	WUPTB1F
	PCR Status Register	W	Write 1 will clear the flags							
0x04	PCR_PRESC (hi)	R	PRESC							
	PCR 1.0 ms prescaler	W								
	PCR_PRESC (lo)	R								
	PCR 1.0 ms prescaler	W								
0x06	PCR_WUE (hi)	R	WUAHTH	WUCTH	WUCAL	WULIN	WUPTB4	WUPTB3	WUPTB2	WUPTB1
	Wake-up Enable Register	W								
0x07	PCR_WUE (lo)	R	WULTC	0	0	0	0	0	HTWF	TSDF
	Wake-up Enable Register	W								
0x08	INT_SRC (hi)	R	TOV	CH3	CH2	CH1	CH0	LFI	HTI	UVI
	Interrupt source register	W								
0x09	INT_SRC (lo)	R	0	0	CAL	LTC	CVMI	RX	TX	ERR
	Interrupt source register	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x0A	INT_VECT	R	0	0	0	0	IRQ			
	Interrupt vector register	W								
0x0B		R	0	0	0	0	0	0	0	0
		W								
0x0C	INT_MSK (hi)	R	TOVM	CH3M	CH2M	CH1M	CH0M	LFIM	HTIM	UVIM
	Interrupt mask register	W								
	INT_MSK (lo)	R	0	0	CALM	LTCM	CVMM	RXM	TXM	ERRM
	Interrupt mask register	W								
0x0E	TRIM_ALF (hi)	R	PRDF	0	0	APRESC				
	Trim for accurate 1.0 ms low freq clock	W								
	TRIM_ALF (hi)	R	APRESC							
	Trim for accurate 1.0 ms low freq clock	W								
0x10	WD_CTL (hi)	R	0	0	0	0	0	0	0	0
	Watchdog control register	W	WDTSTM					WDTOM		
	WD_CTL (lo)	R	WDTST	0	0	0	0	WDTO		
	Watchdog control register	W								
0x12	WD_SR	R	0	0	0	0	0	0	WDOFF	WDWO
	Watchdog status register	W								
0x13		R	0	0	0	0	0	0	0	0
		W								
0x14	WD_RR	R	WDR							
	Watchdog rearm register	W								
0x15		R	0	0	0	0	0	0	0	0
		W								
0x16		R	0	0	0	0	0	0	0	0
		W								
0x17		R	0	0	0	0	0	0	0	0
		W								
0x18	SCIBD (hi)	R	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
	SCI Baud Rate Register	W								
	SCIBD (lo)	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	SCI Baud Rate Register	W								
0x1A	SCIC1	R	LOOPS	DPD1	RSRC	M	DPD0	ILT	PE	PT
	SCI Control Register 1	W								
0x1B	SCIC2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	SCI Control Register 2	W								
0x1C	SCIS1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	SCI Status Register 1	W								
0x1D	SCIS2	R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
	SCI Status Register 2	W								
0x1E	SCIC3	R	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
	SCI Control Register 3	W								
0x1F	SCID	R	R7	R6	R5	R4	R3	R2	R1	R0
	SCI Data Register	W	T7	T6	T5	T4	T3	T2	T1	T0

MM9Z1_638

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x20	TIOS	R	0	0	0	0	IOS3	IOS2	IOS1	IOS0
	Timer Input Capture/Output Compare Select	W								
0x21	CFORC	R	0	0	0	0	0	0	0	0
	Timer Compare Force Register	W					FOC3	FOC2	FOC1	FOC0
0x22	OC3M	R	0	0	0	0	OC3M3	OC3M2	OC3M1	OC3M0
	Output Compare 3 Mask Register	W								
0x23	OC3D	R	0	0	0	0	OC3D3	OC3D2	OC3D1	OC3D0
	Output Compare 3 Data Register	W								
0x24	TCNT (hi)	R	TCNT							
	Timer Count Register	W								
	TCNT (lo)	R								
	Timer Count Register	W								
0x26	TSCR1	R	TEN	0	0	TFFCA	0	0	0	0
	Timer System Control Register 1	W								
0x27	TTOV	R	0	0	0	0	TOV3	TOV2	TOV1	TOV0
	Timer Toggle Overflow Register	W								
0x28	TCTL1	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	Timer Control Register 1	W								
0x29	TCTL2	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
	Timer Control Register 2	W								
0x2A	TIE	R	0	0	0	0	C3I	C2I	C1I	C0I
	Timer Interrupt Enable Register	W								
0x2B	TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
	Timer System Control Register 2	W								
0x2C	TFLG1	R	0	0	0	0	C3F	C2F	C1F	C0F
	Main Timer Interrupt Flag 1	W								
0x2D	TFLG2	R	TOF	0	0	0	0	0	0	0
	Main Timer Interrupt Flag 2	W								
0x2E	TC0 (hi)	R	TC0							
	Timer Input Capture/Output Compare Register 0	W								
	TC0 (lo)	R								
	Timer Input Capture/Output Compare Register 0	W								
0x30	TC1 (hi)	R	TC1							
	Timer Input Capture/Output Compare Register 1	W								
	TC1 (lo)	R								
	Timer Input Capture/Output Compare Register 1	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x32	TC2 (hi)	R	TC2							
	Timer Input Capture/Output Compare Register 2	W								
	TC2 (lo)	R								
	Timer Input Capture/Output Compare Register 2	W								
0x34	TC3 (hi)	R	TC3							
	Timer Input Capture/Output Compare Register 3	W								
	TC3 (lo)	R								
	Timer Input Capture/Output Compare Register 3	W								
0x36	TIMTST	R	0	0	0	0	0	0	TCBYP	0
	Timer Test Register	W								
0x37		R	0	0	0	0	0	0	0	0
		W								
0x38	LTC_CTL (hi)	R	0	0	0	0	0	0	0	0
	Life Time Counter control register	W	LTCIEM							LTCEM
0x39	LTC_CTL (lo)	R	LTCIE	0	0	0	0	0	0	LTCE
	Life Time Counter control register	W								
0x3A	LTC_SR	R	LTCOF	0	0	0	0	0	0	0
	Life Time Counter status register	W	Write 1 will clear the flag							
0x3B		R	0	0	0	0	0	0	0	0
		W								
0x3C	LTC_CNT1 (hi)	R	LTC							
	Life Time Counter Register	W								
0x3D	LTC_CNT1 (lo)	R								
	Life Time Counter Register	W								
0x3E	LTC_CNT0 (hi)	R								
	Life Time Counter Register	W								
0x3F	LTC_CNT0 (lo)	R								
	Life Time Counter Register	W								
0x40	GPIO_CTL (hi)	R	0	0	0	0	0	0	0	0
	GPIO control register	W	DIR3M	DIR2M	DIR1M	PE4M	PE3M	PE2M	PE1M	
0x41	GPIO_CTL (lo)	R	DIR3	DIR2	DIR1	PE4	PE3	PE2	PE1	0
	GPIO control register	W								
0x42	GPIO_PUC	R	0	0	0	PDE4	PUE3	PUE2	PUE1	0
	GPIO pull-up/down configuration	W								
0x43	GPIO_DATA	R	0	0	0	PD4	PD3	PD2	PD1	0
	GPIO port data register	W								
0x44	GPIO_IN1	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 1input configuration	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x45	GPIO_OUT1	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 1output configuration	W								PTBX1
0x46	GPIO_IN2	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 2 input configuration	W								
0x47	GPIO_OUT2	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 2 output configuration	W								PTBX2
0x48	GPIO_IN3	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 3 input configuration	W								
0x49	GPIO_OUT3	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 3 output configuration	W								PTBX3
0x4A	GPIO_IN4	R	PTWU	TCAP3	TCAP2	TCAP1	TCAP0	0	NWUS	NWUE
	Port 4 input configuration	W								
0x4B	GPIO_VSENSE	R	VSSEL		0	VSE4	VSE3	VSE2	VSE1	VSE0
	GPIO V_{SENSE} configuration	W								
0x4C	GPIO_TSENSE	R	0	0	0	TSE4	TSE3	TSE2	TSE1	TSE0
	GPIO T_{SENSE} configuration	W								
0x4D		R	0	0	0	0	0	0	0	0
		W								
0x4E		R	0	0	0	0	0	0	0	0
		W								
0x4F		R	0	0	0	0	0	0	0	0
		W								
0x50	LIN_CTL (hi)	R	0	0	0	0	0	0	0	0
	LIN control register	W	OTIEM	TXDOMIEM			LVSDM	ENM	SRSM	
0x51	LIN_CTL (lo)	R	OTIE	TXDOMIE	0		LVSD	EN	SRS	
	LIN control register	W								
0x52	LIN_SR (hi)	R	OT	TXDOM	HF	0	UV	0	0	0
	LIN status register	W	Write 1 will clear the flags							
0x53	LIN_SR (lo)	R	RDY	0	0	0	0	0	RX	TX
	LIN status register	W								
0x54	LIN_TX	R	0	0	0	0	0	0	FROMPTB	FROMSCI
	LIN transmit line definition	W								
0x55	LIN_RX	R	0	0	0	0	0	0	TOPTB	TOSCI
	LIN receive line definition	W								
0x56		R	0	0	0	0	0	0	0	0
		W								
0x57		R	0	0	0	0	0	0	0	0
		W								
0x58	ACQ_CTL (hi)	R	0	0	0	0	0	0	0	0
	Acquisition control register	W	AHCRM		NVSEM	CVMIEM	ETMENM	ITMENM	VMENM	CMENM
0x59	ACQ_CTL (lo)	R	0		NVSE	CVMIE	ETMEN	ITMEN	VMEN	CMEN
	Acquisition control register	W	AHCR							
0x5A	ACQ_SR (hi)	R	AVRF	PGAG	VMOW	CMOW	ETM	ITM	VM	CM
	Acquisition status register	W	Write 1 will clear the flags							

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x5B	ACQ_SR (lo)	R	0	0	0	VTH	ETCHOP	ITCHOP	VCHOP	CCHOP
	Acquisition status register	W								
0x5C	ACQ_ACC1 (hi)	R	0	0	0	0	0	0	0	0
	Acquisition chain control	W	TCOMPM	VCOMPM	CCOMPM	LPFENM	ETCHOPM	ITCHOPM	CVCHOPM	AGENM
	ACQ_ACC1 (lo)	R	TCOMP	VCOMP	CCOMP	LPFEN	ETCHOP	ITCHOP	CVCHOP	AGEN
	Acquisition chain control	W								
0x5E	ACQ_ACC0 (hi)	R	0	0	0	0	0	0	0	0
	Acquisition chain control	W	ZEROM							
	ACQ_ACC0 (lo)	R	ZERO							
	Acquisition chain control	W								
0x60	ACQ_DEC	R	0	0	0	0	0	DEC[2:0]		
	Decimation rate	W								
0x61		R	0	0	0	0	0	0	0	0
		W								
0x62	ACQ_GAIN	R	G256DIS	G64DIS	G16DIS	G4DIS	LPGEN	IGAIN[1:0]		0
	PGA gain	W								
	ACQ_GCB	R	D							
	GCB threshold	W								
0x64	ACQ_ITEMP (hi)	R	ITEMP[15:8]							
	Internal temperature measurement	W								
	ACQ_ITEMP (lo)	R	ITEMP[7:0]							
	Internal temperature measurement	W								
0x66	ACQ_ETEMP (hi)	R	ETEMP[15:8]							
	External temperature measurement	W								
	ACQ_ETEMP (lo)	R	ETEMP[7:0]							
	External temperature measurement	W								
0x68		R	0	0	0	0	0	0	0	0
		W								
	ACQ_CURR1	R	CURR[23:16]							
	Current measurement	W								
0x6A	ACQ_CURR0 (hi)	R	CURR[15:8]							
	Current measurement	W								
	ACQ_CURR0 (lo)	R	CURR[7:0]							
	Current measurement	W								
0x6C	ACQ_VOLT (hi)	R	VOLT[15:8]							
	Current measurement	W								
	ACQ_VOLT (lo)	R	VOLT[7:0]							
	Current measurement	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x6E	ACQ_LPFC	R	0	0	0	0	LPFC			
	Low pass filter coefficient number	W								
		R	0	0	0	0	0	0	0	0
		W								
0x70	ACQ_TCMP	R	TCMP							
	Low power trigger current measurement period	W								
		R								
		W								
0x72	ACQ_THF	R	THF							
	Low power current threshold filtering period	W								
		R	0	0	0	0	0	0	0	0
		W								
0x74	ACQ_CVCR (hi)	R	0	0	0	0	0	0	0	0
	Current and voltage chopper control register	W					IIRCM			
	ACQ_CVCR (lo)	R	0	0	0	0	IIRC			0
	Current and voltage chopper control register	W								
0x76	ACQ_CTH	R	CTH							
	Low power current threshold	W								
		R	0	0	0	0	0	0	0	0
		W								
0x78	ACQ_AHTH1 (hi)	R	0	AHTH[30:0]						
	Low power Ah counter threshold	W								
0x79	ACQ_AHTH1 (lo)	R								
	Low power Ah counter threshold	W								
0x7A	ACQ_AHTH0 (hi)	R								
	Low power Ah counter threshold	W								
0x7B	ACQ_AHTH0 (lo)	R								
	Low power Ah counter threshold	W								
0x7C	ACQ_AHC1 (hi)	R	AHC[31:0]							
	Low power Ah counter	W								
0x7D	ACQ_AHC1 (lo)	R	AHC[23:16]							
	Low power Ah counter	W								
0x7E	ACQ_AHC0 (hi)	R	AHC[15:8]							
	Low power Ah counter	W								
0x7F	ACQ_AHC0 (lo)	R	AHC[7:0]							
	Low power Ah counter	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x80	LPF_A0 (hi)	R	A0[15:0]							
	A0 filter coefficient	W								
	LPF_A0 (lo)	R								
	A0 filter coefficient	W								
0x82	LPF_A1 (hi)	R	A1[15:0]							
	A1 filter coefficient	W								
	LPF_A1 (lo)	R								
	A1 filter coefficient	W								
0x84	LPF_A2 (hi)	R	A2[15:0]							
	A2 filter coefficient	W								
	LPF_A2 (lo)	R								
	A2 filter coefficient	W								
0x86	LPF_A3 (hi)	R	A3[15:0]							
	A3 filter coefficient	W								
	LPF_A3 (lo)	R								
	A3 filter coefficient	W								
0x88	LPF_A4 (hi)	R	A4[15:0]							
	A4 filter coefficient	W								
	LPF_A4 (lo)	R								
	A4 filter coefficient	W								
0x8A	LPF_A5 (hi)	R	A5[15:0]							
	A5 filter coefficient	W								
	LPF_A5 (lo)	R								
	A5 filter coefficient	W								
0x8C	LPF_A6 (hi)	R	A6[15:0]							
	A6 filter coefficient	W								
	LPF_A6 (lo)	R								
	A6 filter coefficient	W								
0x8E	LPF_A7 (hi)	R	A7[15:0]							
	A7 filter coefficient	W								
	LPF_A7 (lo)	R								
	A7 filter coefficient	W								
0x90	LPF_A8 (hi)	R	A8[15:0]							
	A8 filter coefficient	W								
	LPF_A8 (lo)	R								
	A8 filter coefficient	W								
0x92	LPF_A9 (hi)	R	A9[15:0]							
	A9 filter coefficient	W								
	LPF_A9 (lo)	R								
	A9 filter coefficient	W								
0x94	LPF_A10 (hi)	R	A10[15:0]							
	A10 filter coefficient	W								
	LPF_A10 (lo)	R								
	A10 filter coefficient	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0x96	LPF_A11 (hi)	R	A11[15:0]							
	A11 filter coefficient	W								
	LPF_A11 (lo)	R								
	A11 filter coefficient	W								
0x98	LPF_A12 (hi)	R	A12[15:0]							
	A12 filter coefficient	W								
	LPF_A12 (lo)	R								
	A12 filter coefficient	W								
0x9A	LPF_A13 (hi)	R	A13[15:0]							
	A13 filter coefficient	W								
	LPF_A13 (lo)	R								
	A13 filter coefficient	W								
0x9C	LPF_A14 (hi)	R	A14[15:0]							
	A14 filter coefficient	W								
	LPF_A14 (lo)	R								
	A14 filter coefficient	W								
0x9E	LPF_A15 (hi)	R	A15[15:0]							
	A15 filter coefficient	W								
	LPF_A15 (lo)	R								
	A15 filter coefficient	W								
0xA0	COMP_CTL (hi)	R	0	0	0	0	0	0	0	0
	Compensation control register	W	OPENEM		PGAZM	PGAOM	DIAGVM	DIAGIM	DIAGTM	CALIEM
	COMP_CTL (lo)	R	OPENE	0	PGAZ	PGAO	DIAGV	DIAGI	DIAGT	CALIE
	Compensation control register	W								
0xA2	COMP_SR	R	OPEN	BGRF	0	PGAOF	0	0	0	CALF
	Compensation status register	W	Write 1 will clear the flags							
0xA3	COMP_TF	R	0	IRSEL			ATGCE	TMF		
	Temperature filtering period	W								
0xA4	COMP_TMAX (hi)	R	TCMAX							
	Max. temp. before recalibration	W								
	COMP_TMAX (lo)	R	0	0	0	0	0	0	0	0
	Max. temp. before recalibration	W								
0xA6	COMP_TMIN (hi)	R	TCMIN							
	Min. temp. before recalibration	W								
	COMP_TMIN (lo)	R	0	0	0	0	0	0	0	0
	Min. temp. before recalibration	W								
0xA8		R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xAA	COMP_VO	R	VOC							
	Offset voltage compensation	W								
0xAB	COMP_IO	R	COC							
	Offset current compensation	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0xAC	COMP_VSG (hi)	R	0	0	0	0	0	0	VSGC	
	Gain voltage compensation vsense channel	W								
	COMP_VSG (lo)	R	VSGC							
	Gain voltage compensation vsense channel	W								
0xAE	COMP_TVSG (hi)	R	0	0	0	TVSGCP				
	Voltage gain temp compensation above 25 °C	W								
	COMP_TVSG (lo)	R	0	0	0	TVSGCN				
	Voltage gain temp compensation below 25 °C	W								
0xB0	COMP_IG4 (hi)	R	0	0	0	0	0	0	IGC4	
	Gain current compensation	W								
	COMP_IG4 (lo)	R	IGC4							
	Gain current compensation	W								
0xB2	COMP_TIG4 (hi)	R	0	0	0	TIGC4P				
	Gain current compensation above 25 °C	W								
	COMP_TIG4 (lo)	R	0	0	0	TIGC4N				
	Gain current compensation below 25 °C	W								
0xB4	COMP_IG16 (hi)	R	0	0	0	0	0	0	IGC16	
	Gain current compensation	W								
	COMP_IG16 (lo)	R	IGC16							
	Gain current compensation	W								
0xB6	COMP_TIG16 (hi)	R	0	0	0	TIGC16P				
	Gain current compensation above 25 °C	W								
	COMP_TIG16 (lo)	R	0	0	0	TIGC16N				
	Gain current compensation below 25 °C	W								
0xB8	COMP_IG64 (hi)	R	0	0	0	0	0	0	IGC64	
	Gain current compensation	W								
	COMP_IG64 (lo)	R	IGC64							
	Gain current compensation	W								
0xBA	COMP_TIG64 (hi)	R	0	0	0	TIGC64P				
	Gain current compensation above 25 °C	W								
	COMP_TIG64 (lo)	R	0	0	0	TIGC64N				
	Gain current compensation below 25 °C	W								
0xBC	COMP_IG256 (hi)	R	0	0	0	0	0	0	IGC256	
	Gain current compensation	W								
	COMP_IG256 (lo)	R	IGC256							
	Gain current compensation	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0xBE	COMP_TIG256 (hi)	R	0	0	0	TIGC256P				
	Current gain temp compensation above 25 °C	W								
	COMP_TIG256 (lo)	R	0	0	0	TIGC256N				
	Current gain temp compensation below 25 °C	W								
0xC0	COMP_PGAO4 (hi)	R	0	0	0	0	0	PGAOC4		
	Offset PGA compensation	W								
	PGAO4 (lo)	R	PGAOC4							
	Offset PGA compensation	W								
0xC2		R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xC4	COMP_PGAO16 (hi)	R	0	0	0	0	0	PGAOC16		
	Offset PGA compensation	W								
	COMP_PGAO16 (lo)	R	PGAOC16							
	Offset PGA compensation	W								
0xC6		R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xC8	COMP_PGAO64 (hi)	R	0	0	0	0	0	PGAOC64		
	Offset PGA compensation	W								
	COMP_PGAO64 (lo)	R	PGAOC64							
	Offset PGA compensation	W								
0xCA	Reserved	R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xCC	COMP_PGAO256 (hi)	R	0	0	0	0	0	PGAOC256		
	Offset PGA compensation	W								
	COMP_PGAO256 (lo)	R	PGAOC256							
	Offset PGA compensation	W								
0xCE		R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xD0	COMP_ITO	R	ITOC							
	Internal temp. offset compensation	W								
0xD1	COMP_ITG	R	ITGC							
	Internal temp. gain compensation	W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0xD2	COMP_ETO	R	ETOC							
	External temp. offset compensation	W								
0xD3	COMP_ETG	R	ETGC							
	External temp. gain compensation	W								
0xD4		R	0	0	0	0	0	0	0	0
		W								
0xD5		R	0	0	0	0	0	0	0	0
		W								
0xD6		R	0	0	0	0	0	0	0	0
		W								
0xD7		R	0	0	0	0	0	0	0	0
		W								
0xD8		R	0	0	0	0	0	0	0	0
		W								
0xD9		R	0	0	0	0	0	0	0	0
		W								
0xDA		R	0	0	0	0	0	0	0	0
		W								
0xDB		R	0	0	0	0	0	0	0	0
		W								
0xDC		R	0	0	0	0	0	0	0	0
		W								
0xDD		R	0	0	0	0	0	0	0	0
		W								
0xDE		R	0	0	0	0	0	0	0	0
		W								
0xDF		R	0	0	0	0	0	0	0	0
		W								
0xE0	TRIM_BG0 (hi)	R	IBG1[2:1]		TCIBG2			TCIBG1		
	Trim bandgap 0	W								
0xE1	TRIM_BG0 (lo)	R	IBG1[0]	LVT	TCBG2			TBG1		
	Trim bandgap 0	W								
0xE2	TRIM_BG1 (hi)	R	V1P2BG2				V2P2BG1			
	Trim bandgap 1	W								
0xE3	TRIM_BG1 (lo)	R	VDDXLPMODE				V2P5BG1			
	Trim bandgap 1	W								
0xE4	TRIM_OSC (hi)	R	SLPBG			LPOSC[12:8]				
	Trim LP oscillator	W								
0xE5	TRIM_OSC (lo)	R	LPOSC[7:0]							
	Trim LP oscillator	W								
0xE6		R	0	0	0	0	0	0	0	0
		W								
0xE7		R	0	0	0	0	0	0	0	0
		W								

MM9Z1_638

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0xE8		R	0	0	0	0	0	0	0	0
		W								
0xE9		R	0	0	0	0	0	0	0	0
		W								
0xEA		R	0	0	0	0	0	0	0	0
		W								
0xEB		R	0	0	0	0	0	0	0	0
		W								
0xEC		R	0	0	0	0	0	0	0	0
		W								
0xED		R	0	0	0	0	0	0	0	0
		W								
0xEE		R	0	0	0	0	0	0	0	0
		W								
0xEF		R	0	0	0	0	0	0	0	0
		W								
0xF0	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF1	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF2	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF3	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF4	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF5	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF6	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF7	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF8	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xF9	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFA	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFB	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFC	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFD	Reserved	R	0	0	0	0	0	0	0	0
		W								

Table 60. Analog die registers - 0x0E00–0x0EFF D2D blocking access (D2DI) 0x0F00–0x0FFF D2D non blocking access (D2DI)

Offset[73]	Name		7	6	5	4	3	2	1	0
0xFE	Reserved	R	0	0	0	0	0	0	0	0
		W								
0xFF	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes:
73.Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

Bit legend

= Unimplemented, Reserved

= Implemented (do not alter)

X = Indeterminate

0 = Always read zero

6.2 Clock, reset, and power management unit (S12ZCPMU + PCR)

Table 61. Modes and watchdogs.

MM9Z1_638				
Mode	S12Z die		Analog Die	
	Mode	COP watchdog (COP WD)	Mode	Window watchdog (WWD)
Normal	normal	run / or not	normal	run / or not
Sleep	power off	no	sleep	no
Stop	stop	run / or not	stop	no
	pseudo-stop	run / or not		

6.2.1 S12Z clock, reset and power management unit (S12ZCPMU)

6.2.1.1 Introduction

This specification describes the function of the Clock, Reset and Power Management Unit (S12ZCPMU).

- The Pierce oscillator (XOSCLCP) provides a robust, low-noise and low-power external clock source. It is designed for optimal start-up margin with typical crystal oscillators.
- The voltage regulator (IVREG) operates from the range 3.13 to 5.5 V. It provides all the required chip internal voltages and voltage monitors.
- The phase locked loop (PLL) provides a highly accurate frequency multiplier with internal filter.
- The internal reference clock (IRC1M) provides a 1.024 MHz internal clock.

6.2.1.2 Features

The Pierce Oscillator (XOSCLCP) contains circuitry to dynamically control current gain in the output amplitude. This ensures a signal with low harmonic distortion, low-power and good noise immunity.

- Supports crystals or resonators from 4.0 to 16.384 MHz
- High noise immunity due to input hysteresis and spike filtering
- Low RF emissions with peak-to-peak swing limited dynamically
- Transconductance (gm) sized for optimum start-up margin for typical crystals
- Dynamic gain control eliminates the need for external current limiting resistor
- Integrated resistor eliminates the need for external bias resistor
- Low-power consumption: Operates from internal 1.8 V (nominal) supply, amplitude control limits power

- Optional oscillator clock monitor reset
- Optional full swing mode for higher immunity against noise injection on the cost of higher power consumption and increased emission

The voltage regulator (IVREG) has the following features:

- Input voltage range from 3.13 to 5.5V
- Low-voltage detect (LVD) with low-voltage interrupt (LVI)
- Power-on reset (POR)
- Low-voltage reset (LVR)
- Voltage regulator providing Full Performance mode (FPM) and Reduced Performance mode (RPM)

The Phase Locked Loop (PLL) has the following features:

- Highly accurate and phase locked frequency multiplier
- Configurable internal filter for best stability and lock time
- Frequency modulation for defined jitter and reduced emission
- Automatic frequency lock detector
- Interrupt request on entry or exit from locked condition
- PLL clock monitor reset
- Reference clock either external (crystal) or internal square wave (1.024 MHz IRC1M) based.
- PLL stability is sufficient for LIN communication in slave mode, even if using IRC1M as reference clock

The Internal Reference Clock (IRC1M) has the following features:

- Frequency trimming
  (A factory trim value for 1.024 MHz is loaded from flash memory into the IRCTRIM register after reset, which can be overwritten by application if required)
- Temperature coefficient (TC) trimming.
  (A factory trim value is loaded from Flash Memory into the IRCTRIM register to turn off TC trimming after reset. Application can trim the TC if required by overwriting the IRCTRIM register).

Other features of the S12ZCPMU include

- Oscillator clock monitor to detect loss of crystal
- Autonomous periodical interrupt (API)
- Bus Clock Generator
  - Clock switch to select either PLLCLK or external crystal/resonator based bus clock
  - PLLCLK divider to adjust system speed
- System reset generation from the following possible sources:
  - Power-on reset (POR)
  - Low-voltage reset (LVR)
  - COP timeout
  - Loss of oscillation (oscillator clock monitor fail)
  - Loss of PLL clock (PLL clock monitor fail)
  - External pin $\overline{\text{RESET}}$

6.2.1.3 Modes of operation

This subsection lists and briefly describes all operating modes supported by the S12ZCPMU.

6.2.1.3.1 Run mode

The voltage regulator is in Full Performance mode (FPM).

Note

The voltage regulator is active, providing the nominal supply voltages with full current sourcing capability (see Electrical specification for voltage regulator). The features ACLK clock source, Low Voltage Interrupt (LVI), Low Voltage Reset (LVR) and Power-On Reset (POR) are available.

The phase locked loop (PLL) is on.

The internal reference clock (IRC1M) is on.

The API is available.

- PLL Engaged Internal (PEI)
  - This is the default mode after System Reset and Power-On Reset.
  - The Bus Clock is based on the PLLCLK.
  - After reset the PLL is configured for 50 MHz VCOCLK operation.
    Post divider is 0x03, so PLLCLK is VCOCLK divided by 4, that is 12.5 MHz and bus clock is 6.25 MHz.
    The PLL can be re-configured for other bus frequencies.
  - The reference clock for the PLL (REFCLK) is based on internal reference clock IRC1M.
- PLL engaged external (PEE)
  - The bus clock is based on the PLLCLK.
  - This mode can be entered from default mode PEI by performing the following steps:
    - Configure the PLL for desired bus frequency.
    - Program the reference divider (REFDIV[3:0] bits) to divide down oscillator frequency if necessary.
    - Enable the external oscillator (OSCE bit).
    - Wait for oscillator to start-up (UPOSC=1) and PLL to lock (LOCK=1).
- PLL bypassed external (PBE)
  - The bus clock is based on the oscillator clock (OSCCLK).
  - The PLLCLK is always on to qualify the external oscillator clock. Therefore it is necessary to make sure a valid PLL configuration is used for the selected oscillator frequency.
  - This mode can be entered from default mode PEI by performing the following steps:
    - Make sure the PLL configuration is valid for the selected oscillator frequency.
    - Enable the external oscillator (OSCE bit).
    - Wait for oscillator to start up (UPOSC=1).
    - Select the oscillator clock (OSCCLK) as bus clock (PLLSEL=0).
  - The PLLCLK is on and used to qualify the external oscillator clock.

6.2.1.3.2 Wait mode

For S12ZCPMU Wait mode is the same as Run mode.

6.2.1.3.3 Stop mode

This mode is entered by executing the CPU STOP instruction.

The voltage regulator is in Reduced Performance mode (RPM).

Note

The voltage regulator output voltage may degrade to a lower value than in Full Performance mode (FPM), additionally the current sourcing capability is substantially reduced (see Electrical specification for voltage regulator). Only clock source ACLK is available and the Power-On-Reset (POR) circuitry is functional. The low-voltage interrupt (LVI) and low-voltage reset (LVR) are disabled.

The API is available.

The phase locked loop (PLL) is off.

The internal reference clock (IRC1M) is off.

Core clock and bus clock are stopped.

Depending on the setting of the PSTP and the OSCE bit, Stop mode can be differentiated between Full Stop mode (PSTP = 0 or OSCE=0) and Pseudo Stop mode (PSTP = 1 and OSCE=1). In addition, the behavior of the COP in each mode changes based on the clocking method selected by COPOSCSEL[1:0].

- **Full Stop mode (PSTP = 0 or OSCE=0)**
  External oscillator (XOSCLCP) is disabled.
  - If COPOSCSEL1=0:
    The COP and RTI counters halt during Full Stop mode.
    After wake-up from Full Stop mode the core clock and bus clock are running on PLLCLK (PLLSEL=1). COP and RTI are running on IRCCLK (COPOSCSEL0=0, RTIOSCSEL=0).
  - If COPOSCSEL1=1:
    The clock for the COP is derived from ACLK (trimmable internal RC-Oscillator clock). During Full Stop mode the ACLK for the COP can be stopped (COP static) or running (COP active) depending on the setting of bit CSAD. When bit CSAD is set the ACLK clock source for the COP is stopped during Full Stop mode and COP continues to operate after exit from Full Stop mode. For this

COP configuration (ACLK clock source, CSAD set) a latency time occurs when entering or exiting (Full, Pseudo) Stop mode. When bit CSAD is clear the ACLK clock source is on for the COP during Full Stop mode and COP is operating.
During Full Stop mode the RTI counter halts.
After wake-up from Full Stop mode the core clock and bus clock are running on PLLCLK (PLLSEL=1). The COP runs on ACLK and RTI is running on IRCCLK (COPOSCSEL0=0, RTIOSCSEL=0).

- **Pseudo Stop mode (PSTP = 1 and OSCE=1)**
  External oscillator (XOSCLCP) continues to run.
  — If COPOSCSEL1=0:
  If the respective enable bits are set (PCE=1 and PRE=1) the COP and RTI continues to run with a clock derived from the oscillator clock.
  The clock configuration bits PLLSEL, COPOSCSEL0, RTIOSCSEL are unchanged.
  — If COPOSCSEL1=1:
  If the respective enable bit for the RTI is set (PRE=1) the RTI continues to run with a clock derived from the oscillator clock.
  The clock for the COP is derived from ACLK (trimmable internal RC-oscillator clock). During Pseudo Stop mode the ACLK for the COP can be stopped (COP static) or running (COP active) depending on the setting of bit CSAD. When bit CSAD is set the ACLK for the COP is stopped during Pseudo Stop mode and COP continues to operate after exit from Pseudo Stop mode.
  For this COP configuration (ACLK clock source, CSAD set) a latency time occurs when entering or exiting (Pseudo, Full) Stop mode. When bit CSAD is clear the ACLK clock source is on for the COP during Pseudo Stop mode and COP is operating.
  The clock configuration bits PLLSEL, COPOSCSEL0, RTIOSCSEL are unchanged.

Note

When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the start-up time of the external oscillator t_{UPOSC} before entering Pseudo Stop mode.

6.2.1.3.4 Freeze mode (BDM active)

For S12ZCPMU Freeze mode is the same as Run mode except for RTI and COP which can be stopped in Active BDM mode with the RSBCK bit in the CPMUCOP register. Additionally the COP can be forced to the maximum timeout period in Active BDM mode. For details see the RSBCK and CR[2:0] bit description field of Table 85 in S12ZCPMU COP control register (CPMUCOP)

6.2.1.4 S12ZCPMU block diagram

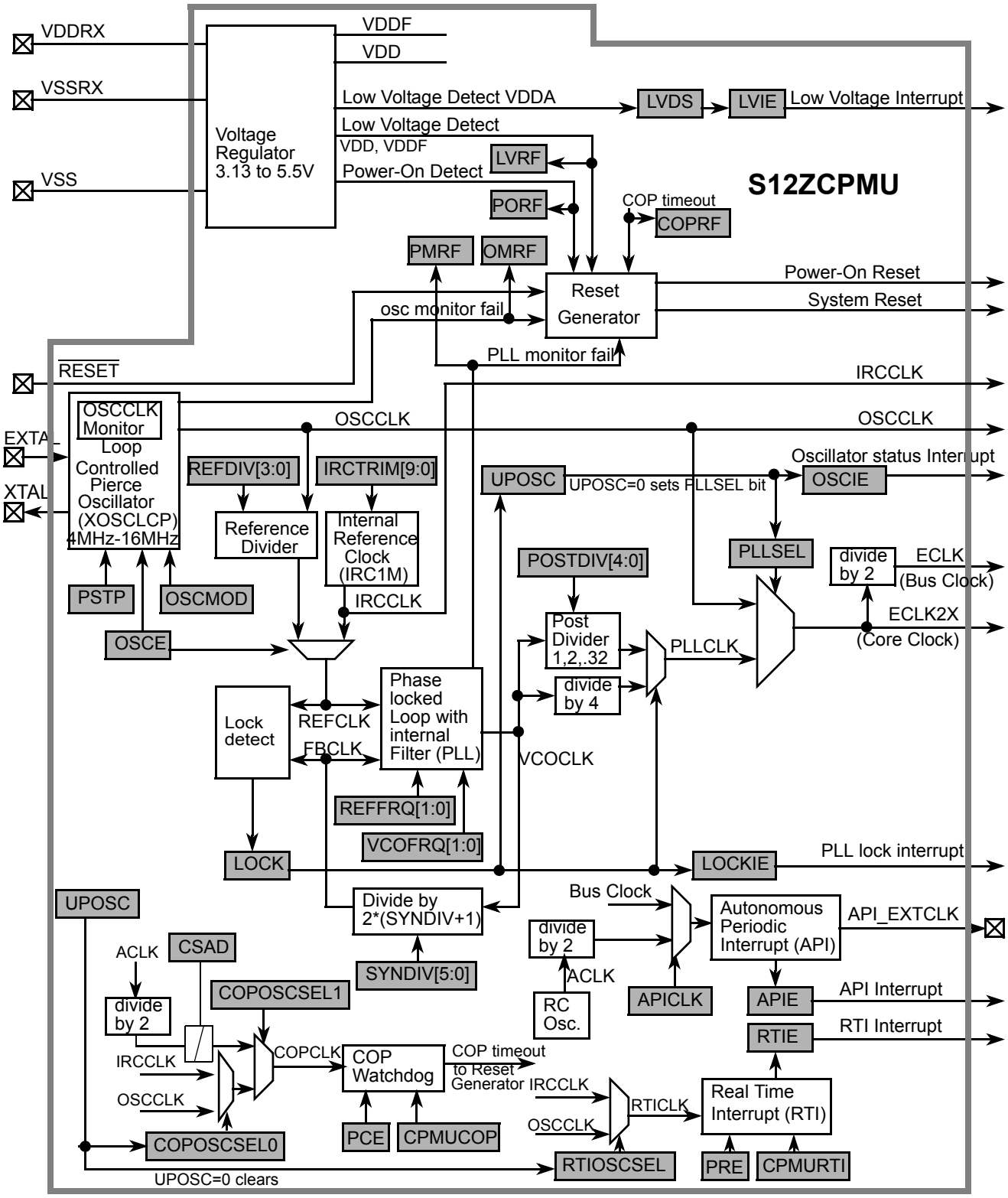


Figure 19. Block diagram of S12ZCPMU

Figure 20 shows a block diagram of the XOSCLCP.

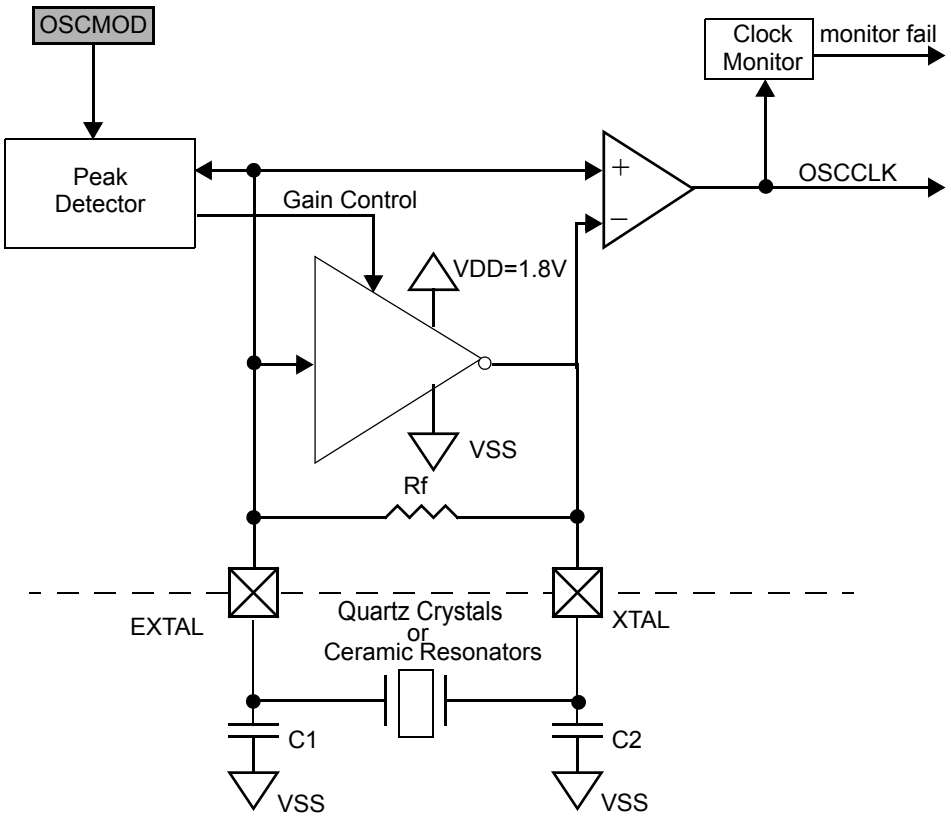


Figure 20. XOSCLCP block diagram

6.2.1.5 Signal description

This section lists and describes the signals that connect off chip as well as internal supply nodes and special signals.

6.2.1.5.1 RESET

The $\overline{\text{RESET}}$ pin is an active-low bidirectional pin. As an input it initializes the MCU asynchronously to a known start-up state. As an open-drain output it indicates that an MCU-internal reset has been triggered.

6.2.1.5.2 EXTAL and XTAL

These pins provide the interface for a crystal to control the internal clock generator circuitry. EXTAL is the input to the crystal oscillator amplifier. XTAL is the output of the crystal oscillator amplifier. If XOSCLCP is enabled, the MCU internal OSCCLK is derived from the EXTAL input frequency. If OSCE=0, the EXTAL pin is pulled down by an internal resistor of approximately 200 kΩ and the XTAL pin is pulled down by an internal resistor of approximately 700 kΩ.

Note

NXP recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier. The XOSCLCP is not suited for overtone resonators and crystals.

6.2.1.5.3 VDDRX, VSSRX— regulator power input pin and 5V supply pins

VDDRX is the power input of IVREG and the PAD positive supply Pin.

All currents sourced into the regulator loads flow through this pin.

The VDDRX/VSSX supply domain is monitored by the Low Voltage Reset circuit.

An off-chip decoupling capacitor (100 nF... 220 nF, X7R ceramic) between VDDRX and VSSX can further improve the quality of this supply.

6.2.1.5.4 VSS — core logic ground pin

VSS is the logic supply return pin. It must be grounded.

6.2.1.5.5 VDD — internal regulator output supply (core logic)

Node VDD is a device internal supply output of the voltage regulator that provides the power supply for the core logic.

This supply domain is monitored by the Low Voltage Reset circuit.

6.2.1.5.6 VDDF — internal regulator output supply (NVM logic)

Node VDDF is a device internal supply output of the voltage regulator that provides the power supply for the NVM logic.

This supply domain is monitored by the Low Voltage Reset circuit

6.2.1.5.7 API_EXTCLK — API external clock output pin

This pin provides the signal selected via APIES and is enabled with APIEA bit. See the device specification if this clock output is available on this device and to which pin it might be connects.

6.2.1.6 Memory map and registers

This section provides a detailed description of all registers accessible in the S12ZCPMU.

6.2.1.6.1 Module memory map

The S12ZCPMU registers are shown in Table 62.

Table 62. CPMU register summary

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000	CPMU RESERVED00	R	0	0	0	0	0	0	0	0
		W								
0x0001	CPMU RESERVED01	R	0	0	0	0	0	0	0	0
		W								
0x0002	CPMU RESERVED02	R	0	0	0	0	0	0	0	0
		W								
0x0003	CPMURFLG	R	0	PORF	LVRF	0	COPRF	0	OMRF	PMRF
		W								
0x0004	CPMU SYNR	R	VCOFRQ[1:0]		SYNDIV[5:0]					
		W								
0x0005	CPMU REFDIV	R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
		W								
0x0006	CPMU POSTDIV	R	0	0	0	POSTDIV[4:0]				
		W								
0x0007	CPMUIFLG	R	RTIF	0	0	LOCKIF	LOCK	0	OSCIF	UPOSC
		W								
0x0008	CPMUINT	R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
		W								
0x0009	CPMUCLKS	R	PLLSEL	PSTP	CSAD	COP OSCSEL1	PRE	PCE	RTI OSCSEL	COP OSCSEL0
		W								

= Unimplemented or Reserved

MM9Z1_638

Table 62. CPMU register summary (continued)

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x000A	CPMUPLL	R	0	0	FM1	FM0	0	0	0	0
		W								
0x000B	CPMURTI	R	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
		W								
0x000C	CPMUCOP	R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
		W			WRTMASK					
0x000D	RESERVED CPMUTEST0	R	0	0	0	0	0	0	0	0
		W								
0x000E	RESERVED CPMUTEST1	R	0	0	0	0	0	0	0	0
		W								
0x000F	CPMU ARMCOP	R	0	0	0	0	0	0	0	0
		W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x0010	CPMU RESERVED10	R	0	0	0	0	0	0	0	0
		W								
0x0011	CPMU LVCTL	R	0	0	0	0	0	LVDS	LVIE	LVIF
		W								
0x0012	CPMU APICTL	R	APICLK	0	0	APIES	APIEA	APIFE	APIE	APIF
		W								
0x0013	CPMUACLKTR	R	ACLKTR5	ACLKTR4	ACLKTR3	ACLKTR2	ACLKTR1	ACLKTR0	0	0
		W								
0x0014	CPMUAPIRH	R	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8
		W								
0x0015	CPMUAPIRL	R	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0
		W								
0x0016	RESERVED CPMUTEST3	R	0	0	0	0	0	0	0	0
		W								
0x0017	CPMU RESERVED17	R	0	0	0	0	0	0	0	0
		W								
0x0018	CPMU IRCTRIMH	R	TCTRIM[4:0]					0	IRCTRIM[9:8]	
		W								
0x0019	CPMU IRCTRIML	R	IRCTRIM[7:0]							
		W								
0x001A	CPMUOSC	R	OSCE	0	Reserved	0	0	0	0	0
		W								
0x001B	CPMUPROT	R	0	0	0	0	0	0	0	PROT
		W								
0x001C	RESERVED CPMUTEST2	R	0	0	0	0	0	0	0	0
		W								
0x001D	CPMU RESERVED1D	R	0	0	0	0	0	0	0	0
		W								
0x001E	CPMUOSC2	R	0	0	0	0	0	0	OMRE	OSCMOD
		W								
			= Unimplemented or Reserved							

Table 62. CPMU register summary (continued)

Address Offset	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x001F	CPMU RESERVED1F	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented or Reserved

6.2.1.6.2 Register descriptions

This section describes all the S12ZCPMU registers and their individual bits.

Address order is as listed in Table 62

6.2.1.6.2.1 S12ZCPMU reset flags register (CPMURFLG)

This register provides S12ZCPMU reset flags.

Table 63. S12ZCPMU flags register (CPMURFLG)

Module Base + 0x0003

	7	6	5	4	3	2	1	0
R	0	PORF	LVRF	0	COPRF	0	OMRF	PMRF
W								
Reset	0	(74)	(75)	0	(76)	0	(77)	(78)

= Unimplemented or Reserved

Notes:
74.PORF is set to 1 when a power on reset occurs. Unaffected by System Reset.
75.LVRF is set to 1 when a low voltage reset occurs. Unaffected by System Reset. Set by power on reset.
76.COPRF is set to 1 when COP reset occurs. Unaffected by System Reset. Cleared by power on reset.
77.OMRF is set to 1 when an oscillator clock monitor reset occurs. Unaffected by System Reset. Cleared by power on reset.
78.PMRF is set to 1 when a PLL clock monitor reset occurs. Unaffected by System Reset. Cleared by power on reset.
79.Read: Anytime
Write: Refer to each bit for individual write conditions

Table 64. CPMURFLG field descriptions

Field	Description
6 PORF	**Power on Reset Flag** — PORF is set to 1 when a power on reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Power on reset has not occurred. 1 Power on reset has occurred.
5 LVRF	**Low Voltage Reset Flag** — LVRF is set to 1 when a low voltage reset occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Low voltage reset has not occurred. 1 Low voltage reset has occurred.
3 COPRF	**COP Reset Flag** — COPRF is set to 1 when a COP (computer operating properly) reset occurs. Refer to Computer operating properly watchdog (COP) reset and S12ZCPMU COP control register (CPMUCOP) for details.This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 COP reset has not occurred. 1 COP reset has occurred.

Table 64. CPMURFLG field descriptions (continued)

Field	Description
1 OMRF	**Oscillator Clock Monitor Reset Flag** — OMRF is set to 1 when a loss of oscillator (crystal) clock occurs and the oscillator clock monitor reset is enabled (OMRE bit in CPMUOSC2 register). Refer to Oscillator clock monitor reset for details.This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Loss of oscillator clock reset has not occurred. 1 Loss of oscillator clock reset has occurred.
0 PMRF	**PLL Clock Monitor Reset Flag** — PMRF is set to 1 when a loss of oscillator (crystal) clock occurs. This flag can only be cleared by writing a 1. Writing a 0 has no effect. 0 Loss of PLL clock reset has not occurred. 1 Loss of PLL clock reset has occurred.

6.2.1.6.2.2 S12ZCPMU synthesizer register (CPMUSYNR)

The CPMUSYNR register controls the multiplication factor of the PLL and selects the VCO frequency range.

Table 65. S12ZCPMU synthesizer register (CPMUSYNR)

Module Base + 0x0004

	7	6	5	4	3	2	1	0
R W	VCOFRQ[1:0]		SYNDIV[5:0]					
Reset	0	1	0	1	1	0	0	0

Notes:

80.Read: Anytime

Write: If PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register), then write anytime. Else write has no effect.

Note

Writing to this register clears the LOCK and UPOSC status bits.

If PLL has locked (LOCK=1) $$f_{VCO} = 2 \times f_{REF} \times (SYNDIV + 1)$$

Note

f_{VCO} must be within the specified VCO frequency lock range. Bus frequency f_{BUS} must not exceed the specified maximum.

The VCOFRQ[1:0] bits are used to configure the VCO gain for optimal stability and lock time. For correct PLL operation the VCOFRQ[1:0] bits have to be selected according to the actual target VCOCLK frequency as shown in Table 66. Setting the VCOFRQ[1:0] bits incorrectly can result in a non functional PLL (no locking and/or insufficient stability).

Table 66. VCO clock frequency selection

VCOCLK frequency ranges	VCOFRQ[1:0]
32 MHz ≤ f_{VCO} ≤ 48 MHz	00
48 MHz < f_{VCO} ≤ 80 MHz	01
Reserved	10
80 MHz < f_{VCO} ≤ 100 MHz	11

6.2.1.6.2.3 S12ZCPMU reference divider register (CPMUREFDIV)

The CPMUREFDIV register provides a finer granularity for the PLL multiplier steps when using the external oscillator as reference.

Table 67. S12ZCPMU reference divider register (CPMUREFDIV)

Module Base + 0x0005

	7	6	5	4	3	2	1	0
R	REFFRQ[1:0]		0	0	REFDIV[3:0]			
W								
Reset	0	0	0	0	1	1	1	1

Notes:

81.Read: Anytime

Write: If PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register), then write anytime. Else write has no effect.

Note

Write to this register clears the LOCK and UPOSC status bits.

If XOSCLCP is enabled (OSCE=1) $$f_{REF} = \frac{f_{OSC}}{(REFDIV + 1)}$$

If XOSCLCP is disabled (OSCE=0) $$f_{REF} = f_{IRC1M}$$

The REFFRQ[1:0] bits are used to configure the internal PLL filter for optimal stability and lock time. For correct PLL operation the REFFRQ[1:0] bits have to be selected according to the actual REFCLK frequency as shown in Table 68.

If IRC1M is selected as REFCLK (OSCE=0) the PLL filter is fixed configured for the 1.0 MHz ≤ f_{REF} ≤ 2.0 MHz range. The bits can still be written but will have no effect on the PLL filter configuration.

For OSCE=1, setting the REFFRQ[1:0] bits incorrectly can result in a non functional PLL (no locking and/or insufficient stability).

Table 68. Reference clock frequency selection if XOSCLCP is enabled

REFCLK frequency ranges (OSCE=1)	REFFRQ[1:0]
1.0 MHz ≤ f_{REF} ≤ 2.0 MHz	00
2.0 MHz < f_{REF} ≤ 6.0 MHz	01
6.0 MHz < f_{REF} ≤ 12 MHz	10
f_{REF} > 12 MHz	11

6.2.1.6.2.4 S12ZCPMU post divider register (CPMUPOSTDIV)

The POSTDIV register controls the frequency ratio between the VCOCLK and the PLLCLK.

Table 69. S12ZCPMU post divider register (CPMUPOSTDIV)

Module Base + 0x0006

	7	6	5	4	3	2	1	0
R	0	0	0	POSTDIV[4:0]				
W								
Reset	0	0	0	0	0	0	1	1

= Unimplemented or Reserved

Notes:

82.Read: Anytime

Write: If PLLSEL=1 write anytime, else write has no effect.

If PLL is locked (LOCK=1) $f_{PLL} = \frac{f_{VCO}}{(POSTDIV + 1)}$

If PLL is not locked (LOCK=0) $f_{PLL} = \frac{f_{VCO}}{4}$

If PLL is selected (PLLSEL=1) $f_{bus} = \frac{f_{PLL}}{2}$

When changing the POSTDIV[4:0] value or PLL transitions to locked stated (lock = 1), it takes up to 32 bus clock cycles until f_{PLL} is at the desired target frequency. This is because the post divider gradually changes (increases or decreases) f_{PLL} to avoid sudden load changes for the on-chip voltage regulator.

6.2.1.6.2.5 S12ZCPMU Interrupt Flags Register (CPMUIFLG)

This register provides S12ZCPMU status bits and interrupt flags.

Table 70. S12ZCPMU flags register (CPMUIFLG)

Module base + 0x0007

	7	6	5	4	3	2	1	0
R	RTIF	0	0	LOCKIF	LOCK	0	OSCIF	UPOSC
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:

83.Read: Anytime

Write: Refer to each bit for individual write conditions.

Table 71. CPMUIFLG field descriptions

Field	Description
7 RTIF	**Real Time Interrupt Flag** — RTIF is set to 1 at the end of the RTI period. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (RTIE=1), RTIF causes an interrupt request. 0 RTI timeout has not yet occurred. 1 RTI timeout has occurred.
4 LOCKIF	**PLL Lock Interrupt Flag** — LOCKIF is set to 1 when LOCK status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LOCKIE=1), LOCKIF causes an interrupt request. 0 No change in LOCK bit. 1 LOCK bit has changed.
3 LOCK	**Lock Status Bit** — LOCK reflects the current state of PLL lock condition. Writes have no effect. While PLL is unlocked (LOCK=0) f_{PLL} is f_{VCO} / 4 to protect the system from high core clock frequencies during the PLL stabilization time t_{LOCK}. 0VCOCLK is not within the desired tolerance of the target frequency. $f_{PLL} = f_{VCO}/4$. 1 VCOCLK is within the desired tolerance of the target frequency. $f_{PLL} = f_{VCO}/(POSTDIV+1)$.
1 OSCIF	**Oscillator Interrupt Flag** — OSCIF is set to 1 when UPOSC status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (OSCIE=1), OSCIF causes an interrupt request. 0 No change in UPOSC bit. 1 UPOSC bit has changed.
0 UPOSC	**Oscillator Status Bit** — UPOSC reflects the status of the oscillator. Writes have no effect. Entering Full Stop mode UPOSC is cleared. 0 The oscillator is off or oscillation is not qualified by the PLL. 1 The oscillator is qualified by the PLL.

6.2.1.6.2.6 S12ZCPMU interrupt enable register (CPMUINT)

This register enables S12ZCPMU interrupt requests.

Table 72. S12ZCPMU interrupt enable register (CPMUINT)

Module base + 0x0008

	7	6	5	4	3	2	1	0
R	RTIE	0	0	LOCKIE	0	0	OSCIE	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:

84.Read: Anytime
Write: Anytime

Table 73. CPMUINT field descriptions

Field	Description
7 RTIE	**Real Time Interrupt Enable Bit** 0 Interrupt requests from RTI are disabled. 1 Interrupt will be requested whenever RTIF is set.
4 LOCKIE	**PLL Lock Interrupt Enable Bit** 0 PLL LOCK interrupt requests are disabled. 1 Interrupt will be requested whenever LOCKIF is set.
1 OSCIE	**Oscillator Corrupt Interrupt Enable Bit** 0 Oscillator Corrupt interrupt requests are disabled. 1 Interrupt will be requested whenever OSCIF is set.

6.2.1.6.2.7 S12ZCPMU clock select register (CPMUCLKS)

This register controls S12ZCPMU clock selection.

Table 74. S12ZCPMU clock select register (CPMUCLKS)

Module Base + 0x0009

	7	6	5	4	3	2	1	0
R	PLLSEL	PSTP	CSAD	COP OSCSEL1	PRE	PCE	RTI OSCSEL	COP OSCSEL0
W								
Reset	1	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:

85.Read: Anytime
Write: Anytime

- Only possible if PROT=0 (CPMUPROT register) in all MCU Modes (Normal and Special mode).
- All bits in Special mode (if PROT=0).
- PLLSEL, PSTP, PRE, PCE, RTIOSCSEL: In Normal mode (if PROT=0).
- CSAD: In Normal mode (if PROT=0) until CPMUCOP write once has taken place.
- COPOSCSEL0: In Normal mode (if PROT=0) until CPMUCOP write once has taken place.
  If COPOSCSEL0 was cleared by UPOSC=0 (entering Full Stop mode with COPOSCSEL0=1 or insufficient OSCCLK quality), then COPOSCSEL0 can be set once again.
- COPOSCSEL1: In Normal mode (if PROT=0) until CPMUCOP write once has taken place. COPOSCSEL1 will not be cleared by UPOSC=0 (entering Full Stop mode with COPOSCSEL1=1 or insufficient OSCCLK quality if OSCCLK is used as clock source for other clock domains: for instance core clock etc.).

Note

After writing CPMUCLKS register, it is strongly recommended to read back CPMUCLKS register to make sure that write of PLLSEL, RTIOSCSEL and COPOSCSEL was successful. This is because under certain circumstances writes have no effect or bits are automatically changed (see Table 75).

Note

When using the oscillator clock as system clock (write PLLSEL = 0) it is highly recommended to enable the oscillator clock monitor reset feature (write OMRE = 1 in CPMUOSC2 register). If the oscillator monitor reset feature is disabled (OMRE = 0) and the oscillator clock is used as system clock, the system will stall in case of loss of oscillation.

Table 75. CPMUCLKS descriptions

Field	Description
7 PLLSEL	**PLL Select Bit** This bit selects the PLLCLK as source of the System Clocks (Core Clock and Bus Clock). PLLSEL can only be set to 0, if UPOSC=1. UPOSC = 0 sets the PLLSEL bit. Entering Full Stop mode sets the PLLSEL bit. 0 System clocks are derived from OSCCLK if oscillator is up (UPOSC = 1, $f_{BUS} = f_{OSC}/2$). 1 System clocks are derived from PLLCLK, $f_{BUS} = f_{PLL}/2$.
6 PSTP	**Pseudo Stop Bit** This bit controls the functionality of the oscillator during Stop mode. 0 Oscillator is disabled in Stop mode (Full Stop mode). 1 Oscillator continues to run in Stop mode (Pseudo Stop mode), option to run RTI and COP. Pseudo Stop mode allows for faster STOP recovery and reduces the mechanical stress and aging of the resonator in case of frequent STOP conditions at the expense of a slightly increased power consumption. When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the start-up time of the external oscillator t_{UPOSC} before entering Pseudo Stop mode.
4 CSAD	**COP in Stop mode ACLK Disable** — This bit disables the ACLK for the COP in Stop mode. Hence the COP is static while in Stop mode and continues to operate after exit from Stop mode. Due to clock domain crossing synchronization there is a latency time to enter and exit Stop mode if COP clock source is ACLK and this clock is stopped in Stop mode. This maximum latency time is 4 ACLK cycles which must be added to the Stop mode recovery time t_{STP_REC} from exit of current Stop mode to entry of next Stop mode. This latency time occurs no matter which Stop mode (Full, Pseudo) is currently exited or entered next. After exit from Stop mode (Pseudo, Full) for 2 ACLK cycles no Stop mode request (STOP instruction) should be generated to make sure the COP counter increments at each Stop mode exit. This bit does not influence the ACLK for the API. 0 COP running in Stop mode (ACLK for COP enabled in Stop mode). 1 COP stopped in Stop mode (ACLK for COP disabled in Stop mode)
4 COP OSCSEL1	**COP Clock Select 1** — COPOSCSEL0 and COPOSCSEL1 combined determine the clock source to the COP (see also **Table 76**). If COPOSCSEL1 = 1, COPOSCSEL0 has no effect regarding clock select and changing the COPOSCSEL0 bit does not re-start the COP timeout period. COPOSCSEL1 selects the clock source to the COP to be either ACLK (derived from trimmable internal RC-Oscillator) or clock selected via COPOSCSEL0 (IRCCLK or OSCCLK). Changing the COPOSCSEL1 bit re-starts the COP timeout period. COPOSCSEL1 can be set independent from value of UPOSC. UPOSC = 0 does not clear the COPOSCSEL1 bit. 0 COP clock source defined by COPOSCSEL0 1 COP clock source is ACLK derived from a trimmable internal RC-Oscillator
3 PRE	**RTI Enable During Pseudo Stop Bit** — PRE enables the RTI during Pseudo Stop mode. 0 RTI stops running during Pseudo Stop mode. 1 RTI continues running during Pseudo Stop mode if RTIOSCSEL=1. If PRE = 0 or RTIOSCSEL = 0 then the RTI will go static while Stop mode is active. The RTI counter will not be reset.
2 PCE	**COP Enable During Pseudo Stop Bit** — PCE enables the COP during Pseudo Stop mode. 0 COP stops running during Pseudo Stop mode 1 COP continues running during Pseudo Stop mode if COPOSCSEL = 1 If PCE = 0 or COPOSCSEL = 0 then the COP will go static while Stop mode is active. The COP counter will not be reset.

Table 75. CPMUCLKS descriptions (continued)

Field	Description
1 RTIOSCSEL	**RTI Clock Select**— RTIOSCSEL selects the clock source to the RTI. Either IRCCLK or OSCCLK. Changing the RTIOSCSEL bit re-starts the RTI timeout period. RTIOSCSEL can only be set to 1, if UPOSC = 1. UPOSC = 0 clears the RTIOSCSEL bit. 0 RTI clock source is IRCCLK. 1 RTI clock source is OSCCLK.
0 COP OSCSEL0	**COP Clock Select 0** — COPOSCSEL0 and COPOSCSEL1 combined determine the clock source to the COP (see also Table 76) If COPOSCSEL1 = 1, COPOSCSEL0 has no effect regarding clock select and changing the COPOSCSEL0 bit does not re-start the COP timeout period. When COPOSCSEL1=0,COPOSCSEL0 selects the clock source to the COP to be either IRCCLK or OSCCLK. Changing the COPOSCSEL0 bit re-starts the COP timeout period. COPOSCSEL0 can only be set to 1, if UPOSC = 1. UPOSC = 0 clears the COPOSCSEL0 bit. 0 COP clock source is IRCCLK. 1 COP clock source is OSCCLK

Table 76. COPOSCSEL1, COPOSCSEL0 clock source

COPOSCSEL1	COPOSCSEL0	COP clock source
0	0	IRCCLK
0	1	OSCCLK
1	x	ACLK

6.2.1.6.2.8 S12ZCPMU PLL control register (CPMUPLL)

This register controls the PLL functionality.

Table 77. S12ZCPMU PLL control register (CPMUPLL)

Module Base + 0x000A

	7	6	5	4	3	2	1	0
R	0	0	FM1	FM0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:

86.Read: Anytime

Write: Anytime if PROT=0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect.

Note

Write to this register clears the LOCK and UPOSC status bits.

Note

Care should be taken to ensure that the bus frequency does not exceed the specified maximum when frequency modulation is enabled.

Table 78. CPMUPLL field descriptions

Field	Description
5, 4 FM1, FM0	PLL **Frequency Modulation Enable Bits** — FM1 and FM0 enable frequency modulation on the VCOCLK. This is to reduce noise emission. The modulation frequency is f_{ref} divided by 16. See Table 79 for coding.

Table 79. FM amplitude selection

FM1	FM0	FM Amplitude / f_{VCO} Variation
0	0	FM off
0	1	±1%
1	0	±2%
1	1	±4%

6.2.1.6.2.9 S12ZCPMU RTI control register (CPMURTI)

This register selects the timeout period for the Real Time Interrupt.

The clock source for the RTI is either IRCCLK or OSCCLK depending on the setting of the RTIOSCSEL bit. In Stop mode with PSTP=1 (Pseudo Stop mode) and RTIOSCSEL=1 the RTI continues to run, else the RTI counter halts in Stop mode.

Table 80. S12ZCPMU RTI control register (CPMURTI)

Module Base + 0x000B

	7	6	5	4	3	2	1	0
R W	RTDEC	RTR6	RTR5	RTR4	RTR3	RTR2	RTR1	RTR0
Reset	0	0	0	0	0	0	0	0

Notes:
87.Read: Anytime
Write: Anytime

Note

A write to this register starts the RTI timeout period. A change of the RTIOSCSEL bit (writing a different value or loosing UPOSC status) re-starts the RTI timeout period.

Table 81. CPMURTI field descriptions

Field	Description
7 RTDEC	**Decimal or Binary Divider Select Bit** — RTDEC selects decimal or binary based prescaler values. 0 Binary based divider value. See Table 82 1 Decimal based divider value. See Table 83
6–4 RTR[6:4]	**Real Time Interrupt Prescale Rate Select Bits** — These bits select the prescale rate for the RTI.See Table 82 and Table 83.
3–0 RTR[3:0]	**Real Time Interrupt Modulus Counter Select Bits** — These bits select the modulus counter target value to provide additional granularity.Table 82 and Table 83 show all possible divide values selectable by the CPMURTI register.

Table 82. RTI frequency divide rates for RTDEC = 0

RTR[3:0]	RTR[6:4] =							
	000 (OFF)	001 (2^{10})	010 (2^{11})	011 (2^{12})	100 (2^{13})	101 (2^{14})	110 (2^{15})	111 (2^{16})
0000 (÷1)	OFF[88]	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}
0001 (÷2)	OFF	$2x2^{10}$	$2x2^{11}$	$2x2^{12}$	$2x2^{13}$	$2x2^{14}$	$2x2^{15}$	$2x2^{16}$
0010 (÷3)	OFF	$3x2^{10}$	$3x2^{11}$	$3x2^{12}$	$3x2^{13}$	$3x2^{14}$	$3x2^{15}$	$3x2^{16}$
0011 (÷4)	OFF	$4x2^{10}$	$4x2^{11}$	$4x2^{12}$	$4x2^{13}$	$4x2^{14}$	$4x2^{15}$	$4x2^{16}$
0100 (÷5)	OFF	$5x2^{10}$	$5x2^{11}$	$5x2^{12}$	$5x2^{13}$	$5x2^{14}$	$5x2^{15}$	$5x2^{16}$
0101 (÷6)	OFF	$6x2^{10}$	$6x2^{11}$	$6x2^{12}$	$6x2^{13}$	$6x2^{14}$	$6x2^{15}$	$6x2^{16}$

Table 82. RTI frequency divide rates for RTDEC = 0 (continued)

RTR[3:0]	RTR[6:4] =							
	000 (OFF)	001 (2^{10})	010 (2^{11})	011 (2^{12})	100 (2^{13})	101 (2^{14})	110 (2^{15})	111 (2^{16})
0110 (÷7)	OFF	$7x2^{10}$	$7x2^{11}$	$7x2^{12}$	$7x2^{13}$	$7x2^{14}$	$7x2^{15}$	$7x2^{16}$
0111 (÷8)	OFF	$8x2^{10}$	$8x2^{11}$	$8x2^{12}$	$8x2^{13}$	$8x2^{14}$	$8x2^{15}$	$8x2^{16}$
1000 (÷9)	OFF	$9x2^{10}$	$9x2^{11}$	$9x2^{12}$	$9x2^{13}$	$9x2^{14}$	$9x2^{15}$	$9x2^{16}$
1001 (÷10)	OFF	$10x2^{10}$	$10x2^{11}$	$10x2^{12}$	$10x2^{13}$	$10x2^{14}$	$10x2^{15}$	$10x2^{16}$
1010 (÷11)	OFF	$11x2^{10}$	$11x2^{11}$	$11x2^{12}$	$11x2^{13}$	$11x2^{14}$	$11x2^{15}$	$11x2^{16}$
1011 (÷12)	OFF	$12x2^{10}$	$12x2^{11}$	$12x2^{12}$	$12x2^{13}$	$12x2^{14}$	$12x2^{15}$	$12x2^{16}$
1100 (÷13)	OFF	$13x2^{10}$	$13x2^{11}$	$13x2^{12}$	$13x2^{13}$	$13x2^{14}$	$13x2^{15}$	$13x2^{16}$
1101 (÷14)	OFF	$14x2^{10}$	$14x2^{11}$	$14x2^{12}$	$14x2^{13}$	$14x2^{14}$	$14x2^{15}$	$14x2^{16}$
1110 (÷15)	OFF	$15x2^{10}$	$15x2^{11}$	$15x2^{12}$	$15x2^{13}$	$15x2^{14}$	$15x2^{15}$	$15x2^{16}$
1111 (÷16)	OFF	$16x2^{10}$	$16x2^{11}$	$16x2^{12}$	$16x2^{13}$	$16x2^{14}$	$16x2^{15}$	$16x2^{16}$

Notes:
88.Denotes the default value out of reset.This value should be used to disable the RTI to ensure future backwards compatibility.

Table 83. RTI frequency divide rates for RTDEC=1

RTR[3:0]	RTR[6:4] =							
	000 ($1x10^3$)	001 ($2x10^3$)	010 ($5x10^3$)	011 ($10x10^3$)	100 ($20x10^3$)	101 ($50x10^3$)	110 ($100x10^3$)	111 ($200x10^3$)
0000 (÷1)	$1x10^3$	$2x10^3$	$5x10^3$	$10x10^3$	$20x10^3$	$50x10^3$	$100x10^3$	$200x10^3$
0001 (÷2)	$2x10^3$	$4x10^3$	$10x10^3$	$20x10^3$	$40x10^3$	$100x10^3$	$200x10^3$	$400x10^3$
0010 (÷3)	$3x10^3$	$6x10^3$	$15x10^3$	$30x10^3$	$60x10^3$	$150x10^3$	$300x10^3$	$600x10^3$
0011 (÷4)	$4x10^3$	$8x10^3$	$20x10^3$	$40x10^3$	$80x10^3$	$200x10^3$	$400x10^3$	$800x10^3$
0100 (÷5)	$5x10^3$	$10x10^3$	$25x10^3$	$50x10^3$	$100x10^3$	$250x10^3$	$500x10^3$	$1x10^6$
0101 (÷6)	$6x10^3$	$12x10^3$	$30x10^3$	$60x10^3$	$120x10^3$	$300x10^3$	$600x10^3$	$1.2x10^6$
0110 (÷7)	$7x10^3$	$14x10^3$	$35x10^3$	$70x10^3$	$140x10^3$	$350x10^3$	$700x10^3$	$1.4x10^6$
0111 (÷8)	$8x10^3$	$16x10^3$	$40x10^3$	$80x10^3$	$160x10^3$	$400x10^3$	$800x10^3$	$1.6x10^6$
1000 (÷9)	$9x10^3$	$18x10^3$	$45x10^3$	$90x10^3$	$180x10^3$	$450x10^3$	$900x10^3$	$1.8x10^6$
1001 (÷10)	$10 x10^3$	$20x10^3$	$50x10^3$	$100x10^3$	$200x10^3$	$500x10^3$	$1x10^6$	$2x10^6$
1010 (÷11)	$11 x10^3$	$22x10^3$	$55x10^3$	$110x10^3$	$220x10^3$	$550x10^3$	$1.1x10^6$	$2.2x10^6$
1011 (÷12)	$12x10^3$	$24x10^3$	$60x10^3$	$120x10^3$	$240x10^3$	$600x10^3$	$1.2x10^6$	$2.4x10^6$
1100 (÷13)	$13x10^3$	$26x10^3$	$65x10^3$	$130x10^3$	$260x10^3$	$650x10^3$	$1.3x10^6$	$2.6x10^6$
1101 (÷14)	$14x10^3$	$28x10^3$	$70x10^3$	$140x10^3$	$280x10^3$	$700x10^3$	$1.4x10^6$	$2.8x10^6$
1110 (÷15)	$15x10^3$	$30x10^3$	$75x10^3$	$150x10^3$	$300x10^3$	$750x10^3$	$1.5x10^6$	$3x10^6$
1111 (÷16)	$16x10^3$	$32x10^3$	$80x10^3$	$160x10^3$	$320x10^3$	$800x10^3$	$1.6x10^6$	$3.2x10^6$

6.2.1.6.2.10 S12ZCPMU COP control register (CPMUCOP)

This register controls the COP (computer operating properly) watchdog.

The clock source for the COP is either ACLK, IRCCLK or OSCCLK depending on the setting of the COPOSCSEL0 and COPOSCSEL1 bit (see also **Table 76**).
In Stop Mode with PSTP=1 (Pseudo Stop mode), COPOSCSEL0=1 and COPOSCEL1=0 and PCE=1 the COP continues to run, else the COP counter halts in Stop mode with COPOSCSEL1 =0.
In Full Stop mode and Pseudo Stop mode with COPOSCSEL1=1 the COP continues to run.

Table 84. S12ZCPMU COP control register (CPMUCOP)

Module Base + 0x000C

	7	6	5	4	3	2	1	0
R	WCOP	RSBCK	0	0	0	CR2	CR1	CR0
W			WRTMASK					
Reset	F	0	0	0	0	F	F	F

After de-assert of System Reset the values are automatically loaded from the Flash memory. See Device specification for details.

= Unimplemented or Reserved

Notes:

89. Read: Anytime
Write:
1. RSBCK: Anytime in Special mode; write to “1” but not to “0” in Normal mode
2. WCOP, CR2, CR1, CR0:
   - Anytime in Special mode, when WRTMASK is 0, otherwise it has no effect
   - Write once in Normal mode, when WRTMASK is 0, otherwise it has no effect.
     - Writing CR[2:0] to “000” has no effect, but counts for the “write once” condition.
     - Writing WCOP to “0” has no effect, but counts for the “write once” condition.

When a non-zero value is loaded from Flash to CR[2:0] the COP timeout period is started.

A change of the COPOSCSEL0 or COPOSCSEL1 bit (writing a different value) or loosing UPOSC status while COPOSCSEL1 is clear and COPOSCSEL0 is set, re-starts the COP timeout period.

In Normal Mode the COP timeout period is restarted if either of these conditions is true:

1. Writing a non-zero value to CR[2:0] (anytime in special mode, once in normal mode) with WRTMASK = 0.
2. Writing WCOP bit (anytime in Special mode, once in Normal mode) with WRTMASK = 0.
3. Changing RSBCK bit from “0” to “1”.

In Special mode, any write access to CPMUCOP register restarts the COP timeout period.

Table 85. CPMUCOP field descriptions

Field	Description
7 WCOP	**Window COP Mode Bit** — When set, a write to the CPMUARMCOP register must occur in the last 25% of the selected period. A write during the first 75% of the selected period generates a COP reset. As long as all writes occur during this window, $55 can be written as often as desired. Once $AA is written after the $55, the timeout logic restarts and the user must wait until the next window before writing to CPMUARMCOP. Table 86 shows the duration of this window for the seven available COP rates. 0 Normal COP operation 1 Window COP operation
6 RSBCK	COP and RTI Stop in Active BDM mode Bit 0 Allows the COP and RTI to keep running in Active BDM mode. 1 Stops the COP and RTI counters whenever the part is in Active BDM mode.

Table 85. CPMUCOP field descriptions (continued)

Field	Description
5 WRTMASK	**Write Mask for WCOP and CR[2:0] Bit** — This write-only bit serves as a mask for the WCOP and CR[2:0] bits while writing the CPMUCOP register. It is intended for BDM writing the RSBCK without changing the content of WCOP and CR[2:0]. 0 Write of WCOP and CR[2:0] has an effect with this write of CPMUCOP 1 Write of WCOP and CR[2:0] has no effect with this write of CPMUCOP. (Does not count for “write once”.)
2–0 CR[2:0]	**COP Watchdog Timer Rate Select** — These bits select the COP timeout rate (see Table 86 and Table 87). Writing a nonzero value to CR[2:0] enables the COP counter and starts the timeout period. A COP counter timeout causes a System Reset. This can be avoided by periodically (before timeout) initializing the COP counter via the CPMUARMCOP register. While all of the following four conditions are true the CR[2:0], WCOP bits are ignored and the COP operates at highest timeout period (2^{24} cycles) in normal COP mode (Window COP mode disabled): 1) COP is enabled (CR[2:0] is not 000) 2) BDM mode active 3) RSBCK = 0 4) Operation in Special Mode

Table 86. COP watchdog rates if COPOSCSEL1=0. (default out of reset)

CR2	CR1	CR0	COPCLK Cycles to timeout (COPCLK is either IRCCLK or OSCCLK depending on the COPOSCSEL0 bit)
0	0	0	COP disabled
0	0	1	2^{14}
0	1	0	2^{16}
0	1	1	2^{18}
1	0	0	2^{20}
1	0	1	2^{22}
1	1	0	2^{23}
1	1	1	2^{24}

Table 87. COP watchdog rates if COPOSCSEL1=1.

CR2	CR1	CR0	COPCLK Cycles to timeout (COPCLK is ACLK divided by 2)
0	0	0	COP disabled
0	0	1	2^{7}
0	1	0	2^{9}
0	1	1	2^{11}
1	0	0	2^{13}
1	0	1	2^{15}
1	1	0	2^{16}
1	1	1	2^{17}

6.2.1.6.2.11 Reserved register CPMUTEST0

Note

This reserved register is designed for factory test purposes only and is not intended for general user access. Writing to this register when in Special mode can alter the S12ZCPMU’s functionality.

Table 88. Reserved register (CPMUTEST0)

Module Base + 0x000D

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
90.Read: Anytime
Write: Only in Special mode

6.2.1.6.2.12 Reserved register CPMUTEST1

Note

This reserved register is designed for factory test purposes only and is not intended for general user access. Writing to this register when in Special Mode can alter the S12ZCPMU's functionality.

Table 89. Reserved register (CPMUTEST1)

Module Base + 0x000E

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
91.Read: Anytime
Write: Only in Special mode

6.2.1.6.2.13 S12ZCPMU COP timer arm/reset register (CPMUARMCOP)

This register is used to restart the COP timeout period.

Table 90. S12ZCPMU CPMUARMCOP register

Module Base + 0x000F

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W	ARMCOP-Bit 7	ARMCOP-Bit 6	ARMCOP-Bit 5	ARMCOP-Bit 4	ARMCOP-Bit 3	ARMCOP-Bit 2	ARMCOP-Bit 1	ARMCOP-Bit 0
Reset	0	0	0	0	0	0	0	0

Notes:
92.Read: Always reads $00
Write: Anytime

When the COP is disabled (CR[2:0] = "000") writing to this register has no effect.

When the COP is enabled by setting CR[2:0] nonzero, the following applies:

Writing any value other than $55 or $AA causes a COP reset. To restart the COP timeout period write $55 followed by a write of $AA. These writes do not need to occur back-to-back, but the sequence ($55, $AA) must be completed prior to COP end of timeout period to avoid a COP reset. Sequences of $55 writes are allowed. When the WCOP bit is set, $55 and $AA writes must be done in the last 25% of the selected timeout period; writing any value in the first 75% of the selected period will cause a COP reset.

6.2.1.6.2.14 Low voltage control register (CPMULVCTL)

The CPMULVCTL register allows the configuration of the low-voltage detect features.

Table 91. Low-voltage control register (CPMULVCTL)

Module Base + 0x0011

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	LVDS	LVIE	LVIF
W								
Reset	0	0	0	0	0	U	0	U

The Reset state of LVDS and LVIF depends on the external supplied VDDRX level

= Unimplemented or Reserved

Notes:
93.Read: Anytime
Write: LVIE and LVIF are write anytime, LVDS is read only

Table 92. CPMULVCTL field descriptions

Field	Description
2 LVDS	**Low-voltage Detect Status Bit** — This read-only status bit reflects the voltage level on VDDRX. Writes have no effect. 0 Input voltage VDDA is above level V_{LVID} or RPM. 1 Input voltage VDDA is below level V_{LVIA} and FPM.
1 LVIE	**Low-voltage Interrupt Enable Bit** 0 Interrupt request is disabled. 1 Interrupt will be requested whenever LVIF is set.
0 LVIF	**Low-voltage Interrupt Flag** — LVIF is set to 1 when LVDS status bit changes. This flag can only be cleared by writing a 1. Writing a 0 has no effect. If enabled (LVIE = 1), LVIF causes an interrupt request. 0 No change in LVDS bit. 1 LVDS bit has changed.

6.2.1.6.2.15 Autonomous periodical interrupt control register (CPMUAPICTL)

The CPMUAPICTL register allows the configuration of the autonomous periodical interrupt features.

Table 93. Autonomous periodical interrupt control register (CPMUAPICTL)

Module Base + 0x0012

	7	6	5	4	3	2	1	0
R	APICLK	0	0	APIES	APIEA	APIFE	APIE	APIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
94.Read: Anytime
Write: Anytime

Table 94. CPMUAPICTL field descriptions

Field	Description
7 APICLK	**Autonomous Periodical Interrupt Clock Select Bit** — Selects the clock source for the API. Writable only if APIFE = 0. APICLK cannot be changed if APIFE is set by the same write operation. 0 Autonomous Clock (ACLK) used as source. 1 Bus clock used as source.
4 APIES	**Autonomous Periodical Interrupt External Select Bit** — Selects the waveform at the external pin API_EXTCLK as shown in Figure 21. See device level specification for connectivity of API_EXTCLK pin. 0 If APIEA and APIFE are set, at the external pin API_EXTCLK periodic high pulses are visible at the end of every selected period with the size of half of the minimum period (APIR=0x0000 in Table 101). 1 If APIEA and APIFE are set, at the external pin API_EXTCLK a clock is visible with two times the selected API Period.
3 APIEA	**Autonomous Periodical Interrupt External Access Enable Bit** — If set, the waveform selected by bit APIES can be accessed externally. See device level specification for connectivity. 0 Waveform selected by APIES can not be accessed externally. 1 Waveform selected by APIES can be accessed externally, if APIFE is set.
2 APIFE	**Autonomous Periodical Interrupt Feature Enable Bit** — Enables the API feature and starts the API timer when set. 0 Autonomous periodical interrupt is disabled. 1 Autonomous periodical interrupt is enabled and timer starts running.
1 APIE	Autonomous Periodical Interrupt Enable Bit 0 API interrupt request is disabled. 1 API interrupt will be requested whenever APIF is set.
0 APIF	**Autonomous Periodical Interrupt Flag** — APIF is set to 1 when the in the API configured time has elapsed. This flag can only be cleared by writing a 1.Writing a 0 has no effect. If enabled (APIE = 1), APIF causes an interrupt request. 0 API timeout has not yet occurred. 1 API timeout has occurred.

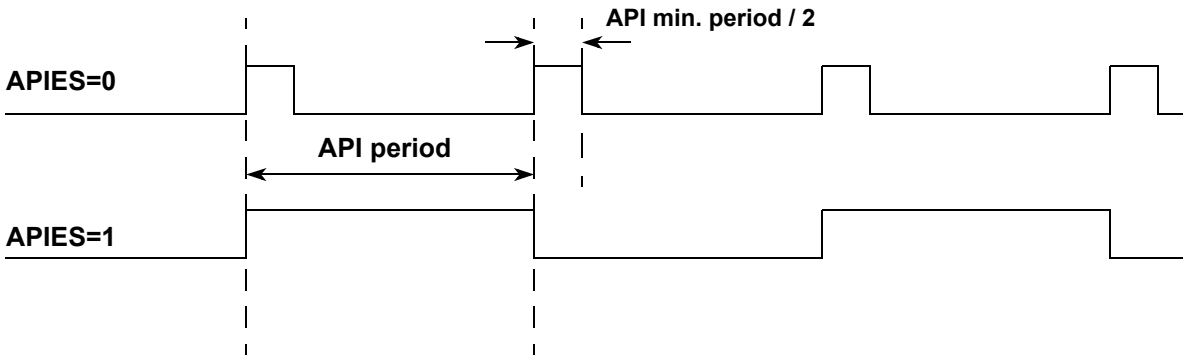


Figure 21. Waveform selected on API_EXTCLK pin (APIEA=1, APIFE=1)

6.2.1.6.2.16 Autonomous clock trimming register (CPMUACLKTR)

The CPMUACLKTR register configures the trimming of the Autonomous Clock (ACLK - trimmable internal RC-Oscillator) which can be selected as clock source for some CPMU features.

Table 95. Autonomous clock trimming register (CPMUACLKTR)

Module Base + 0x0013

	7	6	5	4	3	2	1	0
R	ACLKTR5	ACLKTR4	ACLKTR3	ACLKTR2	ACLKTR1	ACLKTR0	0	0
W								
Reset	F	F	F	F	F	F	0	0

After de-assert of System Reset a value is automatically loaded from the Flash memory.

Notes:
95.Read: Anytime
Write: Anytime

Table 96. CPMUACLKTR field descriptions

Field	Description
7–2 ACLKTR[5:0]	**Autonomous Clock Period Trimming Bits** — See Table 97 for trimming effects. The ACLKTR[5:0] value represents a signed number influencing the ACLK period time.

Table 97. Trimming effect of ACLKTR

Bit	Trimming Effect
ACLKTR[5]	Increases period
ACLKTR[4]	Decreases period less than ACLKTR[5] increased it
ACLKTR[3]	Decreases period less than ACLKTR[4]
ACLKTR[2]	Decreases period less than ACLKTR[3]
ACLKTR[1]	Decreases period less than ACLKTR[2]
ACLKTR[0]	Decreases period less than ACLKTR[1]

6.2.1.6.2.17 Autonomous periodical interrupt rate high and low register (CPMUAPIRH / CPMUAPIRL)

The CPMUAPIRH and CPMUAPIRL registers allow the configuration of the autonomous periodical interrupt rate.

Table 98. Autonomous periodical interrupt rate high register (CPMUAPIRH)

Module Base + 0x0014

	7	6	5	4	3	2	1	0
R W	APIR15	APIR14	APIR13	APIR12	APIR11	APIR10	APIR9	APIR8
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Table 99. Autonomous periodical interrupt rate low register (CPMUAPIRL)

Module Base + 0x0015

	7	6	5	4	3	2	1	0
R W	APIR7	APIR6	APIR5	APIR4	APIR3	APIR2	APIR1	APIR0
Reset	0	0	0	0	0	0	0	0

Notes:
96.Read: Anytime
Write: Anytime if APIFE=0, Else writes have no effect

Table 100. CPMUAPIRH / CPMUAPIRL field descriptions

Field	Description
15-0 APIR[15:0]	**Autonomous Periodical Interrupt Rate Bits** — These bits define the timeout period of the API. See Table 101 for details of the effect of the autonomous periodical interrupt rate bits.

The period can be calculated as follows depending on logical value of the APICLK bit:

APICLK = 0: Period = 2*(APIR[15:0] + 1) * (ACLK Clock Period * 2)
APICLK = 1: Period = 2*(APIR[15:0] + 1) * Bus Clock Period

Note

For APICLK bit clear the first timeout period of the API will show a latency time between two to three f_{ACLK} cycles due to synchronous clock gate release when the API feature gets enabled (APIFE bit set).

Table 101. Selectable autonomous periodical interrupt periods

APICLK	APIR[15:0]	Selected Period
0	0000	0.2 ms (97)
0	0001	0.4 ms (97)
0	0002	0.6 ms (97)
0	0003	0.8 ms (97)
0	0004	1.0 ms (97)
0	0005	1.2 ms (97)
0	.....	.....
0	FFFD	13106.8 ms (97)
0	FFFE	13107.0 ms (97)
0	FFFF	13107.2 ms (97)
1	0000	2 * Bus Clock period
1	0001	4 * Bus Clock period
1	0002	6 * Bus Clock period
1	0003	8 * Bus Clock period
1	0004	10 * Bus Clock period
1	0005	12 * Bus Clock period
1	.....	.....
1	FFFD	131068 * Bus Clock period
1	FFFE	131070 * Bus Clock period
1	FFFF	131072 * Bus Clock period

Notes:
97.When f_{ACLK} is trimmed to 20 kHz

6.2.1.6.2.18 Reserved register CPMUTEST3

Note

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special Mode can alter the S12ZCPMU's functionality.

Table 102. Reserved register (CPMUTEST3)

Module Base + 0x0016

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
98.Read: Anytime
Write: Only in Special mode

6.2.1.6.2.19 S12ZCPMU IRC1M trim registers (CPMUIRCTRIMH / CPMUIRCTRIML)

Table 103. S12ZCPMU IRC1M trim high register (CPMUIRCTRIMH)

Module Base + 0x0018

	15	14	13	12	11	10	9	8
R	TCTRIM[4:0]					0	IRCTRIM[9:8]	
W								
Reset	F	F	F	F	F	0	F	F

After de-assert of System Reset, a factory programmed trim value is automatically loaded from the Flash memory to provide trimmed Internal Reference Frequency f_{IRC1M_TRIM}.

Table 104. S12ZCPMU IRC1M trim low register (CPMUIRCTRIML)

Module Base + 0x0019

	7	6	5	4	3	2	1	0
R	IRCTRIM[7:0]							
W								
Reset	F	F	F	F	F	F	F	F

After de-assert of System Reset, a factory programmed trim value is automatically loaded from the Flash memory to provide trimmed Internal Reference Frequency f_{IRC1M_TRIM}.

Notes:
99.Read: Anytime
Write: if PROT = 0 (CPMUPROT register). Else write has no effect

Note

Writes to these registers while PLLSEL = 1 clears the LOCK and UPOSC status bits.

Table 105. CPMUIRCTRIMH/L field descriptions

Field	Description
15-11 TCTRIM[4:0]	**IRC1M temperature coefficient Trim Bits** Trim bits for the temperature coefficient (TC) of the IRC1M frequency. Table shows the influence of the bits TCTRIM[4:0] on the relationship between frequency and temperature. Figure 23 shows an approximate TC variation, relative to the nominal TC of the IRC1M (i.e. for TCTRIM[4:0] = 0x00000 or 0x10000).
9-0 IRCTRIM[9:0]	**IRC1M Frequency Trim Bits** — Trim bits for internal reference clock After System Reset, the factory programmed trim value is automatically loaded into these registers, resulting in an internal reference frequency f_{IRC1M_TRIM}.See device electrical characteristics for the value of f_{IRC1M_TRIM}. The frequency trimming consists of two different trimming methods: A rough trimming controlled by bits IRCTRIM[9:6] can be done with frequency leaps of about 6% in average. A fine trimming controlled by bits IRCTRIM[5:0] can be done with frequency leaps of about 0.3% (this trimming determines the precision of the frequency setting of 0.15%, i.e. 0.3% is the distance between two trimming values). Figure 22 shows the relationship between the trim bits and the resulting IRC1M frequency.

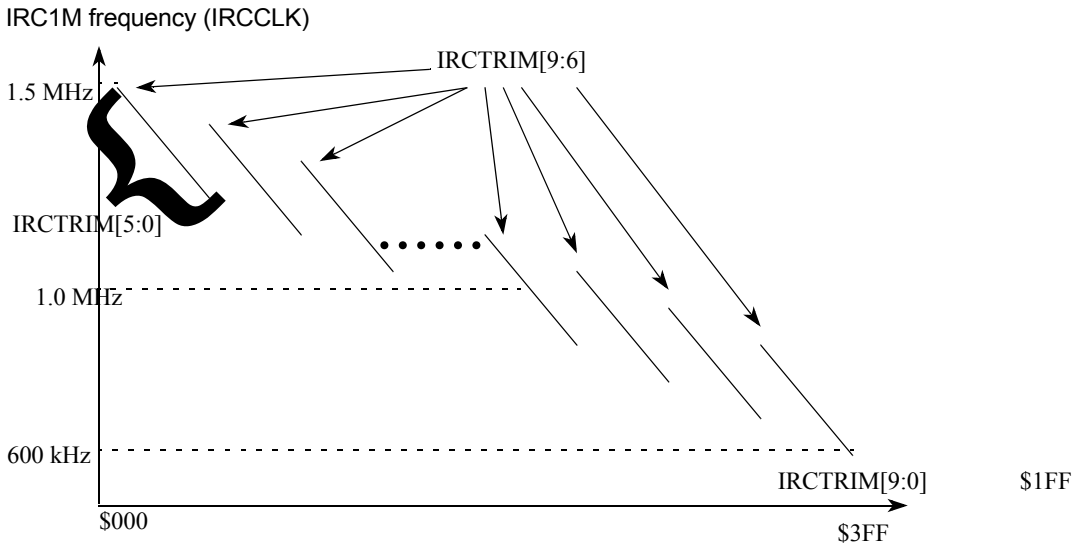


Figure 22. IRC1M frequency trimming diagram

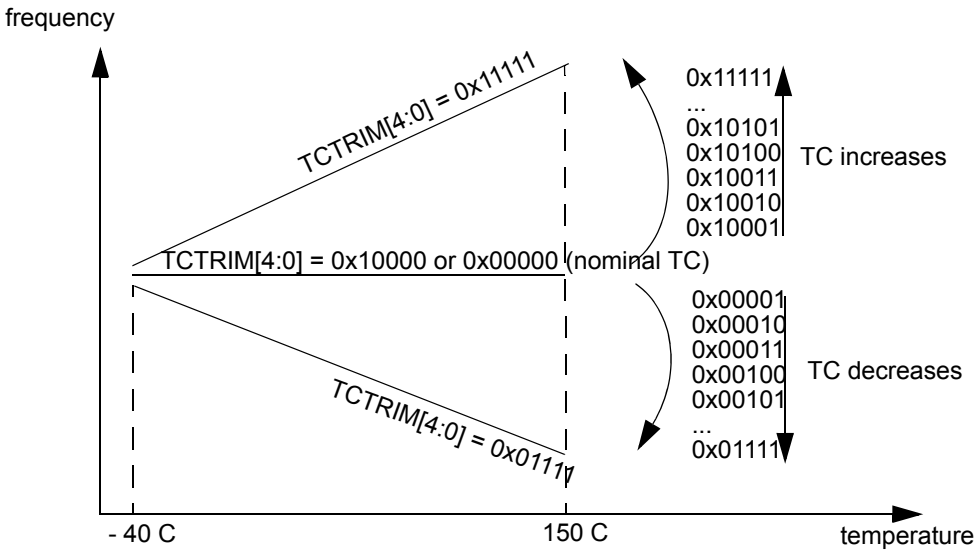


Figure 23. Influence of TCTRIM[4:0] on the temperature coefficient

Note

The frequency is not necessarily linear with the temperature (in most cases it will not be). Figure 23 is meant only to give the direction (positive or negative) of the variation of the TC, relative to the nominal TC.

Setting TCTRIM[4:0] at 0x00000 or 0x10000 does not mean that the temperature coefficient will be zero. These two combinations basically switch off the TC compensation module, which results in the nominal TC of the IRC1M.

Table 106. TC trimming of the frequency of the IRC1M at ambient temperature

TCTRIM[4:0]	IRC1M Indicative relative TC variation	IRC1M indicative frequency drift for relative TC variation
00000	0 (nominal TC of the IRC)	0%
00001	-0.27%	-0.5%
00010	-0.54%	-0.9%
00011	-0.81%	-1.3%

Table 106. TC trimming of the frequency of the IRC1M at ambient temperature

TCTRIM[4:0]	IRC1M Indicative relative TC variation	IRC1M indicative frequency drift for relative TC variation
00100	-1.08%	-1.7%
00101	-1.35%	-2.0%
00110	-1.63%	-2.2%
00111	-1.9%	-2.5%
01000	-2.20%	-3.0%
01001	-2.47%	-3.4%
01010	-2.77%	-3.9%
01011	-3.04	-4.3%
01100	-3.33%	-4.7%
01101	-3.6%	-5.1%
01110	-3.91%	-5.6%
01111	-4.18%	-5.9%
10000	0 (nominal TC of the IRC)	0%
10001	+0.27%	+0.5%
10010	+0.54%	+0.9%
10011	+0.81%	+1.3%
10100	+1.07%	+1.7%
10101	+1.34%	+2.0%
10110	+1.59%	+2.2%
10111	+1.86%	+2.5%
11000	+2.11%	+3.0%
11001	+2.38%	+3.4%
11010	+2.62%	+3.9%
11011	+2.89%	+4.3%
11100	+3.12%	+4.7%
11101	+3.39%	+5.1%
11110	+3.62%	+5.6%
11111	+3.89%	+5.9%

Note

Since the IRC1M frequency is not a linear function of the temperature, but more like a parabola, the Table 106 relative variation is only an indication and should be considered with care.

Be aware that the output frequency varies with the TC trimming. A frequency trimming correction is therefore necessary. The values provided in **Table** are typical values at ambient temperature which can vary from device to device.

6.2.1.6.2.20 S12ZCPMU oscillator register (CPMUOSC)

This registers configures the external oscillator (XOSCLCP).

Table 107. S12ZCPMU oscillator register (CPMUOSC)

Module Base + 0x001A

	7	6	5	4	3	2	1	0
R	OSCE	0	Reserved (100)	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
100.Do not alter these bits from their reset value. These are for Manufacturer use only and can change the Oscillator and the PLL behavior.
101.Read: Anytime
Write: if PROT = 0 (CPMUPROT register) and PLLSEL=1 (CPMUCLKS register). Else write has no effect

Note

Write to this register clears the LOCK and UPOSC status bits.

Table 108. CPMUOSC field descriptions

Field	Description
7 OSCE	**Oscillator Enable Bit** — This bit enables the external oscillator (XOSCLCP). The UPOSC status bit in the CPMIUFLG register indicates when the oscillation is stable and when OSCCLK can be selected as Bus Clock or source of the COP or RTI. If the oscillator clock monitor reset is enabled (OMRE = 1 in CPMUOSC2 register), then a loss of oscillation will lead to an oscillator clock monitor reset. 0 External oscillator is disabled. REFCLK for PLL is IRCCLK. 1 External oscillator is enabled. REFCLK for PLL is the external oscillator clock divided by REFDIV. If OSCE bit has been set (write “1”) the EXTAL and XTAL pins are exclusively reserved for the oscillator and they can not be used anymore as general purpose I/O until the next system reset. When starting up the external oscillator (either by programming OSCE bit to 1 or on exit from Full Stop mode with OSCE bit already 1) the software must wait for a minimum time equivalent to the start-up time of the external oscillator t_{UPOSC} before entering Pseudo Stop mode.
5 Reserved	Do not alter this bit from its reset value.It is for Manufacturer use only and can change the Oscillator behavior.

6.2.1.6.2.21 S12ZCPMU protection register (CPMUPROT)

This register protects the clock configuration registers from accidental overwrite:

CPMUSYNR, CPMUREFDIV, CPMUCLKS, CPMUPLL, CPMUIRCTRIMH/L, CPMUOSC, and CPMUOSC2

Table 109. S12ZCPMU protection register (CPMUPROT)

Module Base + 0x001B

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	PROT
W								
Reset	0	0	0	0	0	0	0	0

Notes:
102.Read: Anytime
Write: Anytime

Table 110. CPMUPROT field description

Field	Description
PROT	**Clock Configuration Registers Protection Bit** - This bit protects the clock configuration registers from accidental overwrite (see list of protected registers in Table 109): Writing 0x26 to the CPMUPROT register clears the PROT bit, other write accesses set the PROT bit. 0 Protection of clock configuration registers is disabled. 1 Protection of clock configuration registers is enabled. (see list of protected registers in Table 109).

6.2.1.6.2.22 Reserved register CPMUTEST2

Note

This reserved register is designed for factory test purposes only, and is not intended for general user access. Writing to this register when in Special mode can alter the S12ZCPMU's functionality.

Table 111. Reserved register CPMUTEST2

Module Base + 0x001C

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
103.Read: Anytime
Write: Only in Special mode

6.2.1.6.2.23 S12ZCPMU oscillator register 2 (CPMUOSC2)

This registers configures the external oscillator (XOSCLCP).

Table 112. S12ZCPMU oscillator register 2 (CPMUOSC2)

Module Base + 0x001E

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	OMRE	OSCMOD
W								
Reset	0	0	0	0	0	0	0	0

Notes:
104.Read: Anytime
Write: Anytime if PROT = 0 (CPMUPROT register) and PLLSEL = 1 (CPMUCLKS register). Else write has no effect.

Table 113. CPMUOSC field descriptions

Field	Description
0 OSCMOD	**This bit selects the mode of the external oscillator (XOSCLCP)** If OSCE bit in CPMUOSC register is 1, then the OSCMOD bit can not be changed (writes will have no effect). 0 External oscillator configured for loop controlled mode (reduced amplitude on EXTAL and XTAL)) 1 External oscillator configured for full swing mode (full swing amplitude on EXTAL and XTAL)
1 OMRE	**This bit enables the oscillator clock monitor reset.** If OSCE bit in CPMUOSC register is 1, then the OMRE bit can not be changed (writes will have no effect). 0 Oscillator clock monitor reset is disabled 1 Oscillator clock monitor reset is enabled

6.2.1.7 Functional description

6.2.1.7.1 Phase locked loop with internal filter (PLL)

The PLL is used to generate a high speed PLLCLK based on a low frequency REFCLK.

The REFCLK is by default the IRCCLK which is trimmed to f_{IRC1M_TRIM} = 1.024 MHz.

If using the oscillator (OSCE = 1), REFCLK will be based on OSCCLK. For increased flexibility, OSCCLK can be divided in a range of 1 to 16, to generate the reference frequency REFCLK using the REFDIV[3:0] bits. Based on the SYNDIV[5:0] bits, the PLL generates the VCOCLK by multiplying the reference clock by a 2, 4, 6,... 126, 128. Based on the POSTDIV[4:0] bits the VCOCLK can be divided in a range of 1,2, 3, 4, 5, 6,... to 32 to generate the PLLCLK.

If oscillator is enabled (OSCE=1) $$f_{REF} = \frac{f_{OSC}}{(REFDIV+1)}$$

If oscillator is disabled (OSCE=0) $$f_{REF} = f_{IRC1M}$$

$$f_{VCO} = 2 \times f_{REF} \times (SYNDIV+1)$$

If PLL is locked (LOCK=1) $$f_{PLL} = \frac{f_{VCO}}{(POSTDIV+1)}$$

If PLL is not locked (LOCK=0) $$f_{PLL} = \frac{f_{VCO}}{4}$$

If PLL is selected (PLLSEL=1) $$f_{bus} = \frac{f_{PLL}}{2}$$

Note

Although it is possible to set the dividers to command a very high clock frequency, do not exceed the specified bus frequency limit for the MCU.

Several examples of PLL divider settings are shown in Table 114. The following rules help to achieve optimum stability and shortest lock time:

- Use lowest possible f_{VCO} / f_{REF} ratio (SYNDIV value).
- Use highest possible REFCLK frequency f_{REF}.

Table 114. Examples of PLL divider settings

f_{osc}	REFDIV[3:0]	f_{REF}	REFFRQ[1:0]	SYNDIV[5:0]	f_{VCO}	VCOFRQ[1:0]	POSTDIV[4:0]	f_{PLL}	f_{bus}
off	$00	1.0 MHz	00	$18	50 MHz	01	$03	12.5 MHz	6.25 MHz
off	$00	1.0 MHz	00	$18	50 MHz	01	$00	50 MHz	25 MHz
4MHz	$00	4.0 MHz	01	$05	48 MHz	00	$00	48 MHz	24 MHz

The phase detector inside the PLL compares the feedback clock (FBCLK = VCOCLK/(SYNDIV+1)) with the reference clock (REFCLK = (IRC1M or OSCCLK)/(REFDIV+1)). Correction pulses are generated based on the phase difference between the two signals. The loop filter alters the DC voltage on the internal filter capacitor, based on the width and direction of the correction pulse which leads to a higher or lower VCO frequency.

The user must select the range of the REFCLK frequency (REFFRQ[1:0] bits) and the range of the VCOCLK frequency (VCOFRQ[1:0] bits) to ensure that the correct PLL loop bandwidth is set.

The lock detector compares the frequencies of the FBCLK and the REFCLK. Therefore the speed of the lock detector is directly proportional to the reference clock frequency. The circuit determines the lock condition based on this comparison.

If PLL LOCK interrupt requests are enabled, the software can wait for an interrupt request and for instance check the LOCK bit. If interrupt requests are disabled, software can poll the LOCK bit continuously (during PLL start-up) or at periodic intervals. In either case, only when the LOCK bit is set, the VCOCLK will have stabilized to the programmed frequency.

- The LOCK bit is a read-only indicator of the locked state of the PLL.

- The LOCK bit is set when the VCO frequency is within the tolerance, Δ_{LOCK}, and is cleared when the VCO frequency is out of the tolerance, Δ_{UNL}.
- Interrupt requests can occur if enabled (LOCKIE = 1) when the lock condition changes, toggling the LOCK bit.

6.2.1.7.2 Startup from reset

An example for startup of the clock system from Reset is given in Figure 24.

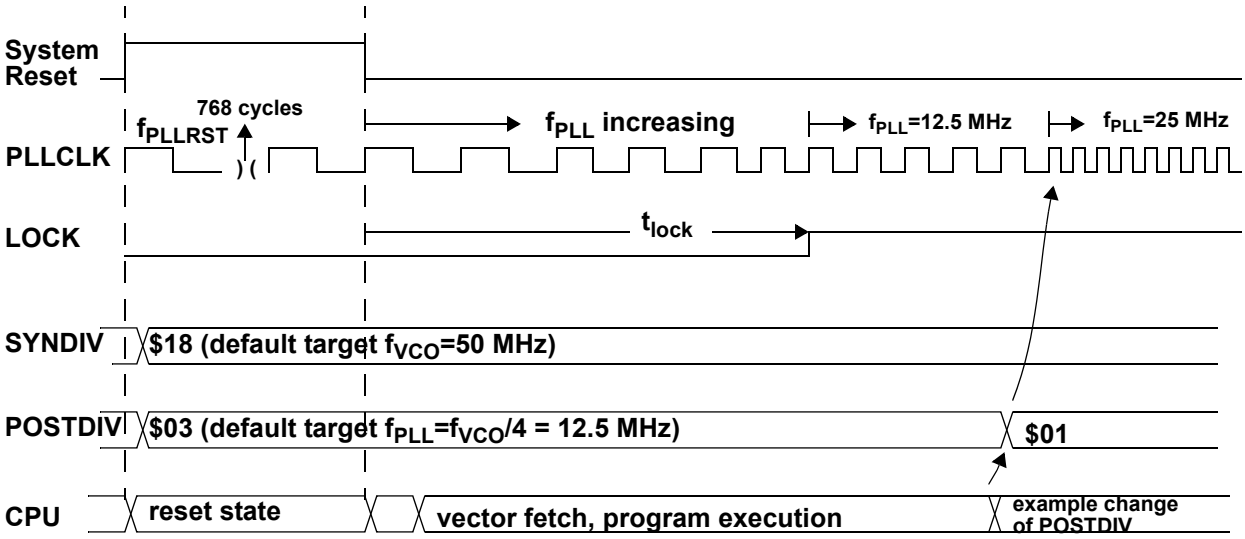


Figure 24. Startup of clock system after reset

6.2.1.7.3 Stop mode using PLLCLK as bus clock

An example of what happens going into Stop mode and exiting Stop mode after an interrupt is shown in Figure 25. Disable PLL Lock interrupt (LOCKIE=0) before going into Stop mode.

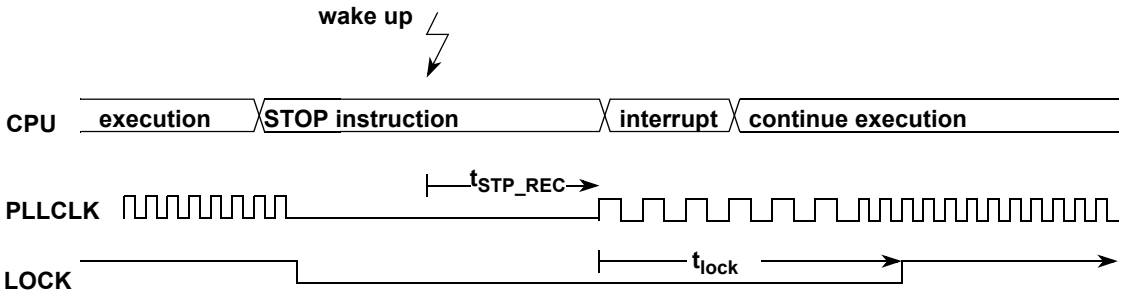


Figure 25. Stop mode using PLLCLK as bus clock

Depending on the COP configuration, there might be an additional significant latency time until COP is active again after exit from Stop mode due to clock domain crossing synchronization. This latency time of 2 ACLK cycles occurs if the COP clock source is ACLK and the CSAD bit is set and must be added to the device Stop mode recovery time t_{STP_REC}. After exit from Stop mode (Pseudo, Full) for this latency time of 2 ACLK cycles, no Stop mode request (STOP instruction) should be generated to make sure the COP counter can increment at each Stop mode exit.

6.2.1.7.4 Full stop mode using oscillator clock as bus clock

An example of what happens going into Full Stop mode and exiting Full Stop mode after an interrupt is shown in Figure 26.

Disable PLL Lock interrupt (LOCKIE=0) and oscillator status change interrupt (OSCIE=0) before going into Full Stop mode.

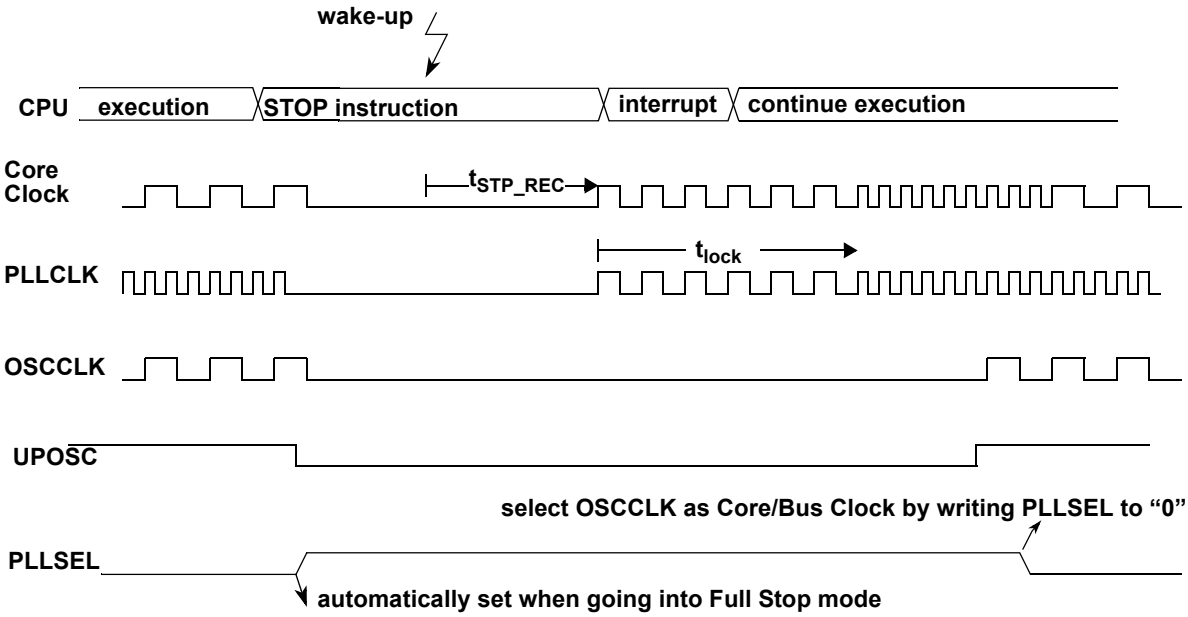


Figure 26. Full stop mode using oscillator clock as bus clock

Depending on the COP configuration, there might be a significant latency time until COP is active again after exit from Stop mode due to clock domain crossing synchronization. This latency time of 2 ACLK cycles occurs if COP clock source is ACLK and the CSAD bit is set and must be added to the device Stop mode recovery time t_{STP_REC}. After exit from Stop mode (Pseudo, Full) for this latency time of 2 ACLK cycles, no Stop mode request (STOP instruction) should be generated to make sure the COP counter can increment at each Stop mode exit.

6.2.1.7.5 External oscillator

6.2.1.7.5.1 Enabling the external oscillator

An example of how to use the oscillator as Bus Clock is shown in **Figure 27**.

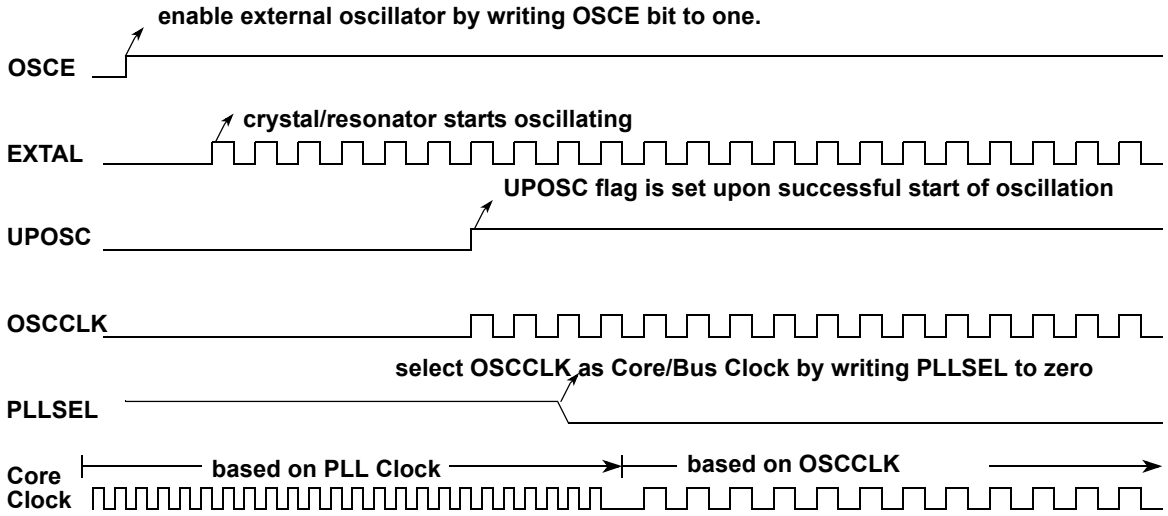


Figure 27. Enabling the external oscillator

6.2.1.7.6 System clock configurations

6.2.1.7.6.1 PLL engaged internal mode (PEI)

This mode is the default mode after system reset or Power-On reset.

The bus clock is based on the PLLCLK, the reference clock for the PLL is internally generated (IRC1M). The PLL is configured to 50 MHz VCOCLK with POSTDIV set to 0x03. If locked (LOCK=1). This results in a PLLCLK of 12.5 MHz and a bus clock of 6.25 MHz. The PLL can be re-configured to other bus frequencies. The clock sources for COP and RTI can be based on the internal reference clock generator (IRC1M) or the RC-oscillator (ACLK).

6.2.1.7.6.2 PLL engaged external mode (PEE)

In this mode, the bus clock is based on the PLLCLK as well (like PEI). The reference clock for the PLL is based on the external oscillator.

The clock sources for COP and RTI can be based on the internal reference clock generator or on the external oscillator clock or the RC-Oscillator (ACLK).

This mode can be entered from default mode PEI by performing the following steps:

1. Configure the PLL for desired bus frequency.
2. Enable the external Oscillator (OSCE bit).
3. Wait for oscillator to start-up and the PLL being locked (LOCK = 1) and (UPOSC = 1).
4. Clear all flags in the CPMUIFLG register to be able to detect any future status bit change.
5. Optionally status interrupts can be enabled (CPMUINT register).

Loosing PLL lock status (LOCK=0) means loosing the oscillator status information as well (UPOSC=0).

The impact of loosing the oscillator status (UPOSC=0) in PEE mode is as follows:

- The PLLCLK is derived from the VCO clock (with its actual frequency) divided by four until the PLL locks again.

Application software needs to be prepared to deal with the impact of loosing the oscillator status at any time.

6.2.1.7.6.3 PLL bypassed external mode (PBE)

In this mode, the bus clock is based on the external oscillator clock. The reference clock for the PLL is based on the external oscillator.

The clock sources for COP and RTI can be based on the internal reference clock generator or on the external oscillator clock or the RC-Oscillator (ACLK).

This mode can be entered from default mode PEI by performing the following steps:

1. Make sure the PLL configuration is valid.
2. Enable the external oscillator (OSCE bit)
3. Wait for the oscillator to start-up and the PLL being locked (LOCK = 1) and (UPOSC = 1)
4. Clear all flags in the CPMUIFLG register to be able to detect any status bit change.
5. Optionally status interrupts can be enabled (CPMUINT register).
6. Select the oscillator clock as bus clock (PLLSEL=0)

Loosing PLL lock status (LOCK=0) means loosing the oscillator status information as well (UPOSC=0).

The impact of loosing the oscillator status (UPOSC=0) in PBE mode is as follows:

- PLLSEL is set automatically and the bus clock is switched back to the PLL clock.
- The PLLCLK is derived from the VCO clock (with its actual frequency) divided by four until the PLL locks again.

Application software needs to be prepared to deal with the impact of loosing the oscillator status at any time.

6.2.1.8 Resets

6.2.1.8.1 General

All reset sources are listed in Table 115. There is only one reset vector for all these reset sources. Refer to MCU specification for reset vector address.

Table 115. Reset summary

Reset Source	Local Enable
Power-On Reset (POR)	None
Low-voltage Reset (LVR)	None
External pin $\overline{\text{RESET}}$	None
PLL Clock Monitor Reset	None
Oscillator Clock Monitor Reset	OSCE Bit in CPMUOSC register and OMRE Bit in CPMUOSC2 register
COP Reset	CR[2:0] in CPMUCOP register

6.2.1.8.2 Description of reset operation

Detection of any reset of an internal circuit drives the $\overline{\text{RESET}}$ pin low for 512 PLLCLK cycles. After 512 PLLCLK cycles, the $\overline{\text{RESET}}$ pin is released. The internal reset of the MCU remains asserted while the reset generator completes the 768 PLLCLK cycles long reset sequence. In case the $\overline{\text{RESET}}$ pin is externally driven low for more than these 768 PLLCLK cycles (External Reset), the internal reset remains asserted longer.

Note

While System Reset is asserted the PLLCLK runs with the frequency f_{VCORST}.

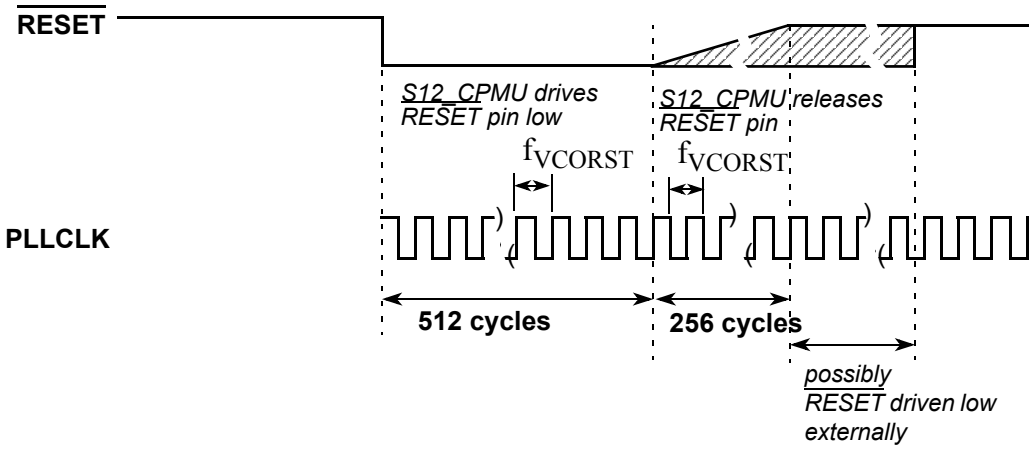


Figure 28. RESET timing

6.2.1.8.3 Oscillator clock monitor reset

If the external oscillator is enabled (OSCE=1) and the oscillator clock monitor reset is enabled (OMRE=1), then in case of loss of oscillation or the oscillator frequency drops below the failure assert frequency f_{CMFA} (see device electrical characteristics for values), the S12ZCPMU generates an Oscillator Clock Monitor Reset. In Full Stop mode the external oscillator and the oscillator clock monitor reset are both disabled.

In case of loss of PLL clock oscillation or the PLL clock frequency is below the failure assert frequency f_{PMFA} (see device electrical characteristics for values), the S12ZCPMU generates a PLL Clock Monitor Reset. In Full Stop mode the PLL and the PLL clock monitor are disabled.

6.2.1.8.4 Computer operating properly watchdog (COP) reset

The COP (free running watchdog timer) enables the user to check that a program is running and sequencing properly. When the COP is being used, software is responsible for keeping the COP from timing out. If the COP times out, it is an indication that the software is no longer being executed in the intended sequence; thus COP reset is generated.

The clock source for the COP is either ACLK, IRCCLK or OSCCLK depending on the setting of the COPOSCSEL0 and COPOSCSEL1 bit. Due to clock domain crossing synchronization there is a latency time to enter and exit Stop mode if the COP clock source is ACLK and this clock is stopped in Stop mode. This maximum total latency time is 4 ACLK cycles (2 ACLK cycles for Stop mode entry and exit each) which must be added to the Stop mode recovery time t_{STP_REC} from exit of current Stop mode to entry of next Stop mode. This latency time occurs no matter which Stop mode (Full, Pseudo) is currently exited or entered next.

After exit from Stop mode (Pseudo, Full) for this latency time of 2 ACLK cycles no Stop mode request (STOP instruction) should be generated to make sure the COP counter can increment at each Stop mode exit.

Table 116 gives an overview of the COP condition (run, static) in Stop mode depending on legal configuration and status bit settings:

Table 116. COP condition (run, static) in stop mode

COPOSCSEL1	CSAD	PSTP	PCE	COPOSCSEL0	OSCE	UPOSC	COP counter behavior in stop mode (clock source)
1	0	x	x	x	x	x	Run (ACLK)
1	1	x	x	x	x	x	Static (ACLK)
0	x	1	1	1	1	1	Run (OSCCLK)
0	x	1	1	0	0	x	Static (IRCCLK)
0	x	1	1	0	1	x	Static (IRCCLK)
0	x	1	0	0	x	x	Static (IRCCLK)
0	x	1	0	1	1	1	Static (OSCCLK)
0	x	0	1	1	1	1	Static (OSCCLK)
0	x	0	1	0	1	x	Static (IRCCLK)
0	x	0	1	0	0	0	Static (IRCCLK)
0	x	0	0	1	1	1	Satic (OSCCLK)
0	x	0	0	0	1	1	Static (IRCCLK)
0	x	0	0	0	1	0	Static (IRCCLK)
0	x	0	0	0	0	0	Static (IRCCLK)

Three control bits in the CPMUCOP register allow selection of seven COP timeout periods.

When COP is enabled, the program must write $55 and $AA (in this order) to the CPMUARMCOP register during the selected timeout period. Once this is done, the COP timeout period is restarted. If the program fails to do this and the COP times out, a COP reset is generated. Also, if any value other than $55 or $AA is written, a COP reset is generated.

Windowed COP operation is enabled by setting WCOP in the CPMUCOP register. In this mode, writes to the CPMUARMCOP register to clear the COP timer must occur in the last 25% of the selected timeout period. A premature write will immediately reset the part.

In MCU Normal mode, the COP timeout period (CR[2:0]) and COP window (WCOP) setting can be automatically pre-loaded at reset release from NVM memory (if values are defined in the NVM by the application). By default, the COP is off and no window COP feature is enabled after reset release via NVM memory. The COP control register CPMUCOP can be written once in an application in MCU Normal mode to update the COP timeout period (CR[2:0]) and COP window (WCOP) setting loaded from NVM memory at reset release. Any value for the new COP timeout period and COP window setting is allowed except COP off value, if the COP was enabled during pre-load via NVM memory.

The COP clock source select bits can not be pre-loaded via NVM memory at reset release. The IRC clock is the default COP clock source out of reset.

The COP clock source select bits (COPOSCSEL0/1) and ACLK clock control bit in Stop mode (CSAD) can be modified until the CPMUCOP register write once has taken place. Therefore these control bits should be modified before the final COP timeout period and window COP setting is written.

The CPMUCOP register access to modify the COP timeout period and window COP setting in MCU Normal mode after reset release must be done with the WRTMASK bit cleared, otherwise the update is ignored and this access does not count as the write once.

6.2.1.8.5 Power-on reset (POR)

The on-chip POR circuitry detects when the internal supply V_{DD} drops below an appropriate voltage level. The POR is deasserted, if the internal supply V_{DD} exceeds an appropriate voltage level (voltage levels not specified, because the internal supply can not be monitored externally. The POR circuitry is always active. It acts as LVR in Stop mode.

6.2.1.8.6 Low-voltage reset (LVR)

The on-chip LVR circuitry detects when one of the supply voltages VDD, VDDRX, and VDDF drops below an appropriate voltage level. If LVR is deasserted, the MCU is fully operational at the specified maximum speed. The LVR assert and deassert levels for the supply voltage VDDRX are V_{LVRXA} and V_{LVRXD}, and are specified in the device Reference Manual. The LVR circuitry is active in Run- and Wait mode.

6.2.1.9 Interrupts

The interrupt vectors requested by the S12ZCPMU are listed in Table 117. Refer to the MCU specification for related vector addresses and priorities.

Table 117. S12ZCPMU interrupt vectors

Interrupt Source	CCR Mask	Local Enable
RTI timeout interrupt	I bit	CPMUINT (RTIE)
PLL lock interrupt	I bit	CPMUINT (LOCKIE)
Oscillator status interrupt	I bit	CPMUINT (OSCIE)
Low-voltage interrupt	I bit	CPMULVCTL (LVIE)
Autonomous Periodical Interrupt	I bit	CPMUAPICTL (APIE)

6.2.1.9.1 Description of interrupt operation

6.2.1.9.1.1 Real time interrupt (RTI)

The clock source for the RTI is either IRCCLK or OSCCLK depending on the setting of the RTIOSCSEL bit. In Stop mode with PSTP = 1 (Pseudo Stop mode), RTIOSCSEL = 1 and PRE = 1 the RTI continues to run, else the RTI counter halts in Stop mode.

The RTI can be used to generate hardware interrupts at a fixed periodic rate. If enabled (by setting RTIE=1), this interrupt will occur at the rate selected by the CPMURTI register. At the end of the RTI timeout period the RTIF flag is set to one and a new RTI timeout period starts immediately.

A write to the CPMURTI register restarts the RTI timeout period.

6.2.1.9.1.2 PLL lock interrupt

The S12ZCPMU generates a PLL Lock interrupt when the lock condition (LOCK status bit) of the PLL changes, either from a locked state to an unlocked state or vice versa. Lock interrupts are locally disabled by setting the LOCKIE bit to zero. The PLL Lock interrupt flag (LOCKIF) is set to1 when the lock condition has changed, and is cleared to 0 by writing a 1 to the LOCKIF bit.

6.2.1.9.1.3 Oscillator status interrupt

When the OSCE bit is 0, then UPOSC stays 0. When OSCE = 1 the UPOSC bit is set after the LOCK bit is set.

Upon detection of a status change (UPOSC) the OSCIF flag is set. Going into Full Stop mode or disabling the oscillator can also cause a status change of UPOSC. Any change in PLL configuration or any other event which causes the PLL lock status to be cleared leads to a loss of the oscillator status information as well (UPOSC = 0).

Oscillator status change interrupts are locally enabled with the OSCIE bit.

Note

Loosing the oscillator status (UPOSC = 0) affects the clock configuration of the system(105). This needs to be dealt with in application software.

Notes:
105.For details refer to “System clock configurations”

6.2.1.9.1.4 Low-voltage interrupt (LVI)

In FPM the input voltage VDDRX is monitored. Whenever VDDRX drops below level $V_{LVIA,}$ the status bit LVDS is set to 1. When VDDRX rises above level V_{LVID} the status bit LVDS is cleared to 0. An interrupt, indicated by flag LVIF = 1, is triggered by any change of the status bit LVDS if interrupt enable bit LVIE = 1.

6.2.1.9.1.5 Autonomous periodical interrupt (API)

The API sub-block can generate periodical interrupts independent of the clock source of the MCU. To enable the timer, the bit APIFE needs to be set.

The API timer is either clocked by the autonomous clock (ACLK - trimmable internal RC oscillator) or the bus clock. Timer operation will freeze when MCU clock source is selected and bus clock is turned off. The clock source can be selected with bit APICLK. APICLK can only be written when APIFE is not set.

The APIR[15:0] bits determine the interrupt period. APIR[15:0] can only be written when APIFE is cleared. As soon as APIFE is set, the timer starts running for the period selected by APIR[15:0] bits. When the configured time has elapsed, the flag APIF is set. An interrupt, indicated by flag APIF = 1, is triggered if interrupt enable bit APIE = 1. The timer is re-started automatically again after it has set APIF.

The procedure to change APICLK or APIR[15:0] is first to clear APIFE, then write to APICLK or APIR[15:0], and afterwards set APIFE.

The API Trimming bits ACLKTR[5:0] must be set so the minimum period equals 0.2 ms if stable frequency is desired.

See Table 97 for the trimming effect of ACLKTR[5:0].

Note

The first period after enabling the counter by APIFE might be reduced by API start up delay t_{SDEL}.

It is possible to generate with the API a waveform at the external pin API_EXTCLK by setting APIFE and enabling the external access with setting APIEA.

6.2.1.10 Initialization/Application Information

6.2.1.10.1 General initialization information

Usually applications run in MCU Normal mode.

It is recommended to write the CPMUCOP register in any case from the application program initialization routine after reset no matter if the COP is used in the application or not, even if a configuration is loaded via the flash memory after reset. By doing a “controlled” write access in MCU Normal mode (with the right value for the application) the write once for the COP configuration bits (WCOP,CR[2:0]) takes place which protects these bits from further accidental change. In case of a program sequencing issue (code runaway) the COP configuration can not be accidentally modified anymore.

6.2.1.10.2 Application information for COP and API use

In many applications the COP is used to check that the program is running and sequencing properly. Often the COP is kept running during Stop mode and periodic wake-up events are needed to service the COP on time and maybe to check the system status.

For such an application it is recommended to use the ACLK as clock source for both COP and API. This guarantees lowest possible IDD current during Stop mode. Additionally it eases software implementation using the same clock source for both, COP and API.

The interrupt service routine (ISR) of the autonomous periodic interrupt API should contain the write instruction to the CPMUARMCOP register. The value (byte) written is derived from the “main routine” (alternating sequence of $55 and $AA) of the application software.

Using this method, then in the case of a runtime or program sequencing issue the application “main routine” is not executed properly anymore and the alternating values are not provided properly. Hence the COP is written at the correct time (due to independent API interrupt request) but the wrong value is written (alternating sequence of $55 and $AA is no longer maintained) which causes a COP reset.

If the COP is stopped during any Stop mode it is recommended to service the COP shortly before Stop mode is entered.

6.2.1.10.3 Application information for PLL and oscillator startup

The following C-code example shows a recommended way of setting up the system clock system using the PLL and Oscillator:

```
/* Procedure proposed by to setup PLL and Oscillator */
/* example for OSC = 4 MHz and Bus Clock = 25MHz, That is VCOCLK = 50MHz */

/* Initialize */
/* PLL Clock = 50 MHz, divide by one */
CPMUPOSTDIV = 0x00;

/* Generally: Whenever changing PLL reference clock (REFCLK) frequency to a higher value */
/* it is recommended to write CPMUSYNR = 0x00 in order to stay within specified */
/* maximum frequency of the MCU */
CPMUSYNR = 0x00;

/* configure PLL reference clock (REFCLK) for usage with Oscillator */
/* OSC=4MHz divide by 4 (3+1) = 1MHz, REFCLK range 1MHz to 2 MHz (REFFRQ[1:0] = 00) */
CPMUREFDV = 0x03;

/* enable external Oscillator, switch PLL reference clock (REFCLK) to OSC */
CPMUOSC = 0x80;

/* multiply REFCLK = 1MHz by 2*(24+1)*1MHz = 50MHz */
/* VCO range 48 to 80 MHz (VCOFRQ[1:0] = 01) */
CPMUSYNR = 0x58;

/* clear all flags, especially LOCKIF and OSCIF */
CPMUIFLG = 0xFF;

/* put your code to loop and wait for the LOCKIF and OSCIF or */
/* poll CPMUIFLG register until both UPOSC and LOCK status are “1” */
/* that is CPMIFLG == 0x1B */

/*..............continue to your main code execution here..............*/

/* in case later in your code you want to disable the Oscillator and use the */
/* 1MHz IRCCLK as PLL reference clock */

/* Generally: Whenever changing PLL reference clock (REFCLK) frequency to a higher value */
/* it is recommended to write CPMUSYNR = 0x00 in order to stay within specified */
/* maximum frequency of the MCU */
CPMUSYNR = 0x00;

/* disable OSC and switch PLL reference clock to IRC */
CPMUOSC = 0x00;

/* multiply REFCLK = 1MHz by 2*(24+1)*1MHz = 50MHz */
/* VCO range 48 to 80 MHz (VCOFRQ[1:0] = 01) */
CPMUSYNR = 0x58;
```

```
/* clear all flags, especially LOCKIF and OSCIF */
CPMUIFLG = 0xFF;

/* put your code to loop and wait for the LOCKIF or */
/* poll CPMUIFLG register until both LOCK status is “1” */
/* that is CPMIFLG == 0x18 */

/*..............continue to your main code execution here..............*/
```

6.2.2 Analog die - power, clock and resets - PCR

6.2.2.1 Introduction

The following chapter describes the MM9Z1_638’s system base functionality primary location on the analog die. The chapter is divided in the following sections:

1. Device operating modes
2. Power management
3. Wake-up sources
4. Device clock tree
5. System resets
6. PCR - memory map and registers

6.2.2.2 Device operating modes

The MM9Z1_638 features three main operation modes: Normal operation, Stop mode, and Sleep mode.

The full signal conditioning and measurements are permanently running in normal operation mode. The total current consumption of the MM9Z1_638 is reduced in the two Low-power modes.

The analog die of the MM9Z1_638 is still partially active and able to monitor the battery current, temperature, activities on the LIN interface and PTB4 terminal, during both Low-power modes.

6.2.2.2.1 Operating mode overview

- Normal mode
  - All device modules active
  - Microcontroller fully supplied
  - D2DCLK active analog die clock source
  - Window watchdog clocked by the low-power oscillator (LPCLK) to operate on independent clock
- Stop Mode
  - MCU in Low-power mode, MCU regulator supply (VDDX) with reduced current capability
  - D2D interface supply disabled (VDDH=OFF)
  - Unused analog blocks disabled
  - Watchdog is disabled
  - LIN wake-up, calibration request wake-up, cyclic wake-up, external wake-up, current threshold wake-up, and lifetime counter wake-up optional
  - Current Measurement / current averaging and temperature measurement optional
- Sleep Mode
  - MCU powered down (VDDH and VDDX = OFF)
  - Unused Analog Blocks disabled
  - Watchdogs = OFF
  - LIN wake-up, calibration request wake-up, cyclic wake-up, external wake-up, current threshold wake-up, and lifetime counter wake-up optional
  - Current measurement / current averaging and temperature measurement optional

- Intermediate mode
  - Every transition from Stop or Sleep into Normal mode will go through an Intermediate mode where the analog die clock is not yet switched to the D2D clock.
- Reset Mode
  - Every reset source within the analog die will bring the system into a Reset state
- Power On Reset Mode
  - For both low voltage thresholds are defined to indicate a loss of internal state.
- Cranking Mode
  - Special Mode implemented to guarantee the RAM content being valid though very low power conditions.

6.2.2.2.2 Operating mode transitions

The device operating modes are controlled by the microcontroller, as well as external and internal wake-up sources. Figure 29 shows the basic principal.

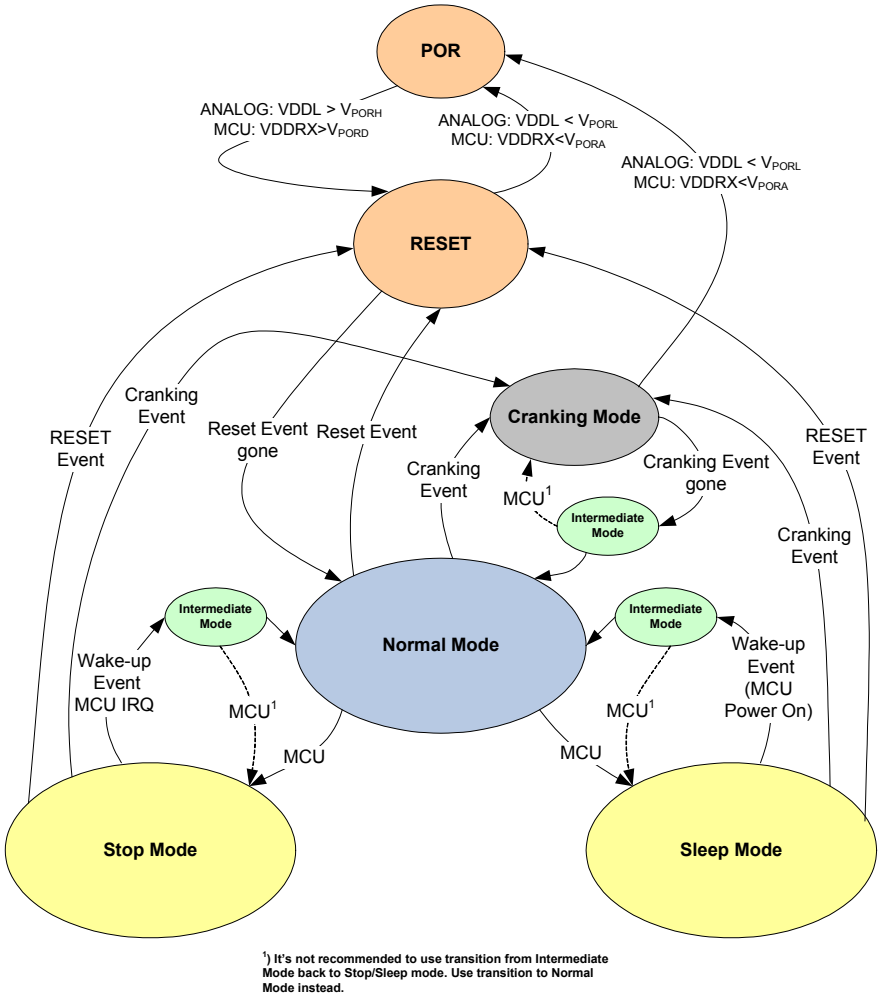


Figure 29. Modes of operation - transitions

6.2.2.2.3 Power on reset - POR

During system startup, or in any other case when MCU_VDD drops below VPORA (MCU), or LTO drops below VPORL (analog die), a Power On Reset (POR) condition is reached. The MCU (PORF) / analog die (LVRF) will indicate this state, setting the corresponding power on reset flag. The primary consequence of entering POR is that the RAM or analog register content can no longer be guaranteed.

6.2.2.2.4 RESET - mode

If any of the analog die reset conditions are present, the MM9Z1_638 analog die will enter Reset mode. During that mode, the analog die will issue the RESET_A pin to be pulled down to reset the microcontroller die. Entering Reset mode will reset the analog die registers to their default values.

The cause of the last reset is flagged in PCR status register (PCR_SR (hi)).

6.2.2.2.5 Normal mode

During Normal mode operation, all modules are operating and the microcontroller is fully supplied.

6.2.2.2.6 Cranking mode

A specific power down behavior has been implemented to allow the MCU memory (RAM) content to be guaranteed during very low supply voltage conditions. The difference between the device behavior, with or without the Cranking mode feature enabled, is described in Power up / power down behavior.

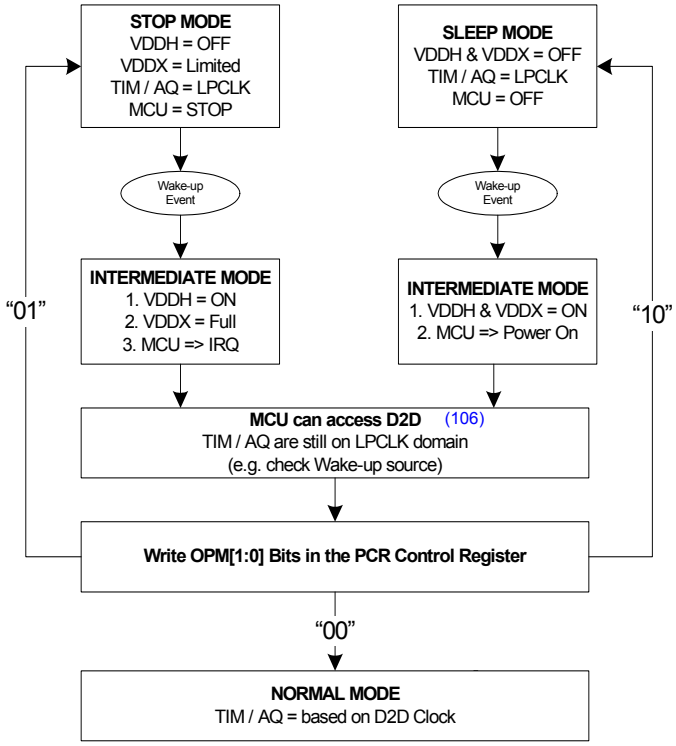
6.2.2.2.7 Intermediate mode

As the channel acquisition and the timer modules are switched to the LPCLK, while the MM9Z1_638 is operating in one of the two Low-power modes, the Intermediate mode has been implemented, to be able to go back to Low-power mode without the transition into the D2D Clock domain.

Note

The flag indicating the last wake-up source must be cleared before re-entering Low-power mode

Once awakened, the MCU instructs the analog die to transit to Normal mode by writing “00” to the OPM bits in the PCR Control Register. See Figure 30 for details.



Notes:

106.As the Life Time Counter has to be configured into Normal mode (no access to LTC_CNT[1..0] is possible during intermediate mode). If not reconfigured it will continue to increment starting from 0.

Figure 30. Low-power mode to normal mode transition through the intermediate mode

6.2.2.2.8 Low-power modes

In Low-power mode, the MM9Z1_638 is still active to monitor the battery current (triggered current measurement for current threshold detection and current accumulator function), and activities on the LIN interface and wake-up inputs. A cyclic wake-up using timer module is implemented for timed wake-up. Temperature measurements are optional to detect an out of calibration condition.

The Life Time counter is also incremented during Low-power mode, to issue a wake-up on overflow. See Life time counter (LTC) for additional details.

The average current consumption is reduced, and based on the actual Low-power mode, the active modules, and the wake-up timing.

Note

To avoid any lock condition, no analog die interrupt should be enabled or pending when entering LPM. To accomplish that condition, the analog die interrupts should be masked and served before writing the PCR_CTL register.

The MCU interrupts should be enabled right before the STOP command, to avoid any interrupt to be handled in between.

A wake-up from any of the Low-power modes will reset the window watchdog equal to a standard reset.

6.2.2.2.8.1 Sleep mode

Writing the PCR Control Register (PCR_CTL) with OPM=10, the MM9Z1_638 will enter Sleep mode with the configured wake-up sources (see Wake-up sources).

Note

The power supply to the MCU will be turned off during Sleep mode. To safely approach this condition, the MCU should be put into a safe state (e.g STOP).

During Sleep mode, the only active voltage regulator is LTO, supplying the low power oscillator (LPOSC), and the permanently supplied digital blocks.

When an enabled wake-up condition occurs, the shutdown voltage regulators are re-enabled, and once their outputs are above reset threshold, the $\overline{\text{RESET_A}}$ signal is released, and the microcontroller will start its normal operation. The wake-up source is flagged in the PCR Status Register (PCR_SR (hi)).

The microcontroller has to acknowledge the Normal mode, by writing the OPM=00, to allow a controlled transition into the D2D Clock domain. If the clock domain transition is not required, the microcontroller may issue a Sleep / Stop mode entry instead (see Device clock tree for details on the limitations during the intermediate state).

6.2.2.2.8.2 Stop mode

Writing the PCR Control Register (PCR_CTL) with OPM=01, the MM9Z1_638 analog die will enter Stop mode with the configured wake-up sources (see Wake-up sources), after the D2DCLK signal has been stopped by the MCU die entering Stop.

Note

After writing the PCR Control Register (PCR_CTL) with OPM=01, the register content of the SCI (S08SCIV4) and TIMER (TIM16B4C) module registers are read only until Normal mode is entered again. This is important in case the MCU does not effectively enter STOP, due to an IRQ pending from one of the two blocks. (Having any analog die IRQ allowed when entering Low-power mode is not recommended).

During Stop mode, the MM9Z1_638 has the same behavior as during Sleep mode, except VDDX is still powered by the internal Clamp_5V, to supply the MCU Stop mode current. As this current is limited, the MCU die must be switched into Stop mode after sending the Stop command for the analog die.

If any enabled wake up condition occurs, the shutdown voltage regulators are re-enabled, and once their outputs are above the reset threshold, VDDX is switched to the main regulator, an D2D interrupt (D2DINT) is issued to wake-up the MCU, and the microcontroller will continue its normal operation. The wake-up source is flagged in the PCR Status Register (PCR_SR (hi)).

The microcontroller has to acknowledge the Normal mode by writing the OPM=00. This allows a controlled transition into the D2D Clock domain. If the clock domain transition is not required, the microcontroller may issue a Sleep / Stop mode entry instead (see Device clock tree for details on the limitations during the intermediate state).

At start-up or after wake-up, it is required to wait until the PLL is locked, if the PLL is enabled previous to access to the analog die through the D2D interface.

Note

After writing the PCR Control Register (PCR_CTL) with OPM=01, the reduced current capability of the MCU regulator supply (VDDX) has to be considered.

6.2.2.3 Power management

To support the various operating modes and modules in the MM9Z1_638, the following power management architecture has been implemented.

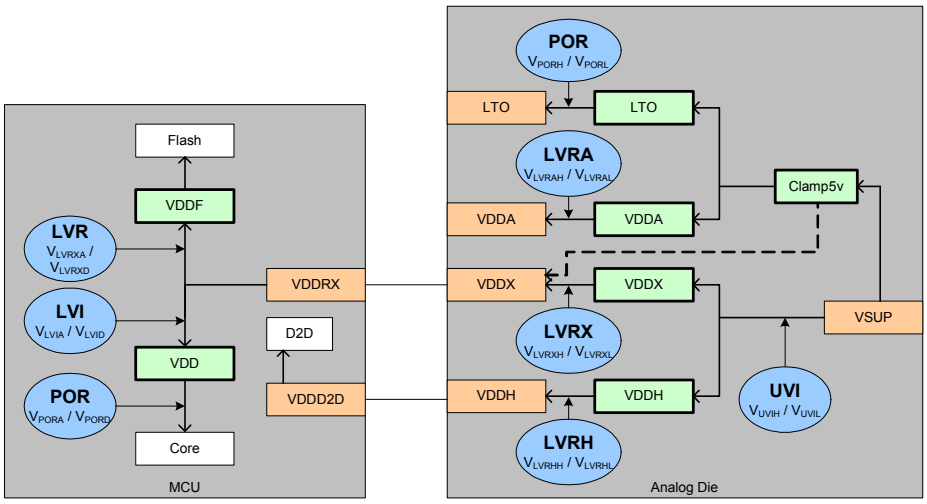


Figure 31. System voltage monitoring

6.2.2.3.1 Detailed power block description

See recommended external components under Typical applications.

6.2.2.3.1.1 VSUP

VSUP is the system power supply input, and must be reverse battery protected by an external diode. VSUP is monitored for undervoltage conditions (UVI). Once VSUP drops below V_{UVIL} an undervoltage interrupt (LVI) is issued.

Note

If the device has the Cranking mode feature enabled, the undervoltage threshold would be V_{UVCIL} instead of V_{UVIL}.

6.2.2.3.1.2 LTO

LTO is the low power 2.5 V digital supply voltage, supplying the permanently active blocks. It is based on the internal Clamp5v voltage and always on.

6.2.2.3.1.3 VDDX

VDDX is the Normal mode 5.0 V regulator output, suppling the LIN block and the microcontroller via the VDDX pin. During Sleep mode operation, the VDDX regulator is shut down. In Stop mode, a sub block of the VDDX regulator remain enabled to supply the microcontroller, while offering a low device current consumption.

6.2.2.3.1.4 VDDH

VDDH is the Normal mode 2.5 V regulator output, suppling only active blocks during Normal mode and the MCU Die to Die Interface, via the VDDH terminal. The VDDH regulator is shut down during both Low-power modes.

6.2.2.3.1.5 VDDA

VDDA is the 2.5 V analog supply voltage, active during Normal mode and I/T acquisitions. Only the circuitry related to usage of external temperature sensor can be connected to this terminal.

6.2.2.3.2 Power supply by module

The following table summarized the active regulators vs. module for the different operating modes.

Table 118. Power supply by module

Module / block	VDDH	VDDA	VDDL	VDDX
Gain Control Block (GCB)(107)	X	X		
Programmable Gain Amplifier (PGA)(108)		X		
I/T - ADC Converters(108)		X		
V - ADC Converters(107)		X		
Temperature Sensor(108)		X		
LIN(107)			X	X
D2D(107)	X			
LPOSC(109)			X	
Permanent Digital(109)			X	
Normal mode Digital(107)	X			

Notes:
107.Enabled in Normal mode only
108.Enabled when a measuring in Low-power mode and always in Normal mode
109.Permanently enabled

6.2.2.3.3 Power up / power down behavior

Several system voltage monitors have been implemented in both die, to guarantee a defined power up and power down system behavior. See Figure 31 for the various sensing points. The individual threshold levels are specified in Table 12 for the analog die and inTable 41 for the microcontroller.

Note

To differentiate between the MCU and analog die thresholds, the following symbol scheme is defined:

V_{xxxxA} - MCU Assert Level (lower threshold for low voltage events)
V_{xxxxD} - MCU Deassert Level (higher threshold for low voltage events)
V_{xxxxH} - Analog Die High Threshold Level (deassert threshold for low voltage events)
V_{xxxxL} - Analog Die Low Threshold Level (assert threshold for low voltage events)

6.2.2.3.4 Low voltage operation - cranking mode

The “Cranking” option is an option, allowing lower voltage operations to guarantee the MCU memory content during a standard cranking situation. As illustrated in Figure 32, the Cranking mode is introduced to maintain both die in a Stop mode alike state. The MCU die will remain in STOP with the RAM content being guaranteed until the PORA level is reached for the VDDRX supply.

The analog die will enter “Cranking Mode” upon the MCU command out of Normal mode, or when it reaches V_{UVCIL} during Stop mode, with the LVT bit set in the TRIM_LVT register.

Note

Executing Stop with VSUP < V_{UVCIL} and LVT = 1, the MM9Z1_638 will immediately enter Cranking mode.

During Cranking mode, the analog die will gate its internal oscillator to stop all ongoing acquisitions during the low power condition. Returning from Cranking mode will appear as a wake-up from undervoltage interrupt (UVI=1). The analog die will be in Intermediate mode after wake-up, and in this case, writing LVT = 0 will have no effect. The analog die can be sent into Normal mode (Stop, Sleep) by writing the OPM bits.

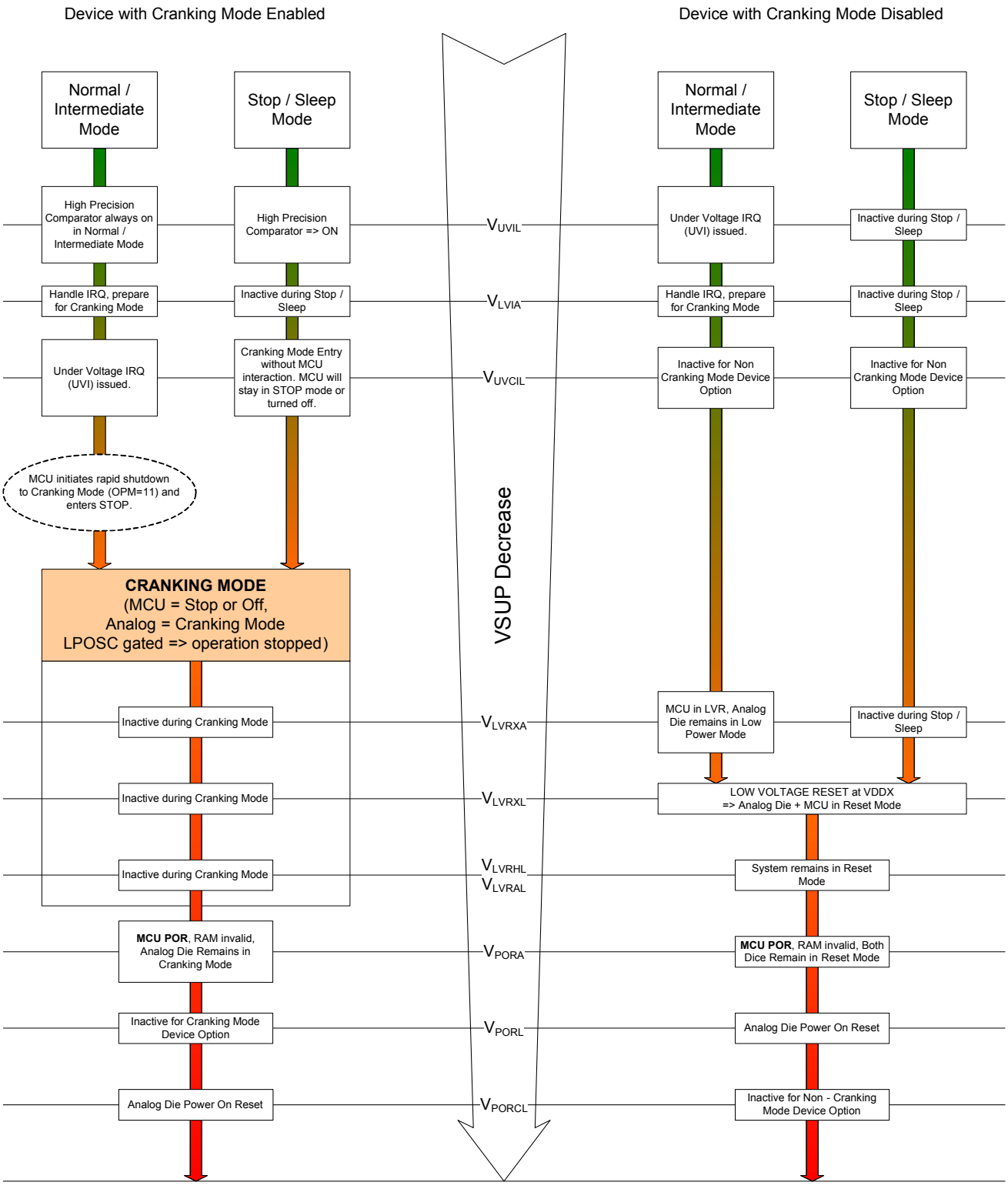


Figure 32. Power down sequence

6.2.2.4 Wake-up sources

Several wake-up sources have been implemented in the MM9Z1_638, to exit from Sleep or Stop mode.

Figure 33 shows the wake-up sources and the corresponding configuration and status bits.

To indicate the internal wake-up signal, a routing of the internal wake-up signal to the PTBx output (WKUP) is implemented. See General purpose I/O - GPIO, for additional details on the required configuration.

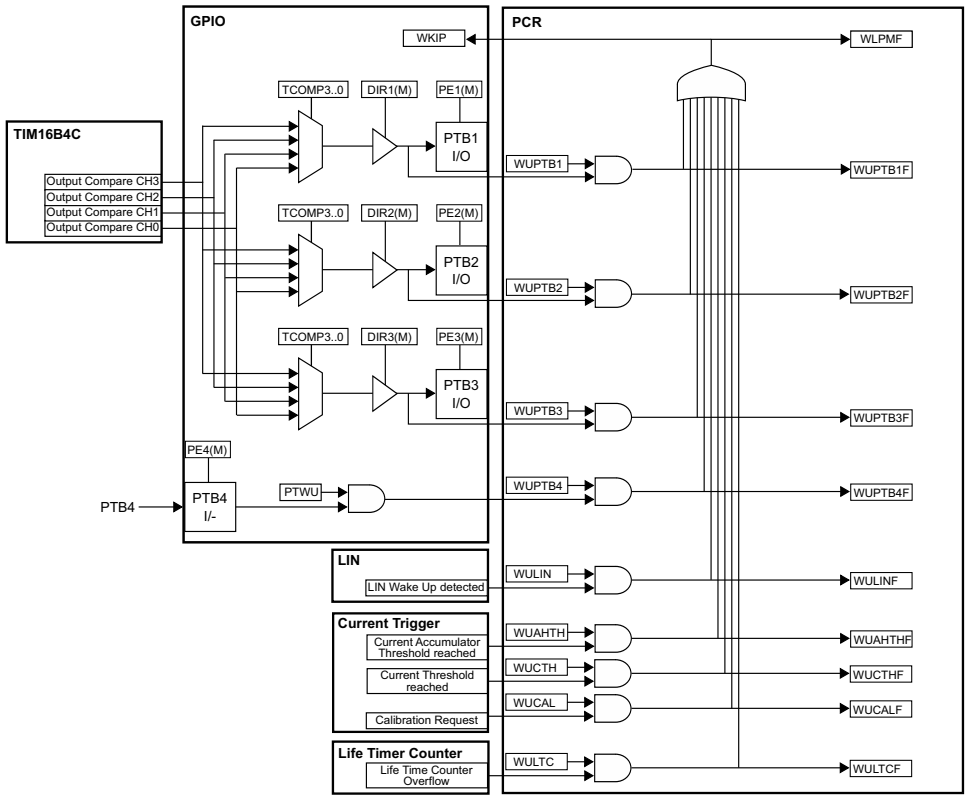


Figure 33. Wake-up sources

6.2.2.4.1 Wake-up source details

6.2.2.4.1.1 Cyclic current acquisition / calibration temperature check

A configurable (ACQ_TCMP) independent Low-power mode counter/trigger, based on the ALFCLK, has been implemented to trigger a cyclic current measurement during the Low-power modes. To validate that the temperature is still within the calibration range, the temperature measurement can be enabled during this event as well.

As a result of the cyclic conversions, three wake-up conditions are implemented.

- Current Threshold Wake-up
- Current Averaging Wake-up
- Calibration Request Wake-up

The configuration of the counter and the cyclic measurements is part of the acquisition paragraph (see Channel acquisition). The actual cyclic measurement does not wake-up the microcontroller unless one of the three wake-up conditions become valid.

6.2.2.4.1.1.1 Current threshold wake-up

Every cyclic current measurement result (absolute content of the ADC result I_CURR register) is compared with a programmable unsigned current threshold (CTH in the ACQ_CTH register).

The comparison is done with the CTH content left - shifted by 1, as shown in Figure 119.

Table 119. Current threshold comparison

	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTH[7:0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CTH[7:0]								0
ABS(CURR[23:0])	X	ABS(CURR[23:0])																						

If the absolute result is greater or equal to the programmed and shifted threshold, a filter counter is incremented (decremented if below). If the filter counter (8-Bit) reaches the programmable low power current threshold filtering period (ACQ_THF), a wake-up initiated if the Current Threshold Wake-up is enabled (WUCTH). The filter counter is reset every time a Low-power mode is entered. The implementation is shown in Figure 34.

The wake-up source is flagged with the WUCTHF Bit.

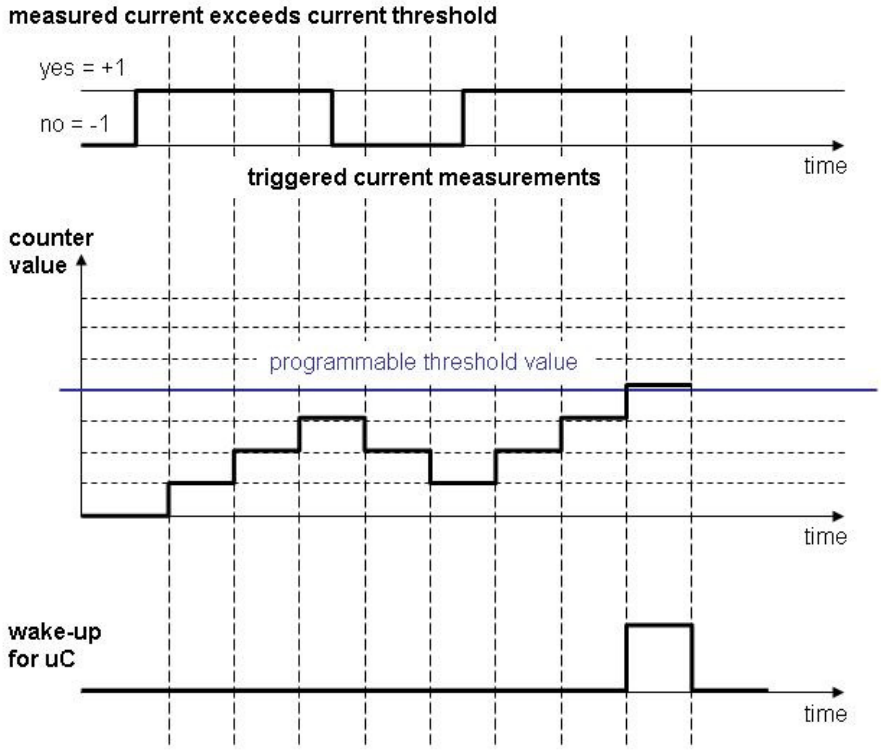


Figure 34. Current threshold - wake-up counter

6.2.2.4.1.1.2 Current ampere hour threshold wake-up

As shown in Figure 35, every cyclic current measurement (signed content of the ADC result ACQ_CURR register) is added to the 32-Bit (signed) current accumulator (ACQ_AHC) (both in two's complement format). If the absolute accumulator value reaches (|ACQ_AHC| ≥ ACQ_AHTH), the absolute programmable 31-Bit current threshold (ACQ_AHTH), a wake-up is initiated if the Current AH Threshold Wake-up is enabled (WUAHTH). The accumulator can be reset by writing a 1 into the AHCR register.The Ampere Hour Counter is counting after wake-up.

In Normal mode, the accumulator register ACQ_AHC can be read out at any time.

The wake-up source is flagged with the WUAHTHF Bit.

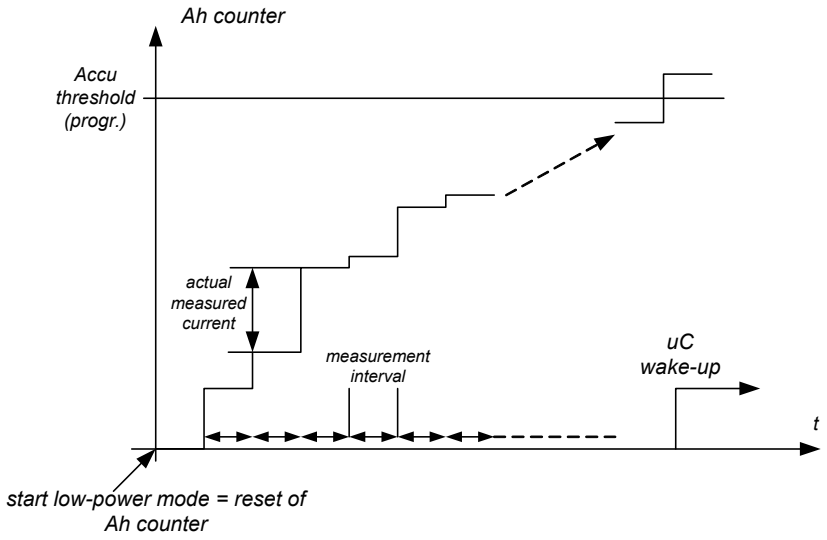


Figure 35. Ah counter function

6.2.2.4.1.1.3 Calibration request wake-up

Once the temperature measured during the cyclic sense is indicating a potential “out of calibration” situation, a wake-up is issued if the Calibration Request Wake-up is enabled (WUCAL). For additional details, refer to .

The wake-up source is flagged with the WUCALF Bit.

6.2.2.4.1.2 Timed wake-up

To generate a programmable wake-up timer, the integrated 4 Channel Timer Module is supplied during both low-power modes and clocked by the ALFCLK clock. To wake-up from one of the Low-power modes, the output compare signal (OC) of any of the 4 channels can be routed to the PTB[3:1] logic (standard feature also in Normal mode). Enabling the corresponding Wake-up Enable Bit (WUPTB[3:1]) will generate the wake up, once the timer output compare becomes active.

Note

Only the internal GPIO logic is active during the Low-power modes. The Port I/O structures will not be active.

To allow an accurate wake-up configuration during the clock transition, the timer should be configured before entering one of the Low-power modes, without the Timer Enable Bit (TEN) being set. Setting the Timer Wake-up Enable Bit WUPTB[3:1] will enable the TIMER OC as wake-up sources, and cause the Timer Enable Bit (TEN) to be set, once the timer clock domain was changed to the LPOSC clock.

During Low-power mode, only current and temperature measurements are performed, so only the current measurement channel is active with the temperature channel being optional - the voltage measurement channel is inactive. To reduce further the power consumption, only triggered current measurements are done. For this purpose, an independent Timer Module is used to periodically start a current measurement after a programmable time (ACQ_TCMP).

6.2.2.4.1.3 Wake-up from LIN

During Low-power mode, operation of the transmitter of the physical layer is disabled. The receiver remains active and able to detect wake-up events on the LIN bus line. For further details, refer to Section 6.18. The LIN wake-up occurs at the LIN dominant to recessive transition, following a LIN dominant level longer than $t_{PROPWUL}$. The wake-up source is flagged with the WULINF Bit.

6.2.2.4.1.4 Wake-up on PTB4 external wake-up pin

Once a Wake-up signal (low or high level, depending on setting of the NWUE bit in the GPIO_IN4 register) is detected on the PTB4 input, with the Wake-up Enable Bit (WUPTB4) and the port configuration bit (PTWU) set, a wake-up is issued. The wake-up source is flagged with the WUPTB4F Bit. This pin can be routed to an external CAN interface to detect CAN bus wake-up events.

6.2.2.4.1.5 Wake-up on life time counter overflow

The life time counter continues to run during Low-power mode, if configured. Once the counter overflows with the life time counter wake-up enabled (WULTC=1), a wake-up is issued. The wake-up source is flagged with the WULTC Bit. The Life Time Counter has to be configured in Normal mode only.

6.2.2.4.1.6 General wake-up indicator

The WLPMF bit can be used after a wake-up from low-power mode (STOP and SLEEP), to indicate the system is awake (clock domain is in normal mode). Prior requires to set analog die to normal mode, by writing OPM=00 in the PCR_CTL register.

6.2.2.5 Device clock tree

6.2.2.5.1 Clock scheme overview

There are two system oscillators implemented. The low power oscillator is located on the analog die, and is supplied permanently and has a nominal frequency of f_{OSCL}, providing a LPCLK clock signal. It is primarily used in Low-power mode, and as an independent clock source for the watchdog during Normal mode.

The high power oscillator is basically the internal or external microcontroller oscillator (active only during Normal mode). The high power oscillator is distributed to the analog die via the D2DCLK (via configurable MCU prescalers), and there it's divided into two clocks (D2DSCLK and D2DFCLK), based on the PRESC[15:0] prescaler. For the D2DSCLK, an additional 2 Bit divider PF[1:0] is implemented[110]. During Normal mode, D2DSCLK is continuously synchronizing the LPCLK, to create the accurate ALFCLK (See ALFCLK calibration), it's clock source of the TIM16B4C (Timer), and S08SCIV4 (SCI) module with a fixed by 4 divider.

Notes:

110.PF[1:0] is not implemented as a simple divider. To accomplish a D2DSCLK period ranging from 1.0 ms to 8.0 ms, the following scheme is used: 00 - 1; 01 - 2; 10 - 4; 11 - 8.

D2DSCLK - D2D slow clock (1... 0.125 kHz) ***Eqn. 1***

$$\left(D2DSCLK = \frac{D2DCLK}{(2^{PF[1,0]}) \times (PRESC[15,0])}\right)$$

D2DFCLK - D2D fast clock (512 kHz) ***Eqn. 2***

$$\left(D2DFCLK = \frac{D2DCLK}{2 \times (PRESC[15,10] + PRESC[9])}\right)$$

During Low-power mode, D2DCLK is not available. The low power oscillator is the only system clock.

Figure 36 and Figure 37 show the different clock sources for Normal and Low-power mode.

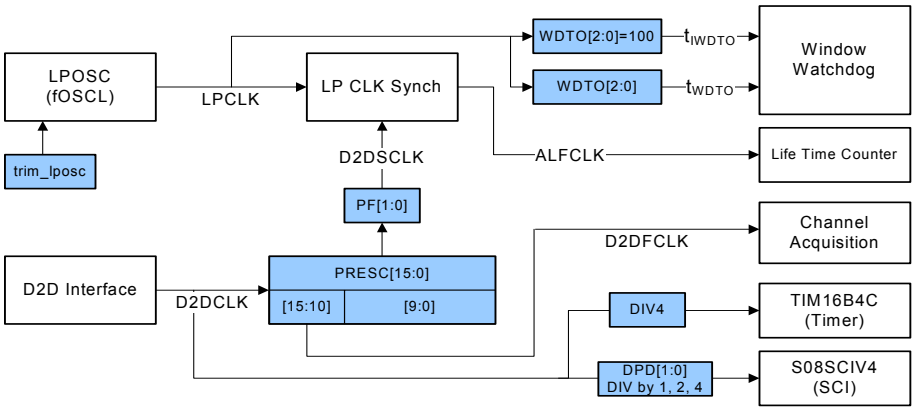


Figure 36. Clock tree overview - normal mode

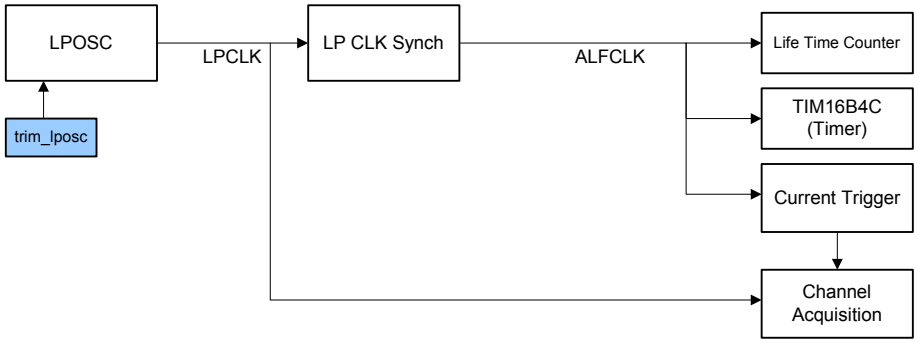


Figure 37. Clock tree overview - low-power modes

6.2.2.5.2 ALFCLK calibration

To increase the accuracy of the 1.0 kHz (or 2.0, 4.0, 8.0 kHz based on PF[1:0]) system clock (ALFCLK), the low power oscillator (LPCLK) is synchronized to the more precise D2DCLK, via the D2DSCLK signal. The “Calibrated Low Power Clock” (ALFCLK) could be trimmed to the D2DCLK accuracy plus a maximum error adder of 1 LPCLK period, by internally counting the number of periods of the LPCLK (512 kHz) during a D2DSCLK period. The APRESC[12:0] register will represent the calculated internal prescaler. The PRDF bit (Prescaler Ready flag) will indicate the synchronization complete after a power up or prescaler (PRESC/PF) change.

The adjustment is continuously performed during Normal mode. During Low-power mode (STOP or SLEEP), the last adjustment factor would be used.

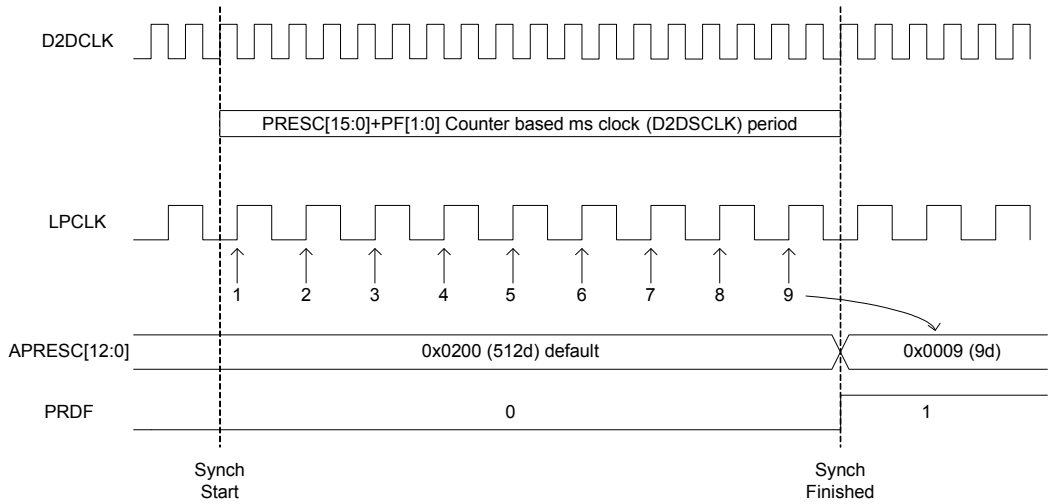


Figure 38. ALF clock calibration procedure during normal mode

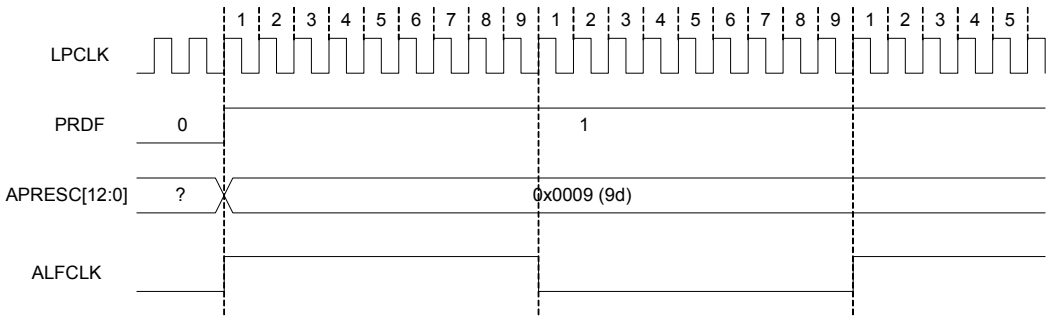


Figure 39. ALFCLK after calibration

6.2.2.5.3 Recommended clock settings

The prescaler (PCR_PRESC) settings has to be done using the following formula: PCR_PRESC = |D2DCLK(kHz)| (integer value of D2D clock in kHz). For details on the MCU divider settings, including POSTDIV and SYNDIV, see S12Z clock, reset and power management unit (S12ZCPMU).

6.2.2.6 System resets

To guarantee safe operation, several RESET sources have been implemented in the MM9Z1_638 device. Both the MCU and the analog die are designed to initiate reset events on internal sources and the MCU is capable of being reset by external events including the analog die reset output. The analog die is capable of being reset by the MCU in Stop and Cranking mode only.

Note

In normal mode the RESET_A is only an output. In sleep mode VDDRX = 0 V and the RESET_A/RESET lines are 0 V.

6.2.2.6.1 Device reset overview

The MM9Z1_638 reset concept includes two external reset signals, $\overline{\text{RESET}}$ (MCU) and $\overline{\text{RESET_A}}$ (analog Die). Figure 40 illustrates the general configuration.

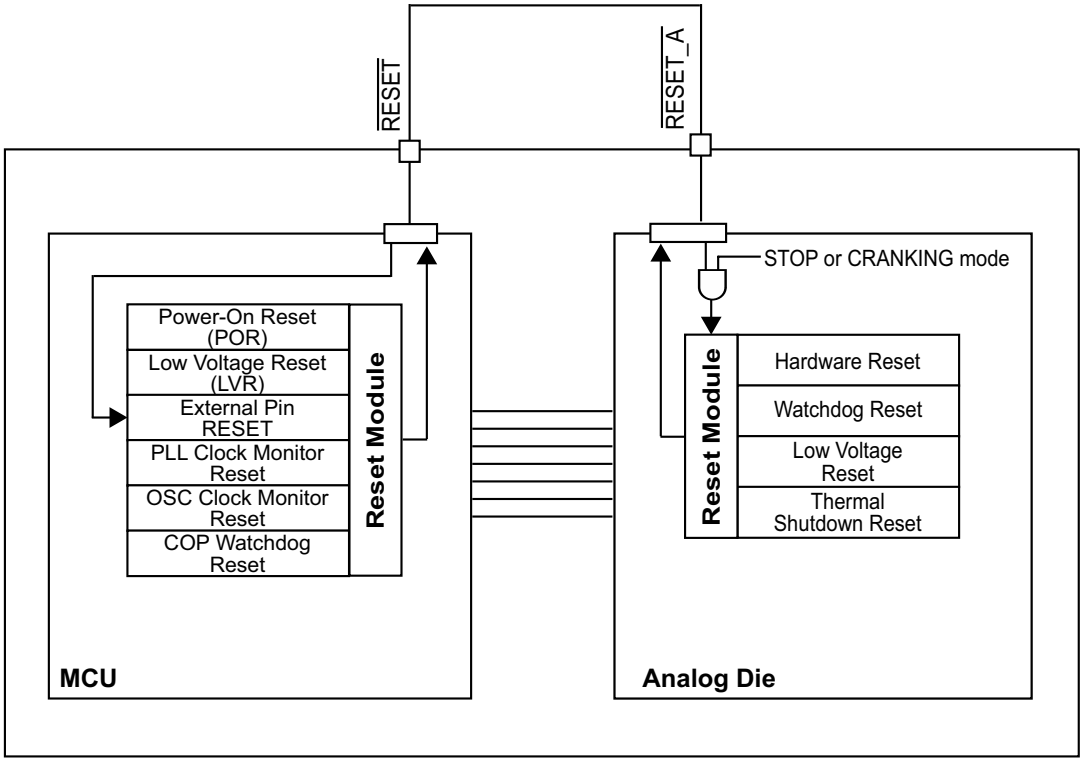


Figure 40. Device reset overview

Both $\overline{\text{RESET}}$ and $\overline{\text{RESET_A}}$ signals are low active I/Os, based on the 5.0 V supply (VDDRX for $\overline{\text{RESET}}$ and VDDX for $\overline{\text{RESET_A}}$).

6.2.2.6.2 Analog die reset implementation

There are 7 internal reset sources implemented in the analog die of the MM9Z1_638 that causing the internal analog die status to be reset to default (Internal analog RST), and to trigger an external reset, activating the $\overline{\text{RESET_A}}$ pin. In addition, during Stop and Cranking mode, an external reset at the $\overline{\text{RESET_A}}$ pin will also reset the analog die.

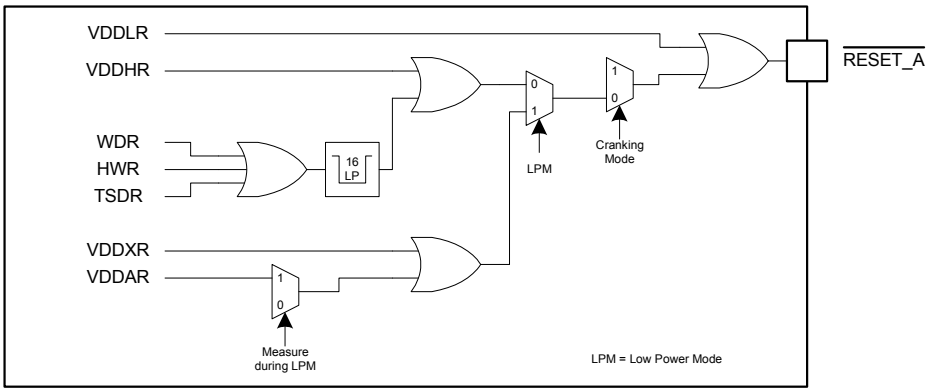


Figure 41. Analog die reset implementation

With the exception of the WDR, HWR, and TSDR, the $\overline{\text{RESET_A}}$ pin is driven active as long the condition is pending. The WDR, HWR, and TSDR will issue a 2 x LPCLK cycle active at the pin. During Cranking mode, only the VDDLR is active. During Low-power modes, only VDDXR and VDDAR are active reset sources. VDDAR is only active during active measurement in LPM. VDDXR and VDDAR are not active in Normal mode.

6.2.2.6.3 Reset source summary

- HWR - Hardware Reset
  - Forced internal reset caused by writing the HWR bin in the PCR_CTL register. The source will be indicated by the HWRF bit.
- WDR - Watchdog Reset
  - Window watchdog failure. The source will be indicated by the WDRF bit.
- LVR - Low Voltage Reset
  - The Voltage at the LTO, VDDH, VDDX, or VDDA has dropped below its reset threshold level. The source will be indicated for the LTO by the LVRF + HVRF, for the VDDA by the AVRF, and for the VDDH by the HVRF bit. VDDX resets are not indicated via individual reset flags. See Figure 41 for dependencies.
- TSDR - Temperature Shutdown Reset
  - The critical shutdown temperature threshold has been reached. VDDA, VDDX, and VDDH will be disabled as long as the overtemperature condition is pending(111) and the reset source is indicated by the HTF bit.
- External Reset
  - During Stop and Cranking(111) mode, a low signal at the $\overline{\text{RESET_A}}$ pin will reset the analog die. Since this condition can only be initiated by the microcontroller, no specific indicator flag is implemented.

Notes:
111.Resulting in a VDDH Low Voltage Reset taking over the reset after the 2 LPCLK reset pulse

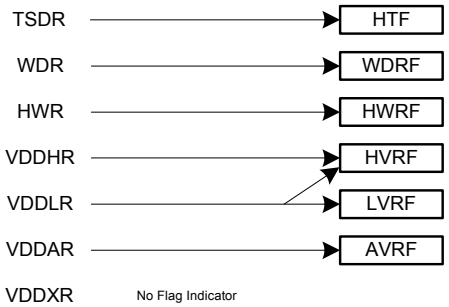


Figure 42. Reset status information

6.2.2.7 PCR - memory map and registers

6.2.2.7.1 Overview

This section provides a detailed description of the memory map and registers.

6.2.2.7.2 Module memory map

The memory map for the Analog Die - Power, Clock and Resets - PCR module is given in Figure 60

Table 120. Module memory map

Offset (112),(113)	Name		7	6	5	4	3	2	1	0
0x00	PCR_CTL (hi)	R	0	0	0	0	0	0	0	0
	PCR Control Register	W	HTIEM	UVIEM	HWRM		PFM		OPMM	
	PCR_CTL (lo)	R	HTIE	UVIE	0		PF		OPM	
	PCR Control Register	W			HWR					
0x02	PCR_SR (hi)	R	HTF	UVF	HWRF	WDRF	HVRF	LVRF	WULTCF	WLPMF
	PCR Status Register	W	Write 1 will clear the flags							
0x03	PCR_SR (lo)	R	WUAHTHF	WUCTHF	WUCALF	WULINF	WUPTB4F	WUPTB3F	WUPTB2F	WUPTB1F
	PCR Status Register	W	Write 1 will clear the flags							
0x04	PCR_PRESC (hi)	R	PRESC							
	PCR 1.0 ms prescaler	W								
	PCR_PRESC (lo)	R								
	PCR 1.0 ms prescaler	W								
0x06	PCR_WUE (hi)	R	WUAHTH	WUCTH	WUCAL	WULIN	WUPTB4	WUPTB3	WUPTB2	WUPTB1
	Wake-up Enable Register	W								
0x07	PCR_WUE (lo)	R	WULTC	0	0	0	0	0	HTWF	TSDF
	Wake-up Enable Register	W								
0x0E	TRIM_ALF (hi)	R	PRDF	0	0	APRESC[12:8]				
	Trim for accurate 1.0 ms low freq clock	W								
0x0F	TRIM_ALF (lo)	R	APRESC[7:0]							
	Trim for accurate 1.0 ms low freq clock	W								

Notes:
112.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address Space.
113.Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

6.2.2.7.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

6.2.2.7.3.1 PCR control register (PCR_CTL)

Table 121. PCR control register (PCR_CTL)

Offset (114), (115)	0x00						Access: User read/write	
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R	0	0	0	0	0	0	0	0
W	HTIEM	UVIEM	HWRM	0	PFM[1:0]		OPMM[1:0]	
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	HTIE	UVIE	0	0	PF[1:0]		OPM[1:0]	
W			HWR	0				
Reset	0	0	0	0	0	0	0	0

Notes:
114.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
115.Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

Table 122. PCR control register (PCR_CTL) - register field descriptions

Field	Description
15 HTIEM	**High temperature interrupt enable mask** 0 - writing the HTIE bit will have no effect 1 - writing the HTIE bit will be effective
14 UVIEM	**Supply undervoltage interrupt enable mask** 0 - writing the UVIE bit will have no effect 1 - writing the UVIE bit will be effective
13 HWRM	**Hardware reset mask** 0 - writing the HWR bit will have no effect 1 - writing the HWR bit will be effective
12 Reserved	Reserved. Must remain “0”
11-10 PFM[1:0]	**Prescaler factor mask** 00,01,10 - writing the PF bits will have no effect 1 - writing the PF bits will be effective
9-8 OPMM[1:0]	**Operation mode mask** 00,01,10 - writing the OPM bits will have no effect 11 - writing the OPM bits will be effective
7 HTIE	**High Temperature Interrupt enable.** Writing only effective with corresponding mask bit HTIEM set. 0 - High temperature interrupt (HTI) enabled 1 - High temperature interrupt (HTI) disabled
6 UVIE	**Low supply voltage interrupt enable.** Writing only effective with corresponding mask bit UVIEM set. 0 - Low supply voltage interrupt (UVI) enabled 1 - Low supply voltage interrupt (UVI) disabled
5 HWR	**Hardware Reset.** Writing only effective with corresponding mask bit HWRM set. Write only. 0 - No effect 1 - All analog die digital logic is reset and external reset ($\overline{\text{RESET_A}}$) is set to reset the MCU.
4 Reserved	Reserved. Must remain “0”

Table 122. PCR control register (PCR_CTL) - register field descriptions (continued)

Field	Description
3-2 PF[1:0]	**1.0 ms Prescaler.** Writing only effective with corresponding mask bits PFM set to 11. 00 - 1 01 - 2 10 - 4 11 - 8
1-0 OPM[1:0]	**Operation mode select.** Writing only effective with “11” mask bits OPMM set to 11. 00 - Normal mode 01 - Stop mode 10 - Sleep mode 11 - Cranking mode

6.2.2.7.3.2 PCR status register (PCR_SR (hi))

Table 123. PCR status register (PCR_SR (hi))

Offset[116]	**0x02**						**Access: User read/write**	
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	HTF	UVF	HWRF	WDRF	HVRF	LVRF	WULTCF	WLPMF
W	Write 1 will clear the flags[117]							
Reset	0	0	0	0	0	0	0	0

Notes:

116.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

117.HTF and UVF represent the current status and cannot be cleared. Writing 1 to HTF / UVF will clear the Interrupt flag in the Interrupt Source Register and Interrupt Vector Register instead.

Table 124. PCR status register (PCR_SR (hi)) - register field descriptions

Field	Description
7 HTF	**High Temperature Condition Flag.** This bit is set once a temperature warning is detected, or the last reset being caused by a temperature shutdown event (TSDR). Writing HTF=1 will clear the flag and the interrupt flag in the Interrupt Source Register and Interrupt Vector Register, if the condition is gone. 0 - No High Temperature condition detected. 1 - High Temperature condition detected or last reset = TSDR.
6 UVF	**Supply Undervoltage Condition Flag.** This bit is set once a undervoltage warning is detected. Writing UVF=1 will clear the flag and the Interrupt flag in the Interrupt Source Register and Interrupt Vector Register, if the condition is gone (UVF=0). 0 - No undervoltage condition detected. 1 - Undervoltage condition detected.
5 HWRF	**Hardware Reset Flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by a HWR command.
4 WDRF	**Watchdog Reset Flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by the analog die window watchdog.
3 HVRF	**VDDH Low Voltage Reset Flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by a low voltage condition at the VDDH regulator. (LVRF = 0) 1 - Last reset was caused by a low voltage condition at the LTO regulator. (LVRF = 1)
2 LVRF	**LTO Low Voltage (POR) Reset Flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by a low voltage condition at the LTO regulator. (Power on Reset - POR)

Table 124. PCR status register (PCR_SR (hi)) - register field descriptions (continued)

Field	Description
1 WULTCF	**Life Time Counter Wake-up Flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last Wake-up was caused by a life time counter overflow
0 WLPMF	**Wake-up after Low-power Mode Flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after Low-power mode.

6.2.2.7.3.3 PCR status register (PCR_SR (lo))

Table 125. PCR status register (PCR_SR (lo))

Offset[118]	0x03					Access: User read/write		
	7	6	5	4	3	2	1	0
R	WUAHTHF	WUCTHF	WUCALF	WULINF	WUPTB4F	WUPTB3F	WUPTB2F	WUPTB1F
W	Write 1 will clear the flags							
Reset	0	0	0	0	0	0	0	0

Notes:
118.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 126. PCR status register (PCR_SR (lo)) - register field descriptions

Field	Description
7 WUAHTHF	**Wake-up on Ah counter threshold Flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after Ah counter threshold reached.
6 WUCTHF	**Wake-up on current threshold Flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after current threshold reached.
5 WUCALF	**Wake-up on calibration request flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after calibration request.
4 WULINF	**Wake-up on LIN flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after LIN wake-up detected
3 WUPTB4F	**Wake-up on GPIO 4 event (TIMER output compare or external wake up) flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after GPIO 4 event
2 WUPTB3F	**Wake-up on GPIO 3 event (TIMER output compare) flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after GPIO 3 event
1 WUPTB2F	**Wake-up on GPIO 2 event (TIMER output compare) flag**. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after GPIO 2 event
0 WUPTB1F	**Wake-up on GPIO 1 event (TIMER output compare) flag.** Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Indicates wake-up after GPIO 1 event

6.2.2.7.3.4 PCR 1.0 ms prescaler (PCR_PRESC)

Table 127. PCR 1.0 ms prescaler (PCR_PRESC)

Offset (119),(120)	0x04	Access: User read/write

	15	14	13	12	11	10	9	8
R W	PRESC[15:8]							
Reset	0	1	1	1	1	1	0	1
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	PRESC[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes:
119.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
120.This Register is 16 Bit access only.

Table 128. PCR 1.0 ms prescaler (PCR_PRESC) - register field descriptions

Field	Description
15-0 PRESC[15:0]	1.0 ms Prescaler, used to derive D2DSCLK and D2DFCLK from the D2DCLK signal. See Device clock tree for details.

6.2.2.7.3.5 Wake-up enable register (PCR_WUE (hi))

Table 129. Wake-up enable register (PCR_WUE (hi))

Offset(121)	0x06	Access: User read/write

	7	6	5	4	3	2	1	0
R W	WUAHTH	WUCTH	WUCAL	WULIN	WUPTB4	WUPTB3	WUPTB2	WUPTB1
Reset	0	0	0	0	0	0	0	0

Notes:
121.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 130. Wake-up enable register (PCR_WUE (hi)) - register field descriptions

Field	Description
7 WUAHTH	0 - Wake-up on Ah counter disabled 1 - Wake-up on Ah counter enabled
6 WUCTH	0 - Wake-up on current threshold disabled 1 - Wake-up on current threshold enabled
5 WUCAL	0 - Wake-up on calibration request disabled 1 - Wake-up on calibration request enabled
4 WULIN	0 - Wake-up on LIN disabled 1 - Wake-up on LIN enabled
3 WUPTB4	0 - Wake-up on GPIO 4 event disabled 1 - Wake-up on GPIO 4 event enabled
2 WUPTB3	0 - Wake-up on GPIO 3 event disabled 1 - Wake-up on GPIO 3 event enabled

Table 130. Wake-up enable register (PCR_WUE (hi)) - register field descriptions (continued)

Field	Description
1 WUPTB2	0 - Wake-up on GPIO 2 event disabled 1 - Wake-up on GPIO 2 event enabled
0 WUPTB1	0 - Wake-up on GPIO 1 event disabled 1 - Wake-up on GPIO 1 event enabled

6.2.2.7.3.6 Wake-up enable register (PCR_WUE (lo))

Table 131. Wake-up enable register (PCR_WUE (lo))

Offset[122]	0x07					Access: User read/write		
	7	6	5	4	3	2	1	0
R	WULTC	0	0	0	0	0	HTWF	TSDF
W								
Reset	0	0	0	0	0	0	0	0

Notes:
122.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 132. Wake-up enable register (PCR_WUE (lo)) - register field descriptions

Field	Description
7 WULTC	0 - Wake-up on Life Timer Counter Overflow disabled 1 - Wake-up on Life Timer Counter Overflow enabled
1 HTWF [123]	0 - Thermal prewarning flag not set 1 - Thermal prewarning flag set
0 TSDF	0 - Thermal shutdown flag not set 1 - Thermal shutdown fag set

Notes:
123.The HTWF flag can be cleared even if the overtemperature condition remains. The HTWF flag will be set again only if the temperature falls below the threshold and rises again above the threshold.

6.2.2.7.3.7 Trim for accurate 1.0 ms low freq clock (TRIM_ALF (hi))

Table 133. Trim for accurate 1.0 ms low freq clock (TRIM_ALF (hi))

Offset[124]	0x0E					Access: User read		
	15	14	13	12	11	10	9	8
R	PRDF	0	0	APRESC[12:8]				
W								

Notes:
124.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

6.2.2.7.3.8 Trim for accurate 1.0 ms low freq clock (TRIM_ALF (lo))

Table 134. Trim for accurate 1.0 ms low freq clock (TRIM_ALF (lo))

Offset[125]	0x0F						Access: User read	
	7	6	5	4	3	2	1	0
R	APRESC[7:0]							
W								

Notes:
125.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 135. Trim for accurate 1.0 ms low freq clock (TRIM_ALF (lo)) - register field descriptions

Field	Description
15 PRDF	**ALFCLK Prescaler ready Flag** 0 - The ALFCLK synchronization after power up or PRESC[15:0] / PF[1:0] change is not completed. 1 - The ALFCLK synchronization is complete. The ALFCLK signal is synchronized to the D2DCLK.
12-0 APRESC[12:0]	**ALFCLK Prescaler** This read only value represents the current ALFCLK prescaler value. With the synchronization complete (PRDF=1), the prescaler is used to create the calibrated clock for the Life Time Counter (Normal mode and Low-power mode), and Timer and Current trigger (Low-power mode only), based on the low power oscillator. After Power Up, the APRESC register is reset to 0x0200 (512dec) until the first synchronization is complete. This will initialize the ALFCLK to 1.0 kHz.

6.2.3 Window watchdog

The MM9Z1_638 analog die includes a configurable window watchdog which is active in Normal mode. The watchdog module is based on the Low Power Oscillator (LPCLK) to operate independently from the MCU based D2DCLK clock. The watchdog timeout (t_{WDTO}) can be configured between 4.0 ms and 2048 ms using the watchdog control register (WD_CTL).

Note

As the watchdog timing is based on the LPCLK, its accuracy is based on the trimming applied to the TRIM_OSC register. The given timeout values are typical values only.

During Low-power mode, the watchdog feature is not active, a D2D read during Stop mode will have the WDOFF bit set. After wake-up and transition to Normal mode, the watchdog is reset to the same state as when following a Power-On-Reset (POR).

To clear the watchdog counter, an alternating write has to be performed to the watchdog rearm register (WD_RR). The first write after the wake-up or $\overline{\text{RESET_A}}$ has been released has to be 0xAA, the next one has to be 0x55.

After the wake-up or $\overline{\text{RESET_A}}$ has been released, there will be a standard (non window) watchdog active with a fixed timeout of t_{IWDTO} (t_{WDTO} = b100 = 256 ms). The Watchdog Window Open (WDWO) bit is set during that time.

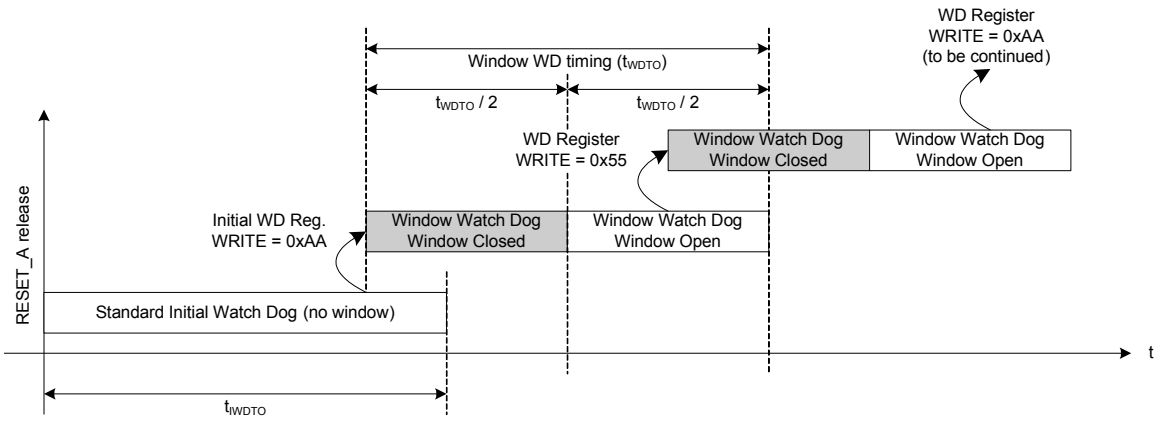


Figure 43. MM9Z1_638 analog die watchdog operation

To change from the standard initial watchdog to the window watchdog, the initial counter reset has to be performed by writing 0xAA to the Watchdog rearm register (WD_RR) before t_{IWDTO} is reached.

If the t_{IWDTO} timeout is reached with no counter reset or a value different from 0xAA written to the WD_RR, a watchdog reset will occur.

Once entering Window Watchdog mode, the first half of the time, t_{WDTO} is forbidden for a counter reset. To reset the watchdog counter, a alternating write of 0x55 and 0xAA has to be performed within the second half of the t_{WDTO}. A Window Open (WDWO) flag will indicate the current status of the window. A timeout or wrong value written to the WD_RR will force a watchdog reset.

If the first write to the WD_CTL register is 000 (WD OFF), the WD will be disabled(126).

Notes:
126.The Watchdog can be enabled any time later.

6.2.3.1 Memory map and registers

6.2.3.1.1 Overview

This section provides a detailed description of the memory map and registers.

6.2.3.1.2 Module memory map

The memory map for the Watchdog module is given in Table 60.

Table 136. Module memory map

Offset (127),(128)	Name		7	6	5	4	3	2	1	0
0x10	WD_CTL Watchdog control register	R	0	0	0	0	0	0	0	0
		W	WDTSTM					WDTOM[2:0]		
		R	WDTST	0	0	0	0	WDTO[2:0]		
		W								
0x12	WD_SR	R	0	0	0	0	0	0	WDOFF	WDWO
	Watchdog status register	W								
0x13	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x14	WD_RR	R	WDR[7:0]							
	Watchdog rearm register	W								
0x15	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x16	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x17	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes:
127.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
128.Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

6.2.3.1.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bits and field function follow the register diagrams, in bit order.

6.2.3.1.3.1 Watchdog control register (WD_CTL)

Table 137. Watchdog control register (WD_CTL)

Offset (129),(130)	0x10						Access: User write	
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R	0	0	0	0	0	0	0	0
W	WDTSTM					WDTOM		
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	WDTST	0	0	0	0	WDTO		
W								
Reset	1	0	0	0	0	1	0	0

Notes:
129.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
130.This Register is 16 Bit access only.

Table 138. Watchdog control register (WD_CTL) - register field descriptions

Field	Description
15 WDTSTM	**Watchdog Test - Mask** 0 - writing the WDTST bit will have no effect 1 - writing the WDTST bit will be effective
10-8 WDTOM[2:0]	**Watchdog Timeout - Mask** 0 - writing the WDTO bits will have no effect 1 - writing the WDTO bits will be effective
7 WDTST	**Watchdog Test** This bit is implemented for test purpose and has no function in Normal mode.
2-0 WDTO[2:0]	**Watchdog Timeout Configuration** - configuring the watchdog timeout duration t_{WDTO}. 000 - Watchdog OFF 001 - 4.0 ms 010 - 16.0 ms 011 - 64.0 ms 100 - 256 ms (default) 101 - 512 ms 110 - 1024 ms 111 - 2048 ms

6.2.3.1.3.2 Watchdog status register (WD_SR)

Table 139. Watchdog status register (WD_SR)

Offset(131)	0x12						Access: User read	
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	0	0	0	0	0	0	WDOFF	WDWO
W								

Notes:
131.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 140. Watchdog status register (WD_SR) - register field descriptions

Field	Description
1 WDOFF	**Watchdog Status** - Indicating the watchdog module being enabled/disabled 1 - Watchdog Off 0 - Watchdog Active
0 WDWO	**Watchdog Window Status** 1 - Open - Indicating the watchdog window is currently open for counter reset. 0 - Closed - Indicating the watchdog window is currently closed for counter reset. Resetting the watchdog with the window closed will cause a watchdog - reset.

6.2.3.1.3.3 Watchdog rearm register (WD_RR)

Table 141. Watchdog rearm register (WD_RR)

Offset[132] **0x14** **Access: User read/write**

	7	6	5	4	3	2	1	0
R W	WDR							
Reset	0	0	0	0	0	0	0	0

Notes:
132.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 142. Watchdog rearm register (WD_RR) - register field descriptions

Field	Description
7-0 WDR[7:0]	**Watchdog rearm register** - Writing this register with the correct value (0xAA alternating 0x55) while the window is open will reset the watchdog counter. Writing the register while the watchdog is disabled will have no effect.

6.3 Channel acquisition

6.3.1 Features

6.3.1.1 Current measurement - ISENSE features

- Dedicated 16 Bit Sigma Delta (ΣΔ) ADC
- Programmable gain amplifier (PGA) with 4 programmable gain factors
- Gain control block (GCB) for automatic gain adjustment
- Simultaneous sampling with voltage channel
- Programmable gain and offset compensation
- Chopper mode with moving average
- Optional automatic temperature dependant gain compensation
- SINC3 + IIR stage
- Calibration mode to compute compensation buffers
- Programmable low pass filter (LPF), configuration shared with the voltage measurement channel
- Optional Shunt resistor sensing feature
- Triggered sampling during Low-power mode with programmable wake-up conditions

6.3.1.2 Voltage measurement - VSENSE features

- Dedicated 16 Bit Sigma Delta (ΣΔ) ADC
- Four external voltage inputs with individual resistor divider (VSENSE0, VSENSE1, VENSE2 and VSENSE3)
- Five external voltage inputs with direct access (no divider) to Sigma delta (PTB0, PTB1, PTB2, PTB3, PTB4)
- Fixed high precision and calibrated divider for each VSENSE input

- Simultaneous sampling with current channel
- Programmable gain and offset compensation
- Optional automatic temperature dependant gain compensation
- Calibration mode to compute compensation buffers
- Optional Chopper mode with moving average (chopper mode recommended to achieve better performance)
- SINC3 + IIR stage
- Programmable low pass filter (LPF), configuration shared with current measurement channel

6.3.1.3 Temperature measurement - TSENSE features

- Internal on chip temperature sensor
- Five optional external temperature sensor inputs: PTB0, PTB1, PTB2, PTB3 and PTB4
- Dedicated PTB5 input for connection to GND of external temperature sensors and disconnection in Sleep and Stop modes.
- Dedicated 16-Bit Sigma Delta ADC
- Programmable gain and offset compensation
- External sensor supply (VDDA) with decoupling capacitor
- Optional measurement during Low-power mode to trigger recalibration

6.3.2 Block diagrams

6.3.2.1 Current measurement - ISENSE block diagram

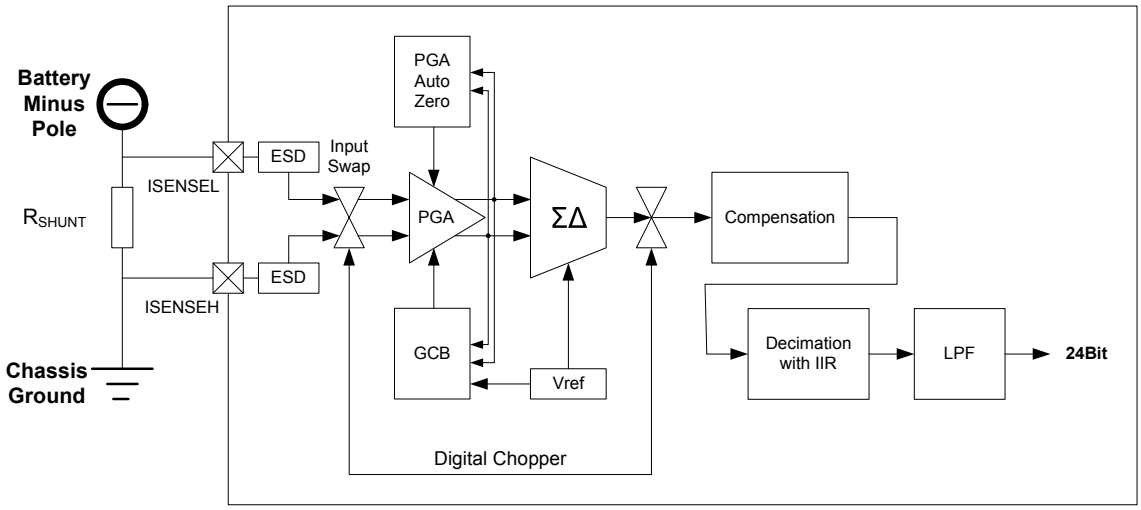


Figure 44. Current measurement channel

The battery current is measured by measuring the voltage drop V_{DROP} over an external shunt resistor, connected to ISENSEH and ISENSEL. V_{DROP} is defined as the differential voltage between the ISENSEL and ISENSEH inputs (V_{DROP}=ISENSEL-ISENSEH). A positive voltage drop means a positive current is flowing, and vice versa.

If the GND pin of the module is connected to ISENSEH, the measured current includes the supply current of the MM9Z1_638 (current flows back to negative battery pole). If the GND pin is connected to the ISENSEL input, the supply current of the MM9Z1_638 is not measured. However, the voltage at the ISENSEH input could go below GND (see Absolute maximum ratings). In this case, the current measurement still functions as specified.

6.3.2.2 Voltage measurement - VSENSE block diagram

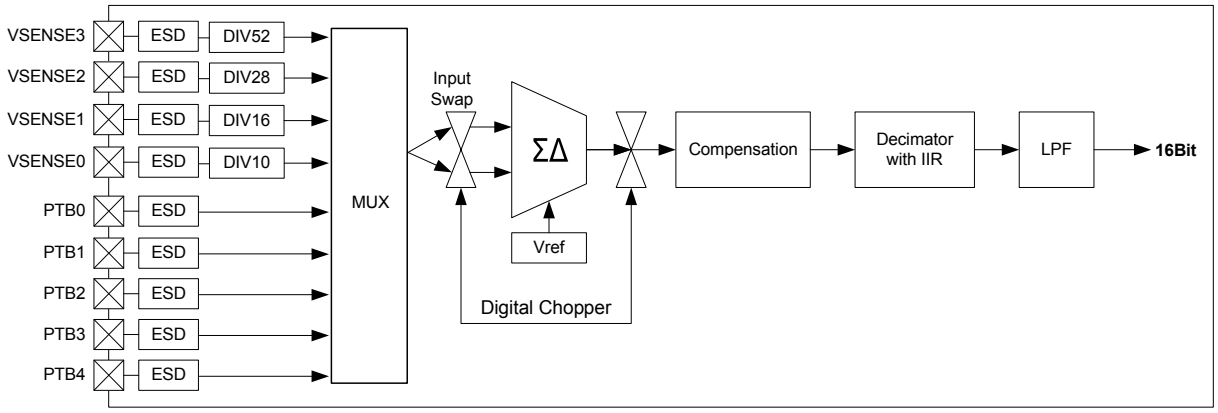


Figure 45. Voltage measurement channel

The voltage can be measured via the on VSENSEX or via PTB0, PTB1,PTB2, PTB3, and PTB4 inputs. For VSENSEX, high precision divider stage scales down the voltage by a fixed factor (K =1/10, 1/16, 1/28, and 1/52 for VSENSE0, VSENSE1, VSENSE2 and VSENSE3 respectively), to a voltage below the internal reference voltage of the Sigma Delta ADC (VSENSEX * K < V_{REF}). For PTB0 to PTB4, the voltage is directly applied to the Sigma Delta ADC.

6.3.2.3 Temperature measurement - TSENSE block diagram

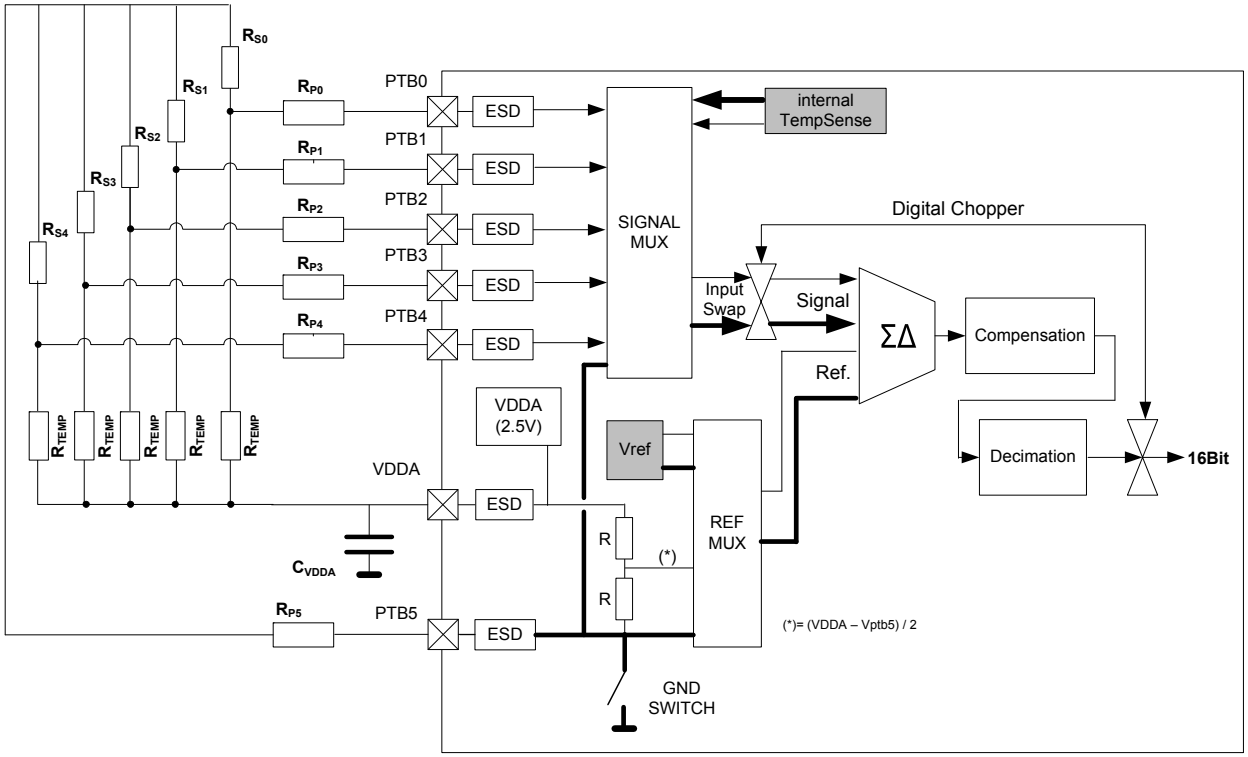


Figure 46. Temperature measurement channel

The external temperature can be measured via external temperature sensors supplied from VDDA. Up to five external sensor can be connected. The GND return path of the temperature sensor should be connected to PTB5. The GND SWITCH is automatically closed when external temperature sense is selected (ETMEN bit). Otherwise, the GND SWITCH is open to reduce consumption of temperature sensor resistor dividers.

6.3.3 Channel acquisition

6.3.3.1 Introduction

This chapter documents the current, voltage, and temperature acquisition flow. The chapter is structured in the following sections.

- Channel structure overview
- Current and voltage measurement
  - Shunt sense, PGA, and GCB (current channel only)
  - Voltage sense multiplexer (voltage channel only)
  - Sigma delta converter
  - Compensation
  - IIR / decimation / chopping stage
  - Low pass filter
  - Format and clamping
- Temperature measurement channel
  - Compensation
- Calibration
- Memory map and registers

6.3.3.2 Channel structure overview

The MM9Z1_638 offers three parallel measurement channels: Current, voltage, and temperature. The voltage channel is shared between the four VSENSEX and the five PTB0 to 4 input sources, the temperature channel is shared between internal temperature sensor and up to five external temperature sensors.

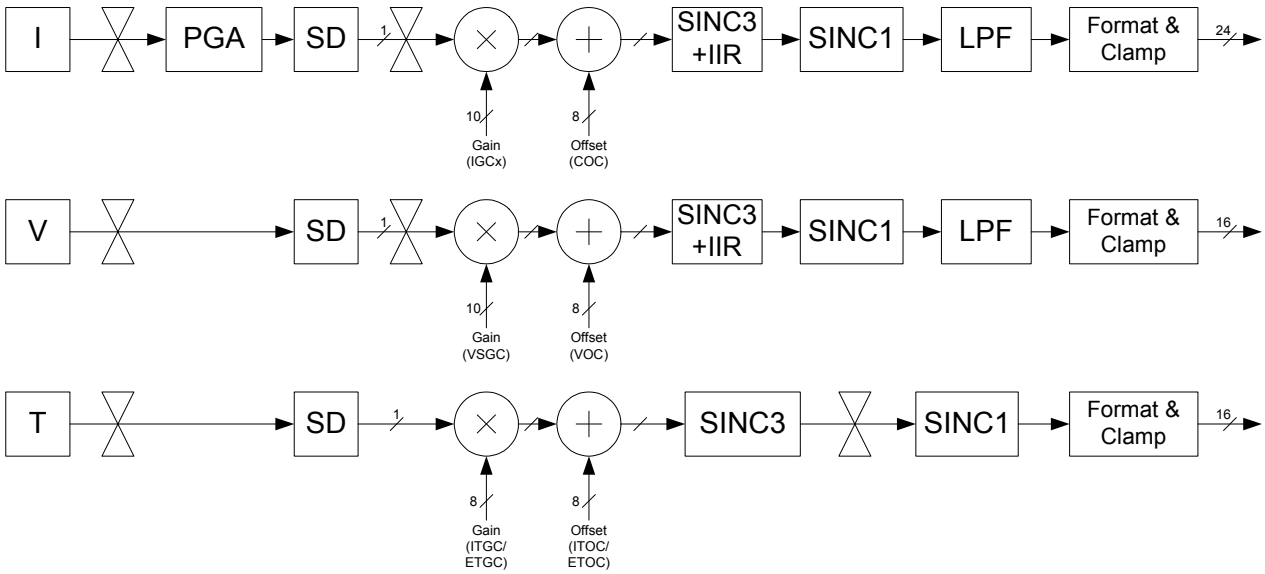


Figure 47. Simplified measurement channel

6.3.3.3 Current and voltage measurement

To guarantee synchronous voltage and current acquisition, both channels are implemented equal in terms of digital signal conditioning and timing. The analog signal conditioning, before the Sigma Delta Converter, is different to match the different sources.

The current channel is set into chopper mode. The voltage channel can be configured into chopper or non chopper mode (according to the value of CVCHOP bit).

The voltage and current channel output will be synchronized whatever the setting of the voltage channel. Both channels will perform synchronized conversions when enabled with a single write to the ACQ_CTL register.

6.3.3.3.1 Shunt sense, PGA, and GCB (current channel only)

Current channel specific analog signal conditioning.

6.3.3.3.1.1 Shunt sense

An optional current sense feature is implemented to sense the presence of the current shunt resistor. Setting the OPENE bit (COMP_CTL register), will activate the feature. The OPEN bit (COMP_SR register) will indicate the shunt resistor open.

The sense feature will detect an open condition for a shunt resistance R_{SHUNT} > ROPEN, or for an open GND.

Note

The sense feature influences the current sense result.

6.3.3.3.1.2 Programmable gain amplifier (PGA)

To allow a wide range of current levels to be measured, a programmable gain amplifier is implemented. Following the input chopper (see IIR / decimation / chopping stage), the differential voltage is amplified by one of the four gains controlled by the Gain Control Block.

The programmable gain amplifier (PGA) has a temperature depending offset error which can lead to errors in the current measurement result. Due to the differential architecture of the PGA, the offset error itself is no issue as long as no saturation occurs. Therefore it is required to regularly cancel the offset using the PGA auto zero sequence (Section 6.3.3.5.2, “PGA auto zero sequence"). The PGA Autozero Sequence should be performed, for example after startup/wake-up and if the chip temperature has changed.

6.3.3.3.1.3 Gain control block (GCB)

To allow a transparent Gain adjustment with minimum MCU load, an automatic gain control has been implemented. The absolute output of the PGA is constantly compared with a programmable up and down threshold (ACQ_GCB register). The threshold is a D/A output (V_{REF} = 1.25 V) according Table 143.

Table 143. Gain control block - register

ACQ_GCB D[7:0]	GCB high (up) threshold	ACQ_GCB D[7:0]	GCB low (down) threshold
0000xxxx	1/16 V_{REF}	xxxx0000	0
0001xxxx	2/16 V_{REF}	xxxx0001	1/16 V_{REF}
0010xxxx	3/16 V_{REF}	xxxx0010	2/16 V_{REF}
0011xxxx	4/16 V_{REF}	xxxx0011	3/16 V_{REF}
0100xxxx	5/16 V_{REF}	xxxx0100	4/16 V_{REF}
0101xxxx	6/16 V_{REF}	xxxx0101	5/16 V_{REF}
0110xxxx	7/16 V_{REF}	xxxx0110	6/16 V_{REF}
0111xxxx	8/16 V_{REF}	xxxx0111	7/16 V_{REF}
1000xxxx	9/16 V_{REF}	xxxx1000	8/16 V_{REF}
1001xxxx	10/16 V_{REF}	xxxx1001	9/16 V_{REF}
1010xxxx	11/16 V_{REF}	xxxx1010	10/16 V_{REF}
1011xxxx	12/16 V_{REF}	xxxx1011	11/16 V_{REF}
1100xxxx	13/16 V_{REF}	xxxx1100	12/16 V_{REF}
1101xxxx	14/16 V_{REF}	xxxx1101	13/16 V_{REF}
1110xxxx	15/16 V_{REF}	xxxx1110	14/16 V_{REF}
1111xxxx	16/16 V_{REF}	xxxx1111	15/16 V_{REF}

Once the programmed threshold is reached, the gain is adjusted to the next level. The currently active gain setting can be read in the IGAIN[1:0] register. Once the gain has been adjusted by the GCB, the PGAG bit will be set.

The automatic Gain Control can be disabled by clearing the AGEN bit. In this case, writing the IGAIN[1:0] register will allow manual gain control.

Note

The IGAIN[1:0] register content does determine the offset compensation register access, as there are 4 individual offset register buffers implemented, accessed through the same COC[7:0] register (paged register).

6.3.3.3.2 Voltage sense multiplexer (voltage channel only)

A multiplexer is implemented to select between nine different inputs: VSENSE3, VSENSE2, VSENSE1, VSENSE0, PTB4, PTB3, PTB2, PTB1, and PTB0. The multiplexer is controlled by the GPIO_VSENSE register.

Note

There is no further state machine separation of the various voltage channels. The software has to assure all compensation registers are configured properly after changing the multiplexer. Both voltage source conversion results will be stored in the same result register.

The divided and multiplexed voltages will be routed through the optional chopper (see IIR / decimation / chopping stage) before entering the Sigma Delta converter stage.

6.3.3.3.3 Sigma delta converter

A high resolution ADC is needed for current and battery voltage measurements of the MM9Z1_638. A second order sigma delta modulator based architecture is chosen.

6.3.3.3.4 Compensation

Following the optional chopper stage, the sigma delta bit stream is first gain and then offset compensated using the compensation registers.

The compensation stages for both channels can be completely bypassed by clearing the CCOMP / VCOMP bits.

6.3.3.3.5 IIR / decimation / chopping stage

6.3.3.3.5.1 Functional description

The chopper frequency is set to one fourth of the decimator frequency (512 kHz typ). On each phase, four decimation cycles are necessary to get a steady signal.

The equation of the IIR is $y_{n+1}=\alpha.x_n+(1-\alpha).y_n$.

The α parameter can be configured by the IIRC[2:0] register. See I and V chopper control register (ACQ_CVCR).

The decimation process is then completed by a programmable (DEC[2:0]) sinc3 filter, which outputs a 0.5...8 kS/s signal. The modulated noise is removed by an averaging filter (SINC1; L=4), which has an infinite rejection at the chopping frequency.

6.3.3.3.5.2 Latency and throughput (sampling rate)

Due to the nature of the digital (post-) processing of the sigma-delta converter, a latency has to be considered until the output signal/result is valid. The latency depends on the sigma-delta channel configuration and is handled in hardware if the conversion is started (setting of VMEN(M) respective CMEN(M) bits in the ACQ_CTL register). If the input signal changes (i.e. VSENSE_GPIO register) while the conversion is running, the latency has to be handled in software (by ignoring samples). Once the output signal/result is valid, all following data samples are ready with the sampling rate (throughput).

- The throughput (sampling rate) is 512 kHz/DF with DF configurable from 64 to 1024.
- The latency is given by (4+3*IIR+3*Avger+N_LPF)*DF/512 kHz where:
  - IIR=1 if IIR is enabled (0 otherwise),
  - Avger=1 as the chopper mode filter of current sense channel is always activated (independant of the chopper mode of the voltage channel),
  - N_LPF is the LPF coefficient number.

6.3.3.3.6 Low pass filter

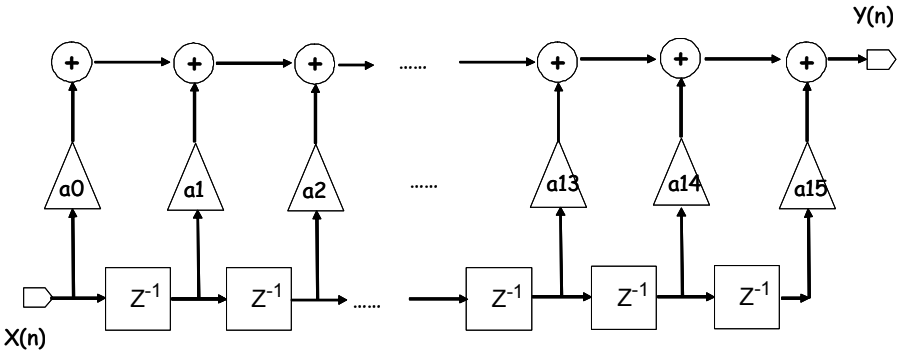
To achieve the required attenuation of the measured voltage and current signals in the frequency domain, a programmable low-pass filter following the SINC3+IIR filter, is implemented for both channels with shared configuration registers to deliver the equivalent filtering.

The following filter characteristic is implemented:

- F_{PASS} = 100 Hz (Att100 Hz)
- F_{STOPP} = 500 Hz (Att500 Hz)

The number of filter coefficients used can be programmed in the ACQ_LPFC[3:0] register. The filter can be bypassed completely clearing the LPFEN bit.

The filter uses an algorithmic and logic unit (ALU) for calculating the filtered output data, depending on the incoming data stream at “DATA IN” and the low-pass coefficients (A0...15) at the input “COEFF”, 16-bit width of each coefficient (See Low pass filter coefficient Ax (LPF_Ax (hi))). The filter structure calculates during one cycle (Tcyc=1/Fadc) the filtered data output.



y(n) = a0.x(n)+a1.x(n-1)+a2.x(n-2)+a3.x(n-3)+a4.x(n-4)+a5.x(n-5)+a6.x(n-6)+a7.x(n-7)
+a8.x(n-8)+a9.x(n-9)+a10.x(n-10)+a11.x(n-11)+a12.x(n-12)+a13.x(n-13)+a14.x(n-14)+a15.x(n-15)

Figure 48. FIR Structure

Z^{-1} Unit delay is done at a programmable frequency, depending on the decimation factor programmed in the DEC[2:0] register. See Table 163.

Note

There is no decimation from SINC3 to the LPF output, LPF uses same output rate than decimator. It's therefore possible to select an output update rate independent of the filter characteristic and bandwidth.

The coefficient vector consists of 16*16-bit elements and is free programmable, the maximum response time for 16 coefficients structure is 16*1/output rate. The following filter function can be realized.

$$H_{LP}(z) = \sum_{i=0}^{M} a_i * z^{-i}$$

LP filter function ***Eqn. 1***

The coefficients a_i are the elements of the coefficient vector and determine the filter function. M <= 15. It's possible to realize FIR filter functions. A typical total frequency response of the decimator and the programmable LP filter is given in Figure 49.

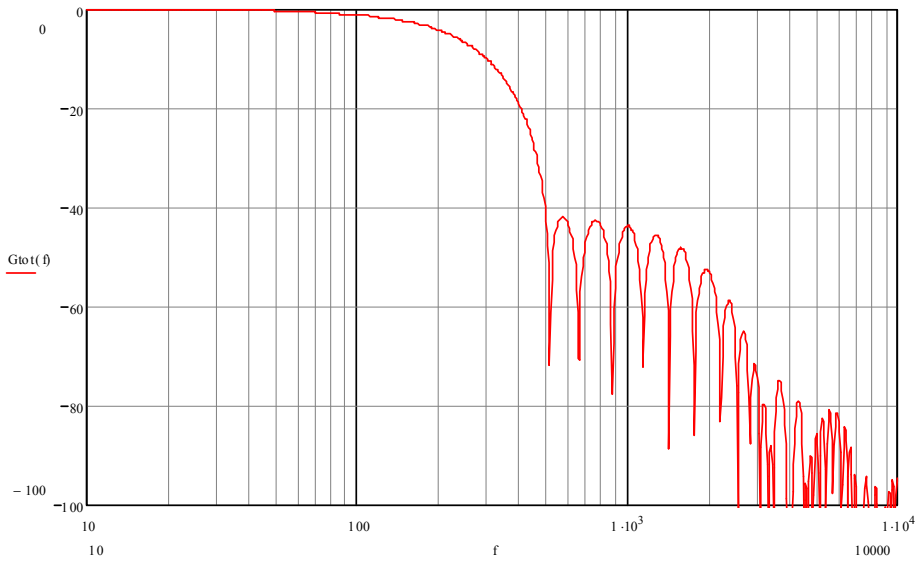


Figure 49. Typical total filter response Sinc3 (D = 128), LP filter (FIR type with 15 coefficients used)

6.3.3.3.7 Format and clamping

The output data stream is formatted into its final size for both channels (16-Bit for Voltage and 24-Bit for Current).

The current result will contain the gain information as part of the result. See Current measurement result (ACQ_CURR1 / ACQ_CURR0) and Voltage measurement result (ACQ_VOLT). Both results are written into the corresponding result registers and will issue an IRQ if enabled.

The internal voltage measurement results (no compensation active) are clamped to maximum and minimum values of 0xFFFF and 0x0000 respectively. Terminal voltages outside this range will result in the respective max or min. clamped values.

The internal current measurement results (no compensation active) are clamped to maximum and minimum values of 0x0FFFF and 0x10000 respectively. Terminal voltages outside this range will result in the respective max or min. clamped values.

Note

Both channels will perform synchronized conversions when enabled with a single write to the ACQ_CTL register.

As the voltage channel is not active during Low-power mode, the synchronicity might not be given after wake-up, and has to be re-established by restarting both channels.

Entering Low-power mode with the current / temperature channel enabled will have the channel(s) remain active during Low-power mode.

6.3.3.4 Temperature measurement channel

The MM9Z1_638 can measure the temperature from an internal built-in temperature sensor, or from an external temperature sensor connected to the PTB0, PTB1, PTB2, PTB3, or PTB4 pins. The external temperature sensor is supplied via the VDDA pin. The measurement channel is the same for the internal and external temperature sensor.

The temperature measurement channel uses the same Sigma Delta (SD) converter implementation as the current and voltage channel, followed by a fixed decimation (L=128). A selectable Chopper mode is implemented to compensate for offset errors. Once the chopper is enabled, an average (sinc1, L=2) is active.

Once the measurement is enabled, the temperature result registers are updated with the channel update rate. When both measurements are enabled, both temperature sensors are measured successively where the measurement is started with the internal sensor.

The internal temperature measurement result (no compensation active) of 0x0000 represents 0K, the maximum 0xFFFF = 523 k (typ). The result data is stored into the result registers ACQ_ITEMP and ACQ_ETEMP (both 16-bit).

During an over range event, the ADC is limited to the maximum value. The result of the internal temperature measurement is utilized to generate the calibration request (if ATGCE=0) or to perform auto temperature gain compensation (if ATGCE=1). See Calibration.

6.3.3.4.1 Compensation

The compensation for the temperature channels is implemented similar to the current and voltage channel.

Note

Factory trimmed compensation values are only available for the internal temperature channel.

6.3.3.5 Calibration

To ensure the maximum precision of the current and voltage sense module, several stages of calibration are implemented to compensate temperature effects. The calibration concept combines the availability of FLASH and the temperature information to guarantee the measurement accuracy under all functional conditions.

Several device parameters are guaranteed with full precision after system calibration only. During final test of the device, trim values are computed, verified, and stored into the system FLASH memory.

To ensure optimum system performance, the procedure has to be performed during power on. As the device is typically constantly powered during its operation, this operation has to be performed typically one time only. During a system power loss or low power reset condition, the application software has to ensure the procedure executes again.

Table 144. Acquisition channel compensation values

Channel Input		offset compensation					gain compensation				
		unit	LSB	default value	default code	format	unit	LSB	default value	default code	format
voltage sense	VSENSE3	mV	3.998	0	0x00	sint8	1	0.000488	0.9922	0x200	10-bit with offset
	VSENSE2	mV	2.000	0	0x00	sint8	1	0.000488	1.0679	0x200	10-bit with offset
	VSENSE1	mV	1.000	0	0x00	sint8	1	0.000488	1.2207	0x200	10-bit with offset
	VSENSE0	mV	1.000	0	0x00	sint8	1	0.000488	0.7632	0x200	10-bit with offset
	PTB[4..0]	mV	0.100	0	0x00	sint8	1	0.000488	0.7632	0x200	10-bit with offset
current sense	gain 4	mV	0.025602	0	0x00	sint8	1	0.000977	1.4902	0x200	10-bit with offset
	gain 16	mV	0.006400	0	0x00	sint8	1	0.000977	1.4902	0x200	10-bit with offset
	gain 64	mV	0.001600	0	0x00	sint8	1	0.000977	1.4902	0x200	10-bit with offset
	gain 256	mV	0.000400	0	0x00	sint8	1	0.000977	1.4902	0x200	10-bit with offset
temperature sense	ETS PTB[4..0]	mV	1.000	0	0x00	sint8	1	0.000977	1	0x80	8-bit with offset
	ITS	K	0.06384	0	0x00	sint8	1	0.000977	1	0x80	8-bit with offset

6.3.3.5.1 Startup trimming

To ensure all analog die modules are being trimmed properly, the following FLASH information (located in the MCU IFR from 0x0_1FC0C0 to 0x0_1FC0C5) has to be copied to the analog die register 0xE0 to 0xE5. (include related Offset of 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.)

Table 145. IFR content: trim and compensation values

Block	Addr (hex)	Register name	7	6	5	4	3	2	1	0
Trim values	1FC0C0	TRIM_BG0 (hi)	IBG1[2:1]			TCIBG2			TCIBG1	
	1FC0C1	TRIM_BG0 (lo)	IBG1[0]	LVT		TCBG2			TCBG1	
	1FC0C2	TRIM_BG1 (hi)	V1P2BG2				V1P2BG1			
	1FC0C3	TRIM_BG1 (lo)	VDDXLPMODE				V2P5BG1			
	1FC0C4	TRIM_OSC (hi)	SLPBG			LPOSC[12:8]				
	1FC0C5	TRIM_OSC (lo)	LPOSC[7:0]							
	1FC0C6	COMP_IG4 (hi)	0	0	0	0	0	0	IGC4	
	1FC0C7	COMP_IG4 (lo)	IGC4							
	1FC0C8	COMP_TIG4 (hi)	Reserved	0	0	TIGC4P				
	1FC0C9	COMP_TIG4 (lo)	Reserved	0	0	TIGC4N				
	1FC0CA	COMP_IG16 (hi)	0	0	0	0	0	0	IGC16	
	1FC0CB	COMP_IG16 (lo)	IGC16							
	1FC0CC	COMP_TIG16 (hi)	Reserved	0	0	TIGC16P				
	1FC0CD	COMP_TIG16 (lo)	Reserved	0	0	TIGC16N				
	1FC0CE	COMP_IG64 (hi)	0	0	0	0	0	0	IGC64	
	1FC0CF	COMP_IG64 (lo)	IGC64							
	1FC0D0	COMP_TIG64 (hi)	Reserved	0	0	TIGC64P				
	1FC0D1	COMP_TIG64 (lo)	Reserved	0	0	TIGC64N				
	1FC0D2	COMP_IG256 (hi)	0	0	0	0	0	0	IG256	
	1FC0D3	COMP_IG256 (lo)	IGC256							
	1FC0D4	COMP_TIG256 (hi)	Reserved	0	0	TIGC256P				
	1FC0D5	COMP_TIG256 (lo)	Reserved	0	0	TIGC256N				
	1FC0D6	COMP_VO_VSENSE0	VOC_VSENSE0							
	1FC0D7	COMP_VO_VSENSE1	VOC_VSENSE1							
	1FC0D8	COMP_VO_VSENSE2	VOC_VSENSE2							
	1FC0D9	COMP_VO_VSENSE3	VOC_VSENSE3							
	1FC0DA	COMP_VO_VSENSE_EXT	VOC_VSENSE_EXT							
	1FC0DB	COMP_VSG_VSENSE0 (hi)	0	0	0	0	0	0	VSGC_VSENSE0	
	1FC0DC	COMP_VSG_VSENSE0 (lo)	VSGC_VSENSE0							
	1FC0DD	COMP_TVSG_VSENSE0 (hi)	TVSOCP_VSENSE0[4]	0	0	TVSGCP_VSENSE0				
	1FC0DE	COMP_TVSG_VSENSE0 (lo)	TVSOCN_VSENSE0[4]	0	0	TVSGCN_VSENSE0				

Table 145. IFR content: trim and compensation values (continued)

Block	Addr (hex)	Register name	7	6	5	4	3	2	1	0
Trim values	1FC0DF	COMP_VSG_VSENSE1 (hi)	0	0	0	0	0	0	VSGC_VSENSE1	
	1FC0E0	COMP_VSG_VSENSE1 (lo)	VSGC_VSENSE1							
	1FC0E1	COMP_TVSG_VSENSE1 (hi)	TVSOCP_VSENSE1[4]	0	0	TVSGCP_VSENSE1				
	1FC0E2	COMP_TVSG_VSENSE1 (lo)	TVSOCN_VSENSE1[4]	0	0	TVSGCN_VSENSE1				
	1FC0E3	COMP_VSG_VSENSE2 (hi)	0	0	0	0	0	0	VSGC_VSENSE2	
	1FC0E4	COMP_VSG_VSENSE2 (lo)	VSGC_VSENSE2							
	1FC0E5	COMP_TVSG_VSENSE2 (hi)	TVSOCP_VSENSE2[4]	0	0	TVSGCP_VSENSE2				
	1FC0E6	COMP_TVSG_VSENSE2 (lo)	TVSOCN_VSENSE2[4]	0	0	TVSGCN_VSENSE2				
	1FC0E7	COMP_VSG_VSENSE3 (hi)	0	0	0	0	0	0	VSGC_VSENSE3	
	1FC0E8	COMP_VSG_VSENSE3 (lo)	VSGC_VSENSE3							
	1FC0E9	COMP_TVSG_VSENSE3 (hi)	TVSOCP_VSENSE3[4]	0	0	TVSGCP_VSENSE3				
	1FC0EA	COMP_TVSG_VSENSE3 (lo)	TVSOCN_VSENSE3[4]	0	0	TVSGCN_VSENSE3				
	1FC0EB	COMP_VSG_VSENSE_EXT (hi)	0	0	0	0	0	0	VSGC_VSENSE_EXT	
	1FC0EC	COMP_VSG_VSENSE_EXT (lo)	VSGC_VSENSE_EXT							
	1FC0ED	COMP_TVSG_VSENSE_EXT (hi)	TVSOCP_VSENSE_EXT[4]	0	0	TVSGCP_VSENSE_EXT				
	1FC0EE	COMP_TVSG_VSENSE_EXT (lo)	TVSOCN_VSENSE_EXT[4]	0	0	TVSGCN_VSENSE_EXT				
	1FC0EF	COMP_ITO	ITOC							
	1FC0F0	COMP_ITG	ITGC							
	1FC0F1	COMP_ETG_ROOM	ETGC_ROOM							
	1FC0F2	COMP_ETG_HOT	ETGC_HOT							
	1FC0F3	COMP_ETG_COLD	ETGC_COLD							
	1FC0F4	DIAG_IG4_ROOM (hi)	DIAG_IG4							
	1FC0F5	DIAG_IG4_ROOM (med)								
	1FC0F6	DIAG_IG4_ROOM (lo)								
	1FC0F7	DIAG_VSENSE_ROOM (hi)	DIAG_VSENSE							
	1FC0F8	DIAG_VSENSE_ROOM (lo)								
	1FC0F9	DIAG_TSENSE_ROOM (hi)	DIAG_TSENSE							
	1FC0FA	DIAG_TSENSE_ROOM (lo)								
	1FC0FB	COMP_TVSO_VSENSE0	TVSOCP_VSENSE0[3..0]				TVSOCN_VSENSE0[3…0]			
	1FC0FC	COMP_TVSO_VSENSE1	TVSOCP_VSENSE1[3..0]				TVSOCN_VSENSE1[3…0]			
	1FC0FD	COMP_TVSO_VSENSE2	TVSOCP_VSENSE2[3..0]				TVSOCN_VSENSE2[3…0]			
	1FC0FE	COMP_TVSO_VSENSE3	TVSOCP_VSENSE3[3..0]				TVSOCN_VSENSE3[3…0]			
	1FC0FF	COMP_TVSO_VSENSE_EXT	TVSOCP_VSENSE_EXT[3…0]				TVSOCN_VSENSE_EXT[3…0]			

Table 146. IFR content: trim and compensation values - register field descriptions

Field	Description
TRIM_OSC	Low Power Oscillator Trim repository
TRIM_BG[1…0]	BandGaps [1…0] (Voltage References) Trim repository
LVT	This bit enables the Cranking Pulse function availability (this bit is part of the register TRIM_BG1)
COMP_IG4	Current Channel Gain 4 Gain Compensation Code at Room Temperature (25 °C)
COMP_TIG4P	Current Channel Gain 4 Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TIG4N	Current Channel Gain 4 Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_IG16	Current Channel Gain 16 Gain Compensation Code at Room Temperature (25 °C)
COMP_TIG16P	Current Channel Gain 16 Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TIG16N	Current Channel Gain 16 Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_IG64	Current Channel Gain 64 Gain Compensation Code at Room Temperature (25 °C)
COMP_TIG64P	Current Channel Gain 64 Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TIG64N	Current Channel Gain 64 Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_IG256	Current Channel Gain 256 Gain Compensation Code at Room Temperature (25 °C)
COMP_TIG256P	Current Channel Gain 256 Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TIG256N	Current Channel Gain 256 Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_VO_VSENSE0	VSENSE0 Offset Compensation Code at Room Temperature (25 °C)
COMP_VO_VSENSE1	VSENSE1 Offset Compensation Code at Room Temperature (25 °C)
COMP_VO_VSENSE2	VSENSE2 Offset Compensation Code at Room Temperature (25 °C)
COMP_VO_VSENSE3	VSENSE3 Offset Compensation Code at Room Temperature (25 °C)
COMP_VO_VSENSE_EXT	VSENSE_EXT (PTB[4…0] as Voltage Channel ADC Input) Offset Compensation Code at Room Temperature (25 °C)
COMP_VSG_VSENSE0	VSENSE0 Gain Compensation Code at Room Temperature (25 °C)
COMP_TVSGCP_VSENSE0	VSENSE0 Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TVSGCN_VSENSE0	VSENSE0 Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_VSG_VSENSE1	VSENSE1 Gain Compensation Code at Room Temperature (25 °C)
COMP_TVSGCP_VSENSE1	VSENSE1 Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TVSGCN_VSENSE1	VSENSE1 Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_VSG_VSENSE2	VSENSE2 Gain Compensation Code at Room Temperature (25 °C)
COMP_TVSGCP_VSENSE2	VSENSE2 Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TVSGCN_VSENSE2	VSENSE2 Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_VSG_VSENSE3	VSENSE3 Gain Compensation Code at Room Temperature (25 °C)
COMP_TVSGCP_VSENSE3	VSENSE3 Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TVSGCN_VSENSE3	VSENSE3 Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_VSG_VSENSE_EXT	VSENSE_EXT Gain Compensation Code at Room Temperature (25 °C)
COMP_TVSGCP_VSENSE_EXT	VSENSE_EXT Gain Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
COMP_TVSGCN_VSENSE_EXT	VSENSE_EXT Gain Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
COMP_ITO	Internal Temperature Offset Compensation Code
COMP_ITG	Internal Temperature Gain Compensation Code
COMP_ETG_ROOM	External Temperature (PTB[4…0] as Temperature Channel ADC Input) Gain Compensation Code at Room Temperature (25 °C)
COMP_ETG_HOT	External Temperature Gain Compensation Code at Hot Temperature (125 °C)
COMP_ETG_COLD	External Temperature Gain Compensation Code at Cold Temperature (-40 °C)
DIAG_IG4_ROOM	Current Sense Channel Diagnostics Value for gain 4 at Room Temperature (25 °C)
DIAG_VSENSE_ROOM	Voltage Sense Channel Diagnostics Value at Room Temperature (25 °C)
DIAG_TSENSE_ROOM	Temperature Sense Channel Diagnostics Value at Room Temperature (25 °C)

Table 146. IFR content: trim and compensation values - register field descriptions

Field	Description
TVSOCP_VSENSE0[4…0]	VSENSE0 Offset Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
TVSOCN_VSENSE0[4…0]	VSENSE0 Offset Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
TVSOCP_VSENSE1[4…0]	VSENSE1 Offset Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
TVSOCN_VSENSE1[4…0]	VSENSE1 Offset Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
TVSOCP_VSENSE2[4…0]	VSENSE2 Offset Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
TVSOCN_VSENSE2[4…0]	VSENSE2 Offset Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
TVSOCP_VSENSE3[4…0]	VSENSE3 Offset Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
TVSOCN_VSENSE3[4…0]	VSENSE3 Offset Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement
TVSOCP_VSENSE_EXT[4…0]	VSENSE_EXT Offset Compensation Delta Code at Hot Temperature (125 °C) - 2's Complement
TVSOCN_VSENSE_EXT[4…0]	VSENSE_EXT Offset Compensation Delta Code at Cold Temperature (-40 °C) - 2's Complement

6.3.3.5.2 PGA auto zero sequence

The following procedure has to be performed for the PGA (Programmable Gain Amplifier) Auto Zero (AZ).

1. Write a “1” to the PGAO bit and its mask in the COMP_CTL register (0xA0)
2. Approximately 6.5 ms later, PGAOF will become set at to “1” (Flag needs to be polled)
3. Exit the PGAO mode by writing “0” in PGA0 and its mask being a “1”
4. Clear the PGAOF flag by writing “1”

Note

The new offset compensation data can be observed in the (PGAOC4...256[10:0]) registers.

The sequence will require 3352 clock cycles of the D2DFCLK (512kHz), typically 6.5 ms.

6.3.3.6 Memory map and registers

6.3.3.6.1 Overview

This section provides a detailed description of the memory map and registers.

6.3.3.6.2 Module memory map

The memory map for the Acquisition, Compensation, and LPF module is given in Table 60.

Table 147. Module memory map

Offset (133)(134)	Name		7	6	5	4	3	2	1	0
0x4B	GPIO_VSENSE	R	VSSEL		0	VSE4	VSE3	VSE2	VSE1	VSE0
	GPIO V_{SENSE} configuration	W								
0x4C	GPIO_TSENSE	R	0	0	0	TSE4	TSE3	TSE2	TSE1	TSE0
	GPIO T_{SENSE} configuration	W								
0x58	ACQ_CTL (hi)	R	0	0	0	0	0	0	0	0
	Acquisition control register	W	AHCRM		NVSEM	CVMIEM	ETMENM	ITMENM	VMENM	CMENM
0x59	ACQ_CTL (lo)	R	0		NVSE	CVMIE	ETMEN	ITMEN	VMEN	CMEN
	Acquisition control register	W	AHCR							
0x5A	ACQ_SR (hi)	R	AVRF	PGAG	VMOW	CMOW	ETM	ITM	VM	CM
	Acquisition status register	W	Write 1 will clear the flags							
0x5B	ACQ_SR (lo)	R	0	0	0	VTH	ETCHOP	ITCHOP	VCHOP	CCHOP
	Acquisition status register	W								

Table 147. Module memory map (continued)

Offset (133)(134)	Name		7	6	5	4	3	2	1	0
0x5C	ACQ_ACC1 (hi)	R	0	0	0	0	0	0	0	0
	Acquisition chain control	W	TCOMPM	VCOMPM	CCOMPM	LPFENM	ETCHOPM	ITCHOPM	CVCHOPM	AGENM
	ACQ_ACC1 (lo)	R	TCOMP	VCOMP	CCOMP	LPFEN	ETCHOP	ITCHOP	CVCHOP	AGEN
	Acquisition chain control	W								
0x5E	ACQ_ACC0 (hi)	R	0	0	0	0	0	0	0	0
	Acquisition chain control	W	ZEROM							
	ACQ_ACC0 (lo)	R	ZERO							
	Acquisition chain control	W								
0x60	ACQ_DEC	R	0	0	0	0	0	DEC[2:0]		
	Decimation rate	W								
0x61	ACQ_BGC	R	0	0	0	0	0	0	0	0
	BandGap control	W								
0x62	ACQ_GAIN	R	G256DIS	G64DIS	G16DIS	G4DIS	LPGEN	IGAIN[1:0]		0
	PGA gain	W								
	ACQ_GCB	R	D							
	GCB threshold	W								
0x64	ACQ_ITEMP (hi)	R	ITEMP[15:8]							
	Internal temperature measurement	W								
	ACQ_ITEMP (lo)	R	ITEMP[7:0]							
	Internal temperature measurement	W								
0x66	ACQ_ETEMP (hi)	R	ETEMP[15:8]							
	External temperature measurement	W								
	ACQ_ETEMP (lo)	R	ETEMP[7:0]							
	External temperature measurement	W								
0x68		R	0	0	0	0	0	0	0	0
		W								
	ACQ_CURR1	R	CURR[23:16]							
	Current measurement	W								
0x6A	ACQ_CURR0 (hi)	R	CURR[15:8]							
	Voltage measurement	W								
	ACQ_CURR0 (lo)	R	CURR[7:0]							
	Voltage measurement	W								
0x6C	ACQ_VOLT (hi)	R	VOLT[15:8]							
	Voltage measurement	W								
	ACQ_VOLT (lo)	R	VOLT[7:0]							
	Voltage measurement	W								
0x6E	ACQ_LPFC	R	0	0	0	0	LPFC			
	Low pass filter coefficient number	W								
		R	0	0	0	0	0	0	0	0
		W								

MM9Z1_638

Table 147. Module memory map (continued)

Offset (133)(134)	Name		7	6	5	4	3	2	1	0
0x70	ACQ_TCMP	R	TCMP							
	Low power trigger current measurement period	W								
		R								
		W								
0x72	ACQ_THF	R	THF							
	Low power current threshold filtering period	W								
		R	0	0	0	0	0	0	0	0
		W								
0x74	ACQ_CVCR (hi)	R	0	0	0	0	0	0	0	0
	Current and voltage chopper control register	W					IIRCM			
	ACQ_CVCR (lo)	R	0	0	0	0	IIRC			0
	Current and voltage chopper control register	W								
0x76	ACQ_CTH	R	CTH							
	Low power current threshold	W								
		R	0	0	0	0	0	0	0	0
		W								
0x78	ACQ_AHTH1 (hi)	R	0	AHTH[30:0]						
	Low power Ah counter threshold	W								
0x79	ACQ_AHTH1 (lo)	R								
	Low power Ah counter threshold	W								
0x7A	ACQ_AHTH0 (hi)	R								
	Low power Ah counter threshold	W								
0x7B	ACQ_AHTH0 (lo)	R								
	Low power Ah counter threshold	W								
0x7C	ACQ_AHC1 (hi)	R	AHC[31:0]							
	Low power Ah counter	W								
0x7D	ACQ_AHC1 (lo)	R	AHC[23:16]							
	Low power Ah counter	W								
0x7E	ACQ_AHC0 (hi)	R	AHC[15:8]							
	Low power Ah counter	W								
0x7F	ACQ_AHC0 (lo)	R	AHC[7:0]							
	Low power Ah counter	W								
0x80	LPF_A0 (hi)	R	A0[15:0]							
	A0 filter coefficient	W								
	LPF_A0 (lo)	R								
	A0 filter coefficient	W								
0x82	LPF_A1 (hi)	R	A1[15:0]							
	A1 filter coefficient	W								
	LPF_A1 (lo)	R								
	A1 filter coefficient	W								

Table 147. Module memory map (continued)

Offset (133)(134)	Name		7	6	5	4	3	2	1	0
0x84	LPF_A2 (hi)	R	A2[15:0]							
	A2 filter coefficient	W								
	LPF_A2 (lo)	R								
	A2 filter coefficient	W								
0x86	LPF_A3 (hi)	R	A3[15:0]							
	A3 filter coefficient	W								
	LPF_A3 (lo)	R								
	A3 filter coefficient	W								
0x88	LPF_A4 (hi)	R	A4[15:0]							
	A4 filter coefficient	W								
	LPF_A4 (lo)	R								
	A4 filter coefficient	W								
0x8A	LPF_A5 (hi)	R	A5[15:0]							
	A5 filter coefficient	W								
	LPF_A5 (lo)	R								
	A5 filter coefficient	W								
0x8C	LPF_A6 (hi)	R	A6[15:0]							
	A6 filter coefficient	W								
	LPF_A6 (lo)	R								
	A6 filter coefficient	W								
0x8E	LPF_A7 (hi)	R	A7[15:0]							
	A7 filter coefficient	W								
	LPF_A7 (lo)	R								
	A7 filter coefficient	W								
0x90	LPF_A8 (hi)	R	A8[15:0]							
	A8 filter coefficient	W								
	LPF_A8 (lo)	R								
	A8 filter coefficient	W								
0x92	LPF_A9 (hi)	R	A9[15:0]							
	A9 filter coefficient	W								
	LPF_A9 (lo)	R								
	A9 filter coefficient	W								
0x94	LPF_A10 (hi)	R	A10[15:0]							
	A10 filter coefficient	W								
	LPF_A10 (lo)	R								
	A10 filter coefficient	W								
0x96	LPF_A11 (hi)	R	A11[15:0]							
	A11 filter coefficient	W								
	LPF_A11 (lo)	R								
	A11 filter coefficient	W								
0x98	LPF_A12 (hi)	R	A12[15:0]							
	A12 filter coefficient	W								
	LPF_A12 (lo)	R								
	A12 filter coefficient	W								

Table 147. Module memory map (continued)

Offset (133)(134)	Name		7	6	5	4	3	2	1	0
0x9A	LPF_A13 (hi)	R	A13[15:0]							
	A13 filter coefficient	W								
	LPF_A13 (lo)	R								
	A13 filter coefficient	W								
0x9C	LPF_A14 (hi)	R	A14[15:0]							
	A14 filter coefficient	W								
	LPF_A14 (lo)	R								
	A14 filter coefficient	W								
0x9E	LPF_A15 (hi)	R	A15[15:0]							
	A15 filter coefficient	W								
	LPF_A15 (lo)	R								
	A15 filter coefficient	W								
0xA0	COMP_CTL (hi)	R	0	0	0	0	0	0	0	0
	Compensation control register	W	OPENEM		PGAZM	PGAOM	DIAGVM	DIAGIM	DIAGTM	CALIEM
	COMP_CTL (lo)	R	OPENE		PGAZ	PGAO	DIAGV	DIAGI	DIAGT	CALIE
	Compensation control register	W								
0xA2	COMP_SR	R	OPEN	BGRF	0	PGAOF	0	0	0	CALF
	Compensation status register	W	Write 1 will clear the flags							
0xA3	COMP_TF	R	0	IRSEL			ATGCE	TMF		
	Temperature filtering period	W								
0xA4	COMP_TMAX (hi)	R	TCMAX							
	Max. temp before recalibration	W								
	COMP_TMAX (lo)	R	0	0	0	0	0	0	0	0
	Max. temp before recalibration	W								
0xA6	COMP_TMIN (hi)	R	TCMIN							
	Min. temp before recalibration	W								
	COMP_TMIN (lo)	R	0	0	0	0	0	0	0	0
	Min. temp before recalibration	W								
0xA8		R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xAA	COMP_VO	R	VOC							
	Offset voltage compensation	W								
0xAB	COMP_IO	R	COC							
	Offset current compensation	W								
0xAC	COMP_VSG (hi)	R	0	0	0	0	0	0	VSGC	
	Gain voltage compensation vsense channel	W								
	COMP_VSG (lo)	R	VSGC							
	Gain voltage compensation vsense channel	W								

Table 147. Module memory map (continued)

Offset (133)(134)	Name		7	6	5	4	3	2	1	0
0xAE	COMP_TVSG (hi)	R	0	0	0	TVSGCP				
	Voltage gain temp compensation above 25 °C	W								
	COMP_TVSG (lo)	R	0	0	0	TVSGCN				
	Voltage gain temp compensation below 25 °C	W								
0xB0	COMP_IG4 (hi)	R	0	0	0	0	0	0	IGC4	
	Gain current compensation	W								
	COMP_IG4 (lo)	R	IGC4							
	Gain current compensation	W								
0xB2	COMP_TIG4 (hi)	R	0	0	0	TIGC4P				
	Gain current compensation above 25°C	W								
	COMP_TIG4 (lo)	R	0	0	0	TIGC4N				
	Gain current compensation below 25°C	W								
0xB4	COMP_IG16 (hi)	R	0	0	0	0	0	0	IGC16	
	Gain current compensation	W								
	COMP_IG16 (lo)	R	IGC16							
	Gain current compensation	W								
0xB6	COMP_TIG16 (hi)	R	0	0	0	TIGC16P				
	Gain current compensation above 25 °C	W								
	COMP_TIG16 (lo)	R	0	0	0	TIGC16N				
	Gain current compensation above 25 °C	W								
0xB8	COMP_IG64 (hi)	R	0	0	0	0	0	0	IGC64	
	Gain current compensation	W								
	COMP_IG64 (lo)	R	IGC64							
	Gain current compensation	W								
0xBA	COMP_TIG64 (hi)	R	0	0	0	TIGC64P				
	Gain current compensation above 25°C	W								
	COMP_TIG64 (lo)	R	0	0	0	TIGC64N				
	Gain current compensation above 25°C	W								
0xBC	COMP_IG256 (hi)	R	0	0	0	0	0	0	IGC256	
	Gain current compensation	W								
	COMP_IG256 (lo)	R	IGC256							
	Gain current compensation	W								
0xBE	COMP_TIG256 (hi)	R	0	0	0	TIGC256P				
	Current gain temp compensation above 25°C	W								
	COMP_TIG256 (lo)	R	0	0	0	TIGC256N				
	Current gain temp compensation above 25°C	W								

Table 147. Module memory map (continued)

Offset (133)(134)	Name		7	6	5	4	3	2	1	0
0xC0	COMP_PGAO4 (hi)	R	0	0	0	0	0	PGAOC4		
	Offset PGA compensation	W								
	PGAO4 (lo)	R	PGAOC4							
	Offset PGA compensation	W								
0xC2		R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xC4	COMP_PGAO16 (hi)	R	0	0	0	0	0	PGAOC16		
	Offset PGA compensation	W								
	COMP_PGAO16 (lo)	R	PGAOC16							
	Offset PGA compensation	W								
0xC6		R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xC8	COMP_PGAO64 (hi)	R	0	0	0	0	0	PGAOC64		
	Offset PGA compensation	W								
	COMP_PGAO64 (lo)	R	PGAOC64							
	Offset PGA compensation	W								
0xCA	Reserved	R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xCC	COMP_PGAO256 (hi)	R	0	0	0	0	0	PGAOC256		
	Offset PGA compensation	W								
	COMP_PGAO256 (lo)	R	PGAOC256							
	Offset PGA compensation	W								
0xCE		R	0	0	0	0	0	0	0	0
		W								
		R	0	0	0	0	0	0	0	0
		W								
0xD0	COMP_ITO	R	ITOC							
	Internal temp. offset compensation	W								
0xD1	COMP_ITG	R	ITGC							
	Internal temp. gain compensation	W								
0xD2	COMP_ETO	R	ETOC							
	External temp. offset compensation	W								
0xD3	COMP_ETG	R	ETGC							
	External temp. gain compensation	W								
0xD4		R	0	0	0	0	0	0	0	0
		W								

Table 147. Module memory map (continued)

Offset (133)(134)	Name		7	6	5	4	3	2	1	0
0xD5		R	0	0	0	0	0	0	0	0
		W								
0xD6		R	0	0	0	0	0	0	0	0
		W								
0xD7		R	0	0	0	0	0	0	0	0
		W								

Notes:
133.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
134.Register Offset with the “lo” address value not shown have to be accessed in 16Bit mode. 8 Bit access will not function.

6.3.3.6.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

6.3.3.6.3.1 GPIO VSENSE configuration register (GPIO_VSENSE)

Table 148. GPIO VSENSE configuration register (GPIO_VSENSE)

Offset (135)	0x4B, 0x4C						Access: User read/write	
	7	6	5	4	3	2	1	0
R	VSSEL		0	VSE4	VSE3	VSE2	VSE1	VSE0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
135.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 149. GPIO VSENSE configuration register (GPIO_VSENSE) - register field descriptions

Field	Description
7-6 VSSEL	**Vsense selection** 00 - selection of Vsense0 01 - selection of Vsense1 10 - selection of Vsense2 11 - selection of Vsense3
4 VSE4	**PTB4 selection for voltage measurement** 0 - PTB4 not selected 1- PTB4 selected
3 VSE3	**PTB3 selection for voltage measurement** 0 - PTB3 not selected 1- PTB3 selected
2 VSE2	**PTB2 selection for voltage measurement** 0 - PTB2 not selected 1- PTB2 selected
1 VSE1	**PTB1 selection for voltage measurement** 0 - PTB1 not selected 1- PTB1 selected
0 VSE0	**PTB0 selection for voltage measurement** 0 - PTB0 not selected 1- PTB0 selected

Note

The selection between V_{SENSE} and PTBx voltage sense is done as follows:

Measurement via VSENSE[3:0] pin:

VSE4 and VSE3 and VSE2 and VSE1 and VSE0 = 0. VSENSE[3:0] selection according to VSSEL[1:0].

Measurement via PTB[4:0] pin:

VSSEL[1:0] = [0:0], PTB[4:0] selection according to VSE4, VSE3, VSE2, VSE1, VSE0 setting. In this case only 1 bit (1 PTB channel) must be selected.

6.3.3.6.3.2 GPIO TSENSE configuration register (GPIO_TSENSE)

Table 150. GPIO TSENSE configuration register (GPIO_TSENSE)

Offset (136)	0x4B, 0x4C					Access: User read/write		
	7	6	5	4	3	2	1	0
R	0	0	0	TSE4	TSE3	TSE2	TSE1	TSE0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
136.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 151. GPIO TSENSE configuration register (GPIO_TSENSE) - register field descriptions

Field	Description
4 TSE4	**PTB4 selection for temperature measurement** 0 - PTB4 not selected 1- PTB4 selected
3 TSE3	**PTB3 selection for temperature measurement** 0 - PTB3 not selected 1- PTB3 selected
2 TSE2	**PTB2 selection for temperature measurement** 0 - PTB2 not selected 1- PTB2 selected
1 TSE1	**PTB1 selection for temperature measurement** 0 - PTB1 not selected 1- PTB1 selected
0 TSE0	**PTB0 selection for temperature measurement** 0 - PTB0 not selected 1- PTB0 selected

Note

Only 1 of the following bits (TSE4, TSE3, TSE2, TSE1, TSE0) must be selected.

6.3.3.6.3.3 ACQ_CTL acquisition control register (ACQ_CTL)

Table 152. Acquisition control register (ACQ_CTL)

Offset (137),(138)	0x58					Access: User read/write		
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R	0	0	0	0	0	0	0	0
W	AHCRM		NVSEM	CVMIEM	ETMENM	ITMENM	VMENM	CMENM
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	0	0	NVSE	CVMIE	ETMEN	ITMEN	VMEN	CMEN
W	AHCR							
Reset	0	0	0	0	0	0	0	0

Notes:
137.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
138.This Register is 16-Bit access only.

Table 153. Acquisition control register (ACQ_CTL) - register field descriptions

Field	Description
15 AHCRM	**Ampere Hour Counter Reset** - Mask 0 - writing the AHCR Bit will have no effect 1 - writing the AHCR Bit will be effective
13 NVSEM	**Negative Voltage Enable** - Mask 0 - NVSE bit will have no effect 1 - NVSE bit will be effective
12 CVMIEM	**Current / Voltage Measurement Interrupt Enable** - Mask 0 - writing the CVMIE Bit will have no effect 1 - writing the CVMIE Bit will be effective
11 ETMENM	**External Temperature Measurement Enable** - Mask 0 - writing the ETMEN Bit will have no effect 1 - writing the ETMEN Bit will be effective
10 ITMENM	**Internal Temperature Measurement Enable** - Mask 0 - writing the ITMEN Bit will have no effect 1 - writing the ITMEN Bit will be effective
9 VMENM	**Voltage Measurement Sense Enable** - Mask 0 - writing the VMEN Bit will have no effect 1 - writing the VMEN Bit will be effective
8 CMENM	**Current Measurement Enable** - Mask 0 - writing the CMEN Bit will have no effect 1 - writing the CMEN Bit will be effective
7 AHCR	**Ampere Hour Counter Reset**, this write only bit will reset the ACQ_AHC register. 0 - no effect 1 - ACQ_AHC reset to 0x00000000
5 NVSE	**Negative Voltage Sense Enable on selected PTB4...0**, on the Voltage Acquisition Channel 0 - Negative Voltage Measurement Sense disabled 1 - Negative Voltage Measurement Sense enabled
4 CVMIE	**Current / Voltage Measurement Interrupt Enable** 0 - current and voltage measurement interrupt disabled 1 - current and voltage measurement interrupt enabled
3 ETMEN	**External Temperature Measurement Enable** 0 - external temperature measurement disabled 1 - external temperature measurement enabled

Table 153. Acquisition control register (ACQ_CTL) - register field descriptions (continued)

Field	Description
2 ITMEN	**Internal Temperature Measurement Enable** 0 - internal temperature measurement disabled 1 - internal temperature measurement enabled
1 VMEN	**Voltage Measurement Enable** 0 - voltage measurement disabled 1 - voltage measurement enabled
0 CMEN	**Current Measurement Enable** 0 - current measurement disabled 1 - current measurement enabled

6.3.3.6.3.4 Acquisition status register (ACQ_SR (hi))

Table 154. Acquisition status register (ACQ_SR (hi))

Offset[139] **0x5A** **Access: User read/write**

	7	6	5	4	3	2	1	0
R	AVRF	PGAG	VMOW	CMOW	ETM	ITM	VM	CM
W	Write 1 will clear the flags							
Reset	0	0	0	0	0	0	0	0

Notes:
139.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 155. Acquisition status register (ACQ_SR (hi)) - register field descriptions

Field	Description
7 AVRF	**VDDA Low Voltage Reset Flag**. Writing this bit to logic 1 will clear the flag. 0 - n.a. 1 - Last reset was caused by a low voltage condition at the VDDA regulator.
6 PGAG	**PGA Gain Change Flag**[140]. Writing this bit to logic 1 will clear the flag. 0 - PGA gain has not changed since last flag clear 1 - PGA gain has changed since last flag clear
5 VMOW	**Voltage Measurement Result Overwritten**[140]. Writing this bit to logic 1 will clear the flag. 0 - Voltage measurement result register VOLT[15:0] not overwritten[141] since last VMOW flag clear 1 - Voltage measurement result register VOLT[15:0] overwritten[141] since last VMOW flag clear
4 CMOW	**Current Measurement Result Overwritten**[140]. Writing this bit to logic 1 will clear the flag. 0 - Current measurement result register CURR[15:0] not overwritten[141] since last CMOW flag clear 1 - Current measurement result register CURR[15:0] overwritten[141] since last CMOW flag clear
3 ETM	**End of Measurement - External Temperature**[140]. Writing this bit to logic 1 will clear the flag. 0 - No external temperature measurement completed since last ETM clear 1 - External temperature measurement completed since last ETM clear
2 ITM	**End of Measurement - Internal Temperature**[140]. Writing this bit to logic 1 will clear the flag. 0 - No internal temperature measurement completed since last ITM clear 1 - Internal temperature measurement completed since last ITM clear
1 VM	**End of Measurement** - Voltage. Writing this bit to logic 1 will clear the flag. 0 - No voltage measurement completed since last VM clear 1 - Voltage measurement completed since last VM clear
0 CM	**End of Measurement** - Current. Writing this bit to logic 1 will clear the flag. 0 - No current measurement completed since last CM clear 1 - Current measurement completed since last CM clear

Notes:
140.No Interrupts issued for those flags
141.Overwritten - new result latched before previous result was read

6.3.3.6.3.5 Acquisition status register (ACQ_SR (lo))

Table 156. Acquisition status register (ACQ_SR (lo))

Offset[142]	0x5B						Access: User read	
	7	6	5	4	3	2	1	0
R	0	0	0	VTH	ETCHOP	ITCHOP	VCHOP	CCHOP
W								

Notes:
142.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 157. Acquisition status register (ACQ_SR (lo)) - register field descriptions

Field	Description
7	
4 VTH	**Voltage High Threshold Reached for selected channel and voltage acquisition on going** 0: maximum functional operating voltage according to Table 6 not reached. 1: maximum functional operating voltage according toTable 6 reached.
3 ETCHOP	**Chopping Active Status** - External Temperature 0 - Chopper for external temperature measurement disabled 1 - Chopper for external temperature measurement enabled
2 ITCHOP	**Chopping Active Status** - Internal Temperature 0 - Chopper for internal temperature measurement disabled 1 - Chopper for internal temperature measurement enabled
1 VCHOP	**Chopping Active Status** - Voltage 0 - Chopper for voltage measurement disabled 1 - Chopper for voltage measurement enabled
0 CCHOP	**Chopping Active Status** - Current 0 - Chopper for current measurement disabled 1 - Chopper for current measurement enabled

6.3.3.6.3.6 Acquisition chain control 1 (ACQ_ACC1)

Table 158. Acquisition chain control 1 (ACQ_ACC1)

Offset (143)(144)	0x5C						Access: User read/write	
	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	TCOMPM	VCOMPM	CCOMPM	LPFENM	ETCHOPM	ITCHOPM	CVCHOPM	AGENM
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	TCOMP	VCOMP	CCOMP	LPFEN	ETCHOP	ITCHOP	CVCHOP	AGEN
Reset	1	1	1	0	0	0	0	1

Notes:
143.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
144.This Register is 16 Bit access only.

Table 159. Acquisition chain control 1 (ACQ_ACC1) - register field descriptions

Field	Description
15 TCOMPM	**Temperature Measurement Channel** - Compensation Enable - Mask 0 - writing the TCOMP bit will have no effect 1 - writing the TCOMP bit will be effective
14 VCOMPM	**Voltage Measurement Channel** - Compensation Enable - Mask 0 - writing the VCOMP bit will have no effect 1 - writing the VCOMP bit will be effective
13 CCOMPM	**Current Measurement Channel** - Compensation Enable - Mask 0 - writing the CCOMP bit will have no effect 1 - writing the CCOMP bit will be effective
12 LPFENM	**LPF Enable** - Mask 0 - writing the LPFEN bit will have no effect 1 - writing the LPFEN bit will be effective
11 ETCHOPM	**Chopping Enable** - External Temperature Measurement Channel - Mask 0 - writing the ETCHOP bit will have no effect 1 - writing the ETCHOP bit will be effective
10 ITCHOPM	**Chopping Enable** - Internal Temperature Measurement Channel - Mask 0 - writing the ITCHOP bit will have no effect 1 - writing the ITCHOP bit will be effective
9 CVCHOPM	**Chopping Enable** - Voltage Measurement Channel - Mask 0 - writing the CVCHOP bit will have no effect 1 - writing the CVCHOP bit will be effective
8 AGENM	**Automatic Gain Control Enable** - Mask 0 - writing the AGEN bit will have no effect 1 - writing the AGEN bit will be effective
7 TCOMP	**Temperature Measurement Channel** - Compensation Enable 0 - Temperature measurement channel offset and gain compensation disabled 1 - Temperature measurement channel offset and gain compensation enabled
6 VCOMP	**Voltage Compensation Enable** 0 - Voltage measurement channel offset and gain compensation disabled 1 - Voltage measurement channel offset and gain compensation enabled
5 CCOMP	**Current Compensation Enable** 0 - Current measurement channel offset and gain compensation disabled 1 - Current measurement channel offset and gain compensation enabled
4 LPFEN	**LPF Enable** 0 - Low pass filter for current and voltage channel disabled 1 - Low pass filter for current and voltage channel enabled
3 ETCHOP	**Chopping Enable** - External Temperature 0 - Chopper mode for external temperature measurement disabled 1 - Chopper mode for external temperature measurement enabled
2 ITCHOP	**Chopping Enable** - Internal Temperature 0 - Chopper mode for internal temperature measurement disabled 1 - Chopper mode for internal temperature measurement enabled
1 CVCHOP	**Chopping Enable** - Voltage 0 - Chopper mode for voltage measurement disabled 1 - Chopper mode for voltage measurement enabled
0 AGEN	**Automatic Gain Control Enable** 0 - Automatic gain control disabled (manual gain control via IGAIN[1:0]) 1 - Automatic gain control enabled

6.3.3.6.3.7 Acquisition chain control 0 (ACQ_ACC0)

Table 160. Acquisition chain control 0 (ACQ_ACC0)

Offset (145),(146)	0x5E					Access: User read/write		
	15	14	13	12	11	10	9	8
R	0							
W	ZEROM							
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	ZERO							
W								
Reset	0	0	0	0	0	0	0	0

Notes:
145.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
146.This Register is 16 Bit access only.

Table 161. Acquisition chain control 0 (ACQ_ACC0) - register field descriptions

Field	Description
15 ZEROM	**Current and Voltage Sigma Delta Input Short** - Mask 0 - writing the ZERO bit will have no effect 1 - writing the ZERO bit will be effective
7 ZERO	**Current and Voltage Sigma Delta Input Short enable** 0 - short of the Current and Voltage Sigma Delta Input disabled 1 - short of the Current and Voltage Sigma Delta Input enabled

6.3.3.6.3.8 Decimation rate (ACQ_DEC)

Table 162. Decimation rate (ACQ_DEC)

Offset(147)	0x60					Access: User read/write		
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	DEC[2:0]		
W								
Reset	0	0	0	0	0	1	0	0

Notes:
147.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 163. Decimation rate (ACQ_DEC) - register field descriptions

Field	Description
2-0 DEC[2:0]	**Decimation Rate Selection** (Combined decimation rate of first and second sinc3 decimator; First decimator is fixed to D=8) 000 - D = 512 (Channel Output Rate = 1.0 kHz) 001 - D = 64 (Channel Output Rate = 8.0 kHz) 010 - D = 128 (Channel Output Rate = 4.0 kHz) 011 - D = 256 (Channel Output Rate = 2.0 kHz) 100 - D = 512 (Channel Output Rate = 1.0 kHz), (default) 101 - D = 1024 (Channel Output Rate = 500 Hz) 110 - D = 512 (Channel Output Rate = 1.0 kHz) 111 - D = 512 (Channel Output Rate = 1.0 kHz)

6.3.3.6.3.9 PGA gain (ACQ_GAIN)

Table 164. PGA gain (ACQ_GAIN)

Offset[148]	0x62					Access: User read/write		
	7	6	5	4	3	2	1	0
R	G256DIS	G64DIS	G16DIS	G4DIS	LPGEN	IGAIN[1:0]		
W								
Reset	0	0	0	0	0	0	0	0

Notes:
148.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 165. PGA gain (ACQ_GAIN) - register field descriptions

Field	Description
7 G256DIS	**Disable usage of Gain 256** (available gains must be consecutive) 0 - Gain usage is enable 1 - Gain usage is disable
6 G64DIS	**Disable usage of Gain 64** (available gains must be consecutive) 0 - Gain usage is enable 1 - Gain usage is disable
5 G16DIS	**Disable usage of Gain 16** (available gains must be consecutive) 0 - Gain usage is enable 1 - Gain usage is disable
4 G4DIS	**Disable usage of Gain 4** (available gains must be consecutive) 0 - Gain usage is enable 1 - Gain usage is disable
3 LPGEN	**Enable usage of gain in Low-power mode, otherwise gain 256 is selected** 0 - Gain 256 is used in Low-power mode 1 - Gain selected by bits 7,6,5,4 is used in Low-power mode
2-1 IGAIN[1:0]	**PGA Gain.** 00 - PGA Gain = 4 01 - PGA Gain = 16 10 - PGA Gain = 64 11 - PGA Gain = 256

6.3.3.6.3.10 GCB threshold (ACQ_GCB)

Table 166. GCB threshold (ACQ_GCB)

Offset[149]	0x63					Access: User read/write		
	7	6	5	4	3	2	1	0
R	D (hi)				D (lo)			
W								
Reset	0	0	0	0	0	0	0	0

Notes:
149.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 167. GCB threshold (ACQ_GCB) - register field descriptions

Field	Description
7-4 D[7:4]	**Gain Control Block (GCB)** - 4 Bit Gain “Up” Threshold. See Gain control block (GCB).
3-0 D[3:0]	**Gain Control Block (GCB)** - 4 Bit Gain “Down” Threshold. See Gain control block (GCB).

6.3.3.6.3.11 Internal temp. measurement result (ACQ_ITEMP (hi) / ACQ_ITEMP (lo))

Table 168. Internal temp. measurement result (ACQ_ITEMP (hi) / ACQ_ITEMP (lo))

Offset[150]	0x64 / 0x65							Access: User read
	7	6	5	4	3	2	1	0
R	ITEMP[15:8]							
W								
R	ITEMP[7:0]							
W								

Notes:
150.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 169. Internal temp. measurement result (ACQ_ITEMP (hi) / ACQ_ITEMP (lo)) - register field descriptions

Field	Description
15-0 ITEMP[15:0]	**Internal Temperature Measurement** - 16 Bit ADC Result Register (unsigned Integer)

6.3.3.6.3.12 External temp. measurement result (ACQ_ETEMP (hi) / ACQ_ETEMP (lo))

Table 170. External temp. measurement result (ACQ_ETEMP (hi) / ACQ_ETEMP (lo))

Offset[151]	0x66 / 0x67							Access: User read
	7	6	5	4	3	2	1	0
R	ETEMP[15:8]							
W								
R	ETEMP[7:0]							
W								

Notes:
151.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 171. External temp. measurement result (ACQ_ETEMP (hi) / ACQ_ETEMP (lo)) - register field descriptions

Field	Description
15-0 ETEMP[15:0]	**External Temperature Measurement** - 16 Bit ADC Result Register (unsigned Integer)

6.3.3.6.3.13 Current measurement result (ACQ_CURR1 / ACQ_CURR0)

Table 172. Current measurement result (ACQ_CURR1 / ACQ_CURR0)

Offset[152]	0x69[153] / 0x6A[154]	Access: User read

	7	6	5	4	3	2	1	0
R	CURR1[23:16]							
W								
R	CURR0[15:8]							
W								
R	CURR0[7:0]							
W								

Notes:
152.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
153.0x69 for 8-Bit access. 0x68 for 16-Bit access.
154.This Register is 16-Bit access only.

Table 173. Current measurement result (ACQ_CURR1 / ACQ_CURR0) - register field descriptions

Field	Description
CURR[23:0]	**Two's complement 24** - Bit signed integer result register for the current measurement channel.
23-16 CURR1[23:16]	**Current Measurement** - High Byte Result Register, 8 or 16-Bit read operation. A read of the ACQ_CURR1 (ACQ_CURR0) register latches the ACQ_CURR0 (ACQ_CURR1) register. Therefore the access is atomic.
15-0 CURR0[15:0]	**Current Measurement** - Low Word Result Register, 16-Bit read operation only. A read of the ACQ_CURR0 (ACQ_CURR1) register latches the ACQ_CURR1 (ACQ_CURR0) register. Therefore the access is atomic.

6.3.3.6.3.14 Voltage measurement result (ACQ_VOLT)

Table 174. Voltage measurement result (ACQ_VOLT)

Offset (155)(156)	0x6C	Access: User read

	7	6	5	4	3	2	1	0
R	VOLT[15:8]							
W								
R	VOLT[7:0]							
W								

Notes:
155.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
156.This Register is 16-Bit access only.

Table 175. Voltage measurement result (ACQ_VOLT) - register field descriptions

Field	Description
15-0 VOLT[15:0]	**Unsigned 16** - Bit integer result register for the voltage measurement channel.

6.3.3.6.3.15 Low pass filter coefficient number (ACQ_LPFC)

Table 176. Low pass filter coefficient number (ACQ_LPFC)

Offset[157]	0x6E						Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	0	0	0	LPFC[3:0]			
W								
Reset	0	0	0	0	1	1	1	0

Notes:
157.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 177. Low pass filter coefficient number (ACQ_LPFC) - register field descriptions

Field	Description
3-0 LPFC[3:0]	**Low Pass Filter Coefficient Number.** Defines the highest coefficient Number used. 0000 - LPF used with Coefficient A0 0001 - LPF used with Coefficient A0…A1 … 1111 - LPF used with Coefficient A0…A15

6.3.3.6.3.16 Low power trigger current measurement period (ACQ_TCMP)

Table 178. Low power trigger current measurement period (ACQ_TCMP)

Offset (158)(159)	0x70						Access: User read / write	
	7	6	5	4	3	2	1	0
R	TCMP[15:8]							
W								
Reset	0	0	0	0	0	0	0	0
R	TCMP[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes:
158.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
159.This Register is 16-Bit access only.

Table 179. Low power trigger current measurement period (ACQ_TCMP) - register field descriptions

Field	Description
15-0 TCMP[15:0]	**Low power trigger current measurement period** (Trigger counter based on ALFCLK). See Cyclic current acquisition / calibration temperature check.

Note

The cyclic acquisition period must be greater than the acquisition time. See Latency and throughput (sampling rate)" for estimation. A continuous acquisition is still possible by using TCMP=0.

The Low Power Trigger Current counter is an up counting counter starting at 0. It increments according to the Low Power Clock.

In Low Power mode, when the Low Power Trigger Current counter is equal to the Low Power Trigger Currrent Measurement Period, the device will start a current and temperature acquisition, according to the setting of the current and internal temperature acquisition channels.

6.3.3.6.3.17 Low power current threshold filtering period (ACQ_THF)

Table 180. Low power current threshold filtering period (ACQ_THF)

Offset[160]	0x72						Access: User read / write	
	7	6	5	4	3	2	1	0
R	THF[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes:
160.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 181. Low power current threshold filtering period (ACQ_THF) - register field descriptions

Field	Description
7-0 THF[7:0]	**Low power current threshold wake-up filtering period.** See Cyclic current acquisition / calibration temperature check

6.3.3.6.3.18 I and V chopper control register (ACQ_CVCR)

Table 182. I and V chopper control register (ACQ_CVCR)

Offset[161], [162]	0x74						Access: User write	
	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W					IIRCM			
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	0	0	0	0	IIRC			0
W								
Reset	0	0	0	0	1	1	1	0

Notes:
161.This register it 16-bit access only.
162.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 183. I and V chopper control register (ACQ_CVCR) - register field descriptions

Field	Description
11-9 IIRCM[2:0]	**IIR Low Pass Filter Configuration** - Mask 000, 001, 010, 011, 100, 101, 110 - writing the IIRC bits has no effect 111 - writing the IIRC bits are effective
3-1 IIRC[2:0]	**IIR Low Pass Filter Coefficient** (α) 000 - 1/8 001 - 1/16 010 - 1/32 011 - 1/64 100 - 1/128 101 - 1/1 (IIR disabled) 110 - 1/1 (IIR disabled) 111 - 1/1 (IIR disabled)

6.3.3.6.3.19 Low power current threshold (ACQ_CTH)

Table 184. Low power current threshold (ACQ_CTH

Offset[163]	0x76						Access: User read / write	
	7	6	5	4	3	2	1	0
R	CTH[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes:
163.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 185. Low power current threshold (ACQ_CTH - register field descriptions

Field	Description
7-0 CTH[7:0]	**Low power current threshold.** See **Current threshold wake-up** for details.

6.3.3.6.3.20 Low power Ah counter threshold (ACQ_AHTH1 (hi) / ACQ_AHTH1 (lo) / ACQ_AHTH0 (hi) / ACQ_AHTH0 (lo))

Table 186. Low power Ah counter threshold (ACQ_AHTH1 (hi) / ACQ_AHTH1 (lo) / ACQ_AHTH0 (hi) / ACQ_AHTH0 (lo))

Offset[164]	0x78 / 0x79 / 0x7A / 0x7B						Access: User read / write	
	7	6	5	4	3	2	1	0
R	AHTH[30:0]							
W								
R								
W								
R								
W								
R								
W								
Reset	0	0	0	0	0	0	0	0

Notes:
164.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 187. Low power Ah counter threshold (ACQ_AHTH1 (hi) / ACQ_AHTH1 (lo) / ACQ_AHTH0 (hi) / ACQ_AHTH0 (lo)) - register field descriptions

Field	Description
30-0 AHTH[30:0]	**Low power Ah counter threshold.** Absolute (unsigned) 31-Bit integer. Reading one 16-Bit part of the register will buffer the second. Reading the second will unlock the buffer. See Current ampere hour threshold wake-up for details on the Register.

6.3.3.6.3.21 Low power Ah Counter (ACQ_AHC1 (hi) / ACQ_AHC1 (lo) / ACQ_AHC0 (hi) / ACQ_AHC0 (lo))

Table 188. Low power Ah counter (ACQ_AHC1 (hi) / ACQ_AHC1 (lo) / ACQ_AHC0 (hi) / ACQ_AHC0 (lo))

Offset[165]	0x7C / 0x7D / 0x7E / 0x7F							Access: User read
	7	6	5	4	3	2	1	0
R	AHC[31:0]							
W								
R	AHC[23:16]							
W								
R	AHC[15:8]							
W								
R	AHC[7:0]							
W								

Notes:
165.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 189. Low power Ah counter (ACQ_AHC1 (hi) / ACQ_AHC1 (lo) / ACQ_AHC0 (hi) / ACQ_AHC0 (lo)) - register field descriptions

Field	Description
31-0 AHC[31:0]	Low power Ah counter (32-Bit signed integer, two's complement). Reading one 16-Bit part of the register will buffer the second. Reading the second will unlock the buffer. See Current ampere hour threshold wake-up.

6.3.3.6.3.22 Low pass filter coefficient Ax (LPF_Ax (hi))

Table 190. Low pass filter coefficient Ax (LPF_Ax (hi))

Offset[166]	0x80...0x9E							Access: User read/write
	7	6	5	4	3	2	1	0
R	Ax[15:8]							
W								
Reset	see Table 193							

Notes:
166.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

6.3.3.6.3.23 Low pass filter coefficient Ax (LPF_Ax (lo))

Table 191. Low pass filter coefficient Ax (LPF_Ax (lo))

Offset[167]	0x81...0x9F							Access: User read/write
	7	6	5	4	3	2	1	0
R	Ax[7:0]							
W								
Reset	see Table 193							

Notes:
167.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 192. Low pass filter coefficient Ax - register field descriptions

Field	Description
15-0 Ax[15:0]	**Low Pass Filter Coefficient Value.** x = 0...15. Data Format: MSB = Sign (“1” minus). [14:0] integer.

Table 193. Low pass filter coefficient Ax - reset values

Field	Reset Value	Field	Reset Value
A0	0x00B5	A8	0x1021
A1	0x0212	A9	0x0E35
A2	0x041B	A10	0x0A44
A3	0x0812	A11	0x0812
A4	0x0A44	A12	0x041B
A5	0x0E35	A13	0x0212
A6	0x1021	A14	0x00B5
A7	0x10E4	A15	0x0000

6.3.3.6.3.24 Compensation control register (COMP_CTL)

Table 194. Compensation control register (COMP_CTL)

Offset (168)(169)	0xA0 / 0xA1					Access: User read/write		
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R	0	0	0	0	0	0	0	0
W	OPENEM		PGAZM	PGAOM	DIAGVM	DIAGIM	DIAGTM	CALIEM
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	OPENE		PGAZ	PGAO	DIAGV	DIAGI	DIAGT	CALIE
W								
Reset	0	0	0	0	0	0	0	0

Notes:
168.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
169.This Register is 16-Bit access only.

Table 195. Compensation control register (COMP_CTL) - register field descriptions

Field	Description
15 OPENEM	**Enable Shunt Resistor Open Detection** - Mask 0 - writing the OPENE Bit will have no effect 1 - writing the OPENE Bit will be effective
13 PGAZM	**PGA Input Zero** - Mask 0 - writing the PGAZ bit will have no effect 1 - writing the PGAZ bit will be effective
12 PGAOM	**PGA Offset Calibration** - Mask 0 - writing the PGAO bit will have no effect 1 - writing the PGAO bit will be effective
11 DIAGVM	**Diagnostic Mode Voltage Channel** - Mask 0 - writing the DIAGV bit will have no effect 1 - writing the DIAGV bit will be effective

Table 195. Compensation control register (COMP_CTL) - register field descriptions (continued)

Field	Description
10 DIAGIM	**Diagnostic Mode Current Channel** - Mask 0 - writing the DIAGI bit will have no effect 1 - writing the DIAGI bit will be effective
9 DIAGTM	**Diagnostic Mode Temperature Channel** - Mask 0 - writing the DIAGTM bit will have no effect 1 - writing the DIAGTM bit will have allow measure of V and I reference by the Temperature ADC
8 CALIEM	**Calibration IRQ Enable** - Mask 0 - writing the CALIE bit will have no effect 1 - writing the CALIE bit will be effective
7 OPENE	**Enable Shunt Resistor Open Detection** 0 - Shunt resistor open detection disabled, the OPEN bit must be ignored 1 - Shunt resistor open detection enabled, OPEN bit will indicate status
5 PGAZ	**PGA Input Zero** 0 - Programmable gain amplifier inputs in normal operation 1 - Programmable gain amplifier inputs shorted for Calibration
4 PGAO	**PGA Offset Calibration Start** 0 - PGA normal operation 1 - PGA internal offset calibration start (PGAOF will indicate calibration complete). PGAZ has to be set to 1 during calibration. The bit will remain set after the calibration is complete. It has to be cleared by writing 0 before it can be set to start the next calibration. The current measurement channel has to be enabled (ACQ_CTL[CMEN]=1) in order to perform the PGA offset compensation.
3 DIAGV	**Diagnostic Mode Voltage Channel** 0 - Calibration reference disconnected from the voltage channel input 1 - Calibration reference connected to the voltage channel input for calibration. Manual conversion needed to measure reference
2 DIAGI	**Diagnostic Mode Current Channel** 0 - Calibration reference disconnected from the current channel input 1 - Calibration reference connected to the current channel input for calibration. Manual conversion needed to measure reference
1 DIAGT	**Diagnostic Mode Temperature Channel** 0 - Calibration reference disconnected from the temperature channel input 1 - Calibration reference connected to the temperature channel input for calibration. Manual conversion needed to measure reference
0 CALIE	**Calibration IRQ Enable** 0 - Calibration IRQ disabled 1 - Calibration IRQ Enabled

6.3.3.6.3.25 Compensation status register (COMP_SR)

Table 196. Compensation status register (COMP_SR)

Offset[170]	**0xA2**						**Access: User read/write**	
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	OPEN	BGRF	0	PGAOF	0	0	0	CALF
W	Write 1 will clear the flags							
Reset	0	0	0	0	0	0	0	0

Notes:
170.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 197. Compensation status register (COMP_SR) - register field descriptions

Field	Description
7 OPEN	**Shunt Resistor Open Detection Status** (Normal mode only, only function if OPENE = 1) 0 - Shunt resistor detected 1 - Shunt resistor disconnected
6 BGRF	**Band Gap Reference Status Flag** 0 - Indicates the reference bandgap has not been set / applied 1 - Reference bandgap has been set. Writing 1 will clear the flag
4 PGAOF	**PGA Internal Offset Compensation Complete Flag** 0 - PGA offset compensation ongoing or not started since last flag clear 1 - PGA offset compensation finished since last flag clear. Writing 1 will clear the flag
0 CALF	**Calibration Request Status Flag** 0 - No Temperature out of range condition detected 1 - Temperature out of range condition detected. Writing 1 clears the flag and starts the next calibration steps

6.3.3.6.3.26 Temperature filtering period (COMP_TF)

Table 198. Temperature filtering period (COMP_TF)

Offset[171]	**0xA3**							**Access: User read / write**
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	0	IRSEL[2…0]			ATGCE	TMF[2:0]		
W								
Reset	0	0	1	0	0	0	0	0

Notes:
171.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 199. Temperature filtering period (COMP_TF) - register field descriptions

Field	Description
6-4 IRSEL[2…0]	**Shunt Resistor Typ. Value selection** 000: Shunt = 50 μΩ 001 Shunt = 75 μΩ 010 Shunt = 100 μΩ (default) 011: Shunt = 150 μΩ 100: Shunt = 200 μΩ
3 ATGCE	**Automatic Gain Temperature Compensation** 0: Automatic Gain Temperature Compensation disabled 1: Automatic Gain Temperature Compensation enabled
2-0 TMF[2:0]	**Recalibration Temperature Filtering period.** Defines the number of measurements above / below the Max. / Min. thresholds that are required before a calibration request is detected.

Note

When ATGCE bit is set, the Calibration Request IRQ is disabled. The Intelligent Battery Sensor MM9Z1_638 will calibrate by its own Gain Compensation register.

6.3.3.6.3.27 Max. temp. before recalibration (COMP_TMAX)

Table 200. Max. temp. before recalibration (COMP_TMAX)

Offset	0xA4						Access: User read/write	
	7	6	5	4	3	2	1	0
R W	TCMAX[7:0]							
Reset	0	0	0	0	0	0	0	0

Table 201. Max. temp. before recalibration (COMP_TMAX) - register field descriptions

Field	Description
7-0 TCMAX[7:0]	**Maximum Temperature before recalibration**. Once the internal temperature measurement result is above or equal to TCMAX (high byte compared with ACQ_ITEMP(hi)), the TMF filter counter increases, if below, the counter decreases.

6.3.3.6.3.28 Min. temp. before recalibration (COMP_TMIN)

Table 202. Min. temp. before recalibration (COMP_TMIN)

Offset (172)(173)	0xA6						Access: User read/write	
	7	6	5	4	3	2	1	0
R W	TCMIN[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes:
172.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
173.This Register is 16-Bit access only.

Table 203. Min. temp. before recalibration (COMP_TMIN) - register field descriptions

Field	Description
7-0 TCMIN[7:0]	**Minimum Temperature before recalibration.** Once the internal temperature measurement result is below TCMIN (high byte compared with ACQ_ITEMP(hi)), the TMF filter counter increases, if above or equal, the counter decreases.

6.3.3.6.3.29 Offset voltage compensation (COMP_VO)

Table 204. Offset voltage compensation (COMP_VO)

Offset(174)	0xAA						Access: User read/write	
	7	6	5	4	3	2	1	0
R W	VOC[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes:
174.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 205. Offset voltage compensation (COMP_VO) - register field descriptions

Field	Description
7-0 VOC[7:0]	**Voltage Offset Compensation Buffer.** This register contains the voltage channel offset compensation as an 8-bit signed char (two complement). 0x7F = max, 0x80 = min.

6.3.3.6.3.30 Offset current compensation window (COMP_IO)

Table 206. Offset current compensation window (COMP_IO)

Offset[175]	0xAB						Access: User read/write	
	7	6	5	4	3	2	1	0
R W	COC[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes:
175.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 207. Offset current compensation window (COMP_IO) - register field descriptions

Field	Description
7-0 COC[7:0]	**Paged (4 pages/gains) Current Offset Compensation Buffer.** This register gives access to the 4 (one for each gain) current channel offset compensation values as 8-bit signed char (two complement). 0x7F = max., 0x80 = min. The content of the ACQ_GAIN register IGAIN[1:0] bits select the page (page/gain).

6.3.3.6.3.31 Gain voltage comp. V_{SENSE} channel (COMP_VSG)

Table 208. Gain voltage comp. V_{SENSE} channel (COMP_VSG)

Offset (176)(177)	0xAC						Access: User read/write	
	15	14	13	12	11	10	9	8
R W	0	0	0	0	0	0	VSGC[9:8]	
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	VSGC[7:0]							
Reset	0	0	0	0	0	0	0	0

Offset (176)(177)	0xAE / 0xAF						Access: User read/write	
	15	14	13	12	11	10	9	8
R W	0	0	0	TVSGCP				
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R W	0	0	0	TVSGCN				
Reset	0	0	0	0	0	0	0	0

Notes:
176.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
177.This Register is 16 Bit access only.

Table 209. Gain voltage comp. V_{SENSE} channel (COMP_VSG) - register field descriptions

Field	Description
9-0 VSGC[9:0]	**Voltage Channel Gain Compensation Buffer.** This register contains the voltage channel gain compensation as 10-bit special coded value. Refer to Compensation for details.
12-8 TVSGCP[4:0]	**V_{SENSE} Gain, temperature delta code for 125 °C**
12-8 TVSGCN[4:0]	**VSENSE Gain, temperature delta code for -40 °C**

6.3.3.6.3.32 4 x gain current compensation 4...256 (COMP_IG4... COMP_IG256)

Table 210. 4 x gain current compensation 4...256 (COMP_IG4... COMP_IG256)

Offset (178)(179) — **0xB0 and B1 / 0xB4 and B5 / 0xB8 and B9 / 0xBC and BD** — **Access: User read/write**

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	IGC4...256(hi) [9:8]	
W								
Reset	0	0	0	0	0	0	1	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	IGC4...256 (lo) [7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Offset (178)(179) — **0xB2 and B3 / 0xB6 and B7 / 0xBA and BB / 0xBE and BF** — **Access: User read/write**

	15	14	13	12	11	10	9	8
R	0	0	0	TIGC4…256P				
W								
Reset	0	0	0	0	0	0	1	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	0	0	0	TIGC4...256N				
W								
Reset	0	0	0	0	0	0	0	0

Notes:
178.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
179.This Register is 16 Bit access only.

Table 211. 4 x gain current compensation 4...256 (COMP_IG4... COMP_IG256) - register field descriptions

Field	Description
9-0 IGC4[9:0] IGC16[9:0] IGC64[9:0] IGC256[9:0]	**Individual Current Gain Compensation Buffers for the 4 Gain configurations.** Those registers contain the current channel gain compensation as 10-bit special coded value. Refer to Compensation for details.
12-8 TIGC4P[12:8] TIGC16P[12:8] TIGC64P[12:8] TIGC256P[12:8]	**Isense Gain 4 , 16 , 64 , 256 temperature delta code for 125 °C**
4-0 TIGC4N[4:0] TIGC16N[4:0] TIGC64N[4:0] TIGC256N[4:0]	**Isense Gain 4, 16, 64, 256 temperature delta code for -40 °C**

6.3.3.6.3.33 4 x offset PGA compensation (COMP_PGAO4...COMP_PGAO256)

Table 212. 8 x offset PGA compensation (COMP_PGAO4... COMP_PGAO256)

Offset (178)(179) — **0xC0 and C1 / 0xC4 and C5 / 0xC8 and C9 / 0xCC and CD** — **Access: User read/write**

	15	14	13	12	11	10	9	8
R	0	0	0	0	0	PGAOC4...256(hi) [10:8]		
W								
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	PGAOC4...256 (lo) [7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes:
180.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
181.This Register is 16 Bit access only.

Table 213. 8 x offset PGA compensation (COMP_PGAO4...COMP_PGAO256) - register field descriptions

Field	Description
10-0 PGAOC4[10:0] PGAOC16[10:0] PGAOC64[10:0] PGAOC256[10:0]	**Individual PGA Offset Compensation Buffers for the 4 Gain configurations.** Those registers contain the PGA Offset compensation as 11-bit special coded value. Refer to Compensation for details.

6.3.3.6.3.34 Internal temp. offset compensation (COMP_ITO)

Table 214. Internal temp. offset compensation (COMP_ITO)

Offset[182]	0xD0	Access: User read/write

	7	6	5	4	3	2	1	0
R W	ITOC[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes:
182.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 215. Internal temp. offset compensation (COMP_ITO) - register field descriptions

Field	Description
7-0 ITOC[7:0]	**Internal Temperature Offset Compensation Buffer.** This register contains the Internal Temperature Offset compensation as 8-bit signed char (two complement). Refer to Compensation for details.

6.3.3.6.3.35 Internal temp. gain compensation (COMP_ITG)

Table 216. Internal temp. gain compensation (COMP_ITG)

Offset[183]	0xD1	Access: User read/write

	7	6	5	4	3	2	1	0
R W	ITGC[7:0]							
Reset	1	0	0	0	0	0	0	0

Notes:
183.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 217. Internal temp. gain compensation (COMP_ITG) - register field descriptions

Field	Description
7-0 ITGC[7:0]	**Internal Temperature Gain Compensation Buffer.** This register contains the Internal Temperature Gain compensation as 8-bit special coded value. Refer to Compensation for details.

6.3.3.6.3.36 External temp. offset compensation (COMP_ETO)

Table 218. External temp. offset compensation (COMP_ETO)

Offset[184]	0xD2	Access: User read/write

	7	6	5	4	3	2	1	0
R W	ETOC[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes:
184.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 219. External temp. offset compensation (COMP_ETO) - register field descriptions

Field	Description
7-0 ETOC[7:0]	**External Temperature Offset Compensation Buffer.** This register contains the External Temperature Offset compensation as 8-bit signed char (two complement). Refer to Compensation for details.

6.3.3.6.3.37 External temp. gain compensation (COMP_ETG)

Table 220. External temp. gain compensation (COMP_ETG)

Offset[185]	0xD3						Access: User read/write	
	7	6	5	4	3	2	1	0
R W	ETGC[7:0]							
Reset	1	0	0	0	0	0	0	0

Notes:
185.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 221. External temp. gain compensation (COMP_ETG) - register field descriptions

Field	Description
7-0 ETGC[7:0]	**External Temperature Gain Compensation Buffer.** This register contains the External Temperature Gain compensation as 8-bit special coded value. Refer to Compensation for details.

6.4 General purpose I/O - GPIO

6.4.1 Introduction

The four general purpose pins (PTB1…4) are multipurpose ports, providing additional digital input/output capability.

6.4.2 Features

General purpose I/O PTB1, PTB2, and PTB3:

- 5.0 V (VDDX) digital input/output
- selectable pull-up
- selectable routing to LIN/SCI or timer module
- deactivated in low-power mode

High voltage wake-up input PTB4:

- 5.0 V (VDDX) digital input
- high voltage input with internal clamping structure (requires ext. serial resistor)
- selectable pull-down
- wake-up from low-power mode (high or low level)
- selectable routing to timer IC (Input capture)
- input filter

6.4.3 Block diagram

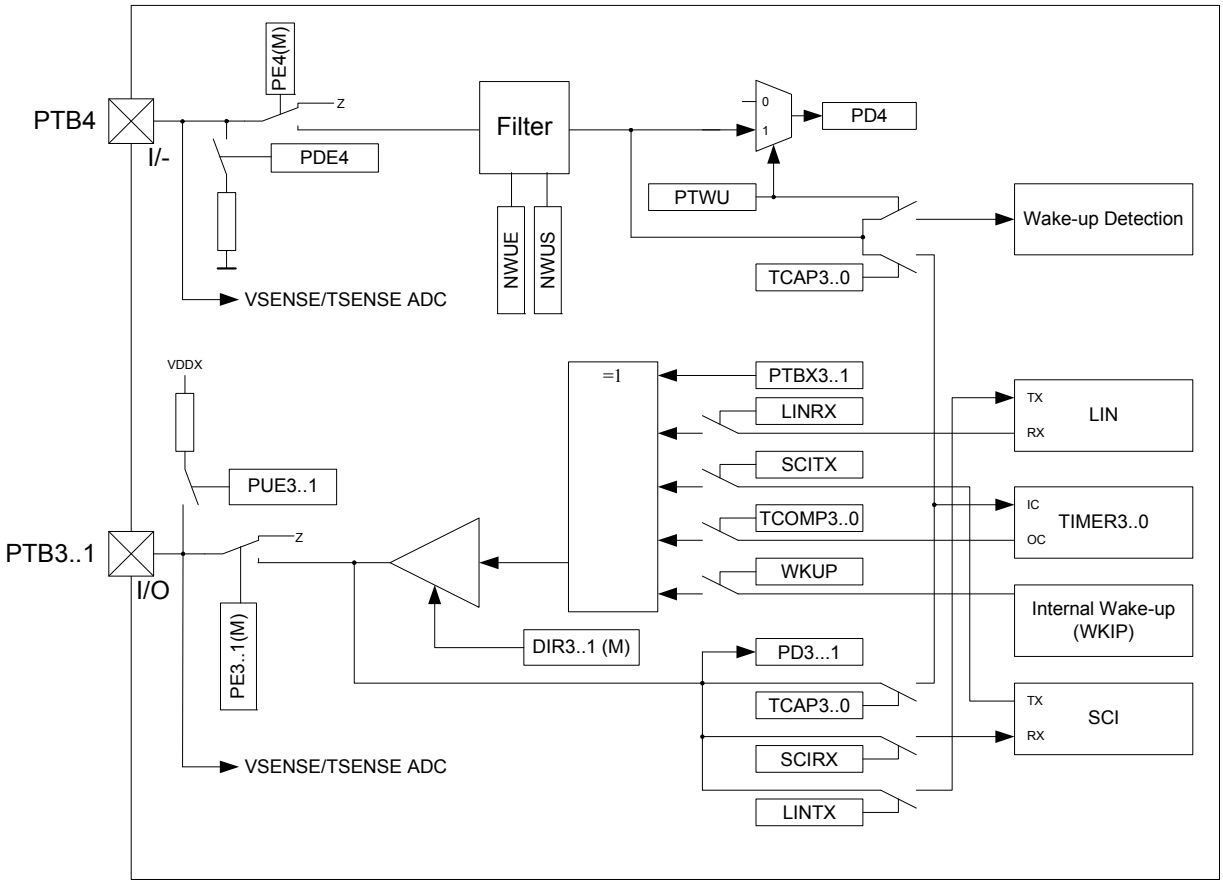


Figure 50. General purpose I/O - block diagram

6.4.3.1 High voltage wake-up input - PTB4

To offer robust high voltage wake-up capabilities, the following structure is implemented for PTB4.

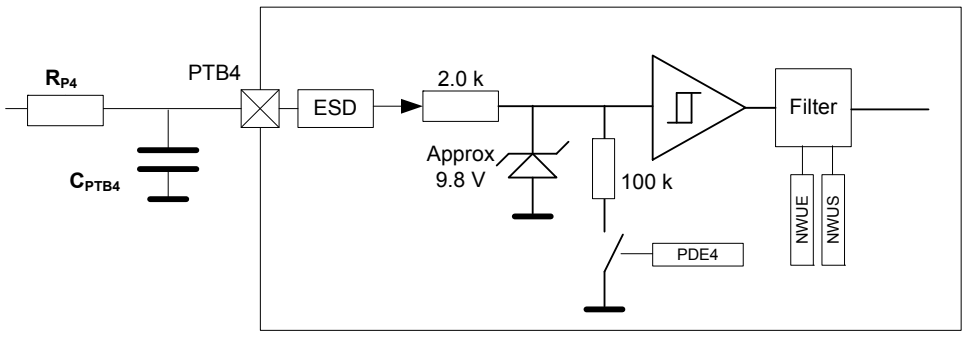


Figure 51. PTB4 input structure (typical values indicated)

Note

Due the different implementation of the PTB4, the PTWU bit needs to be set in the GPIO_IN4 register, to read the port status PD4 during Normal mode.

6.4.4 Modes of operation

The full GPIO functionality is only available during Normal mode. The only features available in both Low-power modes is the PTB4 external wake-up and the wake-up routing of the timer output compare.

Note

TCOMP3...1 needs to be configured to allow timer output compare (OC) to generate a system wake-up.

6.4.5 Memory map and registers

6.4.5.1 Overview

This section provides a detailed description of the memory map and registers.

6.4.5.2 Module Memory Map

The memory map for the GPIO module is given in Table 60

Table 222. Module memory map

Offset (186), (187)	Name		7	6	5	4	3	2	1	0
0x40	GPIO_CTL (hi)	R	0	0	0	0	0	0	0	0
	GPIO control register	W	DIR3M	DIR2M	DIR1M	PE4M	PE3M	PE2M	PE1M	
0x41	GPIO_CTL (lo)	R	DIR3	DIR2	DIR1	PE4	PE3	PE2	PE1	0
	GPIO control register	W								
0x42	GPIO_PUC	R	0	0	0	PDE4	PUE3	PUE2	PUE1	0
	GPIO pull-up/down configuration	W								
0x43	GPIO_DATA	R	0	0	0	PD4	PD3	PD2	PD1	0
	GPIO port data register	W								
0x44	GPIO_IN1	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 1input configuration	W								
0x45	GPIO_OUT1	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 1output configuration	W								PTBX1
0x46	GPIO_IN2	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 2 input configuration	W								
0x47	GPIO_OUT2	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 2 output configuration	W								PTBX2
0x48	GPIO_IN3	R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
	Port 3 input configuration	W								
0x49	GPIO_OUT3	R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
	Port 3 output configuration	W								PTBX3
0x4A	GPIO_IN4	R	PTWU	TCAP3	TCAP2	TCAP1	TCAP0	0	NWUS	NWUE
	Port 4 input configuration	W								
0x4B	GPIO_VSENSE	R	VSSEL		0	VSE4	VSE3	VSE2	VSE1	VSE0
	GPIO V_{SENSE} configuration	W								
0x4C	GPIO_TSENSE	R	0	0	0	TSE4	TSE3	TSE2	TSE1	TSE0
	GPIO T_{SENSE} configuration	W								
0x4D		R	0	0	0	0	0	0	0	0
		W								

MM9Z1_638

Table 222. Module memory map (continued)

Offset (186), (187)	Name		7	6	5	4	3	2	1	0
0x4E		R	0	0	0	0	0	0	0	0
		W								
0x4F		R	0	0	0	0	0	0	0	0
		W								

Notes:
186.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
187.Register Offset with the “lo” address value not shown have to be accessed in 16-Bit mode. 8-Bit access will not function.

6.4.5.3 Register descriptions

6.4.5.3.1 GPIO control register (GPIO_CTL)

Table 223. GPIO control register (GPIO_CTL)

Offset (188),(189)	0x40						Access: User read/write	
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R	0	0	0	0	0	0	0	0
W	DIR3M	DIR2M	DIR1M	PE4M	PE3M	PE2M	PE1M	0
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	DIR3	DIR2	DIR1	PE4	PE3	PE2	PE1	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
188.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
189.This Register is 16-Bit access only.

Table 224. GPIO control register (GPIO_CTL)

Field	Description
15 DIR3M	**Data Direction PTB3** - Mask 0 - writing the DIR3 bit will have no effect 1 - writing the DIR3 bit will be effective
14 DIR2M	**Data Direction PTB2** - Mask 0 - writing the DIR2 bit will have no effect 1 - writing the DIR2 bit will be effective
13 DIR1M	**Data Direction PTB1** - Mask 0 - writing the DIR1 bit will have no effect 1 - writing the DIR1 bit will be effective
12 PE4M	**Port 4 Enable** - Mask 0 - writing the PE4 bit will have no effect 1 - writing the PE4 bit will be effective
11 PE3M	**Port 3 Enable** - Mask 0 - writing the PE3 bit will have no effect 1 - writing the PE3 bit will be effective
10 PE2M	**Port 2 Enable** - Mask 0 - writing the PE2 bit will have no effect 1 - writing the PE2 bit will be effective

Table 224. GPIO control register (GPIO_CTL) (continued)

Field	Description
9 PE1M	**Port 1 Enable** - Mask 0 - writing the PE1 bit will have no effect 1 - writing the PE1 bit will be effective
8	
7 DIR3	**Data Direction PTB3** 0 - PTB3 configured as Input 1 - PTB3 configured as Output
6 DIR2	**Data Direction PTB2** 0 - PTB2 configured as Input 1 - PTB2 configured as Output
5 DIR1	**Data Direction PTB1** 0 - PTB1 configured as Input 1 - PTB1 configured as Output
4 PE4	**Port 4 Enable**(190) 0 - PTB4 Disabled (Z state) 1 - PTB4 Enabled (I)
3 PE3	**Port 3 Enable**(190) 0 - PTB3 Disabled (Z state) 1 - PTB3 Enabled (I/O)
2 PE2	**Port 2 Enable**(190) 0 - PTB2 disabled (Z state) 1 - PTB2 enabled (I/O)
1 PE1	**Port 1 Enable**(190) 0 - PTB1 disabled (Z state) 1 - PTB1 enabled (I/O)
0	

Notes:
190.The port logic is always enabled. Setting PEx will connect the logic to the port I/O buffers.

6.4.5.3.2 GPIO pull-up configuration (GPIO_PUC)

Table 225. GPIO pull-up configuration (GPIO_PUC)

Offset(191)			**0x42**				**Access: User read/write**	
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	0	0	0	PDE4	PUE3	PUE2	PUE1	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
191.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 226. GPIO pull-up configuration (GPIO_PUC)

Field	Description
4 PDE4	**PTB4 Pull-down Enable** 0 - PTB4 pull-down disabled 1 - PTB4 pull-down enabled
3 PUE3	**PTB3 Pull-up Enable** 0 - PTB3 pull-up disabled 1 - PTB3 pull-up enabled
2 PUE2	**PTB2 Pull-up Enable** 0 - PTB2 pull-up disabled 1 - PTB2 pull-up enabled

Table 226. GPIO pull-up configuration (GPIO_PUC) (continued)

Field	Description
1 PUE1	**PTB1 Pull-up Enable** 0 - PTB1 pull-up disabled 1 - PTB1 pull-up enabled
0	

6.4.5.3.3 GPIO port data register (GPIO_DATA)

Table 227. GPIO port data register (GPIO_DATA)

Offset[192]	0x43						Access: User read	
	7	6	5	4	3	2	1	0
R	0	0	0	PD4[193]	PD3	PD2	PD1	0
W								

Notes:
192.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
193.Due the different implementation of the PTB4, PTWU needs to be set in the GPIO_IN4 to read the PD4 port status during Normal mode.

Table 228. GPIO port data register (GPIO_DATA)

Field	Description
4 PD4	**PTB4 Data Register** A read returns the value of the PTB4 buffer.
3 PD3	**PTB3 Data Register** A read returns the value of the PTB3 buffer.
2 PD2	**PTB2 Data Register** A read returns the value of the PTB2 buffer.
1 PD1	**PTB1 Data Register** A read returns the value of the PTB1 buffer.

6.4.5.3.4 Port 1 input configuration (GPIO_IN1)

Table 229. Port 1 input configuration (GPIO_IN1)

Offset[194]	0x44						Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
194.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 230. Port 1 input configuration (GPIO_IN1)

Field	Description
6 TCAP3	**PTB1** - Timer Input Capture Channel 3 0 - PTB1 Input buffer disconnected from Timer Channel 3 - Input Capture 1 - PTB1 Input buffer routed to Timer Channel 3 - Input Capture
5 TCAP2	**PTB1** - Timer Input Capture Channel 2 0 - PTB1 Input buffer disconnected from Timer Channel 2 - Input Capture 1 - PTB1 Input buffer routed to Timer Channel 2 - Input Capture

Table 230. Port 1 input configuration (GPIO_IN1) (continued)

Field	Description
4 TCAP1	**PTB1** - Timer Input Capture Channel 1 0 - PTB1 Input buffer disconnected from Timer Channel 1 - Input Capture 1 - PTB1 Input buffer routed to Timer Channel 1 - Input Capture
3 TCAP0	**PTB1** - Timer Input Capture Channel 0 0 - PTB1 Input buffer disconnected from Timer Channel 0 - Input Capture 1 - PTB1 Input buffer routed to Timer Channel 0 - Input Capture
2 SCIRX	**PTB1** - SCI Module Rx Input 0 - PTB1 Input buffer disconnected from SCI Module Rx Input 1 - PTB1 Input buffer routed to SCI Module Rx Input
1 LINTX	**PTB1** - LIN Module Tx Input 0 - PTB1 Input buffer disconnected from LIN Module Tx Input 1 - PTB1 Input buffer routed to LIN Module Tx Input

6.4.5.3.5 Port 1 output configuration (GPIO_OUT1)

Table 231. Port 1 Output Configuration (GPIO_OUT1)

Offset[195] **0x45** **Access: User read/write**

	7	6	5	4	3	2	1	0
R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
W								PTBX1
Reset	0	0	0	0	0	0	0	0

Notes:
195.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 232. Port 1 output configuration (GPIO_OUT1)

Field	Description
7 WKUP	**PTB1** - Wake-up output 0 - Internal wake-up signal disconnected from PTB1 output buffer OR gate 1 - Internal wake-up signal connected to PTB1 output buffer OR gate
6 TCOMP3	**PTB1** - Timer Channel 3 - Output Compare output 0 - Timer Channel 3 - output compare disconnected from PTB1 output buffer OR gate 1 - Timer Channel 3 - output compare connected to PTB1 output buffer OR gate
5 TCOMP2	**PTB1** - Timer Channel 2 - Output Compare output 0 - Timer Channel 2 - output compare disconnected from PTB1 output buffer OR gate 1 - Timer Channel 2 - output compare connected to PTB1 output buffer OR gate
4 TCOMP1	**PTB1** - Timer Channel 1 - Output Compare output 0 - Timer Channel 1 - output compare disconnected from PTB1 output buffer OR gate 1 - Timer Channel 1 - output compare connected to PTB1 output buffer OR gate
3 TCOMP0	**PTB1** - Timer Channel 0 - Output Compare output 0 - Timer Channel 0 - output compare disconnected from PTB1 output buffer OR gate 1 - Timer Channel 0 - output compare connected to PTB1 output buffer OR gate
2 SCITX	**PTB1** - SCI TX Output 0 - SCI TX output disconnected from PTB1 output buffer OR gate 1 - SCI TX output connected to PTB1 output buffer OR gate
1 LINRX	**PTB1** - LIN RX Output 0 - LIN RX output disconnected from PTB1 output buffer OR gate 1 - LIN RX output connected to PTB1 output buffer OR gate
0 PTBX1	**PTB1** - Output Buffer Control 0 - PTB1 output buffer OR gate input = 0 1 - PTB1 output buffer OR gate input = 1

6.4.5.3.6 Port 2 input configuration (GPIO_IN2)

Table 233. Port 1 input configuration (GPIO_IN2)

Offset[196]	0x46						Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	
W								
Reset	0	0	0	0	0	0	0	0

Notes:
196.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 234. Port 2 input configuration (GPIO_IN2)

Field	Description
6 TCAP3	**PTB2** - Timer Input Capture Channel 3 0 - PTB2 Input buffer disconnected from Timer Channel 3 - Input Capture 1 - PTB2 Input buffer routed to Timer Channel 3 - Input Capture
5 TCAP2	**PTB2** - Timer Input Capture Channel 2 0 - PTB2 Input buffer disconnected from Timer Channel 2 - Input Capture 1 - PTB2 Input buffer routed to Timer Channel 2 - Input Capture
4 TCAP1	**PTB2** - Timer Input Capture Channel 1 0 - PTB2 Input buffer disconnected from Timer Channel 1 - Input Capture 1 - PTB2 Input buffer routed to Timer Channel 1 - Input Capture
3 TCAP0	**PTB2** - Timer Input Capture Channel 0 0 - PTB2 Input buffer disconnected from Timer Channel 0 - Input Capture 1 - PTB2 Input buffer routed to Timer Channel 0 - Input Capture
2 SCIRX	**PTB2** - SCI Module Rx Input 0 - PTB2 Input buffer disconnected from SCI Module Rx Input 1 - PTB2 Input buffer routed to SCI Module Rx Input
1 LINTX	**PTB2** - LIN Module Tx Input 0 - PTB2 Input buffer disconnected from LIN Module Tx Input 1 - PTB2 Input buffer routed to LIN Module Tx Input

6.4.5.3.7 Port 2 output configuration (GPIO_OUT2)

Table 235. Port 2 output configuration (GPIO_OUT2)

Offset[197]	0x47						Access: User read/write	
	7	6	5	4	3	2	1	0
R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
W								PTBX2
Reset	0	0	0	0	0	0	0	0

Notes:
197.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 236. Port 2 output configuration (GPIO_OUT2)

Field	Description
7 WKUP	**PTB2** - Wake-up output 0 - Internal wake-up signal disconnected from PTB2 output buffer OR gate 1 - Internal wake-up signal connected to PTB2 output buffer OR gate
6 TCOMP3	**PTB2** - Timer Channel 3 - Output Compare output 0 - Timer Channel 3 - output compare disconnected from PTB2 output buffer OR gate 1 - Timer Channel 3 - output compare connected to PTB2 output buffer OR gate

Table 236. Port 2 output configuration (GPIO_OUT2) (continued)

Field	Description
5 TCOMP2	**PTB2** - Timer Channel 2 - Output Compare output 0 - Timer Channel 2 - output compare disconnected from PTB2 output buffer OR gate 1 - Timer Channel 2 - output compare connected to PTB2 output buffer OR gate
4 TCOMP1	**PTB2** - Timer Channel 1 - Output Compare output 0 - Timer Channel 1 - output compare disconnected from PTB2 output buffer OR gate 1 - Timer Channel 1 - output compare connected to PTB2 output buffer OR gate
3 TCOMP0	**PTB2** - Timer Channel 0 - Output Compare output 0 - Timer Channel 0 - output compare disconnected from PTB2 output buffer OR gate 1 - Timer Channel 0 - output compare connected to PTB2 output buffer OR gate
2 SCITX	**PTB2** - SCI TX Output 0 - SCI TX output disconnected from PTB2 output buffer OR gate 1 - SCI TX output connected to PTB2 output buffer OR gate
1 LINRX	**PTB2** - LIN RX Output 0 - LIN RX output disconnected from PTB2 output buffer OR gate 1 - LIN RX output connected to PTB2 output buffer OR gate
0 PTBX2	**PTB2** - Output Buffer Control 0 - PTB2 output buffer OR gate input = 0 1 - PTB2 output buffer OR gate input = 1

6.4.5.3.8 Port 3 input configuration (GPIO_IN3)

Table 237. Port 3 input configuration (GPIO_IN3)

Offset[198] **0x48** **Access: User read/write**

	7	6	5	4	3	2	1	0
R	0	TCAP3	TCAP2	TCAP1	TCAP0	SCIRX	LINTX	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
198.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 238. Port 3 input configuration (GPIO_IN3)

Field	Description
6 TCAP3	**PTB3** - Timer Input Capture Channel 3 0 - PTB3 Input buffer disconnected from Timer Channel 3 - Input Capture 1 - PTB3 Input buffer routed to Timer Channel 3 - Input Capture
5 TCAP2	**PTB3** - Timer Input Capture Channel 2 0 - PTB3 Input buffer disconnected from Timer Channel 2 - Input Capture 1 - PTB3 Input buffer routed to Timer Channel 2 - Input Capture
4 TCAP1	**PTB3** - Timer Input Capture Channel 1 0 - PTB3 Input buffer disconnected from Timer Channel 1 - Input Capture 1 - PTB3 Input buffer routed to Timer Channel 1 - Input Capture
3 TCAP0	**PTB3** - Timer Input Capture Channel 0 0 - PTB3 Input buffer disconnected from Timer Channel 0 - Input Capture 1 - PTB3 Input buffer routed to Timer Channel 0 - Input Capture
2 SCIRX	**PTB3** - SCI Module Rx Input 0 - PTB3 Input buffer disconnected from SCI Module Rx Input 1 - PTB3 Input buffer routed to SCI Module Rx Input
1 LINTX	**PTB3** - LIN Module Tx Input 0 - PTB3 Input buffer disconnected from LIN Module Tx Input 1 - PTB3 Input buffer routed to LIN Module Tx Input

6.4.5.3.9 Port 3 output configuration (GPIO_OUT3)

Table 239. Port 3 output configuration (GPIO_OUT3)

Offset[198]	0x49						Access: User read/write	
	7	6	5	4	3	2	1	0
R	WKUP	TCOMP3	TCOMP2	TCOMP1	TCOMP0	SCITX	LINRX	0
W								PTBX3
Reset	0	0	0	0	0	0	0	0

Notes:
199.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 240. Port 3 output configuration (GPIO_OUT3)

Field	Description
7 WKUP	**PTB3** - Wake-up output 0 - Internal wake-up signal disconnected from PTB3 output buffer OR gate 1 - Internal wake-up signal connected to PTB3 output buffer OR gate
6 TCOMP3	**PTB3** - Timer Channel 3 - Output Compare output 0 - Timer Channel 3 - output compare disconnected from PTB3 output buffer OR gate 1 - Timer Channel 3 - output compare connected to PTB3 output buffer OR gate
5 TCOMP2	**PTB3** - Timer Channel 2 - Output Compare output 0 - Timer Channel 2 - output compare disconnected from PTB3 output buffer OR gate 1 - Timer Channel 2 - output compare connected to PTB3 output buffer OR gate
4 TCOMP1	**PTB3** - Timer Channel 1 - Output Compare output 0 - Timer Channel 1 - output compare disconnected from PTB3 output buffer OR gate 1 - Timer Channel 1 - output compare connected to PTB3 output buffer OR gate
3 TCOMP0	**PTB3** - Timer Channel 0 - Output Compare output 0 - Timer Channel 0 - output compare disconnected from PTB3 output buffer OR gate 1 - Timer Channel 0 - output compare connected to PTB3 output buffer OR gate
2 SCITX	**PTB3** - SCI TX Output 0 - SCI TX output disconnected from PTB3 output buffer OR gate 1 - SCI TX output connected to PTB3 output buffer OR gate
1 LINRX	**PTB3** - LIN RX Output 0 - LIN RX output disconnected from PTB3 output buffer OR gate 1 - LIN RX output connected to PTB3 output buffer OR gate
0 PTBX3	**PTB3** - Output Buffer Control 0 - PTB3 output buffer OR gate input = 0 1 - PTB3 output buffer OR gate input = 1

6.4.5.3.10 Port 4 input configuration (GPIO_IN4)

Table 241. Port 4 input configuration (GPIO_IN4)

Offset[200]	0x4A						Access: User read/write	
	7	6	5	4	3	2	1	0
R	PTWU	TCAP3	TCAP2	TCAP1	TCAP0	0	NWUS	NWUE
W								
Reset	0	0	0	0	0	0	0	0

Notes:
200.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 242. Port 4 input configuration (GPIO_IN4)

Field	Description
7 PTWU	**PTB4** - Wake-up and input enable. This bit must be set to enable PTB4 wake-up capability and if PTB4 is used as an input (e.g. for timer input capture or to read GPIO4_PD4) 0 - PTB4 Input buffer Low-power mode wake-up circuity disabled 1 - PTB4 Input buffer Low-power mode wake-up circuity enabled
6 TCAP3	**PTB4** - Timer Input Capture Channel 3 0 - PTB4 Input buffer disconnected from Timer Channel 3 - Input Capture 1 - PTB4 Input buffer routed to Timer Channel 3 - Input Capture
5 TCAP2	**PTB4** - Timer Input Capture Channel 2 0 - PTB4 Input buffer disconnected from Timer Channel 2 - Input Capture 1 - PTB4 Input buffer routed to Timer Channel 2 - Input Capture
4 TCAP1	**PTB4** - Timer Input Capture Channel 1 0 - PTB4 Input buffer disconnected from Timer Channel 1 - Input Capture 1 - PTB4 Input buffer routed to Timer Channel 1 - Input Capture
3 TCAP0	**PTB4** - Timer Input Capture Channel 0 0 - PTB4 Input buffer disconnected from Timer Channel 0 - Input Capture 1 - PTB4 Input buffer routed to Timer Channel 0 - Input Capture
1 NWUS	**PTB4** - negative wake-up detection duration (for NWUE = 1) 0 - set to Wake-up filter time 1 (typ. 700 ns) 1 - set to Wake-up filter time 2 (typ. 1.6 µs)
0 NWUE	**PTB4** - negative wake-up detection 0 - positive wake-up detection (typ. 20 µs deglitch filter) 1 - negative wake-up detection enable

6.5 Port integration module (S12ZIPIMV1)

6.5.1 Introduction

The Port Integration Module (PIM) establishes the interface between the S12ZI128 peripheral modules SPI, MSCAN, and Die-To-Die Interface module (D2DI) to the I/O pins of the MCU.

All port A and port B pins support general purpose I/O functionality if not in use with other functions. The PIM controls the signal prioritization and multiplexing on shared pins.

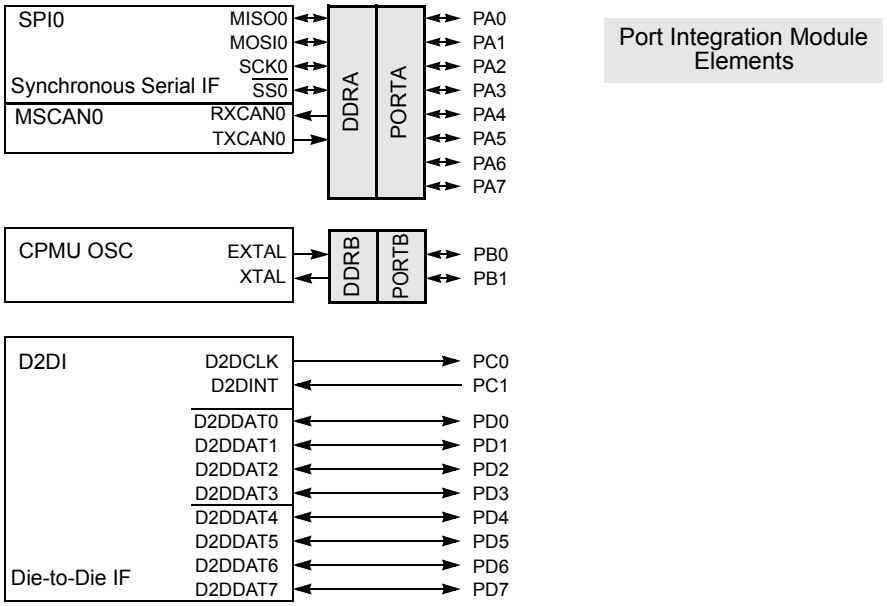


Figure 52. Block diagram

MM9Z1_638

6.5.1.1 Features

- 8-pin port A associated with 1 SPI, 1 MSCAN, and S12ZDBG module
- 2-pin port B associated with the CPMU OSC module
- 2-pin port C used as D2DI clock output and D2DI interrupt input
- 8-pin port D used as 8 or 4 bit data I/O for the D2DI module
- GPIO function shared on port A, B pins
- Pull-down devices on PC1 and PD7-0 if used as D2DI inputs
- Reduced drive capability on PC0 and PD7-0 on per pin basis

The Port Integration Module includes these distinctive registers:

- Data registers for ports A, B when used as general purpose I/O
- Data direction registers for ports A, B when used as general purpose I/O
- Port input register on ports A, B, C and D
- Reduced drive register on port C and D

A standard port A and port B pin has the following features:

- Input/output selection
- 3.15 to 5.5 V output drive
- 3.15 to 5.5 V digital input

A standard port C and D pin has the following features:

- Input/output selection
- 2.25 to 3.6 V output drive
- 2.25 to 3.6 V digital input

6.5.2 External signal description

This section lists and describes the signals that do connect off-chip.

Table 243 shows all the pins and their functions that are controlled by the Port Integration Module.

NOTE:

If there is more than one function associated with a pin, the priority is indicated by the position in the table from top (highest priority) to bottom (lowest priority).

Table 243. Pin functions and priorities

Port	Pin name	Pin function and priority	I/O	Description	Pin function after reset
-	BKGD	MODC [201]	I	MODC input during RESET	BKGD
		BKGD	I/O	S12ZBDC communication	
A	PA7	PDO	O	DBG profiling data	GPI
		DBGEEV	I	DBG external event	
		GPIO	I/O	General Purpose	
	PA6	PDOCLK	O	DBG profiling clock	
		GPIO	I/O	General Purpose	
	PA5	TXCAN0	O	MSCAN0 transmit	
		GPIO	I/O	General Purpose	
	PA4	RXCAN0	I	MSCAN0 receive	
		GPIO	I/O	General Purpose	
	PA3	SS0	I/O	SPI0 slave select	
		GPIO	I/O	General Purpose	
	PA2	SCK0	I/O	SPI0 serial clock	
		GPIO	I/O	General Purpose	
	PA1	MOSI0	I/O	SPI0 master out/slave in	
		GPIO	I/O	General Purpose	
	PA0	MISO0	I/O	SPI0 master in/slave out	
		GPIO	I/O	General Purpose	
B	PB1	XTAL	-	CPMU OSC signal	GPI
		GPIO	I/O	General Purpose	
	PB0	EXTAL	-	CPMU OSC signal	
		GPIO	I/O	General Purpose	
C[202]	PC1	D2DINT	I	Die-to-Die interface interrupt input	
	PC0	D2DCLK	O	Die-to-Die interface clock output	
D[201]	PD7	D2DDAT7	I/O	Die-to-Die interface data bit 7	
	PD6	D2DDAT6	I/O	Die-to-Die interface data bit 6	
	PD5	D2DDAT5	I/O	Die-to-Die interface data bit 5	
	PD4	D2DDAT4	I/O	Die-to-Die interface data bit 4	
	PD3	D2DDAT3	I/O	Die-to-Die interface data bit 3	
	PD2	D2DDAT2	I/O	Die-to-Die interface data bit 2	
	PD1	D2DDAT1	I/O	Die-to-Die interface data bit 1	
	PD0	D2DDAT0	I/O	Die-to-Die interface data bit 0	

Notes:
201.Function active when RESET asserted.
202.Not bonded out if used as internal interface to companion die.

6.5.3 Memory map and register definition

This section provides a detailed description of all Port Integration Module registers.

6.5.3.1 Memory map

Table 244. Memory map

Global address	Register name		Bit 7	6	5	4	3	2	1	Bit 0
0x0200–0x020D	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x020E	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x020F	Reserved	R	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
		W								
0x0210–0x021F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0220	PTA	R	PTA7	PTA6	PTA5	PTA4	PTA3	PTA2	PTA1	PTA0
		W								
0x0221	PTB	R	0	0	0	0	0	0	PTB1	PTB0
		W								
0x0222	PTIA	R	PTIA7	PTIA6	PTIA5	PTIA4	PTIA3	PTIA2	PTIA1	PTIA0
		W								
0x0223	PTIB	R	0	0	0	0	0	0	PTIB1	PTIB0
		W								
0x0224	DDRA	R	DDRA7	DDRA6	DDRA5	DDRA4	DDRA3	DDRA2	DDRA1	DDRA0
		W								
0x0225	DDRB	R	0	0	0	0	0	0	DDRB1	DDRB0
		W								
0x0226	PERA	R	PERA7	PERA6	PERA5	PERA4	PERA3	PERA2	PERA1	PERA0
		W								
0x0227	PERB	R	0	0	0	0	0	0	PERB1	PERB0
		W								
0x0228	PPSA	R	PPSA7	PPSA6	PPSA5	PPSA4	PPSA3	PPSA2	PPSA1	PPSA0
		W								
0x0229	PPSB	R	0	0	0	0	0	0	PPSB1	PPSB0
		W								
0x022A–0x0259	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x025A	RDRC	R	0	0	0	0	0	0	0	RDRC0
		W								
0x025B	RDRD	R	RDRD7	RDRD6	RDRD5	RDRD4	RDRD3	RDRD2	RDRD1	RDRD0
		W								
0x025C–0x02FF	Reserved	R	0	0	0	0	0	0	0	0
		W								

6.5.3.2 PIM registers 0x0200-0x020F

This section details the specific purposes of register implemented in address range 0x0200-0x020F. These registers serve for specific PIM related functions not part of the generic port registers.

- If not stated differently, writing to reserved bits has no effect and read returns zero.
- All register read accesses are synchronous to internal clocks.
- Register bits can be written at any time if not stated differently.

6.5.3.2.1 Reserved register

Table 245. Reserved register

Address	0x020E							Access: User read/write[203]
	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	x	x	x	x	x	x	x	x

Notes:

203.Read: Anytime
Write: Only in special mode
This reserved register is designed for factory test purposes only and is not intended for general user access. Writing to this register when in special modes can alter the modules functionality

6.5.3.2.2 Reserved register

Table 246. Reserved register

Address	0x020F							Access: User read/write[204]
	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Reset	x	x	x	x	x	x	x	x

Notes:

204.Read: Anytime
Write: Only in special mode
This reserved register is designed for factory test purposes only and is not intended for general user access. Writing to this register when in special modes can alter the modules functionality

6.5.3.3 PIM generic registers

This section describes the details of all configuration registers.

- Writing to reserved bits has no effect and read returns zero.
- All register read accesses are synchronous to internal clocks.
- All registers can be written at any time, however a specific configuration might not become active. E.g. a pull-up device does not become active while the port is used as a push-pull output.
- General purpose data output availability depends on prioritization; input data registers always reflect the pin status independent of the use.
- Pull device availability, pull device polarity, wired or mode, key wake-up functionality are independent of the prioritization unless noted differently.
- For availability of individual bits refer to Memory map” and Table 260.

6.5.3.3.1 Port data register

Table 247. Port data register

Address	0x0220 PTA 0x0221 PTB					Access: User read/write[205]		
	7	6	5	4	3	2	1	0
R W	PTx7	PTx6	PTx5	PTx4	PTx3	PTx2	PTx1	PTx0
Reset	0	0	0	0	0	0	0	0

Notes:
205.Read: Anytime. The data source is depending on the data direction value.
Write: Anytime

This is a generic description of the standard port data registers. Refer to Table 260 to determine the implemented bits in the respective register. Unimplemented bits read zero.

Table 248. Port data register field descriptions

Field	Description
7-0 PTx7-0	**Port — General purpose input/output data** This register holds the value driven out to the pin if the pin is used as a general purpose output. When not used with the alternative function (refer to Table 243), these pins can be used as general purpose I/O. If the associated data direction bits of these pins are set to 1, a read returns the value of the port register, otherwise the buffered pin input state is read.

6.5.3.3.2 Port input register

Table 249. Port input register

Address	0x0222 PTIA 0x0223 PTIB					Access: User read only[206]		
	7	6	5	4	3	2	1	0
R	PTIx7	PTIx6	PTIx5	PTIx4	PTIx3	PTIx2	PTIx1	PTIx0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
206.Read: Anytime.
Write: Never

This is a generic description of the standard port input registers. Refer to Table 260 to determine the implemented bits in the respective register. Unimplemented bits read zero.

Table 250. Port input register field descriptions

Field	Description
7-0 PTIx7-0	**Port Input** — Data input A read always returns the buffered input state of the associated pin. It can be used to detect overload or short-circuit conditions on output pins.

6.5.3.3.3 Data direction register

Table 251. Data direction register

Address	0x0224 DDRA 0x0225 DDRB				Access: User read/write[207]			
	7	6	5	4	3	2	1	0
R W	DDRx7	DDRx6	DDRx5	DDRx4	DDRx3	DDRx2	DDRx1	DDRx0
Reset	0	0	0	0	0	0	0	0

Notes:
207.Read: Anytime.
Write: Anytime

This is a generic description of the standard data direction registers. Refer to Table 260 to determine the implemented bits in the respective register. Unimplemented bits read zero.

Table 252. Data direction register field descriptions

Field	Description
7-0 DDRx7-0	**Data Direction** — Select general purpose data direction This bit determines whether the pin is a general purpose input or output. If a peripheral module controls the pin the content of the data direction register is ignored. Independent of the pin usage with a peripheral module this register determines the source of data when reading the associated data register address. Due to internal synchronization circuits, it can take up to two bus clock cycles until the correct value is read on port data and port input registers, when changing the data direction register. 1 Associated pin is configured as output 0 Associated pin is configured as input

6.5.3.3.4 Pull device enable register

Table 253. Pull device enable register

Address	0x0226 PERA 0x0227 PERB				Access: User read/write[208]			
	7	6	5	4	3	2	1	0
R W	PERx7	PERx6	PERx5	PERx4	PERx3	PERx2	PERx1	PERx0
Reset								
Port B:	0	0	0	0	0	0	1	1
Others:	0	0	0	0	0	0	0	0

Notes:
208.Read: Anytime.
Write: Anytime

This is a generic description of the standard pull device enable registers. Refer to Table 260 to determine the implemented bits in the respective register. Unimplemented bits read zero.

Table 254. Pull device enable register field descriptions

Field	Description
7-0 PERx7-0	**Pull Enable** — Activate pull device on input pin This bit controls whether a pull device on the associated port input or open-drain output pin is active. If a pin is used as push-pull output this bit has no effect. The polarity is selected by the related polarity select register bit. On open-drain output pins only a pull-up device can be enabled. 1 Pull device enabled 0 Pull device disabled

6.5.3.3.5 Polarity select register

Table 255. Polarity select register

Address	0x0228 PPSA 0x0229 PPSB					Access: User read/write[209]		
	7	6	5	4	3	2	1	0
R W	PPSx7	PPSx6	PPSx5	PPSx4	PPSx3	PPSx2	PPSx1	PPSx0
Reset								
Port B:	0	0	0	0	0	0	1	1
Others:	0	0	0	0	0	0	0	0

Notes:
209.Read: Anytime.
Write: Anytime

This is a generic description of the standard polarity select registers. Refer to Table 260 to determine the implemented bits in the respective register. Unimplemented bits read zero.

Table 256. Polarity select register field descriptions

Field	Description
7-0 PPSx7-0	**Pull Polarity Select** — Configure pull device and pin interrupt edge polarity on input pin This bit selects a pull-up or a pull-down device if enabled on the associated port input pin. If a port has interrupt functionality this bit also selects the polarity of the active edge. If MSCAN is active a pull-up device can be activated on the RXCAN input; attempting to select a pull-down disables the pull device. 1 Pull-down device selected; rising edge selected 0 Pull-up device selected; falling edge selected

6.5.3.3.6 Reduced drive register

Table 257. Reduced drive register

Address	0x025A RDRC 0x025B RDRD					Access: User read/write[210]		
	7	6	5	4	3	2	1	0
R W	RDRx7	RDRx6	RDRx5	RDRx4	RDRx3	RDRx2	RDRx1	RDRx0
Reset	0	0	0	0	0	0	0	0

Notes:
210.Read: Anytime.
Write: Anytime

This is a generic description of the standard reduced drive registers. Refer to Table 260 to determine the implemented bits in the respective register. Unimplemented bits read zero.

Table 258. Reduced drive register field descriptions

Field	Description
7-0 RDRx7-0	**Reduced Drive Register** — Select reduced drive for output pin This bit configures the drive strength of the associated output pin as either full or reduced. If a pin is used as input, this bit has no effect. The reduced drive function is independent of which function is being used on a particular pin. 1 Reduced drive selected (approx. 1/10 of the full drive strength) 0 Full drive strength enabled

6.5.3.3.7 PIM reserved register

Table 259. PIM reserved register

Address	(any reserved)	Access: User read[211]

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
211.Read: Always reads 0x00.
Write: Unimplemented

6.5.3.4 Functional description

6.5.3.4.1 General

Each pin except BKGD can act as general purpose I/O. In addition each pin can act as an output or input of a peripheral module.

6.5.3.4.2 Registers

Table 260 lists the implemented configuration bits which are available on each port. These registers except the pin input registers can be written at any time, however a specific configuration might not become active. For example a pull-up device does not become active while the port is used as a push-pull output.

Unimplemented bits read zero.

Table 260. Bit indices of implemented register bits per port

Port	Port data register	Port input register	Data direction register	Pull device enable register	Polarity select register	Port interrupt enable register	Port interrupt flag register	Digital input enable register	Reduced drive register	Wired-or mode register
A	7-0	7-0	7-0	7-0	7-0	-	-	-	-	-
B	1-0	1-0	1-0	1-0	1-0	-	-	-	-	-
C	1-0	1-0	1-0	-	-	-	-	-	0	-
D	7-0	7-0	7-0	-	-	-	-	-	7-0	-

Table 261 shows the effect of enabled peripheral features on I/O state and enabled pull devices.

Table 261. Effect of enabled features

Enabled Feature[212]	Related Pin(s)	Effect on I/O State	Effect on Enabled Pull Device
CPMU OSC	EXTAL, XTAL	CPMU takes control	Forced off
SPI0	MISO0, MOSI0, SCK0, $\overline{SS0}$	Controlled input/output	Forced off if output
MSCAN0	TXCAN0	Forced output	Forced off
	RXCAN0	Forced input	Pull-down forced off
S12ZDBG	PDO, PDOCLK	Forced output	Forced off
D2DI	D2DCLK	Forced output	Forced off
	D2DINT	Forced input	Pull-down forced on if D2DCTL0[D2DEN]=1 and used as inputs, no config bits available
	D2DDATx	Controlled input/output	

Notes:
212.If applicable the appropriate routing configuration must be set for the signals to take effect on the pins.

6.5.3.4.3 Pin I/O control

Figure 53 illustrates the data paths to and from an I/O pin. Input and output data can always be read via the input register (PTIx, Port input register") independent if the pin is used as general purpose I/O or with a shared peripheral function. If the pin is configured as input (DDRx=0, Data direction register"), the pin state can also be read through the data register (PTx, Port data register").

The general purpose data direction configuration can be overruled by an enabled peripheral function shared on the same pin (Table 261). If more than one peripheral function is available and enabled at the same time, the highest ranked module according the predefined priority scheme in Table 243 takes precedence on the pin.

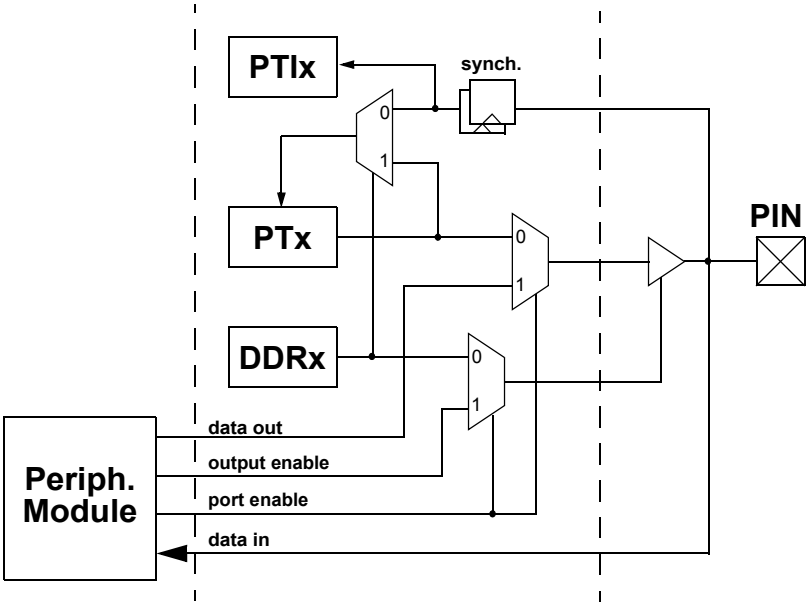


Figure 53. Illustration of I/O pin functionality

6.5.3.5 Initialization information

6.5.3.5.1 Port data and data direction register writes

It is not recommended to write PTx and DDRx in a word access. When changing the register pins from inputs to outputs, the data may have extra transitions during the write access. Initialize the port data register before enabling the outputs.

6.6 Die to die interface

6.6.1 Die-to-die initiator (D2DIV2)

6.6.1.1 Preface

This document contains the user specification of the D2D Initiator.

6.6.1.2 Acronyms and abbreviations

Table 262 contains sample acronyms and abbreviations used in this document.

Table 262. Acronyms and abbreviated terms

Term	Meaning
D2D	Die-to-Die

6.6.1.3 Glossary

Table 340 shows a glossary of the major terms used in this document.

Table 263. Glossary

Term	Definition
Active low	The signal is asserted when it changes to logic-level zero.
Active high	The signal is asserted when it changes to logic-level one.
Asserted	Discrete signal is in active logic state.
Customer	The end user of an SoC design or device.
EOT	End of Transaction
Negated	A discrete signal is in inactive logic state.
Pin	External physical connection.
Signal	Electronic construct whose state or change in state conveys information.
Transfer	A read or write on the CPU bus following the IP-Bus protocol.
Transaction	Command, address and if required data sent on the D2D interface. A transaction is finished by the EOT acknowledge cycle.

6.6.1.4 Introduction

This section describes the functionality of the die-to-die (D2DIV2) initiator block especially designed for low cost connections between a microcontroller die (Interface Initiator) and an analog die (Interface Target) located in the same package.

The D2DI block

- realizes the initiator part of the D2D interface, including supervision and error interrupt generation
- generates the clock for this interface
- disables/enables the interrupt from the D2D interface

6.6.1.4.1 Overview

The D2DI is the initiator for a data transfer to and from a target typically located on another die in the same package. It provides a set of configuration registers and two memory mapped 256 Byte address windows. When writing to a window a transaction is initiated sending a write command, followed by an 8-bit address and the data byte or word to the target. When reading from a window a transaction is initiated sending a read command, followed by an 8-bit address to the target. The target then responds with the data. The basic idea is that a peripheral located on another die, can be addressed like an on-chip peripheral, except for a small transaction delay.

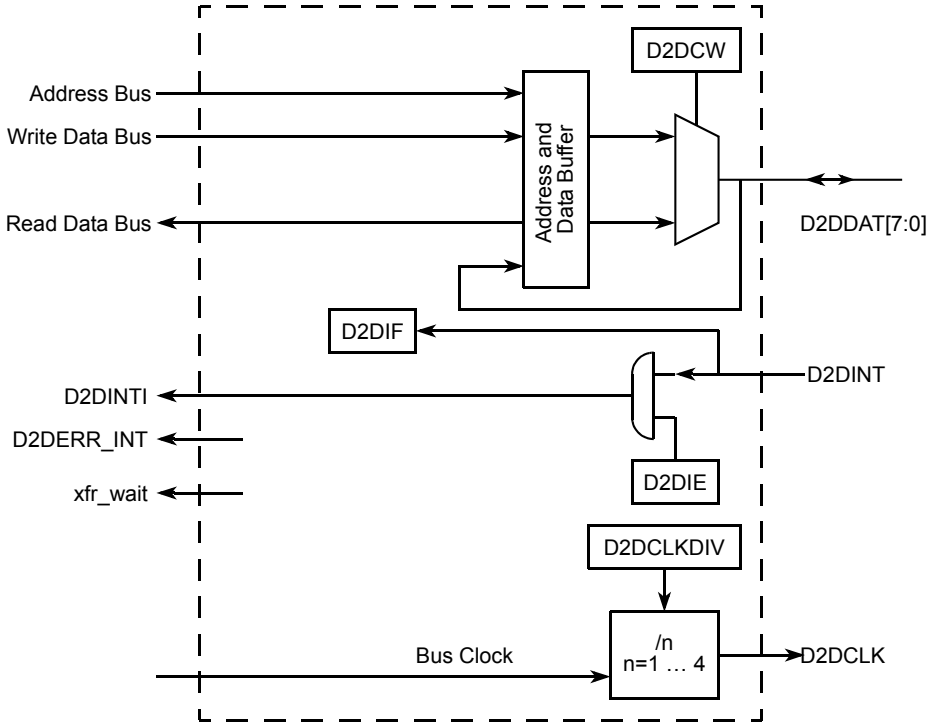


Figure 54. Die-to-die initiator (D2DI) block diagram

6.6.1.4.2 Features

The main features of this block are

- Software transparent, memory mapped access to peripherals on target die
  - 256 Byte address window
  - Supports blocking read or write as well as non-blocking write transactions
- Scalable interface clock divide by 1, 2, 3, or 4 of bus clock
- Clock halt on system STOP
- Configurable for 4- or 8-bit wide transfers
- Configurable timeout period
- Non-maskable interrupt on transaction errors
- Transaction status and error flags
- Interrupt enable for receiving interrupt (from D2D target)

6.6.1.4.3 Modes of operation

6.6.1.4.3.1 D2DI in stop/wait mode

The D2DI stops working in Stop/Wait mode. The D2DCLK signal as well as the data signals used are driven low (only after the end of the current high phase, as defined by D2DCLKDIV).

Waking from Stop/Wait mode, the D2DCLK line starts clocking again and the data lines will be driven low until the first transaction starts.

Stop and Wait mode are entered by different CPU instructions. In the Wait mode the behavior of the D2DI can be configured (D2DSWAI). Every (enabled) interrupt can be used to leave the Stop and Wait mode.

6.6.1.4.3.2 D2DI in special modes

The MCU can enter a special mode (used for test and debugging purposes as well as programming the FLASH). In the D2DI the “write-once” feature is disabled. See the MCU description for details.

6.6.1.5 External signal description

The D2DI optionally uses 6 or 10 port pins. The functions of those pins depends on the settings in the D2DCTL0 register, when the D2DI module is enabled.

6.6.1.5.1 D2DCLK

When the D2DI is enabled this pin is the clock output. This signal is low if the initiator is disabled, in Stop mode or in Wait mode (with D2DSWAI asserted), otherwise it is a continuos clock. This pin may be shared with general purpose functionality if the D2DI is disabled.

6.6.1.5.2 D2DDAT[7:4]

When the D2DI is enabled and the interface connection width D2DCW is set to be 8-bit wide, those lines carry the data bits 7:4 acting as outputs or inputs. When they act as inputs pull-down elements are enabled. If the D2DI is disabled or if the interface connection width is set as 4-bit wide, the pins may be shared with general purpose pin functionality.

6.6.1.5.3 D2DDAT[3:0]

When the D2DI is enabled those lines carry the data bits 3:0 acting as outputs or inputs. When they act as inputs pull-down elements are enabled. If the D2DI is disabled the pins and may be shared with general purpose pin functionality.

6.6.1.5.4 D2DINT

The D2DINT is an active input interrupt input driven by the target device. The pin has an active pull-down device. If the D2DI is disabled the pin may be shared with general purpose pin functionality.

Table 264. Signal properties

Name	Primary (D2DEN=1)	I/O	Secondary (D2DEN=0)	Reset	Comment	Pull-down
D2DDAT[7:0]	Bi-directional Data Lines	I/O	GPIO	0	driven low if in STOP mode	Active[213]
D2DCLK	Interface Clock Signal	O	GPIO	0	low if in STOP mode	—
D2DINT	Active High Interrupt	I	GPIO	—	—	Active[214]

Notes:
213.Active if in input state, only if D2DEN=1
214.only if D2DEN=1

See the port interface module (PIM) guide for details of the GPIO function.

6.6.1.6 Memory map and register definition

6.6.1.6.1 Memory map

The D2DI memory map is split into three sections.

1. An eight byte set of control registers
2. A 256 byte window for blocking transactions
3. A 256 byte window for non-blocking transactions

See Device memory map for the register layout (distribution of these sections).

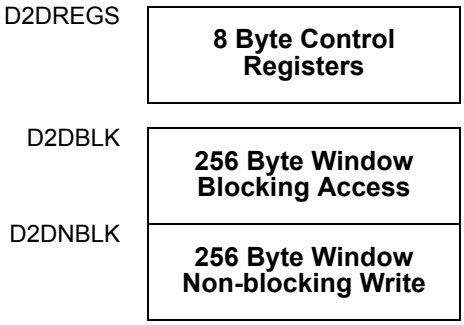


Figure 55. D2DI top level memory map

A summary of the registers associated with the D2DI block is shown in Table 265. Detailed descriptions of the registers and bits are given in the subsections that follow.

Table 265. D2DI register summary

Offset	Register name		Bit 7	6	5	4	3	2	1	Bit 0
0x0	D2DCTL0	R	D2DEN	D2DCW	D2DSWAI	0	0	0	D2DCLKDIV[1:0]	
		W								
0x1	D2DCTL1	R	D2DIE	0	0	0	TIMOUT[3:0]			
		W								
0x2	D2DSTAT0	R	ERRIF	ACKERF	CNCLF	TIMEF	TERRF	PARF	PAR1	PAR0
		W								
0x3	D2DSTAT1	R	D2DIF	D2DBSY	0	0	0	0	0	0
		W								
0x4	D2DADRHI	R	RWB	SZ8	0	NBLK	0	0	0	0
		W								
0x5	D2DADRLO	R	ADR[7:0]							
		W								
0x6	D2DDATAHI	R	DATA[15:8]							
		W								
0x7	D2DDATALO	R	DATA[7:0]							
		W								
			= Unimplemented or Reserved							

6.6.1.6.2 Register definition

6.6.1.6.2.1 D2DI control register 0 (D2DCTL0)

This register is used to enable and configure the interface width, the wait behavior and the frequency of the interface clock.

Table 266. D2DI control register 0 (D2DCTL0)

Offset	0x0						Access: User read/write	
	7	6	5	4	3	2	1	0
R	D2DEN	D2DCW	D2DSWAI	0	0	0	D2DCLKDIV[1:0]	
W								
Reset	0	0	0	0	0	0	0	0

Table 267. D2DCTL0 register field descriptions

Field	Description
7 D2DEN	**D2DI Enable** — Enables the D2DI module. This bit is write-once in normal mode and can always be written in special modes. 0 D2DI initiator is disabled. No lines are not used, the pins have their GPIO (secondary) function. 1 D2DI initiator is enabled. After setting D2DEN=1 the D2DDAT[7:0] (or [3:0], see D2DCW) lines are driven low with the IDLE command; the D2DCLK is driven by the divided bus clock.
6 D2DCW	**D2D Connection Width** — Sets the number of data lines used by the interface. This bit is write-once in normal modes and can always be written in special modes. Must be set to 1 for MM9Z1_638. 0 Lines D2DDAT[3:0] are used for four line data transfer. D2DDAT[7:4] are unused. 1 All eight interface lines D2DDAT[7:0] are used for data transfer.
5 D2DSWAI	**D2D Stop In Wait** — Controls the Wait behaviour. This bit can be written at any time. 0 Interface clock continues to run if the CPU enters Wait mode 1 Interface clock stops if the CPU enters Wait mode.
4:2	Reserved, should be written to 0 to ensure compatibility with future versions of this interface.
1:0 D2DCLKDIV	**Interface Clock Divider** — Determines the frequency of the interface clock. These bits are write-once in normal modes and can be always written in special modes. See Figure 56 for details on the clock waveforms 00 Encoding 0. Bus clock divide by 1. 01 Encoding 1. Bus clock divide by 2. 10 Encoding 2. Bus clock divide by 3. 11 Encoding 3. Bus clock divide by 4.

The Clock Divider will provide the waveforms as shown in Figure 56. The duty cycle of the clock is not 50%. If the D2DCLKDIV = 00, then the interface clock is the bus clock, otherwise the high phase of the interface clock is $t_{BUS}/2$, since this is beneficial for the transaction timing.

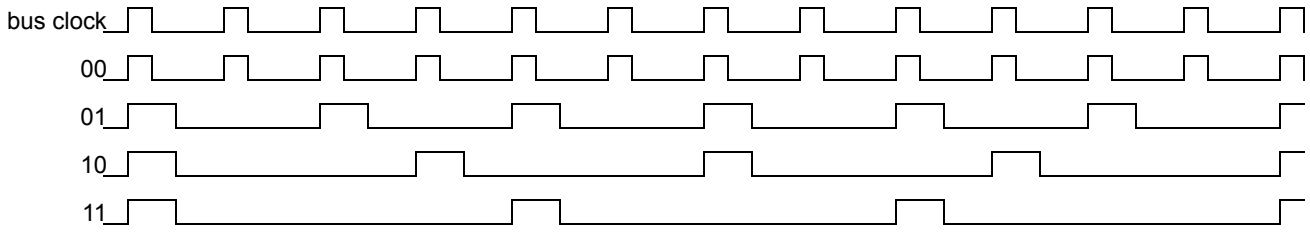


Figure 56. Interface clock waveforms for various D2DCLKDIV encoding

6.6.1.6.2.2 D2DI control register 1 (D2DCTL1)

This register is used to enable the D2DI interrupt and set number of D2DCLK cycles before a timeout error is asserted.

Table 268. D2DI control register 1 (D2DCTL1)

Offset **0x1** **Access: User read/write**

	7	6	5	4	3	2	1	0
R	D2DIE	0	0	0	TIMOUT[3:0]			
W								
Reset	0	0	0	0	0	0	0	0

Table 269. D2DCTL1 register field descriptions

Field	Description
7 D2DIE	**D2D Interrupt Enable** — Enables the external interrupt 0 External Interrupt is disabled 1 External Interrupt is enabled
6:4	Reserved, should be written to 0 to ensure compatibility with future versions of this interface.
3:0 TIMOUT	**Timeout Setting** — Defines the number of D2DCLK cycles to wait after the last transaction cycle until a timeout is asserted. In case of a timeout the TIMEF flag in the D2DSTAT0 register will be set. These bits are write-once in normal modes and can always be written in special modes. 0000 The acknowledge is expected directly after the last transfer, i.e. the target must not insert a wait cycle. 0001 - 1111: The target may insert up to TIMOUT wait states before acknowledging a transaction until a timeout is asserted

Note

"Write-once" means that after writing D2DCNTL0.D2DEN=1 the write accesses to these bits have no effect.

6.6.1.6.2.3 D2DI status register 0 (D2DSTAT0)

This register reflects the status of the D2DI transactions.

Table 270. D2DI status register 0 (D2DSTAT0)

Offset **0x2** **Access: User read/write**

	7	6	5	4	3	2	1	0
R	ERRIF	ACKERF	CNCLF	TIMEF	TERRF	PARF	PAR1	PAR0
W								
Reset	0	0	0	0	0	0	0	0

Table 271. D2DI status register 0 field descriptions

Field	Description
7 ERRIF	**D2DI error interrupt flag** — This status bit indicates that the D2D initiator has detected an error condition (summary of the following five flags).This interrupt is not locally maskable. Write a 1 to clear the flag. Writing a 0 has no effect. 0 D2DI has not detected an error during a transaction. 1 D2DI has detected an error during a transaction.
6 ACKERF	**Acknowledge Error Flag**— This read-only flag indicates that in the acknowledge cycle not all data inputs are sampled high, indicating a potential broken wire. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
5 CNCLF	**CNCLF** — This read-only flag indicates the initiator has canceled a transaction and replaced it by an IDLE command due to a pending error flag (ERRIF). This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
4 TIMEF	**Time Out Error Flag** — This read-only flag indicates the initiator has detected a timeout error. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
3 TERRF	**Transaction Error Flag** — This read-only flag indicates the initiator has detected the error signal during the acknowledge cycle of the transaction. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
2 PARF	**Parity Error Flag** — This read-only flag indicates the initiator has detected a parity error. Parity bits[1:0] contain further information. This flag is cleared when the ERRIF bit is cleared by writing a 1 to the ERRIF bit.
1 PAR1	**Parity Bit** — P[1] as received by the D2DI
0 PAR0	**Parity Bit** — P[0] as received by the D2DI

6.6.1.6.2.4 D2DI status register 1 (D2DSTAT1)

This register holds the status of the external interrupt pin and an indicator about the D2DI transaction status.

Table 272. D2DI status register 1 (D2DSTAT1)

Offset	0x3						Access: User read	
	7	6	5	4	3	2	1	0
R	D2DIF	D2DBSY	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Table 273. D2DSTAT1 register field descriptions

Field	Description
7 D2DIF	**D2D Interrupt Flag** — This read-only flag reflects the status of the D2DINT Pin. The D2D interrupt flag can only be cleared by a target specific interrupt acknowledge sequence. 0 External Interrupt is negated 1 External Interrupt is asserted
6 D2DBSY	**D2D Initiator Busy** — This read-only status bit indicates that a D2D transaction is ongoing. 0 D2D initiator idle. 1 D2D initiator transaction ongoing.
5:0	Reserved, should be masked to ensure compatibility with future versions of this interface.

6.6.1.6.2.5 D2DI address buffer register (D2DADR)

This read-only register contains information about the ongoing D2D interface transaction. The register content will be updated when a new transaction starts. In error cases, the user can track back which transaction failed.

Table 274. D2DI address buffer register (D2DADR)

Offset	0x4/0x5								Access: User read							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RWB	SZ8	0	NBLK	0	0	0	0	ADR[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 275. D2DI address buffer register bit descriptions

Field	Description
15 RWB	**Transaction Read-Write Direction** — This read-only bit reflects the direction of the transaction 0 Write Transaction 1 Read Transaction
14 SZ8	**Transaction Size** — This read-only bit reflects the data size of the transaction 0 16-bit transaction. 1 8-bit transaction.
13	Reserved, should be masked to ensure compatibility with future versions of this interface.
12 NBLK	**Transaction Mode** — This read-only bit reflects the mode of the transaction 0 Blocking transaction. 1 Non-blocking transaction.
11:8	Reserved, should be masked to ensure compatibility with future versions of this interface.
7:0 ADR[7:0]	**Transaction Address** — Those read-only bits contain the address of the transaction

6.6.1.6.2.6 D2DI data buffer register (D2DDATA)

This read-only register contains information about the ongoing D2D interface transaction. For a write transaction the data becomes valid at the begin of the transaction. For a read transaction the data will be updated during the transaction and is finalized when the transaction is acknowledged by the target. In error cases, the user can track back what has happened.

Table 276. D2DI data buffer register (D2DDATA)

Offset	0x6/0x7												Access: User read			
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DATA15:0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 277. D2DI data buffer register bit descriptions

Field	Description
15:0 DATA	**Transaction Data** — Those read-only bits contain the data of the transaction

Both D2DDATA and D2DADR can be read with byte accesses.

6.6.1.7 Functional description

6.6.1.7.1 Initialization

Out of reset the interface is disabled. The interface must be initialized by setting the interface clock speed, the timeout value, the transfer width and finally enabling the interface. This should be done using a 16-bit write or if using 8-bit write D2DCTL1 must be written before D2D2CTL0.D2DEN=1 is written. Once it is enabled in normal modes, only a reset can disable it again (write-once feature).

6.6.1.7.2 Transactions

A transaction on the D2D Interface is triggered by writing to either the 256 byte address window or reading from the address window (see STAA/LDAA 0/1 in the next figure). Depending on which address window is used a blocking or a non-blocking transaction is performed. The address for the transaction is the 8-bit wide window relative address. The data width of the CPU read or write instructions determines if 8-bit or 16-bit wide data are transferred. There is always only one transaction active. Figure 57 shows the various types of transactions explained in more detail below.

For all 16-bit read/write accesses of the CPU the addresses are assigned according the big-endian model:

word [15:8]: addr word[7:0]: addr+1

addr: byte-address (8 bit wide) inside the blocking or non-blocking window, as provided by the CPU and transferred to the D2D target
word: CPU data, to be transferred from/to the D2D target

The application must care for the stretched CPU cycles (limited by the TIMOUT value, caused by blocking or consecutive accesses), which could affect time limits, including COP (computer operates properly) supervision. The stretched CPU cycles cause the “CPU halted” phases (see Figure 57).

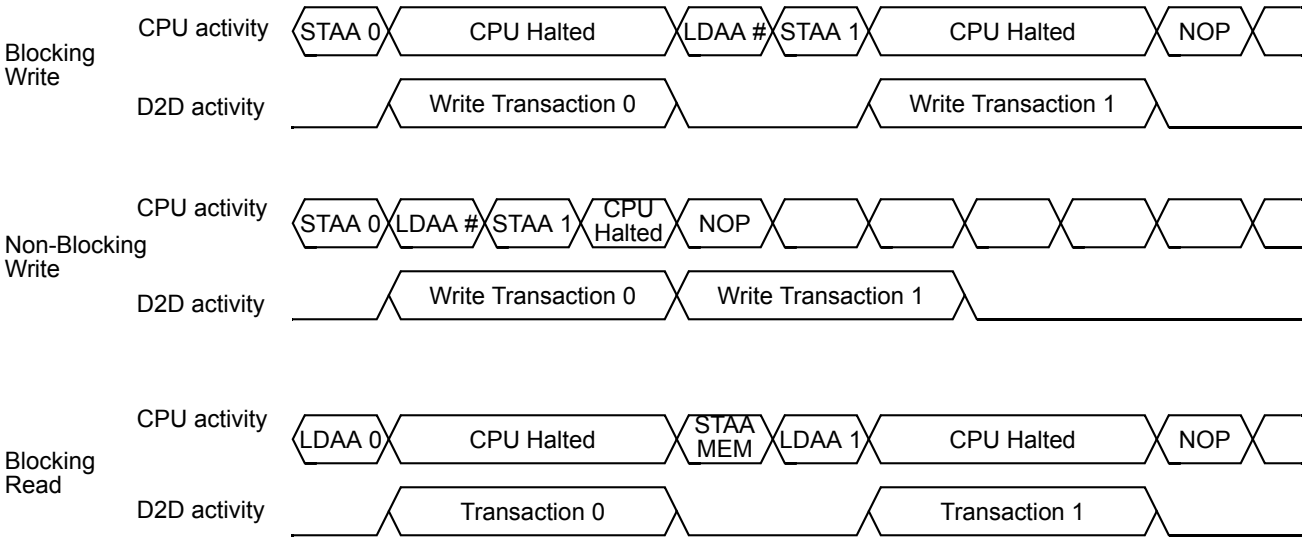


Figure 57. Blocking and non-blocking transfers.

6.6.1.7.2.1 Blocking writes

When writing to the address window associated with blocking transactions, the CPU is held until the transaction is completed, before completing the instruction. Figure 57 shows the behavior of the CPU for a blocking write transaction shown in the following example.

```
STAABLK_WINDOW+OFFS0; WRITE0 8-bit as a blocking transaction
LDAA#BYTE1
STAABLK_WINDOW+OFFS1 ; WRITE1 is executed after WRITE0 transaction is completed
NOP
```

Blocking writes should be used when clearing interrupt flags located in the target or other writes which require that the operation at the target is completed before proceeding with the CPU instruction stream.

6.6.1.7.2.2 Non-blocking writes

When writing to the address window associated with non-blocking transactions, the CPU can continue before the transaction is completed. However if there was a transaction ongoing when doing the 2nd write the CPU is held until the first one is completed, before executing the 2nd one. Figure 57 shows the behavior of the CPU for a blocking write transaction shown in the following example.

```
STAANONBLK_WINDOW+OFFS0; write 8-bit as a non blocking transaction
LDAA#BYTE1; load next byte
STAANONBLK_WINDOW+OFFS1; executed right after the first
NOP
```

As the figure illustrates non-blocking writes have a performance advantage, but care must be taken that the following instructions are not affected by the change in the target caused by the previous transaction.

6.6.1.7.2.3 Blocking read

When reading from the address window associated with blocking transactions, the CPU is held until the data is returned from the target, before completing the instruction.Figure 57 shows the behavior of the CPU for a blocking read transaction shown in the following example.

```
LDAABLK_WINDOW+OFFS0; Read 8-bit as a blocking transaction
STAAMEM; Store result to local Memory
LDAABLK_WINDOW+OFFS1; Read 8-bit as a blocking transaction
```

6.6.1.7.2.4 Non-blocking read

Read access to the non-blocking window is reserved for future use. When reading from the address window associated with non-blocking writes, the read returns an all 0s data byte or word. This behavior can change in future revisions.

6.6.1.7.2.5 Transfer width

8-bit wide writes or reads are translated into 8-bit wide interface transactions. 16-bit wide, aligned writes or reads are translated into a16-bit wide interface transactions. 16-bit wide, misaligned writes or reads are split up into two consecutive 8-bit transactions with the transaction on the odd address first followed by the transaction on the next higher even address. Due to the much more complex error handling (by the MCU), misaligned 16-bit transfers should be avoided.

6.6.1.7.3 Error conditions and handling faults

Since the S12 CPU (as well as the S08) do not provide a method to abort a transfer once started, the D2DI asserts an D2DERRINT. The ERRIF Flag is set in the D2DSTAT0 register. Depending on the error condition further error flags will be set as described below. The content of the address and data buffers are frozen and all transactions are replaced by an IDLE command, until the error flag is cleared. If an error is detected during the read transaction of a read-modify-write instruction or a non-blocking write transaction was followed by another write or read transaction, the second transaction is cancelled. The CNCLF is set in the D2DSTAT0 register to indicate that a transaction has been cancelled. The D2DERRINT handler can read the address and data buffer register to assess the error situation. Any further transaction will be replaced by IDLE until the ERRIF is cleared.

6.6.1.7.3.1 Missing acknowledge

If the target detects a wrong command it will not send back an acknowledge. The same situation occurs if the acknowledge is corrupted. The D2DI detects this missing acknowledge after the timeout period configured in the TIMOUT parameter of the D2DCTL1 register. In case of a timeout the ERRIF and the TIMEF flags in the D2DSTAT0 register will be set.

6.6.1.7.3.2 Parity error

In the final acknowledge cycle of a transaction the target sends two parity bits. If this parity does not match the parity calculated by the initiator, the ERRIF and the PARF flags in the D2DSTAT0 register will be set. The PAR[1:0] bits contain the parity value received by the D2DI.

6.6.1.7.3.3 Error signal

During the acknowledge cycle the target can signal a target specific error condition. If the D2DI finds the error signal asserted during a transaction, the ERRIF and the TERRF flags in the D2DSTAT0 register will be set.

6.6.1.7.4 Low power mode options

6.6.1.7.4.1 D2DI in run mode

In run mode with the D2D Interface enable (D2DEN) bit in the D2D control register 0 clear, the D2DI system is in a low-power, disabled state. D2D registers remain accessible, but clocks to the core of this module are disabled. On D2D lines the GPIO function is activated.

6.6.1.7.4.2 D2DI in wait mode

D2DI operation in Wait mode depends upon the state of the D2DSWAI bit in D2D control register 0.

If D2DSWAI is clear, the D2DI operates normally when the CPU is in the Wait mode

If D2DSWAI is set and the CPU enters the Wait mode, any pending transmission is completed. When the D2DCLK output is driven low then the clock generation is stopped, all internal clocks to the D2DI module are stopped as well and the module enters a power saving state.

6.6.1.7.4.3 D2DI in stop mode

If the CPU enters the Stop mode, the D2DI shows the same behavior as for the Wait mode with an activated D2DSWAI bit.

6.6.1.7.4.4 Reset

In case of reset any transaction is immediately stopped and the D2DI module is disabled.

6.6.1.7.4.5 Interrupts

The D2DI only originates interrupt requests, when D2DI is enabled (D2DIE bit in D2DCTL0 set). There are two different interrupt requests from the D2D module. The interrupt vector offset and interrupt priority are chip dependent.

6.6.1.7.4.5.1 D2D external interrupt

This is a level sensitive active high external interrupt driven by the D2DINT input. This interrupt is enabled if the D2DIE bit in the D2DCTL1 register is set. The interrupt must be cleared using an target specific clearing sequence. The status of the D2D input pin can be observed by reading the D2DIF bit in the D2DSTAT1 register.

The D2DINIT signal is asserted also in the Wait and Stop mode; it can be used to leave these modes.

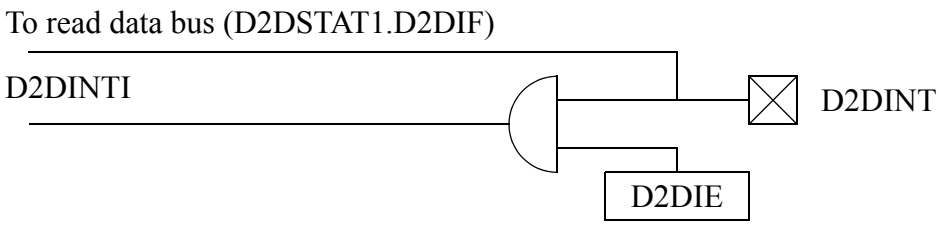


Figure 58. D2D external interrupt scheme

6.6.1.7.4.5.2 D2D error interrupt

Those D2D interface specific interrupts are level sensitive and are all cleared by writing a 1 to the ERRIF flag in the D2DSTAT0 register. This interrupt is not locally maskable and should be tied to the highest possible interrupt level in the system, on an S12 architecture to the XIRQ. See the chapter “Vectors” of the MCU description for details.

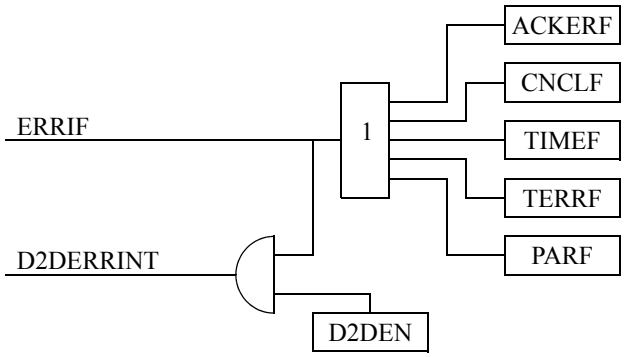


Figure 59. D2D internal interrupts

6.6.1.8 Initialization information

During initialization the transfer width, clock divider and timeout value must be set according to the capabilities of the target device before starting any transaction. See the D2D Target specification for details.

6.6.1.9 Application information

6.6.1.9.1 Entering low power mode

The D2DI module is typically used on a microcontroller along with an analog companion device containing the D2D target interface and supplying the power. Interface specification does not provide special wires for signalling low power modes to the target device. The CPU should determine when it is time to enter one of the above previous modes.The basic flow is as follows:

1. CPU determines there is no more work pending.
2. CPU writes a byte to a register on the analog die using blocking write configuring which mode to enter.
3. Analog die acknowledges that write sending back an acknowledge symbol on the interface.
4. CPU executes WAIT or STOP command.
5. Analog die can enter low power mode - (S12 needs some more cycles to stack data!)

```
; Example shows S12 code
SEI; disable interrupts during test
; check is there is work pending?
; if yes, branch off and re-enable interrupt
; else
LDAA#STOP_ENTRY
STAAMODE_REG; store to the analog die mode reg (use blocking write here)
CLI; re-enable right before the STOP instruction
STOP; stack and turn off all clocks inc. interface clock
```

For wake-up from STOP the basic flow is as follows:

1. Analog die detects a wake-up condition e.g. on a switch input or start bit of a LIN message.
2. Analog die exits voltage regulator low power mode.
3. Analog die asserts the interrupt signal D2DINT.
4. CPU starts clock generation.
5. CPU enters interrupt handler routine.
6. CPU services interrupt and acknowledges the source on the analog die.

Note

Entering Stop mode or Wait mode with D2DSWAI asserted the clock will complete the high duty cycle portion and settle at low level.

6.6.1.9.2 Access to ACQ and COMP registers

The user has to be aware that the acquisition channel is clocked by D2DFCLK. This means that the channel settings provided by the configuration registers become valid after a maximum of 1 cycle of D2DFCLK, which is typically 2.0 µs. As accesses to the registers over the internal bus system are done with a higher speed (clocked by D2DCLK), the software needs to take into account the appropriate wait time before the settings become valid. This is especially important for subsequent accesses to the same register.

6.6.2 Die to die interface - target

The D2D Interface is the bus interface to the Microcontroller. Access to the MM9Z1_638 analog die is controlled by the D2D Interface module. This section describes the functionality of the die-to-die target block (D2D).

6.6.2.1 Overview

The D2D is the target for a data transfer from the target to the initiator (MCU). The initiator provides a set of configuration registers and two memory mapped 256 Byte address windows. When writing to a window, a transaction is initiated sending a write command, followed by an 8-bit address, and the data byte or word is received from the initiator. When reading from a window, a transaction is received with the read command, followed by an 8-bit address. The target then responds with the data. The basic idea is that a peripheral located on the MM9Z1_638 analog die, can be addressed like an on-chip peripheral.

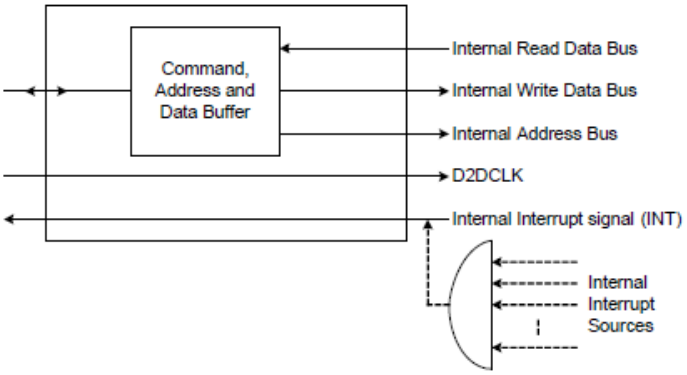


Figure 60. Die to die interface

6.6.2.1.1 Features

- software transparent register access to peripherals on the MM9Z1_638 analog die
- 256 Byte address window
- supports blocking read or write as well as non-blocking write transactions
- 8 bit physical bus width
- automatic synchronization of the target when initiator starts driving the interface clock
- generates transaction and error status as well as EOT acknowledge
- providing single interrupt interface to D2D Initiator

6.6.2.2 Mode of operation

The D2D module is disabled in Sleep and stop Mode (VDDH = 0 V).

In Stop mode, the D2DINT signal is used to wake-up a stopped MCU after re-enabling (VDDH=2.5V) the D2D interface. The MCU should not wake-up while the MM9Z1_638 analog die is in Stop mode (the only way to recover is to generate a Reset / LVR due to limited current capability of VDDXR). See Analog die - power, clock and resets - PCR.

6.6.2.2.1 Normal mode

While in Normal mode, D2DCLK acts as an input only with pull present. D2D[7:0] operates as input/output with a pull-down always present. D2DINT acts as an output only.

6.6.2.2.2 Sleep mode / stop mode

While in Sleep mode, all Interface data pins are pulled down to DGND to reduce power consumption.

6.7 Interrupt module (S12ZINTV0 + IRQ)

6.7.1 Interrupt (S12ZINTV0)

6.7.1.1 Introduction

The INT module decodes the priority of all system exception requests and provides the applicable vector for processing the exception to the CPU. The INT module supports:

- I-bit and X-bit maskable interrupt requests
- One non-maskable unimplemented page1 op-code trap
- One non-maskable unimplemented page2 op-code trap
- One non-maskable software interrupt (SWI)
- One non-maskable system call interrupt (SYS)
- One non-maskable machine exception vector request
- One spurious interrupt vector request
- One system reset vector request

Each of the I-bit maskable interrupt requests can be assigned to one of seven priority levels supporting a flexible priority scheme. The priority scheme can be used to implement nested interrupt capability where interrupts from a lower level are automatically blocked if a higher level interrupt is being processed.

6.7.1.2 Glossary

The following terms and abbreviations are used in the document.

Table 278. Terminology

Term	Meaning
CCW	Condition Code Register (in the S12Z CPU)
DMA	Direct Memory Access
INT	Interrupt
IPL	Interrupt Processing Level
ISR	Interrupt Service Routine
MCU	Micro-Controller Unit
IRQ	refers to the interrupt request associated with the $\overline{\text{IRQ}}$ pin
XIRQ	refers to the interrupt request associated with the $\overline{\text{XIRQ}}$ pin

6.7.1.3 Features

- Interrupt vector base register (IVBR)
- One system reset vector (at address 0xFFFFFC).
- One non-maskable unimplemented page1 op-code trap (SPARE) vector (at address vector base (215) + 0x0001F8).
- One non-maskable unimplemented page2 op-code trap (TRAP) vector (at address vector base (215) + 0x0001F4).
- One non-maskable software interrupt request (SWI) vector (at address vector base (215) + 0x0001F0).
- One non-maskable system call interrupt request (SYS) vector (at address vector base (215) + 0x00001EC).
- One non-maskable machine exception vector request (at address vector base (215) + 0x0001E8).
- One spurious interrupt vector (at address vector base (215) + 0x0001DC).
- One X-bit maskable interrupt vector request associated with $\overline{\text{XIRQ}}$ (at address vector base (215) + 0x0001D8).
- One I-bit maskable interrupt vector request associated with $\overline{\text{IRQ}}$ (at address vector base (215) + 0x0001D4).
- up to 113 additional I-bit maskable interrupt vector requests (at addresses vector base (215) + 0x000010… vector base + 0x0001D0).
- Each I-bit maskable interrupt request has a configurable priority level.
- I-bit maskable interrupts can be nested, depending on their priority levels.
- Wakes up the system from stop or wait mode when an appropriate interrupt request occurs or whenever $\overline{\text{XIRQ}}$ is asserted, even if X interrupt is masked.

Notes:
215.The vector base is a 24-bit address which is accumulated from the contents of the interrupt vector base register (IVBR, used as the upper 15 bits of the address) and 0x000 (used as the lower 9 bits of the address).

6.7.1.4 Modes of operation

- Run mode

This is the basic mode of operation.

- Wait mode

In wait mode, the INT module is capable of waking up the CPU if an eligible CPU exception occurs. Refer to Wake-up from stop or wait mode" for details.

- Stop mode

In stop mode, the INT module is capable of waking up the CPU if an eligible CPU exception occurs. Refer to Wake-up from stop or wait mode" for details.

6.7.1.5 Block diagram

Figure 61 shows a block diagram of the INT module.

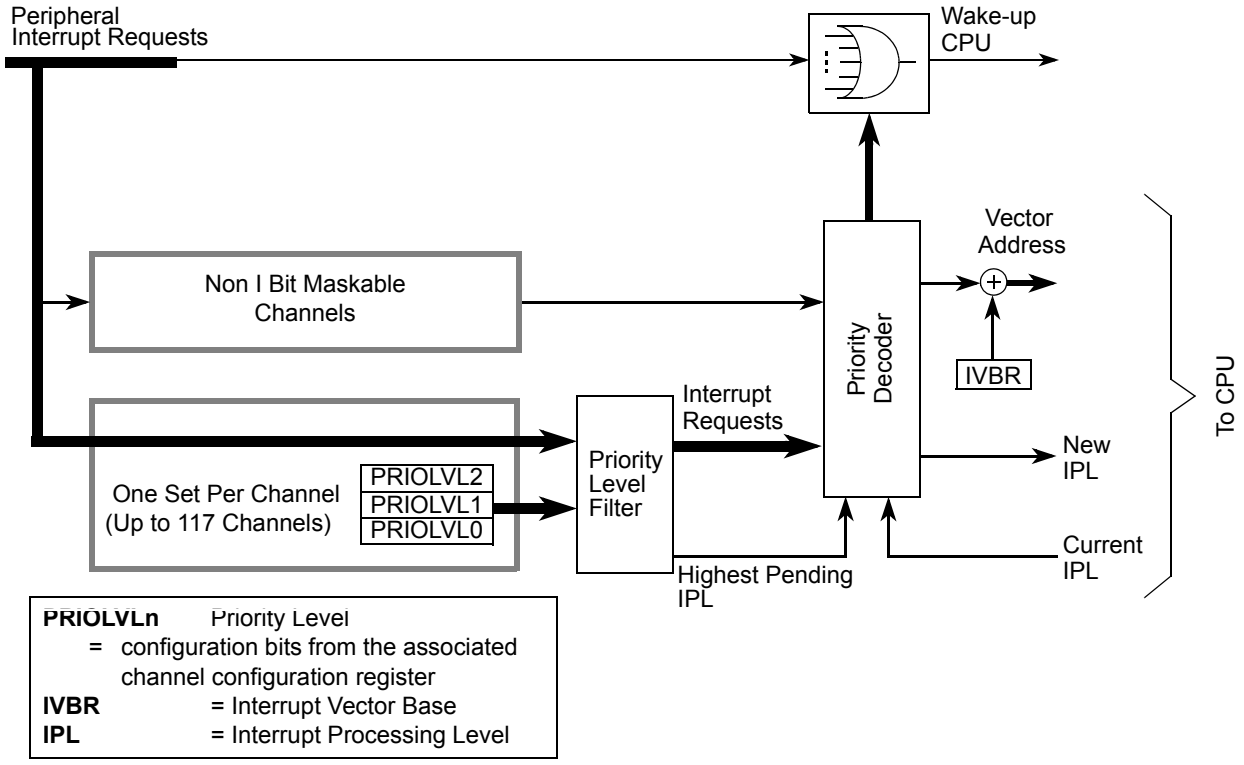


Figure 61. INT block diagram

6.7.1.6 External signal description

The INT module has no external signals.

6.7.1.7 Memory map and register definition

This section provides a detailed description of all registers accessible in the INT module.

6.7.1.7.1 Module memory map

Table 279 gives an overview over all INT module registers.

Table 279. INT memory map

Address	Use	Access
0x000010–0x000011	Interrupt Vector Base Register (IVBR)	R/W
0x000012–0x000016	RESERVED	—
0x000017	Interrupt Request Configuration Address Register (INT_CFADDR)	R/W
0x000018	Interrupt Request Configuration Data Register 0 (INT_CFDATA0)	R/W
0x000019	Interrupt Request Configuration Data Register 1 (INT_CFDATA1)	R/W
0x00001A	Interrupt Request Configuration Data Register 2 (INT_CFDATA2	R/W
0x00001B	Interrupt Request Configuration Data Register 3 (INT_CFDATA3)	R/W
0x00001C	Interrupt Request Configuration Data Register 4 (INT_CFDATA4)	R/W
0x00001D	Interrupt Request Configuration Data Register 5 (INT_CFDATA5)	R/W
0x00001E	Interrupt Request Configuration Data Register 6 (INT_CFDATA6)	R/W
0x00001F	Interrupt Request Configuration Data Register 7 (INT_CFDATA7)	R/W

6.7.1.7.2 Register descriptions

This section describes in address order all the INT module registers and their individual bits.

Table 280. INT Register Summary

Address	Register name		Bit 7	6	5	4	3	2	1	Bit 0
0x000010	IVBR	R	IVB_ADDR[15:8]							
		W								
0x000011		R	IVB_ADDR[7:1]							0
		W								
0x000017	INT_CFADDR	R	0	INT_CFADDR[6:3]				0	0	0
		W								
0x000018	INT_CFDATA0	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x000019	INT_CFDATA1	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001A	INT_CFDATA2	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001B	INT_CFDATA3	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001C	INT_CFDATA4	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001D	INT_CFDATA5	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001E	INT_CFDATA6	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
0x00001F	INT_CFDATA7	R	0	0	0	0	0	PRIOLVL[2:0]		
		W								
			= Unimplemented or Reserved							

6.7.1.7.2.1 Interrupt vector base register (IVBR)

Table 281. Interrupt vector base register (IVBR)

Address: 0x000010

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IVB_ADDR[15:1]															0
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Notes:
216.Read: Anytime
Write: Anytime

Table 282. IVBR field descriptions

Field	Description
15–1 IVB_ADDR [15:1]	**Interrupt Vector Base Address Bits** — These bits represent the upper 15 bits of all vector addresses. Out of reset these bits are set to 0xFFFE (i.e., vectors are located at 0xFFFE00–0xFFFFFF). A system reset will initialize the interrupt vector base register with “0xFFFE” before it is used to determine the reset vector address. Therefore, changing the IVBR has no effect on the location of the reset vector (0xFFFFFC–0xFFFFFF).

6.7.1.7.2.2 Interrupt request configuration address register (INT_CFADDR)

Table 283. Interrupt configuration address register (INT_CFADDR)

Address: 0x000017

	7	6	5	4	3	2	1	0
R	0	INT_CFADDR[6:3]				0	0	0
W								
Reset	0	0	0	0	1	0	0	0

Notes:
217.Read: Anytime
Write: Anytime

Table 284. INT_CFADDR field descriptions

Field	Description
6–3 INT_CFADDR[6:3]	**Interrupt Request Configuration Data Register Select Bits** — These bits determine which of the 128 configuration data registers are accessible in the 8 register window at INT_CFDATA0–7. The hexadecimal value written to this register corresponds to the upper 4 bits of the vector number (multiply with 4 to get the vector address offset). If, for example, the value 0x70 is written to this register, the configuration data register block for the 8 interrupt vector requests starting with vector at address (vector base + (0x70*4 = 0x0001C0)) is selected and can be accessed as INT_CFDATA0–7.

6.7.1.7.2.3 Interrupt request configuration data registers (INT_CFDATA0–7)

The eight register window visible at addresses INT_CFDATA0–7 contains the configuration data for the block of eight interrupt requests (out of 128) selected by the interrupt configuration address register (INT_CFADDR) in ascending order. INT_CFDATA0 represents the interrupt configuration data register of the vector with the lowest address in this block, while INT_CFDATA7 represents the interrupt configuration data register of the vector with the highest address, respectively.

Table 285. Interrupt request configuration data register 0 (INT_CFDATA0)

Address: 0x000018

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	1 (218)

= Unimplemented or Reserved

Notes:
218.Refer to the notes following the PRIOLVL[2:0] description in Table 293.

Table 286. Interrupt request configuration data register 1 (INT_CFDATA1)

Address: 0x000019

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	1 (219)

= Unimplemented or Reserved

Notes:
219.Refer to the notes following the PRIOLVL[2:0] description in Table 293.

Table 287. Interrupt request configuration data register 2 (INT_CFDATA2)

Address: 0x00001A

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	1 (220)

= Unimplemented or Reserved

Notes:
220.Refer to the notes following the PRIOLVL[2:0] description in Table 293.

Table 288. Interrupt request configuration data register 3 (INT_CFDATA3)

Address: 0x00001B

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	1 (221)

= Unimplemented or Reserved

Notes:
221.Refer to the notes following the PRIOLVL[2:0] description in Table 293.

Table 289. Interrupt request configuration data register 4 (INT_CFDATA4)

Address: 0x00001C

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	1 (222)

= Unimplemented or Reserved

Notes:
222.Refer to the notes following the PRIOLVL[2:0] description in Table 293.

Table 290. Interrupt request configuration data register 5 (INT_CFDATA5)

Address: 0x00001D

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	1 (223)

= Unimplemented or Reserved

Notes:
223.Refer to the notes following the PRIOLVL[2:0] description in Table 293.

Table 291. Interrupt request configuration data register 6 (INT_CFDATA6)

Address: 0x00001E

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	1 (224)

= Unimplemented or Reserved

Notes:
224.Refer to the notes following the PRIOLVL[2:0] description in Table 293.

Table 292. Interrupt request configuration data register 7 (INT_CFDATA7)

Address: 0x00001F

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	PRIOLVL[2:0]		
W								
Reset	0	0	0	0	0	0	0	1 (225)

= Unimplemented or Reserved

Notes:
225.Refer to the notes following the PRIOLVL[2:0] description in Table 293.
226.Read: Anytime
Write: Anytime

Table 293. INT_CFDATA0–7 field descriptions

Field	Description
2–0 PRIOLVL[2:0]	**Interrupt Request Priority Level Bits** — The PRIOLVL[2:0] bits configure the interrupt request priority level of the associated interrupt request. Out of reset all interrupt requests are enabled at the lowest active level (“1”). Refer to Table 294 for available interrupt request priority levels. **Note:** Write accesses to configuration data registers of unused interrupt channels are ignored and read accesses return all 0s. For information about what interrupt channels are used in a specific MCU, refer to the Device Reference Manual for that MCU. **Note:** When non I-bit maskable request vectors are selected, writes to the corresponding INT_CFDATA registers are ignored and read accesses return all 0s. The corresponding vectors do not have configuration data registers associated with them. **Note:** Write accesses to the configuration register for the spurious interrupt vector request (vector base + 0x0001DC) are ignored and read accesses return 0x07 (request is handled by the CPU, PRIOLVL = 7).

Table 294. Interrupt priority levels

Priority	PRIOLVL2	PRIOLVL1	PRIOLVL0	Meaning
	0	0	0	Interrupt request is disabled
low	0	0	1	Priority level 1
	0	1	0	Priority level 2
	0	1	1	Priority level 3
	1	0	0	Priority level 4
	1	0	1	Priority level 5
	1	1	0	Priority level 6
high	1	1	1	Priority level 7

6.7.1.8 Functional description

The INT module processes all exception requests to be serviced by the CPU module. These exceptions include interrupt vector requests and reset vector requests. Each of these exception types and their overall priority level is discussed in the following subsections.

6.7.1.8.1 S12Z exception requests

The CPU handles both reset requests and interrupt requests. The INT module contains registers to configure the priority level of each I-bit maskable interrupt request which can be used to implement an interrupt priority scheme. This also includes the possibility to nest interrupt requests. A priority decoder is used to evaluate the relative priority of pending interrupt requests.

6.7.1.8.2 Interrupt prioritization

After system reset all I-bit maskable interrupt requests are configured to be enabled, are set up to be handled by the CPU and have a pre-configured priority level of 1. Exceptions to this rule are the non-maskable interrupt requests and the spurious interrupt vector request at (vector base + 0x0001DC) which cannot be disabled, are always handled by the CPU and have a fixed priority levels. A priority level of 0 effectively disables the associated I-bit maskable interrupt request.

If more than one interrupt request is configured to the same interrupt priority level the interrupt request with the higher vector address wins the prioritization.

The following conditions must be met for an I-bit maskable interrupt request to be processed.

1. The local interrupt enabled bit in the peripheral module must be set.
2. The setup in the configuration register associated with the interrupt request channel must meet the following conditions:
   a) The priority level must be set to non zero.
   b) The priority level must be greater than the current interrupt processing level in the condition code register (CCW) of the CPU (PRIOLVL[2:0] > IPL[2:0]).
3. The I-bit in the condition code register (CCW) of the CPU must be cleared.
4. There is no access violation interrupt request pending.
5. There is no SYS, SWI, SPARE, TRAP, Machine Exception or $\overline{\text{XIRQ}}$ request pending.

Note

All non I-bit maskable interrupt requests always have higher priority than I-bit maskable interrupt requests. If an I-bit maskable interrupt request is interrupted by a non I-bit maskable interrupt request, the currently active interrupt processing level (IPL) remains unaffected. It is possible to nest non I-bit maskable interrupt requests, e.g., by nesting SWI, SYS or TRAP calls.

6.7.1.8.2.1 Interrupt priority stack

The current interrupt processing level (IPL) is stored in the condition code register (CCW) of the CPU. This way the current IPL is automatically pushed to the stack by the standard interrupt stacking procedure. The new IPL is copied to the CCW from the priority level of the highest priority active interrupt request channel which is configured to be handled by the CPU. The copying takes place when the interrupt vector is fetched. The previous IPL is automatically restored from the stack by executing the RTI instruction.

6.7.1.8.3 Priority decoder

The INT module contains a priority decoder to determine the relative priority for all interrupt requests pending for the CPU.

A CPU interrupt vector is not supplied until the CPU requests it. Therefore, it is possible that a higher priority interrupt request could override the original exception which caused the CPU to request the vector. In this case, the CPU will receive the highest priority vector and the system will process this exception first instead of the original request.

If the interrupt source is unknown (for example, in the case where an interrupt request becomes inactive after the interrupt has been recognized, but prior to the vector request), the vector address supplied to the CPU defaults to that of the spurious interrupt vector.

Note

Care must be taken to ensure that all exception requests remain active until the system begins execution of the applicable service routine; otherwise, the exception request may not get processed at all or the result may be a spurious interrupt request (vector at address (vector base + 0x0001DC)).

6.7.1.8.4 Reset exception requests

The INT module supports one system reset exception request. The different reset types are mapped to this vector (for details refer to the Clock and Power Management Unit module (CPMU)):

1. Pin reset
2. Power-on reset
3. Low-voltage reset
4. Clock monitor reset request
5. COP watchdog reset request

6.7.1.8.5 Exception priority

The priority (from highest to lowest) and address of all exception vectors issued by the INT module upon request by the CPU are shown in Table 295. Generally, all non-maskable interrupts have higher priorities than maskable interrupts. Note that between the four software interrupts (Unimplemented op-code trap page1/page2 requests, SWI request, SYS request) there is no real priority defined, since they cannot occur simultaneously (the S12Z CPU executes one instruction at a time).

Table 295. Exception vector map and priority

Vector Address [227]	Source
0xFFFFFC	Pin reset, power-on reset, low-voltage reset, clock monitor reset, COP watchdog reset
(Vector base + 0x0001F8)	Unimplemented page1 op-code trap (SPARE) vector request
(Vector base + 0x0001F4)	Unimplemented page2 op-code trap (TRAP) vector request
(Vector base + 0x0001F0)	Software interrupt instruction (SWI) vector request
(Vector base + 0x0001EC)	System call interrupt instruction (SYS) vector request
(Vector base + 0x0001E8)	Machine exception vector request
(Vector base + 0x0001E4)	Reserved
(Vector base + 0x0001E0)	Reserved
(Vector base + 0x0001DC)	Spurious interrupt
(Vector base + 0x0001D8)	$\overline{\text{XIRQ}}$ interrupt request
(Vector base + 0x0001D4)	$\overline{\text{IRQ}}$ interrupt request
(Vector base + 0x000010 … Vector base + 0x0001D0)	Device specific I-bit maskable interrupt sources (priority determined by the associated configuration registers, in descending order)

Notes:
227.24 bits vector address based

6.7.1.8.6 Interrupt vector table layout

The interrupt vector table contains 128 entries, each 32 bits (4 bytes) wide. Each entry contains a 24-bit address (3 bytes) which is stored in the 3 low-significant bytes of the entry. The content of the most significant byte of a vector-table entry is ignored. Table 296 illustrates the vector table entry format.

Table 296. Interrupt vector table entry

Bits	[31:24]	[23:0]
	(unused)	ISR Address

6.7.1.9 Initialization/application information

6.7.1.9.1 Initialization

After system reset, software should:

- Initialize the interrupt vector base register if the interrupt vector table is not located at the default location (0xFFFE00–0xFFFFFB).
- Initialize the interrupt processing level configuration data registers (INT_CFADDR, INT_CFDATA0–7) for all interrupt vector requests with the desired priority levels. It might be a good idea to disable unused interrupt requests.
- Enable I-bit maskable interrupts by clearing the I-bit in the CCW.
- Enable the X-bit maskable interrupt by clearing the X-bit in the CCW (if required).

6.7.1.9.2 Interrupt nesting

The interrupt request priority level scheme makes it possible to implement priority based interrupt request nesting for the I-bit maskable interrupt requests.

- I-bit maskable interrupt requests can be interrupted by an interrupt request with a higher priority, so that there can be up to seven nested I-bit maskable interrupt requests at a time (refer to Figure 62 for an example using up to three nested interrupt requests).

I-bit maskable interrupt requests cannot be interrupted by other I-bit maskable interrupt requests per default. To make an interrupt service routine (ISR) interruptible, the ISR must explicitly clear the I-bit in the CCW (CLI). After clearing the I-bit, I-bit maskable interrupt requests with higher priority can interrupt the current ISR.

An ISR of an interruptible I-bit maskable interrupt request could basically look like this:

- Service interrupt, e.g., clear interrupt flags, copy data, etc.
- Clear I-bit in the CCW by executing the CPU instruction CLI (thus allowing interrupt requests with higher priority)
- Process data
- Return from interrupt by executing the instruction RTI

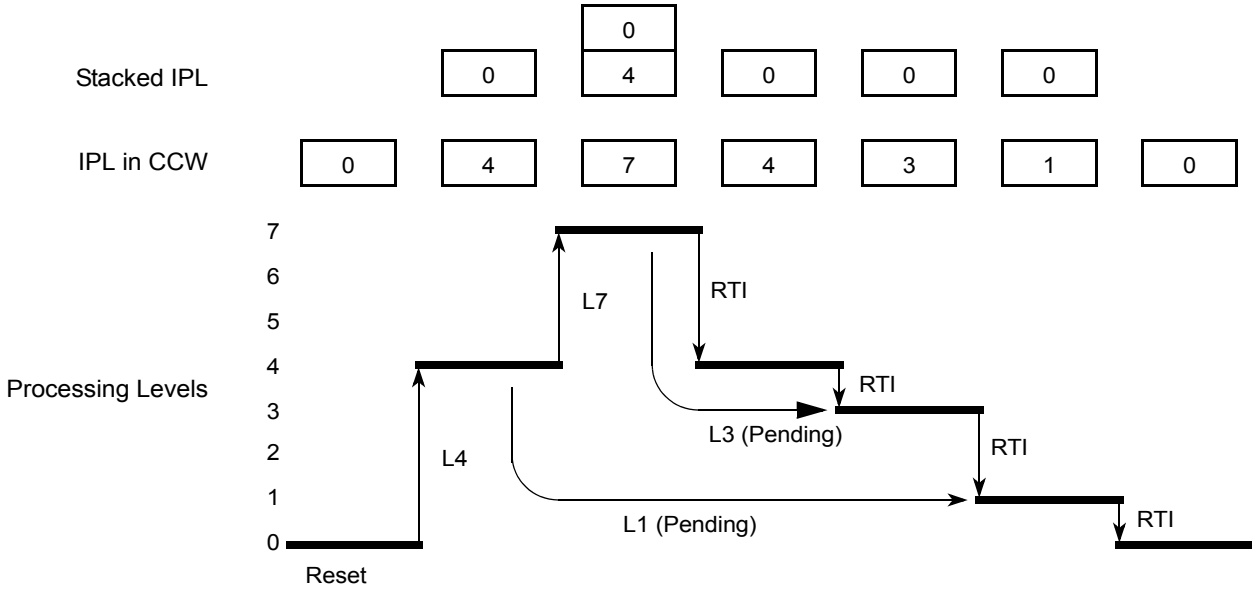


Figure 62. Interrupt processing example

6.7.1.9.3 Wake-up from stop or wait mode

6.7.1.9.3.1 CPU wake-up from stop or wait mode

Every I-bit maskable interrupt request which is configured to be handled by the CPU is capable of waking the MCU from Stop or Wait mode. Additionally machine exceptions can wake-up the MCU from Stop or Wait mode.

To determine whether an I-bit maskable interrupts is qualified to wake-up the CPU, the same settings as in Normal Run mode are applied during Stop or Wait mode:

- If the I-bit in the CCW is set, all I-bit maskable interrupts are masked from waking up the MCU.
- An I-bit maskable interrupt is ignored if it is configured to a priority level below or equal to the current IPL in CCW.

The X-bit maskable interrupt request can wake-up the MCU from Stop or Wait mode at anytime, even if the X-bit in CCW is set If the X-bit maskable interrupt request is used to wake-up the MCU with the X-bit in the CCW set, the associated ISR is not called. The CPU then resumes program execution with the instruction following the WAIT or STOP instruction. This feature works following the same rules like any interrupt request, i.e. care must be taken that the X-bit maskable interrupt request used for wake-up remains active, at least until the system begins execution of the instruction following the WAIT or STOP instruction; otherwise, wake-up may not occur.

6.7.2 Interrupt module - IRQ

6.7.2.1 Introduction

Several interrupt sources are implemented on the analog die to indicate important system conditions. Those Interrupt events are signalized via the D2DINT signal to the microcontroller. See Interrupt (S12ZINTV0)

6.7.2.2 Interrupt source identification

Once an Interrupt is signalized, there are two options to identify the corresponding source(s).

Note

The following Interrupt source registers (Interrupt Source Mirror and Interrupt Vector Emulation by Priority) are indicators only. After identifying the interrupt source, the acknowledgement of the interrupt has to be performed in the corresponding block.

6.7.2.2.1 Interrupt source mirror

All Interrupt sources in the MM9Z1_638 analog die are mirrored to a special Interrupt Source Register (INT_SRC). This register is read only and will indicate all currently pending Interrupts. Reading this register will not acknowledge any interrupt. An additional D2D access is necessary to serve the specific module.

6.7.2.2.2 Interrupt vector emulation by priority

To allow a vector based interrupt handling by the MCU, the number of the highest prioritized interrupt pending is returned in the Interrupt Vector Register (INT_VECT). Reading this register will not acknowledge an interrupt. An additional D2D access is necessary to serve the specific module.

6.7.2.3 Interrupt global mask

The Global Interrupt mask registers INT_MSK (hi) and INT_MSK (lo) are implemented to allow a global enable / disable of all analog die Interrupt sources. The individual blocks mask registers should be used to control the individual sources.

6.7.2.4 Interrupt sources

The following Interrupt sources are implemented on the analog die.

Table 297. Interrupt sources

IRQ	Description
UVI	Undervoltage Interrupt (or wake-up from Cranking mode)
HTI	High Temperature Interrupt
LFI	LIN Driver Overtemperature or TxD Dominant Timeout Interrupt
CH0	TIM Channel 0 Interrupt
CH1	TIM Channel 1 Interrupt
CH2	TIM Channel 2 Interrupt
CH3	TIM Channel 3 Interrupt
TOV	TIM Timer Overflow Interrupt
ERR	SCI Error Interrupt
TX	SCI Transmit Interrupt
RX	SCI Receive Interrupt
CVMI	Current / Voltage Measurement Interrupt
LTC	Lifetime Counter Interrupt
CAL	Calibration Request Interrupt

6.7.2.4.1 Undervoltage Interrupt (UVI)

This maskable interrupt signalizes a undervoltage condition on the VSUP supply input.

Acknowledge the interrupt by writing a 1 into the UVF Bit in the PCR Status Register (PCR_SR (hi)). The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to vanish and occur again. The UVF Bit represents the current condition, and might not be set after an interrupt was signalized by the interrupt source registers.

See Clock, reset, and power management unit (S12ZCPMU + PCR) for details on the PCR Status Register (PCR_SR (hi)), including masking information.

Note

The undervoltage interrupt is not active in devices with the Cranking mode enabled. For those devices, the undervoltage threshold is used to enable the high precision low voltage threshold during Stop/Sleep mode.

Once the device wakes up from Cranking mode, the UVI flag is indicating the wake-up source.

6.7.2.4.2 High temperature interrupt (HTI)

This maskable interrupt signalizes a high temperature condition on the analog die. The sensing element is located close to the major thermal contributors, the system voltage regulators.

Acknowledge the interrupt by writing a 1 into the HTF Bit in the PCR Status Register (PCR_SR (hi)). The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to vanish and occur again. The HTF Bit represents the current condition and might not be set after an interrupt was signalized by the interrupt source registers.

See Clock, reset, and power management unit (S12ZCPMU + PCR) for details on the PCR Status Register (PCR_SR (hi)), including masking information.

6.7.2.4.3 LIN driver overtemperature or TxD dominant timeout interrupt (LFI)

Acknowledge the interrupt by reading the LIN Register - LINR. The flag cannot be cleared as long as the condition is present. To issue a new interrupt, the condition has to cease and then reoccur. See LIN for details on the LIN Register, including masking information.

6.7.2.4.4 TIM channel 0 interrupt (CH0)

See Basic timer module - TIM (TIM16B4C).

6.7.2.4.5 TIM channel 1 interrupt (CH1)

See Basic timer module - TIM (TIM16B4C).

6.7.2.4.6 TIM channel 2 interrupt (CH2)

See Basic timer module - TIM (TIM16B4C).

6.7.2.4.7 TIM channel 3 interrupt (CH3)

See Basic timer module - TIM (TIM16B4C).

6.7.2.4.8 TIM timer overflow interrupt (TOV)

See Basic timer module - TIM (TIM16B4C).

6.7.2.4.9 SCI error interrupt (ERR)

See Basic timer module - TIM (TIM16B4C).

6.7.2.4.10 SCI transmit interrupt (TX)

See Basic timer module - TIM (TIM16B4C).

6.7.2.4.11 SCI receive interrupt (RX)

See Basic timer module - TIM (TIM16B4C).

6.7.2.4.12 Current / voltage measurement interrupt (CVMI)

Indicates the current or voltage measurement finished (VM or CM bit set). See Channel acquisition.

6.7.2.4.13 Life time counter interrupt (LTC)

In case a Life Time Counter overflow occurs with the corresponding interrupt enabled, the LTC interrupt is issued. See Life time counter (LTC).

6.7.2.4.14 Calibration request interrupt (CAL)

Once a request for re-calibration is present (Temperature out of pre-set range), the Calibration Interrupt is issued. See full documentation on the interrupt source in Channel acquisition.

6.7.2.5 IRQ - memory map and registers

6.7.2.5.1 Overview

This section provides a detailed description of the memory map and registers.

6.7.2.5.2 Module memory map

The memory map for the IRQ module is given in Table 60

Table 298. Module memory map

Offset[228]	Name		7	6	5	4	3	2	1	0
0x08	INT_SRC (hi)	R	TOV	CH3	CH2	CH1	CH0	LFI	HTI	UVI
	Interrupt source register	W								
0x09	INT_SRC (lo)	R	0	0	CAL	LTC	CVMI	RX	TX	ERR
	Interrupt source register	W								

Table 298. Module memory map

Offset[228]	Name		7	6	5	4	3	2	1	0
0x0A	INT_VECT	R	0	0	0	0	IRQ[3:0]			
	Interrupt vector register	W								
0x0B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0C	INT_MSK (hi)	R	TOVM	CH3M	CH2M	CH1M	CH0M	LFIM	HTIM	UVIM
	Interrupt mask register	W								
0x0D	INT_MSK (lo)	R	0	0	CALM	LTCM	CVMM	RXM	TXM	ERRM
	Interrupt mask register	W								

Notes:
228.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

6.7.2.5.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

6.7.2.5.3.1 Interrupt source register (INT_SRC (hi))

Table 299. Interrupt source register (INT_SRC (hi))

Offset[229]	0x08					Access: User read		
	7	6	5	4	3	2	1	0
R	TOV	CH3	CH2	CH1	CH0	LFI	HTI	UVI
W								

Notes:
229.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 300. Interrupt source register (INT_SRC (hi)) - register field descriptions

Field	Description
7 TOV	**TIM16B4C** - Timer overflow interrupt status 0 - No timer overflow interrupt pending 1 - Timer overflow interrupt pending
6 CH3	**TIM16B4C** - TIM channel 3 interrupt status 0 - No channel 3 interrupt pending 1 - Channel 3 interrupt pending
5 CH2	**TIM16B4C** - TIM channel 2 interrupt status 0 - No channel 2 interrupt pending 1 - Channel 2 interrupt pending
4 CH1	**TIM16B4C** - TIM channel 1 interrupt status 0 - No channel 1 interrupt pending 1 - Channel 1 interrupt pending
3 CH0	**TIM16B4C** - TIM channel 0 interrupt status 0 - No channel 0 interrupt pending 1 - Channel 0 interrupt pending
2 LFI	**LIN Driver overtemperature or TxD dominant timeout interrupt status** 0 - No LIN driver overtemperature or TxD dominant timeout interrupt 1 - LIN driver overtemperature or TxD dominant timeout interrupt

Table 300. Interrupt source register (INT_SRC (hi)) - register field descriptions (continued)

Field	Description
1 HTI	**High temperature interrupt status** 0 - No high temperature interrupt pending 1 - High temperature interrupt pending
0 UVI	**Undervoltage interrupt pending or wake-up from Cranking mode status** 0 - No undervoltage Interrupt pending or wake-up from Cranking mode 1 - Undervoltage interrupt pending or wake-up from Cranking mode

6.7.2.5.4 Interrupt source register (INT_SRC (lo))

Table 301. Interrupt source register (INT_SRC (lo))

Offset[(230)]	0x09					Access: User read		
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	0	0	CAL	LTC	CVMI	RX	TX	ERR
W								

Notes:
230.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 302. Interrupt source register (INT_SRC (lo)) - register field descriptions

Field	Description
5 CAL	**Calibration request interrupt status** 0 - No calibration request interrupt pending 1 - Calibration request interrupt pending
4 LTC	**Life time counter interrupt status** 0 - No life time counter interrupt pending 1 - Life time counter interrupt pending
3 CVMI	**Current / Voltage measurement interrupt status** 0 - No Current / Voltage measurement interrupt pending 1 - Current / Voltage measurement interrupt pending
2 RX	**SCI receive interrupt status** 0 - No SCI receive interrupt pending 1 - SCI receive interrupt pending
1 TX	**SCI transmit interrupt status** 0 - No SCI transmit interrupt pending 1 - SCI transmit interrupt pending
0 ERR	**SCI error interrupt status** 0 - No SCI transmit interrupt pending 1 - SCI transmit interrupt pending

6.7.2.5.4.1 Interrupt vector register (INT_VECT)

Table 303. Interrupt vector register (INT_VECT)

Offset[231]	0x0A						Access: User read	
	7	6	5	4	3	2	1	0
R	0	0	0	0	IRQ			
W								

Notes:
231.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 304. Interrupt vector register (INT_VECT) - register field descriptions

Field	Description
4-0 IRQ	Represents the highest prioritized interrupt pending. See 305. If no interrupt is pending, the result will be 0.

Table 305. Interrupt vector / priority

IRQ	Description	IRQ	Priority
-	No interrupt pending or wake-up from Stop mode	0x00	-
UVI	Undervoltage interrupt or wake-up from Cranking mode	0x01	1 (highest)
HTI	High temperature interrupt	0x02	2
LFI	LIN driver overtemperature or TxD dominant timeout interrupt	0x03	3
CH0	TIM channel 0 interrupt	0x04	4
CH1	TIM channel 1 interrupt	0x05	5
CH2	TIM channel 2 interrupt	0x06	6
CH3	TIM channel 3 interrupt	0x07	7
TOV	TIM timer overflow interrupt	0x08	8
ERR	SCI error interrupt	0x09	9
TX	SCI transmit interrupt	0x0A	10
RX	SCI receive interrupt	0x0B	11
CVMI	Acquisition interrupt	0x0C	12
LTC	Life time counter interrupt	0x0D	13
CAL	Calibration request interrupt	0x0E	14 (lowest)

6.7.2.5.4.2 Interrupt mask register (INT_MSK (hi))

Table 306. Interrupt mask register (INT_MSK (hi))

Offset[232]	0x0C						Access: User read/write	
	7	6	5	4	3	2	1	0
R W	TOVM	CH3M	CH2M	CH1M	CH0M	LFIM	HTIM	UVIM
Reset	0	0	0	0	0	0	0	0

Notes:
232.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 307. Interrupt mask register (INT_MSK (hi)) - register field descriptions

Field	Description
7 TOVM	**Timer overflow interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
6 CH3M	**Timer channel 3 interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
5 CH2M	**Timer channel 2 interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
4 CH1M	**Timer channel 1 interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
3 CH0M	**Timer channel 1 interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
2 LFIM	**LIN driver overtemperature or TxD dominant timeout interrupt** 0 - Interrupt enabled 1 - Interrupt disabled
1 HTIM	**High temperature interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
0 UVIM	**Undervoltage interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled

6.7.2.5.4.3 Interrupt mask register (INT_MSK (lo)

)

Table 308. Interrupt mask register (INT_MSK (lo))

Offset[233]			0x0D				Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	0	CALM	LTCM	CVMM	RXM	TXM	ERRM
W								
Reset	0	0	0	0	0	0	0	0

Notes:
233.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 309. Interrupt mask register (INT_MSK (lo)) - register field descriptions

Field	Description
5 CALM	**Calibration request interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
4 LTCM	**Life time counter interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
3 CVMM	**Current / Voltage measurement interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
2 RXM	**SCI receive interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled

Table 309. Interrupt mask register (INT_MSK (lo)) - register field descriptions (continued)

Field	Description
1 TXM	**SCI transmit interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled
0 ERRM	**SCI error interrupt mask** 0 - Interrupt enabled 1 - Interrupt disabled

6.8 ECC generation module (SRAM_ECCV1)

6.8.1 Introduction

The purpose of ECC logic is to detect and correct as much as possible memory data bit errors. These soft errors can occur randomly during operation, mainly generated by alpha radiation. Soft Error means, that only the information inside the memory cell is corrupt, the memory cell itself is not damaged. A write access with correct data solves the issue. If the ECC algorithm is able to correct the data, then the system can use this corrected data without any issues. If the ECC algorithm is able to detect, but not correct the error, then the system is able to ignore the memory read data to avoid system malfunction.

The ECC value is calculated based on an aligned two byte memory data word. The ECC algorithm is able to detect and correct single bit ECC errors. Double bit ECC errors will be detected but the system is not able to correct these errors. This kind of ECC code is called SECDED code. This ECC code requires six additional parity bits for each two byte data word.

6.8.1.1 Features

The SRAM_ECC module provides the ECC logic for the system memory based on a SECDED algorithm. Main features of the SRAM_ECC module:

- SECDED ECC code
  - single bit error detection and correction per two byte data word
  - double bit error detection per two byte data word
- memory initialization function
- byte wide system memory write access
- automatic single bit ECC error correction for read and write accesses
- debug logic to read and write raw use data and ECC values

6.8.2 Memory map and register definition

This section provides a detailed description of all memory and registers for the SRAM_ECC module.

6.8.2.1 Register summary

Table 310 shows the summary of all implemented registers inside the SRAM_ECC module.

Note

Register Address = Module Base Address + Address Offset, where the Module Base Address is defined at the MCU level and the Address Offset is defined at the module level.

Table 310. SRAM_ECC register summary

Address offset register name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 ECCSTAT	R	0	0	0	0	0	0	0	RDY
	W								
0x0001 ECCIE	R	0	0	0	0	0	0	0	SBEEIE
	W								

MM9Z1_638

Table 310. SRAM_ECC register summary (continued)

Address offset register name		Bit 7	6	5	4	3	2	1	Bit 0
0x0002 ECCIF	R	0	0	0	0	0	0	0	SBEEIF
	W								
0x0003 - 0x0006 Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0007 ECCDPTRH	R	DPTR[23:16]							
0x0008 ECCDPTRM	R	DPTR[15:8]							
	W								
0x0009 ECCDPTRL	R	DPTR[7:1]							0
	W								
0x000A - 0x000B Reserved	R	0	0	0	0	0	0	0	0
	W								
0x0000C ECCDDH	R	DDATA[15:8]							
	W								
0x0000D ECCDDL	R	DDATA[7:0]							
	W								
0x000E ECCDE	R	0	0	DECC[5:0]					
	W								
0x000F ECCDCMD	R	ECCDRR	0	0	0	0	0	ECCDW	ECCDR
	W								
		= Unimplemented, Reserved, Read as zero							

6.8.2.2 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field functions follow the register diagrams, in bit order.

6.8.2.2.1 ECC status register (ECCSTAT)

Table 311. ECC status register (ECCSTAT)

Module Base + 0x00000 **Access: User read only (234)**

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	RDY
W								
Reset	0	0	0	0	0	0	0	0

Notes:
234.Read: Anytime
Write: Never

Table 312. ECCSTAT field description

Field	Description
0 RDY	**ECC Ready**— Shows the status of the ECC module. 0 Internal memory initialization is ongoing, access to the memory is disabled 1 Internal memory initialization is done, access to the memory is enabled

6.8.2.2.2 ECC interrupt enable register (ECCIE)

Table 313. ECC interrupt enable register (ECCIE)

Module Base + 0x00001							Access: User read/write (235)
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	SBEEIE
W								
Reset	0	0	0	0	0	0	0	0

Notes:
235.Read: Anytime
Write: Anytime

Table 314. ECCIE field description

Field	Description
0 SBEEIE	**Single bit ECC Error Interrupt Enable** — Enables Single ECC Error interrupt. 0 Interrupt request is disabled 1 Interrupt will be requested whenever SBEEIF is set

6.8.2.2.3 ECC interrupt flag register (ECCIF)

Table 315. ECC interrupt flag register (ECCIF)

Module Base + 0x0002							Access: User read/write (236)
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	SBEEIF
W								
Reset	0	0	0	0	0	0	0	0

Notes:
236.Read: Anytime
Write: Anytime, write 1 to clear

Table 316. ECCIF field description

Field	Description
0 SBEEIF	**Single bit ECC Error Interrupt Flag** — The flag is set to 1 when a single bit ECC error occurs. 0 No occurrences of single bit ECC error since the last clearing of the flag 1 single bit ECC error occurs since the last clearing of the flag

6.8.2.2.4 ECC debug pointer register (ECCDPTRH, ECCDPTRM, ECCDPTRL)

Table 317. ECC debug pointer register (ECCDPTRH, ECCDPTRM, ECCDPTRL)

Module Base + 0x0007							Access: User read/write (237)
	7	6	5	4	3	2	1	0
R W	DPTR[23:16]							
Reset	0	0	0	0	0	0	0	0
Module Base + 0x0008							Access: User read/write	
	7	6	5	4	3	2	1	0

Table 317. ECC debug pointer register (ECCDPTRH, ECCDPTRM, ECCDPTRL) (continued)

R W	DPTR[15:8]							
Reset	0	0	0	0	0	0	0	0
	Module Base + 0x0009				Access: User read/write			
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	DPTR[7:1]							0
Reset	0	0	0	0	0	0	0	0
	= Unimplemented							

Notes:
237.Read: Anytime
Write: Anytime

Table 318. ECCDPTR register field descriptions

Field	Description
DPTR [23:0]	**ECC Debug Pointer** — This register contains the system memory address which will be used for a debug access. Address bits not relevant for SRAM address space are not writeable, so the SW should read back the pointer value to make sure the register contains the intended memory address.

6.8.2.2.5 ECC debug data (ECCDDH, ECCDDL)

Table 319. ECC debug data (ECCDDH, ECCDDL)

	Module Base + 0x000C						**Access: User read/write** (238)	
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	DDATA[15:8]							
Reset	0	0	0	0	0	0	0	0
	Module Base + 0x000D				Access: User read/write			
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	DDATA[7:0]							
Reset	0	0	0	0	0	0	0	0
	= Unimplemented							

Notes:
238.Read: Anytime
Write: Anytime

Table 320. ECCDD register field descriptions

Field	Description
DDATA [15:0]	**ECC Debug Raw Data** — This register contains the raw data which will be written into the system memory during a debug write command or the read data from the debug read command.

6.8.2.2.6 ECC debug ECC (ECCDE)

Table 321. ECC debug ECC (ECCDE)

Module Base + 0x000E	Access: User read/write [239]							
	7	6	5	4	3	2	1	0
R	0	0	DECC[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

Notes:
239.Read: Anytime
Write: Anytime

Table 322. ECCDE field description

Field	Description
5:0 DECC[5:0]	**ECC Debug ECC** — This register contains the raw ECC value which will be written into the system memory during a debug write command or the ECC read value from the debug read command.

6.8.2.2.7 ECC debug command (ECCDCMD)

Table 323. ECC debug command (ECCDCMD)

Module Base + 0x000F	Access: User read/write [240]							
	7	6	5	4	3	2	1	0
R	ECCDRR	0	0	0	0	0	ECCDW	ECCDR
W								
Reset	0	0	0	0	0	0	0	0

Notes:
240.Read: Anytime
Write: Anytime, in special mode only

Table 324. CCDCMD field description

Field	Description
7 ECCDRR	**ECC Disable Read Repair Function** — Write one to this register bit will disable the automatic single bit ECC error repair function during read access, see also ECC debug behavior". 0 Automatic single ECC error repair function is enabled 1 Automatic single ECC error repair function is disabled
1 ECCDW	**ECC Debug Write Command** — Write one to this register bit will perform a debug write access to the system memory. During this access the debug data word (DDATA) and the debug ECC value (DECC) will be written to the system memory address defined by DPTR. If the debug write access is done this bit is cleared. Writing 0 has no effect. It is not possible to set this bit if the previous debug access is ongoing. (ECCDW or ECCDR bit set)
0 ECCDR	**ECC Debug Read Command** — Write one to this register bit will perform a debug read access from the system memory address defined by DPTR. If the debug read access is done this bit is cleared and the raw memory read data are available in register DDATA and the raw ECC value is available in register DECC. Writing 0 has no effect. If the ECCDW and ECCDR bit are set at the same time, then only the ECCDW bit is set and the Debug Write Command is performed. It is not possible to set this bit if the previous debug access is ongoing. (ECCDW or ECCDR bit set)

6.8.3 Functional description

The bus system allows 1, 2, 3, and 4 byte write access to a 4 byte aligned memory address, but the ECC value is generated based on an aligned 2 byte data word. Depending on the access type, the access is separated into different access cycles. Table 325 shows the different access types with the expected number of access cycles and the performed internal operations.

Table 325. Memory access cycles

Access type	ECC error	Access cycle	Internal operation	Memory content	Error indication
2 and 4 byte aligned write access	-	1	write to memory	new data	-
1 or 3 byte write, non-aligned 2 byte write	no	2	read data from the memory write old + new data to the memory	old + new data	-
	single bit	2	read data from the memory write corrected + new data to the memory	corrected + new data	SBEEIF
	double bit	2	read data from the memory ignore write data	unchanged	initiator module is informed
read access	no	1	read from memory	unchanged	-
	single bit	1 (241)	read data from the memory write corrected data back to memory	corrected data	SBEEIF
	double bit	1	read from memory	unchanged	data mark as invalid

Notes:
241.The next back to back read access to the memory will be delayed by one clock cycle

The single bit ECC errors generates an interrupt when enabled. The double bit ECC errors are reported by the SRAM_ECC module, but are handled outside by MCU level. For more information see Memory mapping control (S12ZMMCV1).

6.8.3.1 Aligned 2 and 4 byte memory write access

During an aligned 2 or 4 byte memory write access no ECC check is performed. The internal ECC logic generates the new ECC value based on the write data and writes the data words together with the generated ECC values into the memory.

6.8.3.2 Other memory write access

Other types of write accesses are separated into read-modify-write operation. During the first cycle, the logic reads the data from the memory and performs an ECC check. If no ECC errors was detected then the logic generates the new ECC value based on the read and write data and writes the new data word together with the new ECC value into the memory. If required both 2 byte data words are updated.

If the module detects a single bit ECC error during the read cycle, then the logic generates the new ECC value based on the corrected read and new write read. In the next cycle new data word and the new ECC value are written into the memory. If required both 2 byte data words are updated. The SBEEIF bit is set. Hence the single bit ECC error was corrected by the write access. Figure 63 shows an example of a 2 byte non-aligned memory write access.

If the module detects a double bit ECC error during the read cycle, then the write access to the memory is blocked and the initiator module is informed about the error.

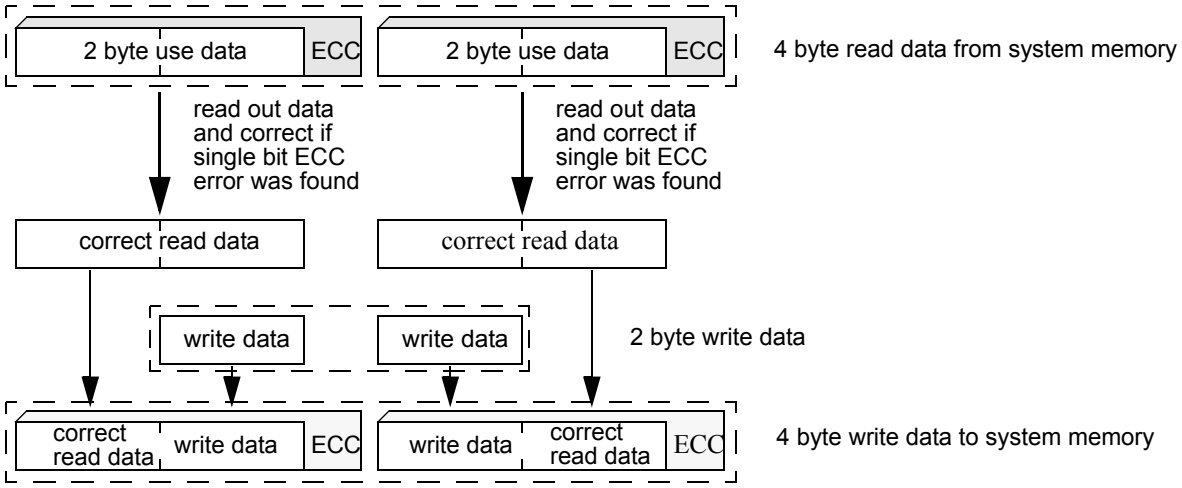


Figure 63. 2 Byte non-aligned write access

6.8.3.3 Memory read access

An ECC check is performed during each memory read access. If the logic detects a single bit ECC error, then the module corrects the data, so that the access initiator module receives correct data. In parallel, the logic writes the corrected data back to the memory, so that this read access repairs the single bit ECC error. This automatic ECC read repair function is disabled by setting the ECCDRR bit.

The SBEEIF flag is set if a single bit ECC error is detected.

The data word is flagged as invalid if the logic detects a double bit ECC error, so the access initiator module can ignore the data.

6.8.3.4 Memory initialization

Memory operation which allows a read before a first write, like the read-modify-write operation of the un-aligned access, requires that the memory contains valid ECC values before the first read-modify-write access is performed to avoid spurious ECC error reporting. The ECC module provides logic to initialize the complete memory content with zero during the power up phase. During the initialization process the access to the memory is disabled and the RDY status bit is cleared. If the initialization process is done, the memory access is possible and the RDY status bit is set.

6.8.3.5 Interrupt handling

This sections describes the interrupts generated by the SRAM_ECC module and their individual sources, Vector addresses and interrupt priority are defined by MCU level.

Table 326. PTU interrupt sources

Module Interrupt Sources	Local Enable
Single bit ECC error	ECCIE[SBEEIE]

6.8.3.6 ECC algorithm

The table below shows the equation for each ECC bit based on the 16 bit data word.

Table 327. ECC calculation

ECC bit	Use data
ECC[0]	~ (^ (data[15:0] & 0x443F))
ECC[1]	~ (^ (data[15:0] & 0x13C7))
ECC[2]	~ (^ (data[15:0] & 0xE1D1))
ECC[3]	~ (^ (data[15:0] & 0xEE60))
ECC[4]	~ (^ (data[15:0] & 0x3E8A))
ECC[5]	~ (^ (data[15:0] & 0x993C))

6.8.3.7 ECC debug behavior

For debug purposes, it is possible to read and write the uncorrected use data and the raw ECC value direct from the memory. For these debug accesses a register interface is available. The debug access is performed with the lowest priority, other memory accesses must be done before the debug access starts. If a debug access is requested during a ongoing memory initialization process, then the debug access is performed if the memory initialization process is done.

If the ECCDRR bit is set, then the automatic single bit ECC error repair function for all read accesses is disabled. In this case, a read access from a system memory location with single bit ECC error will produce correct data, and the single bit ECC error is flagged by the SBEEIF, but the data inside the system memory are unchanged.

By writing wrong ECC values into the system memory the debug access can be used to force single and double bit ECC errors to check the SW error handling.

It is not possible to set the ECCDW or ECCDR bit if the previous debug access is ongoing. (ECCDW or ECCDR bit active) This makes sure that the ECCDD and ECCDE registers contains consistent data. The SW should read out the status of the ECCDW and ECCDR register bit before a new debug access is requested.

6.8.3.7.1 ECC debug memory write access

Writing one to the ECCDW bit performs a debug write access to the memory address defined by register DPTR. During this access, the raw data DDATA and the ECC value DECC are written direct into the system memory. If the debug write access is done, the ECCDW register bit is cleared. The debug write access is always a 2 byte aligned memory access, so that no ECC check is performed and no single or double bit ECC error indication are activated.

6.8.3.7.2 ECC debug memory read access

Writing one to the ECCDR bit performs a debug read access from the memory address defined by register DPTR. If the ECCDR bit is cleared, then the register DDATA contains the uncorrected read data from the memory. The register DECC contains the ECC value read from the memory. Independent of the ECCDRR register bit setting, the debug read access will not perform an automatic ECC repair during read access. During the debug read access, no ECC check is performed, so no single or double bit ECC error indication are activated.

If the ECCDW and the ECCDR bits are set at the same time, then only the debug write access is performed.

6.9 Memory mapping control (S12ZMMCV1)

6.9.1 Introduction

The S12ZMMC module controls the access to all internal memories and peripherals for the S12ZCPU, and the S12ZBDC module. It also provides access to the RAM for ADCs and the PTU module. The S12ZMMC determines the address mapping of the on-chip resources, regulates access priorities and enforces memory protection. Figure 64 shows a block diagram of the S12ZMMC module.

6.9.1.1 Glossary

Table 328. Glossary of terms

Term	Definition
MCU	Microcontroller Unit
CPU	S12Z Central Processing Unit
BDC	S12Z Background Debug Controller
ADC	Analog-to-Digital Converter
PTU	Programmable Trigger Unit
unmapped address range	Address space that is not assigned to a memory
reserved address range	Address space that is reserved for future use cases
illegal access	Memory access, that is not supported or prohibited by the S12ZMMC, e.g. a data store to NVM
access violation	Either an illegal access or an uncorrectable ECC error
byte	8-bit data
word	16-bit data

6.9.1.2 Overview

The S12ZMMC provides access to on-chip memories and peripherals for the S12ZCPU, the S12ZBDC, the PTU, and the ADC. It arbitrates memory accesses and determines all of the MCU memory maps. Furthermore, the S12ZMMC is responsible for selecting the MCUs functional mode.

6.9.1.3 Features

- S12ZMMC mode operation control
- Memory mapping for S12ZCPU and S12ZBDC, PTU, and ADCs
  - Maps peripherals and memories into a 16 Mbyte address space for the S12ZCPU, the S12ZBDC, the PTU, and the ADCs
  - Handles simultaneous accesses to different on-chip resources (NVM, RAM, and peripherals)
- Access violation detection and logging
  - Triggers S12ZCPU machine exceptions upon detection of illegal memory accesses and uncorrectable ECC errors
  - Logs the state of the S12ZCPU and the cause of the access error

6.9.1.4 Modes of operation

6.9.1.4.1 Chip configuration modes

The S12ZMMC determines the chip configuration mode of the device. It captures the state of the MODC pin at reset and provides the ability to switch from Special-single Chip mode to Normal Single Chip mode.

6.9.1.4.2 Power modes

The S12ZMMC module is only active in Run and Wait mode.There is no bus activity in Stop mode.

6.9.1.5 Block diagram

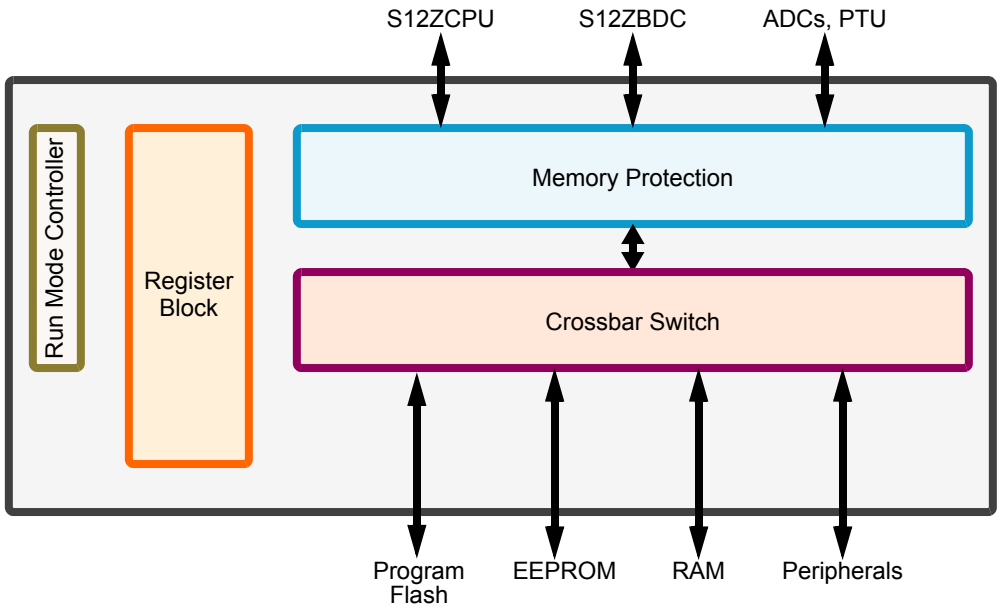


Figure 64. S12ZMMC block diagram

6.9.2 External signal description

The S12ZMMC uses two external pins to determine the devices operating mode: RESET and MODC (Table 329)
See device overview for the mapping of these signals to device pins.

Table 329. External system pins associated with S12ZMMC

Pin Name	Description
$\overline{\text{RESET}}$	External reset signal. The $\overline{\text{RESET}}$ signal is active low.
MODC	This input is captured in bit MODC of the MODE register when the external $\overline{\text{RESET}}$ pin deasserts.

6.9.3 Memory map and register definition

6.9.3.1 Memory map

A summary of the registers associated with the MMC block is shown in Table 330. Detailed descriptions of the registers and bits are given in the subsections that follow.

Table 330. S12ZMMC register summary

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0070	MODE	R	MODC	0	0	0	0	0	0	0
		W								
0x0071-0x007F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0080	MMCECH	R	ITR[3:0]				TGT[3:0]			
		W								
0x0081	MMCECL	R	ACC[3:0]				ERR[3:0]			
		W								
			= Unimplemented or Reserved							

Table 330. S12ZMMC register summary (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0082	MMCCCRH	R	CPUU	0	0	0	0	0	0	0
		W								
0x0083	MMCCCRL	R	0	CPUX	0	CPUI	0	0	0	0
		W								
0x0084	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0085	MMCPCH	R	CPUPC[23:16]							
		W								
0x0086	MMCPCM	R	CPUPC[15:8]							
		W								
0x0087	MMCPCL	R	CPUPC[7:0]							
		W								
0x0088-0x00FF	Reserved	R	0	0	0	0	0	0	0	0
		W								

= Unimplemented or Reserved

6.9.3.2 Register descriptions

This section consists of the S12ZMMC control and status register descriptions in address order.

6.9.3.2.1 Mode register (MODE)

Table 331. Mode register (MODE)

Address: 0x0070

	7	6	5	4	3	2	1	0
R	MODC	0	0	0	0	0	0	0
W								
Reset	MODC[242]	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
242.External signal (see Table 329).
243.Read: Anytime
Write: Only if a transition is allowed (see Figure 65)

The MODE register determines the operating mode of the MCU.

Table 332. MODE field descriptions

Field	Description
7 MODC	**Mode Select Bit** — This bit determines the current operating mode of the MCU. Its reset value is captured from the MODC pin at the rising edge of the RESET pin. Figure 65 illustrates the only valid mode transition from special single-chip mode to normal single chip mode.

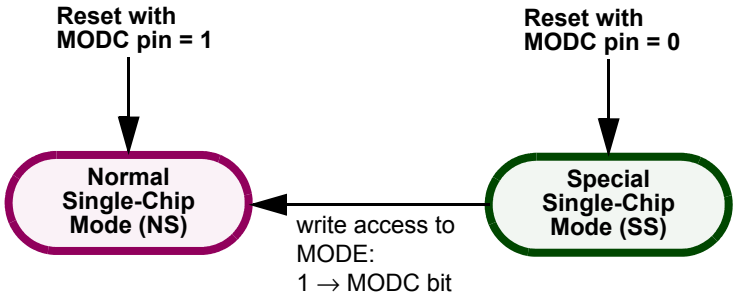


Figure 65. Mode transition diagram

6.9.3.2.2 Error code register (MMCECH, MMCECL)

Table 333. Error code register (MMCEC)

Address: 0x0080 (MMCECH)

	7	6	5	4	3	2	1	0
R W	ITR[3:0]				TGT[3:0]			
Reset	0	0	0	0	0	0	0	0

Address: 0x0081 (MMCECL)

	7	6	5	4	3	2	1	0
R W	ACC[3:0]				ERR[3:0]			
Reset	0	0	0	0	0	0	0	0

Notes:
244.Read: Anytime
Write: Write of 0xFFFF to MMCECH:MMCECL resets both registers to 0x0000

Table 334. MMCECH and MMCECL field descriptions

Field	Description
7-4 (MMCECH) ITR[3:0]	**Initiator Field** — The ITR[3:0] bits capture the initiator which caused the access violation. The initiator is captured in form of a 4 bit value which is assigned as follows: 0: none (no error condition detected) 1:S12ZCPU 2:reserved 3:ADC0 4:ADC1 5:PTU 6-15: reserved
3-0 (MMCECH) TGT[3:0]	**Target Field** — The TGT[3:0] bits capture the target of the faulty access. The target is captured in form of a 4 bit value which is assigned as follows: 0:none 1:register space 2:RAM 3:EEPROM 4:program flash 5:IFR 6-15: reserved

Table 334. MMCECH and MMCECL field descriptions (continued)

Field	Description
7-4 (MMCECL) ACC[3:0]	**Access Type Field** — The ACC[3:0] bits capture the type of memory access, which caused the access violation. The access type is captured in form of a 4 bit value which is assigned as follows: 0:none (no error condition detected) 1:opcode fetch 2:vector fetch 3:data load 4:data store 5-15: reserved
3-0 (MMCECL) ERR[3:0]	**Error Type Field** — The EC[3:0] bits capture the type of the access violation. The type is captured in form of a 4 bit value which is assigned as follows: 0:none (no error condition detected) 1:access to an illegal address range 2:uncorrectable ECC error 3-15:reserved

The MMCEC register captures debug information about access violations. It is set to a non-zero value if a S12ZCPU access violation has occurred. At the same time this register is set to a non-zero value, access information is captured in the MMCPCn and MMCCCRn registers. The MMCECn, the MMCPCn and the MMCCCRn registers are not updated if the MMCECn registers contain a non-zero value. The MMCECn registers are cleared by writing the value 0xFFFF.

6.9.3.2.3 Captured S12ZCPU condition code register (MMCCCRH, MMCCCRL)

Table 335. Captured S12ZCPU condition code register (MMCCCRH, MMCCCRL)

Address: 0x0082 (MMCCCRH)

	7	6	5	4	3	2	1	0
R	CPUU	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Address: 0x0083 (MMCCCRL)

	7	6	5	4	3	2	1	0
R	0	CPUX	0	CPUI	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
245.Read: Anytime
Write: Never

Table 336. MMCCCRH and MMCCCRL field descriptions

Field	Description
7 (MMCCCRH) CPUU	**S12ZCPU User State Flag** — This bit shows the state of the user/supervisor mode bit in the S12ZCPU's CCR at the time the access violation has occurred. The S12ZCPU user state flag is read-only; it will be automatically updated when the next error condition is flagged through the MMCEC register.
6 (MMCCCRL) CPUX	**S12ZCPU X-Interrupt Mask**— This bit shows the state of the X-interrupt mask in the S12ZCPU's CCR at the time the access violation has occurred. The S12ZCPU X-interrupt mask is read-only; it will be automatically updated when the next error condition is flagged through the MMCEC regiter.
4 (MMCCCRL) CPUI	**S12ZCPU I-Interrupt Mask**— This bit shows the state of the I-interrupt mask in the CPU's CCR at the time the access violation has occurred. The S12ZCPU I-interrupt mask is read-only; it will be automatically updated when the next error condition is flagged through the MMCEC register.

6.9.3.2.4 Captured S12ZCPU program counter (MMCPCH, MMCPCM, MMCPCL)

Table 337. Captured S12ZCPU program counter (MMCPCH, MMCPCM, MMCPCL)

Address: 0x0085 (MMCPCH)

	7	6	5	4	3	2	1	0
R	CPUPC[23:16]							
W								
Reset	0	0	0	0	0	0	0	0

Address: 0x0086 (MMCPCM)

	7	6	5	4	3	2	1	0
R	CPUPC[15:8]							
W								
Reset	0	0	0	0	0	0	0	0

Address: 0x0087 (MMCPCL)

	7	6	5	4	3	2	1	0
R	CPUPC[7:0]							
W								
Reset	0	0	0	0	0	0	0	0

Notes:
246. Read: Anytime
Write: Never

Table 338. MMCPCH, MMCPCM, and MMCPCL field descriptions

Field	Description
7–0 (MMCPCH) 7–0 (MMCPCM) 7–0 (MMCPCL) CPUPC[23:0]	**S12ZCPU Program Counter Value**— The CPUPC[23:0] stores the CPU's program counter value at the time the access violation occurred. CPUPC[23:0] always points to the instruction which triggered the violation. These bits are undefined if the error code registers (MMCECn) are cleared.

6.9.4 Functional description

This section provides a complete functional description of the S12ZMMC module.

6.9.4.1 Global memory map

The S12ZMMC maps all on-chip resources into an 16MB address space, the global memory map. The exact resource mapping is shown in Figure 66. The global address space is used by S12ZCPU, the S12ZBDC and the S12ZDBG module.

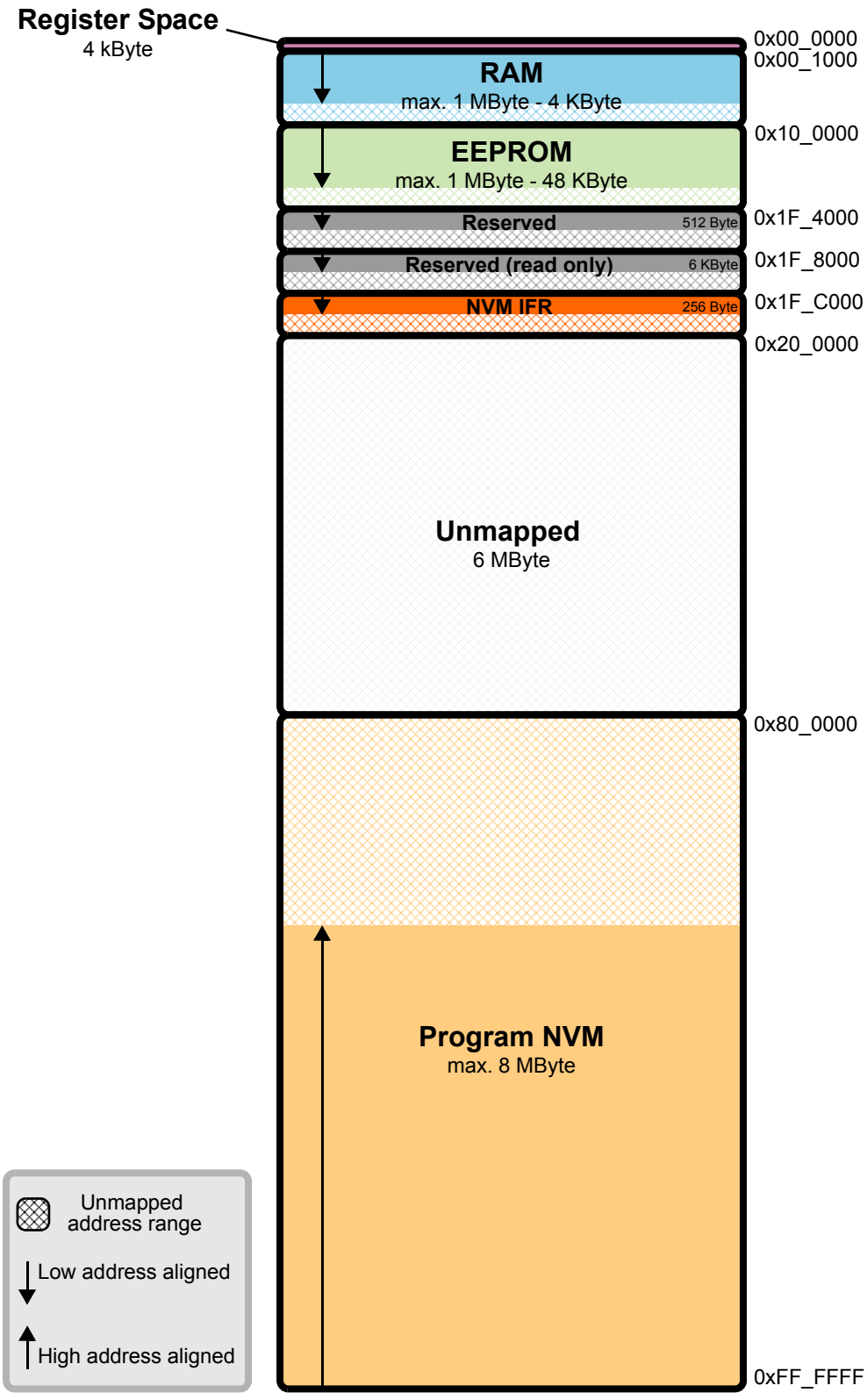


Figure 66. Global memory map

6.9.4.2 Illegal accesses

The S12ZMMC module monitors all memory traffic for illegal accesses. See Table 339 for a complete list of all illegal accesses.

Table 339. Illegal memory accesses

		S12ZCPU	S12ZBDC	ADCs and PTU
Register space	Read access	ok	ok	illegal access
	Write access	ok	ok	illegal access
	Code execution	illegal access		
RAM	Read access	ok	ok	ok
	Write access	ok	ok	ok
	Code execution	ok		
EEPROM	Read access	ok [(247)]	ok[(247)]	ok[(247)]
	Write access	illegal access	illegal access	illegal access
	Code execution	ok		
Reserved Space	Read access	ok	ok	illegal access
	Write access	only permitted in SS mode	ok	illegal access
	Code execution	illegal access		
Reserved Read-only Space	Read access	ok	ok	illegal access
	Write access	illegal access	illegal access	illegal access
	Code execution	illegal access		
NVM IFR	Read access	ok[(247)]	ok[(247)]	illegal access
	Write access	illegal access	illegal access	illegal access
	Code execution	illegal access		
Program NVM	Read access	ok[(247)]	ok[(247)]	ok
	Write access	illegal access	illegal access	illegal access
	Code execution	ok[(247)]		
Unmapped Space	Read access	illegal access	illegal access	illegal access
	Write access	illegal access	illegal access	illegal access
	Code execution	illegal access		

Notes:
247.Unsupported NVM accesses during NVM command execution (see section FTMRZ), are treated as illegal accesses.

Illegal accesses are reported in several ways:

- All illegal accesses performed by the S12ZCPU trigger machine exceptions.
- All illegal accesses performed through the S12ZBDC interface, are captured in the ILLACC bit of the BDCCSRL register.
- All illegal accesses performed by an ADC or PTU module trigger error interrupts. See ADC and PTU section for details.

Note

Illegal accesses caused by S12ZCPU opcode prefetches will also trigger machine exceptions, even if those opcodes might not be executed in the program flow. To avoid these machine exceptions, S12ZCPU instructions must mot be executed from the last (high addresses) 8 bytes of RAM, EEPROM, and Flash.

6.9.4.3 Uncorrectable ECC faults

RAM and flash use error correction codes (ECC) to detect and correct memory corruption. Each uncorrectable memory corruption, which is detected during a S12ZCPU, ADC or PTU access triggers a machine exception. Uncorrectable memory corruptions which are detected during a S12ZBDC access, are captured in the RAMWF or the RDINV bit of the BDCCSRL register.

6.10 S12Z debug module (S12ZDBGV2)

6.10.1 Introduction

The DBG module provides an on-chip trace buffer with flexible triggering capability to allow non-intrusive debug of application software. The DBG module is optimized for the S12Z architecture and allows debugging of CPU module operations.

Typically the DBG module is used in conjunction with the BDC module, whereby the user configures the DBG module for a debugging session over the BDC interface. Once configured the DBG module is armed and the device leaves active BDM returning control to the user program, which is then monitored by the DBG module. Alternatively the DBG module can be configured over a serial interface using SWI routines.

6.10.1.1 Glossary

Table 340. Glossary of terms

Term	Definition
COF	Change Of Flow. Change in the program flow due to a conditional branch, indexed jump or interrupt
PC	Program Counter
BDM	Background Debug Mode. In this mode CPU application code execution is halted. Execution of BDC “active BDM” commands is possible.
BDC	Background Debug Controller
WORD	16 bit data entity
Data Line	64 bit data entity
CPU	S12Z CPU module
Trigger	A trace buffer input that triggers tracing start, end or mid point

6.10.1.2 Overview

The comparators monitor the bus activity of the CPU. A single comparator match or a series of matches can trigger bus tracing and/or generate breakpoints. A state sequencer determines if the correct series of matches occurs. Similarly an external event can trigger bus tracing and/or generate breakpoints.

Independent of comparator matches, CPU breakpoints can be forced by an external pin event.

The trace buffer is visible through a 2-byte window in the register address map and can be read out using standard 16-bit word reads.

6.10.1.3 Features

- Four comparators (A, B, C, and D)
  - Comparators A and C compare the full address bus and full 32-bit data bus
  - Comparators A and C feature a data bus mask register
  - Comparators B and D compare the full address bus only
  - Each comparator can be configured to monitor CPU buses
  - Each comparator can be configured to monitor PC addresses or addresses of data accesses
  - Each comparator can select either read or write access cycles
  - Comparator matches can force state sequencer state transitions
- Three comparator modes
  - Simple address/data comparator match mode
  - Inside address range mode, Addmin ≤ Address ≤ Addmax
  - Outside address range match mode, Address < Addmin or Address > Addmax
- State sequencer control
  - State transitions forced by comparator matches
  - State transitions forced by software write to TRIG
  - State transitions forced by an external event

- The following types of breakpoints
  - CPU breakpoint entering active BDM on breakpoint (BDM)
  - CPU breakpoint executing SWI on breakpoint (SWI)
- Trace control
  - Tracing session triggered by state sequencer
  - Begin, End, and Mid alignment of tracing to trigger
- Four trace modes
  - Normal: change of flow (COF) PC information is stored (see Normal mode) for change of flow definition.
  - Loop1: same as Normal but inhibits consecutive duplicate source address entries
  - Detail: address and data for all read/write access cycles are stored
  - Pure PC: All program counter addresses are stored.
- 2 Pin (data and clock) profiling interface
  - Output of code flow information

6.10.1.4 Modes of operation

The DBG module can be used in all MCU functional modes.

The DBG module can issue breakpoint requests to force the device to enter active BDM or an SWI ISR. The BDC BACKGROUND command is also handled by the DBG to force the device to enter active BDM. When the device enters active BDM through a BACKGROUND command with the DBG module armed, the DBG remains armed.

6.10.1.5 Block diagram

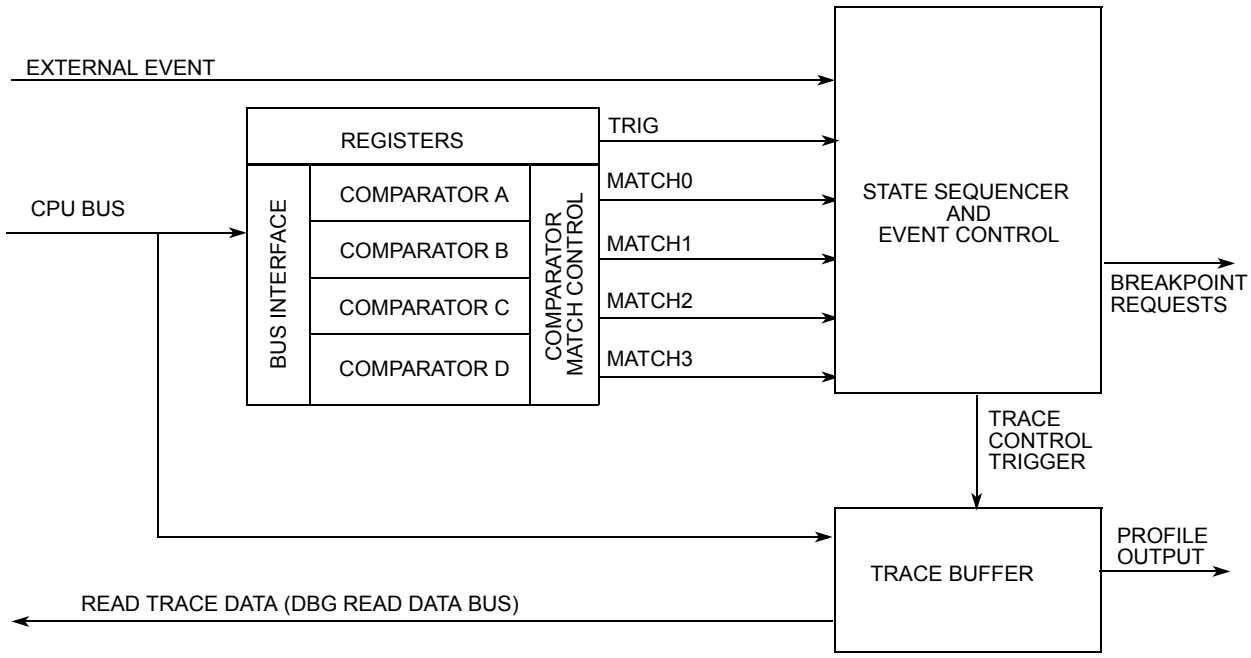


Figure 67. Debug module block diagram

6.10.2 External signal description

6.10.2.1 External event input

The DBG module features an external event input signal, DBGEEV. The mapping of this signal to a device pin is specified in the device specific documentation. This function can be enabled and configured by the EEVE field in the DBGC1 control register. This signal is input only and allows an external event to force a state sequencer transition, or trace buffer entry, or to gate trace buffer entries. With the external event function enabled, a falling edge at the external event pin constitutes an event. Rising edges have no effect. If configured for gating trace buffer entries, then a low level at the pin allows entries, but a high level suppresses entries. The maximum frequency of events is half the internal core bus frequency. The function is explained in the EEVE field description.

Note

Due to input pin synchronization circuitry, the DBG module sees external events two bus cycles after they occur at the pin. Thus an external event occurring less than two bus cycles before arming the DBG module is perceived to occur while the DBG is armed.

When the device is in Stop mode the synchronizer clocks are disabled and the external events are ignored.

6.10.2.2 Profiling output

The DBG module features a profiling data output signal PDO. The mapping of this signal to a device pin is specified in the device specific documentation. The device pin is enabled for profiling by setting the PDOE bit. The profiling function can be enabled by the PROFILE bit in the DBGTCRL control register. This signal is output only and provides a serial, encoded data stream that can be used by external development tools to reconstruct the internal CPU code flow, as specified in Code profiling. During code profiling the device PDOCLK output is used as a clock signal.

6.10.3 Memory map and registers

6.10.3.1 Module memory map

A summary of the registers associated with the DBG module is shown in Table 341. Detailed descriptions of the registers and bits are given in the subsections that follow.

Table 341. Quick reference to DBG registers

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0100	DBGC1	R	ARM	0	reserved	BDMBP	BRKCPU	reserved	EEVE	
		W		TRIG						
0x0101	DBGC2	R	0	0	0	0	CDCM		ABCM	
		W								
0x0102	DBGTCRH	R	reserved	TSOURCE	TRANGE		TRCMOD		TALIGN	
		W								
0x0103	DBGTCRL	R	0	0	0	0	DSTAMP	PDOE	PROFILE	STAMP
		W								
0x0104	DBGTB	R	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Bit 11**	**Bit 10**	**Bit 9**	**Bit 8**
		W								
0x0105	DBGTB	R	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
		W								
0x0106	DBGCNT	R	0	CNT						
		W								
0x0107	DBGSCR1	R	C3SC1	C3SC0	C2SC1	C2SC0	C1SC1	C1SC0	C0SC1	C0SC0
		W								

MM9Z1_638

Table 341. Quick reference to DBG registers (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0108	DBGSCR2	R	C3SC1	C3SC0	C2SC1	C2SC0	C1SC1	C1SC0	C0SC1	C0SC0
		W								
0x0109	DBGSCR3	R	C3SC1	C3SC0	C2SC1	C2SC0	C1SC1	C1SC0	C0SC1	C0SC0
		W								
0x010A	DBGEFR	R	PTBOVF	TRIGF	0	EEVF	ME3	ME2	ME1	ME0
		W								
0x010B	DBGSR	R	TBF	0	0	PTACT	0	SSF2	SSF1	SSF0
		W								
0x010C-0x010F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0110	DBGACTL	R	0	NDB	INST	0	RW	RWE	reserved	COMPE
		W								
0x0111-0x0114	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0115	DBGAAH	R	DBGAA[23:16]							
		W								
0x0116	DBGAAM	R	DBGAA[15:8]							
		W								
0x0117	DBGAAL	R	DBGAA[7:0]							
		W								
0x0118	DBGAD0	R	**Bit 31**	**30**	**29**	**28**	**27**	**26**	**25**	**Bit 24**
		W								
0x0119	DBGAD1	R	**Bit 23**	**22**	**21**	**20**	**19**	**18**	**17**	**Bit 16**
		W								
0x011A	DBGAD2	R	**Bit 15**	**14**	**13**	**12**	**11**	**10**	**9**	**Bit 8**
		W								
0x011B	DBGAD3	R	**Bit 7**	**6**	**5**	**4**	**3**	**2**	**1**	**Bit 0**
		W								
0x011C	DBGADM0	R	**Bit 31**	**30**	**29**	**28**	**27**	**26**	**25**	**Bit 24**
		W								
		W								
0x011E	DBGADM2	R	**Bit 15**	**14**	**13**	**12**	**11**	**10**	**9**	**Bit 8**
		W								
0x011F	DBGADM3	R	**Bit 7**	**6**	**5**	**4**	**3**	**2**	**1**	**Bit 0**
		W								
0x0120	DBGBCTL	R	0	0	INST	0	RW	RWE	reserved	COMPE
		W								
0x0121-0x0124	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0125	DBGBAH	R	DBGBA[23:16]							
		W								
0x0126	DBGBAM	R	DBGBA[15:8]							
		W								
0x0127	DBGBAL	R	DBGBA[7:0]							
		W								

Table 341. Quick reference to DBG registers (continued)

Address	Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0128-0x012F	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0130	DBGCCTL	R	0	NDB	INST	0	RW	RWE	reserved	COMPE
		W								
0x0131-0x0134	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0135	DBGCAH	R	DBGCA[23:16]							
		W								
0x0136	DBGCAM	R	DBGCA[15:8]							
		W								
0x0137	DBGCAL	R	DBGCA[7:0]							
		W								
0x0138	DBGCD0	R	**Bit 31**	**30**	**29**	**28**	**27**	**26**	**25**	**Bit 24**
		W								
0x0139	DBGCD1	R	**Bit 23**	**22**	**21**	**20**	**19**	**18**	**17**	**Bit 16**
		W								
0x013A	DBGCD2	R	**Bit 15**	**14**	**13**	**12**	**11**	**10**	**9**	**Bit 8**
		W								
0x013B	DBGCD3	R	**Bit 7**	**6**	**5**	**4**	**3**	**2**	**1**	**Bit 0**
		W								
0x013C	DBGCDM0	R	**Bit 31**	**30**	**29**	**28**	**27**	**26**	**25**	**Bit 24**
		W								
0x013D	DBGCDM1	R	**Bit 23**	**22**	**21**	**20**	**19**	**18**	**17**	**Bit 16**
		W								
0x013E	DBGCDM2	R	**Bit 15**	**14**	**13**	**12**	**11**	**10**	**9**	**Bit 8**
		W								
0x013F	DBGCDM3	R	**Bit 7**	**6**	**5**	**4**	**3**	**2**	**1**	**Bit 0**
		W								
0x0140	DBGDCTL	R	0	0	INST	0	RW	RWE	reserved	COMPE
		W								
0x0141-0x0144	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x0145	DBGDAH	R	DBGDA[23:16]							
		W								
0x0146	DBGDAM	R	DBGDA[15:8]							
		W								
0x0147	DBGDAL	R	DBGDA[7:0]							
		W								
0x0148-0x017F	Reserved	R	0	0	0	0	0	0	0	0
		W								

6.10.3.2 Register descriptions

This section consists of the DBG control and trace buffer register descriptions in address order. When ARM is set in DBGC1, the only bits in the DBG module registers that can be written are ARM, and TRIG.

6.10.3.2.1 Debug control register 1 (DBGC1)

Table 342. Debug control register (DBGC1)

Address: 0x0100

	7	6	5	4	3	2	1	0
R	ARM	0	reserved	BDMBP	BRKCPU	reserved	EEVE	
W		TRIG						
Reset	0	0	0	0	0	0	0	0

Notes:

248.Read: Anytime
Write: Bit 7 Anytime with the exception that it cannot be cleared if PTACT is set.
Bit 6 can be written anytime but always reads back as 0.
Bits 5:0 anytime DBG is not armed and PTACT is clear.

Note

On a write access to DBGC1 and simultaneous hardware disarm from an internal event, the hardware disarm has highest priority, clearing the ARM bit and generating a breakpoint, if enabled.

Note

When disarming the DBG by clearing ARM with software, the contents of bits[5:0] are not affected by the write, since up until the write operation, ARM = 1 preventing these bits from being written. These bits must be cleared using a second write if required.

Table 343. DBGC1 field descriptions

Field	Description
7 ARM	**Arm Bit** — The ARM bit controls whether the DBG module is armed. This bit can be set and cleared by register writes and is automatically cleared when the state sequencer returns to State0 on completing a debugging session. On setting this bit the state sequencer enters State1. 0 Debugger disarmed. No breakpoint is generated when clearing this bit by software register writes. 1 Debugger armed
6 TRIG	**Immediate Trigger Request Bit** — This bit when written to 1 requests an immediate transition to final state independent of comparator status. This bit always reads back a 0. Writing a 0 to this bit has no effect. 0 No effect. 1 Force state sequencer immediately to final state.
4 BDMBP	**Background Debug Mode Enable** — This bit determines if a CPU breakpoint causes the system to enter Background Debug mode (BDM) or initiate a Software Interrupt (SWI). If this bit is set but the BDC is not enabled, then no breakpoints are generated. 0 Breakpoint to Software Interrupt if BDM inactive. Otherwise no breakpoint. 1 Breakpoint to BDM, if BDC enabled. Otherwise no breakpoint.
3 BRKCPU	**CPU Breakpoint Enable** — The BRKCPU bit controls whether the debugger requests a breakpoint to CPU upon transitions to State 0. If tracing is enabled, the breakpoint is generated on completion of the tracing session. If tracing is not enabled, the breakpoint is generated immediately. Refer to Breakpoints for further details.
1–0 EEVE	**External Event Enable** — The EEVE bits configure the external event function. Table 344 explains the bit encoding.

Table 344. EEVE bit encoding

EEVE	Description
00	External event function disabled
01	External event forces a trace buffer entry if tracing is enabled
10	External event is mapped to the state sequencer, replacing comparator channel 3
11	External event pin gates trace buffer entries

6.10.3.2.2 Debug control register2 (DBGC2)

Table 345. Debug control register2 (DBGC2)

Address: 0x0101

	7	6	5	4	3	2	1	0
R	0	0	0	0	CDCM		ABCM	
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
249.Read: Anytime
Write: Anytime the module is disarmed and PTACT is clear
This register configures the comparators for range matching

Table 346. DBGC2 field descriptions

Field	Description
3–2 CDCM[1:0]	**C and D Comparator Match Control** — These bits determine the C and D comparator match mapping as described in Table 347.
1–0 ABCM[1:0]	**A and B Comparator Match Control** — These bits determine the A and B comparator match mapping as described in Table 348.

Table 347. CDCM encoding

CDCM	Description
00	Match2 mapped to comparator C match… Match3 mapped to comparator D match.
01	Match2 mapped to comparator C/D inside range… Match3 disabled.
10	Match2 mapped to comparator C/D outside range… Match3 disabled.
11	Reserved (250)

Notes:
250.Currently defaults to Match2 mapped to inside range: Match3 disabled.

Table 348. ABCM encoding

ABCM	Description
00	Match0 mapped to comparator A match… Match1 mapped to comparator B match.
01	Match0 mapped to comparator A/B inside range… Match1 disabled.
10	Match0 mapped to comparator A/B outside range… Match1 disabled.
11	Reserved (251)

Notes:
251.Currently defaults to Match0 mapped to inside range: Match1 disabled

6.10.3.2.3 Debug trace control register High (DBGTCRH)

Table 349. Debug trace control register (DBGTCRH)

Address: 0x0102

	7	6	5	4	3	2	1	0
R W	reserved	TSOURCE	TRANGE		TRCMOD		TALIGN	
Reset	0	0	0	0	0	0	0	0

Notes:

252.Read: Anytime
Write: Anytime the module is disarmed and PTACT is clear.

CAUTION

DBGTCR[7] is reserved. Setting this bit maps the tracing to an unimplemented bus, thus preventing proper operation.

This register configures the trace buffer for tracing and profiling.

Table 350. DBGTCRH field descriptions

Field	Description
6 TSOURCE	**Trace Control Bits** — The TSOURCE enables the tracing session. 0 No CPU tracing/profiling selected 1 CPU tracing/profiling selected
5–4 TRANGE	**Trace Range Bits** — The TRANGE bits allow filtering of trace information from a selected address range when tracing from the CPU in Detail mode. These bits have no effect in other tracing modes. To use a comparator for range filtering, the corresponding COMPE bit must remain cleared. If the COMPE bit is set then the comparator is used to generate events and the TRANGE bits have no effect. See Table 351 for range boundary definition.
3–2 TRCMOD	**Trace Mode Bits** — See Trace modes for detailed Trace Mode descriptions. In Normal mode, change of flow information is stored. In Loop1 mode, change of flow information is stored but redundant entries into trace memory are inhibited. In Detail mode, address and data for all memory and register accesses is stored. See Table 352.
1–0 TALIGN	**Trigger Align Bits** — These bits control whether the trigger is aligned to the beginning, end or the middle of a tracing or profiling session. See Table 353.

Table 351. TRANGE trace range encoding

TRANGE	Tracing Range
00	Trace from all addresses (No filter)
01	Trace only in address range from $00000 to Comparator D
10	Trace only in address range from Comparator C to $FFFFFF
11	Trace only in range from Comparator C to Comparator D

Table 352. TRCMOD trace mode bit encoding

TRCMOD	Description
00	Normal
01	Loop1
10	Detail
11	Pure PC

Table 353. TALIGN trace alignment encoding

TALIGN	Description
00	Trigger ends data trace
01	Trigger starts data trace
10	32 lines of data trace follow trigger
11 (253)	Reserved

Notes:
253.Tracing/Profiling disabled

6.10.3.2.4 Debug trace control register low (DBGTCRL)

Table 354. Debug control register low (DBGTCRL)

Address: 0x0103

	7	6	5	4	3	2	1	0
R	0	0	0	0	DSTAMP	PDOE	PROFILE	STAMP
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
254.Read: Anytime
Write: Anytime the module is disarmed and PTACT is clear

This register configures the profiling and timestamp features

Table 355. DBGTCRL field descriptions

Field	Description
3 DSTAMP	**Comparator D Timestamp Enable** — This bit, when set, enables Comparator D matches to generate timestamps in Detail, Normal and Loop1 trace modes. 0 Comparator D match does not generate timestamp 1 Comparator D match generates timestamp if timestamp function is enabled
2 PDOE	**Profile Data Out Enable** — This bit, when set, configures the device profiling pins for profiling. 0 Device pins not configured for profiling 1 Device pins configured for profiling
1 PROFILE	**Profile Enable** — This bit, when set, enables the profile function, whereby a subsequent arming of the DBG activates profiling. When PROFILE is set, the TRCMOD bits are ignored. 0 Profile function disabled 1 Profile function enabled
0 STAMP	**Timestamp Enable** — This bit, when set, enables the timestamp function. The timestamp function adds a timestamp to each trace buffer entry in Detail, Normal and Loop1 trace modes. 0 Timestamp function disabled 1 Timestamp function enabled

6.10.3.2.5 Debug trace buffer register (DBGTB)

Table 356. Debug trace buffer register (DBGTB)

Address: 0x0104, 0x0105

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
POR	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Other Resets	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Notes:

255.Read: Only when unlocked AND not armed AND the TSOURCE bit is set. Only aligned word read operations are supported. Misaligned word reads or byte reads return the error code 0xEE for each byte.
Write: Aligned word writes when the DBG is disarmed and PTACT is clear unlock the trace buffer for reading but do not affect trace buffer contents.

Table 357. DBGTB field descriptions

Field	Description
15–0 Bit[15:0]	**Trace Buffer Data Bits** — The Trace Buffer Register is a window through which the lines of the trace buffer may be read 16 bits at a time. Each valid read of DBGTB increments an internal trace buffer pointer which points to the next address to be read. When the ARM bit is written to 1 the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by writing to DBGTB with an aligned word write when the module is disarmed. The DBGTB register can be read only as an aligned word. Byte reads or misaligned access of these registers returns 0xEE and does not increment the trace buffer pointer. Similarly word reads while the debugger is armed or trace buffer is locked return 0xEEEE. The POR state is undefined Other resets do not affect the trace buffer contents.

6.10.3.2.6 Debug count register (DBGCNT)

Table 358. Debug count register (DBGCNT)

Address: 0x0106

	7	6	5	4	3	2	1	0
R	0	CNT						
W								
Reset	0	—	—	—	—	—	—	—
POR	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:

256.Read: Anytime
Write: Never

Table 359. DBGCNT field descriptions

Field	Description
6–0 CNT[6:0]	**Count Value** — The CNT bits [6:0] indicate the number of valid data lines stored in the trace buffer. Table 360 shows the correlation between the CNT bits and the number of valid data lines in the trace buffer. When the CNT rolls over to zero, the TBF bit in DBGSR is set. Thereafter incrementing of CNT continues if configured for end-alignment or mid-alignment. The DBGCNT register is cleared when ARM in DBGC1 is written to a one. The DBGCNT register is cleared by power-on-reset initialization but is not cleared by other system resets. If a reset occurs during a debug session, the DBGCNT register still indicates after the reset, the number of valid trace buffer entries stored before the reset occurred. The DBGCNT register is not decremented when reading from the trace buffer.

Table 360. CNT decoding table

TBF (DBGSR)	CNT[6:0]	Description
0	0000000	No data valid
0	0000001	32 bits of one line valid
0	0000010 0000100 0000110 … 1111100	1 line valid 2 lines valid 3 lines valid … 62 lines valid
0	1111110	63 lines valid
1	0000000	64 lines valid; if using Begin trigger alignment, ARM bit is cleared and the tracing session ends.
1	0000010 … 1111110	64 lines valid, oldest data has been overwritten by most recent data

6.10.3.2.7 Debug state control register 1 (DBGSCR1)

Table 361. Debug state control register 1 (DBGSCR1)

Address: 0x0107

	7	6	5	4	3	2	1	0
R W	C3SC1	C3SC0	C2SC1	C2SC0	C1SC1	C1SC0	C0SC1	C0SC0
Reset	0	0	0	0	0	0	0	0

Notes:
257.Read: Anytime
Write: If DBG is not armed and PTACT is clear

The state control register 1 selects the targeted next state while in State1. The matches refer to the outputs of the comparator match control logic as depicted in Figure 67 and described in Debug comparator A control register (DBGACTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 362. DBGSCR1 field descriptions

Field	Description
1–0 C0SC[1:0]	These bits select the targeted next state while in State1 following a match0.
3–2 C1SC[1:0]	These bits select the targeted next state while in State1 following a match1.
5–4 C2SC[1:0]	These bits select the targeted next state while in State1 following a match2.
7–6 C3SC[1:0]	If EEVE!=10, these bits select the targeted next state while in State1 following a match3. If EEVE = 10, these bits select the targeted next state while in State1 following an external event.

Table 363. State1 match state sequencer transitions

CxSC[1:0]	Function
00	Match has no effect
01	Match forces sequencer to State2
10	Match forces sequencer to State3
11	Match forces sequencer to Final State

In the case of simultaneous matches, the match on the higher channel number (3,2,1,0) has priority.

6.10.3.2.8 Debug state control register 2 (DBGSCR2)

Table 364. Debug state control register 2 (DBGSCR2)

Address: 0x0108

	7	6	5	4	3	2	1	0
R W	C3SC1	C3SC0	C2SC1	C2SC0	C1SC1	C1SC0	C0SC1	C0SC0
Reset	0	0	0	0	0	0	0	0

Notes:
258.Read: Anytime
Write: If DBG is not armed and PTACT is clear

The state control register 2 selects the targeted next state while in State2. The matches refer to the outputs of the comparator match control logic as depicted in Figure 67 and described in Debug comparator A control register (DBGACTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGXCTL control register.

Table 365. DBGSCR2 field descriptions

Field	Description
1–0 C0SC[1:0]	These bits select the targeted next state while in State2 following a match0.
3–2 C1SC[1:0]	These bits select the targeted next state while in State2 following a match1.
5–4 C2SC[1:0]	These bits select the targeted next state while in State2 following a match2.
7–6 C3SC[1:0]	If EEVE!=10, these bits select the targeted next state while in State2 following a match3. If EEVE =10, these bits select the targeted next state while in State2 following an external event.

Table 366. State2 match state sequencer transitions

CxSC[1:0]	Function
00	Match has no effect
01	Match forces sequencer to State1
10	Match forces sequencer to State3
11	Match forces sequencer to Final State

In the case of simultaneous matches, the match on the higher channel number (3,2,1,0) has priority.

6.10.3.2.9 Debug state control register 3 (DBGSCR3)

Table 367. Debug state control register 3 (DBGSCR3)

Address: 0x0109

	7	6	5	4	3	2	1	0
R W	C3SC1	C3SC0	C2SC1	C2SC0	C1SC1	C1SC0	C0SC1	C0SC0
Reset	0	0	0	0	0	0	0	0

Notes:
259.Read: Anytime
Write: If DBG is not armed and PTACT is clear

The state control register three selects the targeted next state while in State3. The matches refer to the outputs of the comparator match control logic as depicted in Figure 67 and described in Debug comparator A control register (DBGACTL)". Comparators must be enabled by setting the comparator enable bit in the associated DBGxCTL control register.

Table 368. DBGSCR3 field descriptions

Field	Description
1–0 C0SC[1:0]	These bits select the targeted next state while in State3 following a match0.
3–2 C1SC[1:0]	These bits select the targeted next state while in State3 following a match1.
5–4 C2SC[1:0]	These bits select the targeted next state while in State3 following a match2.
7–6 C3SC[1:0]	If EEVE!=10, these bits select the targeted next state while in State3 following a match3. If EEVE =10, these bits select the targeted next state while in State3 following an external event.

Table 369. State3 match state sequencer transitions

CxSC[1:0]	Function
00	Match has no effect
01	Match forces sequencer to State1
10	Match forces sequencer to State2
11	Match forces sequencer to Final State

In the case of simultaneous matches, the match on the higher channel number (3,2,1,0) has priority.

6.10.3.2.10 Debug event flag register (DBGEFR)

Table 370. Debug event flag register (DBGEFR)

Address: 0x010A

	7	6	5	4	3	2	1	0
R	PTBOVF	TRIGF	0	EEVF	ME3	ME2	ME1	ME0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
260.Read: Anytime
Write: Never

DBGEFR contains flag bits each mapped to events while armed. Should an event occur, then the corresponding flag is set. With the exception of TRIGF, the bits can only be set when the ARM bit is set. The TRIGF bit is set if a TRIG event occurs when ARM is already set, or if the TRIG event occurs simultaneous to setting the ARM bit. All other flags can only be cleared by arming the DBG module. Thus the contents are retained after a debug session for evaluation purposes.

A set flag does not inhibit the setting of other flags.

Table 371. DBGEFR field descriptions

Field	Description
7 PTBOVF	**Profiling Trace Buffer Overflow Flag** — Indicates the occurrence of a trace buffer overflow event during a profiling session. 0 No trace buffer overflow event 1 Trace buffer overflow event
6 TRIGF	**TRIG Flag** — Indicates the occurrence of a TRIG event during the debug session. 0 No TRIG event 1 TRIG event
4 EEVF	**External Event Flag** — Indicates the occurrence of an external event during the debug session. 0 No external event 1 External event
3–0 ME[3:0]	**Match Event[3:0]**— Indicates a comparator match event on the corresponding comparator channel**.**

6.10.3.2.11 Debug status register (DBGSR)

Table 372. Debug status register (DBGSR)

Address: 0x010B

	7	6	5	4	3	2	1	0
R	TBF	0	0	PTACT	0	SSF2	SSF1	SSF0
W								
Reset	—	0	0	0	0	0	0	0
POR	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
261.Read: Anytime
Write: Never

Table 373. DBGSR field descriptions

Field	Description
7 TBF	**Trace Buffer Full** — The TBF bit indicates that the trace buffer has been filled with data since it was last armed. If this bit is set, then all trace buffer lines contain valid data, regardless of the value of DBGCNT bits CNT[6:0]. The TBF bit is cleared when ARM in DBGC1 is written to a one. The TBF is cleared by the power on reset initialization. Other system generated resets have no affect on this bit
4 PTACT	**Profiling Transmission Active** — The PTACT bit, when set, indicates that the profiling transmission is still active. When clear, PTACT then profiling transmission is not active. The PTACT bit is set when profiling begins with the first PTS format entry to the trace buffer. The PTACT bit is cleared when the profiling transmission ends.
2–0 SSF[2:0]	**State Sequencer Flag Bits** — The SSF bits indicate the current State Sequencer state. During a debug session on each transition to a new state these bits are updated. If the debug session is ended by software clearing the ARM bit, then these bits retain their value to reflect the last state of the state sequencer before disarming. If a debug session is ended by an internal event, then the state sequencer returns to State0 and these bits are cleared to indicate that State0 was entered during the session. On arming the module the state sequencer enters State1 and these bits are forced to SSF[2:0] = 001. See Table 374**.**

Table 374. SSF[2:0] — State sequence flag bit encoding

SSF[2:0]	Current State
000	State0 (disarmed)
001	State1
010	State2
011	State3
100	Final State
101,110,111	Reserved

6.10.3.2.12 Debug comparator A control register (DBGACTL)

Table 375. Debug comparator A control register

Address: 0x0110

	7	6	5	4	3	2	1	0
R	0	NDB	INST	0	RW	RWE	reserved	COMPE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
262.Read: Anytime
Write: If DBG is not armed and PTACT is clear

Table 376. DBGACTL field descriptions

Field	Description
6 NDB	**Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. This bit is ignored if the INST bit in the same register is set. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
5 INST	**Instruction Select** — This bit configures the comparator to compare PC or data access addresses. 0 Comparator compares addresses of data accesses 1 Comparator compares PC address
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is ignored if RWE is clear or INST is set. 0 Write cycle is matched 1 Read cycle is matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is ignored when INST is set. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
0 COMPE	**Enable Bit** — Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled

Table 377 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if INST is set, because matches based on opcodes reaching the execution stage are data independent.

Table 377. Read or write comparison logic table

RWE Bit	RW Bit	RW Signal	Comment
0	x	0	RW not used in comparison
0	x	1	RW not used in comparison
1	0	0	Write match
1	0	1	No match
1	1	0	No match
1	1	1	Read match

6.10.3.2.13 Debug comparator A address register (DBGAAH, DBGAAM, DBGAAL)

Table 378. Debug comparator A address register

Address: 0x0115, DBGAAH

	23	22	21	20	19	18	17	16
R W	DBGAA[23:16]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0116, DBGAAM

	15	14	13	12	11	10	9	8
R W	DBGAA[15:8]							
Reset	0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0

Address: 0x0117, DBGAAL

	7	6	5	4	3	2	1	0
R W	DBGAA[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes:
263.Read: Anytime
Write: If DBG is not armed and PTACT is clear

Table 379. DBGAAH, DBGAAM, DBGAAL field descriptions

Field	Description
23–16 DBGAA [23:16]	**Comparator Address Bits [23:16]**— These comparator address bits control whether the comparator compares the address bus bits [23:16] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one
15–0 DBGAA [15:0]	**Comparator Address Bits [15:0]**— These comparator address bits control whether the comparator compares the address bus bits [15:0] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

6.10.3.2.14 Debug comparator A data register (DBGAD)

Table 380. Debug comparator A data register (DBGAD)

Address: 0x0118, 0x0119, 0x011A, 0x011B

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Notes:
264.Read: Anytime
Write: If DBG is not armed and PTACT is clear

This register can be accessed with a byte resolution, whereby DBGAD0, DBGAD1, DBGAD2, DBGAD3 map to DBGAD[31:0] respectively.

Table 381. DBGAD field descriptions

Field	Description
31–16 Bits[31:16] (DBGAD0, DBGAD1)	**Comparator Data Bits** — These bits control whether the comparator compares the data bus bits to a logic one or logic zero. The comparator data bits are only used in comparison if the corresponding data mask bit is logic 1. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one
15–0 Bits[15:0] (DBGAD2, DBGAD3)	**Comparator Data Bits** — These bits control whether the comparator compares the data bus bits to a logic one or logic zero. The comparator data bits are only used in comparison if the corresponding data mask bit is logic 1. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

6.10.3.2.15 Debug comparator A data mask register (DBGADM)

Table 1. Debug comparator A data mask register (DBGADM)

Address: 0x011C, 0x011D, 0x011E, 0x011F

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Notes:
265.Read: Anytime
Write: If DBG is not armed

This register can be accessed with a byte resolution, whereby DBGADM0, DBGADM1, DBGADM2, DBGADM3 map to DBGADM[31:0] respectively.

Table 382. DBGADM field descriptions

Field	Description
31–16 Bits[31:16] (DBGADM0, DBGADM1)	**Comparator Data Mask Bits** — These bits control whether the comparator compares the data bus bits to the corresponding comparator data compare bits. 0 Do not compare corresponding data bit 1 Compare corresponding data bit
15-0 Bits[15:0] (DBGADM2, DBGADM3)	**Comparator Data Mask Bits** — These bits control whether the comparator compares the data bus bits to the corresponding comparator data compare bits. 0 Do not compare corresponding data bit 1 Compare corresponding data bit

6.10.3.2.16 Debug comparator B control register (DBGBCTL)

Table 383. Debug comparator B control register

Address: 0x0120

	7	6	5	4	3	2	1	0
R	0	0	INST	0	RW	RWE	reserved	COMPE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
266.Read: Anytime
Write: If DBG is not armed and PTACT is clear

Table 384. DBGBCTL field descriptions

Field [(267)]	Description
5 INST	**Instruction Select** — This bit configures the comparator to compare PC or data access addresses. 0 Comparator compares addresses of data accesses 1 Comparator compares PC address
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is ignored if RWE is clear or INST is set. 0 Write cycle is matched 1 Read cycle is matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is ignored when INST is set. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
0 COMPE	**Enable Bit** — Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled

Notes:
267.If the ABCM field selects range mode comparisons, then DBGACTL bits configure the comparison, DBGBCTL is ignored.

Table 385 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if INST is set, as matches based on instructions reaching the execution stage are data independent.

Table 385. Read or write comparison logic table

RWE Bit	RW Bit	RW Signal	Comment
0	x	0	RW not used in comparison
0	x	1	RW not used in comparison
1	0	0	Write match
1	0	1	No match
1	1	0	No match
1	1	1	Read match

6.10.3.2.17 Debug comparator B address register (DBGBAH, DBGBAM, DBGBAL)

Table 386. Debug Comparator B Address Register

Address: 0x0125, DBGBAH

	23	22	21	20	19	18	17	16
R W	DBGBA[23:16]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0126, DBGBAM

	15	14	13	12	11	10	9	8
R W	DBGBA[15:8]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0127, DBGBAL

	7	6	5	4	3	2	1	0
R W	DBGBA[7:0]							

Notes:
268.Read: Anytime
Write: If DBG is not armed and PTACT is clear

Table 387. DBGBAH, DBGBAM, DBGBAL field descriptions

Field	Description
23–16 DBGBA [23:16]	**Comparator Address Bits [23:16]**— These comparator address bits control whether the comparator compares the address bus bits [23:16] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one
15–0 DBGBA [15:0]	**Comparator Address Bits**[15:0]— These comparator address bits control whether the comparator compares the address bus bits [15:0] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

6.10.3.2.18 Debug comparator C control register (DBGCCTL)

Table 388. Debug comparator C control register

Address: 0x0130

	7	6	5	4	3	2	1	0
R	0	NDB	INST	0	RW	RWE	reserved	COMPE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
269.Read: Anytime
Write: If DBG is not armed and PTACT is clear

Table 389. DBGCCTL field descriptions

Field	Description
6 NDB	**Not Data Bus** — The NDB bit controls whether the match occurs when the data bus matches the comparator register value or when the data bus differs from the register value. This bit is ignored if the INST bit in the same register is set. 0 Match on data bus equivalence to comparator register contents 1 Match on data bus difference to comparator register contents
5 INST	**Instruction Select** — This bit configures the comparator to compare PC or data access addresses. 0 Comparator compares addresses of data accesses 1 Comparator compares PC address
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is ignored if RWE is clear or INST is set. 0 Write cycle is matched 1 Read cycle is matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is not used if INST is set. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
0 COMPE	**Enable Bit** — Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled

Table 390 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if INST is set, because matches based on opcodes reaching the execution stage are data independent.

Table 390. Read or write comparison logic table

RWE Bit	RW Bit	RW Signal	Comment
0	x	0	RW not used in comparison
0	x	1	RW not used in comparison
1	0	0	Write match
1	0	1	No match
1	1	0	No match
1	1	1	Read match

6.10.3.2.19 Debug comparator C address register (DBGCAH, DBGCAM, DBGCAL)

Table 391. Debug comparator C address register

Address: 0x0135, DBGCAH

	23	22	21	20	19	18	17	16
R W	DBGCA[23:16]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0136, DBGCAM

	15	14	13	12	11	10	9	8
R W	DBGCA[15:8]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0137, DBGCAL

	7	6	5	4	3	2	1	0
R W	DBGCA[7:0]							
Reset	0	0	0	0	0	0	0	0

Notes:
270.Read: Anytime
Write: If DBG is not armed and PTACT is clear

Table 392. DBGCAH, DBGCAM, DBGCAL field descriptions

Field	Description
23–16 DBGCA [23:16]	**Comparator Address Bits [23:16]**— These comparator address bits control whether the comparator compares the address bus bits [23:16] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one
15–0 DBGCA [15:0]	**Comparator Address Bits**[15:0]— These comparator address bits control whether the comparator compares the address bus bits [15:0] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

6.10.3.2.20 Debug comparator C data register (DBGCD)

Table 393. Debug comparator C data register (DBGCD)

Address: 0x0138, 0x0139, 0x013A, 0x013B

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Table 393. Debug comparator C data register (DBGCD) (continued)

Address:	0x0138, 0x0139, 0x013A, 0x013B															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Notes:
271.Read: Anytime
Write: If DBG is not armed and PTACT is clear

This register can be accessed with a byte resolution, whereby DBGCD0, DBGCD1, DBGCD2, DBGCD3 map to DBGCD[31:0] respectively.

Table 394. DBGCD field descriptions

Field	Description
31–16 Bits[31:16] (DBGCD0, DBGCD1)	**Comparator Data Bits** — These bits control whether the comparator compares the data bus bits to a logic one or logic zero. The comparator data bits are only used in comparison if the corresponding data mask bit is logic 1. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one
15–0 Bits[15:0] (DBGCD2, DBGCD3)	**Comparator Data Bits** — These bits control whether the comparator compares the data bus bits to a logic one or logic zero. The comparator data bits are only used in comparison if the corresponding data mask bit is logic 1. 0 Compare corresponding data bit to a logic zero 1 Compare corresponding data bit to a logic one

6.10.3.2.21 Debug comparator C data mask register (DBGCDM)

Table 395. Debug comparator C data mask register (DBGCDM)

Address:	0x013C, 0x013D, 0x013E, 0x013F															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Notes:
272.Read: Anytime
Write: If DBG is not armed

This register can be accessed with a byte resolution, whereby DBGCDM0, DBGCDM1, DBGCDM2, DBGCDM3 map to DBGCDM[31:0] respectively.

Table 396. DBGCDM field descriptions

Field	Description
31–16 Bits[31:16] (DBGCDM0, DBGCDM1)	**Comparator Data Mask Bits** — These bits control whether the comparator compares the data bus bits to the corresponding comparator data compare bits. 0 Do not compare corresponding data bit 1 Compare corresponding data bit
15–0 Bits[15:0] (DBGCDM2, DBGCDM3)	**Comparator Data Mask Bits** — These bits control whether the comparator compares the data bus bits to the corresponding comparator data compare bits. 0 Do not compare corresponding data bit 1 Compare corresponding data bit

6.10.3.2.22 Debug comparator D control register (DBGDCTL)

Table 397. Debug comparator D control register

Address: 0x0140

	7	6	5	4	3	2	1	0
R	0	0	INST	0	RW	RWE	reserved	COMPE
W								
Reset	0	0	0	0	0	0	0	0

Notes:
273.Read: Anytime
Write: If DBG is not armed and PTACT is clear

Table 398. DBGDCTL field descriptions

Field [(274)]	Description
5 INST	**Instruction Select** — This bit configures the comparator to compare PC or data access addresses. 0 Comparator compares addresses of data accesses 1 Comparator compares PC address
3 RW	**Read/Write Comparator Value Bit** — The RW bit controls whether read or write is used in compare for the associated comparator. The RW bit is ignored if RWE is clear or INST is set. 0 Write cycle is matched 1 Read cycle is matched
2 RWE	**Read/Write Enable Bit** — The RWE bit controls whether read or write comparison is enabled for the associated comparator. This bit is ignored if INST is set. 0 Read/Write is not used in comparison 1 Read/Write is used in comparison
0 COMPE	**Enable Bit** — Determines if comparator is enabled 0 The comparator is not enabled 1 The comparator is enabled

Notes:
274.If the CDCM field selects range mode comparisons, then DBGCCTL bits configure the comparison, DBGDCTL is ignored.

Table 399 shows the effect for RWE and RW on the comparison conditions. These bits are ignored if INST is set, because matches based on opcodes reaching the execution stage are data independent.

Table 399. Read or write comparison logic table

RWE Bit	RW Bit	RW Signal	Comment
0	x	0	RW not used in comparison
0	x	1	RW not used in comparison
1	0	0	Write match
1	0	1	No match
1	1	0	No match
1	1	1	Read match

6.10.3.2.23 Debug comparator D address register (DBGDAH, DBGDAM, DBGDAL)

Table 400. Debug comparator D address register

Address: 0x0145, DBGDAH

	23	22	21	20	19	18	17	16
R W	DBGDA[23:16]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0146, DBGDAM

	15	14	13	12	11	10	9	8
R W	DBGDA[15:8]							
Reset	0	0	0	0	0	0	0	0

Address: 0x0147, DBGDAL

	7	6	5	4	3	2	1	0
R W	DBGDA[7:0]							
Reset	0	0	0	0	0	0	0	0
	23	22	21	20	19	18	17	16

Notes:
275.Read: Anytime
Write: If DBG is not armed and PTACT is clear

Table 401. DBGDAH, DBGDAM, DBGDAL field descriptions

Field	Description
23–16 DBGDA [23:16]	**Comparator Address Bits [23:16]**— These comparator address bits control whether the comparator compares the address bus bits [23:16] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one
15–0 DBGDA [15:0]	**Comparator Address Bits**[15:0]— These comparator address bits control whether the comparator compares the address bus bits [15:0] to a logic one or logic zero. 0 Compare corresponding address bit to a logic zero 1 Compare corresponding address bit to a logic one

6.10.4 Functional description

This section provides a complete functional description of the DBG module.

6.10.4.1 DBG operation

The DBG module operation is enabled by setting ARM in DBGC1. When armed it supports storing of data in the trace buffer and can be used to generate breakpoints to the CPU. The DBG module is made up of comparators, control logic, the state sequencer, and the trace buffer, Figure 67.

The comparators monitor the bus activity of the CPU. Comparators can be configured to monitor opcode addresses (effectively the PC address) or data accesses. Comparators can be configured during data accesses to mask out individual data bus bits and to use R/W access qualification in the comparison. Comparators can be configured to monitor a range of addresses.

When configured for data access comparisons, the match is generated if the address (and optionally data) of a data access matches the comparator value.

Configured for monitoring opcode addresses, the match is generated when the associated opcode reaches the execution stage of the instruction queue, but before execution of that opcode.

When a match with a comparator register value occurs, the associated control logic can force the state sequencer to another state (see Figure 68).

The state sequencer can transition freely between the states 1, 2, and 3. On transition to Final State, bus tracing is triggered.

Independent of the comparators, state sequencer transitions can be forced by the external event input or by writing to the TRIG bit in the DBGC1 control register.

The trace buffer is visible through a 2-byte window in the register address map and can be read out using standard 16-bit word reads.

6.10.4.2 Comparator modes

The DBG contains four comparators, A, B, C, and D. Each comparator compares the address stored in DBGXAH, DBGXAM, and DBGXAL with the PC (opcode addresses) or selected address bus (data accesses). Furthermore, comparators A and C can compare the data buses to values stored in DBGXD3-0 and allow data bit masking.

The comparators can monitor the buses for an exact address or an address range. The comparator configuration is controlled by the control register contents and the range control by the DBGC2 contents.

The comparator control register also allows the type of data access to be included in the comparison through the use of the RWE and RW bits. The RWE bit controls whether the access type is compared for the associated comparator and the RW bit selects either a read or write access for a valid match.

The INST bit in each comparator control register is used to determine the matching condition. By setting INST, the comparator matches opcode addresses, whereby the databus, data mask, RW and RWE bits are ignored. The comparator register must be loaded with the exact opcode address.

The comparator can be configured to match memory access addresses by clearing the INST bit.

Each comparator match can force a transition to another state sequencer state (see Events").

Once a successful comparator match has occurred, the condition that caused the original match is not verified again on subsequent matches. Thus if a particular data value is verified at a given address, this address may not contain that data value when a subsequent match occurs.

Comparators C and D can also be used to select an address range to trace from, when tracing CPU accesses in Detail mode. This is determined by the TRANGE bits in the DBGTCRH register. The TRANGE encoding is shown in Table 351. If the TRANGE bits select a range definition using comparator D and the COMPE bit is clear, then comparator D is configured for trace range definition. By setting the COMPE bit, the comparator is configured for address bus comparisons, the TRANGE bits are ignored and the tracing range function is disabled. Similarly if the TRANGE bits select a range definition using comparator C and the COMPE bit is clear, then comparator C is configured for trace range definition.

Match[0, 1, 2, 3] map directly to Comparators [A, B, C, D] respectively, except in range modes (see Debug control register2 (DBGC2)"). Comparator priority rules are described in the event priority section (Event priorities").

6.10.4.2.1 Exact address comparator match

With range comparisons disabled, the match condition is an exact equivalence of address bus with the value stored in the comparator address registers. Qualification of the type of access (R/W) is also possible.

Code may contain various access forms of the same address, for example a 16-bit access of ADDR[n] or byte access of ADDR[n+1] both access n+1. The comparators ensure that any access of the address defined by the comparator address register generates a match, as shown in the example of Table 402. Thus if the comparator address register contains ADDR[n+1] any access of ADDR[n+1] matches. This means that a 16-bit access of ADDR[n] or 32-bit access of ADDR[n-1] also match because they also access ADDR[n+1]. The right hand columns show the contents of DBGxA that would match for each access.

Table 402. Comparator address bus matches

Access	Address	ADDR[n]	ADDR[n+1]	ADDR[n+2]	ADDR[n+3]
32-bit	ADDR[n]	Match	Match	Match	Match
16-bit	ADDR[n]	Match	Match	No Match	No Match
16-bit	ADDR[n+1]	No Match	Match	Match	No Match
8-bit	ADDR[n]	Match	No Match	No Match	No Match

If the comparator INST bit is set, the comparator address register contents are compared with the PC, the data register contents and access type bits are ignored. The comparator address register must be loaded with the address of the first opcode byte.

6.10.4.2.2 Address and data comparator match

Comparators A and C feature data comparators, for data access comparisons. The comparators do not evaluate if accessed data is valid. Accesses across aligned 32-bit boundaries are split internally into consecutive accesses. The data comparator mapping to accessed addresses for the CPU is shown in Table 403, whereby the Address column refers to the lowest 2 bits of the lowest accessed address. This corresponds to the most significant data byte.

Table 403. Comparator data byte alignment

Address[1:0]	Data Comparator
00	DBGxD0
01	DBGxD1
10	DBGxD2
11	DBGxD3

The fixed mapping of data comparator bytes to addresses within a 32-bit data field ensures data matches independent of access size. To compare a single data byte within the 32-bit field, the other bytes within that field must be masked using the corresponding data mask registers. This ensures that any access of that byte (32-bit,16-bit, or 8-bit) with matching data causes a match. If no bytes are masked then the data comparator always compares all 32-bits and can only generate a match on a 32-bit access with correct 32-bit data value. In this case, 8-bit or 16-bit accesses within the 32-bit field cannot generate a match even if the contents of the addressed bytes match, because all 32-bits must match. In Table 404, the Access Address column refers to the address bits[1:0] of the lowest accessed address (most significant data byte).

Table 404. Data register use dependency on CPU access type

			Memory Address[2:0]						
Case	Access Address	Access Size	000	001	010	011	100	101	110
1	00	32-bit	DBGxD0	DBGxD1	DBGxD2	DBGxD3			
2	01	32-bit		DBGxD1	DBGxD2	DBGxD3	DBGxD0		
3	10	32-bit			DBGxD2	DBGxD3	DBGxD0	DBGxD1	
4	11	32-bit				DBGxD3	DBGxD0	DBGxD1	DBGxD0
5	00	16-bit	DBGxD0	DBGxD1					
6	01	16-bit		DBGxD1	DBGxD2				
7	10	16-bit			DBGxD2	DBGxD3			
8	11	16-bit				DBGxD3	DBGxD0		
9	00	8-bit	DBGxD0						
10	01	8-bit		DBGxD1					
11	10	8-bit			DBGxD2				
12	11	8-bit				DBGxD3			
13	00	8-bit					DBGxD0		
				Denotes byte that is not accessed.					

For a match of a 32-bit access with data compare, the address comparator must be loaded with the address of the lowest accessed byte. For Case1 Table 404 this corresponds to 000, for Case2 it corresponds to 001. To compare all 32-bits, it is required that no bits are masked.

6.10.4.2.3 Data bus comparison NDB dependency

Comparators A and C feature NDB control bits, which allow data bus comparators to be configured to either match on equivalence or on difference. This allows monitoring of a difference in the contents of an address location from an expected value.

When matching on an equivalence (NDB = 0), each individual data bus bit position can be masked out by clearing the corresponding mask bit, so that it is ignored in the comparison. A match occurs when all data bus bits with corresponding mask bits set are equivalent. If all mask register bits are clear, then a match is based on the address bus only, the data bus is ignored.

When matching on a difference, mask bits can be cleared to ignore bit positions. A match occurs when any data bus bit with corresponding mask bit set is different. Clearing all mask bits, causes all bits to be ignored and prevents a match because no difference can be detected.

In this case, address bus equivalence does not cause a match. Bytes that are not accessed are ignored. Thus when monitoring a multi byte field for a difference, partial accesses of the field only return a match if a difference is detected in the accessed bytes.

Table 405. NDB and MASK bit dependency

NDB	DBGADM	Comment
0	0	Do not compare data bus bit.
0	1	Compare data bus bit. Match on equivalence.
1	0	Do not compare data bus bit.
1	1	Compare data bus bit. Match on difference.

6.10.4.2.4 Range comparisons

Range comparisons are accurate to byte boundaries. Thus for data access comparisons a match occurs if at least one byte of the access is in the range (inside range) or outside the range (outside range). For opcode comparisons only the address of the first opcode byte is compared with the range.

When using the AB comparator pair for a range comparison, the data bus can be used for qualification by using the comparator A data and data mask registers. Similarly when using the CD comparator pair for a range comparison, the data bus can be used for qualification by using the comparator C data and data mask registers. The DBGACTL/DBGCCTL RW and RWE bits can be used to qualify the range comparison on either a read or a write access. The corresponding DBGBCTL/DBGDCTL bits are ignored. The DBGACTL/DBGCCTL COMPE, INST and SRC bits are used for range comparisons of the AB and CD ranges respectively. The DBGBCTL/DBGDCTL COMPE, INST and SRC bits are ignored in range modes.

6.10.4.2.4.1 Inside range (CompAC_Addr $\leq$ address $\leq$ CompBD_Addr)

In the Inside Range comparator mode, either comparator pair A and B or comparator pair C and D can be configured for range comparisons by the control register (DBGC2). The match condition requires a simultaneous valid match for both comparators. A match condition on only one comparator is not valid.

6.10.4.2.4.2 Outside range (address < CompAC_Addr or address > CompBD_Addr)

In the Outside Range comparator mode, either comparator pair A and B or comparator pair C and D can be configured for range comparisons. A single match condition on either of the comparators is recognized as valid. Outside range mode in combination with opcode address matches can be used to detect if opcodes are from an unexpected range.

Note

When configured for data access matches, an outside range match would typically occur at any interrupt vector fetch or register access. This can be avoided by setting the upper or lower range limit to $FFFFFF or $000000 respectively. Interrupt vector fetches do not cause opcode address matches.

6.10.4.3 Events

Events are used as qualifiers for a state sequencer change of state. The state control register for the current state determines the next state for each event. An event can immediately initiate a transition to the next state sequencer state whereby the corresponding flag in DBGSR is set.

6.10.4.3.1 Comparator match events

6.10.4.3.1.1 Opcode address comparator match

The comparator is loaded with the address of the selected instruction and the comparator control register INST bit is set. When the opcode reaches the execution stage of the instruction queue, a match occurs just before the instruction executes, allowing a breakpoint immediately before the instruction boundary. The comparator address register must contain the address of the first opcode byte for the match to occur. Opcode address matches are data independent thus the RWE and RW bits are ignored. CPU compares are disabled when BDM becomes active.

6.10.4.3.1.2 Data access comparator match

Data access matches are generated when an access occurs at the address contained in the comparator address register. The match can be qualified by the access data and by the access type (read/write). The breakpoint occurs a maximum of 2 instructions after the access in the CPU flow. Note that if a COF occurs between access and breakpoint, the opcode address of the breakpoint can be elsewhere in the memory map.

Opcode fetches are not classed as data accesses. Thus data access matches are not possible on opcode fetches.

6.10.4.3.2 External event

The DBGEEV input signal can force a state sequencer transition, independent of internal comparator matches. The DBGEEV is an input signal mapped directly to a device pin and configured by the EEVE field in DBGC1. The external events can change the state sequencer state, or force a trace buffer entry, or gate trace buffer entries. Table 344 explains external event configuration.

If configured to change the state sequencer state, then the external match is mapped to DBGSCRx bits C3SC[1:0]. In this configuration, internal comparator channel3 is de-coupled from the state sequencer but can still be used for timestamps. The DBGEFR bit EEVF is set when an external event occurs.

6.10.4.3.3 Setting the TRIG bit

Independent of comparator matches it is possible to initiate a tracing session and/or breakpoint by writing the TRIG bit in DBGC1 to a logic “1”. This forces the state sequencer into the Final State. If configured for End aligned tracing or for no tracing, the transition to Final State is followed immediately by a transition to State0. If configured for Begin- or Mid Aligned tracing, the state sequencer remains in Final State until tracing is complete, then it transitions to State0.

Breakpoints, if enabled, are issued on the transition to State0.

6.10.4.3.4 Profiling trace buffer overflow event

During code profiling, a trace buffer overflow forces the state sequencer into the disarmed State0 and, if breakpoints are enabled, issues a breakpoint request to the CPU.

6.10.4.3.5 Event priorities

If simultaneous events occur, the priority is resolved according to Table 406. Lower priority events are suppressed. It is thus possible to miss a lower priority event if it occurs simultaneously with an event of a higher priority. The event priorities dictate that in the case of simultaneous matches, the match on the higher comparator channel number (3,2,1,0) has priority.

If a write access to DBGC1 with the ARM bit position set occurs simultaneously to a hardware disarm from an internal event, then the ARM bit is cleared due to the hardware disarm.

Table 406. Event priorities

Priority	Source	Action
Highest	TB Overflow	Immediate force to state 0, generate breakpoint and terminate tracing
	TRIG	Force immediately to final state
	DBGEEV	Force to next state as defined by state control registers (EEVE = 2’b10)
	Match3	Force to next state as defined by state control registers
	Match2	Force to next state as defined by state control registers
	Match1	Force to next state as defined by state control registers
Lowest	Match0	Force to next state as defined by state control registers

6.10.4.4 State sequence control

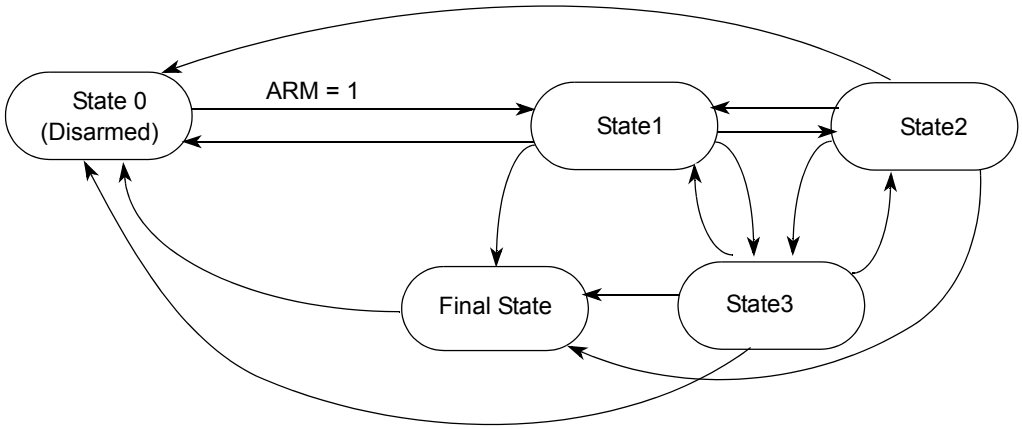


Figure 68. State sequencer diagram

The state sequencer allows a defined sequence of events to provide a breakpoint and/or a trigger point for tracing of data in the trace buffer. When the DBG module is armed by setting the ARM bit in the DBGC1 register, the state sequencer enters State1. Further transitions between the states are controlled by the state control registers and depend upon event occurrences (see Events). From Final State the only permitted transition is back to the disarmed State0. Transition between the states 1 to 3 is not restricted. Each transition updates the SSF[2:0] flags in DBGSR accordingly to indicate the current state. If breakpoints are enabled, then an event based transition to State0 generates the breakpoint request. A transition to State0 resulting from writing “0” to the ARM bit does not generate a breakpoint request.

6.10.4.4.1 Final state

On entering Final State a trigger may be issued to the trace buffer according to the trigger position control as defined by the TALIGN field (see Debug trace control register High (DBGTCRH)”).

If tracing is enabled and either Begin or Mid aligned triggering is selected, the state sequencer remains in Final State until completion of the trace. On completion of the trace the state sequencer returns to State0 and the debug module is disarmed; if breakpoints are enabled, a breakpoint request is generated.

If tracing is disabled or End aligned triggering is selected, then when the Final State is reached the state sequencer returns to State0 immediately and the debug module is disarmed. If breakpoints are enabled, a breakpoint request is generated on transitions to State0.

6.10.4.5 Trace buffer operation

The trace buffer is a 64 lines deep by 64-bits wide RAM array. The DBG module stores trace information in the RAM array in a circular buffer format. Data is stored in mode dependent formats, as described in the following sections. After each trace buffer entry, the counter register DBGCNT is incremented. Trace buffer rollover is possible when configured for End- or Mid-Aligned tracing, such that older entries are replaced by newer entries. Tracing of CPU activity is disabled when the BDC is active.

The RAM array can be accessed through the register DBGTB using 16-bit wide word accesses. After each read, the internal RAM pointer is incremented so that the next read will receive fresh information. Reading the trace buffer while the DBG is armed returns invalid data and the trace buffer pointer is not incremented.

In Detail mode, the address range for CPU access tracing can be limited to a range specified by the TRANGE bits in DBGTCRH. This function uses comparators C and D to define an address range inside which accesses should be traced. Thus traced accesses can be restricted, for example, to particular register or RAM range accesses.

The external event pin can be configured to force trace buffer entries in Normal or Loop1 trace modes. All tracing modes support trace buffer gating. In Pure PC and Detail modes external events do not force trace buffer entries.

If the external event pin is configured to gate trace buffer entries, then any trace mode is valid.

6.10.4.5.1 Trace trigger alignment

Using the TALIGN bits (see Debug trace control register High (DBGTCRH)), it is possible to align the trigger with the end, the middle, or the beginning of a tracing session.

If End or Mid-alignment is selected, tracing begins when the ARM bit in DBGC1 is set and State1 is entered. The transition to Final State if End-alignment is selected, ends the tracing session. The transition to Final State if Mid-alignment is selected signals that another 32 lines are traced before ending the tracing session. Tracing with Begin-alignment starts at the trigger and ends when the trace buffer is full.

Table 407. Tracing alignment

TALIGN	Tracing Begin	Tracing End
00	On arming	At trigger
01	At trigger	When trace buffer is full
10	On arming	When 32 trace buffer lines have been filled after trigger
11	Reserved	

6.10.4.5.1.1 Storing with begin-alignment

Storing with Begin-alignment, data is not stored in the trace buffer until the Final State is entered. Once the trigger condition is met, the DBG module remains armed until 64 lines are stored in the trace buffer. Using Begin-alignment together with opcode address comparisons, if the instruction is about to be executed, then the trace is started. If the trigger is at the address of a COF instruction, while tracing COF addresses, then that COF address is stored to the trace buffer. If breakpoints are enabled, the breakpoint is generated upon entry into State0 on completion of the tracing session; thus the breakpoint does not occur at the instruction boundary.

6.10.4.5.1.2 Storing with mid-alignment

Storing with Mid-alignment, data is stored in the trace buffer as soon as the DBG module is armed. When the trigger condition is met, another 32 lines are traced before ending the tracing session, irrespective of the number of lines stored before the trigger occurred, then the DBG module is disarmed and no more data is stored. Using Mid-alignment with opcode address triggers, if the instruction is about to be executed then the trace is continued for another 32 lines. If breakpoints are enabled, the breakpoint is generated upon entry into State0 on completion of the tracing session. The breakpoint does not occur at the instruction boundary. When configured for Compressed Pure-PC tracing, the MAT info bit is set to indicate the last PC entry before a trigger event.

6.10.4.5.1.3 Storing with end-alignment

Storing with End-alignment, data is stored in the trace buffer until the Final State is entered. Following this trigger, the DBG module immediately transitions to State0. If the trigger is at the address of a COF instruction the trigger event is not stored in the trace buffer.

6.10.4.5.2 Trace modes

The DBG module can operate in four trace modes. The mode is selected using the TRCMOD bits in the DBGTCRH register. Normal, Loop1 and Detail modes can be configured to store a timestamp with each entry, by setting the STAMP bit.The modes are described in the following subsections.

In addition to the listed trace modes it is also possible to use code profiling to fill the trace buffer with a highly compressed COF format. This can be subsequently read out in the same fashion as the listed trace modes (see Code profiling).

6.10.4.5.2.1 Normal mode

In Normal mode, change of flow (COF) program counter (PC) addresses are stored.

CPU COF addresses are defined as follows:

- Source address of taken conditional branches (bit-conditional, and loop primitives)
- Destination address of indexed JMP and JSR instructions
- Destination address of RTI and RTS instructions.
- Vector address of interrupts

BRA, BSR, BGND, as well as non-indexed JMP and JSR instructions, are not classified as change of flow and are not stored in the trace buffer.

COF addresses stored include the full address bus of CPU and an information byte, which contains bits to indicate whether the stored address was a source, destination or vector address.

Note

When a CPU indexed jump instruction is executed, the destination address is stored to the trace buffer on instruction completion, indicating the COF has taken place. If an interrupt occurs simultaneously then the next instruction carried out is actually from the interrupt service routine. The instruction at the destination address of the original program flow gets executed after the interrupt service routine.

In the following example an IRQ interrupt occurs during execution of the indexed JMP at address MARK1. The NOP at the destination (SUB_1) is not executed until after the IRQ service routine but the destination address is entered into the trace buffer to indicate that the indexed JMP COF has taken place.

```
LD X,#SUB_1
MARK1:JMP(0,X); IRQ interrupt occurs during execution of this
MARK2:NOP;

SUB_1:NOP ; JMP Destination address TRACE BUFFER ENTRY 1
; RTI Destination address TRACE BUFFER ENTRY 3
NOP;
ADDR1:DBNED0,PART5; Source address TRACE BUFFER ENTRY 4

IRQ_ISR: LD D1,#$F0; IRQ Vector $FFF2 = TRACE BUFFER ENTRY 2
ST D1,VAR_C1
RTI;
```

The execution flow taking into account the IRQ is as follows

```
LDX,#SUB_1
MARK1:JMP(0,X);
IRQ_ISR: LD D1,#$F0;
STD1,VAR_C1
RTI;
SUB_1:NOP
NOP;
ADDR1:DBNED0,PART5;
```

The Normal mode trace buffer format is shown in the following tables. While tracing in Normal or Loop1 modes each array line contains two data entries, thus in this case the DBGCNT[0] is incremented after each separate entry. Information byte bits indicate if an entry is a source, destination or vector address.

The external event input can force trace buffer entries independent of COF occurrences, in which case the EEVI bit is set and the PC value of the last instruction is stored to the trace buffer. If the external event coincides with a COF buffer entry a single entry is made with the EEVI bit set.

Normal mode profiling with timestamp is possible when tracing from a single source by setting the STAMP bit in DBGTCRL. This results in a different format (see Table 408).

Table 408. Normal and loop1 mode trace buffer format without timestamp

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
CPU	CINF1	CPCH1	CPCM1	CPCL1	CINF0	CPCH0	CPCM0	CPCL0
	CINF3	CPCH3	CPCM3	CPCL3	CINF2	CPCH2	CPCM2	CPCL2

Table 409. Normal and loop1 mode trace buffer format with timestamp

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
CPU	Timestamp	Timestamp	Reserved	Reserved	CINF0	CPCH0	CPCM0	CPCL0
	Timestamp	Timestamp	Reserved	Reserved	CINF1	CPCH1	CPCM1	CPCL1

CINF contains information relating to the CPU.

CPU Information Byte CINF For Normal And Loop1 Modes

Table 410. CPU information byte CINF

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CET		0	0	CTI	EEVI	0	TOVF

Table 411. CINF bit descriptions

Field	Description
7–6 CET	**CPU Entry Type Field** — Indicates the type of stored address of the trace buffer entry as described in Table 412
3 CTI	**Comparator Timestamp Indicator** — This bit indicates if the trace buffer entry corresponds to a comparator timestamp. If a comparator D match occurs then a trace buffer entry is made independent of trace mode settings. 0 Trace buffer entry initiated by trace mode specification conditions or timestamp counter overflow 1 Trace buffer entry initiated by comparator D match
2 EEVI	**External Event Indicator** — This bit indicates if the trace buffer entry corresponds to an external event. 0 Trace buffer entry not initiated by an external event 1 Trace buffer entry initiated by an external event
0 TOVF	**Timestamp Overflow Indicator** — Indicates if the trace buffer entry corresponds to a timestamp overflow 0 Trace buffer entry not initiated by a timestamp overflow 1 Trace buffer entry initiated by a timestamp overflow

Table 412. CET encoding

CET	Entry Type Description
00	Non COF opcode address (entry forced by an external event)
01	Vector destination address
10	Source address of COF opcode
11	Destination address of COF opcode

6.10.4.5.2.2 Loop1 mode

Loop1 mode, similarly to Normal mode also stores only COF address information to the trace buffer, it however allows the filtering out of redundant information.

The intent of Loop1 mode is to prevent the trace buffer from being filled entirely with duplicate information from a looping construct such as delays using the DBNE instruction. The DBG monitors trace buffer entries and prevents consecutive duplicate address entries resulting from repeated branches.

Loop1 mode only inhibits consecutive duplicate source address entries that would typically be stored in most tight looping constructs. It does not inhibit repeated entries of destination addresses or vector addresses, since repeated entries of these could indicate a bug in application code that the DBG module is designed to help find.

The trace buffer format for Loop1 mode is the same as that of Normal mode.

6.10.4.5.2.3 Detail mode

When tracing CPU activity in Detail mode, address and data of data and vector accesses are traced. An information byte (CINF) indicates the size of access and the type of access (read or write).

ADRH, ADRM, and ADRL denote address high, middle and low byte respectively. The numerical suffix indicates which tracing step. DBGCNT increments by 2 for each line completed.

If timestamps are enabled, each CPU entry can span 2 trace buffer lines, whereby the second line includes the timestamp. If a valid PC occurs in the same cycle as the timestamp, it is also stored to the trace buffer and the PC bit is set. The second line featuring the timestamp is only stored if no further data access occurs in the following cycle. Table 414 shows where data accesses 2 and 3 occur in consecutive cycles, suppressing the entry2 timestamp. If 2 lines are used for an entry, then DBGCNT increments by 4. A timestamp line is indicated by bit1 in the TSINF byte.

The timestamp counter is only reset each time a timestamp line entry is made. It is not reset when the data and address trace buffer line entry is made.

Table 413. Detail mode trace buffer format without timestamp

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
CPU Detail	CDATA31	CDATA21	CDATA11	CDATA01	CINF1	CADRH1	CADRM1	CADRL1
	CDATA32	CDATA22	CDATA12	CDATA02	CINF2	CADRH2	CADRM2	CADRL2

Table 414. Detail mode trace buffer format with timestamp

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
CPU Detail	CDATA31	CDATA21	CDATA11	CDATA01	CINF1	CADRH1	CADRM1	CADRL1
	Timestamp	Timestamp	Reserved	Reserved	TSINF1	CPCH1	CPCM1	CPCL1
	CDATA32	CDATA22	CDATA12	CDATA02	CINF2	CADRH2	CADRM2	CADRL2
	CDATA33	CDATA23	CDATA13	CDATA03	CINF3	CADRH3	CADRM3	CADRL3
	Timestamp	Timestamp	Reserved	Reserved	TSINF3	CPCH3	CPCM3	CPCL3

Detail Mode data entries store the bytes aligned to the address of the MSB accessed (Byte1 Table 415). Accesses split across 32-bit boundaries are wrapped around.

Table 415. Detail mode data byte alignment

Access Address	Access Size	CDATA31	CDATA21	CDATA11	CDATA01
00	32-bit	Byte1	Byte2	Byte3	Byte4
01	32-bit	Byte4	Byte1	Byte2	Byte3
10	32-bit	Byte3	Byte4	Byte1	Byte2
11	32-bit	Byte2	Byte3	Byte4	Byte1
00	24-bit	Byte1	Byte2	Byte3	
01	24-bit		Byte1	Byte2	Byte3
10	24-bit	Byte3		Byte1	Byte2
11	24-bit	Byte2	Byte3		Byte1
00	16-bit	Byte1	Byte2		
01	16-bit		Byte1	Byte2	
10	16-bit			Byte1	Byte2
11	16-bit	Byte2			Byte1
00	8-bit	Byte1			
01	8-bit		Byte1		

Table 415. Detail mode data byte alignment (continued)

Access Address	Access Size	CDATA31	CDATA21	CDATA11	CDATA01
10	8-bit			Byte1	
11	8-bit				Byte1
			Denotes byte that is not accessed.		

CINF, TSINF Information Bytes

Table 416. Information bytes CINF and XINF

BYTE	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CINF	CSZ		CRW	0	0	0	0	0
TSINF	0	0	0	0	CTI	PC	1	TOVF

When tracing in Detail mode, CINF provides information about the type of CPU access being made TSINF provides information about a timestamp. Bit1 indicates if the byte is a TSINF byte.

Table 417. CINF field descriptions

Field	Description
7–6 CSZ	**Access Type Indicator** — This field indicates the CPU access size. 00 8-bit Access 01 16-bit Access 10 24-bit Access 11 32-bit Access
5 CRW	**Read/Write Indicator** — Indicates if the corresponding stored address corresponds to a read or write access. 0 Write Access 1 Read Access

\
Table 418. TSINF field descriptions

Field	Description
3 CTI	**Comparator Timestamp Indicator** — This bit indicates if the trace buffer entry corresponds to a comparator timestamp. If a comparator D match occurs then a trace buffer entry is made independent of trace mode settings. 0 Trace buffer entry initiated by trace mode specification conditions or timestamp counter overflow 1 Trace buffer entry initiated by comparator D match
2 PC	**Program Counter Valid Indicator** — Indicates if the PC entry is valid on the timestamp line. 0 Trace buffer entry does not include PC value 1 Trace buffer entry includes PC value
0 TOVF	**Timestamp Overflow Indicator** — Indicates if the trace buffer entry corresponds to a timestamp overflow 0 Trace buffer entry not initiated by a timestamp overflow 1 Trace buffer entry initiated by a timestamp overflow

6.10.4.5.2.4 Pure PC mode

In Pure PC mode, the PC addresses of all opcodes loaded into the execution stage, including illegal opcodes, are stored.

Tracing from a single source, compression is implemented to increase the effective trace depth. A compressed entry consists of the lowest PC byte only. A full entry consists of all PC bytes. If the PC remains in the same 256 byte range, then a compressed entry is made, otherwise a full entry is made. The full entry is always the last entry of a record.

Each trace buffer line consists of 7 payload bytes, PLB0-6, containing full or compressed CPU PC addresses and 1 information byte to indicate the type of entry (compressed or base address) for each payload byte.

Each trace buffer line is filled from right to left. The final entry on each line is always a base address, used as a reference for the previous entries on the same line.

Tracing from the CPU, a base address is typically stored in bytes[6:4], the other payload bytes may be compressed or complete addresses as indicated by the info byte bits.

Table 419. Pure PC mode trace buffer format single source

Mode	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
CPU	CXINF	BASE	BASE	BASE	PLB3	PLB2	PLB1	PLB0

If the info bit for byte3 indicates a full CPU PC address, whereby bytes[5:3] are used, the info bit mapped to byte[4] is redundant and the byte[6] is unused because a line overflow has occurred. Similarly a base address stored in bytes[4:2] causes line overflow, so bytes[6:5] are unused.

CXINF[6:4] indicate how many bytes in a line contain valid data, since tracing may terminate before a complete line has been filled.

CXINF Information Byte Source Tracing

CXINF

7	6	5	4	3	2	1	0
MAT	PLEC			NB3	NB2	NB1	NB0

Table 420. CXINF field descriptions

Field	Description
MAT	**Mid Aligned Trigger**— This bit indicates a mid aligned trigger position. When a mid aligned trigger occurs, the next trace buffer entry is a base address and the counter is incremented to a new line, independent of the number of bytes used on the current line. The MAT bit is set on the current line, to indicate the position of the trigger. When configured for begin or end aligned trigger, this bit has no meaning. NOTE: In the case when ARM and TRIG are simultaneously set together in the same cycle that a new PC value is registered, then this PC is stored to the same trace buffer line and MAT set. 0 Line filled without mid aligned trigger occurrence 1 Line last entry is the last PC entry before a mid aligned trigger
PLEC[2:0]	**Payload Entry Count**— This field indicates the number of valid bytes in the trace buffer line Binary encoding is used to indicate up to 7 valid bytes.
NBx	**Payload Compression Indicator**— This field indicates if the corresponding payload byte is the lowest byte of a base PC entry 0 Corresponding payload byte is a not the lowest byte of a base PC entry 1 Corresponding payload byte is the lowest byte of a base PC entry

Pure PC mode tracing does not support timestamps or external event entries.

6.10.4.5.3 Timestamp

When set, the STAMP bit in DBGTCRL configures the DBG to add a timestamp to trace buffer entries in Normal, Loop1 and Detail trace buffer modes. The timestamp is generated from a 16-bit counter and is stored to the trace buffer line each time a trace buffer entry is made.

The number of core clock cycles since the last entry equals the timestamp + 1. The core clock runs at twice the frequency of the bus clock. The timestamp of the first trace buffer entry is 0x0000. With timestamps enabled trace buffer entries are initiated in the following ways:

- according to the trace mode specification, for example, COF PC addresses in Normal mode
- on a timestamp counter overflow:
  If the timestamp counter reaches 0xFFFF, then a trace buffer entry is made, with timestamp= 0xFFFF and the timestamp overflow bit TOVF is set.
- on a match of comparator D:
  If STAMP and DSTAMP are set, then comparator D is used for forcing trace buffer entries with timestamps. The state control register settings determine if comparator D is also used to trigger the state sequencer. Thus if the state control register configuration does not use comparator D, then it is used solely for the timestamp function. If comparator D initiates a timestamp, then the CTI bit is set in the INFO byte. This can be used in Normal/Loop1 mode to find when a particular data access occurs relative to the PC flow. For example, when the timing of an access may be unclear due to the use of indexes.

Timestamps are disabled in Pure PC mode.

6.10.4.5.4 Reading data from trace buffer

The data stored in the trace buffer can be read using either the background debug controller (BDC) module or the CPU, provided the DBG module is not armed and is configured for tracing by TSOURCE. When the ARM bit is set, the trace buffer is locked to prevent reading. The trace buffer can only be unlocked for reading by an aligned word write to DBGTB when the module is disarmed. The trace buffer can only be read through the DBGTB register using aligned word reads. Reading the trace buffer while the DBG module is armed, or the trace buffer locked, returns 0xEE and no shifting of the RAM pointer occurs. Any byte or misaligned reads return 0xEE and does not cause the trace buffer pointer to increment to the next trace buffer address.

The trace buffer data is read out first-in first-out. By reading CNT in DBGCNT, the number of valid 64-bit lines can be determined. DBGCNT does not decrement as data is read. While reading, an internal pointer is used to determine the next line to be read. After a tracing session, the pointer points to the oldest data entry. If no overflow has occurred, the pointer points to line0. The pointer is initialized by each aligned write to DBGTB to point to the oldest data again. This enables an interrupted trace buffer read sequence to be easily restarted from the oldest data entry. After reading all trace buffer lines, the next read wraps around and returns the contents of line0.

The least significant word of each 64-bit wide array line is read out first. All bytes, including those containing invalid information are read out.

6.10.4.5.5 Trace buffer reset state

The trace buffer contents are not initialized by a system reset. Thus should a system reset occur, the trace session information from immediately before the reset occurred can be read out. The DBGCNT bits are not cleared by a system reset. Should a reset occur, the number of valid lines in the trace buffer is indicated by DBGCNT. The internal pointer is cleared by a system reset. It can be initialized by an aligned word write to DBGTB following a reset during debugging, so that it points to the oldest valid data again. Debugging occurrences of system resets are best handled using mid or end trigger alignment, since the reset may occur before the trace trigger, which in the beginning trigger alignment case means no information would be stored in the trace buffer.

6.10.4.6 Code profiling

6.10.4.6.1 Code profiling overview

Code profiling supplies encoded COF information on the PDO pin and the reference clock on the PDOCLK pin. Code profiling is enabled by setting the PROFILE bit and the associated device pin is configured for code profiling by setting the PDOE bit. Once enabled, code profiling is activated by arming the DBG. During profiling, if PDOE is set, the PDO operates as an output pin at a half the internal bus frequency, driving both high and low.

Independent of PDOE status, profiling data is stored to the trace buffer and can be read out when the debug session ends, in the same fashion as other trace buffer reads.

The external debugger uses both edges of the clock output to strobe the data on PDO. The first PDOCLK edge is used to sample the first data bit on PDO.

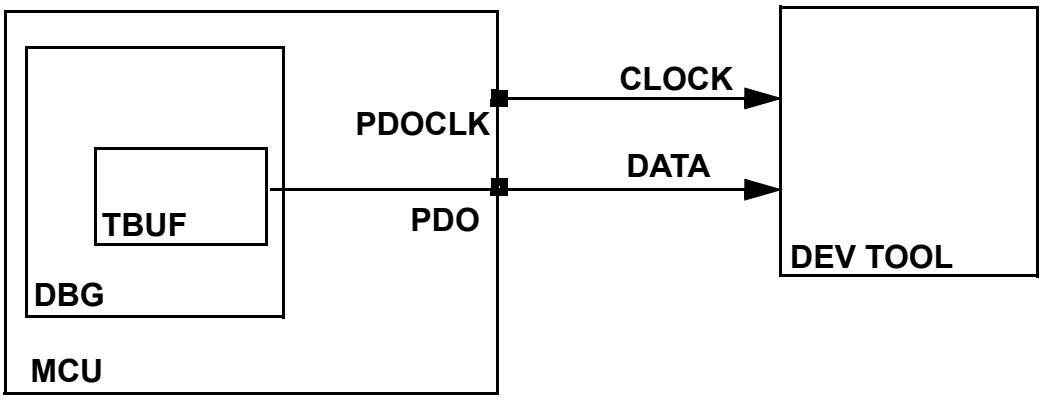


Figure 69. Profiling output interface

Figure 70 shows the profiling clock, PDOCLK, whose edges are offset from the bus clock, to ease setup and hold time requirements relative to PDO, which is synchronous to the bus clock.

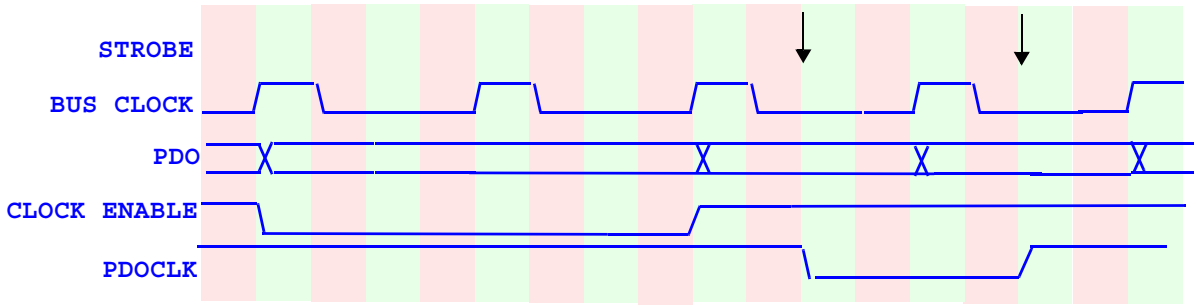


Figure 70. PDO profiling clock control

The trace buffer is used as a temporary storage medium to store COF information before it is transmitted. COF information can be transmitted while new information is written to the trace buffer. The trace buffer data is transmitted at PDO least significant bit first. After the first trace buffer entry is made, transmission begins in the first clock period in which no further data is written to the trace buffer.

If a trace buffer line transmission completes before the next trace buffer line is ready, the clock output is held at a constant level until the line is ready for transfer.

6.10.4.6.2 Profiling configuration, alignment and mode dependencies

The PROFILE bit must be set and the DBG armed to enable profiling. Furthermore, the PDOE bit must be set to configure the PDO and PDOCLK pins for profiling.

If TALIGN is configured for End-aligned tracing, profiling begins as soon as the module is armed.

If TALIGN is configured for Begin-aligned tracing, profiling begins when the state sequencer enters Final State and continues until a software disarm or trace buffer overflow occurs. Profiling does not terminate after 64 line entries have been made.

Mid-align tracing is not supported while profiling. If the TALIGN bits are configured for Mid-align tracing when PROFILE is set, the alignment defaults to end alignment.

Profiling entries continue until either a trace buffer overflow occurs or the DBG is disarmed by a state machine transition to State0. The profiling output transmission continues, even after disarming, until all trace buffer entries have been transmitted. The PTACT bit indicates if a profiling transmission is still active. The PTBOVF indicates if a trace buffer overflow has occurred.

The profiling timestamp feature is used only for the PTVB and PTW formats, differing from timestamps offered in other modes.

Profiling does not support trace buffer gating. The external pin gating feature is ignored during profiling.

When the DBG module is disarmed but profiling transmission is ongoing, register write accesses are suppressed.

When the DBG module is disarmed but profiling transmission is still ongoing, reading from the DBGTB returns the code 0xEE.

6.10.4.6.3 Code profiling internal data storage format

When profiling starts, the first trace buffer entry is made to provide the start address. This uses a 4-byte format (PTS), including the INFO byte and a 3-byte PC start address. In order to avoid trace buffer overflow a fully compressed format is used for direct (conditional branch) COF information.

Table 421. Profiling trace buffer line format

Format	8-Byte Wide Trace Buffer Line							
	7	6	5	4	3	2	1	0
PTS					PC Start Address			INFO
PTIB	Indirect	Indirect	Indirect	Direct	Direct	Direct	Direct	INFO
PTHF			0	Direct	Direct	Direct	Direct	INFO
PTVB	Timestamp	Timestamp	Vector	Direct	Direct	Direct	Direct	INFO
PTW	Timestamp	Timestamp	0	Direct	Direct	Direct	Direct	INFO

The INFO byte indicates the line format used. Up to 4-bytes of each line are dedicated to branch COFs. Further bytes are used for storing indirect COF information (indexed jumps and interrupt vectors). Indexed jumps force a full line entry with the PTIB format and require 3-bytes for the full 24-bit destination address. Interrupts force a full line entry with the PTVB format, whereby vectors are stored as a single byte and a 16-bit timestamp value is stored simultaneously to indicate the number of bus cycles relative to the previous COF. The 16-bit timestamp counter is cleared at each trace buffer entry. The device vectors use address[8:0] whereby address[1:0] are constant zero for vectors. The value stored to the PTVB vector byte is equivalent to (Vector Address[8:1]).

After the PTS entry, the pointer increments and the DBG begins to fill the next line with direct COF information. This continues until the direct COF field is full or an indirect COF occurs, then the INFO byte and, if needed, indirect COF information are entered on that line and the pointer increments to the next line.

If a timestamp overflow occurs, indicating a 65536 bus clock cycles without COF, then an entry is made with the TSOVF bit set, INFO[6] (Table 423) and profiling continues.

If a trace buffer overflow occurs, a final entry is made with the TBOVF bit set, profiling is terminated and the DBG is disarmed. Trace buffer overflow occurs when the trace buffer contains 64 lines pending transmission.

Whenever the DBG is disarmed during profiling, a final entry is made with the TERM bit set to indicate the final entry.

When a final entry is made then by default the PTW line format is used, except if a COF occurs in the same cycle in which case the corresponding PTIB/PTVB/PTHF format is used.

Since the development tool receives the INFO byte first, it can determine in advance the format of data it is about to receive. The transmission of the INFO byte starts when a line is complete. Whole bytes are always transmitted. The grey shaded bytes of Table 423 are not transmitted.

Table 422. INFO byte encoding

7	6	5	4	3	2	1	0
0	TSOVF	TBOVF	TERM	Line Format			

Table 423. Profiling format encoding

INFO[3:0]	Line Format	Source	Description
0000	PTS	CPU	Initial CPU entry
0001	PTIB	CPU	Indexed jump with up to 31 direct COFs
0010	PTHF	CPU	31 direct COFs without indirect COF
0011	PTVB	CPU	Vector with up to 31 direct COFs
0111	PTW	CPU	Error (Error codes in INFO[7:4])
Others	Reserved	CPU	Reserved
INFO[7:4]	**Bit Name**		**Description**
INFO[7]	Reserved	CPU	Reserved
INFO[6]	TSOVF	CPU	Timestamp Overflow
INFO[5]	TBOVF	CPU	Trace Buffer Overflow
INFO[4]	TERM	CPU	Profiling terminated by disarming
Vector[7:0]	Vector[7:0]	CPU	Device Interrupt Vector Address [8:1]

6.10.4.6.4 Direct COF compression

Each branch COF is stored to the trace buffer as a single bit (0 = branch not taken, 1 = branch taken) until an indirect COF (indexed jump, return, or interrupt) occurs. The branch COF entries are stored in the byte fields labelled “Direct” in Table 423. These entries start at byte1[0] and continue through to byte4[7], or until an indirect COF occurs, whichever occurs sooner. The entries use a format whereby the left most asserted bit is always the stop bit, which indicates that the bit to its right is the first direct COF and byte1[0] is the last COF that occurred before the indirect COF. This is shown in Table 424, whereby the Bytes 4 to 1 of the trace buffer are shown for three different cases. The stop bit field for each line is shaded.

In line0, the left most asserted bit is Byte4[7]. This indicates that all remaining 31 bits in the 4-byte field contain valid direct COF information, whereby each one represents branch taken and each 0 represents branch not taken. The stop bit of line1 indicates that all 30 bits to it’s right are valid, after the 30th direct COF entry, an indirect COF occurred, that is stored in bytes 7 to 5. In this case, the bit to the left of the stop bit is redundant. Line2 indicates that an indirect COF occurred after eight direct COF entries. The indirect COF address is stored in bytes 7 to 5. All bits to the left of the stop bit are redundant.

Table 424. Profiling direct COF format

Line	Byte4								Byte3								Byte2								Byte1							
Line0	1	0	0	1	0	0	1	0	0	1	0	1	1	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	1	1	0
Line1	0	1	1	0	0	1	0	1	1	0	0	1	0	0	1	0	1	1	0	0	1	0	0	1	0	1	1	0	0	1	0	0
Line2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1

6.10.4.7 Breakpoints

Breakpoints can be generated by state sequencer transitions to State0. State sequencer transitions to State0 follow automatically and immediately from Final State on tracing completion. Transitions to State0 are forced by the following events:

- Through comparator matches via Final State.
- Through software writing to the TRIG bit in the DBGC1 register via Final State.
- Through the external event input (DBGEEV) via Final State.
- Through a profiling trace buffer overflow event.

Breakpoints are not generated by software writes to DBGC1 that clear the ARM bit.

6.10.4.7.1 Breakpoints from comparator matches or external events

Breakpoints can be generated when the state sequencer transitions to State0 following a comparator match or an external event.

If a tracing session is selected by TSOURCE, the transition to State0 occurs when the tracing session has completed. If Begin or Mid aligned triggering is selected, the breakpoint is requested only on completion of the subsequent trace. If End aligned tracing or no tracing session is selected, the transition to State0 and associated breakpoints are immediate.

6.10.4.7.2 Breakpoints generated via the TRIG bit

When TRIG is written to “1”, the Final State is entered. If a tracing session is selected by TSOURCE, State0 is entered and breakpoints are requested only when the tracing session has completed. If Begin or Mid aligned triggering is selected, the breakpoint is requested only on completion of the subsequent trace. If no tracing session is selected, the state sequencer enters State0 immediately and breakpoints are requested. TRIG breakpoints are possible even if the DBG module is disarmed.

6.10.4.7.3 DBG breakpoint priorities

If a TRIG occurs after Begin or Mid aligned tracing has already been triggered by a comparator instigated transition to Final State, TRIG no longer has an effect. When the associated tracing session is complete, the breakpoint occurs. Similarly, if a TRIG is followed by a subsequent comparator match, it has no effect, since tracing has already started.

6.10.4.7.3.1 DBG breakpoint priorities and BDC interfacing

Breakpoint operation is dependent on the state of the S12ZBDC module. BDM cannot be entered from a breakpoint unless the BDC is enabled (ENBDC bit is set in the BDC). If BDM is already active, breakpoints are disabled. In addition, while executing a BDC STEP1 command, breakpoints are disabled.

When the DBG breakpoints are mapped to BDM (BDMBP set), then if a breakpoint request from either a BDC BACKGROUND command or a DBG event, coincides with an SWI instruction in application code, (i.e. the DBG requests a breakpoint at the next instruction boundary and the next instruction is an SWI) the CPU gives priority to the BDM request over the SWI request.

On returning from BDM, the SWI from user code gets executed. Breakpoint generation control is summarized in Table 425.

Table 425. Breakpoint mapping summary

BRKCPU	BDMBP Bit (DBGC1[4])	BDC Enabled	BDM Active	Breakpoint Mapping
0	X	X	X	No Breakpoint
1	0	X	0	Breakpoint to SWI
1	0	1	1	No Breakpoint
1	1	0	X	No Breakpoint
1	1	1	0	Breakpoint to BDM
1	1	1	1	No Breakpoint

6.10.5 Application information

6.10.5.1 Avoiding unintended breakpoint re-triggering

Returning from an instruction address breakpoint using an RTI or BDC GO command without PC modification, reverts to the instruction that generated the breakpoint. To avoid re-triggering a breakpoint at the same location, the BDC STEP1 command can be used, if configured for BDM breakpoints to increment the PC past the instruction.

If configured for SWI breakpoints, the DBG can be re configured in the SWI routine. If a comparator match occurs at an SWI vector address, then a code SWI and DBG breakpoint SWI could occur simultaneously. In this case, the SWI routine is executed twice before returning.

6.10.5.2 Debugging through reset

To debug through reset, the debugger can recognize a reset occurrence and pull the device BKGD pin low. This forces the device to leave reset in special single chip (SSC) mode, because the BKGD pin is used as the MODC signal in the reset phase. When the device leaves reset in SSC mode, CPU execution is halted and the device is in active BDM. The debugger can configure the DBG for tracing and breakpoints before returning to application code execution. In this way it is possible to analyze the sequence of events emerging from reset. The recommended handling of the internal reset scenario is as follows:

- When a reset occurs the debugger pulls BKGD low until the reset ends, forcing SSC mode entry.
- The debugger reads the reset flags to determine the cause of reset.
- If required, the debugger can read the trace buffer to see what happened just before reset. The trace buffer and DBGCNT register are not affected by resets other than POR.
- The debugger configures and arms the DBG to start tracing on returning to application code.
- The debugger then sets the PC according to the reset flags.
- Then the debugger returns to user code with GO or STEP1.

6.10.5.3 Breakpoints from other S12Z sources

The DBG is neither affected by CPU BGND instructions, nor by BDC BACKGROUND commands.

6.10.5.4 Code profiling

The code profiling data output pin PDO is mapped to a device pin that can also be used as GPIO in an application. If profiling is required and all pins are required in the application, it is recommended to use the device pin for a simple output function in the application, without feedback to the chip. The application can still be profiled, since the pin has no effect on code flow.

The PDO provides a simple bit stream that must be strobed at both edges of the profiling clock when profiling. The external development tool activates profiling by setting the DBG ARM bit, with PROFILE and PDOE already set. Thereafter, the first bit of the profiling bit stream is valid at the first rising edge of the profiling clock. No start bit is provided. The external development tool must detect this first rising edge after arming the DBG. To detect the end of profiling, the DBG ARM bit can be monitored using the BDC.

6.11 Background debug controller (S12ZBDCV1)

6.11.1 Introduction

The background debug controller (BDC) is a single-wire, background debug system implemented in on-chip hardware for minimal CPU intervention. The device BKGD pin interfaces directly to the BDC.

The S12ZBDC maintains the standard S12 serial interface protocol but introduces an enhanced handshake protocol and enhanced BDC command set to support the linear instruction set family of S12Z devices and offer easier, more flexible internal resource access over the BDC serial interface.

6.11.1.1 Glossary

Table 426. Glossary of terms

Term	Definition
DBG	On chip Debug Module
BDM	Active Background Debug mode
CPU	S12Z CPU
SSC	Special Single Chip mode (device operating mode
NSC	Normal Single Chip mode (device operating mode)
BDCSI	Background Debug Controller Serial Interface. This refers to the single pin BKGD serial interface.

6.11.1.2 Features

The BDC includes these distinctive features:

- Single-wire communication with host development system
- SYNC command to determine communication rate
- Genuine non-intrusive handshake protocol
- Enhanced handshake protocol for error detection and stop mode recognition
- Active out of reset in special single chip mode
- Most commands not requiring active BDM, for minimal CPU intervention
- Full global memory map access without paging
- Simple flash mass erase capability

6.11.1.3 Modes of operation

S12 devices feature power modes (run, wait, and stop) and operating modes (normal single chip, special single chip). Furthermore, the operation of the BDC is dependent on the device security status.

6.11.1.3.1 BDC modes

The BDC features module specific modes, namely disabled, enabled and active. These modes are dependent on the device security and operating mode. In active BDM, the CPU ceases execution, to allow BDC system access to all internal resources including CPU internal registers.

6.11.1.3.2 Security and operating mode dependency

In device run mode the BDC dependency is as follows:

- Normal modes, unsecure device

General BDC operation available. The BDC is disabled out of reset.

- Normal modes, secure device BDC disabled. No BDC access possible.
- Special Single Chip mode, unsecure

BDM active out of reset. All BDC commands are available.

- Special single chip mode, secure

BDM active out of reset. Restricted command set available.

When operating in secure mode, BDC operation is restricted to allow checking and clearing security by mass erasing the on-chip flash memory. Secure operation prevents BDC access to on-chip memory other than mass erase. The BDC command set is restricted to those commands classified as Always-available.

6.11.1.3.3 Low power modes

6.11.1.3.3.1 Stop mode

The execution of the CPU STOP instruction leads to stop mode only when all bus masters (CPU, or others, depending on the device) have finished processing. The operation during Stop mode depends on the ENBDC and BDCCIS bit settings, as summarized in Table 427

Table 427. BDC STOP operation dependencies

ENBDC	BDCCIS	Description Of Operation
0	0	BDC has no effect on STOP mode.
0	1	BDC has no effect on STOP mode.
1	0	Only BDCSI clock continues
1	1	All clocks continue

A disabled BDC has no influence on Stop mode operation. The BDCSI clock is disabled in Stop mode, so it is not possible to enable the BDC from within sTop mode.

If the BDC is enabled and BDCCIS is clear, then the BDC prevents the BDCCLK clock (Figure 72) from being disabled in Stop mode. This allows BDC communication to continue throughout Stop mode in order to access the BDCCSR register. All other device level clock signals are disabled on entering Stop mode.

Note

This is intended for application debugging, not for fast flash programming. The CLKSW bit must be clear to map the BDCSI to BDCCLK.

If handshaking is enabled, then the first ACK, following a stop mode entry is long to indicate a stop exception. Attempted internal accesses set the NORESP flag bit. The BDC indicates a Stop mode occurrence by setting the BDCCSR bit STOP.

If the BDC is enabled and BDCCIS is set, the BDC prevents clocks being disabled in Stop mode. This allows BDC communication, including access of internal memory mapped resources to continue throughout Stop mode. If handshaking is enabled, the first ACK, following a Stop mode entry, is long to indicate a stop exception. The BDC indicates a stop mode occurrence by setting the BDCCSR bit STOP.

6.11.1.3.3.2 Wait mode

The device enters Wait mode when the CPU starts to execute the WAI instruction. The second part of the WAI instruction (return from Wait mode) can only be performed when an interrupt occurs. On entering Wait mode, the CPU is in the middle of the WAI instruction and cannot permit access to CPU internal resources, nor allow entry to active BDM. Only commands classified as Non-intrusive or Always-available are possible in Wait mode.

On entering Wait mode, the WAIT flag in BDCCSR is set. If the ACK handshake protocol is enabled, the first ACK generated after WAIT has been set is a long-ACK pulse. The host can recognize a Wait mode occurrence. The WAIT flag remains set and cannot be cleared when the device remains in Wait mode. After the device leaves Wait mode the WAIT flag can be cleared by writing a “1” to it.

A BACKGROUND command, issued while in Wait mode with ACK disabled, sets the NORESP bit and the BDM active request remains pending internally until the CPU leaves Wait mode due to an interrupt. The device then enters BDM with the PC pointing to the address of the first instruction of the ISR. Further Non-intrusive or Always-available commands are possible in this pending state, but attempted Active-background commands set NORESP and ILLCMD because the BDC is not in active BDM state.

A BACKGROUND command, issued while in Wait mode with ACK enabled, remains pending internally until the CPU leaves Wait mode due to an interrupt. A long ACK is generated on leaving Wait mode. If the host attempts communication before the ACK pulse generation, the OVRUN bit is set.

6.11.1.4 Block diagram

A block diagram of the BDC is shown in Figure 71.

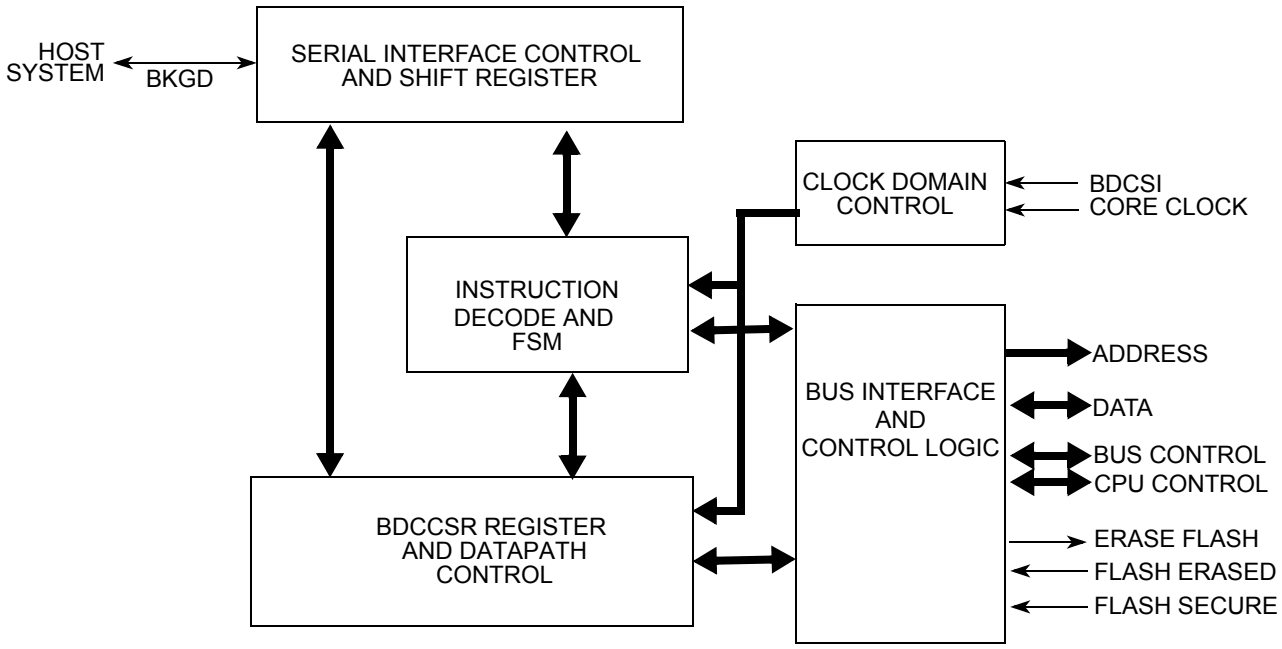


Figure 71. BDC block diagram

MM9Z1_638

6.11.2 External signal description

A single-wire interface pin (BKGD) is used to communicate with the BDC system. During reset, this pin is a device mode select input. After reset, this pin becomes the dedicated serial interface pin for the BDC.

BKGD is a pseudo-open-drain pin with an on-chip pull-up. Unlike typical open-drain pins, the external RC time constant on this pin due to external capacitance, plays almost no role in signal rise time. The custom protocol provides for brief, actively driven speed-up pulses to force rapid rise times on this pin without risking harmful drive level conflicts. Refer to BDC access of internal resources" for more details.

6.11.3 Memory map and register definition

6.11.3.1 Module memory map

Table 59 shows the BDC memory map.

Table 428. BDC memory map

Global Address	Module	Size (Bytes)
Not Applicable	BDC registers	2

6.11.3.2 Register descriptions

The BDC registers are shown in Table 429. Registers are accessed only by host-driven communications to the BDC hardware using READ_BDCCSR and WRITE_BDCCSR commands. They are not accessible in the device memory map.

Table 429. BDC register summary

Global Address	Register Name		Bit 7	6	5	4	3	2	1	Bit 0
Not Applicable	BDCCSRH	R	ENBDC	BDMACT	BDCCIS	0	STEAL	CLKSW	UNSEC	ERASE
		W								
Not Applicable	BDCCSRL	R	WAIT	STOP	RAMWF	OVRUN	NORESP	RDINV	ILLACC	ILLCMD
		W								
			= Unimplemented, Reserved				0	= Always read zero		

6.11.3.2.1 BDC control status register high (BDCCSRH)

Table 430. BDC control status register high (BDCCSRH)

	7	6	5	4	3	2	1	0
R	ENBDC	BDMACT	BDCCIS	0	STEAL	CLKSW	UNSEC	ERASE
W								
Reset								
Secure AND SSC-Mode	1	1	0	0	0	0	0	0
Unsecure AND SSC-Mode	1	1	0	0	0	0	1	0
Secure AND NSC-Mode	0	0	0	0	0	0	0	0
Unsecure AND NSC-Mode	0	0	0	0	0	0	1	0

= Unimplemented, Reserved

0 = Always read zero

Register Address: This register is not in the device memory map. It is accessible using BDC inherent addressing commands

Notes:

276.Read: All modes through BDC operation only.
Write: All modes through BDC operation only, when not secured, but subject to the following:
Bits 7,5,3 and 2 can only be written by WRITE_BDCCSR commands.
Bits 6, 1 and 0 cannot be written. They can only be updated by internal hardware.

Table 431. BDCCSRH field descriptions

Field	Description
7 ENBDC	**Enable BDC** — This bit controls whether the BDC is enabled or disabled. When enabled, active BDM can be entered and non-intrusive commands can be carried out. When disabled, active BDM is not possible and the valid command set is restricted. Further information is provided in Table 434. 0 BDC disabled 1 BDC enabled ENBDC is set out of reset in special single chip mode.
6 BDMACT	**BDM Active Status** — This bit becomes set upon entering active BDM. BDMACT is cleared as part of the active BDM exit sequence. 0 BDM not active 1 BDM active BDMACT is set out of reset in special single chip mode.
5 BDCCIS	**BDC Continue In Stop** — If ENBDC is set, then BDCCIS selects the type of BDC operation in Stop mode (as shown in Table 427). If ENBDC is clear, then the BDC has no effect on Stop mode and no BDC communication is possible. If ACK pulse handshaking is enabled, then the first ACK pulse following Stop mode entry is a long ACK. 0 Only the BDCSI clock continues in Stop mode 1 All clocks continue in Stop mode
3 STEAL	**Steal enabled with ACK**— This bit forces immediate internal accesses with the ACK handshaking protocol enabled. If ACK handshaking is disabled, then this bit has no effect. 0 If ACK is enabled then BDC accesses await a free cycle, with a timeout of 512 cycles 1 If ACK is enabled then BDC accesses are carried out in the next bus cycle
2 CLKSW	**Clock Switch** — The CLKSW bit controls the BDCSI clock source. This bit is initialized to “0” by each reset and can be written to “1”. Once it has been set, it can only be cleared by a reset. When setting CLKSW a minimum delay of 150 cycles at the initial clock speed must elapse before the next command can be sent. This guarantees that the start of the next BDC command uses the new clock for timing subsequent BDC communications. 0 BDCCLK used as BDCSI clock source 1 Device fast clock used as BDCSI clock source Refer to the device specification to determine which clock connects to the BDCCLK and fast clock inputs.
1 UNSEC	**Unsecure** — If the device is unsecure, the UNSEC bit is set automatically. 0 Device is secure. 1 Device is unsecure. When UNSEC is set, the device is unsecure and the state of the secure bits in the on-chip Flash EEPROM can be changed.
0 ERASE	**Erase Flash** — This bit can only be set by the dedicated ERASE_FLASH command. ERASE is unaffected by write accesses to BDCCSR. ERASE is cleared either when the mass erase sequence is completed, independent of the actual status of the flash array, or by a soft reset. Reading this bit indicates the status of the requested mass erase sequence. 0 No flash mass erase sequence pending completion 1 Flash mass erase sequence pending completion.

6.11.3.2.2 BDC control status register low (BDCCSRL)

Table 432. BDC Control Status Register Low (BDCCSRL)

	7	6	5	4	3	2	1	0
R W	WAIT	STOP	RAMWF	OVRUN	NORESP	RDINV	ILLACC	ILLCMD
Reset	0	0	0	0	0	0	0	0

Register Address: This register is not in the device memory map. It is accessible using BDC inherent addressing commands

Notes:
277.Read: BDC access only.
Write: Bits [7:5], [3:0] BDC access only, restricted to flag clearing by writing a “1” to the bit position.
Write: Bit 4 never. It can only be cleared by a SYNC pulse.

If ACK handshaking is enabled then BDC commands with ACK causing a BDCCSRL[3:1] flag setting condition also generate a long ACK pulse. Subsequent commands that are executed correctly generate a normal ACK pulse. Subsequent commands that are not correctly executed generate a long ACK pulse. The first ACK pulse after WAIT or STOP have been set also generates a long ACK. Subsequent ACK pulses are normal, while STOP and WAIT remain set.

Long ACK pulses are not immediately generated if an overrun condition is caused by the host driving the BKGD pin low while a target ACK is pending, because this would conflict with an attempted host transmission following the BKGD edge. When a whole byte has been received following the offending BKGD edge, the OVRUN bit is still set, forcing subsequent ACK pulses to be long.

Unimplemented BDC opcodes causing the ILLCMD bit to be set do not generate a long ACK because this could conflict with further transmission from the host. If the ILLCMD is set for another reason, then a long ACK is generated for the current command if it is a BDC command with ACK.

Table 433. BDCCSRL field descriptions

Field	Description
7 WAIT	**WAIT Indicator Flag** — Indicates that the device entered Wait mode. Writing a “1” to this bit while in Wait mode has no effect. Writing a “1” after exiting Wait mode, clears the bit. 0 Device did not enter Wait mode 1 Device entered Wait mode.
6 STOP	**STOP Indicator Flag** — Indicates that the CPU requested Stop mode following a STOP instruction. Writing a “1” to this bit while not in Stop mode clears the bit. Writing a “1” to this bit while in Stop mode has no effect. This bit can only be set when the BDC is enabled. 0 Device did not enter Stop mode 1 Device entered Stop mode.
5 RAMWF	**RAM Write Fault** — Indicates an ECC double fault during a BDC write access to RAM. Writing a “1” to this bit, clears the bit. 0 No RAM write double fault detected. 1 RAM write double fault detected.
4 OVRUN	**Overrun Flag** — Indicates unexpected host activity before command completion. This occurs if a new command is received before the current command completion. With ACK enabled, this also occurs if the host drives the BKGD pin low while a target ACK pulse is pending To protect internal resources from misinterpreted BDC accesses following an overrun, internal accesses are suppressed until a SYNC clears this bit. A SYNC clears the bit. 0 No overrun detected. 1 Overrun detected when issuing a BDC command.
3 NORESP	**No Response Flag** — Indicates that the BDC internal data access did not complete. This occurs if no free cycle for an access is found within 512 core clock cycles. This could typically happen if a code loop without free cycles is executing with ACK enabled and STEAL clear. With ACK disabled or STEAL set, this occurs when an internal access is not complete before the host starts data/BDCCSRL retrieval, or an internal write access is not complete before the host starts the next BDC command. On setting NORESP, the BDC aborts the access if permitted. (BDC external accesses with EWAIT assertions, prevent a command from being aborted until EWAIT is deasserted). If a BACKGROUND command is issued while the device is in Wait mode, the NORESP bit is set but the command is not aborted. The active BDM request is completed when the device leaves wait mode. Furthermore subsequent CPU register access commands during wait mode set the NORESP bit, should it have been cleared. When NORESP is set, a value of 0xEE is returned for each data byte associated with the current access. Writing a “1” to this bit, clears the bit. 0 Internal access completed. 1 Internal access did not complete.
2 RDINV	**Read Data Invalid Flag** — Indicates invalid read data due to an ECC error during a BDC initiated read access. The access returns the actual data read from the location. Writing a “1” to this bit, clears the bit. 0 No invalid read data detected. 1 Invalid data returned during a BDC read access.
1 ILLACC	**Illegal Access Flag** — Indicates an attempted illegal access. This is set in the following cases: When the attempted access addresses unimplemented memory When the access attempts to write to the flash array When a CPU register access is attempted with an invalid CRN (BDC access of CPU registers). Illegal accesses return a value of 0xEE for each data byte. Writing a “1” to this bit, clears the bit. 0 No illegal access detected. 1 Illegal BDC access detected.
0 ILLCMD	**Illegal Command Flag** — Indicates an illegal BDC command. This bit is set in the following cases: When an unimplemented BDC command opcode is received. When a DUMP_MEM{_WS}, FILL_MEM{_WS} or READ_SAME{_WS} is attempted in an illegal sequence. When an active BDM command is received while BDM is not active When a non Always-available command is received while the BDC is disabled or a flash mass erase is ongoing. When a non Always-available command is received while the device is secure Read commands return a value of 0xEE for each data byte Writing a “1” to this bit, clears the bit. 0 No illegal command detected. 1 Illegal BDC command detected.

6.11.4 Functional description

6.11.4.1 Security

If the device resets with the system secured, the device clears the BDCCSR UNSEC bit. In the secure state BDC access is restricted to the BDCCSR register. A mass erase can be requested using the ERASE_FLASH command. If the mass erase is completed successfully, the device programs the security bits to the unsecure state and sets the BDC UNSEC bit. If the mass erase is unsuccessful, the device remains secure and the UNSEC bit is not set.

For more information regarding security, Refer to device specific security information.

6.11.4.2 Enabling BDC and entering active BDM

BDM can be activated only after being enabled. BDC is enabled by setting the ENBDC bit in the BDCCSR register, via the single-wire interface, using the hardware command WRITE_BDCCSR.

After being enabled, BDM is activated by one of the following:

- The BDC BACKGROUND command
- A CPU BGND instruction
- The DBG Breakpoint mechanism

Notes:
278.BDM active immediately out of special single-chip reset

Alternatively BDM can be activated directly from reset when resetting into Special Single Chip mode.

The BDC is ready for receiving the first command 10 core clock cycles after the deassertion of the internal reset signal. This is delayed relative to the external pin reset as specified in the device reset documentation. On S12Z devices, an NVM initialization phase follows reset. During this phase the BDC commands classified as always-available are carried out immediately, whereas other BDC commands are subject to delayed response due to the NVM initialization phase.

When BDM is activated, the CPU finishes executing the current instruction. Thereafter only BDC commands can affect CPU register contents until the BDC GO command returns from active BDM to user code or a device reset occurs. When BDM is activated by a breakpoint, the type of breakpoint used determines if BDM becomes active before or after execution of the next instruction.

Note

When attempting to activate BDM using a BGND instruction while the BDC is disabled, the CPU requires clock cycles for the attempted BGND execution. BACKGROUND commands issued while the BDC is disabled are ignored by the BDC and the CPU execution is not delayed.

6.11.4.3 Clock source

The BDC clock source can be mapped to a constant frequency clock source or a PLL based fast clock. The clock source for the BDC is selected by the CLKSW bit as shown in Figure 72. The BDC internal clock is named BDCSI clock. If BDCSI clock is mapped to the BDCCLK by CLKSW then the serial interface communication is not affected by bus/core clock frequency changes. If the BDC is mapped to BDCFCLK then the clock is connected to a PLL derived source at device level (typically bus clock), thus can be subject to frequency changes in application. Debugging through frequency changes requires SYNC pulses to resychronize. The sources of BDCCLK and BDCFCLK are specified at device level.

BDC accesses of internal device resources always use the device core clock. Thus if the ACK handshake protocol is not enabled, the clock frequency relationship must be taken into account by the host.

When changing the clock source via the CLKSW bit a minimum delay of 150 cycles at the initial clock speed must elapse before a SYNC can be sent. This guarantees that the start of the next BDC command uses the new clock for timing subsequent BDC communications.

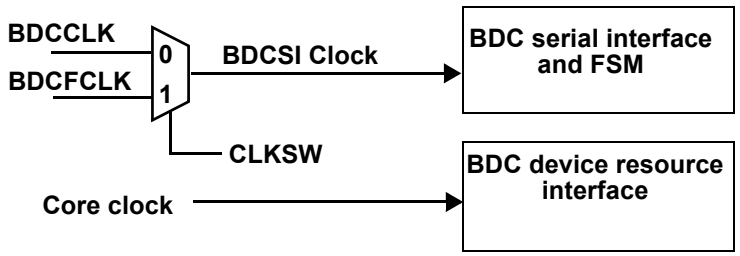


Figure 72. Clock switch

6.11.4.4 BDC commands

BDC commands can be classified into three types as shown in Table 434.

Table 434. BDC command types

Command Type	Secure Status	BDC Status	CPU Status	Command Set
Always-available	Secure or Unsecure	Enabled or Disabled	—	Read/write access to BDCCSR Mass erase flash memory using ERASE_FLASH SYNC ACK enable/disable
Non-intrusive	Unsecure	Enabled	Code execution allowed	Read/write access to BDCCSR Memory access Memory access with status Mass erase flash memory using ERASE_FLASH Debug register access BACKGROUND SYNC ACK enable/disable
Active background	Unsecure	Active	Code execution halted	Read/write access to BDCCSR Memory access Memory access with status Mass erase flash memory using ERASE_FLASH Debug register access Read or write CPU registers Single-step the application Exit active BDM to return to the application program (GO) SYNC ACK enable/disable

Non-intrusive commands are used to read and write target system memory locations and to enter active BDM. Target system memory includes all memory and registers within the global memory map, including external memory.

Active background commands are used to read and write all memory locations and CPU resources. Furthermore they allow single stepping through application code and to exit from active BDM.

Non-intrusive commands can only be executed when the BDC is enabled and the device unsecure. Active background commands can only be executed when the system is not secure and is in active BDM.

Non-intrusive commands do not require the system to be in active BDM for execution, although, they can still be executed in this mode. When executing a non-intrusive command with the ACK pulse handshake protocol disabled, the BDC steals the next bus cycle for the access. If an operation requires multiple cycles, then multiple cycles can be stolen. If stolen cycles are not free cycles, the application code execution is delayed. The delay is negligible because the BDC serial transfer rate dictates that such accesses occur infrequently.

For data read commands, the external host must wait at least 16 BDCSI clock cycles after sending the address before attempting to obtain the read data. This is to be certain that valid data is available in the BDC shift register, ready to be shifted out. For write commands, the external host must wait 16 bdcsi cycles after sending the data to be written before attempting to send a new command. This is to avoid disturbing the BDC shift register before the write has been completed. The external host must wait at least for 16 bdcsi cycles after a control command before starting any new serial command.

If the ACK pulse handshake protocol is enabled and STEAL is cleared, then the BDC waits for the first free bus cycle to make a non-intrusive access. If no free bus cycle occurs within 512 core clock cycles then the BDC aborts the access, sets the NORESP bit and uses a long ACK pulse to indicate an error condition to the host.

Table 435 summarizes the BDC command set. The subsequent sections describe each command in detail and illustrate the command structure in a series of packets, each consisting of eight bit times starting with a falling edge. The bar across the top of the blocks indicates that the BKGD line idles in the high state. The time for an 8-bit command is 8 × 16 target BDCSI clock cycles.

The nomenclature below is used to describe the structure of the BDC commands. Commands begin with an 8-bit hexadecimal command code in the host-to-target direction (most significant bit first)

```
/=separates parts of the command
d=delay 16 target BDCSI clock cycles (DLY)
dack =delay (16 cycles) no ACK; or delay (=> 32 cycles) then ACK.(DACK)
ad24=24-bit memory address in the host-to-target direction
rd8=8 bits of read data in the target-to-host direction
rd16=16 bits of read data in the target-to-host direction
rd24=24 bits of read data in the target-to-host direction
rd32=32 bits of read data in the target-to-host direction
rd64=64 bits of read data in the target-to-host direction
rd.sz=read data, size defined by sz, in the target-to-host direction
wd8=8 bits of write data in the host-to-target direction
wd16=16 bits of write data in the host-to-target direction
wd32=32 bits of write data in the host-to-target direction
wd.sz=write data, size defined by sz, in the host-to-target direction
ss=the contents of BDCCSRL in the target-to-host direction
sz=memory operand size (00 = byte, 01 = word, 10 = long)
(sz = 11 is reserved and currently defaults to long)
crn=core register number, 32-bit data width
WS=command suffix signaling the operation is with status
```

Table 435. BDC command summary

Command Mnemonic	Command Classification	ACK	Command Structure	Description
SYNC	Always Available	N/A	N/A [(279)]	Request a timed reference pulse to determine the target BDC communication speed
ACK_DISABLE	Always Available	No	0x03/d	Disable the communication handshake. This command does not issue an ACK pulse.
ACK_ENABLE	Always Available	Yes	0x02/dack	Enable the communication handshake. Issues an ACK pulse after the command is executed.
BACKGROUND	Non-intrusive	Yes	0x04/dack	Halt the CPU if ENBDC is set. Otherwise, ignore as illegal command.
DUMP_MEM.sz	Non-intrusive	Yes	(0x32+4 x sz)/dack/rd.sz	Dump (read) memory based on operand size (sz). Used with READ_MEM to dump large blocks of memory. An initial READ_MEM is executed to set up the starting address of the block and to retrieve the first result. Subsequent DUMP_MEM commands retrieve sequential operands.
DUMP_MEM.sz_WS	Non-intrusive	No	(0x33+4 x sz)/d/ss/rd.sz	Dump (read) memory based on operand size (sz) and report status. Used with READ_MEM{_WS} to dump large blocks of memory. An initial READ_MEM{_WS} is executed to set up the starting address of the block and to retrieve the first result. Subsequent DUMP_MEM{_WS} commands retrieve sequential operands.

Table 435. BDC command summary (continued)

Command Mnemonic	Command Classification	ACK	Command Structure	Description
FILL_MEM.sz	Non-intrusive	Yes	(0x12+4 x sz)/wd.sz/dack	Fill (write) memory based on operand size (sz). Used with WRITE_MEM to fill large blocks of memory. An initial WRITE_MEM is executed to set up the starting address of the block and to write the first operand. Subsequent FILL_MEM commands write sequential operands.
FILL_MEM.sz_WS	Non-intrusive	No	(0x13+4 x sz)/wd.sz/d/ss	Fill (write) memory based on operand size (sz) and report status. Used with WRITE_MEM{_WS} to fill large blocks of memory. An initial WRITE_MEM{_WS} is executed to set up the starting address of the block and to write the first operand. Subsequent FILL_MEM{_WS} commands write sequential operands.
GO	Active Background	Yes	0x08/dack	Resume CPU user code execution
GO_UNTIL (280)	Active Background	Yes	0x0C/dack	Go to user program. ACK is driven upon returning to active background mode.
NOP	Non-intrusive	Yes	0x00/dack	No operation
READ_Rn	Active Background	Yes	(0x60+CRN)/dack/rd32	Read the requested CPU register
READ_MEM.sz	Non-intrusive	Yes	(0x30+4 x sz)/ad24/dack/rd.sz	Read the appropriately-sized (sz) memory value from the location specified by the 24-bit address
READ_MEM.sz_WS	Non-intrusive	No	(0x31+4 x sz)/ad24/d/ss/rd.sz	Read the appropriately-sized (sz) memory value from the location specified by the 24-bit address and report status
READ_DBGTB	Non-intrusive	Yes	(0x07)/dack/rd32/dack/rd32	Read 64-bits of DBG trace buffer
READ_SAME.sz	Non-intrusive	Yes	(0x50+4 x sz)/dack/rd.sz	Read from location. An initial READ_MEM defines the address, subsequent READ_SAME reads return content of same address
READ_SAME.sz_WS	Non-intrusive	No	(0x51+4 x sz)/d/ss/rd.sz	Read from location. An initial READ_MEM defines the address, subsequent READ_SAME reads return content of same address
READ_BDCCSR	Always Available	No	0x2D/rd16	Read the BDCCSR register
SYNC_PC	Non-intrusive	Yes	0x01/dack/rd24	Read current PC
WRITE_MEM.sz	Non-intrusive	Yes	(0x10+4 x sz)/ad24/wd.sz/dack	Write the appropriately-sized (sz) memory value to the location specified by the 24-bit address
WRITE_MEM.sz_WS	Non-intrusive	No	(0x11+4 x sz)/ad24/wd.sz/d/ss	Write the appropriately-sized (sz) memory value to the location specified by the 24-bit address and report status
WRITE_Rn	Active Background	Yes	(0x40+CRN)/wd32/dack	Write the requested CPU register
WRITE_BDCCSR	Always Available	No	0x0D/wd16	Write the BDCCSR register
ERASE_FLASH	Always Available	No	0x95/d	Mass erase internal flash
STEP1 (TRACE1)	Active Background	Yes	0x09/dack	Execute one CPU command.

Notes:
279.The SYNC command is a special operation which does not have a command code.
280.The GO_UNTIL command is identical to the GO command if ACK is not enabled.

6.11.4.4.1 SYNC

The SYNC command is unlike other BDC commands because the host does not necessarily know the correct speed to use for serial communications until after it has analyzed the response to the SYNC command.

To issue a SYNC command, the host:

1. Ensures that the BKGD pin is high for at least four cycles of the slowest possible BDCSI clock without reset asserted.
2. Drives the BKGD pin low for at least 128 cycles of the slowest possible BDCSI clock.
3. Drives BKGD high for a brief speed-up pulse to get a fast rise time. (This speed-up pulse is typically one cycle of the host clock, which is as fast as the maximum target BDCSI clock).
4. Removes all drive to the BKGD pin so it reverts to high-impedance.
5. Listens to the BKGD pin for the sync response pulse.

Upon detecting the sync request from the host (which is a much longer low time than would ever occur during normal BDC communications), the target:

1. Discards any incomplete command
2. Waits for BKGD to return to a logic high.
3. Delays 16 cycles to allow the host to stop driving the high speed-up pulse.
4. Drives BKGD low for 128 BDCSI clock cycles.
5. Drives a 1-cycle high speed-up pulse to force a fast rise time on BKGD.
6. Removes all drive to the BKGD pin so it reverts to high-impedance.
7. Clears the OVRRUN flag (if set).

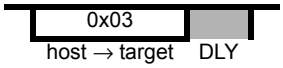
The host measures the low time of this 128-cycle SYNC response pulse and determines the correct speed for subsequent BDC communications. Typically, the host can determine the correct communication speed within a few percent of the actual target speed and the serial protocol can easily tolerate this speed error.

If the SYNC request is detected by the target, any partially executed command is discarded. This is referred to as a soft-reset, equivalent to a timeout in the serial communication. After the SYNC response, the target interprets the next negative edge (issued by the host) as the start of a new BDC command or the start of new SYNC request.

A SYNC command can also be used to abort a pending ACK pulse. This is explained in Hardware handshake abort procedure.

6.11.4.4.2 ACK_DISABLE

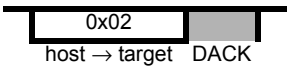
Disable host/target handshake protocol **Always Available**



Disables the serial communication handshake protocol. The subsequent commands, issued after the ACK_DISABLE command, do not execute the hardware handshake protocol. This command is not followed by an ACK pulse.

6.11.4.4.3 ACK_ENABLE

Enable host/target handshake protocol **Always Available**

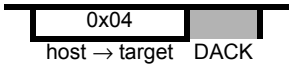


Enables the hardware handshake protocol in the serial communication. The hardware handshake is implemented by an acknowledge (ACK) pulse issued by the target MCU in response to a host command. The ACK_ENABLE command is interpreted and executed in the BDC logic without the need to interface with the CPU. An ACK pulse is issued by the target device after this command is executed. This command can be used by the host to evaluate if the target supports the hardware handshake protocol. If the target supports the hardware handshake protocol, subsequent commands are enabled to execute the hardware handshake protocol, otherwise this command is ignored by the target. Table 435 indicates which commands support the ACK hardware handshake protocol.

For additional information about the hardware handshake protocol, refer to Serial interface hardware handshake (ACK pulse) protocol," and Hardware handshake abort procedure."

6.11.4.4.4 Background

Enter active background mode (if enabled) **Non-intrusive**



Provided ENBDC is set, the BACKGROUND command causes the target MCU to enter active BDM as soon as the current CPU instruction finishes. If ENBDC is cleared, the BACKGROUND command is ignored.

A delay of 16 BDCSI clock cycles is required after the BACKGROUND command to allow the target MCU to finish its current CPU instruction, and enter active background mode before a new BDC command can be accepted.

The host debugger must set ENBDC before attempting to send the BACKGROUND command the first time. Normally the host sets ENBDC once at the beginning of a debug session or after a target system reset. During debugging, the host uses GO commands to move from active BDM to application program execution and uses the BACKGROUND command or DBG breakpoints to return to active BDM.

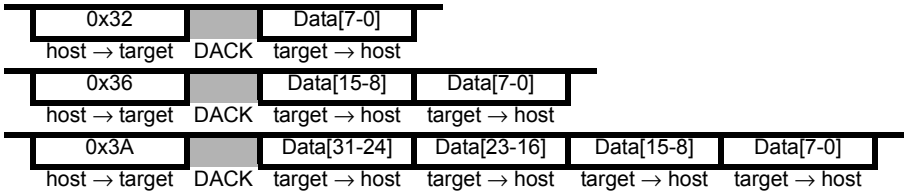
A BACKGROUND command issued during Wait mode cannot immediately force active BDM, because the WAI instruction does not end until an interrupt occurs. The BDM entry is stalled by wait, but remains pending, the NORESP bit is set and, if enabled, a long ACK is returned. When an interrupt brings the device out of Wait mode, the WAI instruction is completed and the pending BDM request is applied with the CPU PC pointing to the address of the first instruction of the interrupt service routine. A further ACK related to the BACKGROUND request is not generated.

The host can recognize this pending BDM request condition because both NORESP and WAIT are set, but BDMACT is clear. While in Wait mode, with the pending BDM request, non-intrusive BDC commands are allowed.

6.11.4.4.5 DUMP_MEM.sz, DUMP_MEM.sz_WS

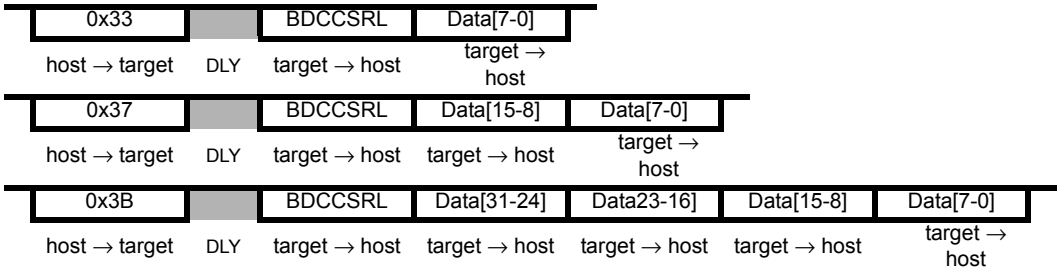
DUMP_MEM.sz

Read memory specified by debug address register, then increment address **Non-intrusive**



DUMP_MEM.sz_WS

Read memory specified by debug address register with status, then increment address **Non-intrusive**



DUMP_MEM{_WS} is used with the READ_MEM{_WS} command to access large blocks of memory. An initial READ_MEM{_WS} is executed to set-up the starting address of the block and to retrieve the first result. The DUMP_MEM{_WS} command retrieves subsequent operands. The initial address is incremented by the operand size (1, 2, or 4) and saved in a temporary register. Subsequent DUMP_MEM{_WS} commands use this address, perform the memory read, increment it by the current operand size, and store the updated address in the temporary register. If the with-status option is specified, the BDCCSRL status byte is returned before the read data. This status byte reflects the state after the memory read was performed. If enabled, an ACK pulse is driven before the data bytes are transmitted. The effect of the access size and alignment on the next address to be accessed is explained in more detail in BDC access of device memory mapped resources.

Note

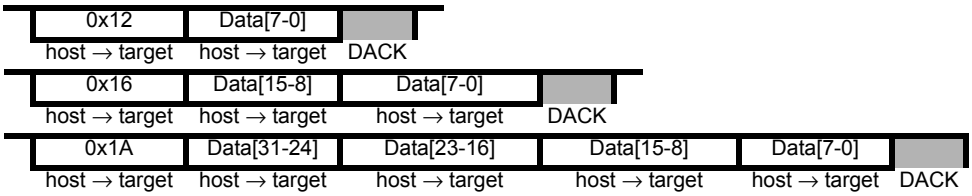
DUMP_MEM{_WS} is a valid command only when preceded by SYNC, NOP, READ_MEM{_WS}, or another DUMP_MEM{_WS} command. Otherwise, an illegal command response is returned, setting the ILLCMD bit. NOP can be used for inter-command padding without corrupting the address pointer.

The size field (sz) is examined each time a DUMP_MEM{_WS} command is processed, allowing the operand size to be dynamically altered. The examples show the DUMP_MEM.B{_WS}, DUMP_MEM.W{_WS} and DUMP_MEM.L{_WS} commands.

6.11.4.4.6 FILL_MEM.sz, FILL_MEM.sz_WS

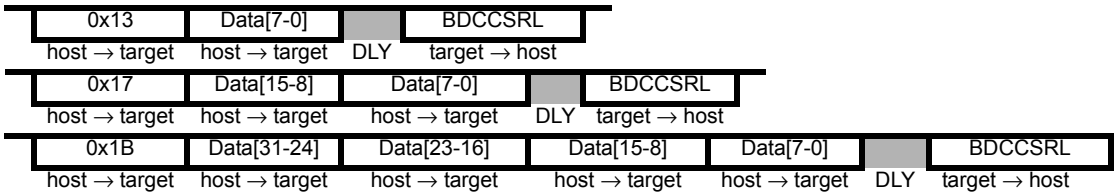
FILL_MEM.sz

Write memory specified by debug address register, then increment address **Non-intrusive**



FILL_MEM.sz_WS

Write memory specified by debug address register with status, then increment address **Non-intrusive**



FILL_MEM{_WS} is used with the WRITE_MEM{_WS} command to access large blocks of memory. An initial WRITE_MEM{_WS} is executed to set up the starting address of the block and write the first datum. If an initial WRITE_MEM{_WS} is not executed before the first FILL_MEM{_WS}, an illegal command response is returned. The FILL_MEM{_WS} command stores subsequent operands. The initial address is incremented by the operand size (1, 2, or 4) and saved in a temporary register. Subsequent FILL_MEM{_WS} commands use this address, perform the memory write, increment it by the current operand size, and store the updated address in the temporary register. If the with-status option is specified, the BDCCSRL status byte is returned after the write data. This status byte reflects the state after the memory write was performed. If enabled an ACK pulse is generated after the internal write access has been completed or aborted. The effect of the access size and alignment on the next address to be accessed is explained in more detail in BDC access of device memory mapped resources.

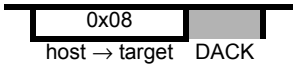
Note

FILL_MEM{_WS} is a valid command only when preceded by SYNC, NOP, WRITE_MEM{_WS}, or another FILL_MEM{_WS} command. Otherwise, an illegal command response is returned, setting the ILLCMD bit. NOP can be used for inter command padding without corrupting the address pointer.

The size field (sz) is examined each time a FILL_MEM{_WS} command is processed, allowing the operand size to be dynamically altered. The examples show the FILL_MEM.B{_WS}, FILL_MEM.W{_WS} and FILL_MEM.L{_WS} commands.

6.11.4.4.7 Go

Go **Non-intrusive**



This command is used to exit active BDM and begin (or resume) execution of CPU application code. The CPU pipeline is flushed and refilled before normal instruction execution resumes. Prefetching begins at the current address in the PC. If any register (such as the PC) is altered by a BDC command while in BDM, the updated value is used when prefetching resumes. If enabled, an ACK is driven on exiting active BDM.

If a GO command is issued while the BDM is inactive, an illegal command response is returned and the ILLCMD bit is set.

6.11.4.4.8 GO_UNTIL

This command is used to exit active BDM and begin (or resume) execution of application code. The CPU pipeline is flushed and refilled before normal instruction execution resumes. Prefetching begins at the current address in the PC. If any register (such as the PC) is altered by a BDC command while in BDM, the updated value is used when prefetching resumes.

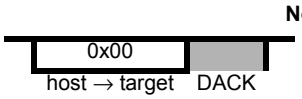
After resuming application code execution, if ACK is enabled, the BDC awaits a return to active BDM before driving an ACK pulse. Timeouts do not apply when awaiting a GO_UNTIL command ACK.

If a GO_UNTIL is not acknowledged then a SYNC command must be issued to end the pending GO_UNTIL.

If a GO_UNTIL command is issued while BDM is inactive, an illegal command response is returned and the ILLCMD bit is set.

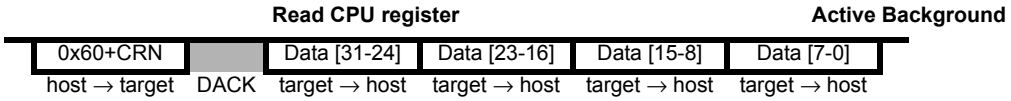
If ACK handshaking is disabled, the GO_UNTIL command is identical to the GO command.

6.11.4.4.9 NOP



NOP performs no operation and may be used as a null command where required.

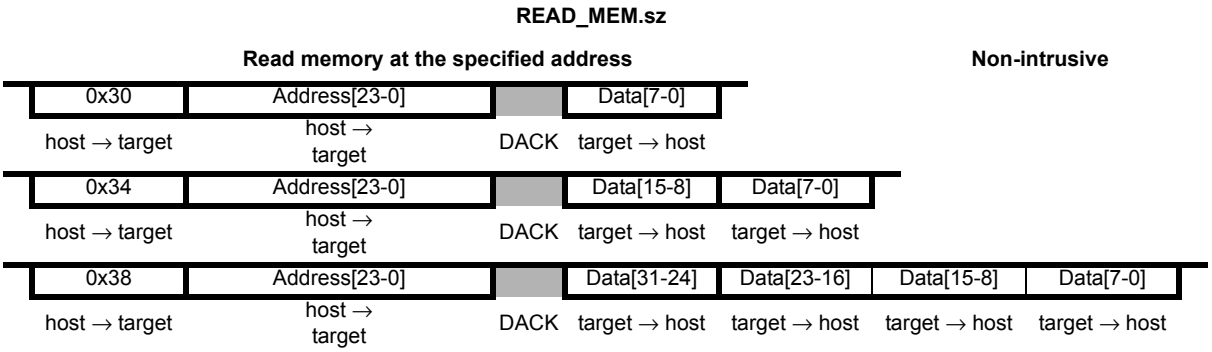
6.11.4.4.10 READ_Rn



This command reads the selected CPU registers and returns the 32-bit result. Accesses to CPU registers are always 32-bits wide, regardless of implemented register width. Bytes that are not implemented return zero. The register is addressed through the CPU register number (CRN). See BDC access of CPU registers for the CRN address decoding. If enabled, an ACK pulse is driven before the data bytes are transmitted.

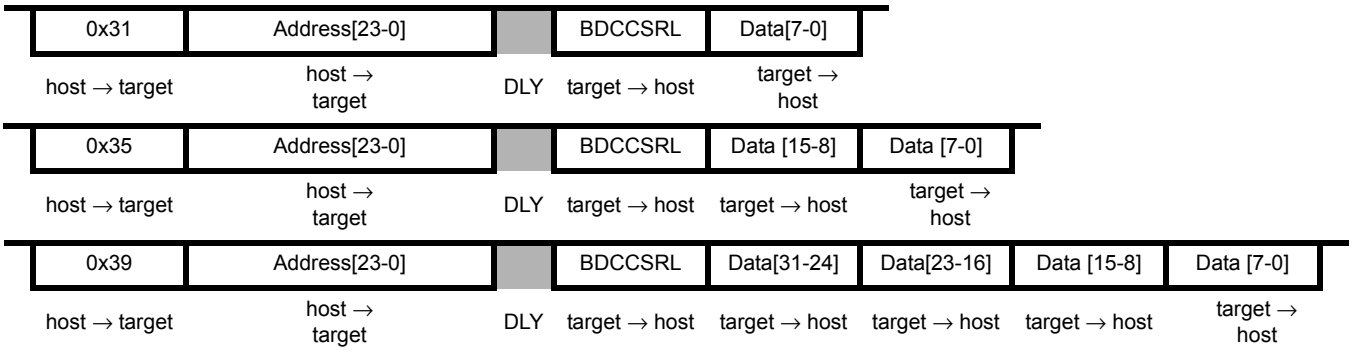
If the device is not in active BDM, this command is illegal, the ILLCMD bit is set and no access is performed.

6.11.4.4.11 READ_MEM.sz, READ_MEM.sz_WS



READ_MEM.sz_WS

Read memory at the specified address with status **Non-intrusive**



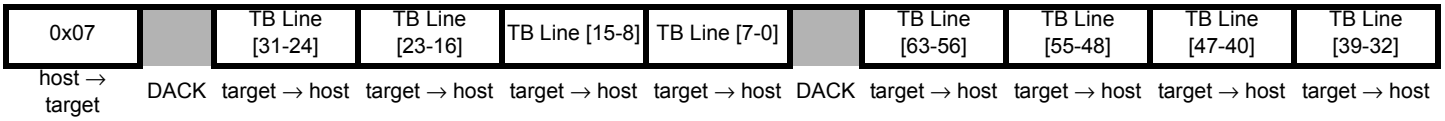
Read data at the specified memory address. The address is transmitted as three 8-bit packets (msb to lsb) immediately after the command.

The hardware forces low-order address bits to zero longword accesses to ensure these accesses are on 0-modulo-size alignments. Byte alignment details are described in BDC access of device memory mapped resources". If the with-status option is specified, the BDCCSR status byte is returned before the read data. This status byte reflects the state after the memory read was performed. If enabled, an ACK pulse is driven before the data bytes are transmitted.

The examples show the READ_MEM.B{_WS}, READ_MEM.W{_WS} and READ_MEM.L{_WS} commands.

6.11.4.4.12 READ_DBGTB

Read DBG trace buffer **Non-intrusive**

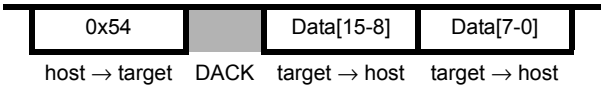


Read 64 bits from the DBG trace buffer. Refer to the DBG module description for more detailed information. If enabled an ACK pulse is generated before each 32-bit long word is ready to be read by the host. After issuing the first ACK a timeout is still possible while accessing the second 32-bit long word, since this requires separate internal accesses. The first 32-bit long word corresponds to trace buffer line bits[31:0]; the second to trace buffer line bits[63:32]. If ACK handshaking is disabled, the host must wait 16 clock cycles (DLY) after completing the first 32-bit read before starting the second 32-bit read.

6.11.4.4.13 READ_SAME.sz, READ_SAME.sz_WS

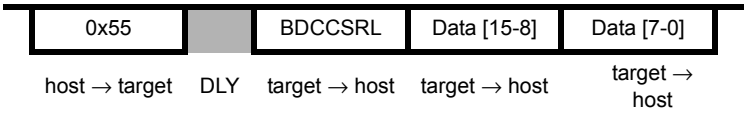
READ_SAME

Read same location specified by previous READ_MEM{_WS} **Non-intrusive**



READ_SAME_WS

Read same location specified by previous READ_MEM{_WS} **Non-intrusive**



MM9Z1_638

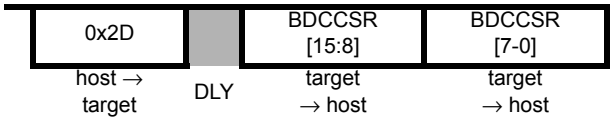
Read from location defined by the previous READ_MEM. The previous READ_MEM command defines the address, subsequent READ_SAME commands return contents of same address. The example shows the sequence for reading a 16-bit word size. Byte alignment details are described in BDC access of device memory mapped resources". If enabled, an ACK pulse is driven before the data bytes are transmitted.

Note

READ_SAME{_WS} is a valid command only when preceded by SYNC, NOP, READ_MEM{_WS}, or another READ_SAME{_WS} command. Otherwise, an illegal command response is returned, setting the ILLCMD bit. NOP can be used for inter-command padding without corrupting the address pointer.

6.11.4.4.14 READ_BDCCSR

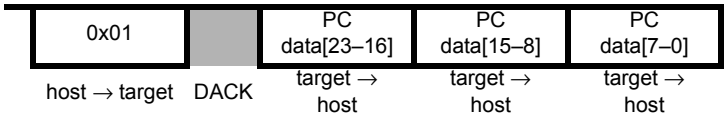
Read BDCCSR Status Register **Always Available**



Read the BDCCSR status register. This command can be executed in any mode.

6.11.4.4.15 SYNC_PC

Sample current PC **Non-intrusive**



This command returns the 24-bit CPU PC value to the host. Unsuccessful SYNC_PC accesses return 0xEE for each byte. If enabled, an ACK pulse is driven before the data bytes are transmitted. The value of 0xEE is returned if a timeout occurs, whereby NORESP is set. This can occur if the CPU is executing the WAI or STOP instruction or if a CPU access is delayed considerably by EWAIT.

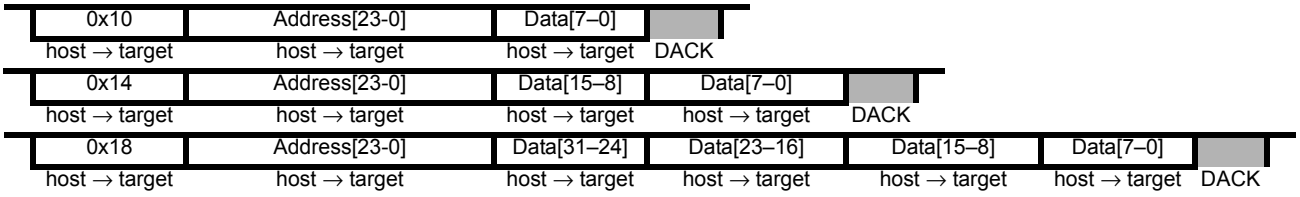
This command can be used to dynamically access the PC for performance monitoring as the execution of this command is considerably less intrusive to the real-time operation of an application than a BACKGROUND/read-PC/GO command sequence.

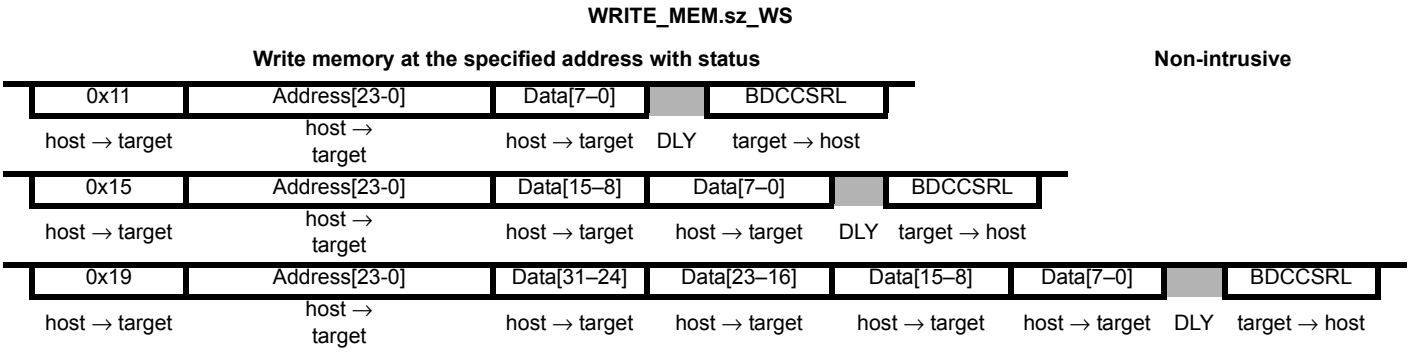
While the BDC is not in active BDM, SYNC_PC returns the PC address of the instruction currently being executed by the CPU. In active BDM, SYNC_PC returns the address of the next instruction to be executed on returning from active BDM. Thus following a write to the PC in active BDM, a SYNC_PC returns that written value.

6.11.4.4.16 WRITE_MEM.sz, WRITE_MEM.sz_WS

WRITE_MEM.sz

Write memory at the specified address **Non-intrusive**



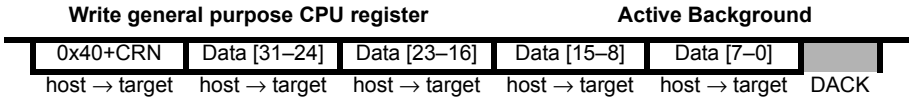


Write data to the specified memory address. The address is transmitted as three 8-bit packets (msb to lsb) immediately after the command.

If the with-status option is specified, the status byte contained in BDCCSRL is returned after the write data. This status byte reflects the state after the memory write was performed. The examples show the WRITE_MEM.B{_WS}, WRITE_MEM.W{_WS}, and WRITE_MEM.L{_WS} commands. If enabled an ACK pulse is generated after the internal write access has been completed or aborted.

The hardware forces low-order address bits to zero longword accesses to ensure these accesses are on 0-modulo-size alignments. Byte alignment details are described in BDC access of device memory mapped resources.

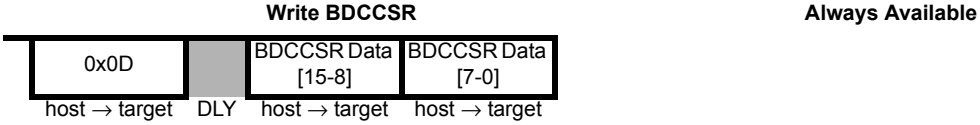
6.11.4.4.17 WRITE_Rn



If the device is in active BDM, this command writes the 32-bit operand to the selected CPU general purpose register. See BDC access of CPU registers for the CRN details. Accesses to CPU registers are always 32-bits wide, regardless of implemented register width. If enabled an ACK pulse is generated after the internal write access has been completed or aborted.

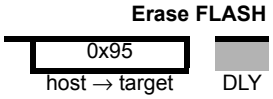
If the device is not in active BDM, this command is rejected as an illegal operation, the ILLCMD bit is set and no operation is performed.

6.11.4.4.18 WRITE_BDCCSR



16-bit write to the BDCCSR register. No ACK pulse is generated. Writing to this register can be used to configure control bits or clear flag bits. Refer to the register bit descriptions.

6.11.4.4.19 ERASE_FLASH



Mass erase the internal flash. This command can always be issued. On receiving this command twice in succession, the BDC sets the ERASE bit in BDCCSR and requests a flash mass erase. Any other BDC command following a single ERASE_FLASH initializes the sequence, such that the ERASE_FLASH must then be applied twice in succession to request a mass erase. If 512 BDCSI clock cycles elapse between the consecutive ERASE_FLASH commands then a timeout occurs, which forces a soft reset and initializes the sequence. The ERASE bit is cleared when the mass erase sequence has been completed. No ACK is driven.

During the mass erase operation, which takes many clock cycles, the command status is indicated by the ERASE bit in BDCCSR. While a mass erase operation is ongoing, Always-available commands can be issued. This allows the status of the erase operation to be polled by reading BDCCSR to determine when the operation is finished. The status of the flash array can be verified by subsequently reading the flash error flags to determine if the erase completed successfully.

ERASE_FLASH can be aborted by a SYNC pulse forcing a soft reset.

Note: Device bus frequency considerations

The ERASE_FLASH command requires the default device bus clock frequency after reset. Thus the bus clock frequency must not be changed following reset before issuing an ERASE_FLASH command.

6.11.4.4.20 Step1

This command is used to step through application code. In active BDM, this command executes the next CPU instruction in application code. If enabled an ACK is driven.

If a STEP1 command is issued and the CPU is not halted, the command is ignored.

6.11.4.5 BDC access of internal resources

Unsuccessful read accesses of internal resources return a value of 0xEE for each data byte. This enables a debugger to recognize a potential error, even if neither the ACK handshaking protocol nor a status command is currently being executed. The value of 0xEE is returned in the following cases.

- Illegal address access, whereby ILLACC is set
- Invalid READ_SAME or DUMP_MEM sequence
- Invalid READ_Rn command (BDM inactive or CRN incorrect)
- Internal resource read with timeout, whereby NORESP is set

6.11.4.5.1 BDC access of CPU registers

The CRN field of the READ_Rn and WRITE_Rn commands contains a pointer to the CPU registers. The mapping of CRN to CPU registers is shown in Table 436. Accesses to CPU registers are always 32-bits wide, regardless of implemented register width. This means that the BDC data transmission for these commands is 32-bits long. The valid bits of the transfer are listed in the Valid Data Bits column. The other bits of the transmission are redundant.

Attempted accesses of CPU registers using a CRN of 0xD,0xE or 0xF is invalid, returning the value 0xEE for each byte and setting the ILLACC bit.

Table 436. CPU register number (CRN) mapping

CPU Register	Valid Data Bits	Command	Opcode	Command	Opcode
D0	[7:0]	WRITE_D0	0x40	READ_D0	0x60
D1	[7:0]	WRITE_D1	0x41	READ_D1	0x61
D2	[15:0]	WRITE_D2	0x42	READ_D2	0x62
D3	[15:0]	WRITE_D3	0x43	READ_D3	0x63
D4	[15:0]	WRITE_D4	0x44	READ_D4	0x64
D5	[15:0]	WRITE_D5	0x45	READ_D5	0x65
D6	[31:0]	WRITE_D6	0x46	READ_D6	0x66
D7	[31:0]	WRITE_D7	0x47	READ_D7	0x67
X	[23:0]	WRITE_X	0x48	READ_X	0x68
Y	[23:0]	WRITE_Y	0x49	READ_Y	0x69
SP	[23:0]	WRITE_SP	0x4A	READ_SP	0x6A
PC	[23:0]	WRITE_PC	0x4B	READ_PC	0x6B
CCR	[15:0]	WRITE_CCR	0x4C	READ_CCR	0x6C

6.11.4.5.2 BDC access of device memory mapped resources

The device memory map is accessed using READ_MEM, DUMP_MEM, WRITE_MEM, FILL_MEM and READ_SAME, which support different access sizes, as explained in the command descriptions.

When an unimplemented command occurs during a DUMP_MEM, FILL_MEM or READ_SAME sequence, then that sequence is ended.

Illegal read accesses return a value of 0xEE for each byte. After an illegal access FILL_MEM and READ_SAME commands are not valid, and it is necessary to restart the internal access sequence with READ_MEM or WRITE_MEM. An illegal access does not break a DUMP_MEM sequence. After read accesses that cause the RDINV bit to be set, DUMP_MEM and READ_SAME commands are valid, it is not necessary to restart the access sequence with a READ_MEM.

The hardware forces low-order address bits to zero for longword accesses to ensure these accesses are realigned to 0-modulo-size alignments.

Word accesses map to 2-bytes from within a 4-byte field as shown in Table . Thus if address bits [1:0] are both logic “1” the access is realigned so that it does not straddle the 4-byte boundary but accesses data from within the addressed 4-byte field.

Table 437. Field location to byte access mapping

Address[1:0]	Access Size	00	01	10	11	Note
00	32-bit	Data[31:24]	Data[23:16]	Data [15:8]	Data [7:0]	
01	32-bit	Data[31:24]	Data[23:16]	Data [15:8]	Data [7:0]	Realigned
10	32-bit	Data[31:24]	Data[23:16]	Data [15:8]	Data [7:0]	Realigned
11	32-bit	Data[31:24]	Data[23:16]	Data [15:8]	Data [7:0]	Realigned
00	16-bit	Data [15:8]	Data [7:0]			
01	16-bit		Data [15:8]	Data [7:0]		
10	16-bit			Data [15:8]	Data [7:0]	
11	16-bit			Data [15:8]	Data [7:0]	Realigned
00	8-bit	Data [7:0]				
01	8-bit		Data [7:0]			
10	8-bit			Data [7:0]		
11	8-bit				Data [7:0]	
			Denotes byte that is not transmitted			

6.11.4.5.2.1 FILL_MEM and DUMP_MEM increments and alignment

FILL_MEM and DUMP_MEM increment the previously accessed address by the previous access size to calculate the address of the current access. On misaligned longword accesses, the address bits [1:0] are forced to zero, therefore the following FILL_MEM or DUMP_MEM increment to the first address in the next 4-byte field. This is shown in Table 438, the address of the first DUMP_MEM.32 following READ_MEM.32 being calculated from 0x004000+4.

When misaligned word accesses are realigned, then the original address (not the realigned address) is incremented for the following FILL_MEM, DUMP_MEM command.

Misaligned word accesses can cause the same locations to be read twice as shown in rows 6 and 7. The hardware ensures alignment at an attempted misaligned word access across a 4-byte boundary, as shown in row 7. The following word access in row 8 continues from the realigned address of row 7.

Table 438. Field location to byte access mapping

Row	Command	Address	Address[1:0]	00	01	10	11
1	READ_MEM.32	0x004003	11	Accessed	Accessed	Accessed	Accessed
2	DUMP_MEM.32	0x004004	00	Accessed	Accessed	Accessed	Accessed
3	DUMP_MEM.16	0x004008	00	Accessed	Accessed		
4	DUMP_MEM.16	0x00400A	10			Accessed	Accessed
5	DUMP_MEM.08	0x00400C	00	Accessed			
6	DUMP_MEM.16	0x00400D	01		Accessed	Accessed	

Table 438. Field location to byte access mapping (continued)

Row	Command	Address	Address[1:0]	00	01	10	11
7	DUMP_MEM.16	0x00400E	10			Accessed	Accessed
8	DUMP_MEM.16	0x004010	01	Accessed	Accessed		

6.11.4.5.2.2 READ_SAME effects of variable access size

R EAD_SAME uses the unadjusted address given in the previous READ_MEM command as a base address for subsequent READ_SAME commands. When the READ_MEM and READ_SAME size parameters differ then READ_SAME uses the original base address buts aligns 32-bit and 16-bit accesses, where those accesses would otherwise cross the aligned 4-byte boundary. Table 439 shows some examples of this.

Table 439. Consecutive READ_SAME accesses with variable size

Row	Command	Base Address	00	01	10	11
1	READ_MEM.32	0x004003	Accessed	Accessed	Accessed	Accessed
2	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
3	READ_SAME.16	—			Accessed	Accessed
4	READ_SAME.08	—				Accessed
5	READ_MEM.08	0x004000	Accessed			
6	READ_SAME.08	—	Accessed			
7	READ_SAME.16	—	Accessed	Accessed		
8	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
9	READ_MEM.08	0x004002			Accessed	
10	READ_SAME.08	—			Accessed	
11	READ_SAME.16	—			Accessed	Accessed
12	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
13	READ_MEM.08	0x004003				Accessed
14	READ_SAME.08	—				Accessed
15	READ_SAME.16	—			Accessed	Accessed
16	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
17	READ_MEM.16	0x004001		Accessed	Accessed	
18	READ_SAME.08	—		Accessed		
19	READ_SAME.16	—		Accessed	Accessed	
20	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed
21	READ_MEM.16	0x004003			Accessed	Accessed
22	READ_SAME.08	—				Accessed
23	READ_SAME.16	—			Accessed	Accessed
24	READ_SAME.32	—	Accessed	Accessed	Accessed	Accessed

6.11.4.6 BDC serial interface

The BDC communicates with external devices serially via the BKGD pin. During reset, this pin is a mode select input which selects between normal and special modes of operation. After reset, this pin becomes the dedicated serial interface pin for the BDC.

The BDC serial interface uses an internal clock source, selected by the CLKSW bit in the BDCCSR register. This clock is referred to as the target clock in the following explanation.

The BDC serial interface uses a clocking scheme in which the external host generates a falling edge on the BKGD pin to indicate the start of each bit time. This falling edge is sent for every bit whether data is transmitted or received. Data is transferred most significant bit (MSB) first at 16 target clock cycles per bit. The interface times out if during a command 512 clock cycles occur between falling edges from the host. The timeout forces the current command to be discarded.

The BKGD pin is a pseudo open-drain pin and has a weak on-chip active pull-up that is enabled at all times. It is assumed that there is an external pull-up and that drivers connected to BKGD do not typically drive the high level. Since R-C rise time could be unacceptably long, the target system and host provide brief drive-high (speed-up) pulses to drive BKGD to a logic 1. The source of this speedup pulse is the host for transmit cases and the target for receive cases.

The timing for host-to-target is shown in Figure 73 and that of target-to-host in Figure 74 and Figure 75. All cases begin when the host drives the BKGD pin low to generate a falling edge. Since the host and target operate from separate clocks, it can take the target up to one full clock cycle to recognize this edge. This synchronization uncertainty is illustrated in Figure 73. The target measures delays from this perceived start of the bit time while the host measures delays from the point it actually drove BKGD low to start the bit up to one target clock cycle earlier. Synchronization between the host and target is established in this manner at the start of every bit time.

Figure 73 shows an external host transmitting a logic 1 and transmitting a logic 0 to the BKGD pin of a target system. The host is asynchronous to the target, so there is up to a one clock-cycle delay from the host-generated falling edge to where the target recognizes this edge as the beginning of the bit time. Ten target clock cycles later, the target senses the bit level on the BKGD pin. Internal glitch detect logic requires the pin be driven high no later than eight target clock cycles after the falling edge for a logic 1 transmission.

Since the host drives the high speedup pulses in these two cases, the rising edges look like digitally driven signals.

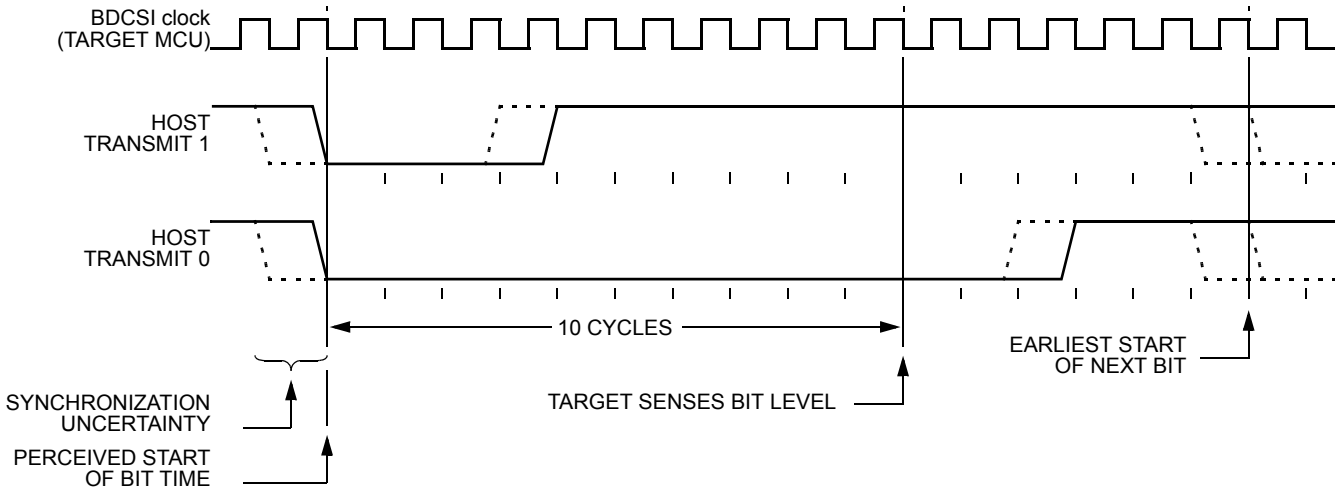


Figure 73. BDC host-to-target serial bit timing

Figure 74 shows the host receiving a logic 1 from the target system. The host holds the BKGD pin low long enough for the target to recognize it (at least two target clock cycles). The host must release the low drive at the latest after 6 clock cycles, before the target drives a brief high speedup pulse seven target clock cycles after the perceived start of the bit time. The host should sample the bit level about 10 target clock cycles after it started the bit time.

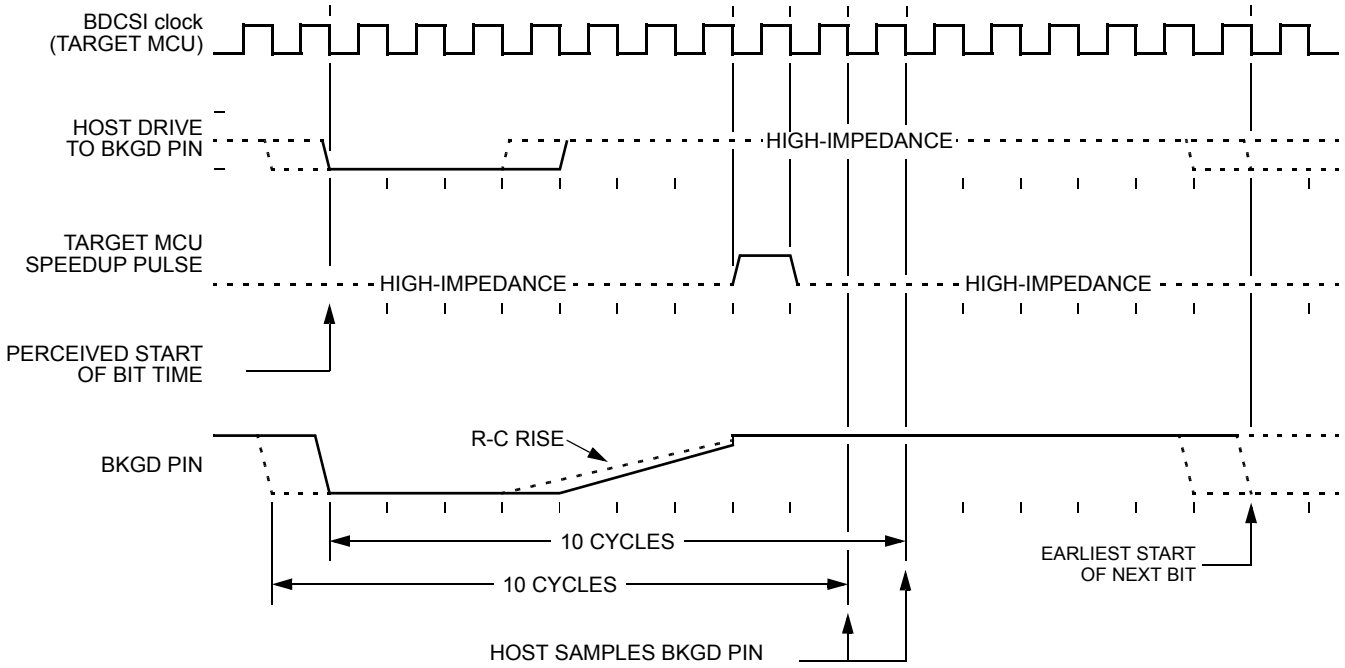


Figure 74. BDC target-to-host serial bit timing (logic 1)

Figure 75 shows the host receiving a logic 0 from the target. The host initiates the bit time but the target finishes it. Since the target wants the host to receive a logic 0, it drives the BKGD pin low for 13 target clock cycles then briefly drives it high to speed up the rising edge. The host samples the bit level about 10 target clock cycles after starting the bit time.

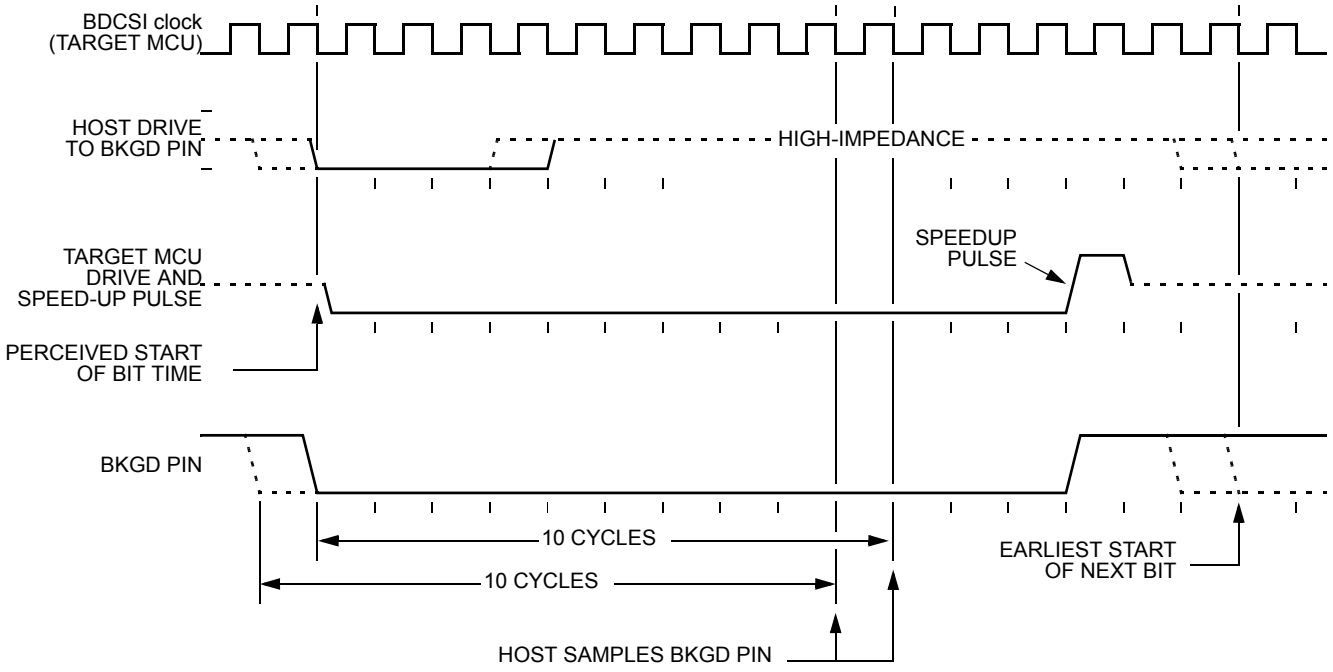


Figure 75. BDC target-to-host serial bit timing (logic 0)

6.11.4.7 Serial interface hardware handshake (ACK pulse) protocol

BDC commands are processed internally at the device core clock rate. Since the BDCSI clock can be asynchronous relative to the bus frequency, a handshake protocol is provided so the host can determine when an issued command has been executed. This section describes the hardware handshake protocol.

The hardware handshake protocol signals to the host controller when a BDC command has been executed by the target. This protocol is implemented by a low pulse (16 BDCSI clock cycles) followed by a brief speedup pulse on the BKGD pin, generated by the target MCU when a command, issued by the host, has been successfully executed (see Figure 76). This pulse is referred to as the ACK pulse. After the ACK pulse has finished, the host can start the bit retrieval if the last issued command was a read command, or start a new command if the last command was a write command or a control command.

If a command results in an error condition, whereby a BDCCSRL flag is set, then the target generates a “Long-ACK” low pulse of 64 BDCSI clock cycles, followed by a brief speed pulse. This indicates to the host that an error has occurred. The host can subsequently read BDCCSR to determine the type of error. Whether normal ACK or Long-ACK, the ACK pulse is not issued earlier than 32 BDCSI clock cycles after the BDC command was issued. The end of the BDC command is assumed to be the 16th BDCSI clock cycle of the last bit. The 32 cycle minimum delay differs from the 16 cycle delay time with ACK disabled.

If the BDC does not gain access within 512 core clock cycles, the request is aborted, the NORESP flag is set and a Long-ACK pulse is transmitted to indicate an error case.

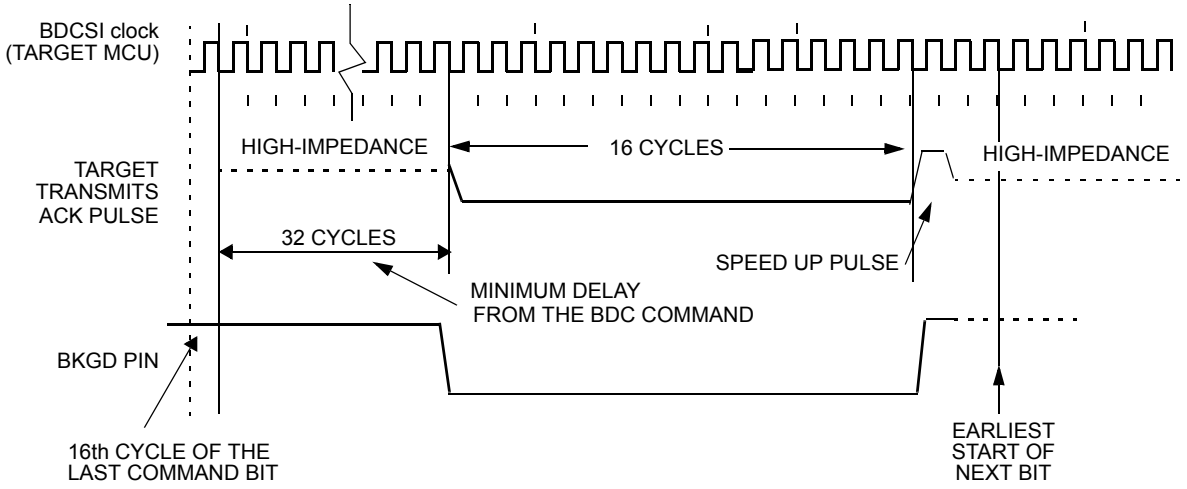


Figure 76. Target acknowledge pulse (ACK)

The handshake protocol is enabled by the ACK_ENABLE command. The BDC sends an ACK pulse when the ACK_ENABLE command has been completed. This feature can be used by the host to evaluate if the target supports the hardware handshake protocol. If an ACK pulse is issued in response to this command, the host knows that the target supports the hardware handshake protocol.

Unlike the normal bit transfer, where the host initiates the transmission by issuing a negative edge on the BKGD pin, the serial interface ACK handshake pulse is initiated by the target MCU by issuing a negative edge on the BKGD pin. Figure 76 specifies the timing when the BKGD pin is being driven. The host must follow this timing constraint in order to avoid the risk of an electrical conflict at the BKGD pin.

When the handshake protocol is enabled, the STEAL bit in BDCCSR selects if bus cycle stealing is used to gain immediate access. If STEAL is cleared, the BDC is configured for low priority bus access using free cycles, without stealing cycles. This guarantees that BDC accesses remain truly non-intrusive to not affect the system timing during debugging. If STEAL is set, the BDC gains immediate access, if necessary stealing an internal bus cycle.

Note

If bus steals are disabled then a loop with no free cycles cannot allow access. In this case the host must recognize repeated NORESP messages and then issue a BACKGROUND command to stop the target and access the data.

Following a STOP instruction, if the BDC is enabled, the first ACK, following stop mode entry is a long ACK to indicate an exception.

Figure 77 shows the ACK handshake protocol without steal in a command level timing diagram. The READ_MEM.B command is used as an example. First, the 8-bit command code is sent by the host, followed by the address of the memory location to be read. The target BDC decodes the command. Then an internal access is requested by the BDC. When a free bus cycle occurs the READ_MEM.B operation is carried out. If no free cycle occurs within 512 core clock cycles then the access is aborted, the NORESP flag is set and the target generates a Long-ACK pulse.

Having retrieved the data, the BDC issues an ACK pulse to the host controller, indicating that the addressed byte is ready to be retrieved. After detecting the ACK pulse, the host initiates the data read part of the command.

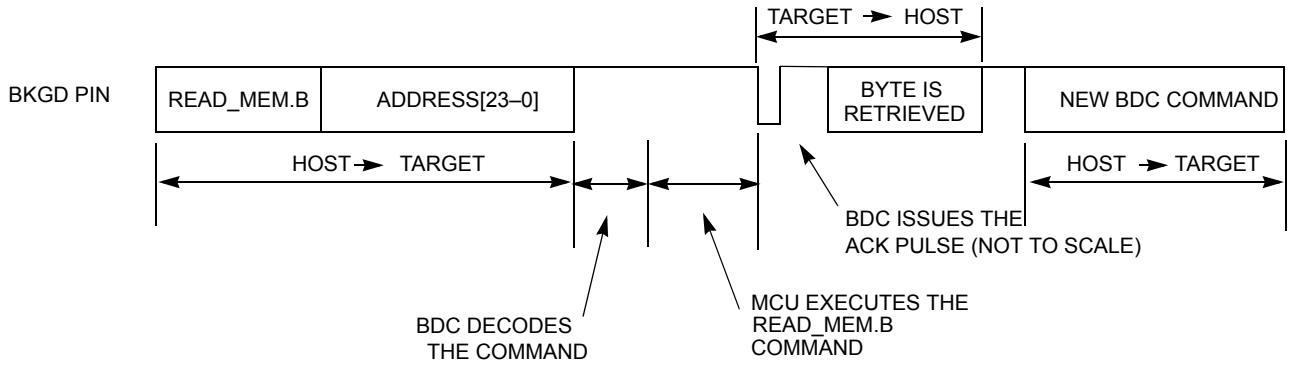


Figure 77. Handshake protocol at command level

Alternatively, setting the STEAL bit configures the handshake protocol to make an immediate internal access, independent of free bus cycles.

The ACK handshake protocol does not support nested ACK pulses. If a BDC command is not acknowledged by an ACK pulse, the host needs to abort the pending command first in order to be able to issue a new BDC command. The host can decide to abort any possible pending ACK pulse in order to be sure a new command can be issued. Therefore, the protocol provides a mechanism in which a command, and its corresponding ACK, can be aborted.

Commands With-Status do not generate an ACK, thus if ACK is enabled and a With-Status command is issued, the host must use the 512 cycle timeout to calculate when the data is ready for retrieval.

6.11.4.8 Hardware handshake abort procedure

The abort procedure is based on the SYNC command. To abort a command that has not responded with an ACK pulse, the host controller generates a sync request (by driving BKGD low for at least 128 BDCSI clock cycles and then driving it high for one BDCSI clock cycle as a speedup pulse). By detecting this long low pulse in the BKGD pin, the target executes the SYNC protocol, see SYNC, and assumes that the pending command and therefore the related ACK pulse are being aborted. After the SYNC protocol has been completed the host is free to issue new BDC commands.

The host can issue a SYNC close to the 128 clock cycles length, providing a small overhead on the pulse length to assure the sync pulse is not misinterpreted by the target. See SYNC".

Figure 78 shows a SYNC command being issued after a READ_MEM, which aborts the READ_MEM command. Note that, after the command is aborted a new command is issued by the host.

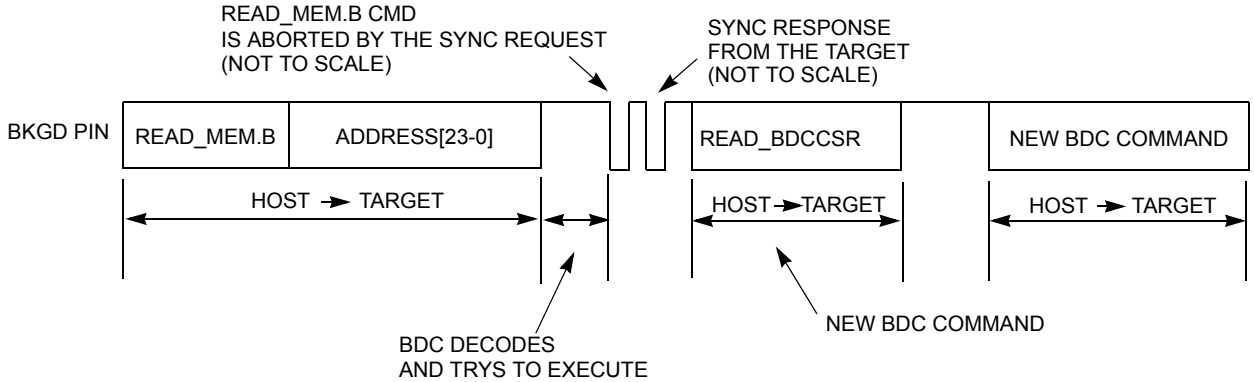


Figure 78. ACK abort procedure at the command level (not to scale)

Figure 79 shows a conflict between the ACK pulse and the SYNC request pulse. The target is executing a pending BDC command at the exact moment the host is being connected to the BKGD pin. In this case, an ACK pulse is issued simultaneously to the SYNC command. Thus there is an electrical conflict between the ACK speed-up pulse and the SYNC pulse. As this is not a probable situation, the protocol does not prevent this conflict from happening.

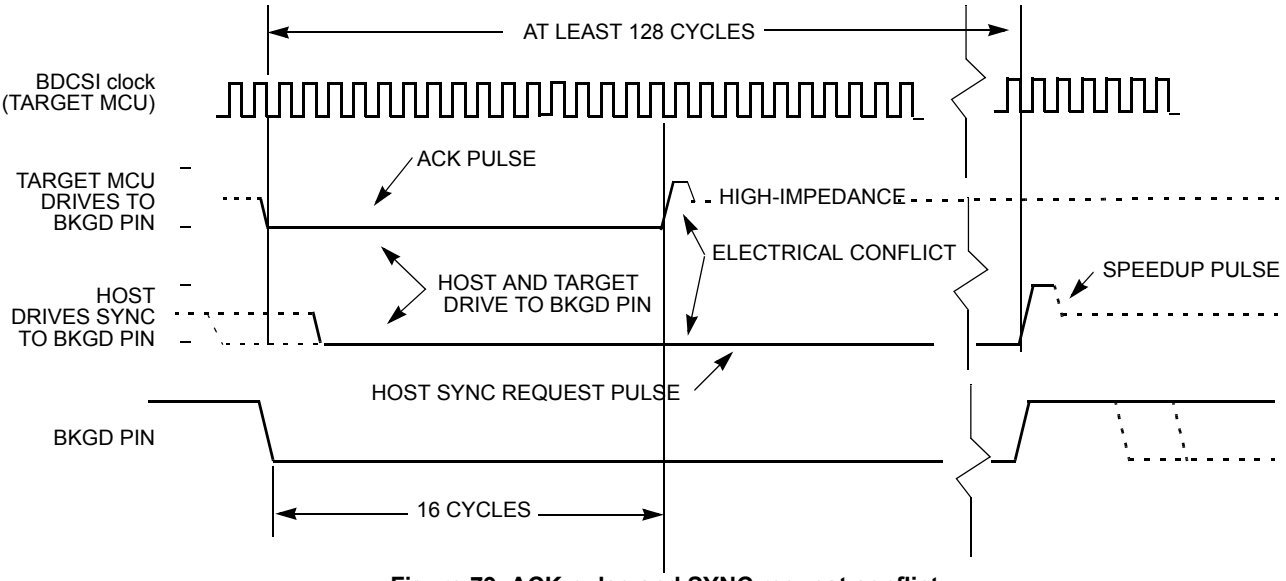


Figure 79. ACK pulse and SYNC request conflict

6.11.4.9 Hardware handshake disabled (ACK pulse disabled)

The default state of the BDC after reset is hardware handshake protocol disabled. It can also be disabled by the ACK_DISABLE BDC command. This provides backwards compatibility with the existing host devices which are not able to execute the hardware handshake protocol. For host devices that support the hardware handshake protocol, true non-intrusive debugging and error flagging is offered.

If the ACK pulse protocol is disabled, the host needs to use the worst case delay time at the appropriate places in the protocol.

If the handshake protocol is disabled, the access is always independent of free cycles, whereby BDC has higher priority than CPU. Since at least 2 bytes (command byte + data byte) are transferred over BKGD the maximum intrusiveness is only once every few hundred cycles.

After decoding an internal access command, the BDC then awaits the next internal core clock cycle. The relationship between BDCSI clock and core clock must be considered. If the host retrieves the data immediately, then the BDCSI clock frequency must not be more than 4 times the core clock frequency, in order to guarantee that the BDC gains bus access within 16 the BDCSI cycle DLY period following an access command. If the BDCSI clock frequency is more than 4 times the core clock frequency, then the host must use a suitable delay time before retrieving data (see Clock frequency considerations). Furthermore, for stretched read accesses to external resources via a device expanded bus (if implemented) the potential extra stretch cycles must be taken into consideration before attempting to obtain read data.

If the access does not succeed before the host starts data retrieval then the NORESP flag is set but the access is not aborted. The NORESP state can be used by the host to recognize an unexpected access conflict due to stretched expanded bus accesses. Although the NORESP bit is set when an access does not succeed before the start of data retrieval, the access may succeed in following bus cycles if the internal access has already been initiated.

6.11.4.10 Single stepping

When a STEP1 command is issued to the BDC in active BDM, the CPU executes a single instruction in the user code and returns to active BDM. The STEP1 command can be issued repeatedly to step through the user code one instruction at a time.

If an interrupt is pending when a STEP1 command is issued, the interrupt stacking operation occurs but no user instruction is executed. In this case the stacking counts as one instruction. The device re-enters active BDM with the program counter pointing to the first instruction in the interrupt service routine.

When stepping through the user code, the execution of the user code is done step by step but peripherals are free running. Some peripheral modules include a freeze feature, whereby their clocks are halted when the device enters active BDM. Timer modules typically include the freeze feature. Serial interface modules typically do not include the freeze feature. Hence possible timing relations between CPU code execution and occurrence of events of peripherals no longer exist.

If the handshake protocol is enabled and BDCCIS is set then stepping over the STOP instruction causes the Long-ACK pulse to be generated and the BDCCSR STOP flag to be set. When stop mode is exited due to an interrupt the device enters active BDM and the PC points to the start of the corresponding interrupt service routine. Stepping can be continued.

Stepping over a WAI instruction, the STEP1 command cannot be finished because active BDM cannot be entered after CPU starts to execute the WAI instruction.

Stepping over the WAI instruction causes the BDCCSR WAIT and NORESP flags to be set and, if the handshake protocol is enabled, then the Long-ACK pulse is generated. Then the device enters wait mode, clears the BDMACT bit and awaits an interrupt to leave wait mode. In this time non-intrusive BDC commands are possible, although the STEP1 has actually not finished. When an interrupt occurs the device leaves wait mode, enters active BDM and the PC points to the start of the corresponding interrupt service routine. A further ACK related to stepping over the WAI is not generated.

6.11.4.11 Serial communication timeout

The host initiates a host-to-target serial transmission by generating a falling edge on the BKGD pin. If BKGD is kept low for more than 128 target clock cycles, the target understands that a SYNC command was issued. In this case, the target waits for a rising edge on BKGD in order to answer the SYNC request pulse. When the BDC detects the rising edge a soft reset is generated, whereby the current BDC command is discarded. If the rising edge is not detected, the target keeps waiting forever without any timeout limit.

If a falling edge is not detected by the target within 512 clock cycles since the last falling edge, a timeout occurs and the current command is discarded without affecting memory or the operating mode of the MCU. This is referred to as a soft-reset. This timeout also applies if 512 cycles elapse between 2 consecutive ERASE_FLASH commands. The soft reset is disabled while the internal flash mass erase operation is pending completion.

Timeouts are also possible if a BDC command is partially issued, or data partially retrieved. Thus if a time greater than 512 BDCSI clock cycles is observed between two consecutive negative edges, a soft-reset occurs causing the partially received command or data retrieved to be discarded. The next negative edge at the BKGD pin, after a soft-reset has occurred, is considered by the target as the start of a new BDC command, or the start of a SYNC request pulse.

6.11.4.12 Application information

6.11.4.12.1 Clock frequency considerations

Read commands without status and without ACK must consider the frequency relationship between BDCSI and the internal core clock. If the core clock is slow, then the internal access may not have been carried out within the standard 16 BDCSI cycle delay period (DLY). The host must then extend the DLY period or clock frequencies accordingly. Taking internal clock domain synchronizers into account, the minimum number of BDCSI periods required for the DLY is expressed by:

$\#DLY > 3(f_{(BDCSI\ clock)} / f_{(core\ clock)}) + 4$

and the minimum core clock frequency with respect to BDCSI clock frequency is expressed by

Minimum $f_{(core\ clock)} = (3/(\#DLY\ cycles\ -4))f_{(BDCSI\ clock)}$

For the standard 16 period DLY this yields $f_{(core\ clock)} >= (1/4)f_{(BDCSI\ clock)}$

6.12 Serial peripheral interface (S12SPIV5)

6.12.1 Introduction

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or the SPI operation can be interrupt driven.

6.12.1.1 Glossary of terms

SPI	Serial Peripheral Interface
SS	Slave Select
SCK	Serial Clock
MOSI	Master Output, Slave Input
MISO	Master Input, Slave Output
MOMI	Master Output, Master Input
SISO	Slave Input, Slave Output

6.12.1.2 Features

The SPI includes these distinctive features:

- Master mode and slave mode
- Selectable 8 or 16-bit transfer width
- Bidirectional mode
- Slave select output
- Mode fault error flag with CPU interrupt capability
- Double-buffered data register
- Serial clock with programmable polarity and phase
- Control of SPI operation during wait mode

6.12.1.3 Modes of operation

The SPI functions in three modes: run, wait, and stop.

- Run mode

  This is the basic mode of operation.

- Wait mode

  SPI operation in wait mode is a configurable low power mode, controlled by the SPISWAI bit located in the SPICR2 register. In Wait mode, if the SPISWAI bit is clear, the SPI operates like in Run mode. If the SPISWAI bit is set, the SPI goes into a power conservative state, with the SPI clock generation turned off. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into Run mode. If the SPI is configured as a slave, reception and transmission of data continues, so that the slave stays synchronized to the master.

- Stop mode

  The SPI is inactive in stop mode for reduced power consumption. If the SPI is configured as a master, any transmission in progress stops, but is resumed after CPU goes into Run mode. If the SPI is configured as a slave, reception and transmission of data continues, so that the slave stays synchronized to the master.

For a detailed description of operating modes, refer to Low power mode options.

6.12.1.4 Block diagram

Figure 80 gives an overview on the SPI architecture. The main parts of the SPI are status, control and data registers, shifter logic, baud rate generator, master/slave control logic, and port control logic.

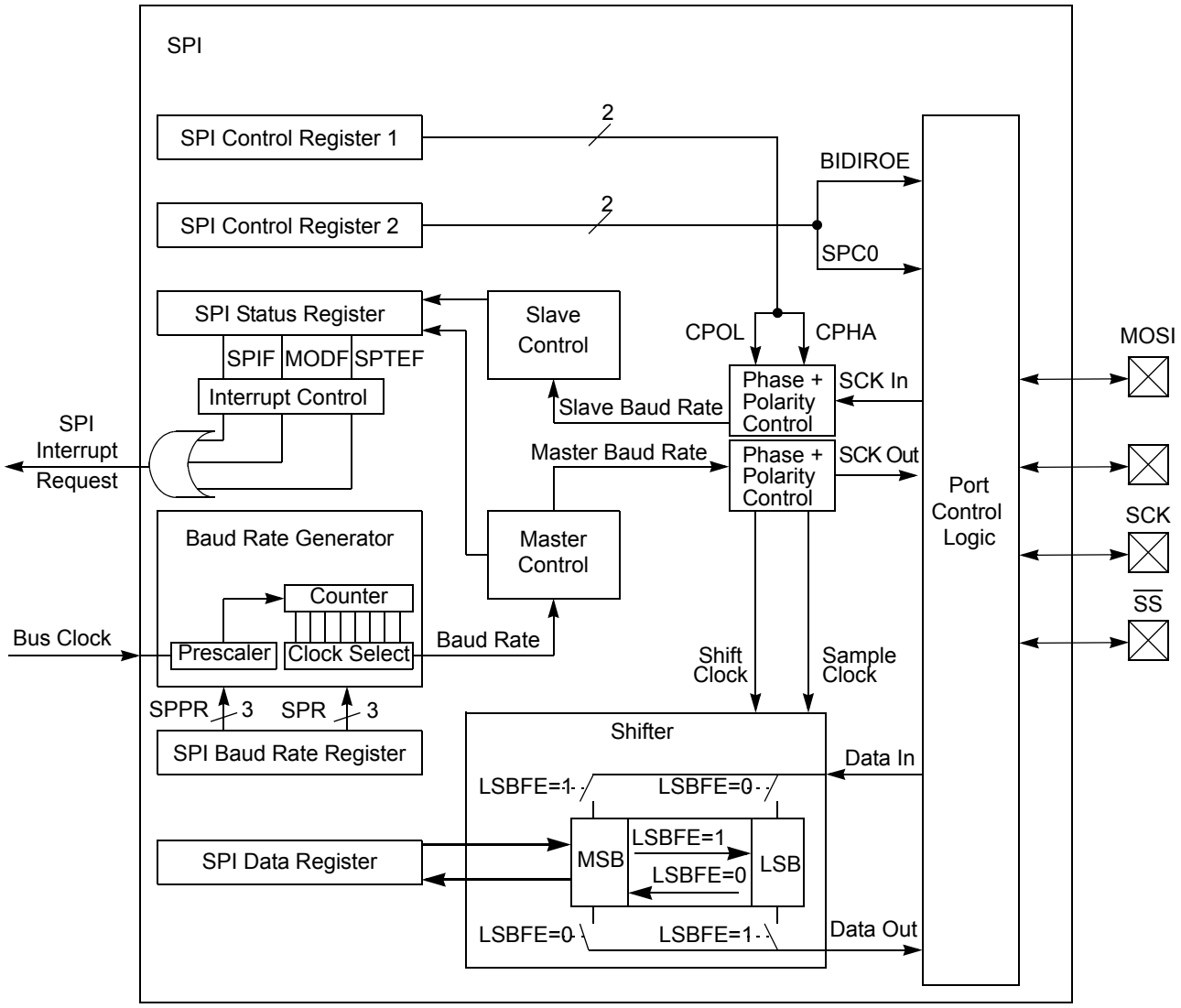


Figure 80. SPI block diagram

6.12.2 External signal description

This section lists the name and description of all ports including inputs and outputs that do, or may, connect off chip. The SPI module has a total of four external pins.

6.12.2.1 MOSI — master out/slave in pin

This pin is used to transmit data out of the SPI module when it is configured as a master and receive data when it is configured as slave.

6.12.2.2 MISO — master in/slave out pin

This pin is used to transmit data out of the SPI module when it is configured as a slave and receive data when it is configured as master.

6.12.2.3 $\overline{SS}$ — slave select pin

This pin is used to output the select signal from the SPI module to another peripheral with which a data transfer is to take place when it is configured as a master and it is used as an input to receive the slave select signal when the SPI is configured as slave.

6.12.2.4 SCK — serial clock pin

In Master mode, this is the synchronous output clock. In Slave mode, this is the synchronous input clock.

6.12.3 Memory map and register definition

This section provides a detailed description of address space and registers used by the SPI.

6.12.3.1 Module memory map

The memory map for the SPI is given in Table 440. The address listed for each register is the sum of a base address and an address offset. The base address is defined at the SoC level and the address offset is defined at the module level. Reads from the reserved bits return zeros and writes to the reserved bits have no effect.

Table 440. SPI register summary

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 SPICR1	R	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
	W								
0x0001 SPICR2	R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
	W								
0x0002 SPIBR	R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
	W								
0x0003 SPISR	R	SPIF	0	SPTEF	MODF	0	0	0	0
	W								
0x0004 SPIDRH	R	R15	R14	R13	R12	R11	R10	R9	R8
	W	T15	T14	T13	T12	T11	T10	T9	T8
0x0005 SPIDRL	R	R7	R6	R5	R4	R3	R2	R1	R0
	W	T7	T6	T5	T4	T3	T2	T1	T0
0x0006 Reserved	R								
	W								
0x0007 Reserved	R								
	W								
		= Unimplemented or Reserved							

6.12.3.2 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

6.12.3.2.1 SPI control register 1 (SPICR1)

)

Table 441. SPI control register 1 (SPICR1)

Module Base +0x0000

	7	6	5	4	3	2	1	0
R W	SPIE	SPE	SPTIE	MSTR	CPOL	CPHA	SSOE	LSBFE
Reset	0	0	0	0	0	1	0	0

Notes:
281.Read: Anytime
Write: Anytime

Table 442. SPICR1 field descriptions

Field	Description
7 SPIE	**SPI Interrupt Enable Bit** — This bit enables SPI interrupt requests, if SPIF or MODF status flag is set. 0 SPI interrupts disabled. 1 SPI interrupts enabled.
6 SPE	**SPI System Enable Bit** — This bit enables the SPI system and dedicates the SPI port pins to SPI system functions. If SPE is cleared, SPI is disabled and forced into idle state, status bits in SPISR register are reset. 0 SPI disabled (lower power consumption). 1 SPI enabled, port pins are dedicated to SPI functions.
5 SPTIE	**SPI Transmit Interrupt Enable** — This bit enables SPI interrupt requests, if SPTEF flag is set. 0 SPTEF interrupt disabled. 1 SPTEF interrupt enabled.
4 MSTR	**SPI Master/Slave Mode Select Bit** — This bit selects whether the SPI operates in master or slave mode. Switching the SPI from master to slave or vice versa forces the SPI system into idle state. 0 SPI is in slave mode. 1 SPI is in master mode.
3 CPOL	**SPI Clock Polarity Bit** — This bit selects an inverted or non-inverted SPI clock. To transmit data between SPI modules, the SPI modules must have identical CPOL values. In Master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Active-high clocks selected. In idle state SCK is low. 1 Active-low clocks selected. In idle state SCK is high.
2 CPHA	**SPI Clock Phase Bit** — This bit is used to select the SPI clock format. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Sampling of data occurs at odd edges (1,3,5,….) of the SCK clock. 1 Sampling of data occurs at even edges (2,4,6,….) of the SCK clock.
1 SSOE	**Slave Select Output Enable** — The $\overline{SS}$ output feature is enabled only in master mode, if MODFEN is set, by asserting the SSOE as shown in Table 443. In Master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.
0 LSBFE	**LSB-First Enable** — This bit does not affect the position of the MSB and LSB in the data register. Reads and writes of the data register always have the MSB in the highest bit position. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 Data is transferred most significant bit first. 1 Data is transferred least significant bit first.

Table 443. $\overline{SS}$ input / output selection

MODFEN	SSOE	Master Mode	Slave Mode
0	0	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
0	1	$\overline{SS}$ not used by SPI	$\overline{SS}$ input
1	0	$\overline{SS}$ input with MODF feature	$\overline{SS}$ input
1	1	$\overline{SS}$ is slave select output	$\overline{SS}$ input

6.12.3.2.2 SPI control register 2 (SPICR2)

)

Table 444. SPI control register 2 (SPICR2)

Module Base +0x0001

	7	6	5	4	3	2	1	0
R	0	XFRW	0	MODFEN	BIDIROE	0	SPISWAI	SPC0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
282.Read: Anytime
Write: Anytime. Writes to the reserved bits have no effect

Table 445. SPICR2 field descriptions

Field	Description
6 XFRW	**Transfer Width —** This bit is used for selecting the data transfer width. If 8-bit transfer width is selected, SPIDRL becomes the dedicated data register and SPIDRH is unused. If 16-bit transfer width is selected, SPIDRH and SPIDRL form a 16-bit data register. Refer to SPI status register (SPISR) for information about transmit/receive data handling and the interrupt flag clearing mechanism. In Master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 8-bit Transfer Width (n = 8) (283) 1 16-bit Transfer Width (n = 16)(283)
4 MODFEN	**Mode Fault Enable Bit** — This bit allows the MODF failure to be detected. If the SPI is in master mode and MODFEN is cleared, then the $\overline{SS}$ port pin is not used by the SPI. In Slave mode, the $\overline{SS}$ is available only as an input regardless of the value of MODFEN. For an overview on the impact of the MODFEN bit on the $\overline{SS}$ port pin configuration, refer to SS input / output selection. In Master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state. 0 $\overline{SS}$ port pin is not used by the SPI. 1 $\overline{SS}$ port pin with MODF feature.
3 BIDIROE	**Output Enable in the Bidirectional Mode of Operation** — This bit controls the MOSI and MISO output buffer of the SPI, when in bidirectional mode of operation (SPC0 is set). In Master mode, this bit controls the output buffer of the MOSI port, in slave mode it controls the output buffer of the MISO port. In Master mode, with SPC0 set, a change of this bit will abort a transmission in progress and force the SPI into idle state. 0 Output buffer disabled. 1 Output buffer enabled.
1 SPISWAI	**SPI Stop in Wait Mode Bit** — This bit is used for power conservation while in wait mode. 0 SPI clock operates normally in Wait mode. 1 Stop SPI clock generation when in Wait mode.
0 SPC0	**Serial Pin Control Bit 0** — This bit enables bidirectional pin configurations as shown in Table 446. In master mode, a change of this bit will abort a transmission in progress and force the SPI system into idle state.

Notes:
283.n is used later in this document as a placeholder for the selected transfer width.

Table 446. Bidirectional pin configurations

Pin Mode	SPC0	BIDIROE	MISO	MOSI
Master Mode of Operation				
Normal	0	X	Master In	Master Out
Bidirectional	1	0	MISO not used by SPI	Master In
		1		Master I/O
Slave Mode of Operation				
Normal	0	X	Slave Out	Slave In
Bidirectional	1	0	Slave In	MOSI not used by SPI
		1	Slave I/O	

6.12.3.2.3 SPI baud rate register (SPIBR)

Table 447. SPI baud rate register (SPIBR)

Module Base +0x0002

	7	6	5	4	3	2	1	0
R	0	SPPR2	SPPR1	SPPR0	0	SPR2	SPR1	SPR0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
284.Read: Anytime
Write: Anytime. Writes to the reserved bits have no effect

Table 448. SPIBR field descriptions

Field	Description
6–4 SPPR[2:0]	**SPI Baud Rate Preselection Bits** — These bits specify the SPI baud rates as shown in Table 449. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.
2–0 SPR[2:0]	**SPI Baud Rate Selection Bits** — These bits specify the SPI baud rates as shown in Table 449. In master mode, a change of these bits will abort a transmission in progress and force the SPI system into idle state.

The baud rate divisor equation is as follows:

$BaudRateDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}$

The baud rate can be calculated with the following equation:

Baud Rate = BusClock / BaudRateDivisor

Note

For maximum allowed baud rates, refer to the SPI Electrical Specification in the Electricals chapter of this data sheet.

Table 449. Example SPI baud rate selection (25 MHz bus clock)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	0	0	0	0	0	2	12.5 Mbit/s
0	0	0	0	0	1	4	6.25 Mbit/s
0	0	0	0	1	0	8	3.125 Mbit/s
0	0	0	0	1	1	16	1.5625 Mbit/s
0	0	0	1	0	0	32	781.25 kbit/s
0	0	0	1	0	1	64	390.63 kbit/s
0	0	0	1	1	0	128	195.31 kbit/s
0	0	0	1	1	1	256	97.66 kbit/s
0	0	1	0	0	0	4	6.25 Mbit/s
0	0	1	0	0	1	8	3.125 Mbit/s
0	0	1	0	1	0	16	1.5625 Mbit/s
0	0	1	0	1	1	32	781.25 kbit/s
0	0	1	1	0	0	64	390.63 kbit/s
0	0	1	1	0	1	128	195.31 kbit/s
0	0	1	1	1	0	256	97.66 kbit/s
0	0	1	1	1	1	512	48.83 kbit/s
0	1	0	0	0	0	6	4.16667 Mbit/s
0	1	0	0	0	1	12	2.08333 Mbit/s
0	1	0	0	1	0	24	1.04167 Mbit/s
0	1	0	0	1	1	48	520.83 kbit/s
0	1	0	1	0	0	96	260.42 kbit/s
0	1	0	1	0	1	192	130.21 kbit/s
0	1	0	1	1	0	384	65.10 kbit/s
0	1	0	1	1	1	768	32.55 kbit/s
0	1	1	0	0	0	8	3.125 Mbit/s
0	1	1	0	0	1	16	1.5625 Mbit/s
0	1	1	0	1	0	32	781.25 kbit/s
0	1	1	0	1	1	64	390.63 kbit/s
0	1	1	1	0	0	128	195.31 kbit/s
0	1	1	1	0	1	256	97.66 kbit/s

Table 449. Example SPI baud rate selection (25 MHz bus clock) (continued)

SPPR2	SPPR1	SPPR0	SPR2	SPR1	SPR0	Baud Rate Divisor	Baud Rate
0	1	1	1	1	0	512	48.83 kbit/s
0	1	1	1	1	1	1024	24.41 kbit/s
1	0	0	0	0	0	10	2.5 Mbit/s
1	0	0	0	0	1	20	1.25 Mbit/s
1	0	0	0	1	0	40	625 kbit/s
1	0	0	0	1	1	80	312.5 kbit/s
1	0	0	1	0	0	160	156.25 kbit/s
1	0	0	1	0	1	320	78.13 kbit/s
1	0	0	1	1	0	640	39.06 kbit/s
1	0	0	1	1	1	1280	19.53 kbit/s
1	0	1	0	0	0	12	2.08333 Mbit/s
1	0	1	0	0	1	24	1.04167 Mbit/s
1	0	1	0	1	0	48	520.83 kbit/s
1	0	1	0	1	1	96	260.42 kbit/s
1	0	1	1	0	0	192	130.21 kbit/s
1	0	1	1	0	1	384	65.10 kbit/s
1	0	1	1	1	0	768	32.55 kbit/s
1	0	1	1	1	1	1536	16.28 kbit/s
1	1	0	0	0	0	14	1.78571 Mbit/s
1	1	0	0	0	1	28	892.86 kbit/s
1	1	0	0	1	0	56	446.43 kbit/s
1	1	0	0	1	1	112	223.21 kbit/s
1	1	0	1	0	0	224	111.61 kbit/s
1	1	0	1	0	1	448	55.80 kbit/s
1	1	0	1	1	0	896	27.90 kbit/s
1	1	0	1	1	1	1792	13.95 kbit/s
1	1	1	0	0	0	16	1.5625 Mbit/s
1	1	1	0	0	1	32	781.25 kbit/s
1	1	1	0	1	0	64	390.63 kbit/s
1	1	1	0	1	1	128	195.31 kbit/s
1	1	1	1	0	0	256	97.66 kbit/s
1	1	1	1	0	1	512	48.83 kbit/s
1	1	1	1	1	0	1024	24.41 kbit/s
1	1	1	1	1	1	2048	12.21 kbit/s

6.12.3.2.4 SPI status register (SPISR)

)

Table 450. SPI status register (SPISR)

Module Base +0x0003

	7	6	5	4	3	2	1	0
R	SPIF	0	SPTEF	MODF	0	0	0	0
W								
Reset	0	0	1	0	0	0	0	0

= Unimplemented or Reserved

Notes:
285.Read: Anytime
Write: Has no effect

Table 451. SPISR field descriptions

Field	Description
7 SPIF	**SPIF Interrupt Flag** — This bit is set after received data has been transferred into the SPI data register. For information about clearing SPIF Flag, refer to Table 452. 0 Transfer not yet complete. 1 New data copied to SPIDR.
5 SPTEF	**SPI Transmit Empty Interrupt Flag** — If set, this bit indicates that the transmit data register is empty. For information about clearing this bit and placing data into the transmit data register, refer to Table 453. 0 SPI data register not empty. 1 SPI data register empty.
4 MODF	**Mode Fault Flag** — This bit is set if the $\overline{SS}$ input becomes low while the SPI is configured as a master and mode fault detection is enabled, MODFEN bit of SPICR2 register is set. Refer to MODFEN bit description in SPI control register 2 (SPICR2)". The flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to the SPI control register 1. 0 Mode fault has not occurred. 1 Mode fault has occurred.

Table 452. SPIF interrupt flag clearing sequence

XFRW Bit	SPIF Interrupt Flag Clearing Sequence			
0	Read SPISR with SPIF == 1	then	Read SPIDRL	
1	Read SPISR with SPIF == 1	then	Byte Read SPIDRL (286)	
			or	
			Byte Read SPIDRH (287)	Byte Read SPIDRL
			or	
			Word Read (SPIDRH:SPIDRL)	

Notes:
286.Data in SPIDRH is lost in this case.
287.SPIDRH can be read repeatedly without any effect on SPIF. SPIF Flag is cleared only by the read of SPIDRL after reading SPISR with SPIF == 1

Table 453. SPTEF interrupt flag clearing sequence

XFRW Bit	SPTEF Interrupt Flag Clearing Sequence			
0	Read SPISR with SPTEF == 1	then	Write to SPIDRL (288)	
1	Read SPISR with SPTEF == 1	then	Byte Write to SPIDRL (288), (289)	
			or	
			Byte Write to SPIDRH (288),(290)	Byte Write to SPIDRL (288)
			or	
			Word Write to (SPIDRH:SPIDRL) (288)	

Notes:
288.Any write to SPIDRH or SPIDRL with SPTEF == 0 is effectively ignored.
289.Data in SPIDRH is undefined in this case.
290.SPIDRH can be written repeatedly without any effect on SPTEF. SPTEF Flag is cleared only by writing to SPIDRL after reading SPISR with SPTEF == 1.

6.12.3.2.5 SPI data register (SPIDR = SPIDRH:SPIDRL)

Table 454. SPI data register high (SPIDRH)

Module Base +0x0004

	7	6	5	4	3	2	1	0
R	R15	R14	R13	R12	R11	R10	R9	R8
W	T15	T14	T13	T12	T11	T10	T9	T8
Reset	0	0	0	0	0	0	0	0

Table 455. SPI data register low (SPIDRL)

Module Base +0x0005

	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Notes:
291.Read: Anytime. Read data only valid when SPIF is set
Write: Anytime

The SPI data register is both the input and output register for SPI data. A write to this register allows data to be queued and transmitted. For an SPI configured as a master, queued data is transmitted immediately after the previous transmission has completed. The SPI transmitter empty flag SPTEF in the SPISR register indicates when the SPI data register is ready to accept new data. Received data in the SPIDR is valid when SPIF is set.

If SPIF is cleared and data has been received, the received data is transferred from the receive shift register to the SPIDR and SPIF is set.

If SPIF is set and not serviced, and a second data value has been received, the second received data is kept as valid data in the receive shift register until the start of another transmission. The data in the SPIDR does not change.

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced before the start of a third transmission, the data in the receive shift register is transferred into the SPIDR and SPIF remains set (see Figure 81).

If SPIF is set and valid data is in the receive shift register, and SPIF is serviced after the start of a third transmission, the data in the receive shift register has become invalid and is not transferred into the SPIDR (see Figure 82).

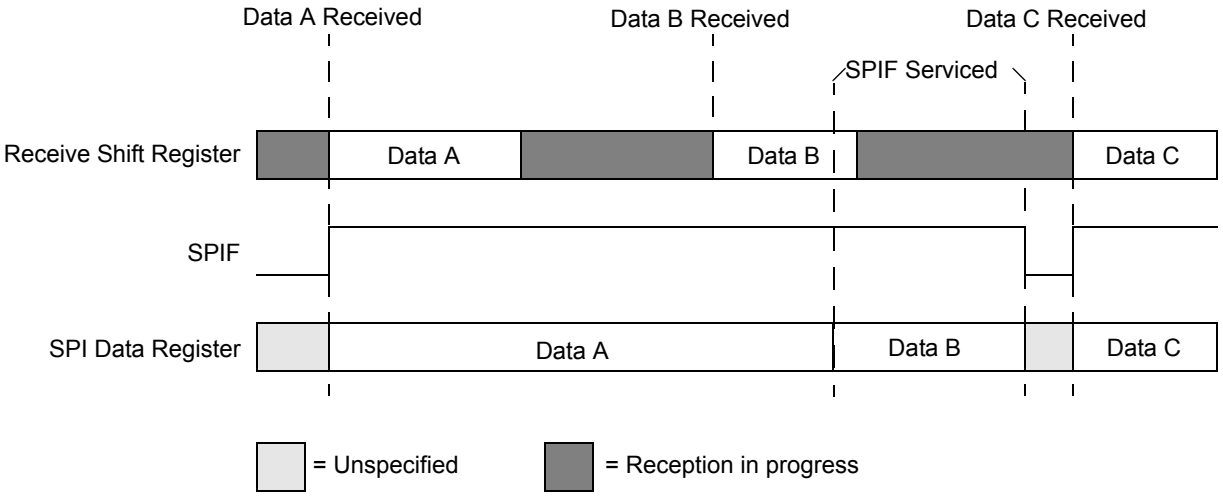


Figure 81. Reception with SPIF serviced in time

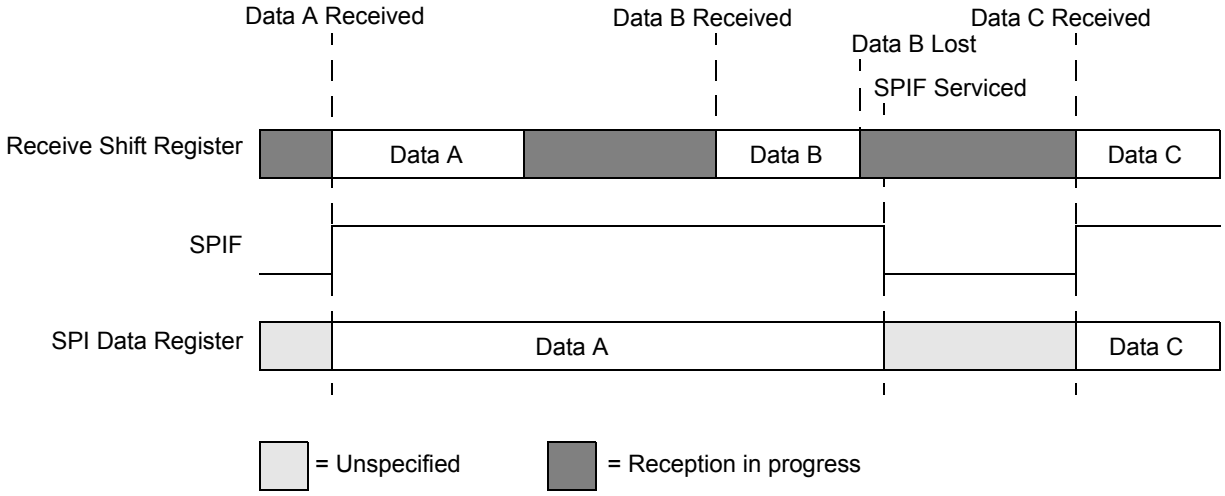


Figure 82. Reception with SPIF serviced too late

6.12.4 Functional description

The SPI module allows a duplex, synchronous, serial communication between the MCU and peripheral devices. Software can poll the SPI status flags or SPI operation can be interrupt driven.

The SPI system is enabled by setting the SPI enable (SPE) bit in SPI control register 1. While SPE is set, the four associated SPI port pins are dedicated to the SPI function as:

- Slave select ($\overline{SS}$)
- Serial clock (SCK)
- Master out/slave in (MOSI)
- Master in/slave out (MISO)

The main element of the SPI system is the SPI data register. The n-bit (292) data register in the master and the n-bit (292) data register in the slave are linked by the MOSI and MISO pins to form a distributed 2n-bit (292) register. When a data transfer operation is performed, this 2n-bit (292) register is serially shifted n (292) bit positions by the S-clock from the master, so data is exchanged between the master and the slave. Data written to the master SPI data register becomes the output data for the slave, and data read from the master SPI data register after a transfer operation is the input data from the slave.

A read of SPISR with SPTEF = 1 followed by a write to SPIDR puts data into the transmit data register. When a transfer is complete and SPIF is cleared, received data is moved into the receive data register. This data register acts as the SPI receive data register for reads and as the SPI transmit data register for writes. A common SPI data register address is shared for reading data from the read data buffer and for writing data to the transmit data register.

The clock phase control bit (CPHA) and a clock polarity control bit (CPOL) in the SPI control register 1 (SPICR1) select one of four possible clock formats to be used by the SPI system. The CPOL bit simply selects a non-inverted or inverted clock. The CPHA bit is used to accommodate two fundamentally different protocols by sampling data on odd numbered SCK edges or on even numbered SCK edges (see Transmission formats").

The SPI can be configured to operate as a master or as a slave. When the MSTR bit in SPI control register1 is set, master mode is selected, when the MSTR bit is clear, slave mode is selected.

Notes:
292.n depends on the selected transfer width, refer to SPI control register 2 (SPICR2)

Note

A change of CPOL or MSTR bit while there is a received byte pending in the receive shift register will destroy the received byte and must be avoided.

6.12.4.1 Master mode

The SPI operates in master mode when the MSTR bit is set. Only a master SPI module can initiate transmissions. A transmission begins by writing to the master SPI data register. If the shift register is empty, data immediately transfers to the shift register. Data begins shifting out on the MOSI pin under the control of the serial clock.

- Serial clock

  The SPR2, SPR1, and SPR0 baud rate selection bits, in conjunction with the SPPR2, SPPR1, and SPPR0 baud rate preselection bits in the SPI baud rate register, control the baud rate generator and determine the speed of the transmission. The SCK pin is the SPI clock output. Through the SCK pin, the baud rate generator of the master controls the shift register of the slave peripheral.

- MOSI, MISO pin

  In master mode, the function of the serial data output pin (MOSI) and the serial data input pin (MISO) is determined by the SPC0 and BIDIROE control bits.

- $\overline{SS}$ pin

  If MODFEN and SSOE are set, the $\overline{SS}$ pin is configured as slave select output. The $\overline{SS}$ output becomes low during each transmission and is high when the SPI is in idle state.

  If MODFEN is set and SSOE is cleared, the $\overline{SS}$ pin is configured as input for detecting mode fault error. If the $\overline{SS}$ input becomes low this indicates a mode fault error where another master tries to drive the MOSI and SCK lines. In this case, the SPI immediately switches to slave mode, by clearing the MSTR bit and also disables the slave output buffer MISO (or SISO in bidirectional mode). The result is that all outputs are disabled and SCK, MOSI, and MISO are inputs. If a transmission is in progress when the mode fault occurs, the transmission is aborted and the SPI is forced into idle state.

  This mode fault error also sets the mode fault (MODF) flag in the SPI status register (SPISR). If the SPI interrupt enable bit (SPIE) is set when the MODF flag becomes set, then an SPI interrupt sequence is also requested.

  When a write to the SPI data register in the master occurs, there is a half SCK-cycle delay. After the delay, SCK is started within the master. The rest of the transfer operation differs slightly, depending on the clock format specified by the SPI clock phase bit, CPHA, in SPI control register 1 (see Transmission formats").

Note

A change of the bits CPOL, CPHA, SSOE, LSBFE, XFRW, MODFEN, SPC0, or BIDIROE with SPC0 set, SPPR2-SPPR0 and SPR2-SPR0 in master mode will abort a transmission in progress and force the SPI into idle state. The remote slave cannot detect this, therefore the master must ensure that the remote slave is returned to idle state.

6.12.4.2 Slave mode

The SPI operates in slave mode when the MSTR bit in SPI control register 1 is clear.

- Serial clock

  In Slave mode, SCK is the SPI clock input from the master.

- MISO, MOSI pin

  In Slave mode, the function of the serial data output pin (MISO) and serial data input pin (MOSI) is determined by the SPC0 bit and BIDIROE bit in SPI control register 2.

- $\overline{SS}$ pin

  The $\overline{SS}$ pin is the slave select input. Before a data transmission occurs, the $\overline{SS}$ pin of the slave SPI must be low. $\overline{SS}$ must remain low until the transmission is complete. If $\overline{SS}$ goes high, the SPI is forced into idle state.

  The $\overline{SS}$ input also controls the serial data output pin, if $\overline{SS}$ is high (not selected), the serial data output pin is high-impedance, and, if $\overline{SS}$ is low, the first bit in the SPI data register is driven out of the serial data output pin. Also, if the slave is not selected ($\overline{SS}$ is high), then the SCK input is ignored and no internal shifting of the SPI shift register occurs.

  Although the SPI is capable of duplex operation, some SPI peripherals are capable of only receiving SPI data in a Slave mode. For these simpler devices, there is no serial data out pin.

Note

When peripherals with duplex capability are used, take care not to simultaneously enable two receivers whose serial outputs drive the same system slave's serial data output line.

As long as no more than one slave device drives the system slave's serial data output line, it is possible for several slaves to receive the same transmission from a master, although the master would not receive return information from all of the receiving slaves.

If the CPHA bit in SPI control register 1 is clear, odd numbered edges on the SCK input cause the data at the serial data input pin to be latched. Even numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

If the CPHA bit is set, even numbered edges on the SCK input cause the data at the serial data input pin to be latched. Odd numbered edges cause the value previously latched from the serial data input pin to shift into the LSB or MSB of the SPI shift register, depending on the LSBFE bit.

When CPHA is set, the first edge is used to get the first data bit onto the serial data output pin. When CPHA is clear and the $\overline{SS}$ input is low (slave selected), the first bit of the SPI data is driven out of the serial data output pin. After the nth [(293)] shift, the transfer is considered complete and the received data is transferred into the SPI data register. To indicate transfer is complete, the SPIF flag in the SPI status register is set.

Notes:
293.n depends on the selected transfer width, refer to Section 6.12.3.2.2, "SPI control register 2 (SPICR2)"

Note

A change of the bits CPOL, CPHA, SSOE, LSBFE, MODFEN, SPC0, or BIDIROE with SPC0 set in slave mode will corrupt a transmission in progress and must be avoided.

6.12.4.3 Transmission formats

During an SPI transmission, data is transmitted (shifted out serially) and received (shifted in serially) simultaneously. The serial clock (SCK) synchronizes shifting and sampling of the information on the two serial data lines. A slave select line allows selection of an individual slave SPI device; slave devices that are not selected do not interfere with SPI bus activities. Optionally, on a master SPI device, the slave select line can be used to indicate multiple-master bus contention.

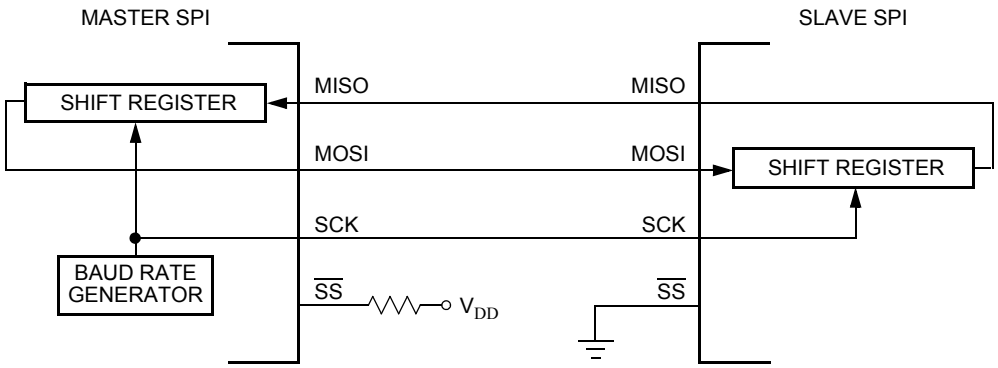


Figure 83. Master/slave transfer block diagram

6.12.4.3.1 Clock phase and polarity controls

Using two bits in the SPI control register 1, software selects one of four combinations of serial clock phase and polarity.

The CPOL clock polarity control bit specifies an active high or low clock and has no significant effect on the transmission format.

The CPHA clock phase control bit selects one of two fundamentally different transmission formats.

Clock phase and polarity should be identical for the master SPI device and the communicating slave device. In some cases, the phase and polarity are changed between transmissions to allow a master device to communicate with peripheral slaves having different requirements.

6.12.4.3.2 CPHA = 0 transfer format

The first edge on the SCK line is used to clock the first data bit of the slave into the master and the first data bit of the master into the slave. In some peripherals, the first bit of the slave's data is available at the slave's data out pin as soon as the slave is selected. In this format, the first SCK edge is issued a half cycle after $\overline{SS}$ has become low.

A half SCK cycle later, the second edge appears on the SCK line. When this second edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the shift register, depending on LSBFE bit.

After this second edge, the next bit of the SPI master data is transmitted out of the serial data output pin of the master to the serial input pin on the slave. This process continues for a total of 16 edges on the SCK line, with data being latched on odd numbered edges and shifted on even numbered edges.

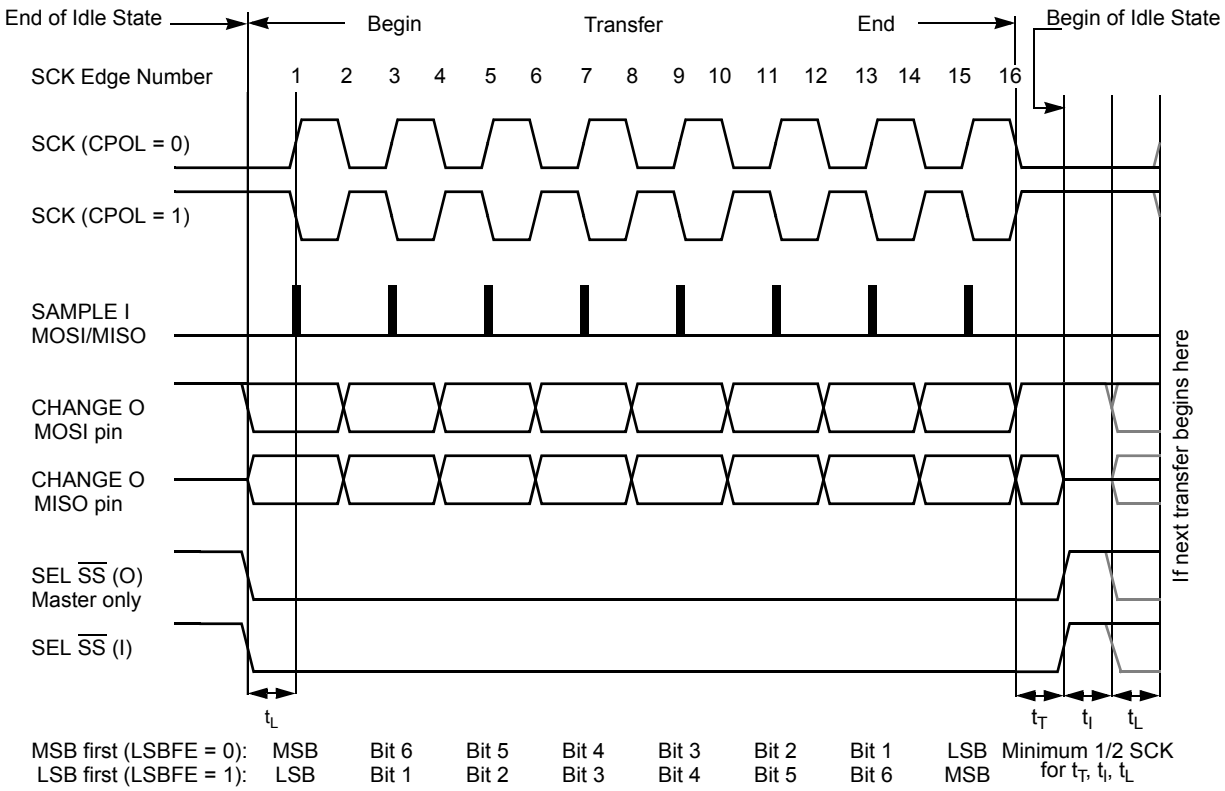
Data reception is double buffered. Data is shifted serially into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After 2n [(294)] (last) SCK edges:

- Data that was previously in the master SPI data register should now be in the slave data register and the data that was in the slave data register should be in the master.
- The SPIF flag in the SPI status register is set, indicating that the transfer is complete.

Notes:
294.n depends on the selected transfer width, refer to SPI control register 2 (SPICR2)

Figure 84 is a timing diagram of an SPI transfer where CPHA = 0. SCK waveforms are shown for CPOL = 0 and CPOL = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave and the MOSI signal is the output from the master. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general purpose output not affecting the SPI.



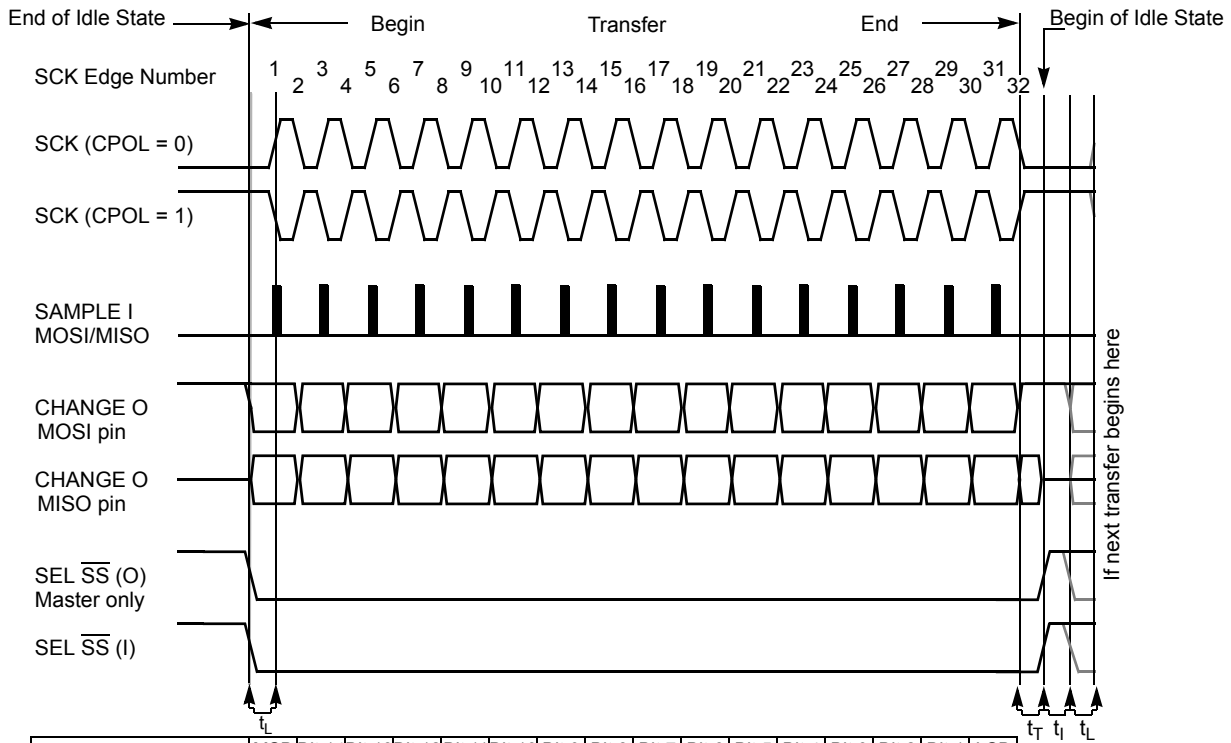
t_L = Minimum leading time before the first SCK edge

t_T = Minimum trailing time after the last SCK edge

t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)

t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 84. SPI clock format 0 (CPHA = 0), with 8-bit transfer width selected (XFRW = 0)



t_L = Minimum leading time before the first SCK edge

t_T = Minimum trailing time after the last SCK edge

t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time)

t_L, t_T, and t_I are guaranteed for the master mode and required for the slave mode.

Figure 85. SPI clock format 0 (CPHA = 0), with 16-bit transfer width selected (XFRW = 1)

In Slave mode, if the $\overline{SS}$ line is not deasserted between the successive transmissions then the content of the SPI data register is not transmitted; instead the last received data is transmitted. If the $\overline{SS}$ line is deasserted for at least minimum idle time (half SCK cycle) between successive transmissions, then the content of the SPI data register is transmitted.

In Master mode with slave select output enabled, the $\overline{SS}$ line is always deasserted and reasserted between successive transfers for at least minimum idle time.

6.12.4.3.3 CPHA = 1 transfer format

Some peripherals require the first SCK edge before the first data bit becomes available at the data out pin, the second edge clocks data into the system. In this format, the first SCK edge is issued by setting the CPHA bit at the beginning of the n [(295)] -cycle transfer operation.

The first edge of SCK occurs immediately after the half SCK clock cycle synchronization delay. This first edge commands the slave to transfer its first data bit to the serial data input pin of the master.

A half SCK cycle later, the second edge appears on the SCK pin. This is the latching edge for both the master and slave.

When the third edge occurs, the value previously latched from the serial data input pin is shifted into the LSB or MSB of the SPI shift register, depending on LSBFE bit. After this edge, the next bit of the master data is coupled out of the serial data output pin of the master to the serial input pin on the slave.

This process continues for a total of n [(295)] edges on the SCK line with data being latched on even numbered edges and shifting taking place on odd numbered edges.

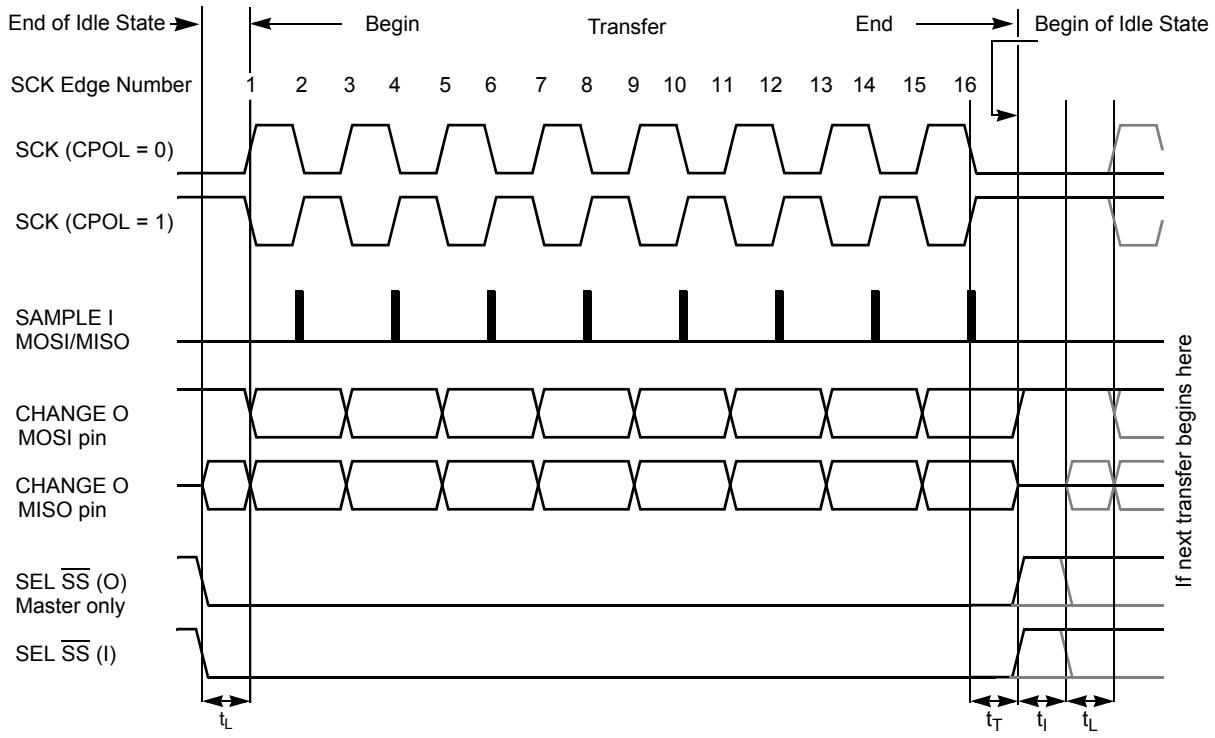
Data reception is double buffered, data is serially shifted into the SPI shift register during the transfer and is transferred to the parallel SPI data register after the last bit is shifted in.

After 2n [(295)] SCK edges:

Notes:
295.n depends on the selected transfer width, refer to SPI control register 2 (SPICR2)

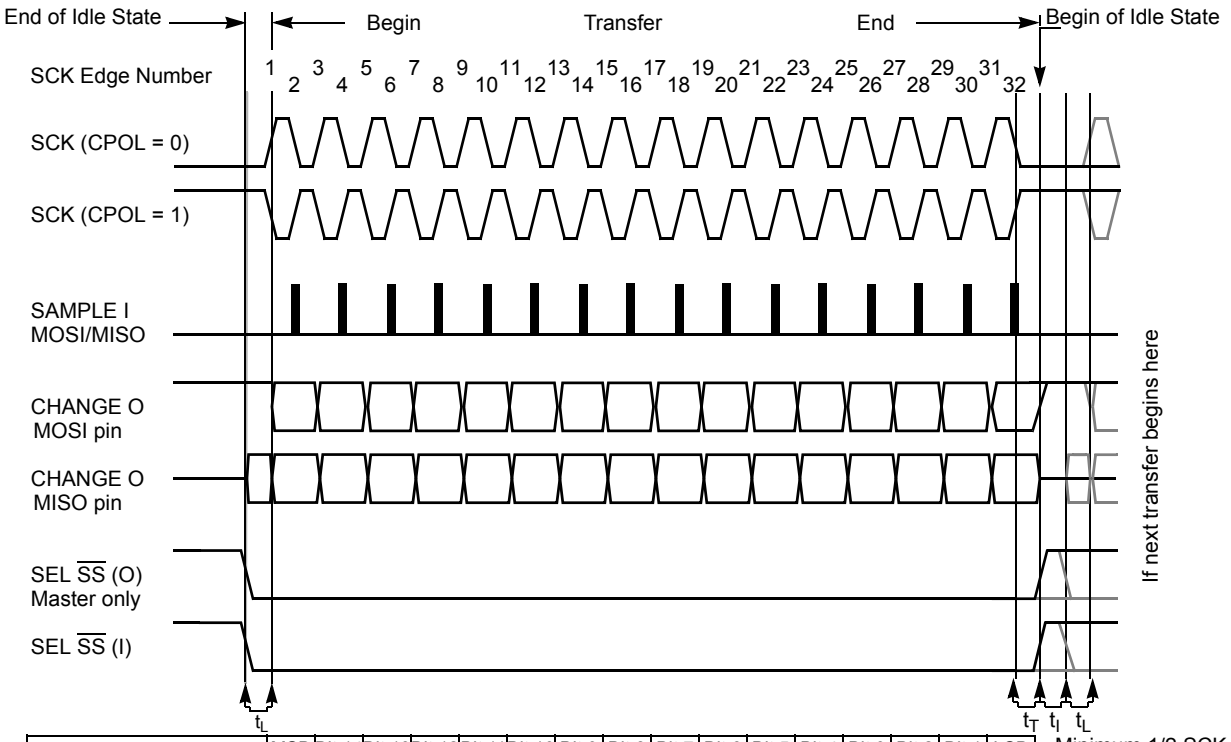
- Data that was previously in the SPI data register of the master is now in the data register of the slave, and data that was in the data register of the slave is in the master.
- The SPIF flag bit in SPISR is set indicating that the transfer is complete.

Figure 86 shows two clocking variations for CPHA = 1. The diagram may be interpreted as a master or slave timing diagram because the SCK, MISO, and MOSI pins are connected directly between the master and the slave. The MISO signal is the output from the slave, and the MOSI signal is the output from the master. The $\overline{SS}$ line is the slave select input to the slave. The $\overline{SS}$ pin of the master must be either high or reconfigured as a general purpose output not affecting the SPI.



t_L = Minimum leading time before the first SCK edge, not required for back-to-back transfers
t_T = Minimum trailing time after the last SCK edge
t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back-to-back transfers

Figure 86. SPI clock format 1 (CPHA = 1), with 8-bit transfer width selected (XFRW = 0)



t_L = Minimum leading time before the first SCK edge, not required for back-to-back transfers

t_T = Minimum trailing time after the last SCK edge

t_I = Minimum idling time between transfers (minimum $\overline{SS}$ high time), not required for back-to-back transfers

Figure 87. SPI clock format 1 (CPHA = 1), with 16-bit transfer width selected (XFRW = 1)

The $\overline{SS}$ line can remain active low between successive transfers (can be tied low at all times). This format is sometimes preferred in systems having a single fixed master and a single slave that drive the MISO data line.

- Back-to-back transfers in master mode

  In Master mode, if a transmission has completed and new data is available in the SPI data register, this data is sent out immediately without a trailing and minimum idle time.

  The SPI interrupt request flag (SPIF) is common to both the Master and Slave modes. SPIF gets set one half SCK cycle after the last SCK edge.

6.12.4.4 SPI baud rate generation

Baud rate generation consists of a series of divider stages. Six bits in the SPI baud rate register (SPPR2, SPPR1, SPPR0, SPR2, SPR1, and SPR0) determine the divisor to the SPI module clock which results in the SPI baud rate.

The SPI clock rate is determined by the product of the value in the baud rate preselection bits (SPPR2–SPPR0) and the value in the baud rate selection bits (SPR2–SPR0). The module clock divisor equation is shown in the following Equation .

$$BaudRateDivisor = (SPPR + 1) \bullet 2^{(SPR + 1)}$$

When all bits are clear (the default condition), the SPI module clock is divided by 2. When the selection bits (SPR2–SPR0) are 001 and the preselection bits (SPPR2–SPPR0) are 000, the module clock divisor becomes 4. When the selection bits are 010, the module clock divisor becomes 8, etc.

When the preselection bits are 001, the divisor determined by the selection bits is multiplied by 2. When the preselection bits are 010, the divisor is multiplied by 3, etc. See Table 449 for baud rate calculations for all bit conditions, based on a 25 MHz bus clock. The two sets of selects allows the clock to be divided by a non-power of two to achieve other baud rates such as divide by 6, divide by 10, etc.

The baud rate generator is activated only when the SPI is in master mode and a serial transfer is taking place. In the other cases, the divider is disabled to decrease I_{DD} current.

Note

For maximum allowed baud rates, refer to the SPI Electrical Specification in the Electricals chapter of this data sheet.

6.12.4.5 Special features

6.12.4.5.1 $\overline{SS}$ output

The $\overline{SS}$ output feature automatically drives the $\overline{SS}$ pin low during transmission to select external devices and drives it high during idle to deselect external devices. When $\overline{SS}$ output is selected, the $\overline{SS}$ output pin is connected to the $\overline{SS}$ input pin of the external device.

The $\overline{SS}$ output is available only in master mode during normal SPI operation by asserting SSOE and MODFEN bit as shown in Table 443.

The mode fault feature is disabled while $\overline{SS}$ output is enabled.

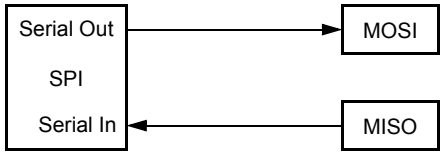
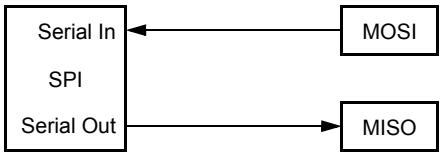
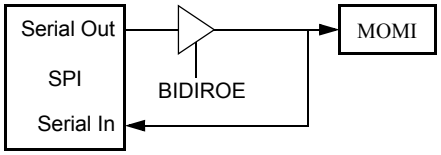
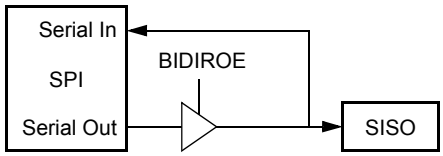
Note

Care must be taken when using the $\overline{SS}$ output feature in a multimaster system because the mode fault feature is not available for detecting system errors between masters.

6.12.4.5.2 Bidirectional mode (MOMI or SISO)

The bidirectional mode is selected when the SPC0 bit is set in SPI control register 2 (see Table 456). In this mode, the SPI uses only one serial data pin for the interface with external device(s). The MSTR bit decides which pin to use. The MOSI pin becomes the serial data I/O (MOMI) pin for the Master mode, and the MISO pin becomes serial data I/O (SISO) pin for the Slave mode. The MISO pin in Master mode and MOSI pin in Slave mode are not used by the SPI.

Table 456. Normal mode and bidirectional mode

When SPE = 1	Master Mode MSTR = 1	Slave Mode MSTR = 0
Normal Mode SPC0 = 0		
Bidirectional Mode SPC0 = 1		

The direction of each serial I/O pin depends on the BIDIROE bit. If the pin is configured as an output, serial data from the shift register is driven out on the pin. The same pin is also the serial input to the shift register.

- The SCK is output for the Master mode and input for the Slave mode.
- The $\overline{SS}$ is the input or output for the Master mode, and it is always the input for the Slave mode.
- The bidirectional mode does not affect SCK and $\overline{SS}$ functions.

Note

In bidirectional Master mode, with mode fault enabled, both data pins MISO and MOSI can be occupied by the SPI, though MOSI is normally used for transmissions in bidirectional mode and MISO is not used by the SPI. If a mode fault occurs, the SPI is automatically switched to Slave mode. In this case, MISO becomes occupied by the SPI and MOSI is not used. This must be considered, if the MISO pin is used for another purpose.

6.12.4.6 Error conditions

The SPI has one error condition:

- Mode fault error

6.12.4.6.1 Mode fault error

If the $\overline{SS}$ input becomes low while the SPI is configured as a master, it indicates a system error where more than one master may be trying to drive the MOSI and SCK lines simultaneously. This condition is not permitted in normal operation, the MODF bit in the SPI status register is set automatically, provided the MODFEN bit is set.

In the special case where the SPI is in Master mode and MODFEN bit is cleared, the $\overline{SS}$ pin is not used by the SPI. In this special case, the mode fault error function is inhibited and MODF remains cleared. In case the SPI system is configured as a slave, the $\overline{SS}$ pin is a dedicated input pin. Mode fault error doesn't occur in Slave mode.

If a mode fault error occurs, the SPI is switched to Slave mode, with the exception that the slave output buffer is disabled. So SCK, MISO, and MOSI pins are forced to be high-impedance inputs to avoid any possibility of conflict with another output driver. A transmission in progress is aborted and the SPI is forced into idle state.

If the mode fault error occurs in the bidirectional mode for a SPI system configured in master mode, output enable of the MOMI (MOSI in bidirectional mode) is cleared if it was set. No mode fault error occurs in the bidirectional mode for SPI system configured in Slave mode.

The mode fault flag is cleared automatically by a read of the SPI status register (with MODF set) followed by a write to SPI control register 1. If the mode fault flag is cleared, the SPI becomes a normal master or slave again.

Note

If a mode fault error occurs and a received data byte is pending in the receive shift register, this data byte will be lost.

6.12.4.7 Low power mode options

6.12.4.7.1 SPI in run mode

In Run mode with the SPI system enable (SPE) bit in the SPI control register clear, the SPI system is in a low-power, disabled state. SPI registers remain accessible, but clocks to the core of this module are disabled.

6.12.4.7.2 SPI in wait mode

SPI operation in Wait mode depends upon the state of the SPISWAI bit in SPI control register 2.

- If SPISWAI is clear, the SPI operates normally when the CPU is in Wait mode
- If SPISWAI is set, SPI clock generation ceases and the SPI module enters a power conservation state when the CPU is in Wait mode.
  - If SPISWAI is set and the SPI is configured for master, any transmission and reception in progress stops at Wait mode entry. The transmission and reception resumes when the SPI exits Wait mode.
  - If SPISWAI is set and the SPI is configured as a slave, any transmission and reception in progress continues if the SCK continues to be driven from the master. This keeps the slave synchronized to the master and the SCK.

    If the master transmits several bytes while the slave is in Wait mode, the slave will continue to send out bytes consistent with the operation mode at the start of Wait mode (i.e., if the slave is currently sending its SPIDR to the master, it will continue to send the same byte. Else if the slave is currently sending the last received byte from the master, it will continue to send each previous master byte).

Note

Care must be taken when expecting data from a master while the slave is in Wait or Stop mode. Even though the shift register will continue to operate, the rest of the SPI is shut down (i.e., a SPIF interrupt will **not** be generated until exiting Stop or Wait mode). The byte from the shift register will not be copied into the SPIDR register until after the slave SPI has exited Wait or Stop mode. In Slave mode, a received byte pending in the receive shift register will be lost when entering Wait or Stop mode. An SPIF flag and SPIDR copy is generated only if Wait mode is entered or exited during a tranmission. If the slave enters Wait mode in idle mode and exits Wait mode in Idle mode, neither a SPIF nor a SPIDR copy will occur.

6.12.4.7.3 SPI in stop mode

Stop mode is dependent on the system. The SPI enters Stop mode when the module clock is disabled (held high or low). If the SPI is in Master mode and exchanging data when the CPU enters Stop mode, the transmission is frozen until the CPU exits Stop mode. After stop, data to and from the external SPI is exchanged correctly. In Slave mode, the SPI will stay synchronized with the master.

The Stop mode is not dependent on the SPISWAI bit.

6.12.4.7.4 Reset

The reset values of registers and signals are described in Memory map and register definition", which details the registers and their bit fields.

- If a data transmission occurs in Slave mode after reset without a write to SPIDR, it will transmit garbage, or the data last received from the master before the reset.
- Reading from the SPIDR after reset will always read zeros.

6.12.4.7.5 Interrupts

The SPI only originates interrupt requests when SPI is enabled (SPE bit in SPICR1 set). The following is a description of how the SPI makes a request and how the MCU should acknowledge that request. The interrupt vector offset and interrupt priority are chip dependent.

The interrupt flags MODF, SPIF, and SPTEF are logically ORed to generate an interrupt request.

6.12.4.7.5.1 MODF

MODF occurs when the master detects an error on the $\overline{SS}$ pin. The master SPI must be configured for the MODF feature (see Table 443). After MODF is set, the current transfer is aborted and the following bit is changed:

- MSTR = 0, The master bit in SPICR1 resets.

The MODF interrupt is reflected in the status register MODF flag. Clearing the flag will also clear the interrupt. This interrupt will stay active while the MODF flag is set. MODF has an automatic clearing process which is described in SPI status register (SPISR)".

6.12.4.7.5.2 SPIF

SPIF occurs when new data has been received and copied to the SPI data register. After SPIF is set, it does not clear until it is serviced. SPIF has an automatic clearing process, which is described in SPI status register (SPISR)".

6.12.4.7.5.3 SPTEF

SPTEF occurs when the SPI data register is ready to accept new data. After SPTEF is set, it does not clear until it is serviced. SPTEF has an automatic clearing process, which is described in SPI status register (SPISR)".

6.13 NXP's scalable controller area network (S12MSCANV3)

6.13.1 Introduction

NXP's scalable controller area network (S12MSCANV3) definition is based on the MSCAN12 definition, which is the specific implementation of the MSCAN concept targeted for the M68HC12 microcontroller family.

The module is a communication controller implementing the CAN 2.0A/B protocol as defined in the Bosch specification dated September 1991. For users to fully understand the MSCAN specification, it is recommended that the Bosch specification be read first to familiarize the reader with the terms and concepts contained within this document.

Though not exclusively intended for automotive applications, CAN protocol is designed to meet the specific requirements of a vehicle serial data bus: real-time processing, reliable operation in the EMI environment of a vehicle, cost-effectiveness, and required bandwidth.

MSCAN uses an advanced buffer arrangement resulting in predictable real-time behavior and simplified application software.

6.13.1.1 Glossary

Table 457. Terminology

ACK	Acknowledge of CAN message
CAN	Controller Area Network
CRC	Cyclic Redundancy Code
EOF	End of Frame
FIFO	First-In-First-Out Memory
IFS	Inter-frame Sequence
SOF	Start of Frame
CPU bus	CPU related read/write data bus
CAN bus	CAN protocol related serial bus
oscillator clock	Direct clock from external oscillator
bus clock	CPU bus related clock
CAN clock	CAN protocol related clock

6.13.1.2 Block diagram

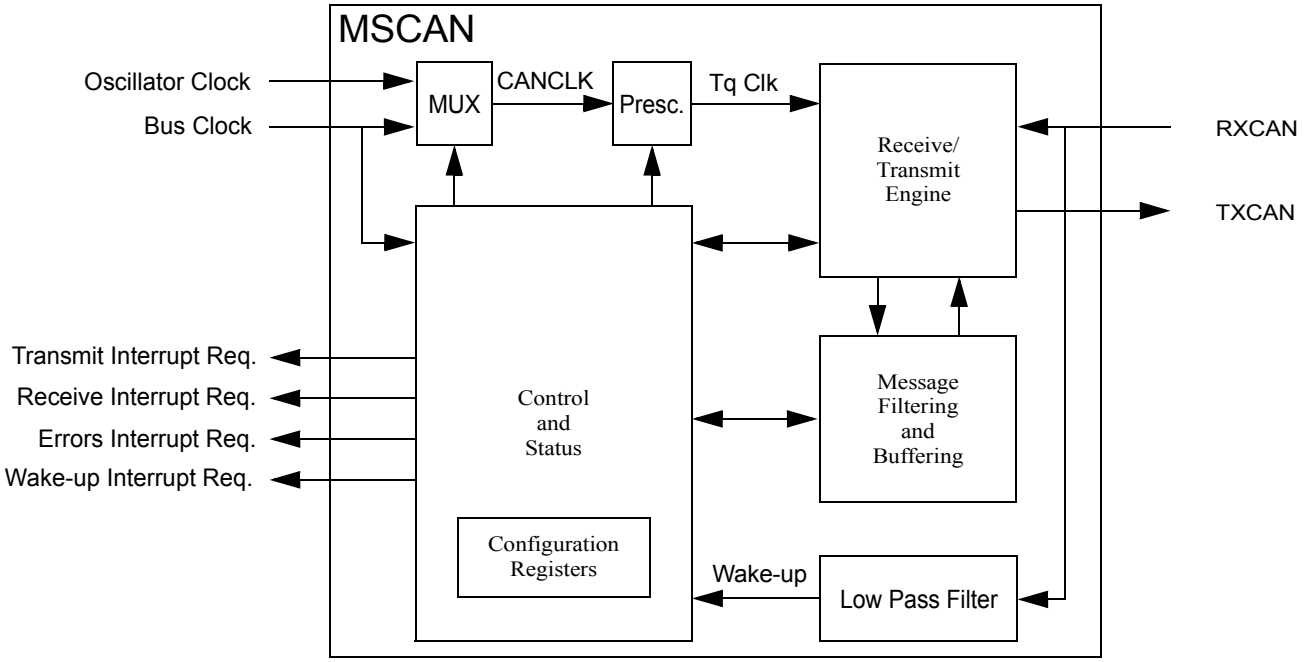


Figure 88. MSCAN block diagram

6.13.1.3 Features

The basic features of the MSCAN are as follows:

- Implementation of the CAN protocol — Version 2.0A/B
  - — Standard and extended data frames
  - — Zero to eight bytes data length
  - — Programmable bit rate up to 1.0 Mbps (296)
  - — Support for remote frames
- Five receive buffers with FIFO storage scheme
- Three transmit buffers with internal prioritization using a “local priority” concept
- Flexible maskable identifier filter supports two full-size (32-bit) extended identifier filters, or four 16-bit filters, or eight 8-bit filters

- Programmable wake-up functionality with integrated low-pass filter
- Programmable loopback mode supports self-test operation
- Programmable listen-only mode for monitoring of CAN bus
- Programmable bus-off recovery functionality
- Separate signalling and interrupt capabilities for all CAN receiver and transmitter error states (warning, error passive, bus-off)
- Programmable MSCAN clock source either bus clock or oscillator clock
- Internal timer for time-stamping of received and transmitted messages
- Three low-power modes: sleep, power down, and MSCAN enable
- Global initialization of configuration registers

Notes:
296.Depending on the actual bit timing and the clock jitter of the PLL.

6.13.1.4 Modes of operation

For a description of the specific MSCAN modes and the module operation related to the system operating modes refer to Modes of operation".

6.13.2 External signal description

The MSCAN uses two external pins.

Note

On MCUs with an integrated CAN physical interface (transceiver) the MSCAN interface is connected internally to the transceiver interface. In these cases, the external availability of signals TXCAN and RXCAN is optional.

6.13.2.1 RXCAN — CAN receiver input pin

RXCAN is the MSCAN receiver input pin.

6.13.2.2 TXCAN — CAN transmitter output pin

TXCAN is the MSCAN transmitter output pin. The TXCAN output pin represents the logic level on the CAN bus:

0 = Dominant state
1 = Recessive state

6.13.2.3 CAN system

A typical CAN system with MSCAN is shown in Figure 89. Each CAN station is connected physically to the CAN bus lines through a transceiver device. The transceiver is capable of driving the large current needed for the CAN bus and has current protection against defective CAN or defective stations.

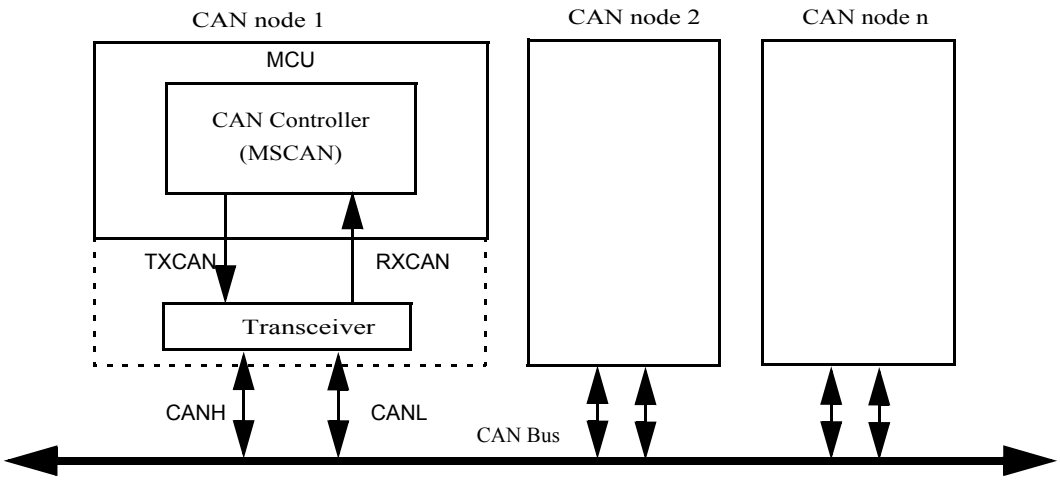


Figure 89. CAN system

6.13.3 Memory map and register definition

This section provides a detailed description of all registers accessible in the MSCAN.

6.13.3.1 Module memory map

Figure 458 gives an overview on all registers and their individual bits in the MSCAN memory map. The register address results from the addition of base address and address offset. The base address is determined at the MCU level and can be found in the MCU memory map description. The address offset is defined at the module level.

The MSCAN occupies 64 bytes in the memory space. The base address of the MSCAN module is determined at the MCU level when the MCU is defined. The register decode map is fixed and begins at the first address of the module address offset.

The detailed register descriptions follow in the order they appear in the register map.

Table 458. MSCAN register summary

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0000 CANCTL0	R	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ
	W								
0x0001 CANCTL1	R	CANE	CLKSRC	LOOPB	LISTEN	BORM	WUPM	SLPAK	INITAK
	W								
0x0002 CANBTR0	R	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
	W								
0x0003 CANBTR1	R	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
	W								
0x0004 CANRFLG	R	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
	W								
0x0005 CANRIER	R	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
	W								
0x0006 CANTFLG	R	0	0	0	0	0	TXE2	TXE1	TXE0
	W								
0x0007 CANTIER	R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
	W								
		= Unimplemented or Reserved							

MM9Z1_638

Table 458. MSCAN register summary (continued)

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
0x0008 CANTARQ	R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
	W								
0x0009 CANTAAK	R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
	W								
0x000A CANTBSEL	R	0	0	0	0	0	TX2	TX1	TX0
	W								
0x000B CANIDAC	R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
	W								
0x000C Reserved	R	0	0	0	0	0	0	0	0
	W								
0x000D CANMISC	R	0	0	0	0	0	0	0	BOHOLD
	W								
0x000E CANRXERR	R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
	W								
0x000F CANTXERR	R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
	W								
0x0010–0x0013 CANIDAR0–3	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
	W								
0x0014–0x0017 CANIDMRx	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
	W								
0x0018–0x001B CANIDAR4–7	R	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
	W								
0x001C–0x001F CANIDMR4–7	R	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
	W								
0x0020–0x002F CANRXFG	R	See Section 6.13.3.3, “Programmer’s model of message storage””							
	W								
0x0030–0x003F CANTXFG	R	See Section 6.13.3.3, “Programmer’s model of message storage””							
	W								

= Unimplemented or Reserved

6.13.3.2 Register descriptions

This section describes in detail all the registers and register bits in the MSCAN module. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order. All bits of all registers in this module are completely synchronous to internal clocks during a register read.

6.13.3.2.1 MSCAN control register 0 (CANCTL0)

The CANCTL0 register provides various control bits of the MSCAN module as described in Table 459.

Table 459. MSCAN Control Register 0 (CANCTL0)

Module Base + 0x0000 — **Access: User read/write** (297)

	7	6	5	4	3	2	1	0
R	RXFRM	RXACT	CSWAI	SYNCH	TIME	WUPE	SLPRQ	INITRQ
W								
Reset:	0	0	0	0	0	0	0	1

= Unimplemented

Notes:

297. Read: Anytime
Write: Anytime when out of initialization mode; exceptions are read-only RXACT and SYNCH, RXFRM (which is set by the module only), and INITRQ (which is also writable in initialization mode)

Note

The CANCTL0 register, except WUPE, INITRQ, and SLPRQ, is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable again as soon as the initialization mode is exited (INITRQ = 0 and INITAK = 0).

Table 460. CANCTL0 register field descriptions

Field	Description
7 RXFRM	**Received Frame Flag** — This bit is read and clear only. It is set when a receiver has received a valid message correctly, independently of the filter configuration. After it is set, it remains set until cleared by software or reset. Clearing is done by writing a 1. Writing a 0 is ignored. This bit is not valid in loopback mode. 0 No valid message was received since last clearing this flag 1 A valid message was received since last clearing of this flag
6 RXACT	**Receiver Active Status** — This read-only flag indicates the MSCAN is receiving a message (298). The flag is controlled by the receiver front end. This bit is not valid in loopback mode. 0 MSCAN is transmitting or idle 1 MSCAN is receiving a message (including when arbitration is lost)
5 CSWAI (299)	**CAN Stops in Wait Mode** — Enabling this bit allows for lower power consumption in wait mode by disabling all the clocks at the CPU bus interface to the MSCAN module. 0 The module is not affected during wait mode 1 The module ceases to be clocked during wait mode
4 SYNCH	**Synchronized Status** — This read-only flag indicates whether the MSCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the MSCAN. 0 MSCAN is not synchronized to the CAN bus 1 MSCAN is synchronized to the CAN bus
3 TIME	**Timer Enable** — This bit activates an internal 16-bit wide free running timer which is clocked by the bit clock rate. If the timer is enabled, a 16-bit time stamp will be assigned to each transmitted/received message within the active TX/RX buffer. Right after the EOF of a valid message on the CAN bus, the time stamp is written to the highest bytes (0x000E, 0x000F) in the appropriate buffer (see Programmer's model of message storage"). The internal timer is reset (all bits set to 0) when disabled. This bit is held low in initialization mode. 0 Disable internal MSCAN timer 1 Enable internal MSCAN timer
2 WUPE (300)	**Wake-Up Enable** — This configuration bit allows the MSCAN to restart from sleep mode or from power down mode (entered from sleep) when traffic on CAN is detected (see MSCAN sleep mode"). This bit must be configured before sleep mode entry for the selected function to take effect. 0 Wake-up disabled — The MSCAN ignores traffic on CAN 1 Wake-up enabled — The MSCAN is able to restart

Table 460. CANCTL0 register field descriptions (continued)

Field	Description
1 SLPRQ (301)	**Sleep Mode Request** — This bit requests the MSCAN to enter sleep mode, which is an internal power saving mode (see MSCAN sleep mode"). The sleep mode request is serviced when the CAN bus is idle, i.e., the module is not receiving a message and all transmit buffers are empty. The module indicates entry to sleep mode by setting SLPAK = 1 (see MSCAN control register 1 (CANCTL1)"). SLPRQ cannot be set while the WUPIF flag is set (see MSCAN receiver flag register (CANRFLG)"). Sleep mode will be active until SLPRQ is cleared by the CPU or, depending on the setting of WUPE, the MSCAN detects activity on the CAN bus and clears SLPRQ itself. 0 Running — The MSCAN functions normally 1 Sleep mode request — The MSCAN enters sleep mode when CAN bus idle
0 INITRQ (302),(303)	**Initialization Mode Request** — When this bit is set by the CPU, the MSCAN skips to initialization mode (see MSCAN initialization mode"). Any ongoing transmission or reception is aborted and synchronization to the CAN bus is lost. The module indicates entry to initialization mode by setting INITAK = 1 (MSCAN control register 1 (CANCTL1)"). The following registers enter their hard reset state and restore their default values: CANCTL0 (304), CANRFLG (305), CANRIER (306), CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0-7, and CANIDMR0-7 can only be written by the CPU when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1). The values of the error counters are not affected by initialization mode. When this bit is cleared by the CPU, the MSCAN restarts and then tries to synchronize to the CAN bus. If the MSCAN is not in bus-off state, it synchronizes after 11 consecutive recessive bits on the CAN bus; if the MSCAN is in bus-off state, it continues to wait for 128 occurrences of 11 consecutive recessive bits. Writing to other bits in CANCTL0, CANRFLG, CANRIER, CANTFLG, or CANTIER must be done only after initialization mode is exited, which is INITRQ = 0 and INITAK = 0. 0 Normal operation 1 MSCAN in initialization mode

Notes:

298.See the Bosch CAN 2.0A/B specification for a detailed definition of transmitter and receiver states.

299.In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the CPU enters wait (CSWAI = 1) or stop mode (see Section 6.13.4.5.2, "Operation in wait mode"" and Section 6.13.4.5.3, "Operation in stop mode"").

300.The CPU has to make sure that the WUPE register and the WUPIE wake-up interrupt enable register (see Section 6.13.3.2.6, "MSCAN receiver interrupt enable register (CANRIER)") is enabled, if the recovery mechanism from stop or wait is required.

301.The CPU cannot clear SLPRQ before the MSCAN has entered sleep mode (SLPRQ = 1 and SLPAK = 1).

302.The CPU cannot clear INITRQ before the MSCAN has entered initialization mode (INITRQ = 1 and INITAK = 1).

303.In order to protect from accidentally violating the CAN protocol, TXCAN is immediately forced to a recessive state when the initialization mode is requested by the CPU. Thus, the recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before requesting initialization mode.

304.Not including WUPE, INITRQ, and SLPRQ.

305.TSTAT1 and TSTAT0 are not affected by initialization mode.

306.RSTAT1 and RSTAT0 are not affected by initialization mode.

6.13.3.2.2 MSCAN control register 1 (CANCTL1)

The CANCTL1 register provides various control bits and handshake status information of the MSCAN module as described in Table 3.

Table 3. MSCAN control register 1 (CANCTL1)

Module Base + 0x0001 — **Access: User read/write (307)**

	7	6	5	4	3	2	1	0
R W	CANE	CLKSRC	LOOPB	LISTEN	BORM	WUPM	SLPAK	INITAK
Reset:	0	0	0	1	0	0	0	1
	= Unimplemented							

Notes:

307.Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1), except CANE which is write once in normal and anytime in special system operation modes when the MSCAN is in initialization mode (INITRQ = 1 and INITAK = 1)

Table 461. CANCTL1 register field descriptions

Field	Description
7 CANE	**MSCAN Enable** 0 MSCAN module is disabled 1 MSCAN module is enabled
6 CLKSRC	**MSCAN Clock Source** — This bit defines the clock source for the MSCAN module (only for systems with a clock generation module; Clock system," and MSCAN clocking scheme,"). 0 MSCAN clock source is the oscillator clock 1 MSCAN clock source is the bus clock
5 LOOPB	**Loopback Self Test Mode** — When this bit is set, the MSCAN performs an internal loopback which can be used for self test operation. The bit stream output of the transmitter is fed back to the receiver internally. The RXCAN input is ignored and the TXCAN output goes to the recessive state (logic 1). The MSCAN behaves as it does normally when transmitting and treats its own transmitted message as a message received from a remote node. In this state, the MSCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated. 0 Loopback self test disabled 1 Loopback self test enabled
4 LISTEN	**Listen Only Mode** — This bit configures the MSCAN as a CAN bus monitor. When LISTEN is set, all valid CAN messages with matching ID are received, but no acknowledgement or error frames are sent out (see Listen-only mode"). In addition, the error counters are frozen. Listen only mode supports applications which require "hot plugging" or throughput analysis. The MSCAN is unable to transmit any messages when listen only mode is active. 0 Normal operation 1 Listen only mode activated
3 BORM	**Bus-Off Recovery Mode** — This bit configures the bus-off state recovery mode of the MSCAN. Refer to Bus-off recovery," for details. 0 Automatic bus-off recovery (see Bosch CAN 2.0A/B protocol specification) 1 Bus-off recovery upon user request
2 WUPM	**Wake-up Mode** — If WUPE in CANCTL0 is enabled, this bit defines whether the integrated low-pass filter is applied to protect the MSCAN from spurious wake-up (see MSCAN sleep mode"). 0 MSCAN wakes up on any dominant level on the CAN bus 1 MSCAN wakes up only in case of a dominant pulse on the CAN bus that has a length of t_{WUP}
1 SLPAK	**Sleep Mode Acknowledge** — This flag indicates whether the MSCAN module has entered Sleep mode (see MSCAN sleep mode"). It is used as a handshake flag for the SLPRQ Sleep mode request. Sleep mode is active when SLPRQ = 1 and SLPAK = 1. Depending on the setting of WUPE, the MSCAN will clear the flag if it detects activity on the CAN bus while in Sleep mode. 0 Running — The MSCAN operates normally 1 Sleep mode active — The MSCAN has entered Sleep mode
0 INITAK	**Initialization Mode Acknowledge** — This flag indicates whether the MSCAN module is in initialization mode (see MSCAN initialization mode"). It is used as a handshake flag for the INITRQ initialization mode request. Initialization mode is active when INITRQ = 1 and INITAK = 1. The registers CANCTL1, CANBTR0, CANBTR1, CANIDAC, CANIDAR0–CANIDAR7, and CANIDMR0–CANIDMR7 can be written only by the CPU when the MSCAN is in Initialization mode. 0 Running — The MSCAN operates normally 1 Initialization mode active — The MSCAN has entered initialization mode

6.13.3.2.3 MSCAN bus timing register 0 (CANBTR0)

The CANBTR0 register configures various CAN bus timing parameters of the MSCAN module.

Table 462. MSCAN bus timing register 0 (CANBTR0)

Module Base + 0x0002 — **Access: User read/write** (308)

	7	6	5	4	3	2	1	0
R W	SJW1	SJW0	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0
Reset:	0	0	0	0	0	0	0	0

Notes:

308.Read: Anytime
Write: Anytime in Initialization mode (INITRQ = 1 and INITAK = 1)

Table 463. CANBTR0 register field descriptions

Field	Description
7-6 SJW[1:0]	**Synchronization Jump Width** — The synchronization jump width defines the maximum number of time quanta (Tq) clock cycles a bit can be shortened or lengthened to achieve resynchronization to data transitions on the CAN bus (see Table 464).
5-0 BRP[5:0]	**Baud Rate Prescaler** — These bits determine the time quanta (Tq) clock which is used to build up the bit timing (see Table 465).

Table 464. Synchronization jump width

SJW1	SJW0	Synchronization Jump Width
0	0	1 Tq clock cycle
0	1	2 Tq clock cycles
1	0	3 Tq clock cycles
1	1	4 Tq clock cycles

Table 465. Baud rate prescaler

BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	Prescaler value (P)
0	0	0	0	0	0	1
0	0	0	0	0	1	2
0	0	0	0	1	0	3
0	0	0	0	1	1	4
:	:	:	:	:	:	:
1	1	1	1	1	1	64

6.13.3.2.4 MSCAN bus timing register 1 (CANBTR1)

The CANBTR1 register configures various CAN bus timing parameters of the MSCAN module.

Table 466. MSCAN bus timing register 1 (CANBTR1)

Module Base + 0x0003 — **Access: User read/write(309)**

	7	6	5	4	3	2	1	0
R W	SAMP	TSEG22	TSEG21	TSEG20	TSEG13	TSEG12	TSEG11	TSEG10
Reset:	0	0	0	0	0	0	0	0

Notes:

309.Read: Anytime
Write: Anytime in Initialization mode (INITRQ = 1 and INITAK = 1)

Table 467. CANBTR1 register field descriptions

Field	Description
7 SAMP	**Sampling** — This bit determines the number of CAN bus samples taken per bit time. 0One sample per bit. 1Three samples per bit (310). If SAMP = 0, the resulting bit value is equal to the value of the single bit positioned at the sample point. If SAMP = 1, the resulting bit value is determined by using majority rule on the three total samples. For higher bit rates, it is recommended that only one sample is taken per bit time (SAMP = 0).
6-4 TSEG2[2:0]	**Time Segment 2** — Time segments within the bit time fix the number of clock cycles per bit time and the location of the sample point (see Figure 95). Time segment 2 (TSEG2) values are programmable as shown in Table 468.

Table 467. CANBTR1 register field descriptions (continued)

Field	Description
3-0 TSEG1[3:0]	**Time Segment 1** — Time segments within the bit time fix the number of clock cycles per bit time and the location of the sample point (see Figure 95). Time segment 1 (TSEG1) values are programmable as shown in Table 469.

Notes:
310.In this case, PHASE_SEG1 must be at least two times quanta (Tq).

Table 468. Time segment 2 values

TSEG22	TSEG21	TSEG20	Time Segment 2
0	0	0	1 Tq clock cycle(311)
0	0	1	2 Tq clock cycles
:	:	:	:
1	1	0	7 Tq clock cycles
1	1	1	8 Tq clock cycles

Notes:
311.This setting is not valid. Refer to Table 526 for valid settings.

Table 469. Time segment 1 values

TSEG13	TSEG12	TSEG11	TSEG10	Time segment 1
0	0	0	0	1 Tq clock cycle (312)
0	0	0	1	2 Tq clock cycles (312)
0	0	1	0	3 Tq clock cycles (312)
0	0	1	1	4 Tq clock cycles
:	:	:	:	:
1	1	1	0	15 Tq clock cycles
1	1	1	1	16 Tq clock cycles

Notes:
312.This setting is not valid. Refer to Table 526 for valid settings.

The bit time is determined by the oscillator frequency, the baud rate prescaler, and the number of time quanta (Tq) clock cycles per bit (as shown in Table 468 and Table 469).

$$Bit\ \ Time = \frac{(Prescaler\ \ value)}{f_{CANCLK}} \times (1 + TimeSegment1 + TimeSegment2)$$

6.13.3.2.5 MSCAN receiver flag register (CANRFLG)

A flag can be cleared only by software (writing a 1 to the corresponding bit position) when the condition which caused the setting is no longer valid. Every flag has an associated interrupt enable bit in the CANRIER register.

Table 470. MSCAN receiver flag register (CANRFLG)

Module Base + 0x0004						Access: User read/write[313]	
7	6	5	4	3	2	1	0
R W	WUPIF	CSCIF	RSTAT1	RSTAT0	TSTAT1	TSTAT0	OVRIF	RXF
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:
313.Read: Anytime
Write: Anytime when not in initialization mode, except RSTAT[1:0] and TSTAT[1:0] flags which are read-only; write of 1 clears flag; write of 0 is ignored

Note

The CANRFLG register is held in the reset state [314] when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable again as soon as the initialization mode is exited (INITRQ = 0 and INITAK = 0).

Notes:
314.The RSTAT[1:0], TSTAT[1:0] bits are not affected by initialization mode.

Table 471. CANRFLG register field descriptions

Field	Description
7 WUPIF	**Wake-up Interrupt Flag** — If the MSCAN detects CAN bus activity while in sleep mode (see MSCAN sleep mode,") and WUPE = 1 in CANTCTL0 (see MSCAN control register 0 (CANCTL0)"), the module will set WUPIF. If not masked, a wake-up interrupt is pending while this flag is set. 0 No wake-up activity observed while in sleep mode 1 MSCAN detected activity on the CAN bus and requested wake-up
6 CSCIF	**CAN Status Change Interrupt Flag** — This flag is set when the MSCAN changes its current CAN bus status due to the actual value of the transmit error counter (TEC) and the receive error counter (REC). An additional 4-bit (RSTAT[1:0], TSTAT[1:0]) status register, which is split into separate sections for TEC/REC, informs the system on the actual CAN bus status (see MSCAN receiver interrupt enable register (CANRIER)"). If not masked, an error interrupt is pending while this flag is set. CSCIF provides a blocking interrupt. That guarantees that the receiver/transmitter status bits (RSTAT/TSTAT) are only updated when no CAN status change interrupt is pending. If the TECs/RECs change their current value after the CSCIF is asserted, which would cause an additional state change in the RSTAT/TSTAT bits, these bits keep their status until the current CSCIF interrupt is cleared again. 0 No change in CAN bus status occurred since last interrupt 1 MSCAN changed current CAN bus status
5-4 RSTAT[1:0]	**Receiver Status Bits** — The values of the error counters control the actual CAN bus status of the MSCAN. As soon as the status change interrupt flag (CSCIF) is set, these bits indicate the appropriate receiver related CAN bus status of the MSCAN. The coding for the bits RSTAT1, RSTAT0 is: 00 RxOK: 0 ≤ receive error counter ≤ 96 01 RxWRN: 96 < receive error counter ≤ 127 10 RxERR:127 < receive error counter 11 Bus-off[315]:transmit error counter > 255
3-2 TSTAT[1:0]	**Transmitter Status Bits** — The values of the error counters control the actual CAN bus status of the MSCAN. As soon as the status change interrupt flag (CSCIF) is set, these bits indicate the appropriate transmitter related CAN bus status of the MSCAN. The coding for the bits TSTAT1, TSTAT0 is: 00 TxOK: 0 ≤ transmit error counter ≤ 96 01 TxWRN: 96 < transmit error counter ≤ 127 10 TxERR:127 < transmit error counter ≤ 255 11 Bus-Off: transmit error counter > 255
1 OVRIF	**Overrun Interrupt Flag** — This flag is set when a data overrun condition occurs. If not masked, an error interrupt is pending while this flag is set. 0 No data overrun condition 1 A data overrun detected

Table 471. CANRFLG register field descriptions (continued)

Field	Description
0 RXF[316]	**Receive Buffer Full Flag** — RXF is set by the MSCAN when a new message is shifted in the receiver FIFO. This flag indicates whether the shifted buffer is loaded with a correctly received message (matching identifier, matching cyclic redundancy code (CRC) and no other errors detected). After the CPU has read that message from the RxFG buffer in the receiver FIFO, the RXF flag must be cleared to release the buffer. A set RXF flag prohibits the shifting of the next FIFO entry into the foreground buffer (RxFG). If not masked, a receive interrupt is pending while this flag is set. 0 No new message available within the RxFG 1 The receiver FIFO is not empty. A new message is available in the RxFG

Notes:

315. Redundant Information for the most critical CAN bus status which is “bus-off”. This only occurs if the Tx error counter exceeds a number of 255 errors. Bus-off affects the receiver state. As soon as the transmitter leaves its bus-off state the receiver state skips to RxOK too. Refer also to TSTAT[1:0] coding in this register.

316. To ensure data integrity, do not read the receive buffer registers while the RXF flag is cleared. For MCUs with dual CPUs, reading the receive buffer registers while the RXF flag is cleared may result in a CPU fault condition.

6.13.3.2.6 MSCAN receiver interrupt enable register (CANRIER)

This register contains the interrupt enable bits for the interrupt flags described in the CANRFLG register.

Table 472. MSCAN receiver interrupt enable register (CANRIER)

Module Base + 0x0005 — **Access: User read/write[317]**

	7	6	5	4	3	2	1	0
R W	WUPIE	CSCIE	RSTATE1	RSTATE0	TSTATE1	TSTATE0	OVRIE	RXFIE
Reset:	0	0	0	0	0	0	0	0

Notes:

317. Read: Anytime
Write: Anytime when not in initialization mode.

Note

The CANRIER register is held in the reset state when the initialization mode is active (INITRQ=1 and INITAK=1). This register is writable when not in initialization mode (INITRQ=0 and INITAK=0).

The RSTATE[1:0], TSTATE[1:0] bits are not affected by initialization mode.

Table 473. CANRIER register field descriptions

Field	Description
7 WUPIE[318]	**Wake-up Interrupt Enable** 0 No interrupt request is generated from this event. 1 A wake-up event causes a Wake-up interrupt request.
6 CSCIE	**CAN Status Change Interrupt Enable** 0 No interrupt request is generated from this event. 1 A CAN Status Change event causes an error interrupt request.
5-4 RSTATE[1:0]	**Receiver Status Change Enable** — These RSTAT enable bits control the sensitivity level in which receiver state changes are causing CSCIF interrupts. Independent of the chosen sensitivity level the RSTAT flags continue to indicate the actual receiver state and are only updated if no CSCIF interrupt is pending. 00 Do not generate any CSCIF interrupt caused by receiver state changes. 01 Generate CSCIF interrupt only if the receiver enters or leaves “bus-off” state. Discard other receiver state changes for generating CSCIF interrupt. 10 Generate CSCIF interrupt only if the receiver enters or leaves “RxErr” or “bus-off”[319] state. Discard other receiver state changes for generating CSCIF interrupt. 11 Generate CSCIF interrupt on all state changes.

Table 473. CANRIER register field descriptions (continued)

Field	Description
3-2 TSTATE[1:0]	**Transmitter Status Change Enable** — These TSTAT enable bits control the sensitivity level in which transmitter state changes are causing CSCIF interrupts. Independent of the chosen sensitivity level, the TSTAT flags continue to indicate the actual transmitter state and are only updated if no CSCIF interrupt is pending. 00 Do not generate any CSCIF interrupt caused by transmitter state changes. 01 Generate CSCIF interrupt only if the transmitter enters or leaves “bus-off” state. Discard other transmitter state changes for generating CSCIF interrupt. 10 Generate CSCIF interrupt only if the transmitter enters or leaves “TxErr” or “bus-off” state. Discard other transmitter state changes for generating CSCIF interrupt. 11 Generate CSCIF interrupt on all state changes.
1 OVRIE	**Overrun Interrupt Enable** 0 No interrupt request is generated from this event. 1 An overrun event causes an error interrupt request.
0 RXFIE	**Receiver Full Interrupt Enable** 0 No interrupt request is generated from this event. 1 A receive buffer full (successful message reception) event causes a receiver interrupt request.

Notes:

318.WUPIE and WUPE (see MSCAN control register 0 (CANCTL0)”) must both be enabled if the recovery mechanism from stop or wait is required.

319.Bus-off state is only defined for transmitters by the CAN standard (see Bosch CAN 2.0A/B protocol specification). Because the only possible state change for the transmitter from bus-off to TxOK also forces the receiver to skip its current state to RxOK, the coding of the RXSTAT[1:0] flags define an additional bus-off state for the receiver (see MSCAN receiver flag register (CANRFLG)”).

6.13.3.2.7 MSCAN transmitter flag register (CANTFLG)

The transmit buffer empty flags each have an associated interrupt enable bit in the CANTIER register.

Table 474. MSCAN transmitter flag register (CANTFLG)

Module Base + 0x0006 — **Access: User read/write[320]**

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TXE2	TXE1	TXE0
W								
Reset:	0	0	0	0	0	1	1	1

= Unimplemented

Notes:

320.Read: Anytime
Write: Anytime when not in initialization mode; write of 1 clears flag, write of 0 is ignored

Note

The CANTFLG register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 475. CANTFLG register field descriptions

Field	Description
2-0 TXE[2:0]	**Transmitter Buffer Empty** — This flag indicates that the associated transmit message buffer is empty, and thus not scheduled for transmission. The CPU must clear the flag after a message is set up in the transmit buffer and is due for transmission. The MSCAN sets the flag after the message is sent successfully. The flag is also set by the MSCAN when the transmission request is successfully aborted due to a pending abort request (see MSCAN transmitter message abort request register (CANTARQ)"). If not masked, a transmit interrupt is pending while this flag is set. Clearing a TXEx flag also clears the corresponding ABTAKx (see MSCAN transmitter message abort acknowledge register (CANTAAK)"). When a TXEx flag is set, the corresponding ABTRQx bit is cleared (see MSCAN transmitter message abort request register (CANTARQ)"). When listen-mode is active (see MSCAN control register 1 (CANCTL1)") the TXEx flags cannot be cleared and no transmission is started. Read and write accesses to the transmit buffer will be blocked, if the corresponding TXEx bit is cleared (TXEx = 0) and the buffer is scheduled for transmission. 0 The associated message buffer is full (loaded with a message due for transmission) 1 The associated message buffer is empty (not scheduled)

6.13.3.2.8 MSCAN transmitter interrupt enable register (CANTIER)

This register contains the interrupt enable bits for the transmit buffer empty interrupt flags.

Table 476. MSCAN transmitter interrupt enable register (CANTIER)

Module Base + 0x0007 — **Access: User read/write (321)**

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TXEIE2	TXEIE1	TXEIE0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:
321.Read: Anytime
Write: Anytime when not in initialization mode

Note

The CANTIER register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 477. CANTIER register field descriptions

Field	Description
2-0 TXEIE[2:0]	**Transmitter Empty Interrupt Enable** 0 No interrupt request is generated from this event. 1 A transmitter empty (transmit buffer available for transmission) event causes a transmitter empty interrupt request.

6.13.3.2.9 MSCAN transmitter message abort request register (CANTARQ)

The CANTARQ register allows abort request of queued messages as described in Table 478.

Table 478. MSCAN transmitter message abort request register (CANTARQ)

Module Base + 0x0008						Access: User read/write (322)	
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ABTRQ2	ABTRQ1	ABTRQ0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:
322.Read: Anytime
Write: Anytime when not in initialization mode

Note

The CANTARQ register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 479. CANTARQ register field descriptions

Field	Description
2-0 ABTRQ[2:0]	**Abort Request** — The CPU sets the ABTRQx bit to request that a scheduled message buffer (TXEx = 0) be aborted. The MSCAN grants the request if the message has not already started transmission, or if the transmission is not successful (lost arbitration or error). When a message is aborted, the associated TXE (see MSCAN transmitter flag register (CANTFLG)") and abort acknowledge flags (ABTAK, see MSCAN transmitter message abort acknowledge register (CANTAAK)") are set and a transmit interrupt occurs if enabled. The CPU cannot reset ABTRQx. ABTRQx is reset whenever the associated TXE flag is set. 0 No abort request 1 Abort request pending

6.13.3.2.10 MSCAN transmitter message abort acknowledge register (CANTAAK)

The CANTAAK register indicates the successful abort of a queued message, if requested by the appropriate bits in the CANTARQ register.

Table 480. MSCAN transmitter message abort acknowledge register (CANTAAK)

Module Base + 0x0009						Access: User read/write(323)	
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	ABTAK2	ABTAK1	ABTAK0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:
323.Read: Anytime
Write: Unimplemented

Note

The CANTAAK register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK = 1).

Table 481. CANTAAK register field descriptions

Field	Description
2-0 ABTAK[2:0]	**Abort Acknowledge** — This flag acknowledges that a message was aborted due to a pending abort request from the CPU. After a particular message buffer is flagged empty, this flag can be used by the application software to identify whether the message was aborted successfully or was sent anyway. The ABTAKx flag is cleared whenever the corresponding TXE flag is cleared. 0 The message was not aborted. 1 The message was aborted.

6.13.3.2.11 MSCAN transmit buffer selection register (CANTBSEL)

The CANTBSEL register allows the selection of the actual transmit message buffer, which then will be accessible in the CANTXFG register space.

Table 482. MSCAN transmit buffer selection register (CANTBSEL)

Module Base + 0x000A — **Access: User read/write** (324)

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	TX2	TX1	TX0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:
324. Read: Find the lowest ordered bit set to 1, all other bits will be read as 0
Write: Anytime when not in initialization mode

Note

The CANTBSEL register is held in the reset state when the initialization mode is active (INITRQ = 1 and INITAK=1). This register is writable when not in initialization mode (INITRQ = 0 and INITAK = 0).

Table 483. CANTBSEL register field descriptions

Field	Description
2-0 TX[2:0]	**Transmit Buffer Select** — The lowest numbered bit places the respective transmit buffer in the CANTXFG register space (e.g., TX1 = 1 and TX0 = 1 selects transmit buffer TX0; TX1 = 1 and TX0 = 0 selects transmit buffer TX1). Read and write accesses to the selected transmit buffer will be blocked, if the corresponding TXEx bit is cleared and the buffer is scheduled for transmission (see MSCAN transmitter flag register (CANTFLG)"). 0 The associated message buffer is deselected 1 The associated message buffer is selected, if lowest numbered bit

The following gives a short programming example of the usage of the CANTBSEL register:

To get the next available transmit buffer, application software must read the CANTFLG register and write this value back into the CANTBSEL register. In this example, Tx buffers TX1 and TX2 are available. The value read from CANTFLG is therefore 0b0000_0110. When writing this value back to CANTBSEL, the Tx buffer TX1 is selected in the CANTXFG because the lowest numbered bit set to 1 is at bit position 1. Reading back this value out of CANTBSEL results in 0b0000_0010, because only the lowest numbered bit position set to 1 is presented. This mechanism eases the application software's selection of the next available Tx buffer.

- LDAA CANTFLG; value read is 0b0000_0110
- STAA CANTBSEL; value written is 0b0000_0110
- LDAA CANTBSEL; value read is 0b0000_0010

If all transmit message buffers are deselected, no accesses are allowed to the CANTXFG registers.

6.13.3.2.12 MSCAN identifier acceptance control register (CANIDAC)

The CANIDAC register is used for identifier acceptance control as described below.

Table 484. MSCAN identifier acceptance control register (CANIDAC)

Module Base + 0x000B							Access: User read/write[325]
	7	6	5	4	3	2	1	0
R	0	0	IDAM1	IDAM0	0	IDHIT2	IDHIT1	IDHIT0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:

325.Read: Anytime

Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1), except bits IDHITx, which are read-only

Table 485. CANIDAC register field descriptions

Field	Description
5-4 IDAM[1:0]	**Identifier Acceptance Mode** — The CPU sets these flags to define the identifier acceptance filter organization (see Identifier acceptance filter"). Figure 486 summarizes the different settings. In filter closed mode, no message is accepted such that the foreground buffer is never reloaded.
2-0 IDHIT[2:0]	**Identifier Acceptance Hit Indicator** — The MSCAN sets these flags to indicate an identifier acceptance hit (see Identifier acceptance filter"). Table 487 summarizes the different settings.

Table 486. Identifier acceptance mode settings

IDAM1	IDAM0	Identifier Acceptance Mode
0	0	Two 32-bit acceptance filters
0	1	Four 16-bit acceptance filters
1	0	Eight 8-bit acceptance filters
1	1	Filter closed

Table 487. Identifier acceptance hit indication

IDHIT2	IDHIT1	IDHIT0	Identifier Acceptance Hit
0	0	0	Filter 0 hit
0	0	1	Filter 1 hit
0	1	0	Filter 2 hit
0	1	1	Filter 3 hit
1	0	0	Filter 4 hit
1	0	1	Filter 5 hit
1	1	0	Filter 6 hit
1	1	1	Filter 7 hit

The IDHITx indicators are always related to the message in the foreground buffer (RxFG). When a message gets shifted into the foreground buffer of the receiver FIFO the indicators are updated as well.

6.13.3.2.13 MSCAN reserved register

This register is reserved for factory testing of the MSCAN module and is not available in normal system operating modes.

Table 488. MSCAN Reserved Register

Module Base + 0x000C to Module Base + 0x000D							Access: User read/write [326]
7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:
326.Read: Always reads zero in normal system operation modes
Write: Unimplemented in normal system operation modes

Note

Writing to this register when in special system operating modes can alter the MSCAN functionality.

6.13.3.2.14 MSCAN miscellaneous register (CANMISC)

This register provides additional features.

Table 489. MSCAN miscellaneous register (CANMISC)

Module Base + 0x000D							Access: User read/write [327]
7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	BOHOLD
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:
327.Read: Anytime
Write: Anytime; write of ‘1’ clears flag; write of ‘0’ ignored

Table 490. CANMISC register field descriptions

Field	Description
0 BOHOLD	**Bus-off State Hold Until User Request** — If BORM is set in MSCAN control register 1 (CANCTL1), this bit indicates whether the module has entered the bus-off state. Clearing this bit requests the recovery from bus-off. Refer to Bus-off recovery," for details. 0 Module is not bus-off or recovery has been requested by user in bus-off state 1 Module is bus-off and holds this state until user request

6.13.3.2.15 MSCAN receive error counter (CANRXERR)

This register reflects the status of the MSCAN receive error counter.

Table 491. MSCAN receive error counter (CANRXERR)

Module Base + 0x000E							Access: User read/write[328]
7	6	5	4	3	2	1	0
R	RXERR7	RXERR6	RXERR5	RXERR4	RXERR3	RXERR2	RXERR1	RXERR0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:
328.Read: Only when in sleep mode (SLPRQ = 1 and SLPAK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
Write: Unimplemented

Note

Reading this register when in any other mode other than sleep or initialization mode may return an incorrect value. For MCUs with dual CPUs, this may result in a CPU fault condition.

Writing to this register when in special modes can alter the MSCAN functionality.

6.13.3.2.16 MSCAN transmit error counter (CANTXERR)

This register reflects the status of the MSCAN transmit error counter.

Table 492. MSCAN transmit error counter (CANTXERR)

Module Base + 0x000F							Access: User read/write (329)
7	6	5	4	3	2	1	0
R	TXERR7	TXERR6	TXERR5	TXERR4	TXERR3	TXERR2	TXERR1	TXERR0
W								
Reset:	0	0	0	0	0	0	0	0

= Unimplemented

Notes:

329.Read: Only when in sleep mode (SLPRQ = 1 and SLPAK = 1) or initialization mode (INITRQ = 1 and INITAK = 1)
Write: Unimplemented

Note

Reading this register when in any other mode other than sleep or initialization mode, may return an incorrect value. For MCUs with dual CPUs, this may result in a CPU fault condition.

Writing to this register when in special modes can alter the MSCAN functionality.

6.13.3.2.17 MSCAN identifier acceptance registers (CANIDAR0-7)

On reception, each message is written into the background receive buffer. The CPU is only signalled to read the message if it passes the criteria in the identifier acceptance and identifier mask registers (accepted); otherwise, the message is overwritten by the next message (dropped).

The acceptance registers of the MSCAN are applied on the IDR0–IDR3 registers (see Identifier registers (IDR0–IDR3)") of incoming messages in a bit by bit manner (see Identifier acceptance filter").

For extended identifiers, all four acceptance and mask registers are applied. For standard identifiers, only the first two (CANIDAR0/1, CANIDMR0/1) are applied.

Table 493. MSCAN identifier acceptance registers (first bank) — CANIDAR0–CANIDAR3

Module Base + 0x0010 to Module Base + 0x0013							Access: User read/write (330)
7	6	5	4	3	2	1	0
R W	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
Reset	0	0	0	0	0	0	0	0

Notes:

330.Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 494. CANIDAR0–CANIDAR3 register field descriptions

Field	Description
7-0 AC[7:0]	**Acceptance Code Bits** — AC[7:0] comprise a user-defined sequence of bits with which the corresponding bits of the related identifier register (IDRn) of the receive message buffer are compared. The result of this comparison is then masked with the corresponding identifier mask register.

Table 495. MSCAN identifier acceptance registers (second bank) — CANIDAR4–CANIDAR7

Module Base + 0x0018 to Module Base + 0x001B — Access: User read/write [331]

	7	6	5	4	3	2	1	0
R W	AC7	AC6	AC5	AC4	AC3	AC2	AC1	AC0
Reset	0	0	0	0	0	0	0	0

Notes:
331.Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 496. CANIDAR4–CANIDAR7 register field descriptions

Field	Description
7-0 AC[7:0]	**Acceptance Code Bits** — AC[7:0] comprise a user-defined sequence of bits with which the corresponding bits of the related identifier register (IDRn) of the receive message buffer are compared. The result of this comparison is then masked with the corresponding identifier mask register.

6.13.3.2.18 MSCAN identifier mask registers (CANIDMR0–CANIDMR7)

The identifier mask register specifies which of the corresponding bits in the identifier acceptance register are relevant for acceptance filtering. To receive standard identifiers in 32 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1 and CANIDMR5 to “don’t care.” To receive standard identifiers in 16 bit filter mode, it is required to program the last three bits (AM[2:0]) in the mask registers CANIDMR1, CANIDMR3, CANIDMR5, and CANIDMR7 to “don’t care.”

Table 497. MSCAN identifier mask registers (first bank) — CANIDMR0–CANIDMR3

Module Base + 0x0014 to Module Base + 0x0017 — Access: User read/write [332]

	7	6	5	4	3	2	1	0
R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset	0	0	0	0	0	0	0	0

Notes:
332.Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 498. CANIDMR0–CANIDMR3 register field descriptions

Field	Description
7-0 AM[7:0]	**Acceptance Mask Bits** — If a particular bit in this register is cleared, this indicates that the corresponding bit in the identifier acceptance register must be the same as its identifier bit before a match is detected. The message is accepted if all such bits match. If a bit is set, it indicates that the state of the corresponding bit in the identifier acceptance register does not affect whether or not the message is accepted. 0 Match corresponding acceptance code register and identifier bits 1 Ignore corresponding acceptance code register bit

Table 499. MSCAN identifier mask registers (second bank) — CANIDMR4–CANIDMR7

Module Base + 0x001C to Module Base + 0x001F — Access: User read/write [333]

	7	6	5	4	3	2	1	0
R W	AM7	AM6	AM5	AM4	AM3	AM2	AM1	AM0
Reset	0	0	0	0	0	0	0	0

Notes:
333.Read: Anytime
Write: Anytime in initialization mode (INITRQ = 1 and INITAK = 1)

Table 500. CANIDMR4–CANIDMR7 register field descriptions

Field	Description
7-0 AM[7:0]	**Acceptance Mask Bits** — If a particular bit in this register is cleared, this indicates that the corresponding bit in the identifier acceptance register must be the same as its identifier bit before a match is detected. The message is accepted if all such bits match. If a bit is set, it indicates that the state of the corresponding bit in the identifier acceptance register does not affect whether or not the message is accepted. 0 Match corresponding acceptance code register and identifier bits 1 Ignore corresponding acceptance code register bit

6.13.3.3 Programmer's model of message storage

The following section details the organization of the receive and transmit message buffers and the associated control registers.

To simplify the programmer interface, the receive and transmit message buffers have the same outline. Each message buffer allocates 16 bytes in the memory map containing a 13 byte data structure.

An additional transmit buffer priority register (TBPR) is defined for the transmit buffers. Within the last two bytes of this memory map, the MSCAN stores a special 16-bit time stamp, which is sampled from an internal timer after successful transmission or reception of a message. This feature is only available for transmit and receiver buffers, if the TIME bit is set (see MSCAN control register 0 (CANCTL0)").

The time stamp register is written by the MSCAN. The CPU can only read these registers.

Table 501. Message buffer organization

Offset Address	Register	Access
0x00X0	Identifier Register 0	R/W
0x00X1	Identifier Register 1	R/W
0x00X2	Identifier Register 2	R/W
0x00X3	Identifier Register 3	R/W
0x00X4	Data Segment Register 0	R/W
0x00X5	Data Segment Register 1	R/W
0x00X6	Data Segment Register 2	R/W
0x00X7	Data Segment Register 3	R/W
0x00X8	Data Segment Register 4	R/W
0x00X9	Data Segment Register 5	R/W
0x00XA	Data Segment Register 6	R/W
0x00XB	Data Segment Register 7	R/W
0x00XC	Data Length Register	R/W
0x00XD	Transmit Buffer Priority Register [334]	R/W
0x00XE	Time Stamp Register (High Byte)	R
0x00XF	Time Stamp Register (Low Byte)	R

Notes:
334.Not applicable for receive buffers

Figure 502 shows the common 13-byte data structure of receive and transmit buffers for extended identifiers. The mapping of standard identifiers into the IDR registers is shown in Figure 503.

All bits of the receive and transmit buffers are ‘x’ out of reset because of RAM-based implementation [(335)]. All reserved or unused bits of the receive and transmit buffers always read ‘x’.

Notes:
335.Exception: The transmit buffer priority registers are 0 out of reset.

Table 502. Receive/transmit message buffer — extended identifier mapping

Register Name		Bit 7	6	5	4	3	2	1	Bit0
0x00X0 IDR0	R	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
	W								
0x00X1 IDR1	R	ID20	ID19	ID18	SRR (=1)	IDE (=1)	ID17	ID16	ID15
	W								
0x00X2 IDR2	R	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
	W								
0x00X3 IDR3	R	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
	W								
0x00X4 DSR0	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X5 DSR1	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X6 DSR2	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X7 DSR3	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X8 DSR4	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00X9 DSR5	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XA DSR6	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XB DSR7	R	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
	W								
0x00XC DLR	R					DLC3	DLC2	DLC1	DLC0
	W								

= Unused, always read ‘x’

Notes:
336.Read:
For transmit buffers, anytime when TXEx flag is set (see MSCAN transmitter flag register (CANTFLG)”) and the corresponding transmit buffer is selected in CANTBSEL (see MSCAN transmit buffer selection register (CANTBSEL)”).
For receive buffers, only when RXF flag is set (see MSCAN receiver flag register (CANRFLG)”).
Write:
For transmit buffers, anytime when TXEx flag is set (see MSCAN transmitter flag register (CANTFLG)”) and the corresponding transmit buffer is selected in CANTBSEL (see MSCAN transmit buffer selection register (CANTBSEL)”).
Unimplemented for receive buffers.
Reset:
Undefined because of RAM-based implementation

Table 503. Receive/transmit message buffer — standard identifier mapping

Register Name		Bit 7	6	5	4	3	2	1	Bit 0
IDR0 0x00X0	R	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
	W								
IDR1 0x00X1	R	ID2	ID1	ID0	RTR	IDE (=0)			
	W								
IDR2 0x00X2	R								
	W								
IDR3 0x00X3	R								
	W								
		= Unused, always read ‘x’							

6.13.3.3.1 Identifier registers (IDR0–IDR3)

The identifier registers for an extended format identifier consist of a total of 32 bits: ID[28:0], SRR, IDE, and RTR. The identifier registers for a standard format identifier consist of a total of 13 bits: ID[10:0], RTR, and IDE.

6.13.3.3.2 IDR0–IDR3 for extended identifier mapping

Table 504. Identifier register 0 (IDR0) — extended identifier mapping

Module Base + 0x00X0

	7	6	5	4	3	2	1	0
R W	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21
Reset:	x	x	x	x	x	x	x	x

Table 505. IDR0 register field descriptions — extended

Field	Description
7-0 ID[28:21]	**Extended Format Identifier** — The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

Table 506. Identifier register 1 (IDR1) — extended identifier mapping

Module Base + 0x00X1

	7	6	5	4	3	2	1	0
R W	ID20	ID19	ID18	SRR (=1)	IDE (=1)	ID17	ID16	ID15
Reset:	x	x	x	x	x	x	x	x

Table 507. IDR1 register field descriptions — extended

Field	Description
7-5 ID[20:18]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
4 SRR	**Substitute Remote Request** — This fixed recessive bit is used only in extended format. It must be set to 1 by the user for transmission buffers and is stored as received on the CAN bus for receive buffers.

Table 507. IDR1 register field descriptions — extended (continued)

Field	Description
3 IDE	**ID Extended** — This flag indicates whether the extended or standard identifier format is applied in this buffer. In the case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In the case of a transmit buffer, the flag indicates to the MSCAN what type of identifier to send. 0 Standard format (11 bit) 1 Extended format (29 bit)
2-0 ID[17:15]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

Table 508. Identifier register 2 (IDR2) — extended identifier mapping

Module Base + 0x00X2

	7	6	5	4	3	2	1	0
R W	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
Reset:	x	x	x	x	x	x	x	x

Table 509. IDR2 register field descriptions — extended

Field	Description
7-0 ID[14:7]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.

Table 510. Identifier register 3 (IDR3) — extended identifier mapping

Module Base + 0x00X3

	7	6	5	4	3	2	1	0
R W	ID6	ID5	ID4	ID3	ID2	ID1	ID0	RTR
Reset:	x	x	x	x	x	x	x	x

Table 511. IDR3 register field descriptions — extended

Field	Description
7-1 ID[6:0]	**Extended Format Identifier —** The identifiers consist of 29 bits (ID[28:0]) for the extended format. ID28 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number.
0 RTR	**Remote Transmission Request** — This flag reflects the status of the remote transmission request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame

6.13.3.3.2.1 IDR0–IDR3 for standard identifier mapping

Table 4. Identifier register 0 — standard mapping

Module Base + 0x00X0

7	6	5	4	3	2	1	0

Table 4. Identifier register 0 — standard mapping

R W	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3
Reset:	x	x	x	x	x	x	x	x

Table 512. IDR0 register field descriptions — standard

Field	Description
7-0 ID[10:3]	**Standard Format Identifier —** The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 514.

Table 513. Identifier register 1 — standard mapping

Module Base + 0x00X1

	7	6	5	4	3	2	1	0
R W	ID2	ID1	ID0	RTR	IDE (=0)			
Reset:	x	x	x	x	x	x	x	x

= Unused; always read ‘x’

Table 514. IDR1 register field descriptions

Field	Description
7-5 ID[2:0]	**Standard Format Identifier —** The identifiers consist of 11 bits (ID[10:0]) for the standard format. ID10 is the most significant bit and is transmitted first on the CAN bus during the arbitration procedure. The priority of an identifier is defined to be highest for the smallest binary number. See also ID bits in Table 512.
4 RTR	**Remote Transmission Request** — This flag reflects the status of the Remote Transmission Request bit in the CAN frame. In the case of a receive buffer, it indicates the status of the received frame and supports the transmission of an answering frame in software. In the case of a transmit buffer, this flag defines the setting of the RTR bit to be sent. 0 Data frame 1 Remote frame
3 IDE	**ID Extended** — This flag indicates whether the extended or standard identifier format is applied in this buffer. In the case of a receive buffer, the flag is set as received and indicates to the CPU how to process the buffer identifier registers. In the case of a transmit buffer, the flag indicates to the MSCAN what type of identifier to send. 0 Standard format (11 bit) 1 Extended format (29 bit)

Table 515. Identifier register 2 — standard mapping

Module Base + 0x00X2

	7	6	5	4	3	2	1	0
R W								
Reset:	x	x	x	x	x	x	x	x

= Unused; always read ‘x’

Table 516. Identifier register 3 — standard mapping

Module Base + 0x00X3

	7	6	5	4	3	2	1	0
R W								

Table 516. Identifier register 3 — standard mapping

Reset:	x	x	x	x	x	x	x	x

= Unused; always read ‘x’

6.13.3.3.3 Data segment registers (DSR0-7)

The eight data segment registers, each with bits DB[7:0], contain the data to be transmitted or received. The number of bytes to be transmitted or received is determined by the data length code in the corresponding DLR register.

Table 517. Data segment registers (DSR0–DSR7) — extended identifier mapping

Module Base + 0x00X4 to Module Base + 0x00XB

	7	6	5	4	3	2	1	0
R W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Reset:	x	x	x	x	x	x	x	x

Table 518. DSR0–DSR7 register field descriptions

Field	Description
7-0 DB[7:0]	Data bits 7-0

6.13.3.3.4 Data length register (DLR)

This register keeps the data length field of the CAN frame.

Table 519. Data length register (DLR) — extended identifier mapping

Module Base + 0x00XC

	7	6	5	4	3	2	1	0
R W					DLC3	DLC2	DLC1	DLC0
Reset:	x	x	x	x	x	x	x	x

= Unused; always read “x”

Table 520. DLR register field descriptions

Field	Description
3-0 DLC[3:0]	**Data Length Code Bits** — The data length code contains the number of bytes (data byte count) of the respective message. During the transmission of a remote frame, the data length code is transmitted as programmed while the number of transmitted data bytes is always 0. The data byte count ranges from 0 to 8 for a data frame. Table 521 shows the effect of setting the DLC bits.

Table 521. Data length codes

Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6

Table 521. Data length codes

Data Length Code				Data Byte Count
DLC3	DLC2	DLC1	DLC0	
0	1	1	1	7
1	0	0	0	8

6.13.3.3.5 Transmit buffer priority register (TBPR)

This register defines the local priority of the associated message buffer. The local priority is used for the internal prioritization process of the MSCAN and is defined to be highest for the smallest binary number. The MSCAN implements the following internal prioritization mechanisms:

- All transmission buffers with a cleared TXEx flag participate in the prioritization immediately before the SOF (start of frame) is sent.
- The transmission buffer with the lowest local priority field wins the prioritization.

In cases of more than one buffer having the same lowest priority, the message buffer with the lower index number wins.

Table 522. Transmit buffer priority register (TBPR)

Module Base + 0x00XD **Access: User read/write [337]**

	7	6	5	4	3	2	1	0
R W	PRIO7	PRIO6	PRIO5	PRIO4	PRIO3	PRIO2	PRIO1	PRIO0
Reset:	0	0	0	0	0	0	0	0

Notes:

337. Read: Anytime when TXEx flag is set (see MSCAN transmitter flag register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see MSCAN transmit buffer selection register (CANTBSEL)")
Write: Anytime when TXEx flag is set (see MSCAN transmitter flag register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see MSCAN transmit buffer selection register (CANTBSEL)"

6.13.3.3.6 Time stamp register (TSRH–TSRL)

If the TIME bit is enabled, the MSCAN will write a time stamp to the respective registers in the active transmit or receive buffer right after the EOF of a valid message on the CAN bus (see MSCAN control register 0 (CANCTL0)"). In case of a transmission, the CPU can only read the time stamp after the respective transmit buffer has been flagged empty.

The timer value, which is used for stamping, is taken from a free running internal CAN bit clock. A timer overrun is not indicated by the MSCAN. The timer is reset (all bits set to 0) during initialization mode. The CPU can only read the time stamp registers.

Table 523. Time stamp register — high byte (TSRH)

Module Base + 0x00XE							Access: User read/write [338]
7	6	5	4	3	2	1	0
R	TSR15	TSR14	TSR13	TSR12	TSR11	TSR10	TSR9	TSR8
W								
Reset:	x	x	x	x	x	x	x	x

Notes:
338.Read: Anytime when TXEx flag is set (see MSCAN transmitter flag register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see MSCAN transmit buffer selection register (CANTBSEL)")
Write: Unimplemented

Table 524. Time stamp register — low byte (TSRL)

Module Base + 0x00XF							Access: User read/write [339]
7	6	5	4	3	2	1	0
R	TSR7	TSR6	TSR5	TSR4	TSR3	TSR2	TSR1	TSR0
W								
Reset:	x	x	x	x	x	x	x	x

Notes:
339.Read: Anytime when TXEx flag is set (see MSCAN transmitter flag register (CANTFLG)") and the corresponding transmit buffer is selected in CANTBSEL (see MSCAN transmit buffer selection register (CANTBSEL)")
Write: Unimplemented

6.13.4 Functional Description

6.13.4.1 General

This section provides a complete functional description of the MSCAN.

6.13.4.2 Message storage

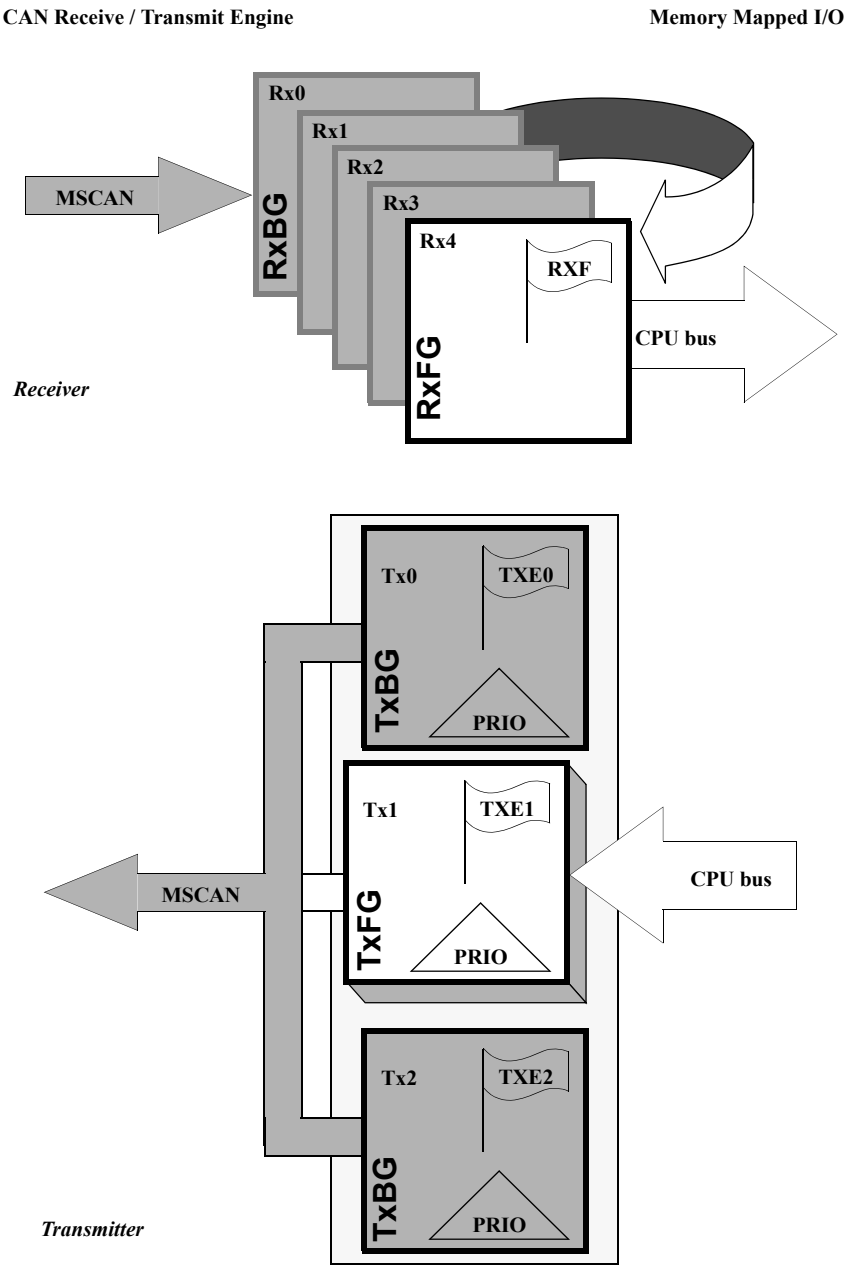


Figure 90. User model for message buffer organization

The MSCAN facilitates a sophisticated message storage system which addresses the requirements of a broad range of network applications.

6.13.4.2.1 Message transmit background

Modern application layer software is built upon two fundamental assumptions:

- Any CAN node is able to send out a stream of scheduled messages without releasing the CAN bus between the two messages. Such nodes arbitrate for the CAN bus immediately after sending the previous message and only release the CAN bus in case of lost arbitration.
- The internal message queue within any CAN node is organized such that the highest priority message is sent out first, if more than one message is ready to be sent.

The behavior described in the bullets above cannot be achieved with a single transmit buffer. That buffer must be reloaded immediately after the previous message is sent. This loading process lasts a finite amount of time and must be completed within the inter-frame sequence (IFS) to be able to send an uninterrupted stream of messages. Even if this is feasible for limited CAN bus speeds, it requires that the CPU reacts with short latencies to the transmit interrupt.

A double buffer scheme de-couples the reloading of the transmit buffer from the actual message sending and, therefore, reduces the reactiveness requirements of the CPU. Problems can arise if the sending of a message is finished while the CPU re-loads the second buffer. No buffer would then be ready for transmission, and the CAN bus would be released.

At least three transmit buffers are required to meet the first of the above requirements under all circumstances. The MSCAN has three transmit buffers.

The second requirement calls for some sort of internal prioritization which the MSCAN implements with the “local priority” concept described in Transmit structures.”

6.13.4.2.2 Transmit structures

The MSCAN triple transmit buffer scheme optimizes real-time performance by allowing multiple messages to be set up in advance. The three buffers are arranged as shown in Figure 90.

All three buffers have a 13-byte data structure similar to the outline of the receive buffers (see Programmer’s model of message storage”). An additional Transmit buffer priority register (TBPR) contains an 8-bit local priority field (PRIO) (see Transmit buffer priority register (TBPR)”). The remaining two bytes are used for time stamping of a message, if required (see Time stamp register (TSRH–TSRL)”).

To transmit a message, the CPU must identify an available transmit buffer, which is indicated by a set transmitter buffer empty (TXEx) flag (see MSCAN transmitter flag register (CANTFLG)”). If a transmit buffer is available, the CPU must set a pointer to this buffer by writing to the CANTBSEL register (see MSCAN transmit buffer selection register (CANTBSEL)”). This makes the respective buffer accessible within the CANTXFG address space (see Programmer’s model of message storage”). The algorithmic feature associated with the CANTBSEL register simplifies the transmit buffer selection. In addition, this scheme makes the handler software simpler because only one address area is applicable for the transmit process, and the required address space is minimized.

The CPU then stores the identifier, the control bits, and the data content into one of the transmit buffers. Finally, the buffer is flagged as ready for transmission by clearing the associated TXE flag.

The MSCAN then schedules the message for transmission and signals the successful transmission of the buffer by setting the associated TXE flag. A transmit interrupt (see Transmit interrupt”) is generated [(340)] when TXEx is set and can be used to drive the application software to re-load the buffer.

If more than one buffer is scheduled for transmission when the CAN bus becomes available for arbitration, the MSCAN uses the local priority setting of the three buffers to determine the prioritization. For this purpose, every transmit buffer has an 8-bit local priority field (PRIO). The application software programs this field when the message is set up. The local priority reflects the priority of this particular message relative to the set of messages being transmitted from this node. The lowest binary value of the PRIO field is defined to be the highest priority. The internal scheduling process takes place whenever the MSCAN arbitrates for the CAN bus. This is also the case after the occurrence of a transmission error.

When a high priority message is scheduled by the application software, it may become necessary to abort a lower priority message in one of the three transmit buffers. Because messages that are already in transmission cannot be aborted, the user must request the abort by setting the corresponding abort request bit (ABTRQ) (see MSCAN transmitter message abort request register (CANTARQ)”.) The MSCAN then grants the request, if possible, by:

1. Setting the corresponding abort acknowledge flag (ABTAK) in the CANTAAK register.
2. Setting the associated TXE flag to release the buffer.
3. Generating a transmit interrupt. The transmit interrupt handler software can determine from the setting of the ABTAK flag whether the message was aborted (ABTAK = 1) or sent (ABTAK = 0).

Notes:

340.The transmit interrupt occurs only if not masked. A polling scheme can also be applied on TXEx.

6.13.4.2.3 Receive structures

The received messages are stored in a five stage input FIFO. The five message buffers are alternately mapped into a single memory area (see Figure 90). The background receive buffer (RxBG) is exclusively associated with the MSCAN, but the foreground receive buffer (RxFG) is addressable by the CPU (see Figure 90). This scheme simplifies the handler software because only one address area is applicable for the receive process.

All receive buffers have a size of 15 bytes to store the CAN control bits, the identifier (standard or extended), the data contents, and a time stamp, if enabled (see Programmer's model of message storage").

The receiver full flag (RXF) (see MSCAN receiver flag register (CANRFLG)") signals the status of the foreground receive buffer. When the buffer contains a correctly received message with a matching identifier, this flag is set.

On reception, each message is checked to see whether it passes the filter (see Identifier acceptance filter") and simultaneously is written into the active RxBG. After successful reception of a valid message, the MSCAN shifts the content of RxBG into the receiver FIFO, sets the RXF flag, and generates a receive interrupt [(341)] (see Receive interrupt") to the CPU. The user's receive handler must read the received message from the RxFG and then reset the RXF flag to acknowledge the interrupt and to release the foreground buffer. A new message, which can follow immediately after the IFS field of the CAN frame, is received into the next available RxBG. If the MSCAN receives an invalid message in its RxBG (wrong identifier, transmission errors, etc.) the actual contents of the buffer is over-written by the next message. The buffer is not shifted into the FIFO.

When the MSCAN module is transmitting, the MSCAN receives its own transmitted messages into the background receive buffer, RxBG, but does not shift it into the receiver FIFO, generate a receive interrupt, or acknowledge its own messages on the CAN bus. The exception to this rule is in loopback mode (see MSCAN control register 1 (CANCTL1)") where the MSCAN treats its own messages exactly like all other incoming messages. The MSCAN receives its own transmitted messages in the event that it loses arbitration. If arbitration is lost, the MSCAN must be prepared to become a receiver.

An overrun condition occurs when all receive message buffers in the FIFO are filled with correctly received messages with accepted identifiers and another message is correctly received from the CAN bus with an accepted identifier. The latter message is discarded and an error interrupt with overrun indication is generated if enabled (see Error interrupt"). The MSCAN remains able to transmit messages while the receiver FIFO is being filled, but all incoming messages are discarded. As soon as a receive buffer in the FIFO is available again, new valid messages will be accepted.

Notes:

341.The receive interrupt occurs only if not masked. A polling scheme can be applied on RXF also.

6.13.4.3 Identifier acceptance filter

The MSCAN identifier acceptance registers (see MSCAN identifier acceptance control register (CANIDAC)") define the acceptable patterns of the standard or extended identifier (ID[10:0] or ID[28:0]). Any of these bits can be marked 'don't care' in the MSCAN identifier mask registers (see MSCAN identifier mask registers (CANIDMR0–CANIDMR7)").

A filter hit is indicated to the application software by a set receive buffer full flag (RXF = 1) and three bits in the CANIDAC register (see MSCAN identifier acceptance control register (CANIDAC)"). These identifier hit flags (IDHIT[2:0]) clearly identify the filter section that caused the acceptance. They simplify the application software's task to identify the cause of the receiver interrupt. If more than one hit occurs (two or more filters match), the lower hit has priority.

A very flexible programmable generic identifier acceptance filter has been introduced to reduce the CPU interrupt loading. The filter is programmable to operate in four different modes:

- Two identifier acceptance filters, each to be applied to:
  - The full 29 bits of the extended identifier and to the following bits of the CAN 2.0B frame:
    - Remote transmission request (RTR)
    - Identifier extension (IDE)
    - Substitute remote request (SRR)
  - The 11 bits of the standard identifier plus the RTR and IDE bits of the CAN 2.0A/B messages. This mode implements two filters for a full length CAN 2.0B compliant extended identifier. Although this mode can be used for standard identifiers, it is recommended to use the four or eight identifier acceptance filters. Figure 91 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces a filter 0 hit. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces a filter 1 hit.
- Four identifier acceptance filters, each to be applied to:
  - The 14 most significant bits of the extended identifier plus the SRR and IDE bits of CAN 2.0B messages.
  - The 11 bits of the standard identifier, the RTR and IDE bits of CAN 2.0A/B messages. Figure 92 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 and 1 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 2 and 3 hits.

- Eight identifier acceptance filters, each to be applied to the first 8 bits of the identifier. This mode implements eight independent filters for the first 8 bits of a CAN 2.0A/B compliant standard identifier or a CAN 2.0B compliant extended identifier. Figure 93 shows how the first 32-bit filter bank (CANIDAR0–CANIDAR3, CANIDMR0–CANIDMR3) produces filter 0 to 3 hits. Similarly, the second filter bank (CANIDAR4–CANIDAR7, CANIDMR4–CANIDMR7) produces filter 4 to 7 hits.
- Closed filter. No CAN message is copied into the foreground buffer RxFG, and the RXF flag is never set.

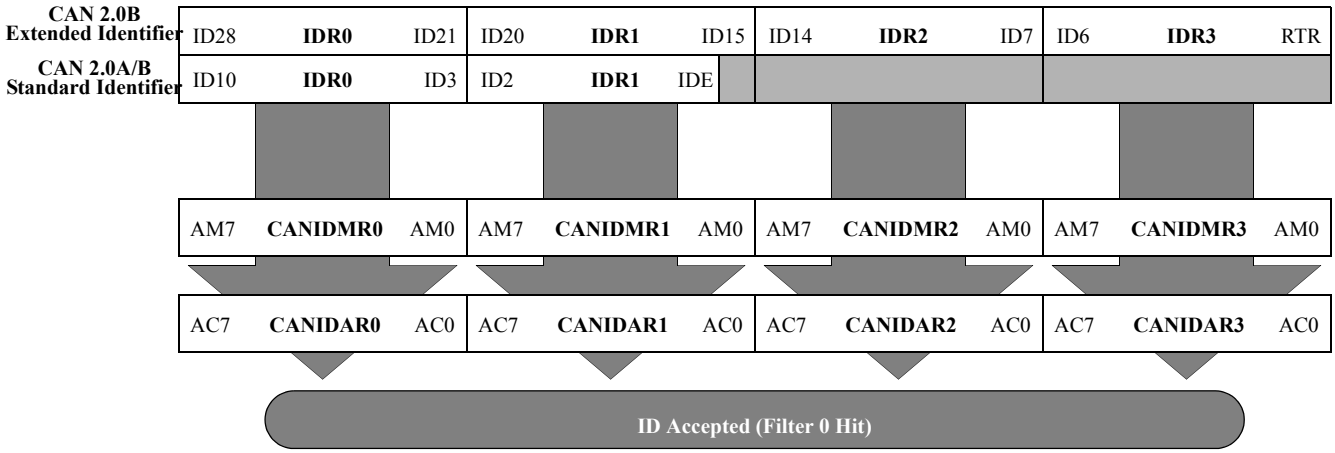


Figure 91. 32-bit maskable identifier acceptance filter

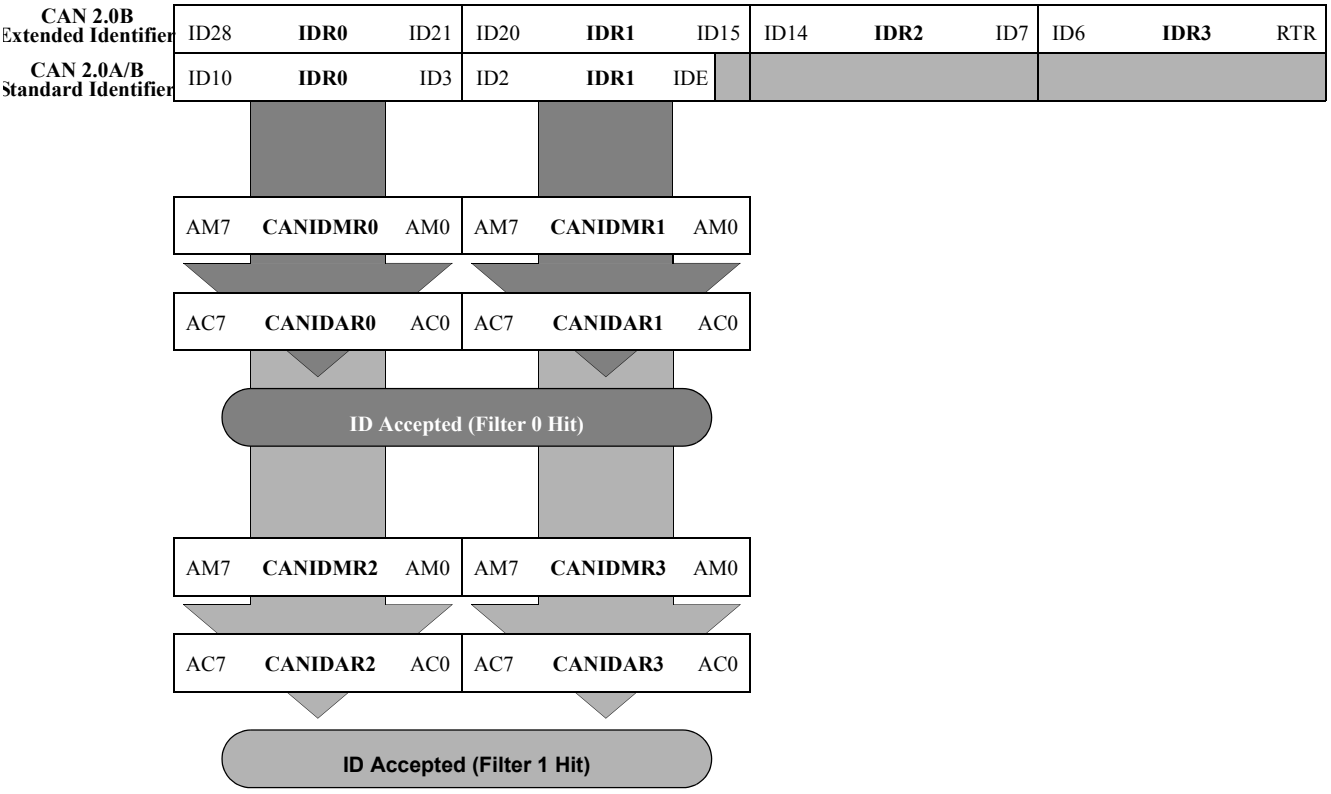


Figure 92. 16-bit maskable identifier acceptance filters

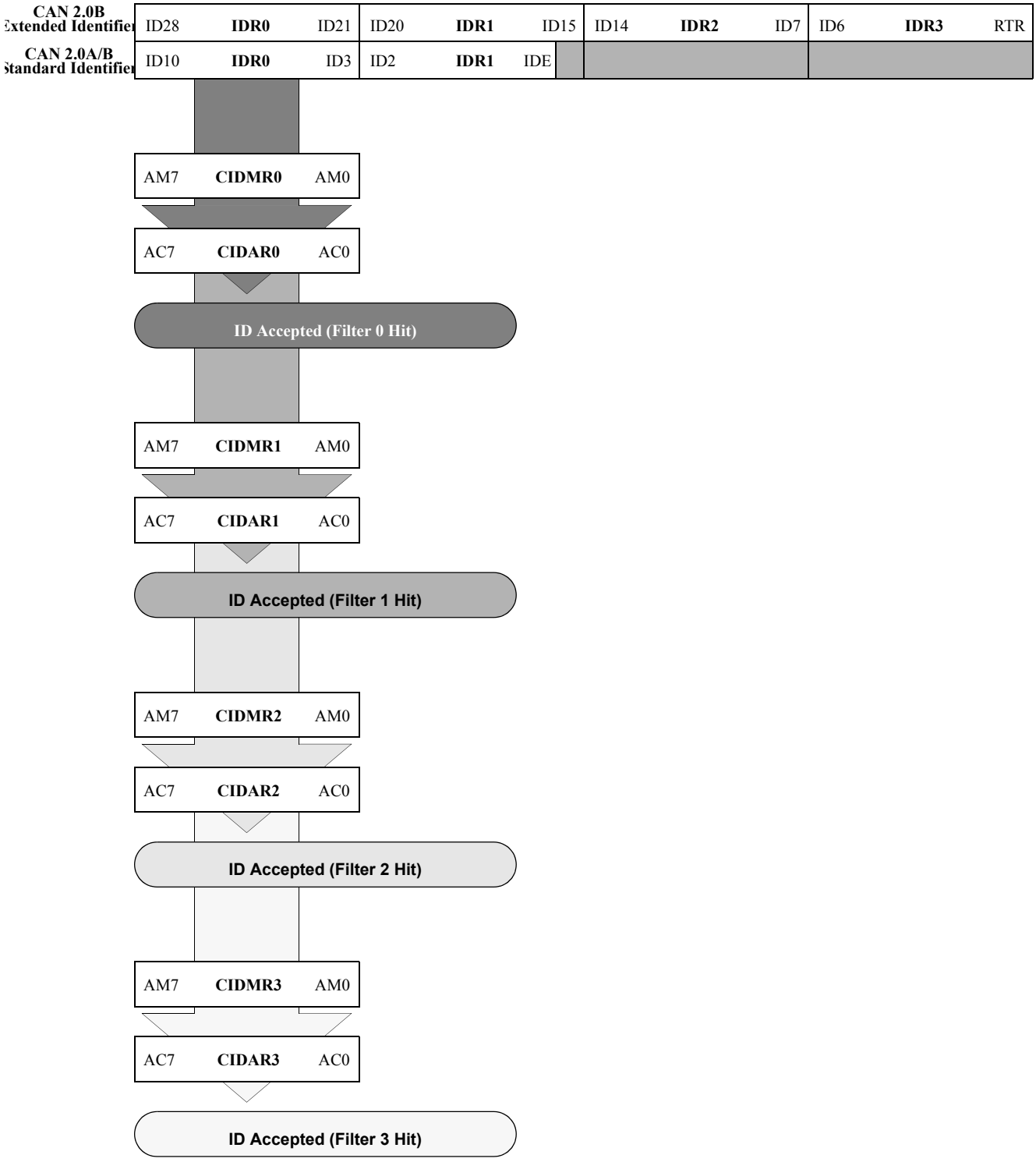


Figure 93. 8-bit maskable identifier acceptance filters

6.13.4.3.1 Protocol violation protection

The MSCAN protects the user from accidentally violating the CAN protocol through programming errors. The protection logic implements the following features:

- The receive and transmit error counters cannot be written or otherwise manipulated.
- All registers which control the configuration of the MSCAN cannot be modified while the MSCAN is on-line. The MSCAN has to be in Initialization Mode. The corresponding INITRQ/INITAK handshake bits in the CANCTL0/CANCTL1 registers (see MSCAN control register 0 (CANCTL0)") serve as a lock to protect the following registers:
  - MSCAN control 1 register (CANCTL1)
  - MSCAN bus timing registers 0 and 1 (CANBTR0, CANBTR1)
  - MSCAN identifier acceptance control register (CANIDAC)
  - MSCAN identifier acceptance registers (CANIDAR0–CANIDAR7)
  - MSCAN identifier mask registers (CANIDMR0–CANIDMR7)
- The TXCAN is immediately forced to a recessive state when the MSCAN goes into the power down mode or initialization mode (see MSCAN power down mode," and MSCAN initialization mode").
- The MSCAN enable bit (CANE) is writable only once in normal system operation modes, which provides further protection against inadvertently disabling the MSCAN.

6.13.4.3.2 Clock system

Figure 94 shows the structure of the MSCAN clock generation circuitry.

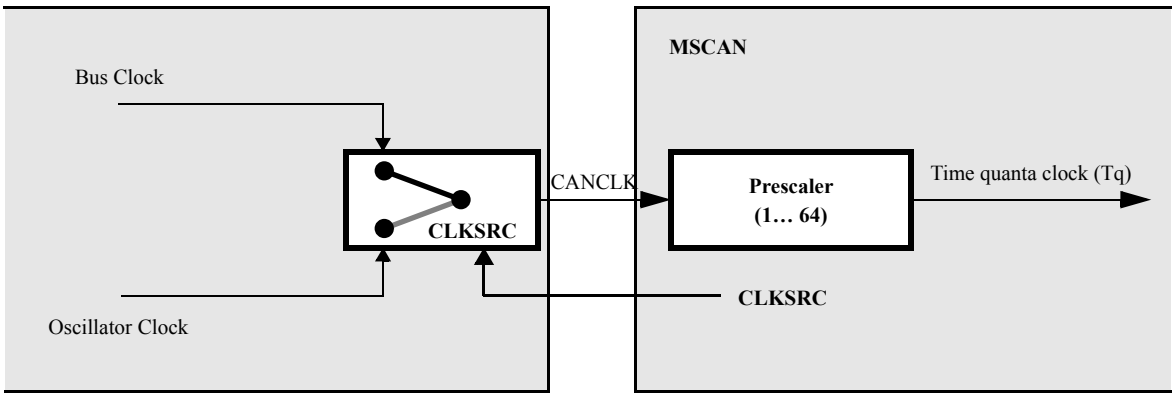


Figure 94. MSCAN clocking scheme

The clock source bit (CLKSRC) in the CANCTL1 register (MSCAN control register 1 (CANCTL1)) defines whether the internal CANCLK is connected to the output of a crystal oscillator (oscillator clock) or to the bus clock.

The clock source has to be chosen such that the tight oscillator tolerance requirements (up to 0.4%) of the CAN protocol are met. Additionally, for high CAN bus rates (1.0 Mbps), a 45% to 55% duty cycle of the clock is required.

If the bus clock is generated from a PLL, it is recommended to select the oscillator clock rather than the bus clock due to jitter considerations, especially at the faster CAN bus rates.

For microcontrollers without a clock and reset generator (CRG), CANCLK is driven from the crystal oscillator (oscillator clock).

A programmable prescaler generates the time quanta (Tq) clock from CANCLK. A time quantum is the atomic unit of time handled by the MSCAN.

$$f_{Tq} = \frac{f_{CANCLK}}{(Prescaler\ \ value)}$$

A bit time is subdivided into three segments as described in the Bosch CAN 2.0A/B specification. (see Figure 95):

- SYNC_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section.
- Time Segment 1: This segment includes the PROP_SEG and the PHASE_SEG1 of the CAN standard. It can be programmed by setting the parameter TSEG1 to consist of 4 to 16 time quanta.
- Time Segment 2: This segment represents the PHASE_SEG2 of the CAN standard. It can be programmed by setting the TSEG2 parameter to be 2 to 8 time quanta long.

$$Bit\ \ Rate = \frac{f_{Tq}}{(number\ \ of\ \ Time\ \ Quanta)}$$

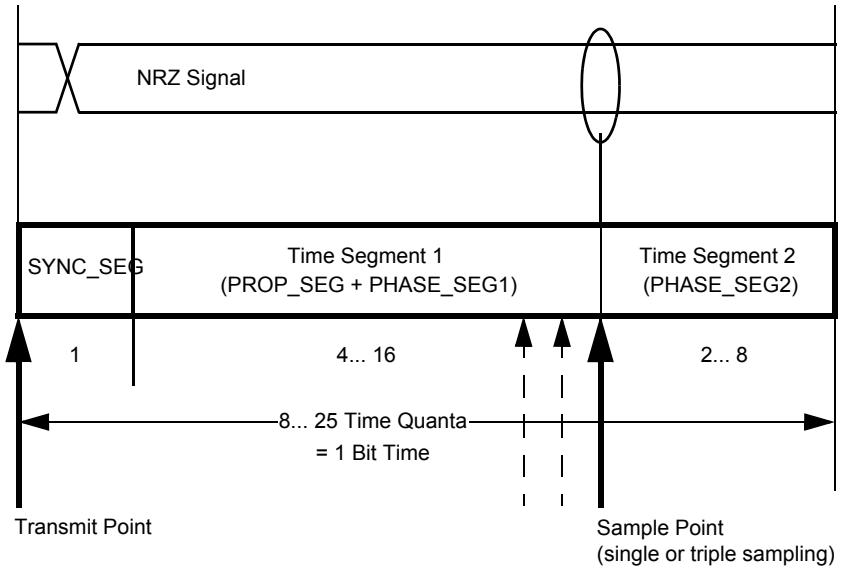


Figure 95. Segments within the bit time

Table 525. Time segment syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the CAN bus during this period.
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node in receive mode samples the CAN bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

The synchronization jump width (see the Bosch CAN 2.0A/B specification for details) can be programmed in a range of 1 to 4 time quanta by setting the SJW parameter.

The SYNC_SEG, TSEG1, TSEG2, and SJW parameters are set by programming the MSCAN bus timing registers (CANBTR0, CANBTR1) (see MSCAN bus timing register 0 (CANBTR0)" and MSCAN bus timing register 1 (CANBTR1)").

Table 526 gives an overview of the Bosch CAN 2.0A/B specification compliant segment settings and the related parameter values.

Note

It is the user's responsibility to ensure the bit time settings are in compliance with the CAN standard.

Table 526. Bosch CAN 2.0A/B compliant bit time segment settings

Time Segment 1	TSEG1	Time Segment 2	TSEG2	Synchronization Jump Width	SJW
5… 10	4… 9	2	1	1… 2	0… 1
4… 11	3 … 10	3	2	1 … 3	0 … 2
5 … 12	4 … 11	4	3	1 … 4	0 … 3
6 … 13	5 … 12	5	4	1 … 4	0 … 3
7 … 14	6 … 13	6	5	1 … 4	0 … 3
8 … 15	7 … 14	7	6	1 … 4	0 … 3
9 … 16	8 … 15	8	7	1 … 4	0 … 3

6.13.4.4 Modes of operation

6.13.4.4.1 Normal system operating modes

The MSCAN module behaves as described within this specification in all normal system operating modes. Write restrictions exist for some registers.

6.13.4.4.2 Special system operating modes

The MSCAN module behaves as described within this specification in all special system operating modes. Write restrictions which exist on specific registers in normal modes are lifted for test purposes in special modes.

6.13.4.4.3 Emulation modes

In all emulation modes, the MSCAN module behaves just like in normal system operating modes as described within this specification.

6.13.4.4.4 Listen-only mode

In an optional CAN bus monitoring mode (listen-only), the CAN node is able to receive valid data frames and valid remote frames, but it sends only “recessive” bits on the CAN bus. In addition, it cannot start a transmission.

If the MAC sub-layer is required to send a “dominant” bit (ACK bit, overload flag, or active error flag), the bit is rerouted internally so that the MAC sub-layer monitors this “dominant” bit, although the CAN bus may remain in recessive state externally.

6.13.4.4.5 MSCAN initialization mode

The MSCAN enters initialization mode when it is enabled (CANE=1).

When entering initialization mode during operation, any on-going transmission or reception is immediately aborted and synchronization to the CAN bus is lost, potentially causing CAN protocol violations. To protect the CAN bus system from fatal consequences of violations, the MSCAN immediately drives TXCAN into a recessive state.

Note

The user is responsible for ensuring that the MSCAN is not active when initialization mode is entered. The recommended procedure is to bring the MSCAN into sleep mode (SLPRQ = 1 and SLPAK = 1) before setting the INITRQ bit in the CANCTL0 register. Otherwise, the abort of an on-going message can cause an error condition and can impact other CAN bus devices.

In initialization mode, the MSCAN is stopped. However, interface registers remain accessible. This mode is used to reset the CANCTL0, CANRFLG, CANRIER, CANTFLG, CANTIER, CANTARQ, CANTAAK, and CANTBSEL registers to their default values. In addition, the MSCAN enables the configuration of the CANBTR0, CANBTR1 bit timing registers; CANIDAC; and the CANIDAR, CANIDMR message filters. See MSCAN control register 0 (CANCTL0),” for a detailed description of the initialization mode.

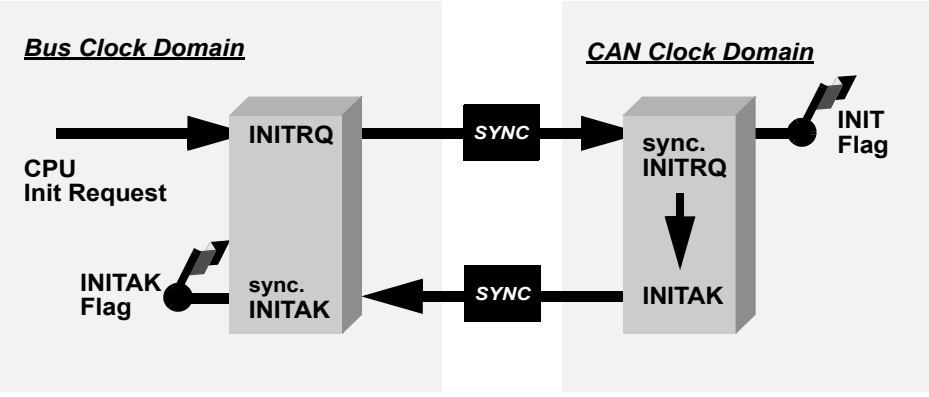


Figure 96. Initialization request/acknowledge cycle

Due to independent clock domains within the MSCAN, INITRQ must be synchronized to all domains by using a special handshake mechanism. This handshake causes additional synchronization delay (see Figure 96).

If there is no message transfer ongoing on the CAN bus, the minimum delay will be two additional bus clocks and three additional CAN clocks. When all parts of the MSCAN are in initialization mode, the INITAK flag is set. The application software must use INITAK as a handshake indication for the request (INITRQ) to go into initialization mode.

Note

The CPU cannot clear INITRQ before initialization mode (INITRQ = 1 and INITAK = 1) is active.

6.13.4.5 Low-power options

If the MSCAN is disabled (CANE = 0), the MSCAN clocks are stopped for power saving.

If the MSCAN is enabled (CANE = 1), the MSCAN has two additional modes with reduced power consumption, compared to Normal mode: Sleep and Power Down mode. In Sleep mode, power consumption is reduced by stopping all clocks except those to access the registers from the CPU side. In Power Down mode, all clocks are stopped and no power is consumed.

Table 527 summarizes the combinations of MSCAN and CPU modes. A particular combination of modes is entered by the given settings on the CSWAI and SLPRQ/SLPAK bits.

Table 527. CPU vs. MSCAN operating modes

CPU Mode	MSCAN Mode			
	Normal	Reduced Power Consumption		
		Sleep	Power Down	Disabled (CANE=0)
RUN	CSWAI = X [342] SLPRQ = 0 SLPAK = 0	CSWAI = X SLPRQ = 1 SLPAK = 1		CSWAI = X SLPRQ = X SLPAK = X
WAIT	CSWAI = 0 SLPRQ = 0 SLPAK = 0	CSWAI = 0 SLPRQ = 1 SLPAK = 1	CSWAI = 1 SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X
STOP			CSWAI = X SLPRQ = X SLPAK = X	CSWAI = X SLPRQ = X SLPAK = X

Notes:

342.'X' means don't care.

6.13.4.5.1 Operation in run mode

As shown in Table 527, only MSCAN sleep mode is available as low power option when the CPU is in run mode.

6.13.4.5.2 Operation in wait mode

The WAI instruction puts the MCU in a low power consumption stand-by mode. If the CSWAI bit is set, additional power can be saved in power down mode because the CPU clocks are stopped. After leaving this Power Down mode, the MSCAN restarts and enters Normal mode again.

While the CPU is in Wait mode, the MSCAN can be operated in Normal mode and generate interrupts (registers can be accessed via background debug mode).

6.13.4.5.3 Operation in stop mode

The STOP instruction puts the MCU in a low power consumption stand-by mode. In Stop mode, the MSCAN is set in power down mode regardless of the value of the SLPRQ/SLPAK and CSWAI bits (Table 527).

6.13.4.5.4 MSCAN normal mode

This is a non-power-saving mode. Enabling the MSCAN puts the module from disabled mode into Normal mode. In this mode the module can either be in initialization mode or out of initialization mode. See MSCAN initialization mode".

6.13.4.5.5 MSCAN sleep mode

The CPU can request the MSCAN to enter this low power mode by asserting the SLPRQ bit in the CANCTL0 register. The time when the MSCAN enters Sleep mode depends on a fixed synchronization delay and its current activity:

- If there are one or more message buffers scheduled for transmission (TXEx = 0), the MSCAN will continue to transmit until all transmit message buffers are empty (TXEx = 1, transmitted successfully or aborted) and then goes into Sleep mode.
- If the MSCAN is receiving, it continues to receive and goes into Sleep mode as soon as the CAN bus next becomes idle.
- If the MSCAN is neither transmitting nor receiving, it immediately goes into Sleep mode.

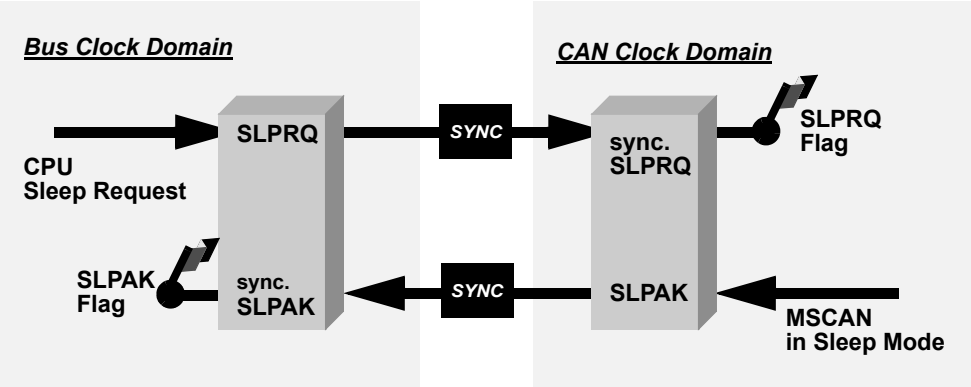


Figure 97. Sleep request / acknowledge cycle

Note

The application software must avoid setting up a transmission (by clearing one or more TXEx flag(s)) and immediately request sleep mode (by setting SLPRQ). Whether the MSCAN starts transmitting or goes into Sleep mode directly depends on the exact sequence of operations.

If Sleep mode is active, the SLPRQ and SLPAK bits are set (Figure 97). The application software must use SLPAK as a handshake indication for the request (SLPRQ) to go into Sleep mode.

When in Sleep mode (SLPRQ = 1 and SLPAK = 1), the MSCAN stops its internal clocks. However, clocks that allow register accesses from the CPU side continue to run.

If the MSCAN is in bus-off state, it stops counting the 128 occurrences of 11 consecutive recessive bits due to the stopped clocks. TXCAN remains in a recessive state. If RXF = 1, the message can be read and RXF can be cleared. Shifting a new message into the foreground buffer of the receiver FIFO (RxFG) does not take place while in sleep mode.

It is possible to access the transmit buffers and to clear the associated TXE flags. No message abort takes place while in sleep mode.

If the WUPE bit in CANCTL0 is not asserted, the MSCAN will mask any activity it detects on CAN. RXCAN is therefore held internally in a recessive state. This locks the MSCAN in sleep mode. WUPE must be set before entering sleep mode to take effect.

The MSCAN is able to leave sleep mode (wake up) only when:

- CAN bus activity occurs and WUPE = 1

  or
- the CPU clears the SLPRQ bit

Note

The CPU cannot clear the SLPRQ bit before sleep mode (SLPRQ = 1 and SLPAK = 1) is active.

After wake-up, the MSCAN waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, if the MSCAN is woken-up by a CAN frame, this frame is not received.

The receive message buffers (RxFG and RxBG) contain messages if they were received before sleep mode was entered. All pending actions will be executed upon wake-up; copying of RxBG into RxFG, message aborts and message transmissions. If the MSCAN remains in bus-off state after sleep mode was exited, it continues counting the 128 occurrences of 11 consecutive recessive bits.

6.13.4.5.6 MSCAN power down mode

The MSCAN is in power down mode (Table 527) when

- CPU is in stop mode

  or

- CPU is in wait mode and the CSWAI bit is set

When entering the power down mode, the MSCAN immediately stops all ongoing transmissions and receptions, potentially causing CAN protocol violations. To protect the CAN bus system from fatal consequences of violations to the above rule, the MSCAN immediately drives TXCAN into a recessive state.

Note

The user is responsible for ensuring that the MSCAN is not active when power down mode is entered. The recommended procedure is to bring the MSCAN into Sleep mode before the STOP or WAI instruction (if CSWAI is set) is executed. Otherwise, the abort of an ongoing message can cause an error condition and impact other CAN bus devices.

In power down mode, all clocks are stopped and no registers can be accessed. If the MSCAN was not in sleep mode before power down mode became active, the module performs an internal recovery cycle after powering up. This causes some fixed delay before the module enters normal mode again.

6.13.4.5.7 Disabled mode

The MSCAN is in disabled mode out of reset (CANE=0). All module clocks are stopped for power saving, however the register map can still be accessed as specified.

6.13.4.5.8 Programmable wake-up function

The MSCAN can be programmed to wake up from sleep or power down mode as soon as CAN bus activity is detected (see control bit WUPE in MSCAN control register 0 (CANCTL0). The sensitivity to existing CAN bus action can be modified by applying a low-pass filter function to the RXCAN input line (see control bit WUPM in MSCAN control register 1 (CANCTL1)").

This feature can be used to protect the MSCAN from wake-up due to short glitches on the CAN bus lines. Such glitches can result from—for example—electromagnetic interference within noisy environments.

6.13.4.6 Reset initialization

The reset state of each individual bit is listed in Register descriptions," which details all the registers and their bit-fields.

6.13.4.7 Interrupts

This section describes all interrupts originated by the MSCAN. It documents the enable bits and generated flags. Each interrupt is listed and described separately.

6.13.4.7.1 Description of interrupt operation

The MSCAN supports four interrupt vectors (see Table 528), any of which can be individually masked (for details see MSCAN receiver interrupt enable register (CANRIER)" to MSCAN transmitter interrupt enable register (CANTIER)").

Refer to the device overview section to determine the dedicated interrupt vector addresses.

Table 528. Interrupt vectors

Interrupt Source	CCR Mask	Local Enable
Wake-up Interrupt (WUPIF)	I bit	CANRIER (WUPIE)
Error Interrupts Interrupt (CSCIF, OVRIF)	I bit	CANRIER (CSCIE, OVRIE)
Receive Interrupt (RXF)	I bit	CANRIER (RXFIE)
Transmit Interrupts (TXE[2:0])	I bit	CANTIER (TXEIE[2:0])

6.13.4.7.2 Transmit interrupt

At least one of the three transmit buffers is empty (not scheduled) and can be loaded to schedule a message for transmission. The TXEx flag of the empty message buffer is set.

6.13.4.7.3 Receive interrupt

A message is successfully received and shifted into the foreground buffer (RxFG) of the receiver FIFO. This interrupt is generated immediately after receiving the EOF symbol. The RXF flag is set. If there are multiple messages in the receiver FIFO, the RXF flag is set as soon as the next message is shifted to the foreground buffer.

6.13.4.7.4 Wake-up interrupt

A wake-up interrupt is generated if activity on the CAN bus occurs during MSCAN sleep or power-down mode.

Note

This interrupt can only occur if the MSCAN was in sleep mode (SLPRQ = 1 and SLPAK = 1) before entering power down mode, the wake-up option is enabled (WUPE = 1), and the wake-up interrupt is enabled (WUPIE = 1).

6.13.4.7.5 Error interrupt

An error interrupt is generated if an overrun of the receiver FIFO, error, warning, or bus-off condition occurrs. MSCAN receiver flag register (CANRFLG) indicates one of the following conditions:

- **Overrun** — An overrun condition of the receiver FIFO as described in Receive structures," occurred.
- **CAN Status Change** — The actual value of the transmit and receive error counters control the CAN bus state of the MSCAN. As soon as the error counters skip into a critical range (Tx/Rx-warning, Tx/Rx-error, bus-off), the MSCAN flags an error condition. The status change, which caused the error condition, is indicated by the TSTAT and RSTAT flags (see MSCAN receiver flag register (CANRFLG)" and MSCAN receiver interrupt enable register (CANRIER)").

6.13.4.7.6 Interrupt acknowledge

Interrupts are directly associated with one or more status flags in either the MSCAN receiver flag register (CANRFLG) or the MSCAN transmitter flag register (CANTFLG). Interrupts are pending as long as one of the corresponding flags is set. The flags in CANRFLG and CANTFLG must be reset within the interrupt handler to handshake the interrupt. The flags are reset by writing a 1 to the corresponding bit position. A flag cannot be cleared if the respective condition prevails.

Note

It must be guaranteed that the CPU clears only the bit causing the current interrupt. For this reason, bit manipulation instructions (BSET) must not be used to clear interrupt flags. These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

6.13.5 Initialization/application information

6.13.5.1 MSCAN initialization

The procedure to initially start up the MSCAN module out of reset is as follows:

1. Assert CANE
2. Write to the configuration registers in initialization mode
3. Clear INITRQ to leave initialization mode

If the configuration of registers which are only writable in initialization mode shall be changed:

1. Bring the module into sleep mode by setting SLPRQ and awaiting SLPAK to assert after the CAN bus becomes idle.
2. Enter initialization mode: assert INITRQ and await INITAK
3. Write to the configuration registers in initialization mode
4. Clear INITRQ to leave initialization mode and continue

6.13.5.2 Bus-off recovery

The bus-off recovery is user configurable. The bus-off state can either be left automatically or on user request.

For reasons of backwards compatibility, the MSCAN defaults to automatic recovery after reset. In this case, the MSCAN will become error active again after counting 128 occurrences of 11 consecutive recessive bits on the CAN bus (see the Bosch CAN 2.0 A/B specification for details).

If the MSCAN is configured for user request (BORM set in MSCAN control register 1 (CANCTL1)), the recovery from bus-off starts after both independent events have become true:

- 128 occurrences of 11 consecutive recessive bits on the CAN bus have been monitored
- BOHOLD in MSCAN miscellaneous register (CANMISC) has been cleared by the user

These two events may occur in any order.

6.14 128 KB flash module (S12ZFTMRZ128K4KV1)

6.14.1 Introduction

The FTMRZ128K4K module implements the following:

- 128 kbytes of P-Flash (Program Flash) memory
- 4.0 kbytes of EEPROM memory

The Flash memory is ideal for single supply applications allowing for field reprogramming without requiring external high voltage sources for program or erase operations. The Flash module includes a memory controller that executes commands to modify Flash memory contents. The user interface to the memory controller consists of the indexed Flash Common Command Object (FCCOB) register which is written to with the command, global address, data, and any required command parameters. The memory controller must complete the execution of a command before the FCCOB register can be written to with a new command.

Note

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

The Flash memory may be read as bytes and aligned words. Read access time is one bus cycle for bytes and aligned words. For misaligned words access, the CPU has to perform twice the byte read access command. For Flash memory, an erased bit reads 1 and a programmed bit reads 0.

It is possible to read from P-Flash memory while some commands are executing on EEPROM memory. It is not possible to read from EEPROM memory while a command is executing on P-Flash memory. Simultaneous P-Flash and EEPROM operations are discussed in Allowed simultaneous p-flash and EEPROM operations.

Both P-Flash and EEPROM memories are implemented with Error Correction Codes (ECC) that can resolve single bit faults and detect double bit faults. For P-Flash memory, the ECC implementation requires that programming be done on an aligned 8 byte basis (a Flash phrase). Since P-Flash memory is always read by half-phrase, only one single bit fault in an aligned 4 byte half-phrase containing the byte or word accessed will be corrected.

6.14.1.1 Glossary

Table 529 shows a glossary of the major terms used in this document.

Table 529. Glossary

Term	Definition
Command Write Sequence	An MCU instruction sequence to execute built-in algorithms (including program and erase) on the Flash memory.
EEPROM Memory	The EEPROM memory constitutes the nonvolatile memory store for data.
EEPROM Sector	The EEPROM sector is the smallest portion of the EEPROM memory that can be erased. The EEPROM sector consists of 4 bytes.
NVM Command Mode	An NVM mode using the CPU to setup the FCCOB register to pass parameters required for Flash command execution.

Table 529. Glossary (continued)

Term	Definition
Phrase	An aligned group of four 16-bit words within the P-Flash memory. Each phrase includes two sets of aligned double words with each set including 7 ECC bits for single bit fault correction and double bit fault detection within each double word.
P-Flash Memory	The P-Flash memory constitutes the main nonvolatile memory store for applications.
P-Flash Sector	The P-Flash sector is the smallest portion of the P-Flash memory that can be erased. Each P-Flash sector contains 512 bytes.
Program IFR	Nonvolatile information register located in the P-Flash block that contains the Version ID, and the Program Once field.

6.14.1.2 Features

6.14.1.2.1 P-flash features

- 128 kbytes of P-Flash composed of one 128 kbyte Flash block divided into 256 sectors of 512 bytes
- Single bit fault correction and double bit fault detection within a 32-bit double word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and phrase program operation
- Ability to read the P-Flash memory while programming a word in the EEPROM memory
- Flexible protection scheme to prevent accidental program or erase of P-Flash memory

6.14.1.2.2 EEPROM features

- 4Kbytes of EEPROM memory composed of one 4 Kbytes Flash block divided into 1024 sectors of 4 bytes
- Single bit fault correction and double bit fault detection within a word during read operations
- Automated program and erase algorithm with verify and generation of ECC parity bits
- Fast sector erase and word program operation
- Protection scheme to prevent accidental program or erase of EEPROM memory
- Ability to program up to four words in a burst sequence

6.14.1.2.3 Other flash module features

- No external high-voltage power supply required for Flash memory program and erase operations
- Interrupt generation on Flash command completion and Flash error detection
- Security mechanism to prevent unauthorized access to the Flash memory

6.14.1.3 Block diagram

The block diagram of the Flash module is shown in Figure 98.

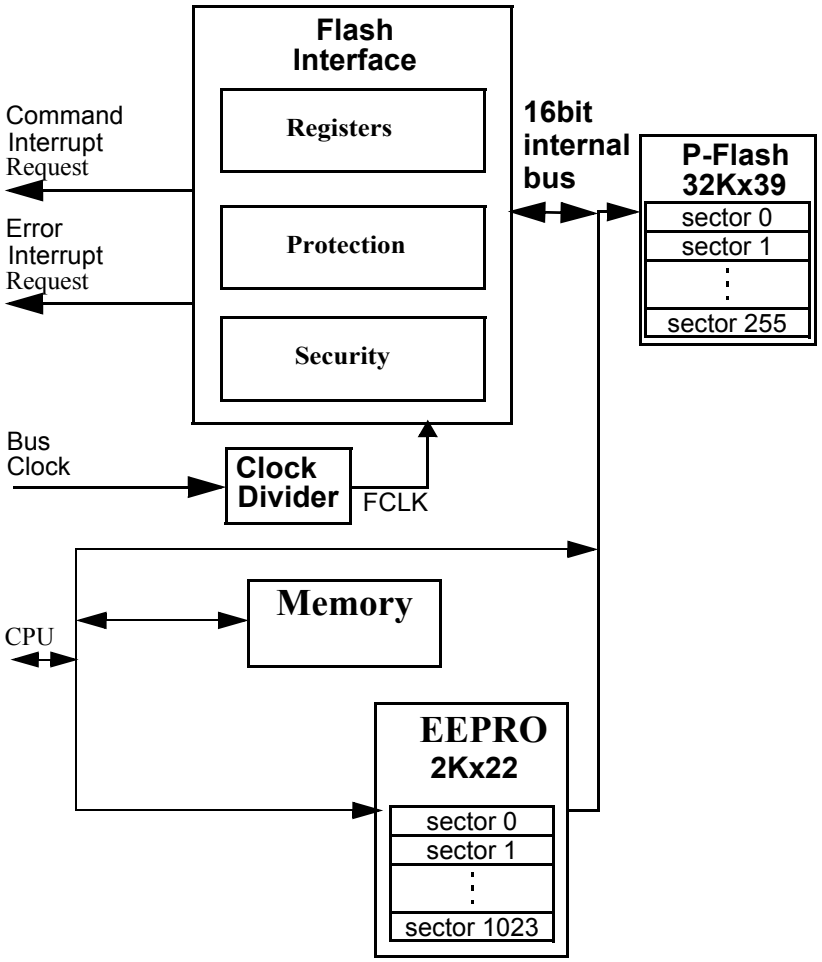


Figure 98. FTMRZ128K4K block diagram

6.14.2 External signal description

The Flash module contains no signals that connect off-chip.

6.14.3 Memory map and registers

This section describes the memory map and registers for the Flash module. Read data from unimplemented memory space in the Flash module is undefined. Write access to unimplemented or reserved memory space in the Flash module will be ignored by the Flash module.

Note

Writing to the Flash registers while a Flash command is executing (that is indicated when the value of flag CCIF reads as '0') is not allowed. If such action is attempted the write operation will not change the register value.

Writing to the Flash registers is allowed when the Flash is not busy executing commands (CCIF = 1) and during initialization right after reset, despite the value of flag CCIF in that case (refer to Initialization for a complete description of the reset sequence).

Table 530. FTMRZ memory map

Global Address (in Bytes)	Size (Bytes)	Description
0x0_0000 – 0x0_0FFF	4,096	Register Space
0x10_0000 – 0x10_0FFF	4,096	EEPROM memory
0x1F_4000 – 0x1F_FFFF	49,152	NVMRES [343] =1: NVM Resource area (see Figure 100)
0xFE_0000 – 0xFF_FFFF	131,072	P-Flash Memory

Notes:
343.See NVMRES description in Internal NVM resource (NVMRES)

6.14.3.1 Module memory map

The S12 architecture places the P-Flash memory between global addresses

0xFE_0000 and 0xFF_FFFF as shown in Table 531

The P-Flash memory map is shown in Figure .

Table 531. P-flash memory addressing

Global Address	Size (Bytes)	Description
0xFE_0000 – 0xFF_FFFF	128 k	P-Flash Block. Contains Flash Configuration Field (see Table 532)

The FPROT register, described in P-flash protection register (FPROT), can be set to protect regions in the Flash memory from accidental program or erase. Three separate memory regions, one growing upward from global address 0xFF_8000 in the Flash memory (called the lower region), one growing downward from global address 0xFF_FFFF in the Flash memory (called the higher region), and the remaining addresses in the Flash memory, can be activated for protectionThe Flash memory addresses covered by these protectable regions are shown in the P-Flash memory map. The higher address region is mainly targeted to hold the boot loader code since it covers the vector space. Default protection settings as well as security information that allows the MCU to restrict access to the Flash module are stored in the Flash configuration field as described in Table 532.

Table 532. Flash configuration field

Global Address	Size (Bytes)	Description
0xFF_FE00-0xFF_FE07	8	Backdoor Comparison Key. Refer to Verify backdoor access key command," and Unsecuring the MCU using backdoor key access"
0xFF_FE08-0xFF_FE09 [344]	2	Protection Override Comparison Key. Refer to Protection override command"
0xFF_FE0A-0xFF_FE0B [344]	2	Reserved
0xFF_FE0C [344]	1	P-Flash Protection byte. Refer to P-flash protection register (FPROT)"
0xFF_FE0D [344]	1	EEPROM Protection byte. Refer to EEPROM protection register (DFPROT)"
0xFF_FE0E [344]	1	Flash Nonvolatile byte. Refer to Flash option register (FOPT)"
0xFF_FE0F [344]	1	Flash Security byte. Refer to Flash security register (FSEC)"

Notes:
344.0xFF_FE08-0xFF_FE0F form a Flash phrase and must be programmed in a single command write sequence. Each byte in the 0xFF_FE0A - 0xFF_FE0B reserved field should be programmed to 0xFF.

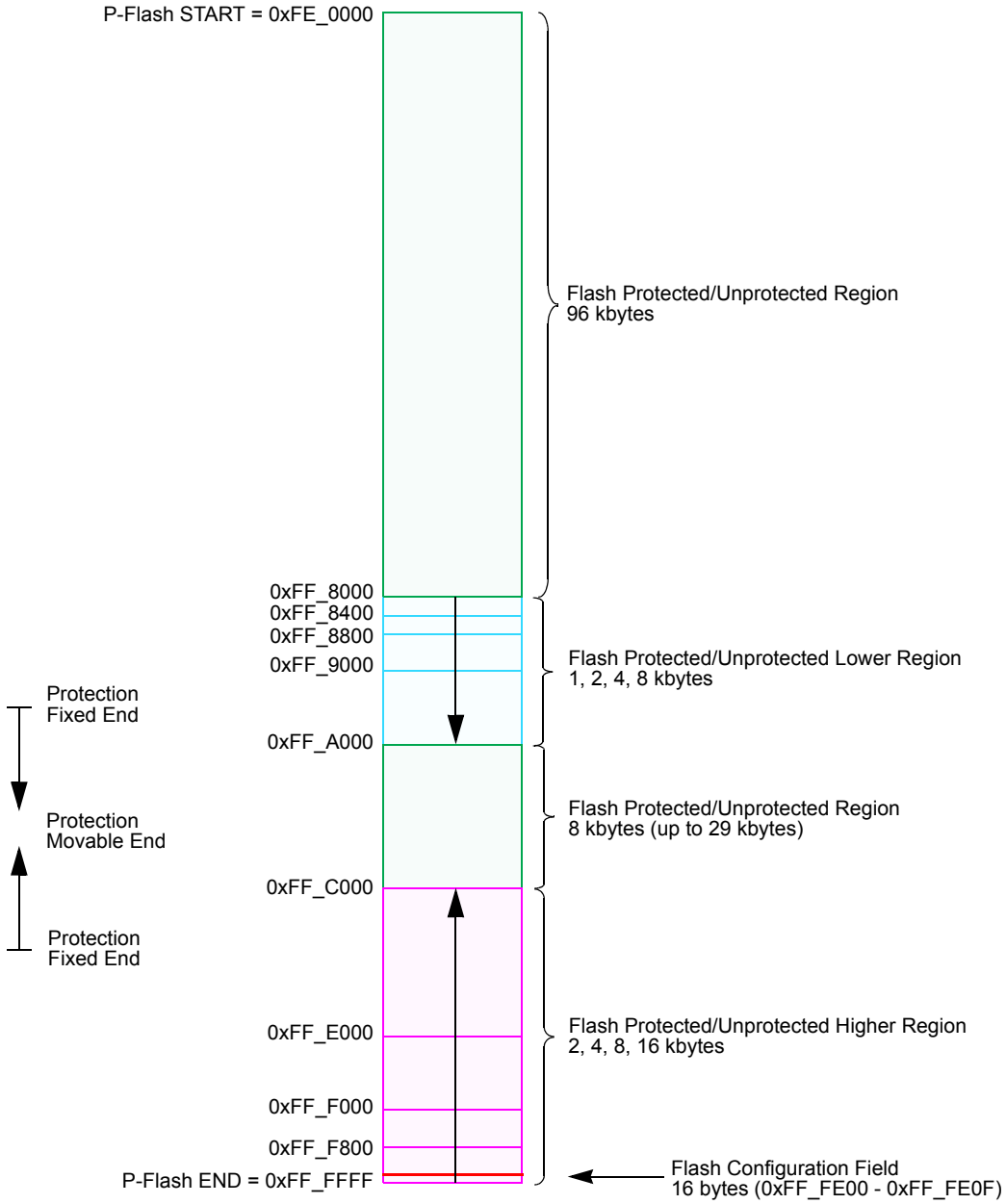


Figure 99. P-flash memory map

Table 533. Program IFR fields

Global Address	Size (Bytes)	Field Description
0x1F_C000 – 0x1F_C007	8	Reserved
0x1F_C008 – 0x1F_C0B5	174	Reserved
0x1F_C0B6 – 0x1F_C0B7	2	Version ID[345]
0x1F_C0B8 – 0x1F_C0BF	8	Reserved
0x1F_C0C0 – 0x1F_C0FF	64	Program Once Field. Refer to Program once command"

Notes:
345.Used to track firmware patch versions, see IFR version ID word

Table 534. Memory controller resource fields (NVMRES =1)[346]

Global Address	Size (Bytes)	Description
0x1F_4000 – 0x1F_41FF	512	Reserved
0x1F_4200 – 0x1F_7FFF	15,872	Reserved
0x1F_8000 – 0x1F_93FF	5,120	Reserved
0x1F_9400 – 0x1F_BFFF	11,264	Reserved
0x1F_C000 – 0x1F_C0FF	256	P-Flash IFR (see Table 533)
0x1F_C100 – 0x1F_FFFF	16,128	Reserved.

Notes:
346.NVMRES - See Internal NVM resource (NVMRES) for NVMRES (NVM Resource) detail.

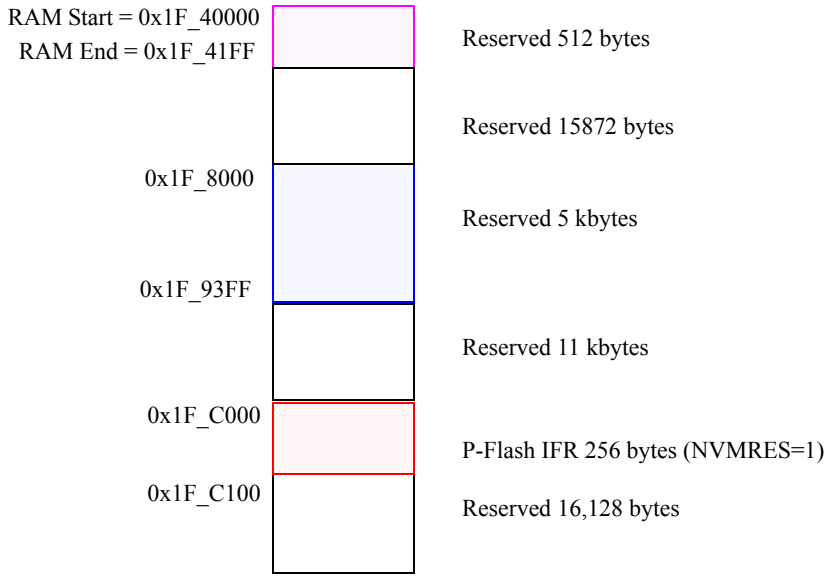


Figure 100. Memory Controller resource memory map (NVMRES=1)

6.14.3.2 Register descriptions

The Flash module contains a set of 24 control and status registers located between Flash module base + 0x0000 and 0x0017.

In the case of the writable registers, the write accesses are forbidden during Flash command execution (for more detail, see note in Memory map and registers).

A summary of the Flash module registers is given in Figure 535 with detailed descriptions in the following subsections.

Table 535. FTMRC128K1 register summary

Address & Name		7	6	5	4	3	2	1	0
0x0000 FCLKDIV	R	FDIVLD	FDIVLCK	FDIV5	FDIV4	FDIV3	FDIV2	FDIV1	FDIV0
	W								
0x0001 FSEC	R	KEYEN1	KEYEN0	RNV5	RNV4	RNV3	RNV2	SEC1	SEC0
	W								
0x0002 FCCOBIX	R	0	0	0	0	0	CCOBIX2	CCOBIX1	CCOBIX0
	W								
0x0003 FPSTAT	R	FPOVRD	0	0	0	0	0	0	0
	W								
0x0004 FCNFG	R	CCIE	0	ERSAREQ	IGNSF	RV1	RV0	FDFD	FSFD
	W								

MM9Z1_638

Table 535. FTMRC128K1 register summary (continued)

Address & Name		7	6	5	4	3	2	1	0
0x0005 FERCNFG	R	0	0	0	0	0	0	0	SFDIE
	W								
0x0006 FSTAT	R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT1	MGSTAT0
	W								
0x0007 FERSTAT	R	0	0	0	0	0	0	DFDIF	SFDIF
	W								
0x0008 FPROT	R	FPOPEN	RNV6	FPHDIS	FPHS1	FPHS0	FPLDIS	FPLS1	FPLS0
	W								
0x0009 DFPROT	R	DPOPEN	DPS6	DPS5	DPS4	DPS3	DPS2	DPS1	DPS0
	W								
0x000A FOPT	R	NV7	NV6	NV5	NV4	NV3	NV2	NV1	NV0
	W								
0x000B FRSV1	R	0	0	0	0	0	0	0	0
	W								
0x000C FCCOB0HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000D FCCOB0LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x000E FCCOB1HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x000F FCCOB1LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x0010 FCCOB2HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x0011 FCCOB2LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x0012 FCCOB3HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x0013 FCCOB3LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x0014 FCCOB4HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x0015 FCCOB4LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								
0x0016 FCCOB5HI	R	CCOB15	CCOB14	CCOB13	CCOB12	CCOB11	CCOB10	CCOB9	CCOB8
	W								
0x0017 FCCOB5LO	R	CCOB7	CCOB6	CCOB5	CCOB4	CCOB3	CCOB2	CCOB1	CCOB0
	W								

= Unimplemented or Reserved

6.14.3.2.1 Flash clock divider register (FCLKDIV)

The FCLKDIV register is used to control timed events in program and erase algorithms.

Table 536. Flash clock divider register (FCLKDIV)

Offset **Module Base + 0x0000**

	7	6	5	4	3	2	1	0
R	FDIVLD	FDIVLCK	FDIV[5:0]					
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

All bits in the FCLKDIV register are readable, bit 7 is not writable, bit 6 is write-once-hi and controls the writability of the FDIV field in normal mode. In Special mode, bits 6-0 are writable any number of times but bit 7 remains unwritable.

Note

The FCLKDIV register should never be written while a Flash command is executing (CCIF=0).

Table 537. FCLKDIV field descriptions

Field	Description
7 FDIVLD	Clock Divider Loaded 0 FCLKDIV register has not been written since the last reset 1 FCLKDIV register has been written since the last reset
6 FDIVLCK	Clock Divider Locked 0 FDIV field is open for writing 1 FDIV value is locked and cannot be changed. Once the lock bit is set high, only reset can clear this bit and restore writability to the FDIV field in normal mode.
5–0 FDIV[5:0]	**Clock Divider Bits** — FDIV[5:0] must be set to effectively divide BUSCLK down to 1.0 MHz to control timed events during Flash program and erase algorithms. Table 538 shows recommended values for FDIV[5:0] based on the BUSCLK frequency. Refer to Flash command operations," for more information.

Table 538. FDIV values for various BUSCLK frequencies

BUSCLK Frequency (MHz)		FDIV[5:0]	BUSCLK Frequency (MHz)		FDIV[5:0]
MIN (347)	MAX (348)		MIN (347)	MAX (348)	
1.0	1.6	0x00	26.6	27.6	0x1A
1.6	2.6	0x01	27.6	28.6	0x1B
2.6	3.6	0x02	28.6	29.6	0x1C
3.6	4.6	0x03	29.6	30.6	0x1D
4.6	5.6	0x04	30.6	31.6	0x1E
5.6	6.6	0x05	31.6	32.6	0x1F
6.6	7.6	0x06	32.6	33.6	0x20
7.6	8.6	0x07	33.6	34.6	0x21
8.6	9.6	0x08	34.6	35.6	0x22
9.6	10.6	0x09	35.6	36.6	0x23
10.6	11.6	0x0A	36.6	37.6	0x24
11.6	12.6	0x0B	37.6	38.6	0x25
12.6	13.6	0x0C	38.6	39.6	0x26
13.6	14.6	0x0D	39.6	40.6	0x27
14.6	15.6	0x0E	40.6	41.6	0x28
15.6	16.6	0x0F	41.6	42.6	0x29
16.6	17.6	0x10	42.6	43.6	0x2A

Table 538. FDIV values for various BUSCLK frequencies (continued)

BUSCLK Frequency (MHz)		FDIV[5:0]	BUSCLK Frequency (MHz)		FDIV[5:0]
MIN (347)	MAX (348)		MIN (347)	MAX (348)	
17.6	18.6	0x11	43.6	44.6	0x2B
18.6	19.6	0x12	44.6	45.6	0x2C
19.6	20.6	0x13	45.6	46.6	0x2D
20.6	21.6	0x14	46.6	47.6	0x2E
21.6	22.6	0x15	47.6	48.6	0x2F
22.6	23.6	0x16	48.6	49.6	0x30
23.6	24.6	0x17	49.6	50.6	0x31
24.6	25.6	0x18			
25.6	26.6	0x19			

Notes:
347.BUSCLK is Greater Than this value.
348.BUSCLK is Less Than or Equal to this value.

6.14.3.2.2 Flash security register (FSEC)

The FSEC register holds all bits associated with the security of the MCU and Flash module.

Table 539. Flash security register (FSEC)

Offset **Module Base + 0x0001**

	7	6	5	4	3	2	1	0
R	KEYEN[1:0]		RNV[5:2]				SEC[1:0]	
W								
Reset	F (349)	F (349)	F (349)	F (349)	F (349)	F (349)	F (349)	F (349)

= Unimplemented or Reserved

Notes:
349.Loaded from Flash configuration field, during reset sequence.

All bits in the FSEC register are readable but not writable.

During the reset sequence, the FSEC register is loaded with the contents of the Flash security byte in the Flash configuration field at global address 0xFF_FE0F located in P-Flash memory (see Table 532) as indicated by reset condition F in Table 539. If a double bit fault is detected while reading the P-Flash phrase containing the Flash security byte during the reset sequence, all bits in the FSEC register will be set to leave the Flash module in a secured state with backdoor key access disabled.

Table 540. FSEC field descriptions

Field	Description
7–6 KEYEN[1:0]	**Backdoor Key Security Enable Bits** — The KEYEN[1:0] bits define the enabling of backdoor key access to the Flash module as shown in Table 541.
5–2 RNV[5:2]	**Reserved Nonvolatile Bits** — The RNV bits should remain in the erased state for future enhancements.
1–0 SEC[1:0]	**Flash Security Bits** — The SEC[1:0] bits define the security state of the MCU as shown in Table 542. If the Flash module is unsecured using backdoor key access, the SEC bits are forced to 10.

Table 541. Flash KEYEN states

KEYEN[1:0]	Status of Backdoor Key Access
00	DISABLED
01	DISABLED (350)
10	ENABLED
11	DISABLED

Notes:
350.Preferred KEYEN state to disable backdoor key access.

Table 542. Flash security states

SEC[1:0]	Status of Security
00	SECURED
01	SECURED (351)
10	UNSECURED
11	SECURED

Notes:
351.Preferred SEC state to set MCU to secured state.

The security function in the Flash module is described in Security.

6.14.3.2.3 Flash CCOB index register (FCCOBIX)

The FCCOBIX register is used to indicate the amount of parameters loaded into the FCCOB registers for Flash memory operations.

Table 543. FCCOB index register (FCCOBIX)

Offset	Module Base + 0x0002							
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CCOBIX[2:0]		
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

CCOBIX bits are readable and writable while remaining bits read 0 and are not writable.

Table 544. FCCOBIX field descriptions

Field	Description
2–0 CCOBIX[1:0]	**Common Command Register Index**— The CCOBIX bits are used to indicate how many words of the FCCOB register array are being read or written to. See Flash common command object registers (FCCOB)," for more details.

6.14.3.2.4 Flash protection status register (FPSTAT)

This Flash register holds the status of the Protection Override feature.

Table 545. Flash protection status register (FPSTAT)

Offset	Module Base + 0x0003							
	7	6	5	4	3	2	1	0
R	FPOVRD	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

All bits in the FPSTAT register are readable but are not writable.

Table 546. FPSTAT field descriptions

Field	Description
7 FPOVRD	**Flash Protection Override Status** — The FPOVRD bit indicates if the Protection Override feature is currently enabled. See Protection override command" for more details. 0 Protection is not overridden 1 Protection is overridden, contents of registers FPROT and/or DFPROT (and effective protection limits determined by their current contents) were determined during execution of command Protection Override

6.14.3.2.5 Flash configuration register (FCNFG)

The FCNFG register enables the Flash command complete interrupt, control generation of wait-states and forces ECC faults on Flash array read access from the CPU.

Table 547. Flash configuration register (FCNFG)

Offset **Module Base + 0x0004**

	7	6	5	4	3	2	1	0
R	CCIE	0	ERSAREQ	IGNSF	0	0	FDFD	FSFD
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

CCIE, IGNSF, FDFD, and FSFD bits are readable and writable, while remaining bits read 0 and are not writable.

Table 548. FCNFG field descriptions

Field	Description
7 CCIE	**Command Complete Interrupt Enable** — The CCIE bit controls interrupt generation when a Flash command has completed. 0 Command complete interrupt disabled 1 An interrupt will be requested whenever the CCIF flag in the FSTAT register is set (see Flash status register (FSTAT))
5 ERSAREQ	**Erase All Request** — Requests the Memory Controller to execute the Erase All Blocks command and release security. ERSAREQ is not directly writable but is under indirect user control. Refer to the Reference Manual for assertion of the *soc_erase_all_req* input to the FTMRZ module. 0 No request or request complete 1 Request to: a) run the Erase All Blocks command b) verify the erased state c) program the security byte in the Flash Configuration Field to the unsecure state d) release MCU security by setting the SEC field of the FSEC register to the unsecure state as defined in Table 540 of Flash security register (FSEC). The ERSAREQ bit sets to 1 when *soc_erase_all_req* is asserted, CCIF=1 and the Memory Controller starts executing the sequence. ERSAREQ will be reset to 0 by the Memory Controller when the operation is completed (see Erase all pin).
4 IGNSF	**Ignore Single Bit Fault** — The IGNSF controls single bit fault reporting in the FERSTAT register (see Flash error status register (FERSTAT)). 0 All single bit faults detected during array reads are reported 1 Single bit faults detected during array reads are not reported and the single bit fault interrupt will not be generated
1 FDFD	**Force Double Bit Fault Detect** — The FDFD bit allows the user to simulate a double bit fault during Flash array read operations and check the associated interrupt routine. The FDFD bit is cleared by writing a 0 to FDFD. 0 Flash array read operations will set the DFDIF flag in the FERSTAT register only if a double bit fault is detected 1 Any Flash array read operation will force the DFDIF flag in the FERSTAT register to be set (see Flash status register (FSTAT)) and an interrupt will be generated as long as the DFDIE interrupt enable in the FERCNFG register is set (see Flash error configuration register (FERCNFG))
0 FSFD	**Force Single Bit Fault Detect** — The FSFD bit allows the user to simulate a single bit fault during Flash array read operations and check the associated interrupt routine. The FSFD bit is cleared by writing a 0 to FSFD. 0 Flash array read operations will set the SFDIF flag in the FERSTAT register only if a single bit fault is detected 1 Flash array read operation will force the SFDIF flag in the FERSTAT register to be set (see Flash status register (FSTAT)) and an interrupt will be generated as long as the SFDIE interrupt enable in the FERCNFG register is set (see Flash error configuration register (FERCNFG))

6.14.3.2.6 Flash error configuration register (FERCNFG)

The FERCNFG register enables the Flash error interrupts for the FERSTAT flags.

Table 549. Flash error configuration register (FERCNFG)

Offset	Module Base + 0x0005							
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	SFDIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

All assigned bits in the FERCNFG register are readable and writable.

Table 550. FERCNFG field descriptions

Field	Description
0 SFDIE	**Single Bit Fault Detect Interrupt Enable** — The SFDIE bit controls interrupt generation when a single bit fault is detected during a Flash block read operation. 0 SFDIF interrupt disabled whenever the SFDIF flag is set (see Flash error status register (FERSTAT)) 1 An interrupt will be requested whenever the SFDIF flag is set (see Flash error status register (FERSTAT))

6.14.3.2.7 Flash status register (FSTAT)

The FSTAT register reports the operational status of the Flash module.

Table 551. Flash status register (FSTAT)

Offset	Module Base + 0x0006							
	7	6	5	4	3	2	1	0
R	CCIF	0	ACCERR	FPVIOL	MGBUSY	RSVD	MGSTAT[1:0]	
W								
Reset	1	0	0	0	0	0	0 (352)	0 (352)

= Unimplemented or Reserved

Notes:
352.Reset value can deviate from the value shown if a double bit fault is detected during the reset sequence (see Initialization).

CCIF, ACCERR, and FPVIOL bits are readable and writable, MGBUSY and MGSTAT bits are readable but not writable, while remaining bits read 0 and are not writable.

Table 5. FSTAT field descriptions

Field	Description
7 CCIF	**Command Complete Interrupt Flag** — The CCIF flag indicates that a Flash command has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command and CCIF will stay low until command completion or command violation. 0 Flash command in progress 1 Flash command has completed
5 ACCERR	**Flash Access Error Flag** — The ACCERR bit indicates an illegal access has occurred to the Flash memory caused by either a violation of the command write sequence (see Command write sequence) or issuing an illegal Flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to ACCERR. Writing a 0 to the ACCERR bit has no effect on ACCERR. 0 No access error detected 1 Access error detected

Table 5. FSTAT field descriptions (continued)

Field	Description
4 FPVIOL	**Flash Protection Violation Flag** —The FPVIOL bit indicates an attempt was made to program or erase an address in a protected area of P-Flash or EEPROM memory during a command write sequence. The FPVIOL bit is cleared by writing a 1 to FPVIOL. Writing a 0 to the FPVIOL bit has no effect on FPVIOL. While FPVIOL is set, it is not possible to launch a command or start a command write sequence. 0 No protection violation detected 1 Protection violation detected
3 MGBUSY	**Memory Controller Busy Flag** — The MGBUSY flag reflects the active state of the Memory Controller. 0 Memory Controller is idle 1 Memory Controller is busy executing a Flash command (CCIF = 0)
2 RSVD	**Reserved Bit** — This bit is reserved and always reads 0.
1–0 MGSTAT[1:0]	**Memory Controller Command Completion Status Flag** — One or more MGSTAT flag bits are set if an error is detected during execution of a Flash command or during the Flash reset sequence. See Flash command description," and Initialization" for details.

6.14.3.2.8 Flash error status register (FERSTAT)

The FERSTAT register reflects the error status of internal Flash operations.

Table 552. Flash error status register (FERSTAT)

Offset **Module Base + 0x0007**

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	DFDIF	SFDIF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

All flags in the FERSTAT register are readable and only writable to clear the flag.

Table 553. FERSTAT field descriptions

Field	Description
1 DFDIF	**Double Bit Fault Detect Interrupt Flag** — The setting of the DFDIF flag indicates that a double bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation.(353) The DFDIF flag is cleared by writing a 1 to DFDIF. Writing a 0 to DFDIF has no effect on DFDIF.(354) 0 No double bit fault detected 1 Double bit fault detected or a Flash array read operation returning invalid data was attempted while command running
0 SFDIF	**Single Bit Fault Detect Interrupt Flag** — With the IGNSF bit in the FCNFG register clear, the SFDIF flag indicates that a single bit fault was detected in the stored parity and data bits during a Flash array read operation or that a Flash array read operation returning invalid data was attempted on a Flash block that was under a Flash command operation.(353) The SFDIF flag is cleared by writing a 1 to SFDIF. Writing a 0 to SFDIF has no effect on SFDIF. 0 No single bit fault detected 1 Single bit fault detected and corrected or a Flash array read operation returning invalid data was attempted while command running

Notes:

353. The single bit fault and double bit fault flags are mutually exclusive for parity errors (an ECC fault occurrence can be either single fault or double fault but never both). A simultaneous access collision (Flash array read operation returning invalid data attempted while command running) is indicated when both SFDIF and DFDIF flags are high.

354. There is a one cycle delay in storing the ECC DFDIF and SFDIF fault flags in this register. At least one NOP is required after a flash memory read before checking FERSTAT for the occurrence of ECC errors.

6.14.3.2.9 P-flash protection register (FPROT)

The FPROT register defines which P-Flash sectors are protected against program and erase operations.

Table 554. Flash protection register (FPROT)

Offset	Module Base + 0x0008							
	7	6	5	4	3	2	1	0
R	FPOPEN	RNV6	FPHDIS	FPHS[1:0]		FPLDIS	FPLS[1:0]	
W								
Reset	F (355)	F (355)	F (355)	F (355)	F (355)	F (355)	F (355)	F (355)

= Unimplemented or Reserved

Notes:
355.Loaded from Flash configuration field, during reset sequence.

The (unreserved) bits of the FPROT register are writable with the restriction that the size of the protected region can only be increased (see P-flash protection restrictions," and Table 559)

During the reset sequence, the FPROT register is loaded with the contents of the P-Flash protection byte in the Flash configuration field at global address 0xFF_FE0C located in P-Flash memory (see Table 532) as indicated by reset condition 'F' in Table 554. To change the P-Flash protection that will be loaded during the reset sequence, the upper sector of the P-Flash memory must be unprotected, then the P-Flash protection byte must be reprogrammed. If a double bit fault is detected while reading the P-Flash phrase containing the P-Flash protection byte during the reset sequence, the FPOPEN bit will be cleared and remaining bits in the FPROT register will be set to leave the P-Flash memory fully protected.

Trying to alter data in any protected area in the P-Flash memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. The block erase of a P-Flash block is not possible if any of the P-Flash sectors contained in the same P-Flash block are protected.

Table 555. FPROT field descriptions

Field	Description
7 FPOPEN	**Flash Protection Operation Enable** — The FPOPEN bit determines the protection function for program or erase operations as shown in Table 556 for the P-Flash block. 0 When FPOPEN is clear, the FPHDIS and FPLDIS bits define unprotected address ranges as specified by the corresponding FPHS and FPLS bits 1 When FPOPEN is set, the FPHDIS and FPLDIS bits enable protection for the address range specified by the corresponding FPHS and FPLS bits
6 RNV[6]	**Reserved Nonvolatile Bit** — The RNV bit should remain in the erased state for future enhancements.
5 FPHDIS	**Flash Protection Higher Address Range Disable** — The FPHDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory ending with global address 0xFF_FFFF. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
4–3 FPHS[1:0]	**Flash Protection Higher Address Size** — The FPHS bits determine the size of the protected/unprotected area in P-Flash memory as shown inTable 557. The FPHS bits can only be written to while the FPHDIS bit is set.
2 FPLDIS	**Flash Protection Lower Address Range Disable** — The FPLDIS bit determines whether there is a protected/unprotected area in a specific region of the P-Flash memory beginning with global address 0xFF_8000. 0 Protection/Unprotection enabled 1 Protection/Unprotection disabled
1–0 FPLS[1:0]	**Flash Protection Lower Address Size** — The FPLS bits determine the size of the protected/unprotected area in P-Flash memory as shown in Table 558. The FPLS bits can only be written to while the FPLDIS bit is set.

Table 556. P-flash protection function

FPOPEN	FPHDIS	FPLDIS	Function (356)
1	1	1	No P-Flash Protection
1	1	0	Protected Low Range
1	0	1	Protected High Range
1	0	0	Protected High and Low Ranges
0	1	1	Full P-Flash Memory Protected
0	1	0	Unprotected Low Range
0	0	1	Unprotected High Range
0	0	0	Unprotected High and Low Ranges

Notes:
356.For range sizes, refer to Table 557 and Table 558.

Table 557. P-flash protection higher address range

FPHS[1:0]	Global Address Range	Protected Size
00	0xFF_F800–0xFF_FFFF	2.0 kbytes
01	0xFF_F000–0xFF_FFFF	4.0 kbytes
10	0xFF_E000–0xFF_FFFF	8.0 kbytes
11	0xFF_C000–0xFF_FFFF	16 kbytes

Table 558. P-flash protection lower address range

FPLS[1:0]	Global Address Range	Protected Size
00	0xFF_8000–0xFF_83FF	1.0 kbyte
01	0xFF_8000–0xFF_87FF	2.0 kbytes
10	0xFF_8000–0xFF_8FFF	4.0 kbytes
11	0xFF_8000–0xFF_9FFF	8.0 kbytes

All possible P-Flash protection scenarios are shown in Figure 101 . Although the protection scheme is loaded from the Flash memory at global address 0xFF_FE0C during the reset sequence, it can be changed by the user. The P-Flash protection scheme can be used by applications requiring reprogramming in single chip mode while providing as much protection as possible if reprogramming is not required.

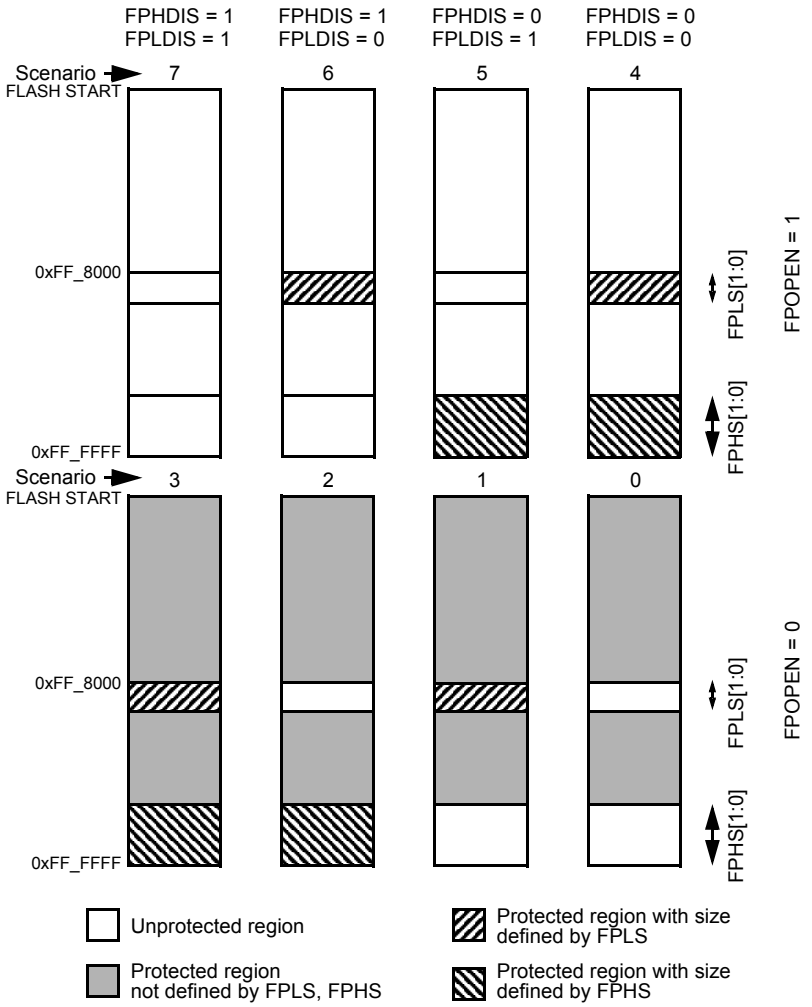


Figure 101. P-flash protection scenarios

6.14.3.2.9.1 P-flash protection restrictions

The general guideline is that P-Flash protection can only be added and not removed. Table 559 specifies all valid transitions between P-Flash protection scenarios. Any attempt to write an invalid scenario to the FPROT register will be ignored. The contents of the FPROT register reflect the active protection scenario. See the FPHS and FPLS bit descriptions for additional restrictions.

Table 559. P-flash protection scenario transitions

From Protection Scenario	To Protection Scenario[357]							
	0	1	2	3	4	5	6	7
0	X	X	X	X				
1		X		X				
2			X	X				
3				X				
4				X	X			
5			X	X	X	X		
6		X		X	X		X	

Table 559. P-flash protection scenario transitions (continued)

From Protection Scenario	To Protection Scenario[357]							
	0	1	2	3	4	5	6	7
7	X	X	X	X	X	X	X	X

Notes:
357.Allowed transitions marked with X, see Figure 101 for a definition of the scenarios.

6.14.3.2.10 EEPROM protection register (DFPROT)

The DFPROT register defines which EEPROM sectors are protected against program and erase operations

Table 560. EEPROM protection register (DFPROT)

Offset	Module Base + 0x0009							
	7	6	5	4	3	2	1	0
R W	DPOPEN	DPS[6:0]						
Reset	F [358]	F [358]	F [358]	F [358]	F [358]	F [358]	F [358]	F [358]

Notes:
358.Loaded from Flash configuration field, during reset sequence.

The (unreserved) bits of the DFPROT register are writable with the restriction that protection can be added but not removed. Writes must increase the DPS value and the DPOPEN bit can only be written from 1 (protection disabled) to 0 (protection enabled). If the DPOPEN bit is set, the state of the DPS bits is irrelevant.

During the reset sequence, fields DPOPEN and DPS of the DFPROT register are loaded with the contents of the EEPROM protection byte in the Flash configuration field at global address 0xFF_FE0D located in P-Flash memory (see Table 532) as indicated by reset condition F in. To change the EEPROM protection that will be loaded during the reset sequence, the P-Flash sector containing the EEPROM protection byte must be unprotected, then the EEPROM protection byte must be programmed. If a double bit fault is detected while reading the P-Flash phrase containing the EEPROM protection byte during the reset sequence, the DPOPEN bit will be cleared and DPS bits will be set to leave the EEPROM memory fully protected.

Trying to alter data in any protected area in the EEPROM memory will result in a protection violation error and the FPVIOL bit will be set in the FSTAT register. Block erase of the EEPROM memory is not possible if any of the EEPROM sectors are protected.

Table 561. DFPROT field descriptions

Field	Description
7 DPOPEN	EEPROM Protection Control 0 Enables EEPROM memory protection from program and erase with protected address range defined by DPS bits 1 Disables EEPROM memory protection from program and erase
6–0 DPS[6:0]	**EEPROM Protection Size** — The DPS[6:0] bits determine the size of the protected area in the EEPROM memory, this size increase in step of 32 bytes, as shown in Table 562 .

Table 562. EEPROM protection address range

DPS[6:0]	Global Address Range	Protected Size
0000000	0x10_0000 – 0x10_001F	32 bytes
0000001	0x10_0000 – 0x10_003F	64 bytes
0000010	0x10_0000 – 0x10_005F	96 bytes
0000011	0x10_0000 – 0x10_007F	128 bytes
0000100	0x10_0000 – 0x10_009F	160 bytes
0000101	0x10_0000 – 0x10_00BF	192 bytes

Table 562. EEPROM protection address range (continued)

DPS[6:0]	Global Address Range	Protected Size
The Protection Size goes on enlarging in step of 32 bytes, for each DPS value increasing of one. . . .		
1111111	0x10_0000 – 0x10_0FFF	4,096 bytes

6.14.3.2.11 Flash option register (FOPT)

The FOPT register is the Flash option register.

Table 563. Flash option register (FOPT)

Offset **Module Base + 0x000A**

	7	6	5	4	3	2	1	0
R	NV[7:0]							
W								
Reset	F (359)	F (359)	F (359)	F (359)	F (359)	F (359)	F (359)	F (359)

= Unimplemented or Reserved

Notes:
359.Loaded from Flash configuration field, during reset sequence.

All bits in the FOPT register are readable but are not writable.

During the reset sequence, the FOPT register is loaded from the Flash nonvolatile byte in the Flash configuration field at global address 0xFF_FE0E located in P-Flash memory (see Table 532) as indicated by reset condition F in Table 563. If a double bit fault is detected while reading the P-Flash phrase containing the Flash nonvolatile byte during the reset sequence, all bits in the FOPT register will be set.

Table 564. FOPT field descriptions

Field	Description
7–0 NV[7:0]	**Nonvolatile Bits** — The NV[7:0] bits are available as nonvolatile bits. Refer to the device user guide for proper use of the NV bits.

6.14.3.2.12 Flash reserved1 register (FRSV1)

This Flash register is reserved for factory testing.

Table 565. Flash reserved1 register (FRSV1)

Offset **Module Base + 0x000B**

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

All bits in the FRSV1 register read 0 and are not writable.

6.14.3.2.13 Flash common command object registers (FCCOB)

The FCCOB is an array of six words. Byte wide reads and writes are allowed to the FCCOB registers.

Table 566. Flash common command object 0 high register (FCCOB0HI)

Offset	Module Base + 0x000C							
	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 567. Flash common command object 0 low register (FCCOB0LO)

Offset	Module Base + 0x000D							
	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

Table 568. Flash common command object 1 high register (FCCOB1HI)

Offset	Module Base + 0x000E							
	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 569. Flash common command object 1 low register (FCCOB1LO)

Offset	Module Base + 0x000F							
	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

Table 570. Flash common command object 2 high register (FCCOB2HI)

Offset	Module Base + 0x0010							
	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 571. Flash common command object 2 low register (FCCOB2LO)

Offset	Module Base + 0x0011							
	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

Table 572. Flash common command object 3 high register (FCCOB3HI)

Offset	Module Base + 0x0012							
	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 573. Flash common command object 3 low register (FCCOB3LO)

Offset	Module Base + 0x0013							
	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

Table 574. Flash common command object 4 high register (FCCOB4HI)

Offset	Module Base + 0x0014							
	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 575. Flash common command object 4 low register (FCCOB4LO)

Offset	Module Base + 0x0015							
	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

Table 576. Flash common command object 5 high register (FCCOB5HI)

Offset	Module Base + 0x0016							
	7	6	5	4	3	2	1	0
R W	CCOB[15:8]							
Reset	0	0	0	0	0	0	0	0

Table 577. Flash common command object 5 low register (FCCOB5LO)

Offset	Module Base + 0x0017							
	7	6	5	4	3	2	1	0
R W	CCOB[7:0]							
Reset	0	0	0	0	0	0	0	0

6.14.3.2.13.1 FCCOB - NVM command mode

NVM command mode uses the FCCOB registers to provide a command code and its relevant parameters to the Memory Controller. The user first sets up all required FCCOB fields and then initiates the command's execution by writing a 1 to the CCIF bit in the FSTAT register (a 1 written by the user clears the CCIF command completion flag to 0). When the user clears the CCIF bit in the FSTAT register all FCCOB parameter fields are locked and cannot be changed by the user until the command completes (as evidenced by the Memory Controller returning CCIF to 1). Some commands return information to the FCCOB register array.

The generic format for the FCCOB parameter fields in NVM command mode is shown in Table 578. The return values are available for reading after the CCIF flag in the FSTAT register has been returned to 1 by the Memory Controller. The value written to the FCCOBIX field must reflect the amount of CCOB words loaded for command execution.

Table 578 shows the generic Flash command format. The high byte of the first word in the CCOB array contains the command code, followed by the parameters for this specific Flash command. For details on the FCCOB settings required by each command, see the Flash command descriptions in Flash command description.

Table 578. FCCOB - NVM command mode (typical usage)

CCOBIX[2:0]	Register	Byte	FCCOB Parameter Fields (NVM Command Mode)
000	FCCOB0	HI	FCMD[7:0] defining Flash command
		LO	Global address [23:16]
001	FCCOB1	HI	Global address [15:8]
		LO	Global address [7:0]
010	FCCOB2	HI	Data 0 [15:8]
		LO	Data 0 [7:0]
011	FCCOB3	HI	Data 1 [15:8]
		LO	Data 1 [7:0]
100	FCCOB4	HI	Data 2 [15:8]
		LO	Data 2 [7:0]
101	FCCOB5	HI	Data 3 [15:8]
		LO	Data 3 [7:0]

6.14.4 Functional description

6.14.4.1 Modes of operation

The FTMRC128K1 module provides the modes of operation normal and special . The operating mode is determined by module-level inputs and affects the FCLKDIV, FCNFG, and DFPROT registers (see 580).

6.14.4.2 IFR version ID word

The version ID word is stored in the IFR at address 0x1F_C0B6. The contents of the word are defined in Table 579.

Table 579. IFR version ID fields

[15:4]	[3:0]
Reserved	VERNUM

- VERNUM: Version number. The first version is number 0b_0001 with both 0b_0000 and 0b_1111 meaning 'none'.

6.14.4.3 Internal NVM resource (NVMRES)

IFR is an internal NVM resource readable by CPU, when NVMRES is active. The IFR fields are shown in Table 533.

The NVMRES global address map is shown in Table 534.

6.14.4.4 Flash command operations

Flash command operations are used to modify Flash memory contents.

The next sections describe:

- How to write the FCLKDIV register that is used to generate a time base (FCLK) derived from BUSCLK for Flash program and erase command operations
- The command write sequence used to set Flash command parameters and launch execution
- Valid Flash commands available for execution, according to MCU functional mode and MCU security state.

6.14.4.4.1 Writing the FCLKDIV register

Prior to issuing any Flash program or erase command after a reset, the user is required to write the FCLKDIV register to divide BUSCLK down to a target FCLK of 1 MHz. Table 538 shows recommended values for the FDIV field based on BUSCLK frequency.

CAUTION

Programming or erasing the Flash memory cannot be performed if the bus clock runs at less than 0.8 MHz. Setting FDIV too high can destroy the Flash memory due to overstress. Setting FDIV too low can result in incomplete programming or erasure of the Flash memory cells.

When the FCLKDIV register is written, the FDIVLD bit is set automatically. If the FDIVLD bit is 0, the FCLKDIV register has not been written since the last reset. If the FCLKDIV register has not been written, any Flash program or erase command loaded during a command write sequence will not execute and the ACCERR bit in the FSTAT register will set.

6.14.4.4.2 Command write sequence

The Memory Controller will launch all valid Flash commands entered using a command write sequence.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be clear (see Flash status register (FSTAT)) and the CCIF flag should be tested to determine the status of the current command write sequence. If CCIF is 0, the previous command write sequence is still active, a new command write sequence cannot be started, and all writes to the FCCOB register are ignored.

6.14.4.4.3 Define FCCOB contents

The FCCOB parameter fields must be loaded with all required parameters for the Flash command being executed. The CCOBIX bits in the FCCOBIX register must reflect the amount of words loaded into the FCCOB registers (see Flash CCOB index register (FCCOBIX)).

The contents of the FCCOB parameter fields are transferred to the Memory Controller when the user clears the CCIF command completion flag in the FSTAT register (writing 1 clears the CCIF to 0). The CCIF flag will remain clear until the Flash command has completed. Upon completion, the Memory Controller will return CCIF to 1 and the FCCOB register will be used to communicate any results. The flow for a generic command write sequence is shown in Figure 102.

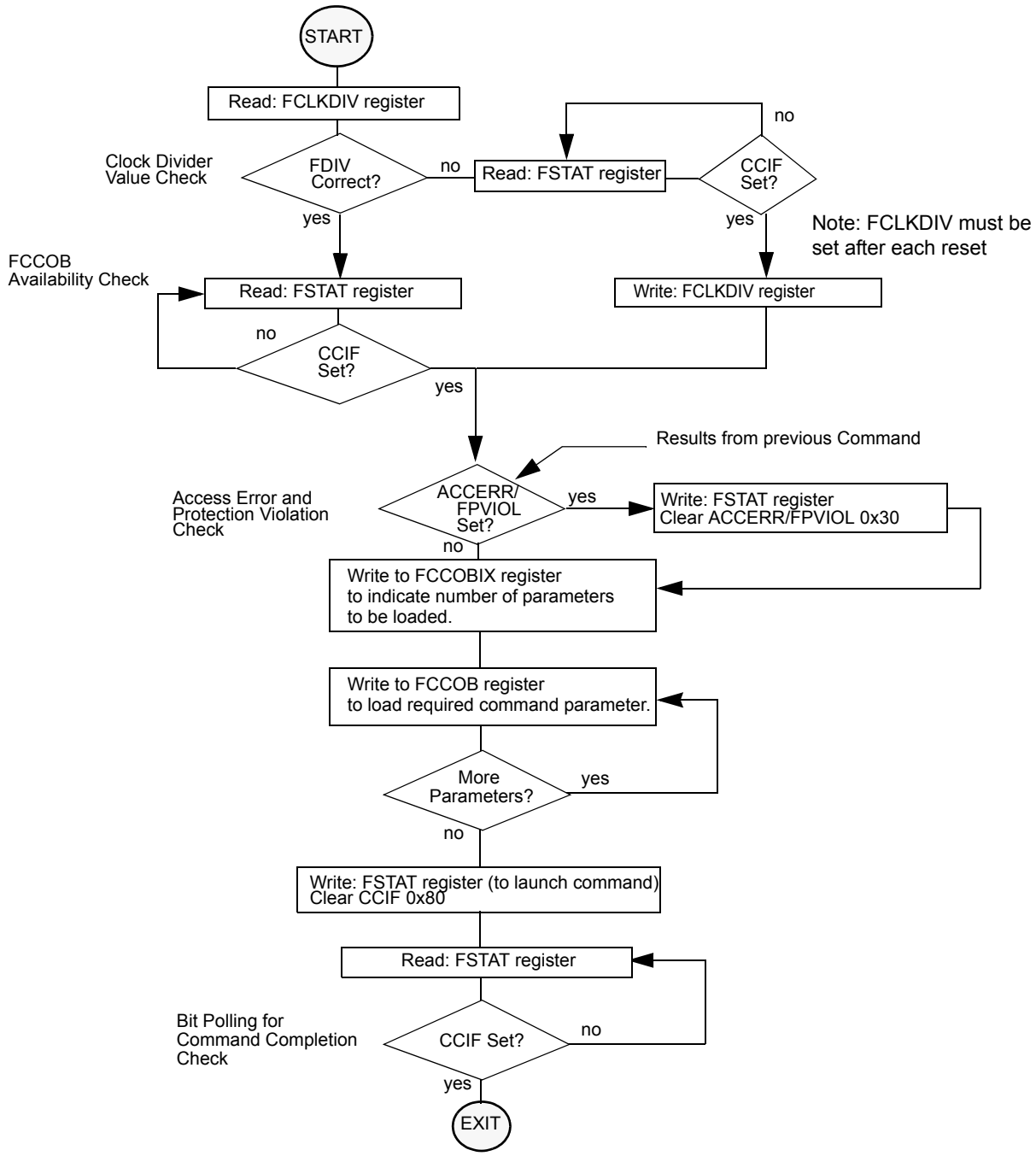


Figure 102. Generic flash command write sequence flowchart

6.14.4.4.4 Valid flash module commands

Table 580 presents the valid Flash commands, as enabled by the combination of the functional MCU mode (Normal SingleChip NS, Special Singlechip SS) with the MCU security state (Unsecured, Secured).

Special Singlechip mode is selected by input mmc_ss_mode_ts2 asserted. MCU Secured state is selected by input mmc_secure input asserted.

Table 580. Flash commands by mode and security state

FCMD	Command	Unsecured		Secured	
		NS (360)	SS (361)	NS (362)	SS (363)
0x01	Erase Verify All Blocks	*	*	*	*
0x02	Erase Verify Block	*	*	*	*
0x03	Erase Verify P-Flash Section	*	*	*	
0x04	Read Once	*	*	*	
0x06	Program P-Flash	*	*	*	
0x07	Program Once	*	*	*	
0x08	Erase All Blocks		*		*
0x09	Erase Flash Block	*	*	*	
0x0A	Erase P-Flash Sector	*	*	*	
0x0B	Unsecure Flash		*		*
0x0C	Verify Backdoor Access Key	*		*	
0x0D	Set User Margin Level	*	*	*	
0x0E	Set Field Margin Level		*		
0x10	Erase Verify EEPROM Section	*	*	*	
0x11	Program EEPROM	*	*	*	
0x12	Erase EEPROM Sector	*	*	*	
0x13	Protection Override	*	*	*	*

Notes:
360.Unsecured Normal Single Chip mode
361.Unsecured Special Single Chip mode.
362.Secured Normal Single Chip mode.
363.Secured Special Single Chip mode.

6.14.4.4.5 P-flash commands

Table 581 summarizes the valid P-Flash commands along with the effects of the commands on the P-Flash block and other resources within the Flash module.

Table 581. P-flash commands

FCMD	Command	Function on P-Flash Memory
0x01	Erase Verify All Blocks	Verify that all P-Flash (and EEPROM) blocks are erased.
0x02	Erase Verify Block	Verify that a P-Flash block is erased.
0x03	Erase Verify P-Flash Section	Verify that a given number of words starting at the address provided are erased.
0x04	Read Once	Read a dedicated 64 byte field in the nonvolatile information register in P-Flash block that was previously programmed using the Program Once command.
0x06	Program P-Flash	Program a phrase in a P-Flash block.
0x07	Program Once	Program a dedicated 64 byte field in the nonvolatile information register in P-Flash block that is allowed to be programmed only once.
0x08	Erase All Blocks	Erase all P-Flash (and EEPROM) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a P-Flash (or EEPROM) block. An erase of the full P-Flash block is only possible when FPLDIS, FPHDIS and FPOPEN bits in the FPROT register are set prior to launching the command.
0x0A	Erase P-Flash Sector	Erase all bytes in a P-Flash sector.

Table 581. P-flash commands (continued)

FCMD	Command	Function on P-Flash Memory
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all P-Flash (and EEPROM) blocks and verifying that all P-Flash (and EEPROM) blocks are erased.
0x0C	Verify Backdoor Access Key	Supports a method of releasing MCU security by verifying a set of security keys.
0x0D	Set User Margin Level	Specifies a user margin read level for all P-Flash blocks.
0x0E	Set Field Margin Level	Specifies a field margin read level for all P-Flash blocks (special modes only).
0x13	Protection Override	Supports a mode to temporarily override Protection configuration (for P-Flash and/or EEPROM) by verifying a key.

6.14.4.4.6 EEPROM commands

Table 582 summarizes the valid EEPROM commands along with the effects of the commands on the EEPROM block.

Table 582. EEPROM commands

FCMD	Command	Function on EEPROM Memory
0x01	Erase Verify All Blocks	Verify that all EEPROM (and P-Flash) blocks are erased.
0x02	Erase Verify Block	Verify that the EEPROM block is erased.
0x08	Erase All Blocks	Erase all EEPROM (and P-Flash) blocks. An erase of all Flash blocks is only possible when the FPLDIS, FPHDIS, and FPOPEN bits in the FPROT register and the DPOPEN bit in the DFPROT register are set prior to launching the command.
0x09	Erase Flash Block	Erase a EEPROM (or P-Flash) block. An erase of the full EEPROM block is only possible when DPOPEN bit in the DFPROT register is set prior to launching the command.
0x0B	Unsecure Flash	Supports a method of releasing MCU security by erasing all EEPROM (and P-Flash) blocks and verifying that all EEPROM (and P-Flash) blocks are erased.
0x0D	Set User Margin Level	Specifies a user margin read level for the EEPROM block.
0x0E	Set Field Margin Level	Specifies a field margin read level for the EEPROM block (special modes only).
0x10	Erase Verify EEPROM Section	Verify that a given number of words starting at the address provided are erased.
0x11	Program EEPROM	Program up to four words in the EEPROM block.
0x12	Erase EEPROM Sector	Erase all bytes in a sector of the EEPROM block.
0x13	Protection Override	Supports a mode to temporarily override Protection configuration (for P-Flash and/or EEPROM) by verifying a key.

6.14.4.5 Allowed simultaneous p-flash and EEPROM operations

Only the operations marked ‘OK’ in Table 583 are permitted to be run simultaneously on the Program Flash and EEPROM blocks. Some operations cannot be executed simultaneously because certain hardware resources are shared by the two memories. The priority has been placed on permitting Program Flash reads while program and erase operations execute on the EEPROM, providing read (P-Flash) while write (EEPROM) functionality.

Table 583. Allowed P-flash and EEPROM simultaneous operations

Program Flash	EEPROM				
	Read	Margin Read (364)	Program	Sector Erase	Mass Erase (365)
Read		OK	OK	OK	
Margin Read (364)					
Program					
Sector Erase					
Mass Erase (365)					OK

Notes:
364. A 'Margin Read' is any read after executing the margin setting commands 'Set User Margin Level' or 'Set Field Margin Level' with anything but the 'normal' level specified. See the Note on margin settings in Set user margin level command and Set field margin level command.
365. The 'Mass Erase' operations are commands 'Erase All Blocks' and 'Erase Flash Block'

6.14.4.6 Flash command description

This section provides details of all available Flash commands launched by a command write sequence. The ACCERR bit in the FSTAT register will be set during the command write sequence if any of the following illegal steps are performed, causing the command not to be processed by the Memory Controller:

- Starting any command write sequence that programs or erases Flash memory before initializing the FCLKDIV register
- Writing an invalid command as part of the command write sequence
- For additional possible errors, refer to the error handling table provided for each command

If a Flash block is read during execution of an algorithm (CCIF = 0) on that same block, the read operation will return invalid data if both flags SFDIF and DFDIF are set. If the SFDIF or DFDIF flags were not previously set when the invalid read operation occurred, both the SFDIF and DFDIF flags will be set.

If the ACCERR or FPVIOL bits are set in the FSTAT register, the user must clear these bits before starting any command write sequence (see Flash status register (FSTAT)).

Note

A Flash word or phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash word or phrase is not allowed.

6.14.4.6.1 Erase verify all blocks command

The Erase Verify All Blocks command will verify that all P-Flash and EEPROM blocks have been erased.

Table 6. Erase verify all blocks command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x01	Not required

Upon clearing CCIF to launch the Erase Verify All Blocks command, the Memory Controller will verify that the entire Flash memory space is erased. The CCIF flag will set after the Erase Verify All Blocks operation has completed. If all blocks are not erased, it means blank check failed, both MGSTAT bits will be set.

Table 584. Erase verify all blocks command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 000 at command launch
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

6.14.4.6.2 Erase verify block command

The Erase Verify Block command allows the user to verify that an entire P-Flash or EEPROM block has been erased.

Table 585. Erase verify block command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x02	Global address [23:16] to identify Flash block
FCCOB1	Global address [15:0] to identify Flash block	

Upon clearing CCIF to launch the Erase Verify Block command, the Memory Controller will verify that the selected P-Flash or EEPROM block is erased. The CCIF flag will set after the Erase Verify Block operation has completed.If the block is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 586. Erase verify block command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if an invalid global address [23:0] is supplied see Flash status register (FSTAT))
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

6.14.4.6.3 Erase verify P-Flash Section Command

The Erase Verify P-Flash Section command will verify that a section of code in the P-Flash memory is erased. The Erase Verify P-Flash Section command defines the starting point of the code to be verified and the number of phrases.

Table 587. Erase verify P-flash section command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x03	Global address [23:16] of a P-Flash block
FCCOB1	Global address [15:0] of the first phrase to be verified	
FCCOB2	Number of phrases to be verified	

Upon clearing CCIF to launch the Erase Verify P-Flash Section command, the Memory Controller will verify the selected section of Flash memory is erased. The CCIF flag will set after the Erase Verify P-Flash Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 588. Erase verify P-flash section command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 010 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied see Table 580)
		Set if a misaligned phrase address is supplied (global address [2:0]!= 000)
		Set if the requested section crosses a the P-Flash address boundary
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

6.14.4.6.4 Read once command

The Read Once command provides read access to a reserved 64 byte field (8 phrases) located in the nonvolatile information register of P-Flash. The Read Once field is programmed using the Program Once command described in Program once command. The Read Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 589. Read once command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x04	Not Required
FCCOB1	Read Once phrase index (0x0000 - 0x0007)	
FCCOB2	Read Once word 0 value	
FCCOB3	Read Once word 1 value	
FCCOB4	Read Once word 2 value	
FCCOB5	Read Once word 3 value	

Upon clearing CCIF to launch the Read Once command, a Read Once phrase is fetched and stored in the FCCOB indexed register. The CCIF flag will set after the Read Once operation has completed. Valid phrase index values for the Read Once command range from 0x0000 to 0x0007. During execution of the Read Once command, any attempt to read addresses within P-Flash block will return invalid data.

Table 590. Read once command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid phrase index is supplied
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read
	MGSTAT0	Set if any non-correctable errors have been encountered during the read

6.14.4.6.5 Program P-flash command

The Program P-Flash operation will program a previously erased phrase in the P-Flash memory using an embedded algorithm.

Note

A P-Flash phrase must be in the erased state before being programmed. Cumulative programming of bits within a Flash phrase is not allowed.

Table 591. Program P-flash command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x06	Global address [23:16] to identify P-Flash block
FCCOB1	Global address [15:0] of phrase location to be programmed (366)	
FCCOB2	Word 0 program value	
FCCOB3	Word 1 program value	
FCCOB4	Word 2 program value	
FCCOB5	Word 3 program value	

Notes:
366.Global address [2:0] must be 000

Upon clearing CCIF to launch the Program P-Flash command, the Memory Controller will program the data words to the supplied global address and will then proceed to verify the data words read back as expected. The CCIF flag will set after the Program P-Flash operation has completed.

Table 592. Program P-flash command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 101 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0]!= 000)
	FPVIOL	Set if the global address [17:0] points to a protected area
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

6.14.4.6.6 Program once command

The Program Once command restricts programming to a reserved 64 byte field (8 phrases) in the nonvolatile information register located in P-Flash. The Program Once reserved field can be read using the Read Once command as described in Read once command. The Program Once command must only be issued once since the nonvolatile information register in P-Flash cannot be erased. The Program Once command must not be executed from the Flash block containing the Program Once reserved field to avoid code runaway.

Table 593. Program once command FCCOB requirements

CCOBIX[2:0]	FCCOB Parameters	
FCCOB0	0x07	Not Required
FCCOB1	Program Once phrase index (0x0000 - 0x0007)	
FCCOB2	Program Once word 0 value	
FCCOB3	Program Once word 1 value	
FCCOB4	Program Once word 2 value	
FCCOB5	Program Once word 3 value	

Upon clearing CCIF to launch the Program Once command, the Memory Controller first verifies that the selected phrase is erased. If erased, then the selected phrase will be programmed and then verified with read back. The CCIF flag will remain clear, setting only after the Program Once operation has completed.

The reserved nonvolatile information register accessed by the Program Once command cannot be erased and any attempt to program one of these phrases a second time will not be allowed. Valid phrase index values for the Program Once command range from 0x0000 to 0x0007. During execution of the Program Once command, any attempt to read addresses within P-Flash will return invalid data.

Table 594. Program once command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 101 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid phrase index is supplied
		Set if the requested phrase has already been programmed (367)
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

Notes:

367. If a Program Once phrase is initially programmed to 0xFFFF_FFFF_FFFF_FFFF, the Program Once command will be allowed to execute again on that same phrase.

6.14.4.6.7 Erase all blocks command

The Erase All Blocks operation will erase the entire P-Flash and EEPROM memory space.

Table 595. Erase all blocks command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x08	Not required

Upon clearing CCIF to launch the Erase All Blocks command, the Memory Controller will erase the entire Flash memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag will set after the Erase All Blocks operation has completed.

Table 596. Erase all blocks command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 000 at command launch
		Set if command not available in current mode (see Table 580)
	FPVIOL	Set if any area of the P-Flash or EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

6.14.4.6.7.1 Erase all pin

The functionality of the Erase All Blocks command is also available in an uncommanded fashion from the *soc_erase_all_req* input pin on the Flash module. Refer to the Reference Manual for information on control of *soc_erase_all_req*.

The erase-all function requires the clock divider register FCLKDIV (see Flash clock divider register (FCLKDIV)) to be loaded before invoking this function using *soc_erase_all_req* input pin. Refer to the Reference Manual for information about the default value of FCLKDIV in case direct writes to register FCLKDIV are not allowed by the time this feature is invoked. If FCLKDIV is not properly set the erase-all operation will not execute and the ACCERR flag in FSTAT register will set. After the execution of the erase-all function the FCLKDIV register will be reset and the value of register FCLKDIV must be loaded before launching any other command afterwards.

Before invoking the erase-all function using the *soc_erase_all_req* pin, the ACCERR and FPVIOL flags in the FSTAT register must be clear. When invoked from *soc_erase_all_req* the erase-all function will erase all P-Flash memory and EEPROM memory space regardless of the protection settings. If the post-erase verify passes, the routine will then release security by setting the SEC field of the FSEC register to the unsecure state (see Flash security register (FSEC)). The security byte in the Flash Configuration Field will be programmed to the unsecure state (see Table 540). The status of the erase-all request is reflected in the ERSAREQ bit in the FCNFG register (see Flash configuration register (FCNFG)). The ERSAREQ bit in FCNFG will be cleared once the operation has completed and the normal FSTAT error reporting will be available as described inTable 597.

At the end of the erase-all sequence Protection will remain configured as it was before executing the erase-all function. If the application requires programming P-Flash and/or EEPROM after the erase-all function completes, the existing protection limits must be taken into account. If protection needs to be disabled the user may need to reset the system right after completing the erase-all function.

Table 597. Erase all pin error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if command not available in current mode (see Table 580)
	MGSTAT1	Set if any errors have been encountered during the erase verify operation, or during the program verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the erase verify operation, or during the program verify operation

6.14.4.6.8 Erase flash block command

The Erase Flash Block operation will erase all addresses in a P-Flash or EEPROM block.

Table 598. Erase flash block command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x09	Global address [23:16] to identify Flash block
FCCOB1	Global address [15:0] in Flash block to be erased	

Upon clearing CCIF to launch the Erase Flash Block command, the Memory Controller will erase the selected Flash block and verify that it is erased. The CCIF flag will set after the Erase Flash Block operation has completed.

Table 599. Erase flash block command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied
		Set if the supplied P-Flash address is not phrase-aligned or if the EEPROM address is not word-aligned
	FPVIOL	Set if an area of the selected Flash block is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

6.14.4.6.9 Erase P-flash sector command

The Erase P-Flash Sector operation will erase all addresses in a P-Flash sector.

Table 600. Erase P-Flash Sector Command FCCOB Requirements

Register	FCCOB Parameters	
FCCOB0	0x0A	Global address [23:16] to identify P-Flash block to be erased
FCCOB1	Global address [15:0] anywhere within the sector to be erased. Refer to P-flash features for the P-Flash sector size.	

Upon clearing CCIF to launch the Erase P-Flash Sector command, the Memory Controller will erase the selected Flash sector and then verify that it is erased. The CCIF flag will be set after the Erase P-Flash Sector operation has completed.

Table 601. Erase P-flash sector command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied
		Set if a misaligned phrase address is supplied (global address [2:0]!= 000)
	FPVIOL	Set if the selected P-Flash sector is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

6.14.4.6.10 Unsecure flash command

The Unsecure Flash command will erase the entire P-Flash and EEPROM memory space and, if the erase is successful, will release security.

Table 602. Unsecure flash command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x0B	Not required

Upon clearing CCIF to launch the Unsecure Flash command, the Memory Controller will erase the entire P-Flash and EEPROM memory space and verify that it is erased. If the Memory Controller verifies that the entire Flash memory space was properly erased, security will be released. If the erase verify is not successful, the Unsecure Flash operation sets MGSTAT1 and terminates without changing the security state. During the execution of this command (CCIF=0) the user must not write to any Flash module register. The CCIF flag is set after the Unsecure Flash operation has completed.

Table 603. Unsecure flash command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 000 at command launch
		Set if command not available in current mode (see Table 580)
	FPVIOL	Set if any area of the P-Flash or EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

6.14.4.6.11 Verify backdoor access key command

The Verify Backdoor Access Key command will only execute if it is enabled by the KEYEN bits in the FSEC register (see Table 541). The Verify Backdoor Access Key command releases security if user-supplied keys match those stored in the Flash security bytes of the Flash configuration field (see Table 532). The Verify Backdoor Access Key command must not be executed from the Flash block containing the backdoor comparison key to avoid code runaway.

Table 604. Verify backdoor access key command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x0C	Not required
FCCOB1	Key 0	
FCCOB2	Key 1	
FCCOB3	Key 2	
FCCOB4	Key 3	

Upon clearing CCIF to launch the Verify Backdoor Access Key command, the Memory Controller will check the FSEC KEYEN bits to verify that this command is enabled. If not enabled, the Memory Controller sets the ACCERR bit in the FSTAT register and terminates. If the command is enabled, the Memory Controller compares the key provided in FCCOB to the backdoor comparison key in the Flash configuration field with Key 0 compared to 0xFF_FE00, etc. If the backdoor keys match, security will be released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are aborted (set ACCERR) until a reset occurs. The CCIF flag is set after the Verify Backdoor Access Key operation has completed.

Table 605. Verify backdoor access key command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 100 at command launch
		Set if an incorrect backdoor key is supplied
		Set if backdoor key access has not been enabled (KEYEN[1:0]!= 10, see Flash security register (FSEC))
		Set if the backdoor key has mismatched since the last reset
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

6.14.4.6.12 Set user margin level command

The Set User Margin Level command causes the Memory Controller to set the margin level for future read operations of the P-Flash or EEPROM block.

Table 606. Set user margin level command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x0D	Global address [23:16] to identify Flash block
FCCOB1	Global address [15:0] to identify Flash block	
FCCOB2	Margin level setting.	

Upon clearing CCIF to launch the Set User Margin Level command, the Memory Controller will set the user margin level for the targeted block and then set the CCIF flag.

Note

When the EEPROM block is targeted, the EEPROM user margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash user margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply user margin levels to the P-Flash block only.

Valid margin level settings for the Set User Margin Level command are defined in 607.

Table 607. Valid set user margin level settings

FCCOB1	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level (369)
0x0002	User Margin-0 Level (369)

Notes:
368.Read margin to the erased state
369.Read margin to the programmed state

Table 608. Set user margin level command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 010 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

Note

User margin levels can be used to check that Flash memory contents have adequate margin for normal level read operations. If unexpected results are encountered when checking Flash memory contents at user margin levels, a potential loss of information has been detected.

6.14.4.6.13 Set field margin level command

The Set Field Margin Level command, valid in special modes only, causes the Memory Controller to set the margin level specified for future read operations of the P-Flash or EEPROM block.

Table 609. Set field margin level command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x0E	Global address [23:16] to identify Flash block
FCCOB1	Global address [15:0] to identify Flash block	
FCCOB2	Margin level setting.	

Upon clearing CCIF to launch the Set Field Margin Level command, the Memory Controller will set the field margin level for the targeted block and then set the CCIF flag.

Note

When the EEPROM block is targeted, the EEPROM field margin levels are applied only to the EEPROM reads. However, when the P-Flash block is targeted, the P-Flash field margin levels are applied to both P-Flash and EEPROM reads. It is not possible to apply field margin levels to the P-Flash block only.

Valid margin level settings for the Set Field Margin Level command are defined in Table 610.

Table 610. Valid set field margin level settings

FCCOB1	Level Description
0x0000	Return to Normal Level
0x0001	User Margin-1 Level (370)
0x0002	User Margin-0 Level (371)
0x0003	Field Margin-1 Level (370)
0x0004	Field Margin-0 Level (371)

Notes:
370.Read margin to the erased state
371.Read margin to the programmed state

Table 611. Set field margin level command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 010 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied
		Set if an invalid margin level setting is supplied
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

Note

Field margin levels must only be used during verify of the initial factory programming.

Note

Field margin levels can be used to check that Flash memory contents have adequate margin for data retention at the normal level setting. If unexpected results are encountered when checking Flash memory contents at field margin levels, the Flash memory contents should be erased and reprogrammed.

6.14.4.6.14 Erase verify EEPROM section command

The Erase Verify EEPROM Section command will verify that a section of code in the EEPROM is erased. The Erase Verify EEPROM Section command defines the starting point of the data to be verified and the number of words.

Table 612. Erase verify EEPROM section command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x10	Global address [23:16] to identify the EEPROM block
FCCOB1	Global address [15:0] of the first word to be verified	
FCCOB2	Number of words to be verified	

Upon clearing CCIF to launch the Erase Verify EEPROM Section command, the Memory Controller will verify the selected section of EEPROM memory is erased. The CCIF flag will set after the Erase Verify EEPROM Section operation has completed. If the section is not erased, it means blank check failed, both MGSTAT bits will be set.

Table 613. Erase verify EEPROM section command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 010 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied
		Set if a misaligned word address is supplied (global address [0]!= 0)
		Set if the requested section breaches the end of the EEPROM block
	FPVIOL	None
	MGSTAT1	Set if any errors have been encountered during the read or if blank check failed.
	MGSTAT0	Set if any non-correctable errors have been encountered during the read or if blank check failed.

6.14.4.6.15 Program EEPROM command

The Program EEPROM operation programs one to four previously erased words in the EEPROM block. The Program EEPROM operation will confirm that the targeted location(s) were successfully programmed upon completion.

Note

A Flash word must be in the erased state before being programmed. Cumulative programming of bits within a Flash word is not allowed.

Table 614. Program EEPROM command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x11	Global address [23:16] to identify the EEPROM block
FCCOB1	Global address [15:0] of word to be programmed	
FCCOB2	Word 0 program value	
FCCOB3	Word 1 program value, if desired	
FCCOB4	Word 2 program value, if desired	
FCCOB5	Word 3 program value, if desired	

Upon clearing CCIF to launch the Program EEPROM command, the user-supplied words will be transferred to the Memory Controller and be programmed if the area is unprotected. The CCOBIX index value at Program EEPROM command launch determines how many words will be programmed in the EEPROM block. The CCIF flag is set when the operation has completed.

Table 615. Program EEPROM command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0] < 010 at command launch
		Set if CCOBIX[2:0] > 101 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied
		Set if a misaligned word address is supplied (global address [0]!= 0)
		Set if the requested group of words breaches the end of the EEPROM block
	FPVIOL	Set if the selected area of the EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

6.14.4.6.16 Erase EEPROM sector command

The Erase EEPROM Sector operation will erase all addresses in a sector of the EEPROM block.

Table 616. Erase EEPROM sector command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x12	Global address [23:16] to identify EEPROM block
FCCOB1	Global address [15:0] anywhere within the sector to be erased. See EEPROM features for EEPROM sector size.	

Upon clearing CCIF to launch the Erase EEPROM Sector command, the Memory Controller will erase the selected Flash sector and verify that it is erased. The CCIF flag will set after the Erase EEPROM Sector operation has completed.

Table 617. Erase EEPROM sector command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= 001 at command launch
		Set if command not available in current mode (see Table 580)
		Set if an invalid global address [23:0] is supplied
		Set if a misaligned word address is supplied (global address [0]!= 0)
	FPVIOL	Set if the selected area of the EEPROM memory is protected
	MGSTAT1	Set if any errors have been encountered during the verify operation
	MGSTAT0	Set if any non-correctable errors have been encountered during the verify operation

6.14.4.6.17 Protection override command

The Protection Override command allows the user to temporarily override the protection limits, either decreasing, increasing or disabling protection limits, on P-Flash and/or EEPROM, if the comparison key provided as a parameter loaded on FCCOB matches the value of the key previously programmed on the Flash Configuration Field (see Table 532). The value of the Protection Override Comparison Key must not be 16'hFFFF, that is considered invalid and if used as argument will cause the Protection Override feature to be disabled. Any valid key value that does not match the value programmed in the Flash Configuration Field will cause the Protection Override feature to be disabled. Current status of the Protection Override feature can be observed on FPSTAT FPOVRD bit (see Flash protection status register (FPSTAT)).

Table 618. Protection override command FCCOB requirements

Register	FCCOB Parameters	
FCCOB0	0x13	Protection Update Selection [1:0] See Table 619
FCCOB1	Comparison Key	
FCCOB2	reserved	New FPROT value
FCCOB3	reserved	New DFPROT value

Table 619. Protection override selection description

Protection Update Selection code [1:0]	Protection register selection
bit 0	Update P-Flash protection 0 - keep unchanged (do not update) 1 - update P-Flash protection with new FPROT value loaded on FCCOB
bit 1	Update EEPROM protection 0 - keep unchanged (do not update) 1 - update EEPROM protection with new DFPROT value loaded on FCCOB

If the comparison key successfully matches the key programmed in the Flash Configuration Field the Protection Override command will preserve the current values of registers FPROT and DFPROT stored in an internal area and will override these registers as selected by the Protection Update Selection field with the value(s) loaded on FCCOB parameters. The new values loaded into FPROT and/or DFPROT can reconfigure protection without any restriction (by increasing, decreasing or disabling protection limits). If the command executes successfully the FPSTAT FPOVRD bit will set.

If the comparison key does not match the key programmed in the Flash Configuration Field, or if the key loaded on FCCOB is 16'hFFFF, the value of registers FPROT and DFPROT will be restored to their original contents before executing the Protection Override command and the FPSTAT FPOVRD bit will be cleared. If the contents of the Protection Override Comparison Key in the Flash Configuration Field is left in the erased state (i.e. 16'hFFFF) the Protection Override feature is permanently disabled. If the command execution is flagged as an error (ACCERR being set for incorrect command launch) the values of FPROT and DFPROT will not be modified.

The Protection Override command can be called multiple times and every time it is launched it will preserve the current values of registers FPROT and DFPROT in a single-entry buffer to be restored later; when the Protection Override command is launched to restore FPROT and DFPROT these registers will assume the values they had before executing the Protection Override command on the last time. If contents of FPROT and/or DFPROT registers were modified by direct register writes while protection is overridden these modifications will be lost. Running Protection Override command to restore the contents of registers FPROT and DFPROT will not force them to the reset values.

Table 620. Protection override command error handling

Register	Error Bit	Error Condition
FSTAT	ACCERR	Set if CCOBIX[2:0]!= (001, 010 or 011) at command launch.
		Set if command not available in current mode (see Table 580).
		Set if protection is supposed to be restored (if key does not match or is invalid) and Protection Override command was not run previously (bit FPSTAT FPOVRD is 0), so there are no previous valid values of FPROT and DFPROT to be re-loaded.
		Set if Protection Update Selection[1:0] = 00 (in case of CCOBIX[2:0] = 010 or 011)
		Set if Protection Update Selection[1:0] = 00, CCOBIX[2:0] = 001 and a valid comparison key is loaded as a command parameter.
	FPVIOL	None
	MGSTAT1	None
	MGSTAT0	None

6.14.4.7 Interrupts

The Flash module can generate an interrupt when a Flash command operation has completed or when a Flash command operation has detected an ECC fault.

Table 621. Flash interrupt sources

Interrupt Source	Interrupt Flag	Local Enable	Global (CCR) Mask
Flash Command Complete	CCIF (FSTAT register)	CCIE (FCNFG register)	I Bit
ECC Double Bit Fault on Flash Read	DFDIF (FERSTAT register)	DFDIE (FERCNFG register)	I Bit
ECC Single Bit Fault on Flash Read	SFDIF (FERSTAT register)	SFDIE (FERCNFG register)	I Bit

Note

Vector addresses and their relative interrupt priority are determined at the MCU level.

6.14.4.7.1 Description of flash interrupt operation

The Flash module uses the CCIF flag in combination with the CCIE interrupt enable bit to generate the Flash command interrupt request. The Flash module uses the DFDIF and SFDIF flags in combination with the DFDIE and SFDIE interrupt enable bits to generate the Flash error interrupt request. For a detailed description of the register bits involved, refer to Flash configuration register (FCNFG)", Flash error configuration register (FERCNFG)", Flash status register (FSTAT)", and Flash error status register (FERSTAT)".

The logic used for generating the Flash module interrupts is shown in Figure 103.

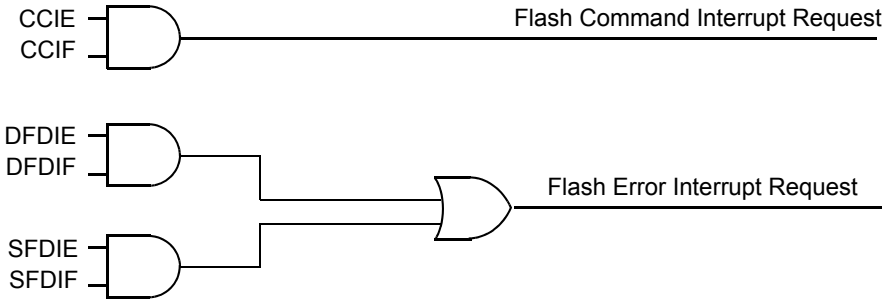


Figure 103. Flash module interrupts implementation

6.14.4.8 Wait mode

The Flash module is not affected if the MCU enters wait mode. The Flash module can recover the MCU from wait via the CCIF interrupt (see Interrupts").

6.14.4.9 Stop mode

If a Flash command is active (CCIF = 0) when the MCU requests stop mode, the current Flash operation will be completed before the MCU is allowed to enter stop mode.

6.14.5 Security

The Flash module provides security information to the MCU. The Flash security state is defined by the SEC bits of the FSEC register (see Table 542). During reset, the Flash module initializes the FSEC register using data read from the security byte of the Flash configuration field at global address 0xFF_FE0F. The security state out of reset can be permanently changed by programming the security byte assuming that the MCU is starting from a mode where the necessary P-Flash erase and program commands are available and that the upper region of the P-Flash is unprotected. If the Flash security byte is successfully programmed, its new value will take affect after the next MCU reset.

The following subsections describe these security-related subjects:

- Unsecuring the MCU using backdoor key access
- Unsecuring the MCU in special single chip mode using BDM
- Mode and security effects on flash command availability

6.14.5.1 Unsecuring the MCU using backdoor key access

The MCU may be unsecured by using the backdoor key access feature which requires knowledge of the contents of the backdoor keys (four 16-bit words programmed at addresses 0xFF_FE00-0xFF_FE07). If the KEYEN[1:0] bits are in the enabled state (see Flash security register (FSEC)), the Verify Backdoor Access Key command (see Verify backdoor access key command) allows the user to present four prospective keys for comparison to the keys stored in the Flash memory via the Memory Controller. If the keys presented in the Verify Backdoor Access Key command match the backdoor keys stored in the Flash memory, the SEC bits in the FSEC register (see Table 542) will be changed to unsecure the MCU. Key values of 0x0000 and 0xFFFF are not permitted as backdoor keys. While the Verify Backdoor Access Key command is active, P-Flash memory and EEPROM memory will not be available for read access and will return invalid data.

The user code stored in the P-Flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN[1:0] bits are in the enabled state (see Flash security register (FSEC)), the MCU can be unsecured by the backdoor key access sequence described below:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Verify backdoor access key command
2. If the Verify Backdoor Access Key command is successful, the MCU is unsecured and the SEC[1:0] bits in the FSEC register are forced to the unsecure state of 10

The Verify Backdoor Access Key command is monitored by the Memory Controller and an illegal key will prohibit future use of the Verify Backdoor Access Key command. A reset of the MCU is the only method to re-enable the Verify Backdoor Access Key command. The security as defined in the Flash security byte (0xFF_FE0F) is not changed by using the Verify Backdoor Access Key command sequence. The backdoor keys stored in addresses 0xFF_FE00-0xFF_FE07 are unaffected by the Verify Backdoor Access Key command sequence. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the Flash protection register, FPROT.

After the backdoor keys have been correctly matched, the MCU will be unsecured. After the MCU is unsecured, the sector containing the Flash security byte can be erased and the Flash security byte can be reprogrammed to the unsecure state, if desired. In the unsecure state, the user has full control of the contents of the backdoor keys by programming addresses 0xFF_FE00-0xFF_FE07 in the Flash configuration field.

6.14.5.2 Unsecuring the MCU in special single chip mode using BDM

A secured MCU can be unsecured in special single chip mode using an automated procedure described in Erase all pin", For a complete description about how to activate that procedure, refer to the Reference Manual. Alternatively, a similar (non-automated) procedure to unsecure the MCU in special single chip mode can be done by using the following method to erase the P-Flash and EEPROM memory:

1. Reset the MCU into special single chip mode
2. Delay while the BDM executes the Erase Verify All Blocks command write sequence to check if the P-Flash and EEPROM memories are erased
3. Send BDM commands to disable protection in the P-Flash and EEPROM memory
4. Execute the Erase All Blocks command write sequence to erase the P-Flash and EEPROM memory. Alternatively the Unsecure Flash command can be executed, if so the steps 5 and 6 below are skipped.
5. After the CCIF flag sets to indicate that the Erase All Blocks operation has completed, reset the MCU into special single chip mode
6. Delay while the BDM executes the Erase Verify All Blocks command write sequence to verify that the P-Flash and EEPROM memory are erased

If the P-Flash and EEPROM memory are verified as erased, the MCU will be unsecured. All BDM commands will now be enabled and the Flash security byte may be programmed to the unsecure state by continuing with the following steps:

7. Send BDM commands to execute the Program P-Flash command write sequence to program the Flash security byte to the unsecured state
8. Reset the MCU

6.14.5.3 Mode and security effects on flash command availability

The availability of Flash module commands depends on the MCU operating mode and security state as shown in Table 580.

6.14.6 Initialization

On each system reset the flash module executes an initialization sequence which establishes initial values for the Flash Block Configuration Parameters, the FPROT and DFPROT protection registers, and the FOPT and FSEC registers. The initialization routine reverts to built-in default values that leave the module in a fully protected and secured state if errors are encountered during execution of the reset sequence. If a double bit fault is detected during the reset sequence, both MGSTAT bits in the FSTAT register will be set.

CCIF is cleared throughout the initialization sequence. The Flash module holds off all CPU access for a portion of the initialization sequence. Flash reads are allowed once the hold is removed. Completion of the initialization sequence is marked by setting CCIF high which enables user commands.

If a reset occurs while any Flash command is in progress, that command will be immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed.

6.15 Basic timer module - TIM (TIM16B4C)

6.15.1 Introduction

6.15.1.1 Overview

The basic timer consists of a 16-bit, software-programmable counter driven by a seven stage programmable prescaler.

This timer can be used for many purposes, including input waveform measurements while simultaneously generating an output waveform. Pulse widths can vary from microseconds to many seconds.

This timer contains four complete input capture/output compare channels [IOC 3:0]. The input capture function is used to detect a selected transition edge and record the time. The output compare function is used for generating output signals or for timer software delays.

Full access for the counter registers or the input capture/output compare registers should take place in a16-bit word access. Accessing high bytes and low bytes separately for all of these registers may not yield the same result as accessing them in one word.

6.15.1.2 Features

The TIM16B4C includes these distinctive features:

- Four input capture/output compare channels.
- Clock prescaler
- 16-bit counter

6.15.1.3 Modes of operation

The TIM16B4C is driven by the D2DCLK / 4 during Normal mode and the ALFCLK during Low-power mode.

6.15.1.4 Block diagram

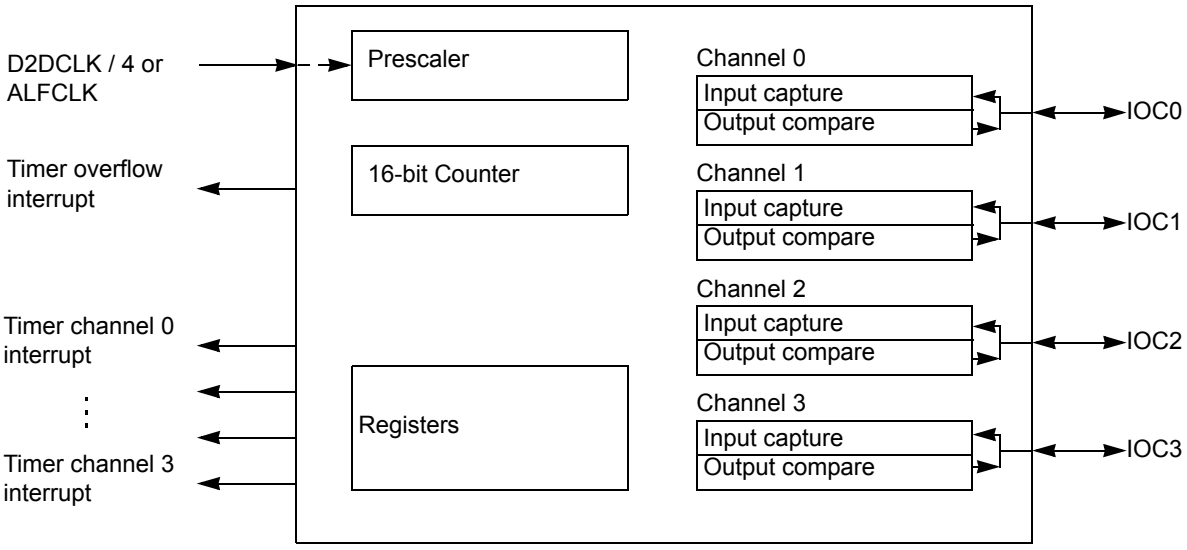


Figure 104. Timer block diagram

For more information on the respective functional descriptions see Functional description of this chapter.

6.15.2 Signal description

6.15.2.1 Overview

The TIM16B4C module can be used as regular time base, or can be internally routed to the PTB and LIN module. Refer to the corresponding sections for further details, see LIN and General purpose I/O - GPIO. In addition, the TIM16B4C module is used during Low-power mode to determine the cyclic wake-up and current measurement timing.

6.15.2.2 Detailed signal descriptions

6.15.2.2.1 IOC3 – input capture and output compare channel 3

This pin serves as the input capture or output compare for channel 3.

6.15.2.2.2 IOC2 – input capture and output compare channel 2

This pin serves as the input capture or output compare for channel 2.

6.15.2.2.3 IOC1 – input capture and output compare channel 1

This pin serves as the input capture or output compare for channel 1.

6.15.2.2.4 IOC0 – input capture and output compare channel 0

This pin serves as the input capture or output compare for channel 0.

6.15.3 Memory map and registers

6.15.3.1 Overview

This section provides a detailed description of all memory and registers.

6.15.3.2 Module memory map

The memory map for the TIM16B4C module is given in 60.

Table 622. Module memory map

Offset [372]	Name		7	6	5	4	3	2	1	0
0x20	TIOS	R	0	0	0	0	IOS3	IOS2	IOS1	IOS0
	Timer Input Capture/Output Compare Select	W								
0x21 [373]	CFORC	R	0	0	0	0	0	0	0	0
	Timer Compare Force Register	W					FOC3	FOC2	FOC1	FOC0
0x22	OC3M	R	0	0	0	0	OC3M3	OC3M2	OC3M1	OC3M0
	Output Compare 3 Mask Register	W								
0x23	OC3D	R	0	0	0	0	OC3D3	OC3D2	OC3D1	OC3D0
	Output Compare 3 Data Register	W								
0x24 [374]	TCNT (hi)	R	TCNT							
	Timer Count Register	W								
0x25 [374]	TCNT (lo)	R								
	Timer Count Register	W								
0x26	TSCR1	R	TEN	0	0	TFFCA	0	0	0	0
	Timer System Control Register 1	W								
0x27	TTOV	R	0	0	0	0	TOV3	TOV2	TOV1	TOV0
	Timer Toggle Overflow Register	W								
0x28	TCTL1	R	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
	Timer Control Register 1	W								
0x29	TCTL2	R	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
	Timer Control Register 2	W								
0x2A	TIE	R	0	0	0	0	C3I	C2I	C1I	C0I
	Timer Interrupt Enable Register	W								
0x2B	TSCR2	R	TOI	0	0	0	TCRE	PR2	PR1	PR0
	Timer System Control Register 2	W								
0x2C	TFLG1	R	0	0	0	0	C3F	C2F	C1F	C0F
	Main Timer Interrupt Flag 1	W								
0x2D	TFLG2	R	TOF	0	0	0	0	0	0	0
	Main Timer Interrupt Flag 2	W								
0x2E [375]	TC0 (hi)	R	TC0							
	Timer Input Capture/Output Compare Register 0	W								
0x2F [375]	TC0 (lo)	R								
	Timer Input Capture/Output Compare Register 0	W								
0x30 [375]	TC1 (hi)	R	TC1							
	Timer Input Capture/Output Compare Register 1	W								
0x31 [375]	TC1 (lo)	R								
	Timer Input Capture/Output Compare Register 1	W								

Table 622. Module memory map (continued)

Offset [372]	Name		7	6	5	4	3	2	1	0
0x32 [375]	TC2 (hi)	R	TC2							
	Timer Input Capture/Output Compare Register 2	W								
0x33 [375]	TC2 (lo)	R								
	Timer Input Capture/Output Compare Register 2	W								
0x34 [375]	TC3 (hi)	R	TC3							
	Timer Input Capture/Output Compare Register 3	W								
0x35 [375]	TC3 (lo)	R								
	Timer Input Capture/Output Compare Register 3	W								
0x36 [374]	TIMTST	R	0	0	0	0	0	0	TCBYP	0
	Timer Test Register	W								

Notes:
372.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
373.Always reads $00.
374.Only writable in special modes. (Refer to the SOC Guide for different modes).
375.A write to these registers has no meaning or effect during input capture.

6.15.3.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

6.15.3.3.1 Timer input capture/output compare select (TIOS)

Table 623. Timer input capture/output compare select (TIOS)

Offset[376]	0x20						Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	0	0	0	IOS3	IOS2	IOS1	IOS0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
376.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 624. TIOS - register field descriptions

Field	Description
3-0 IOS[3-0]	**Input Capture or Output Compare Channel Configuration** 0 - The corresponding channel acts as an input capture. 1 - The corresponding channel acts as an output compare.

6.15.3.3.2 Timer compare force register (CFORC)

Table 625. Timer compare force register (CFORC)

Offset[377]	0x21						Access: User write	
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0
W					FOC3	FOC2	FOC1	FOC0
Reset	0	0	0	0	0	0	0	0

Notes:
377.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 626. CFORC - register field descriptions

Field	Description
3-0 FOC[3-0]	**Force Output Compare Action for Channel 3-0** 0 - Force output compare action disabled. Input capture or output compare channel configuration 1 - Force output compare action enabled

A write to this register with the corresponding (FOC 3:0) data bit(s) set causes the action programmed for output compare on channel “n” to occur immediately.The action taken is the same as if a successful comparison had just taken place with the TCn register, except the interrupt flag does not get set.

Note

A successful channel 3 output compare overrides any channel 2:0 compare. If a forced output compare on any channel occurs at the same time as the successful output compare, then a forced output compare action will take precedence and the interrupt flag will not get set.

6.15.3.3.3 Output compare 3 mask register (OC3M)

Table 627. Output compare 3 mask register (OC3M)

Offset[378]	0x22						Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	0	0	0	OC3M3	OC3M2	OC3M1	OC3M0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
378.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 628. OC3M - register field descriptions

Field	Description
3-0 OC3M[3-0]	**Output Compare 3 Mask “n” Channel bit** 0 - Does not set the corresponding port to be an output port 1 - Sets the corresponding port to be an output port when this corresponding TIOS bit is set to be an output compare

Setting the OC3Mn (n ranges from 0 to 2) will set the corresponding port to be an output port when the corresponding TIOSn (n ranges from 0 to 2) bit is set to be an output compare.

Note

A successful channel 3 output compare overrides any channel 2:0 compares. For each OC3M bit that is set, the output compare action reflects the corresponding OC3D bit.

6.15.3.3.4 Output compare 3 data register (OC3D)

Table 629. Output compare 3 data register (OC3D)

Offset[379]	0x23						Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	0	0	0	OC3D3	OC3D2	OC3D1	OC3D0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
379.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 630. OC3D - register field descriptions

Field	Description
3 OC3D3	Output Compare 3 Data for Channel 3
2 OC3D2	Output Compare 3 Data for Channel 2
1 OC3D1	Output Compare 3 Data for Channel 1
0 OC3D0	Output Compare 3 Data for Channel 0

Note

A channel 3 output compare will cause bits in the output compare 3 data register to transfer to the timer port data register if the corresponding output compare 3 mask register bits are set.

6.15.3.3.5 Timer count register (TCNT)

Table 631. Timer count register (TCNT)

Offset[380]	0x24, 0x25						Access: User read (anytime)/write (special mode)	
	15	14	13	12	11	10	9	8
R	tcnt15	tcnt14	tcnt13	tcnt12	tcnt11	tcnt10	tcnt9	tcnt8
W								
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	tcnt7	tcnt6	tcnt5	tcnt4	tcnt3	tcnt2	tcnt1	tcnt0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
380.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 632. TCNT - register field descriptions

Field	Description
15-0 tcnt[15-0]	16-Bit Timer Count Register

Note

The 16-bit main timer is an up counter. Full access to the counter register should take place in one clock cycle. A separate read/write for high bytes and low bytes will give a different result than accessing them as a word. The period of the first count after a write to the TCNT registers may be a different length, because the write is not synchronized with the prescaler clock.

6.15.3.3.6 Timer system control register 1 (TSCR1)

Table 633. Timer system control register 1 (TSCR1)

Offset[381]	0x26					Access: User read/write		
	7	6	5	4	3	2	1	0
R	TEN	0	0	TFFCA	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
381.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 634. TSCR1 - register field descriptions

Field	Description
7 TEN	**Timer Enable** 1 = Enables the timer. 0 = Disables the timer. (Used for reducing power consumption).
4 TFFCA	**Timer Fast Flag Clear All** 1 = For TFLG1 register, a read from an input capture or a write to the output compare channel [TC 3:0] causes the corresponding channel flag, CnF, to be cleared. For TFLG2 register, any access to the TCNT register clears the TOF flag. This has the advantage of eliminating software overhead in a separate clear sequence. Extra care is required to avoid accidental flag clearing due to unintended accesses. 0 = Allows the timer flag clearing.

6.15.3.3.7 Timer toggle on overflow register 1 (TTOV)

Table 635. Timer toggle on overflow register 1 (TTOV)

Offset[382]	0x27					Access: User read/write		
	7	6	5	4	3	2	1	0
R	0	0	0	0	TOV3	TOV2	TOV1	TOV0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
382.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 636. TTOV - register field descriptions

Field	Description
3-0 TOV[3-0]	**Toggle On Overflow Bits** 1 = Toggle output compare pin on overflow feature enabled. 0 = Toggle output compare pin on overflow feature disabled.

Note

TOVn toggles the output compare pin on overflow. This feature only takes effect when the corresponding channel is configured for an output compare mode. When set, an overflow toggle on the output compare pin takes precedence over forced output compare events.

6.15.3.3.8 Timer control register 1 (TCTL1)

Table 637. Timer control register 1 (TCTL1)

Offset[383]	0x28	Access: User read/write

	7	6	5	4	3	2	1	0
R W	OM3	OL3	OM2	OL2	OM1	OL1	OM0	OL0
Reset	0	0	0	0	0	0	0	0

Notes:
383.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 638. TCTL1 - register field descriptions

Field	Description
7,5,3,1 OMn	Output Mode bit
6,4,2,0 OLn	Output Level bit

Note

These four pairs of control bits are encoded to specify the output action to be taken as a result of a successful Output Compare on “n” channel. When either OMn or OLn, the pin associated with the corresponding channel becomes an output tied to its IOC. To enable output action by the OMn and OLn bits on a timer port, the corresponding bit in OC3M should be cleared.

Table 639. Compare result output action

OMn	OLn	Action
0	0	Timer disconnected from output pin logic
0	1	Toggle OCn output line
1	0	Clear OCn output line to zero
1	1	Set OCn output line to one

6.15.3.3.9 Timer control register 2 (TCTL2)

Table 640. Timer Control Register 2 (TCTL2)

Offset[384]	0x29	Access: User read/write

	7	6	5	4	3	2	1	0
R W	EDG3B	EDG3A	EDG2B	EDG2A	EDG1B	EDG1A	EDG0B	EDG0A
Reset	0	0	0	0	0	0	0	0

Notes:
384.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 641. TCTL2 - register field descriptions

Field	Description
EDGnB,EDGnA	Input Capture Edge Control

These four pairs of control bits configure the input capture edge detector circuits.

Table 642. Edge detector circuit configuration

EDGnB	EDGnA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any edge (rising or falling)

6.15.3.3.10 Timer interrupt enable register (TIE)

Table 643. Timer interrupt enable register (TIE)

Offset[385]	0x2A					Access: User read/write		
	7	6	5	4	3	2	1	0
R	0	0	0	0	C3I	C2I	C1I	C0I
W								
Reset	0	0	0	0	0	0	0	0

Notes:
385.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 644. TIE - register field descriptions

Field	Description
3-0 C[3-0]I	**Input Capture/Output Compare Interrupt Enable.** 1 = Enables corresponding Interrupt flag (CnF of TFLG1 register) to cause a hardware interrupt 0 = Disables corresponding Interrupt flag (CnF of TFLG1 register) from causing a hardware interrupt

6.15.3.3.11 Timer system control register 2 (TSCR2)

Table 645. Timer system control register 2 (TSCR2)

Offset[386]	0x2B					Access: User read/write		
	7	6	5	4	3	2	1	0
R	TOI	0	0	0	TCRE	PR2	PR1	PR0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
386.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 646. TSCR2 - register field descriptions

Field	Description
7 TOI	**Timer Overflow Interrupt Enable** 1 = Hardware interrupt requested when TOF flag set in TFLG2 register. 0 = Hardware Interrupt request inhibited.
3 TCRE	**TCRE** — Timer Counter Reset Enable 1 = Enables timer counter reset by a successful output compare on channel 3 0 = Inhibits timer counter reset and counter continues to run.
3-0 PR[2:0]	Timer Prescaler Select These three bits select the frequency of the timer prescaler clock derived from the bus clock as shown in Table 647.

Note

This mode of operation is similar to an up-counting modulus counter.

If register TC3 = $0000 and TCRE = 1, the timer counter register (TCNT) will stay at $0000 continuously. If register TC3 = $FFFF and TCRE = 1, TOF will not be set when the timer counter register (TCNT) is reset from $FFFF to $0000.

The newly selected prescale factor will not take effect until the next synchronized edge, where all prescale counter stages equal zero.

Table 647. Timer clock selection

PR2	PR1	PR0	Timer Clock(387)
0	0	0	TimerClk / 1
0	0	1	TimerClk / 2
0	1	0	TimerClk / 4
0	1	1	TimerClk / 8
1	0	0	TimerClk / 16
1	0	1	TimerClk / 32
1	1	0	TimerClk / 64
1	1	1	TimerClk / 128

Notes:
387.TimerClk = D2DCLK/4 or ALFCLK

6.15.3.3.12 Main timer interrupt flag 1 (TFLG1)

Table 648. Main timer interrupt flag 1 (TFLG1)

Offset(388)	0x2C						Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	0	0	0	C3F	C2F	C1F	C0F
W								
Reset	0	0	0	0	0	0	0	0

Notes:
388.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 649. TFLG1 - register field descriptions

Field	Description
3-0 C[3:0]F	**Input Capture/Output Compare Channel Flag.** 1 = Input capture or output compare event occurred 0 = No event (input capture or output compare event) occurred.

Note

These flags are set when an input capture or output compare event occurs. Flag set on a particular channel is cleared by writing a one to that corresponding CnF bit. Writing a zero to CnF bit has no effect on its status. When TFFCA bit in TSCR register is set, a read from an input capture or a write into an output compare channel will cause the corresponding channel flag CnF to be cleared.

6.15.3.3.13 Main timer interrupt flag 2 (TFLG2)

Table 650. Main timer interrupt flag 2 (TFLG2)

Offset[389]	0x2D						Access: User read/write	
	7	6	5	4	3	2	1	0
R	TOF	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
389.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 651. TFLG2 - register field descriptions

Field	Description
7 TOF	**Timer Overflow Flag** 1 = Indicates that an interrupt has occurred (Set when 16-bit free-running timer counter overflows from $FFFF to $0000) 0 = Flag indicates an interrupt has not occurred.

Note

The TFLG2 register indicates when an interrupt has occurred. Writing a one to the TOF bit will clear it. Any access to TCNT will clear TOF bit of TFLG2 register if the TFFCA bit in TSCR register is set.

6.15.3.3.14 Timer input capture/output compare registers (TC3 - TC0)

Table 652. Timer input capture/output compare register 0 (TC0)

Offset[390]	0x2E, 0x2F					Access: User read/write		
	15	14	13	12	11	10	9	8
R	tc0_15	tc0_14	tc0_13	tc0_12	tc0_11	tc0_10	tc0_9	tc0_8
W								
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	tc0_7	tc0_6	tc0_5	tc0_4	tc0_3	tc0_2	tc0_1	tc0_0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
390.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 653. Timer input capture/output compare register 1(TC1)

Offset[391]	0x30, 0x31					Access: User read/write		
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R W	tc1_15	tc1_14	tc1_13	tc1_12	tc1_11	tc1_10	tc1_9	tc1_8
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	tc1_7	tc1_6	tc1_5	tc1_4	tc1_3	tc1_2	tc1_1	tc1_0
Reset	0	0	0	0	0	0	0	0

Notes:
391.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 654. Timer input capture/output compare register 2(TC2)

Offset[392]	0x32, 0x33					Access: User read/write		
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R W	tc2_15	tc2_14	tc2_13	tc2_12	tc2_11	tc2_10	tc2_9	tc2_8
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	tc2_7	tc2_6	tc2_5	tc2_4	tc2_3	tc2_2	tc2_1	tc2_0
Reset	0	0	0	0	0	0	0	0

Notes:
392.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 655. Timer input capture/output compare register 3(TC3)

Offset[393]	0x34, 0x35					Access: User read/write		
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R W	tc3_15	tc3_14	tc3_13	tc3_12	tc3_11	tc3_10	tc3_9	tc3_8
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R W	tc3_7	tc3_6	tc3_5	tc3_4	tc3_3	tc3_2	tc3_1	tc3_0
Reset	0	0	0	0	0	0	0	0

Notes:
393.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 656. TCn - register field descriptions

Field	Description
15-0 tcn[15-0]	16 Timer Input Capture/Output Compare Registers

Note

Read anytime. Write anytime for output compare function. Writes to these registers have no effect during input capture.

Depending on the TIOS bit for the corresponding channel, these registers are used to latch the value of the free-running counter when a defined transition is sensed by the corresponding input capture edge detector or to trigger an output action for output compare.

Read/Write access in byte mode for high byte should takes place before low byte otherwise it will give a different result.

6.15.4 Functional description

6.15.4.1 General

This section provides a complete functional description of the timer TIM16B4C block. Refer to the detailed timer block diagram in Figure 105 as necessary.

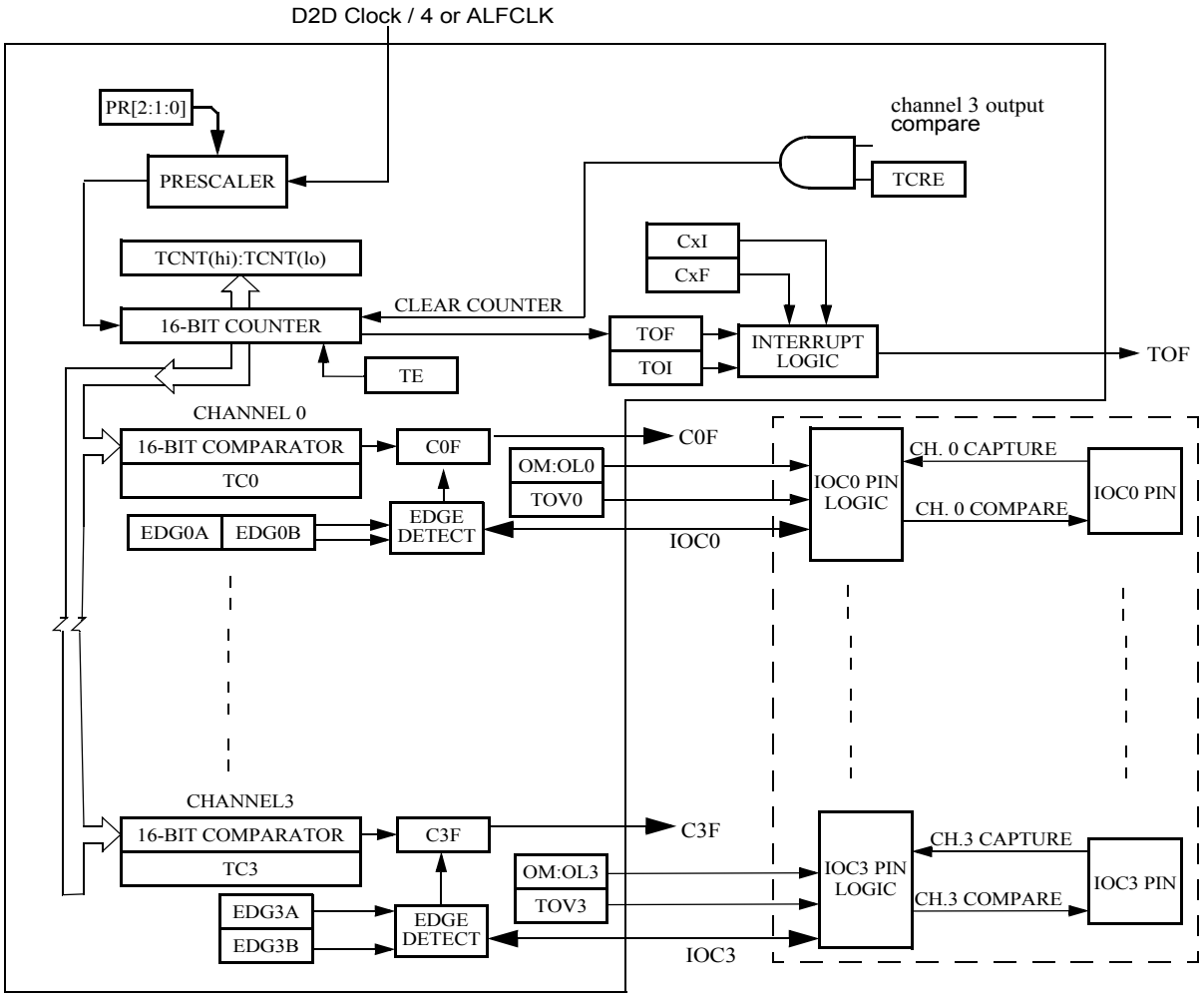


Figure 105. Detailed timer block diagram

6.15.4.2 Prescaler

The prescaler divides the bus clock by 1, 2, 4, 8,16, 32, 64, or 128. The prescaler select bits, PR[2:0], select the prescaler divisor. PR[2:0] are in the timer system control register 2 (TSCR2).

6.15.4.3 Input capture

Clearing the I/O (input/output) select bit, IOSn, configures channel n as an input capture channel. The input capture function captures the time at which an external event occurs. When an active edge occurs on the pin of an input capture channel, the timer transfers the value in the timer counter into the timer channel registers, TCn.

The minimum pulse width for the input capture input is greater than two bus clocks. An input capture on channel n sets the CnF flag. The CnI bit enables the CnF flag to generate interrupt requests.

6.15.4.4 Output compare

Setting the I/O select bit, IOSn, configures channel n as an output compare channel. The output compare function can generate a periodic pulse with a programmable polarity, duration, and frequency. When the timer counter reaches the value in the channel registers of an output compare channel, the timer can set, clear, or toggle the channel pin. An output compare on channel n sets the CnF flag. The CnI bit enables the CnF flag to generate interrupt requests.

The output mode and level bits, OMn and OLn, select set, clear, toggle on output compare. Clearing both OMn and OLn disconnects the pin from the output logic. Setting a force output compare bit, FOCn, causes an output compare on channel n. A forced output compare does not set the channel flag.

A successful output compare on channel 3 overrides output compares on all other output compare channels. The output compare 3 mask register masks the bits in the output compare 3 data register. The timer counter reset enable bit, TCRE, enables channel 3 output compares to reset the timer counter.Writing to the timer port bit of an output compare pin does not affect the pin state. The value written is stored in an internal latch. When the pin becomes available for general purpose output, the last value written to the bit appears at the pin.

6.15.5 Resets

6.15.5.1 General

The reset state of each individual bit is listed within the Register Description Memory map and registers, which details the registers and their bit-fields.

6.15.6 Interrupts

6.15.6.1 General

This section describes interrupts originated by the TIM16B4C block. Table 657 lists the interrupts generated by the TIM16B4C to communicate with the MCU.

Table 657. TIM16B4C interrupts

Interrupt	Offset	Vector	Priority	Source	Description
C[3:0]F	-	-	-	Timer Channel 3-0	Active high timer channel interrupts 3-0
TOF	-	-	-	Timer Overflow	Timer Overflow interrupt

6.15.6.2 Description of interrupt operation

The TIM16B4C uses a total of 5 interrupt vectors. The interrupt vector offsets and interrupt numbers are chip dependent. More information on interrupt vector offsets and interrupt numbers can be found in .

Channel [3:0] Interrupt

These active high outputs is asserted by the module to request a timer channel 3 – 0 interrupt following an input capture or output compare event on these channels [3-0]. For the interrupt to be asserted on a specific channel, the enable, CnI bit of TIE register should be set. These interrupts are serviced by the system controller.

6.15.6.2.1 Timer overflow interrupt (TOF)

This active high output will be asserted by the module to request a timer overflow interrupt, following the timer counter overflow when the overflow enable bit (TOI) bit of TFLG2 register is set. This interrupt is serviced by the system controller.

6.16 Life time counter (LTC)

6.16.1 Introduction

The Life Time Counter is implemented as flexible counter running in both, low power (STOP and SLEEP) and Normal modes.

It is based on the ALFCLK clock featuring IRQ and Wake-up capabilities on the Life Time Counter Overflow. The Wake-up on overflow would be indicated in the PCR_SR register WULTCF bit. The Life Time Counter has to be set in Normal mode. The Life Time Counter is an up counter.

6.16.2 Memory map and registers

6.16.2.1 Overview

This section provides a detailed description of the memory map and registers.

6.16.2.2 Module memory map

The memory map for the LTC module is in Table 60

Table 658. Module memory map

Offset[394]	Name		7	6	5	4	3	2	1	0
0x38	LTC_CTL (hi)	R	0	0	0	0	0	0	0	0
	Life Time Counter control register	W	LTCIEM							LTCEM
	LTC_CTL (lo)	R	LTCIE	0	0	0	0	0	0	LTCE
	Life Time Counter control register	W								
0x3A	LTC_SR	R	LTCOF	0	0	0	0	0	0	0
	Life Time Counter status register	W	1 = clear							
0x3B	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x3C	LTC_CNT1 Life Time Counter Register	R	LTC[31:0]							
		W								
		R								
		W								
0x3E	LTC_CNT0 Life Time Counter Register	R								
		W								
		R								
		W								

Notes:
394.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

6.16.2.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of register bit and field function follow the register diagrams, in bit order.

6.16.2.3.1 Life time counter control register (LTC_CTL)

Table 659. Life time counter control register (LTC_CTL)

Offset[395], [396]	0x38						Access: User write	
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R	0	0	0	0	0	0	0	0
W	LTCIEM							LTCEM
Reset	0	0	0	0	0	0	0	0
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	LTCIE	0	0	0	0	0	0	LTCE
W								
Reset	0	0	0	0	0	0	0	0

Notes:
395.This register is 16-bit access.
396.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 660. Life time counter control register (LTC_CTL) - register field descriptions

Field	Description
15 LTCIEM	**Life Time Counter Interrupt Enable Mask** 0 - writing the LTCIE Bit will have no effect 1 - writing the LTCIE Bit will be effective
8 LTCEM	**Life Time Counter Enable Mask** 0 - writing the LTCE Bit will have no effect 1 - writing the LTCE Bit will be effective
7 LTCIE	**Life Time Counter Interrupt Enable** 1 - Life time counter overflow will generate an interrupt request. 0 - Life time counter overflow will not generate an interrupt request.
0 LTCE	**Life Time Counter Enable** 1 - Life time counter module enabled. Counter will be incremented with based on the ALFCLK frequency. 0 - Life time counter module disabled. Counter content will remain.[397]

Notes:
397.The first period after enable might be shorted due to the asynchronous clocks.

6.16.2.3.2 Life time counter status register (LTC_SR)

Table 661. Life time counter status register (LTC_SR)

Offset[398]	0x3A						Access: User read/write	
	7	6	5	4	3	2	1	0
R	LTCOF	0	0	0	0	0	0	0
W								
Reset	0	0	0	0	0	0	0	0

Notes:
398.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 662. Life time counter status register (LTC_SR) - register field descriptions

Field	Description
0 LTCOF	**Life Time Counter Overflow Flag. Writing 1 will clear the flag.** 1 - Life time counter overflow detected. 0 - No life time counter overflow since last clear

6.16.2.3.3 Life time counter register (LTC_CNT1, LTC_CNT0)

Table 663. Life time counter register (LTC_CNT1, LTC_CNT0)

Offset [399],[400]	0x3C, 0x3E						Access: User read/write	
	7	6	5	4	3	2	1	0
R	LTC[31:16]							
W								
R								
W								
R	LTC[15:0]							
W								
R								
W								
Reset	0	0	0	0	0	0	0	0

Notes:
399.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
400.Those Registers are 16-Bit access only.

Table 664. Life time counter register (LTC_CNT1, LTC_CNT0) - register field descriptions

Field	Description
0-31 LTC[31:0]	Life Time Counter Register The two 16-Bit words of the 32-Bit Life Time Counter register represent the current counter status. Whenever the microcontroller performs a reading operation on one of the 16Bit registers, the Life Time Counter is stopped until the remaining 16-Bit register is read, to prevent loss of information. After the second part is read, the LTC continues automatically. Write operations should be performed with the Life Time Counter disabled to prevent a loss of data.

6.17 Serial communication interface (S08SCIV4)

6.17.1 Introduction

6.17.1.1 Features

Features of SCI module include:

- Full-duplex, standard non-return-to-zero (NRZ) format
- Double-buffered transmitter and receiver with separate enables
- Programmable baud rates (13-bit modulo divider)
- Interrupt-driven or polled operation:
  - Transmit data register empty and transmission complete
  - Receive data register full
  - Receive overrun, parity error, framing error, and noise error
  - Idle receiver detect
  - Active edge on receive pin
  - Break detect supporting LIN
- Hardware parity generation and checking
- Programmable 8-bit or 9-bit character length
- Receiver wake-up by idle-line or address-mark
- Optional 13-bit break character generation / 11-bit break character detection
- Selectable transmitter output polarity
- A clock divider DPD[1..0] of the D2DCLK by 1, 2 or 4

6.17.1.2 Modes of operation

See Functional description, for details concerning SCI operation in these modes:

- 8- and 9-bit data modes
- Loop mode
- Single-wire mode

6.17.1.3 Block diagram

Figure 106 shows the transmitter portion of the SCI.

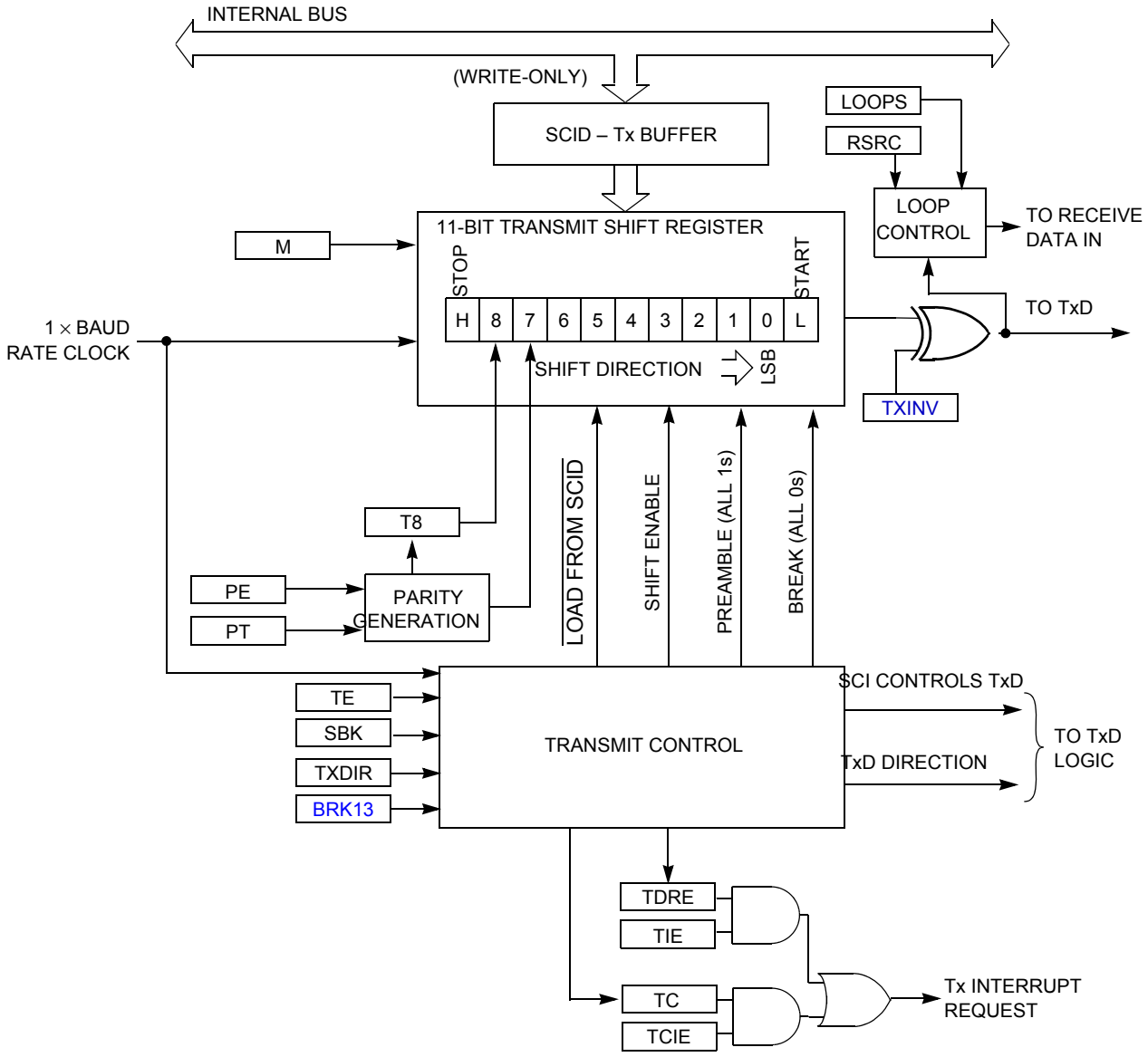


Figure 106. SCI transmitter block diagram

Figure 107 shows the receiver portion of the SCI.

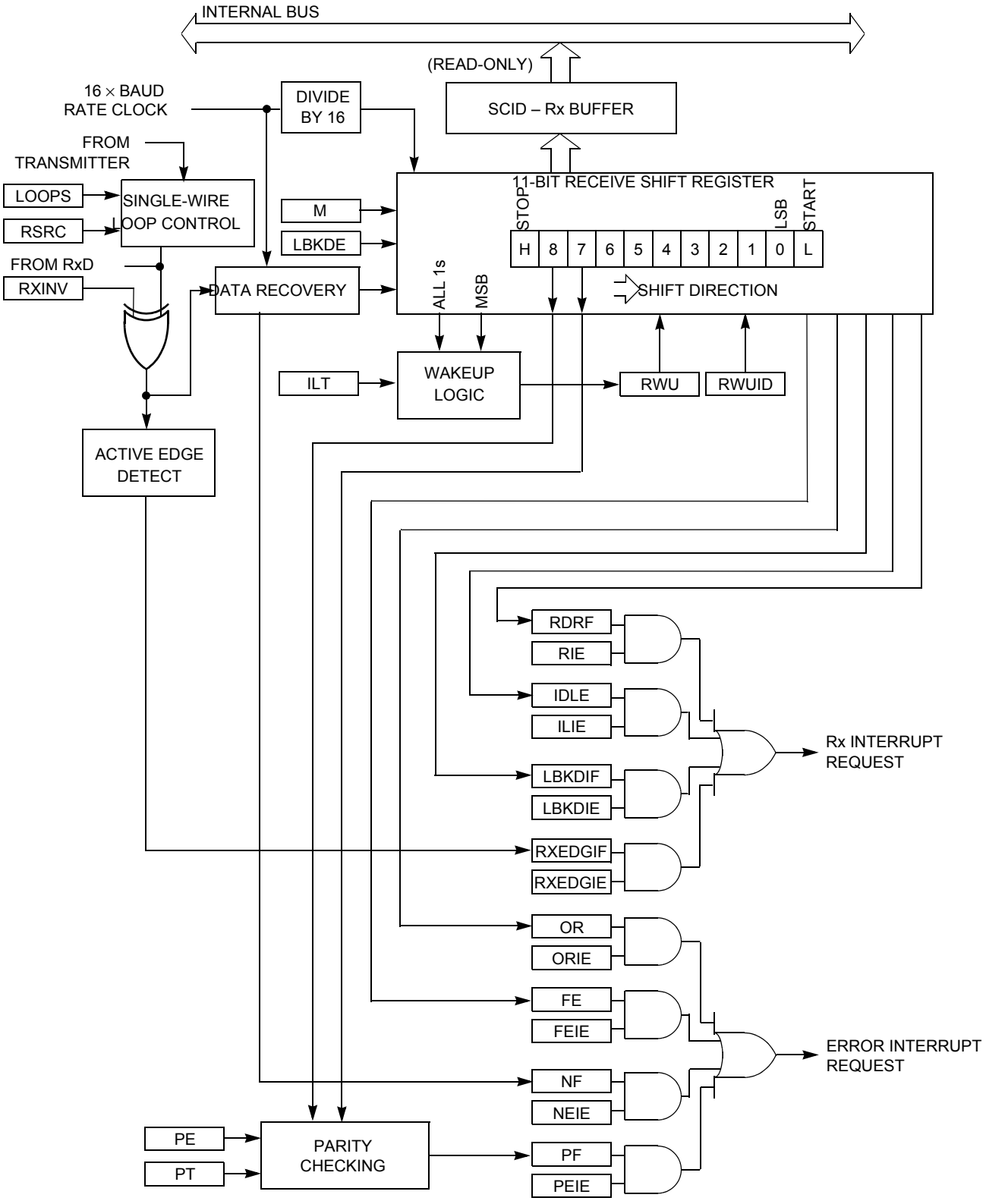


Figure 107. SCI receiver block diagram

MM9Z1_638

6.17.2 Memory map and registers

6.17.2.1 Overview

This section provides a detailed description of the memory map and registers.

6.17.2.2 Module memory map

The memory map for the S08SCIV4 module is given in Table 60.

Table 665. Module memory map

Offset[401]	Name		7	6	5	4	3	2	1	0
0x18	SCIBD (hi)	R	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
	SCI Baud Rate Register	W								
0x19	SCIBD (lo)	R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
	SCI Baud Rate Register	W								
0x1A	SCIC1	R	LOOPS	DPD1	RSRC	M	DPD0	ILT	PE	PT
	SCI Control Register 1	W								
0x1B	SCIC2	R	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
	SCI Control Register 2	W								
0x1C	SCIS1	R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
	SCI Status Register 1	W								
0x1D	SCIS2	R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
	SCI Status Register 2	W								
0x1E	SCIC3	R	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
	SCI Control Register 3	W								
0x1F	SCID	R	R7	R6	R5	R4	R3	R2	R1	R0
	SCI Data Register	W	T7	T6	T5	T4	T3	T2	T1	T0

Notes:
401.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

6.17.2.3 Register definition

The SCI has eight 8-bit registers to control baud rate, select SCI options, report SCI status, and for transmit/receive data.

6.17.2.3.1 SCI baud rate registers (SCIBD (hi), SCIBD (lo))

This pair of registers control the prescale divisor for SCI baud rate generation. To update the 13-bit baud rate setting [SBR12:SBR0], first write to SCIBD (hi) to buffer the high half of the new value, and then write to SCIBD (lo). The working value in SCIBD (hi) does not change until SCIBD (lo) is written.

SCIBDL is reset to a non-zero value, so after reset the baud rate generator remains disabled until the first time the receiver or transmitter is enabled (RE or TE bits in SCIC2 are written to 1).

Table 666. SCI baud rate register (SCIBD (hi))

Offset (402)	0x18					Access: User read/write		
	7	6	5	4	3	2	1	0
R	LBKDIE	RXEDGIE	0	SBR12	SBR11	SBR10	SBR9	SBR8
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
402.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Bit Legend

= Unimplemented, Reserved

= Implemented (do not alter)

X = Indeterminate

0 = Always read zero

Table 667. SCIBD (hi) field descriptions

Field	Description
7 LBKDIE	**LIN Break Detect Interrupt Enable (for LBKDIF)** 0 Hardware interrupts from LBKDIF disabled (use polling). 1 Hardware interrupt requested when LBKDIF flag is 1.
6 RXEDGIE	**RxD Input Active Edge Interrupt Enable (for RXEDGIF)** 0 Hardware interrupts from RXEDGIF disabled (use polling). 1 Hardware interrupt requested when RXEDGIF flag is 1.
4:0 SBR[12:8]	**Baud Rate Modulo Divisor** — The 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(16xDPD[1:0]xBR).

Table 668. SCI baud rate register (SCIBDL)

Offset(403)	0x19					Access: User read/write		
	7	6	5	4	3	2	1	0
R	SBR7	SBR6	SBR5	SBR4	SBR3	SBR2	SBR1	SBR0
W								
Reset	0	0	0	0	0	1	0	0

Notes:
403.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 669. SCIBDL field descriptions

Field	Description
7:0 SBR[7:0]	**Baud Rate Modulo Divisor** — These 13 bits in SBR[12:0] are referred to collectively as BR, and they set the modulo divide rate for the SCI baud rate generator. When BR = 0, the SCI baud rate generator is disabled to reduce supply current. When BR = 1 to 8191, the SCI baud rate = BUSCLK/(16xDPD[1:0]xBR).

6.17.2.3.2 SCI control register 1 (SCIC1)

This read/write register is used to control various optional features of the SCI system.

Table 670. SCI control register 1 (SCIC1)

Offset[404]	0x1A						Access: User read/write	
	7	6	5	4	3	2	1	0
R W	LOOPS	DPD1	RSRC	M	DPD0	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

Notes:
404.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 671. SCIC1 field descriptions

Field	Description
7 LOOPS	**Loop Mode Select** — Selects between loop back modes and normal 2-pin full-duplex modes. When LOOPS = 1, the transmitter output is internally connected to the receiver input. 0 Normal operation — RxD and TxD use separate pins. 1 Loop mode or single-wire mode where transmitter outputs are internally connected to receiver input. (See RSRC bit.) RxD pin is not used by SCI.
6,3 DPD[1…0]	**Die to Die Post Divider** — Selection of pre divider to the SCI operation. 00 die to die clock divide by 4 01 Reserved 10 die to die clock divide by 2 11 die to die clock divide by 1
5 RSRC	**Receiver Source Select** — This bit has no meaning or effect unless the LOOPS bit is set to 1. When LOOPS = 1, the receiver input is internally connected to the TxD pin and RSRC determines whether this connection is also connected to the transmitter output. 0 Provided LOOPS = 1, RSRC = 0 selects internal loop back mode and the SCI does not use the RxD pins. 1 Single-wire SCI mode where the TxD pin is connected to the transmitter output and receiver input.
4 M	**9-Bit or 8-Bit Mode Select** 0 Normal — start + 8 data bits (LSB first) + stop. 1 Receiver and transmitter use 9-bit data characters start + 8 data bits (LSB first) + 9th data bit + stop.
2 ILT	**Idle Line Type Select** — Setting this bit to 1 ensures that the stop bit and logic 1 bits at the end of a character do not count toward the 10 or 11 bit times of logic high level needed by the idle line detection logic. Refer to Idle-line wake-up for more information. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	**Parity Enable** — Enables hardware parity generation and checking. When parity is enabled, the most significant bit (MSB) of the data character (eighth or ninth data bit) is treated as the parity bit. 0 No hardware parity generation or checking. 1 Parity enabled.
0 PT	**Parity Type** — Provided parity is enabled (PE = 1), this bit selects even or odd parity. Odd parity means the total number of 1s in the data character, including the parity bit, is odd. Even parity means the total number of 1s in the data character, including the parity bit, is even. 0 Even parity. 1 Odd parity.

6.17.2.3.3 SCI control register 2 (SCIC2)

This register can be read or written at any time.

Table 672. SCI control register 2 (SCIC2)

Offset[405]	0x1B						Access: User read/write	
	7	6	5	4	3	2	1	0
R W	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

Notes:
405.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 673. SCIC2 field descriptions

Field	Description
7 TIE	**Transmit Interrupt Enable (for TDRE)** 0 Hardware interrupts from TDRE disabled (use polling). 1 Hardware interrupt requested when TDRE flag is 1.
6 TCIE	**Transmission Complete Interrupt Enable (for TC)** 0 Hardware interrupts from TC disabled (use polling). 1 Hardware interrupt requested when TC flag is 1.
5 RIE	**Receiver Interrupt Enable (for RDRF)** 0 Hardware interrupts from RDRF disabled (use polling). 1 Hardware interrupt requested when RDRF flag is 1.
4 ILIE	**Idle Line Interrupt Enable (for IDLE)** 0 Hardware interrupts from IDLE disabled (use polling). 1 Hardware interrupt requested when IDLE flag is 1.
3 TE	**Transmitter Enable** 0 Transmitter off. 1 Transmitter on. TE must be 1 in order to use the SCI transmitter. When TE = 1, the SCI forces the TxD pin to act as an output for the SCI system. When the SCI is configured for single-wire operation (LOOPS = RSRC = 1), TXDIR controls the direction of traffic on the single SCI communication line (TxD pin). TE also can be used to queue an idle character by writing TE = 0 then TE = 1 while a transmission is in progress. Refer to Send break and queued idle for more details. When TE is written to 0, the transmitter keeps control of the port TxD pin until any data, queued idle, or queued break character finishes transmitting before allowing the pin to revert to a general purpose I/O pin.
2 RE	**Receiver Enable** — When the SCI receiver is off, the RxD pin reverts to being a general purpose port I/O pin. If LOOPS = 1 the RxD pin reverts to being a general purpose I/O pin even if RE = 1. 0 Receiver off. 1 Receiver on.
1 RWU	**Receiver Wake-up Control** — This bit can be written to 1 to place the SCI receiver in a standby state where it waits for automatic hardware detection of a selected wake-up condition. The wake-up condition is either an idle line between messages (WAKE = 0, idle-line wake-up), or a logic 1 in the most significant data bit in a character (WAKE = 1, address-mark wake-up). Application software sets RWU and (normally) a selected hardware condition automatically clears RWU. Refer to Receiver wake-up operation for more details. 0 Normal SCI receiver operation. 1 SCI receiver in standby waiting for wake-up condition.
0 SBK	**Send Break** — Writing a 1 and then a 0 to SBK queues a break character in the transmit data stream. Additional break characters of 10 or 11 (13 or 14 if BRK13 = 1) bit times of logic 0 are queued as long as SBK = 1. Depending on the timing of the set and clear of SBK relative to the information currently being transmitted, a second break character may be queued before software clears SBK. Refer to Send break and queued idle for more details. 0 Normal transmitter operation. 1 Queue break character(s) to be sent.

6.17.2.3.4 SCI status register 1 (SCIS1)

This register has eight read-only status flags. Writes have no effect. Special software sequences (which do not involve writing to this register) are used to clear these status flags.

Table 674. SCI status register 1 (SCIS1)

Offset[406]	0x1C						Access: User read/write	
	7	6	5	4	3	2	1	0
R	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
W								
Reset	1	1	0	0	0	0	0	0
		= Unimplemented or Reserved						

Notes:
406.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 675. SCIS1 field descriptions

Field	Description
7 TDRE	**Transmit Data Register Empty Flag** — TDRE is set out of reset and when a transmit data value transfers from the transmit data buffer to the transmit shifter, leaving room for a new character in the buffer. To clear TDRE, read SCIS1 with TDRE = 1 and then write to the SCI data register (SCID). 0 Transmit data register (buffer) full. 1 Transmit data register (buffer) empty.
6 TC	**Transmission Complete Flag** — TC is set out of reset and when TDRE = 1 and no data, preamble, or break character is being transmitted. 0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete). TC is cleared automatically by reading SCIS1 with TC = 1 and then doing one of the following three things: Write to the SCI data register (SCID) to transmit new data Queue a preamble by changing TE from 0 to 1 Queue a break character by writing 1 to SBK in SCIC2
5 RDRF	**Receive Data Register Full Flag** — RDRF becomes set when a character transfers from the receive shifter into the receive data register (SCID). To clear RDRF, read SCIS1 with RDRF = 1 and then read the SCI data register (SCID). 0 Receive data register empty. 1 Receive data register full.
4 IDLE	**Idle Line Flag** — IDLE is set when the SCI receive line becomes idle for a full character time after a period of activity. When ILT = 0, the receiver starts counting idle bit times after the start bit. So if the receive character is all 1s, these bit times and the stop bit time count toward the full character time of logic high (10 or 11 bit times depending on the M control bit) needed for the receiver to detect an idle line. When ILT = 1, the receiver doesn't start counting idle bit times until after the stop bit. So the stop bit and any logic high bit times at the end of the previous character do not count toward the full character time of logic high needed for the receiver to detect an idle line. To clear IDLE, read SCIS1 with IDLE = 1 and then read the SCI data register (SCID). After IDLE has been cleared, it cannot become set again until after a new character has been received and RDRF has been set. IDLE will get set only once even if the receive line remains idle for an extended period. 0 No idle line detected. 1 Idle line was detected.
3 OR	**Receiver Overrun Flag** — OR is set when a new serial character is ready to be transferred to the receive data register (buffer), but the previously received character has not been read from SCID yet. In this case, the new character (and all associated error information) is lost because there is no room to move it into SCID. To clear OR, read SCIS1 with OR = 1 and then read the SCI data register (SCID). 0 No overrun. 1 Receive overrun (new SCI data lost and no more receive/RDRF interrupts)
2 NF	**Noise Flag** — The advanced sampling technique used in the receiver takes seven samples during the start bit and three samples in each data bit and the stop bit. If any of these samples disagrees with the rest of the samples within any bit time in the frame, the flag NF will be set at the same time as the flag RDRF gets set for the character. To clear NF, read SCIS1 and then read the SCI data register (SCID). 0 No noise detected. 1 Noise detected in the received character in SCID.

Table 675. SCIS1 field descriptions (continued)

Field	Description
1 FE	**Framing Error Flag** — FE is set at the same time as RDRF when the receiver detects a logic 0 where the stop bit was expected. This suggests the receiver was not properly aligned to a character frame. To clear FE, read SCIS1 with FE = 1 and then read the SCI data register (SCID). 0 No framing error detected. This does not guarantee the framing is correct. 1 Framing error.
0 PF	**Parity Error Flag** — PF is set at the same time as RDRF when parity is enabled (PE = 1) and the parity bit in the received character does not agree with the expected parity value. To clear PF, read SCIS1 and then read the SCI data register (SCID). 0 No parity error. 1 Parity error.

6.17.2.3.5 SCI status register 2 (SCIS2)

This register has one read-only status flag.

Table 676. SCI status register 2 (SCIS2)

Offset[407] **0x1D** **Access: User read/write**

	7	6	5	4	3	2	1	0
R	LBKDIF	RXEDGIF	0	RXINV	RWUID	BRK13	LBKDE	RAF
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
407.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 677. SCIS2 field descriptions

Field	Description
7 LBKDIF	**LIN Break Detect Interrupt Flag** — LBKDIF is set when the LIN break detect circuitry is enabled and a LIN break character is detected. LBKDIF is cleared by writing a “1” to it. 0 No LIN break character has been detected. 1 LIN break character has been detected.
6 RXEDGIF	**RxD Pin Active Edge Interrupt Flag** — RXEDGIF is set when an active edge (falling if RXINV = 0, rising if RXINV=1) on the RxD pin occurs. RXEDGIF is cleared by writing a “1” to it. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.
4 RXINV[408]	**Receive Data Inversion** — Setting this bit reverses the polarity of the received data input. 0 Receive data not inverted 1 Receive data inverted
3 RWUID	**Receive Wake Up Idle Detect**— RWUID controls whether the idle character that wakes up the receiver sets the IDLE bit. 0 During receive standby state (RWU = 1), the IDLE bit does not get set upon detection of an idle character. 1 During receive standby state (RWU = 1), the IDLE bit gets set upon detection of an idle character.
2 BRK13	**Break Character Generation Length** — BRK13 is used to select a longer transmitted break character length. Detection of a framing error is not affected by the state of this bit. 0 Break character is transmitted with length of 10 bit times (11 if M = 1) 1 Break character is transmitted with length of 13 bit times (14 if M = 1)
1 LBKDE	**LIN Break Detection Enable**— LBKDE is used to select a longer break character detection length. While LBKDE is set, framing error (FE) and receive data register full (RDRF) flags are prevented from setting. 0 Break character detection disabled. 1 Break character detection enabled (11/12 bit times for M=0/1).

Table 677. SCIS2 field descriptions (continued)

Field	Description
0 RAF	**Receiver Active Flag** — RAF is set when the SCI receiver detects the beginning of a valid start bit, and RAF is cleared automatically when the receiver detects an idle line. This status flag can be used to check whether an SCI character is being received before instructing the MCU to go to stop mode. 0 SCI receiver idle waiting for a start bit. 1 SCI receiver active (RxD input not idle).

Notes:
408.Setting RXINV inverts the RxD input for all cases: data bits, start and stop bits, break, and idle.

When using an internal oscillator in a LIN system, it is necessary to raise the break detection threshold by one bit time. Under the worst case timing conditions allowed in LIN, it is possible that a 0x00 data character can appear to be 10.26 bit times long at a slave which is running 14% faster than the master. This would trigger normal break detection circuitry, which is designed to detect a 10 bit break symbol. When the LBKDE bit is set, framing errors are inhibited and the break detection threshold changes from 10 bits to 11 bits, preventing false detection of a 0x00 data character as a LIN break symbol.

6.17.2.3.6 SCI control register 3 (SCIC3)

Table 678. SCI control register 3 (SCIC3)

Offset[409] **0x1E** **Access: User read/write**

	7	6	5	4	3	2	1	0
R	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
W								
Reset	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Notes:
409.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 679. SCIC3 field descriptions

Field	Description
7 R8	**Ninth Data Bit for Receiver** — When the SCI is configured for 9-bit data (M = 1), R8 can be thought of as a ninth receive data bit to the left of the MSB of the buffered data in the SCID register. When reading 9-bit data, read R8 before reading SCID, because reading SCID completes automatic flag clearing sequences, which could allow R8 and SCID to be overwritten with new data.
6 T8	**Ninth Data Bit for Transmitter** — When the SCI is configured for 9-bit data (M = 1), T8 may be thought of as a ninth transmit data bit to the left of the MSB of the data in the SCID register. When writing 9-bit data, the entire 9-bit value is transferred to the SCI shift register after SCID is written, so T8 should be written (if it needs to change from its previous value) before SCID is written. If T8 does not need to change in the new value (such as when it is used to generate mark or space parity), it need not be written each time SCID is written.
5 TXDIR	**TxD Pin Direction in Single-wire Mode** — When the SCI is configured for single-wire half-duplex operation (LOOPS = RSRC = 1), this bit determines the direction of data at the TxD pin. 0 TxD pin is an input in single-wire mode. 1 TxD pin is an output in single-wire mode.
4 TXINV[410]	**Transmit Data Inversion** — Setting this bit reverses the polarity of the transmitted data output. 0 Transmit data not inverted 1 Transmit data inverted
3 ORIE	**Overrun Interrupt Enable** — This bit enables the overrun flag (OR) to generate hardware interrupt requests. 0 OR interrupts disabled (use polling). 1 Hardware interrupt requested when OR = 1.
2 NEIE	**Noise Error Interrupt Enable** — This bit enables the noise flag (NF) to generate hardware interrupt requests. 0 NF interrupts disabled (use polling). 1 Hardware interrupt requested when NF = 1.
1 FEIE	**Framing Error Interrupt Enable** — This bit enables the framing error flag (FE) to generate hardware interrupt requests. 0 FE interrupts disabled (use polling). 1 Hardware interrupt requested when FE = 1.

Table 679. SCIC3 field descriptions (continued)

Field	Description
0 PEIE	**Parity Error Interrupt Enable** — This bit enables the parity error flag (PF) to generate hardware interrupt requests. 0 PF interrupts disabled (use polling). 1 Hardware interrupt requested when PF = 1.

Notes:
410.Setting TXINV inverts the TxD output for all cases: data bits, start and stop bits, break, and idle.

6.17.2.3.7 SCI data register (SCID)

This register is actually two separate registers. Reads return the contents of the read-only receive data buffer and writes go to the write-only transmit data buffer. Reads and writes of this register are also involved in the automatic flag clearing mechanisms for the SCI status flags.

Table 680. SCI data register (SCID)

Offset[411]	0x1D						Access: User read/write	
	7	6	5	4	3	2	1	0
R	R7	R6	R5	R4	R3	R2	R1	R0
W	T7	T6	T5	T4	T3	T2	T1	T0
Reset	0	0	0	0	0	0	0	0

Notes:
411.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

6.17.3 Functional description

The SCI allows full-duplex, asynchronous, NRZ serial communication among the MCU and remote devices, including other MCUs. The SCI comprises a baud rate generator, transmitter, and receiver block. The transmitter and receiver operate independently, although they use the same baud rate generator. During normal operation, the MCU monitors the status of the SCI, writes the data to be transmitted, and processes received data. The following describes each of the blocks of the SCI.

6.17.3.1 Baud rate generation

Figure 108 shows the clock source for the SCI baud rate generator is the D2D clock/DPD[1:0].

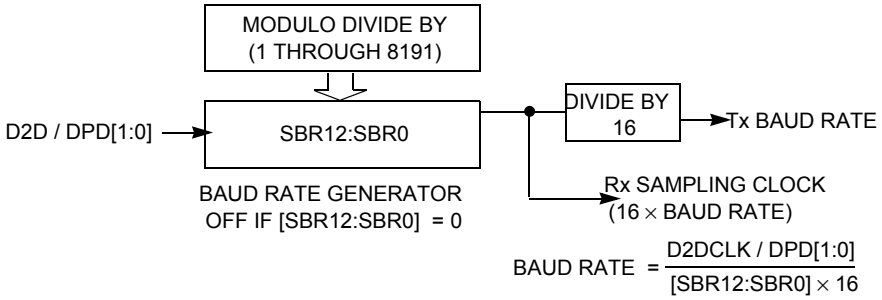


Figure 108. SCI baud rate generation

SCI communications require the transmitter and receiver (which typically derive baud rates from independent clock sources) to use the same baud rate. Allowed tolerance on this baud frequency depends on the details of how the receiver synchronizes to the leading edge of the start bit and how bit sampling is performed.

The MCU resynchronizes to bit boundaries on every high-to-low transition, but in the worst case, there are no such transitions in the full 10- or 11-bit time character frame so any mismatch in baud rate is accumulated for the whole character time. For a NXP Semiconductor SCI system whose bus frequency is driven by a crystal, the allowed baud rate mismatch is about ±4.5 percent for 8-bit data format and about ±4.0 percent for 9-bit data format. Although baud rate modulo divider settings do not always produce baud rates that exactly match standard rates, it is normally possible to get within a few percent, which is acceptable for reliable communications.

6.17.3.2 Transmitter functional description

This section describes the overall block diagram for the SCI transmitter, as well as specialized functions for sending break and idle characters. The transmitter block diagram is shown in Figure 106.

The transmitter output (TxD) idle state defaults to logic high (TXINV = 0 following reset). The transmitter output is inverted by setting TXINV = 1. The transmitter is enabled by setting the TE bit in SCIC2. This queues a preamble character that is one full character frame of the idle state. The transmitter then remains idle until data is available in the transmit data buffer. Programs store data into the transmit data buffer by writing to the SCI data register (SCID).

The central element of the SCI transmitter is the transmit shift register that is either 10 or 11 bits long depending on the setting in the M control bit. For the remainder of this section, we will assume M = 0, selecting the normal 8-bit data mode. In 8-bit data mode, the shift register holds a start bit, eight data bits, and a stop bit. When the transmit shift register is available for a new SCI character, the value waiting in the transmit data register is transferred to the shift register (synchronized with the baud rate clock) and the transmit data register empty (TDRE) status flag is set to indicate another character may be written to the transmit data buffer at SCID.

If no new character is waiting in the transmit data buffer after a stop bit is shifted out the TxD pin, the transmitter sets the transmit complete flag and enters an idle mode, with TxD high, waiting for more characters to transmit.

Writing 0 to TE does not immediately release the pin to be a general purpose I/O pin. Any transmit activity that is in progress must first be completed. This includes data characters in progress, queued idle characters, and queued break characters.

6.17.3.2.1 Send break and queued idle

The SBK control bit in SCIC2 is used to send break characters which were originally used to gain the attention of old teletype receivers. Break characters are a full character time of logic 0 (10 bit times including the start and stop bits). A longer break of 13 bit times can be enabled by setting BRK13 = 1. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 1 and then write 0 to the SBK bit. This action queues a break character to be sent as soon as the shifter is available. If SBK is still 1 when the queued break moves into the shifter (synchronized to the baud rate clock), an additional break character is queued. If the receiving device is another NXP Semiconductors SCI, the break characters will be received as 0s in all eight data bits and a framing error (FE = 1) occurs.

When idle-line wake-up is used, a full character time of idle (logic 1) is needed between messages to wake up any sleeping receivers. Normally, a program would wait for TDRE to become set to indicate the last character of a message has moved to the transmit shifter, then write 0 and then write 1 to the TE bit. This action queues an idle character to be sent as soon as the shifter is available. As long as the character in the shifter does not finish while TE = 0, the SCI transmitter never actually releases control of the TxD pin. If there is a possibility of the shifter finishing while TE = 0, set the general purpose I/O controls so the pin that is shared with TxD is an output driving a logic 1. This ensures that the TxD line will look like a normal idle line even if the SCI loses control of the port pin between writing 0 and then 1 to TE.

The length of the break character is affected by the BRK13 and M bits as shown in Table 681.

Table 681. Break character length

BRK13	M	Break Character Length
0	0	10 bit times
0	1	11 bit times
1	0	13 bit times
1	1	14 bit times

6.17.3.3 Receiver functional description

In this section, the receiver block diagram (Figure 107) is used as a guide for the overall receiver functional description. The data sampling technique used to reconstruct receiver data is then described in more detail. Finally, two variations of the receiver wake-up function are explained.

The receiver input is inverted by setting RXINV = 1. The receiver is enabled by setting the RE bit in SCIC2. Character frames consist of a start bit of logic 0, eight (or nine) data bits (LSB first), and a stop bit of logic 1. For information about 9-bit data mode, refer to 8 and 9-bit data modes. For the remainder of this discussion, we assume the SCI is configured for normal 8-bit data mode.

After receiving the stop bit into the receive shifter, and provided the receive data register is not already full, the data character is transferred to the receive data register and the receive data register full (RDRF) status flag is set. If RDRF was already set indicating the receive data register (buffer) was already full, the overrun (OR) status flag is set and the new data is lost. Because the SCI receiver is double-buffered, the program has one full character time after RDRF is set before the data in the receive data buffer must be read to avoid a receiver overrun.

When a program detects that the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF flag is cleared automatically by a 2-step sequence which is normally satisfied in the course of the user's program that handles receive data. Refer to Interrupts and status flags for more details about flag clearing.

6.17.3.3.1 Data sampling technique

The SCI receiver uses a 16× baud rate clock for sampling. The receiver starts by taking logic level samples at 16 times the baud rate to search for a falling edge on the RxD serial data input pin. A falling edge is defined as a logic 0 sample after three consecutive logic 1 samples. The 16× baud rate clock is used to divide the bit time into 16 segments labeled RT1 through RT16. When a falling edge is located, three more samples are taken at RT3, RT5, and RT7 to make sure this was a real start bit and not merely noise. If at least two of these three samples are 0, the receiver assumes it is synchronized to a receive character.

The receiver then samples each bit time, including the start and stop bits, at RT8, RT9, and RT10 to determine the logic level for that bit. The logic level is interpreted to be that of the majority of the samples taken during the bit time. In the case of the start bit, the bit is assumed to be 0 if at least two of the samples at RT3, RT5, and RT7 are 0 even if one or all of the samples taken at RT8, RT9, and RT10 are 1s. If any sample in any bit time (including the start and stop bits) in a character frame fails to agree with the logic level for that bit, the noise flag (NF) will be set when the received character is transferred to the receive data buffer.

The falling edge detection logic continuously looks for falling edges, and if an edge is detected, the sample clock is resynchronized to bit times. This improves the reliability of the receiver in the presence of noise or mismatched baud rates. It does not improve worst case analysis because some characters do not have any extra falling edges anywhere in the character frame.

In the case of a framing error, provided the received character was not a break character, the sampling logic that searches for a falling edge is filled with three logic 1 samples so that a new start bit can be detected almost immediately. The receiver is inhibited from receiving any new characters until the framing error flag is cleared. The receive shift register continues to function, but a complete character cannot transfer to the receive data buffer if FE is still set.

6.17.3.3.2 Receiver wake-up operation

Receiver wake-up is a hardware mechanism that allows an SCI receiver to ignore the characters in a message that is intended for a different SCI receiver. In such a system, all receivers evaluate the first character(s) of each message, and as soon as they determine the message is intended for a different receiver, they write logic 1 to the receiver wake up (RWU) control bit in SCIC2. When RWU bit is set, the status flags associated with the receiver (with the exception of the idle bit, IDLE, when RWUID bit is set) are inhibited from setting, thus eliminating the software overhead for handling the unimportant message characters. At the end of a message, or at the beginning of the next message, all receivers automatically force RWU to 0 so all receivers wake-up in time to look at the first character(s) of the next message.

6.17.3.3.2.1 Idle-line wake-up

When WAKE = 0, the receiver is configured for idle-line wake-up. In this mode, RWU is cleared automatically when the receiver detects a full character time of the idle-line level. The M control bit selects 8-bit or 9-bit data mode that determines how many bit times of idle are needed to constitute a full character time (10 or 11 bit times because of the start and stop bits).

When RWU is one and RWUID is zero, the idle condition that wakes up the receiver does not set the IDLE flag. The receiver wakes up and waits for the first data character of the next message which will set the RDRF flag and generate an interrupt if enabled. When RWUID is one, any idle condition sets the IDLE flag and generates an interrupt if enabled, regardless of whether RWU is zero or one.

The idle-line type (ILT) control bit selects one of two ways to detect an idle line. When ILT = 0, the idle bit counter starts after the start bit so the stop bit and any logic 1s at the end of a character count toward the full character time of idle. When ILT = 1, the idle bit counter does not start until after a stop bit time, so the idle detection is not affected by the data in the last character of the previous message.

6.17.3.3.2.2 Address-mark wake-up

When WAKE = 1, the receiver is configured for address-mark wake-up. In this mode, RWU is cleared automatically when the receiver detects a logic 1 in the most significant bit of a received character (eighth bit in M = 0 mode and ninth bit in M = 1 mode).

Address-mark wake-up allows messages to contain idle characters but requires that the MSB be reserved for use in address frames. The logic 1 MSB of an address frame clears the RWU bit before the stop bit is received and sets the RDRF flag. In this case, the character with the MSB set is received even though the receiver was sleeping during most of this character time.

6.17.3.4 Interrupts and status flags

The SCI system has three separate interrupt vectors to reduce the amount of software needed to isolate the cause of the interrupt. One interrupt vector is associated with the transmitter for TDRE and TC events. Another interrupt vector is associated with the receiver for RDRF, IDLE, RXEDGIF, and LBKDIF events, and a third vector is used for OR, NF, FE, and PF error conditions. Each of these ten interrupt sources can be separately masked by local interrupt enable masks. The flags can still be polled by software when the local masks are cleared to disable generation of hardware interrupt requests.

The SCI transmitter has two status flags that optionally can generate hardware interrupt requests. Transmit data register empty (TDRE) indicates when there is room in the transmit data buffer to write another transmit character to SCID. If the transmit interrupt enable (TIE) bit is set, a hardware interrupt will be requested whenever TDRE = 1. Transmit complete (TC) indicates that the transmitter is finished transmitting all data, preamble, and break characters and is idle with TxD at the inactive level. This flag is often used in systems with modems to determine when it is safe to turn off the modem. If the transmit complete interrupt enable (TCIE) bit is set, a hardware interrupt will be requested whenever TC = 1. Instead of hardware interrupts, software polling may be used to monitor the TDRE and TC status flags if the corresponding TIE or TCIE local interrupt masks are 0s.

When a program detects that the receive data register is full (RDRF = 1), it gets the data from the receive data register by reading SCID. The RDRF flag is cleared by reading SCIS1 while RDRF = 1 and then reading SCID.

When polling is used, this sequence is naturally satisfied in the normal course of the user program. If hardware interrupts are used, SCIS1 must be read in the interrupt service routine (ISR). Normally, this is done in the ISR anyway to check for receive errors, so the sequence is automatically satisfied.

The IDLE status flag includes logic that prevents it from getting set repeatedly when the RxD line remains idle for an extended period of time. IDLE is cleared by reading SCIS1 while IDLE = 1 and then reading SCID. After IDLE has been cleared, it cannot become set again until the receiver has received at least one new character and has set RDRF.

If the associated error was detected in the received character that caused RDRF to be set, the error flags — noise flag (NF), framing error (FE), and parity error flag (PF) — get set at the same time as RDRF. These flags are not set in overrun cases.

If RDRF was already set when a new character is ready to be transferred from the receive shifter to the receive data buffer, the overrun (OR) flag gets set instead the data along with any associated NF, FE, or PF condition is lost.

At any time, an active edge on the RxD serial data input pin causes the RXEDGIF flag to set. The RXEDGIF flag is cleared by writing a “1” to it. This function does depend on the receiver being enabled (RE = 1).

6.17.3.5 Additional SCI functions

The following sections describe additional SCI functions.

6.17.3.5.1 8 and 9-bit data modes

The SCI system (transmitter and receiver) can be configured to operate in 9-bit data mode by setting the M control bit in SCIC1. In 9-bit mode, there is a ninth data bit to the left of the MSB of the SCI data register. For the transmit data buffer, this bit is stored in T8 in SCIC3. For the receiver, the ninth bit is held in R8 in SCIC3.

For coherent writes to the transmit data buffer, write to the T8 bit before writing to SCID.

If the bit value to be transmitted as the ninth bit of a new character is the same as for the previous character, it is not necessary to write to T8 again. When data is transferred from the transmit data buffer to the transmit shifter, the value in T8 is copied at the same time data is transferred from SCID to the shifter.

9-bit data mode typically is used in conjunction with parity to allow eight bits of data plus the parity in the ninth bit. Or it is used with address-mark wake-up so the ninth data bit can serve as the wake-up bit. In custom protocols, the ninth bit can also serve as a software-controlled marker.

6.17.3.5.2 Stop/sleep mode operation

The SCI is powered down in stop and sleep mode. It is required to re-initialize the SCI module after a wake-up from stop or sleep mode.

6.17.3.5.3 Loop mode

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Loop mode is sometimes used to check software, independent of connections in the external system, to help isolate system problems. In this mode, the transmitter output is internally connected to the receiver input and the RxD pin is not used by the SCI, so it reverts to a general purpose port I/O pin.

6.17.3.5.4 Single-wire operation

When LOOPS = 1, the RSRC bit in the same register chooses between loop mode (RSRC = 0) or single-wire mode (RSRC = 1). Single-wire mode is used to implement a half-duplex serial connection. The receiver is internally connected to the transmitter output and to the TxD pin. The RxD pin is not used and reverts to a general purpose port I/O pin.

In single-wire mode, the TXDIR bit in SCIC3 controls the direction of serial data on the TxD pin. When TXDIR = 0, the TxD pin is an input to the SCI receiver and the transmitter is temporarily disconnected from the TxD pin so an external device can send serial data to the receiver. When TXDIR = 1, the TxD pin is an output driven by the transmitter. In single-wire mode, the internal loop back connection from the transmitter to the receiver causes the receiver to receive characters that are sent out by the transmitter.

6.18 LIN

6.18.1 Introduction

The LIN bus pin provides a physical layer for single-wire communication in automotive applications. The LIN physical layer is designed to meet the LIN physical layer version 2.2 and J2602 specification, and has the following features:

- LIN physical layer 2.2 / J2602 compliant
- Slew rate selection 20 kBit, 10 kBit, and Fast mode (100 kBit)
- Overtemperature shutdown - HTI
- Permanent pull-up in Normal mode 30 kΩ, 1.0 MΩ in low power
- Current limitation
- Special J2602 compliant configuration
- Direct Rx / Tx access
- Optional external Rx / Tx access and routing to the TIMER Input through PTBx
- TXD dominant timeout detection

The LIN driver is a low side MOSFET with current limitation and thermal shutdown. An internal pull-up resistor with a serial diode structure is integrated, so no external pull-up components are required for the application in a slave node. The fall time from dominant to recessive and the rise time from recessive to dominant is controlled. The symmetry between both slopes is guaranteed.

6.18.2 Overview

6.18.2.1 Block diagram

Figure 109 shows the basic function of the LIN module.

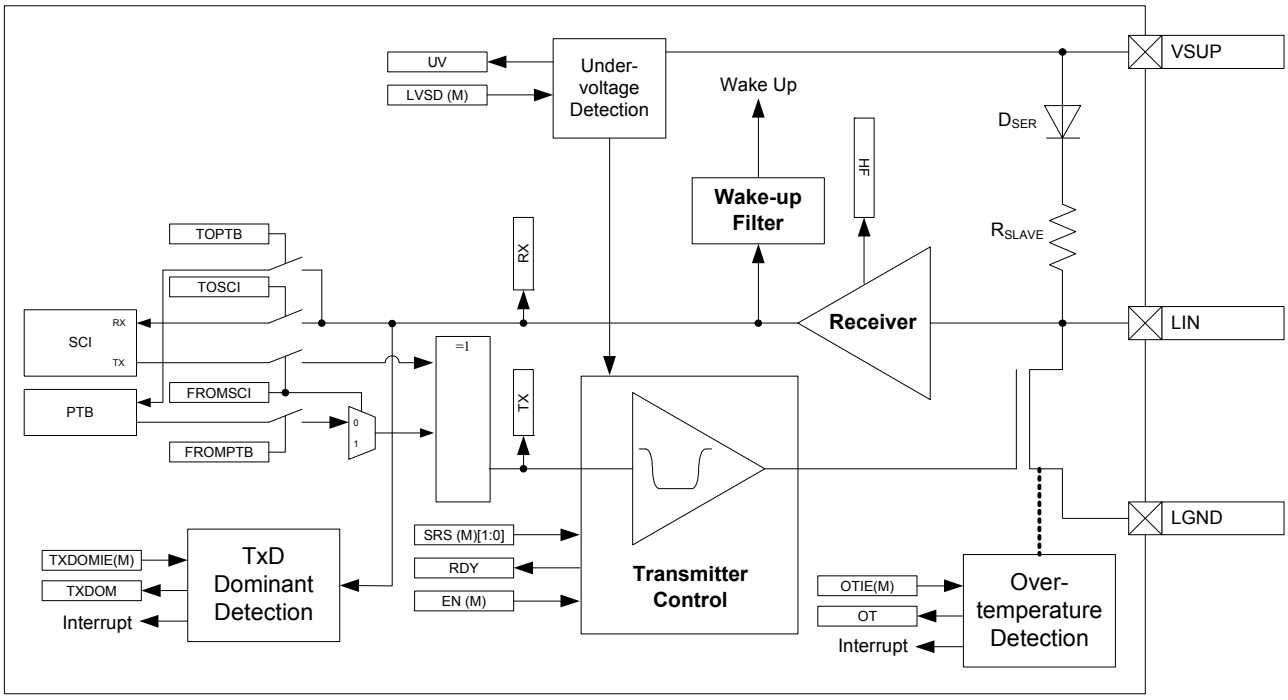


Figure 109. LIN module block diagram

6.18.2.2 LIN pin

The LIN pin offers high susceptibility immunity level from external disturbance, guaranteeing communication during external disturbances. See Electromagnetic compatibility (EMC).

6.18.2.3 Slew rate selection

The slew rate can be selected for optimized operation at 10 kBit/s and 20 kBit/s as well as a fast baud rate (100 kBit/s) for test and programming. The slew rate can be adapted with the bits SRS[1:0] in the LIN Control Register (LIN_CTL). The initial slew rate is 20 kBit/s.

6.18.2.4 Overtemperature shutdown (LIN interrupt)

The output low side FET (transmitter) is protected against overtemperature conditions. In an overtemperature condition, the transmitter will be shut down, and the OT bit in the LIN Control Register (LIN_CTL) is set as long as the condition is present.

If the OTIE bit is set in the LIN Status Register (LIN_SR), an Interrupt IRQ will be generated. Acknowledge the interrupt by writing a “1” in the LIN Status Register (LIN_SR). To issue a new interrupt, the condition has to vanish and reoccur.

The transmitter is automatically re-enabled once the overtemperature condition has ceased and TxD is High.

6.18.2.5 TxD dominant detection (LIN interrupt)

The LIN Physical Layer (transmitter) is protected against TxD Dominant conditions. In a TxD Dominant condition, longer than t_{TxDDOM}, the LIN transmitter will be turned off, so that the LIN line will be released to the recessive state.

If the TXDOMIE bit is set in the LIN Status Register, an interrupt LIN IRQ will be generated when a TxD Dominant Condition is detected.

When LIN IRQ is cleared, and the TxD Dominant condition persists, a new interrupt will not be generated, a new interrupt will only be generated if the TxD Dominant condition is removed and then reoccurs.

6.18.2.6 Low-power mode and wake-up feature

During Low-power mode operation, the transmitter of the physical layer is disabled. The receiver is still active and able to detect wake-up events on the LIN bus line.

A dominant level longer than t_{PROPWL}, followed by a rising edge will generate a wake-up event and will be reported in the PCR status register, bit WULINF.

6.18.2.7 J2602 compliance

A Low Voltage Shutdown feature allows controlled LIN driver behavior under low voltage conditions at VSUP. If LVSD is set, once VSUP is below the threshold VJ2602H, the LIN transmitter is not turned dominant again. The condition is indicated by the UV flag.

6.18.2.8 Transmit / receiving line definition

The LIN module can be connected to the SCI or PTB module.

The module includes a TxD permanent dominant detection, in order to disable the LIN bus driver if TxD signal (internal RX signal is monitored) is low for a time longer than 6.0 ms (typ).

6.18.2.9 Transmitter enable / ready

The LIN transmitter must be enabled before transmission is possible (EN). The RDY bit is set to 1 about 50 µs after the LIN transmitter is enabled. This is due to the initialization time for the LIN transmitter, under some low voltage conditions.

During this period (LIN enabled to RDY = 1), the LIN is forced to a recessive state.

6.18.3 Memory map and registers

6.18.3.1 Overview

This section provides a detailed description of the memory map and registers.

6.18.3.2 Module memory map

The memory map for the LIN module is given in Table 60

Bit Legend

= Unimplemented, Reserved

= Implemented (do not alter)

X = Indeterminate

0 = Always read zero

Table 682. Module memory map

Offset (412),(413)	Name		7	6	5	4	3	2	1	0
0x50	LIN_CTL LIN control register	R	0	0	0	0	0	0	0	0
		W	OTIEM	TXDOMIEM			LVSDM	ENM	SRSM	
		R	OTIE	TXDOMIE	0		LVSD	EN	SRS	
		W								
0x52	LIN_SR (hi)	R	OT	TXDOM	HF	0	UV	0	0	0
	LIN status register	W	Write 1 will clear the flags							
0x53	LIN_SR (lo)	R	RDY	0	0	0	0	0	RX	TX
	LIN status register	W								

MM9Z1_638

Table 682. Module memory map (continued)

Offset (412),(413)	Name		7	6	5	4	3	2	1	0
0x54	LIN_TX	R	0	0	0	0	0	0	FROMPTB	FROMSCI
	LIN transmit line definition	W								
0x55	LIN_RX	R	0	0	0	0	0	0	TOPTB	TOSCI
	LIN receive line definition	W								
0x56	Reserved	R	0	0	0	0	0	0	0	0
		W								
0x57	Reserved	R	0	0	0	0	0	0	0	0
		W								

Notes:
412.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
413.This Register is 16-Bit access only.

6.18.3.3 Register descriptions

This section consists of register descriptions in address order. Each description includes a standard register diagram with an associated figure number. Details of the register bit and field function follow the register diagrams, in bit order.

6.18.3.3.1 LIN control register (LIN_CTL)

Bit Legend

= Unimplemented, Reserved

= Implemented (do not alter)

X = Indeterminate

0 = Always read zero

Table 683. LIN control register (LIN_CTL)

Offset (414) ,(415)	0x50					Access: User write		
	15	14	13	12	11	10	9	8
R	0	0	0	0	0	0	0	0
W	OTIEM	TXDOMIEM			LVSDM	ENM	SRSM	
Reset	0	0	0	0	0	0	0	0
	7	6	5	4	3	2	1	0
R	OTIE	TXDOMIE	0		LVSD	EN	SRS	
W								
Reset	0	0	0	0	0	0	0	0

Notes:
414.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.
415.This Register is 16-Bit access only.

Table 684. LIN control register (LIN_CTL) - register field descriptions

Field	Description
15 OTIEM	**LIN Overtemperature Interrupt Enable** - Mask 0 - writing the OTIE Bit will have no effect 1 - writing the OTIE Bit will be effective
14 TXDOMIEM	**LIN - TxD Permanent Dominant** - Mask 0 - writing the TXDOMIEM Bit will have no effect 1 - writing the TXDOMIEM Bit will be effective
12	
11 LVSDM	**LIN - Low Voltage Shutdown Disable (J2602 Compliance Control)** - Mask 0 - writing the LVSD Bit will have no effect 1 - writing the LVSD Bit will be effective
10 ENM	**LIN Module Enable** - Mask 0 - writing the EN Bit will have no effect 1 - writing the EN Bit will be effective
9-8 SRSM[1:0]	**LIN - Slew Rate Select** - Mask 00,01,10 - writing the SRS Bits will have no effect 11 - writing the SRS Bits will be effective
7 OTIE	**LIN Overtemperature Interrupt Enable** 0 - LIN overtemperature interrupt disabled 1 - LIN overtemperature interrupt enabled
6 TXDOMIE	**LIN TxD permanent dominant Interrupt Enable** 0 - LIN TxD permanent dominant interrupt disabled 1 - LIN TxD permanent dominant interrupt enabled
4	
3 LVSD	**LIN - Low Voltage Shutdown Disable (J2602 Compliance Control)** 0 - LIN will be remain in recessive state in case of V_{SUP} undervoltage condition 1 - LIN will stay functional even with a V_{SUP} undervoltage condition
2 EN	**LIN Module Enable** 0 - LIN module disabled 1 - LIN module enabled
1-0 SRS[1:0]	**LIN - Slew Rate Select** 00 - Normal slew rate (20 kBit) 01 - Slow slew rate (10.4 kBit) 10 - Fast slew rate (100 kbit) 11 - normal Slew Rate (20 kBit)

6.18.3.3.2 LIN status register (LIN_SR (hi))

Table 685. LIN status register (LIN_SR (hi))

Offset[416]	**0x52**						**Access: User read/write**	
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	OT	TXDOM	HF	0	UV	0	0	0
W	Write 1 will clear the flags							
Reset	0	0	0	0	0	0	0	0

Notes:
416.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 686. LIN status register (LIN_SR (hi)) - register field descriptions

Field	Description
7 OT	**LIN Overtemperature Status.** This bit is latched and has to be reset by writing 1 into OT bit. 0 - No LIN overtemperature condition detected 1 - LIN overtemperature condition detected
6 TXDOM	**LIN TxD permanent Dominant Status.** This bit is latched and has to be reset by writing 1 into TXDOM bit. 0 - no TxD permanent Dominant condition detected 1 - TxD permanent Dominant condition detected
5 HF	**LIN HF (High Frequency) Condition Status indicating HF (DPI) disturbance in the LIN module.** This bit is latched and has to be reset by writing 1 into HF bit. 0 - No LIN HF (DPI) condition detected 1 - LIN HF (DPI) condition detected
3 UV	**LIN Undervoltage Status.** This threshold is used for the J2602 feature as well. This bit is latched and has to be reset by writing 1 into UV bit. 0 - No LIN undervoltage condition detected 1 - LIN undervoltage condition detected

6.18.3.3.3 LIN status register (LIN_SR (lo))

Table 687. LIN status register (LIN_SR (lo))

Offset[417]	0x53						Access: User read	
	7	6	5	4	3	2	1	0
R	RDY	0	0	0	0	0	RX	TX
W								

Notes:
417.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 688. LIN status register (LIN_SR (lo)) - register field descriptions

Field	Description
1 RDY	**Transmitter Ready Status** 0 - Transmitter not ready 1 - Transmitter ready
1 RX	**Current RX status** 0 - Rx recessive 1 - Rx dominant
0 TX	**Current TX status** 0 - Tx recessive 1 - Tx dominant

6.18.3.3.4 LIN transmit line definition (LIN_TX)

Table 689. LIN Transmit Line Definition (LIN_TX)

Offset[418]	0x54						Access: User read/write	
	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	FROMPTB	FROMSCI
W								
Reset	0	0	0	0	0	0	0	0

Notes:
418.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 690. LIN transmit line definition (LIN_TX) - register field descriptions

Field	Description
1 FROMPTB	**LIN_TX internally routed from PTB.** See General purpose I/O - GPIO for details.(419) 0 - LIN transmitter disconnected from PTB module. 1 - LIN transmitter connected to the PTB module.
0 FROMSCI	**LIN_TX internally routed from SCI** (419) 0 - LIN transmitter disconnected from SCI module. 1 - LIN transmitter connected to the SCI module.

Notes:
419.In case both, FROMPTB and FROMSCI are selected, the SCI has priority and the PTB signal is ignored.

6.18.3.3.5 LIN receive line definition (LIN_RX)

Table 691. LIN Receive Line Definition (LIN_RX)

Offset(420) **0x55** **Access: User read/write**

	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	TOPTB	TOSCI
W								
Reset	0	0	0	0	0	0	0	0

Notes:
420.Offset related to 0x0E00 for blocking access and 0x0F00 for non blocking access within the global address space.

Table 692. LIN receive line definition (LIN_RX) - register field descriptions

Field	Description
1 TOPTB	**LIN_RX internally routed to PTB** 0 - LIN receiver disconnected from PTB module. 1 - LIN receiver connected to the PTB module.
0 TOSCI	**LIN_RX internally routed to SCI** 0 - LIN receiver disconnected from SCI module. 1 - LIN receiver connected to the SCI module.

Note

In order to route the RX signal to the Timer Input capture, one of the PTBx must be configured as a pass through.

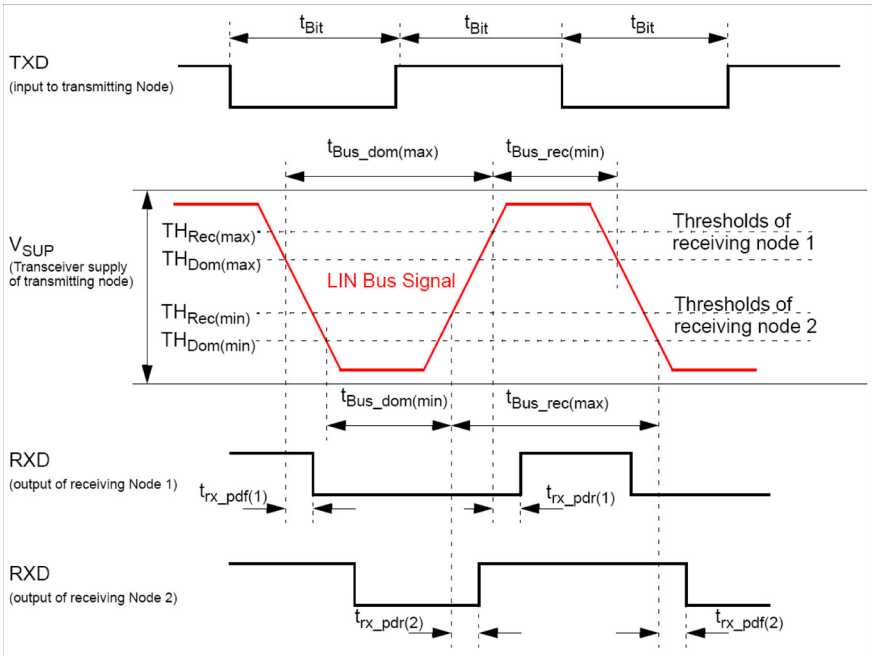
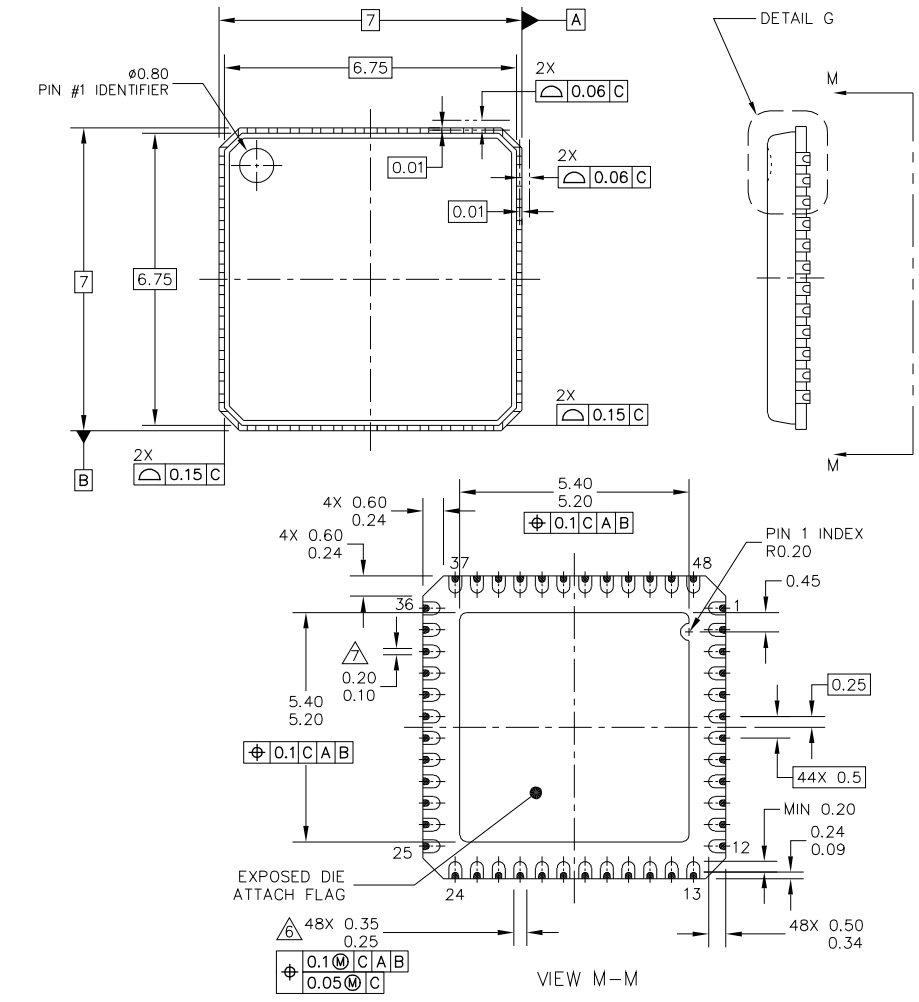


Figure 110. Definition of LIN bus timing parameters

7 Packaging

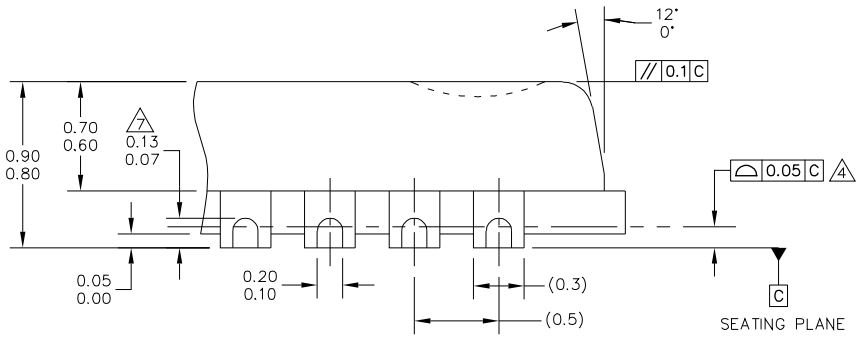
7.1 Package dimensions

For the most current package revision, visit www.nxp.com and perform a keyword search using the “98A” listed below.



© NXP SEMICONDUCTORS N.V. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON–LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D	REV: B
	STANDARD: NON–JEDEC	
	SOT619–16	07 JAN 2016

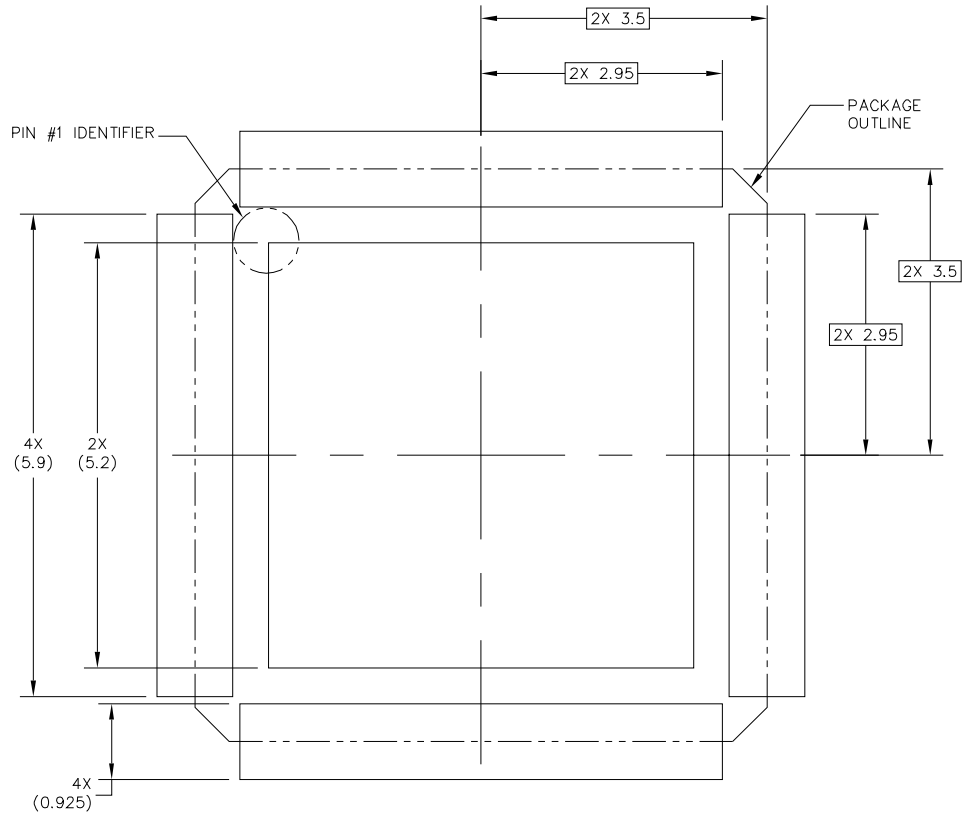
MM9Z1_638



DETAIL G
VIEW ROTATED 90° CW

© NXP SEMICONDUCTORS N.V.
ALL RIGHTS RESERVED

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON-LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D REV: B
	STANDARD: NON-JEDEC
	SOT619-16 07 JAN 2016



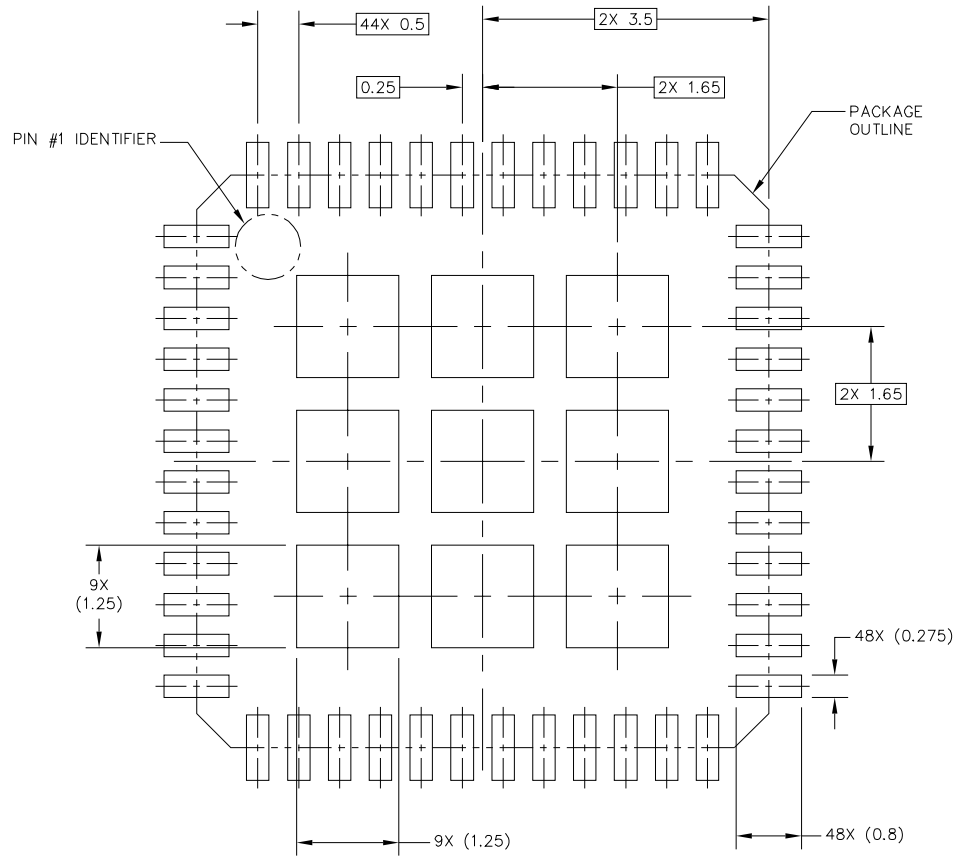
PCB DESIGN GUIDELINES – SOLDER MASK OPENINGS

THIS SHEET SERVES ONLY AS A GUIDELINE TO HELP DEVELOP A USER SPECIFIC SOLUTION. DEVELOPMENT EFFORT WILL STILL BE REQUIRED BY END USERS TO OPTIMIZE PCB MOUNTING PROCESSES AND BOARD DESIGN IN ORDER TO MEET INDIVIDUAL / SPECIFIC REQUIREMENTS.

© NXP SEMICONDUCTORS N.V.
ALL RIGHTS RESERVED

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON-LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D REV: B
	STANDARD: NON-JEDEC
	SOT619-16 07 JAN 2016

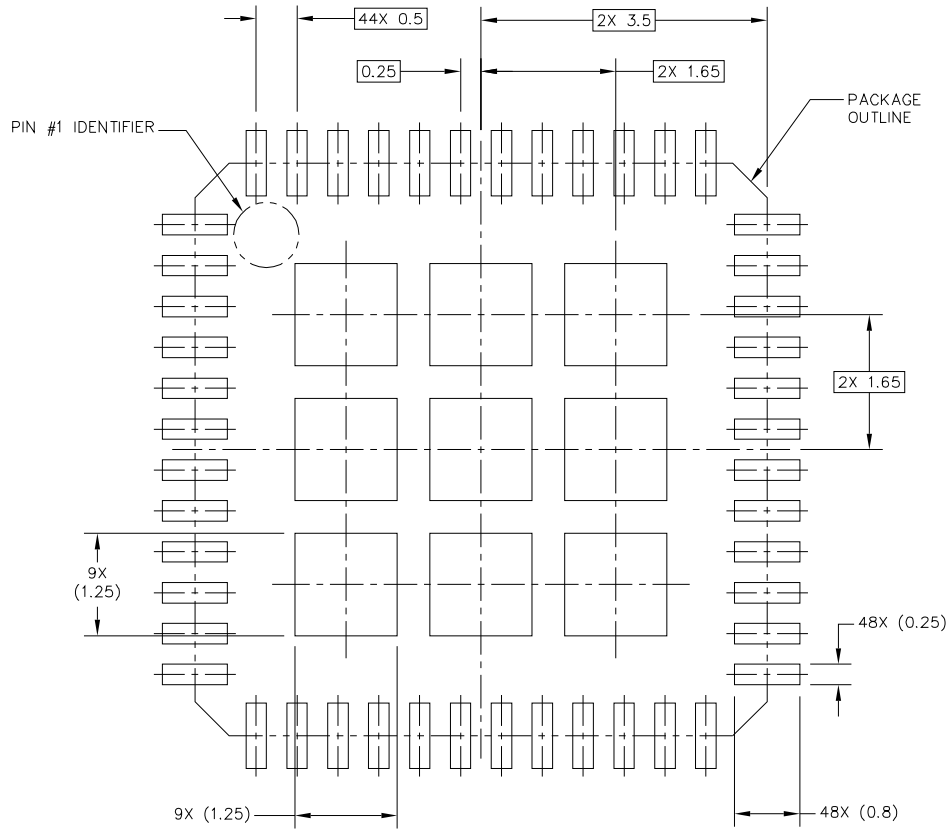


PCB CU GUIDELINES – I/O PADS & SOLDERABLE AREAS

THIS SHEET SERVES ONLY AS A GUIDELINE TO HELP DEVELOP A USER SPECIFIC SOLUTION. DEVELOPMENT EFFORT WILL STILL BE REQUIRED BY END USERS TO OPTIMIZE PCB MOUNTING PROCESSES AND BOARD DESIGN IN ORDER TO MEET INDIVIDUAL / SPECIFIC REQUIREMENTS.

© NXP SEMICONDUCTORS N.V.
ALL RIGHTS RESERVED

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON–LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D REV: B
	STANDARD: NON–JEDEC
	SOT619–16 07 JAN 2016



SOLDER PASTE STENCIL DESIGN GUIDELINES

THIS SHEET SERVES ONLY AS A GUIDELINE TO HELP DEVELOP A USER SPECIFIC SOLUTION. DEVELOPMENT EFFORT WILL STILL BE REQUIRED BY END USERS TO OPTIMIZE PCB MOUNTING PROCESSES AND BOARD DESIGN IN ORDER TO MEET INDIVIDUAL / SPECIFIC REQUIREMENTS.

© NXP SEMICONDUCTORS N.V. ALL RIGHTS RESERVED	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON-LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D	REV: B
	STANDARD: NON-JEDEC	
	SOT619-16	07 JAN 2016

NOTES:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.
3. THIS IS NON JEDEC REGISTERED PACKAGE.
4. COPLANARITY APPLIES TO LEADS, CORNER LEADS AND DIE ATTACH FLAG.
5. MIN METAL GAP BETWEEN TERMINAL AND DIE PADDLE SHALL BE 0.25MM.
6. THIS DIMENSION APPLIES TO PLATED TERMINAL AND IS MEASURED BETWEEN 0.20 AND 0.25MM FROM TERMINAL TIP.
7. THIS DIMENSION APPLIES TO PLATED TERMINAL AND IS MEASURED BETWEEN 0.02MM AND PACKAGE CUTTING LINE.

© NXP SEMICONDUCTORS N.V.
ALL RIGHTS RESERVED

MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE

TITLE: THERMALLY ENHANCED PUNCH QUAD FLAT NON–LEADED (QFN) PACKAGE WITH WETTABLE FLANKS 48 TERMINAL, 7 X 7 X 0.85, 0.5 PITCH	DOCUMENT NO: 98ASA00343D REV: B
	STANDARD: NON–JEDEC
	SOT619–16 07 JAN 2016

8 Revision history

Revision	Date	Description of changes
1.0	1/2013 2/2013	• Initial release. This release joins the Analog, Digital, and Packaging sections into a single document. • Fixed document to standard Freescale format. No changes to content were made.
2.0	5/2013	• Added targeted applications - page 1 • Added wettable flanks - page 1 • Updated description of PTB4, VSENSE[3..1], RESET and RESETA pins into pin description section • Updated Typical Application Schematics (added connection of RESET to RESETA) • Merged current consumption of embedded MCU and Analog die into a unique table • Changed limit of Undervoltage Cranking Interrupt (UVI) Deassert (measured on VSUP) Cranking Mode Enabled • Changed typo on Resistor Threshold for OPEN Detection unit from mΩ to MΩ • Changed limits of Static Electrical Characteristics - Current Sense Module • Changed limits of Static Electrical Characteristics - Voltage Sense Module • Changed limits of Static Electrical Characteristics - PTB0 to 4 voltage and temperature measurements • Changed limits of Low Power Oscillator Tolerance over temperature range • Changed LIN configuration into ESD GUN - ISO10605(35), powered, contact discharge, C_{ZAP} = 330 pF, R_{ZAP} = 2.0 kΩ • Corrected Low Pass Filter Coefficient Ax - Reset Values • Updated description of PTB4 into several chapters • Added requirement to wait that PLL is locked to access to D2D at start-up of after wake-up • Updated ampere hour counter description • Added description of Life Time Counter configuration to be done into normal mode only • Added description of Low Power Current Trigger Counter • Added description of DPD[1..0] registers into overview of SCI block • Corrected LIN Module Block Diagram • Removed references and functions related to XGATE into Interrupt (S12ZINTV0) chapter • Changed limits of VDDRX Low Voltage Reset Deassert

Revision	Date	Description of changes
3.0	10/2013	• Complete reformatting of the document: Moved all functional description blocks into chapter 5. Moved all parametrics data into Chapter 3. • Removed references to MM9Z1_638D2. • Updated Ordering Information. • Added PTB5 Protection Zener Diode into External Components • Updated Ordering Information with two part numbers: MM9Z1J638BM2EP and MM9Z1I638BM2EP • Updated limits of VSDRIFT • Updated Voltage Sense Module parameters • Updated Wake-Up Filter Times 1 and 2 • Updated VDDH/VDDA/VDDX High Temperature Warning (HTI) • Changed VUVIL limits • Changed VUVIH limits • Changed fNVMBUS limits • Changed VLVIA limits • Changed VLVID limits • Updated IFR content: trim and compensation values • Added descriptions of the IFR content: trim and com • Harmonized max ratings of VDDRX and VDDX • Harmonized max ratings of VDDD2D and VDDH • Harmonized Storage Temperature Range • Added Access to ACQ and COMP registers description • normal mode current max = 40mA • Updated stop/sleep/pseudo stop consumption combined for the product • Updated IRC frequency • Updated Low Power Oscillator Frequency • Description updated to fit current implementation on current ampere hour counter: The accumulator could be reset by writing 1 into AHCR register. The Ampere Hour Counter is counting after wake up. In normal mode, the accumulator register ACQ_AHC can be read out any time.
	3/2014	• Removed Xtrinsic logo and document type from Product Preview to Advance Information. No other change to the document.
	5/2014	• Corrected footer typographic error. No other change to the document.

Revision	Date	Description of changes
4.0	10/2016	• Updated document to NXP form and style. • Various clarifications, corrections • updated all frequency values related to 1.024 MHz IRC clock (e.g. f_{BUS} 50 MHz -> 51.2 MHz) • Extended external oscillator to 16.384 MHz • Corrected all wrong references to VDDA -> VDDRX • Updated Figure 1 (VSENSE0 -> VSENSE2) • Updated Table 2, Pin 21 description • Updated Typical applications • Updated Figure 5, Typical application example • Updated Table 3, External components • Updated Tables 17, 19, 20, 21, and 22 • Updated Table 38, I/O characteristics • Updated Table 60, Analog die registers • Updated Figure 29, Operating mode transitions • Updated Stop mode • Updated Figure 33 Wake-up Sources • Updated Timed wake-up • Updated Wake-up from LIN • Updated Wake-up on PTB4 external wake-up pin • Updated General wake-up indicator • Updated System resets • Updated Figure 40, Device Reset Overview • Updated Shunt sense • Updated Programmable gain amplifier (PGA) • Updated Latency and throughput (sampling rate) • Reformatted and corrected Table 144, Acquisition channel compensation values • Corrected Table 200 (Shunt Resistor Typ. Value selection) • Updated General purpose I/O - GPIO • Updated Introduction • Updated Features • Updated Figure 50, General purpose I/O - block diagram • Updated Figure 51, PTB4 input structure • Updated Table 243, Port 4 Input configuration (GPIO_IN4) • Updated Mode of operation • Corrected Table 536, 550 (deleted bit DFDIE from register FERCNFG) • Updated Serial communication interface (S08SCIV4), corrected baud rate divisor (bits DPD[1:0]) • Updated Table 677, added comment that SCIS1 bit OR blocks further RDRF interrupts • Updated Table 679, corrected SCIS2 bit LBKDE description • Corrected Stop/sleep mode operation • Updated LIN • Updated Figure 109 and text (TXD dom. Detection clarifications)

How to Reach Us:

Home Page:
NXP.com

Web Support:
http://www.nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no expressed or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation, consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by the customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address:

http://www.nxp.com/terms-of-use.html.

NXP, the NXP logo, Freescale, the Freescale logo and SMARTMOS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. All rights reserved.

© 2016 NXP B.V.

Document Number: MM9Z1_638D1
Rev. 4.0
10/2016