*Flex*ROM III User's Manual

Version 1.0

(This page intentionally blank)

Table of Contents

Introduction

Thank you for purchasing a TechTools product. We make every attempt to provide quality tools at reasonable prices. If you have any questions, comments or suggestions, please feel free to contact us by FAX, Voice, email or mail and express your opinions.

How To Reach TechTools

You can reach us at any of the following:

Voice:	(972) 272-9392	Monday-Friday, 9:00-5:00 Central
FAX:	(972) 494-5814	
email:	support@tech-tools.com	
web:	http://www.tech-tools.com	
Mail:	TechTools PO Box 462101 Garland, TX 75046-2101	

Up and Running in 15 Minutes!

As engineers, we understand that you do not have time to troubleshoot your test equipment. You need to be able to trust your test equipment. With that in mind, we thoroughly test each and every *FlexROM* III to ensure that you receive a fully functional unit. In addition, we provide a self-test function to enable you to verify proper emulator operation at any time.

The following sections are designed to get you up and running in less than 15 minutes. If your system is not running by the time you complete these instructions, please contact our technical support staff. We have many thousand emulators in use throughout the world. Many have been in service for nearly 10 years. We are confident that we can resolve the problem through email or over the telephone.

Install the Software

The *FlexROM* III software may come on floppy, CDROM or BOTH. If a floppy was included with your emulator, it may be a complete install or a patch. Follow the instructions on the floppy.

If the floppy contains a PATCH, you will be instructed to install the software from the CDROM and THEN apply the patch.

If the floppy contains a full install, the CDROM is not needed and any FlexROM III software it contains is out-dated. We still include the CDROM because it contains a wealth of information.

If a floppy disk was NOT included, then the software on the CDROM was the most current at the time the product was stocked. Follow the directions on the CDROM to install the software. If your system is configured to auto-launch applications from the CDROM, our launch-pad will appear after you insert the CDROM into your drive. Simply select the FlexROM III software option from there.

In any event, it is always advisable to check our web pages for any updates. We always post the most recent versions of our software for free download.

Disable Any In-Circuit Programming Voltages

EPROMs and some FLASH devices are usually programmed with "super voltages" (> 5V) and are designed to handle such voltages on some pins; *FlexROM III* is not! If your target is capable of in-circuit programming, disable or isolate the high voltage, not just the high voltage control signals. Even short surges during power cycling could be damaging.

FlexROM III is a 3-5volt ONLY device. Any voltage greater than 5 Volts or less than 0 Volts will damage it.

Install an ACM module

FlexROM III uses plug-in modules to configure it to support 8bit, 16bit, dual-8bit or custom applications. We call these Active Cable Modules (ACMs) because they contain all of the target interface, control and data steering logic. This design partition allows us to adapt FlexROM III to new target technologies with new ACMs. It also allows you to support several different targets with a single emulator.

Select the ACM appropriate for your target device.

Configure the ACM

Refer to the documentation that came with your particular ACM module to set any configuration jumpers required to match your target device.

Attach Target Interface cable to the ACM module.

Lay the ACM on a flat, firm surface and plug the target interface cable into it.
NEVER attempt to insert or remove the target interface cable while the ACM is plugged into the emulator. The circuit boards and connectors are not designed to handle this kind of mechanical stress.

Insert the ACM into the FlexROM III.

Slide the ACM into the card slots visible from the rear of the emulator. Press the ACM in firmly until fully seated. When properly installed, a portion of the ACM will extend beyond the card-guides. This provides clearance for the target interface cable and for your access to the feature connector, reset connector, etc.

Plug the emulator into your target

Turn the target power off. Plug the ACM target interface cable into the target's memory socket. Be careful to align the PIN 1 end of the cable (designated with a stripe) with the PIN 1 end of the target's socket (usually designated with a notch, a dot or a '1').

FlexROM III ACMs ship with DIP cables. If you are emulating PLCC type devices, you will need to purchase a DIP to PLCC adapter. These are available from TechTools as well as all popular adapter manufacturers.

PLUGGING THE CABLE IN BACKWARDS WILL DAMAGE *FlexROM* III!

The number one cause of emulator failure is reverse voltage. The JEDEC standard pin-out for DIP devices places power and ground on opposite corners of the device. If one plugs an EPROM or an EPROM EMULATOR in backwards, it applies reverse voltage to the device or emulator, causing permanent damage.

We added a new reverse-polarity protection circuit to FlexROM III which should greatly reduce the risk of damage. However, we still strongly recommend that you heed these warnings and make every attempt to avoid plugging the emulator in backwards.

If your target uses DIP sockets or a TechTools' PLCC adapter, it is fairly easy to ensure the cable is plugged in properly. Simply check (twice) that the RED stripe on the cable is aligned with the PIN 1 end of the socket.

If you are using any type of adapter between the DIP cable and your target, it can get a little more confusing. An incorrect adapter or one that is installed incorrectly could apply reverse voltages to the emulator.

If you have ANY doubts, check it out! The simplest way to check out adapters, is to insert the adapter into the target but do NOT insert the DIP cable into the adapter. The DIP socket provided by the adapter should have GROUND on one corner pin and POWER on the opposite corner. On 28 pin sockets, ground should be on pin 14 and power on pin 28. On 32 pin sockets, ground should be on pin 16 and power on pin 32.

Once you have established that the adapter is wired correctly and installed correctly, then insert the DIP cable into the adapter, aligning its RED stripe with the PIN 1 end of the DIP socket.

PLUGGING THE CABLE IN BACKWARDS WILL DAMAGE *FlexROM* III!

The old carpenters' rule applies here: 'MEASURE twice – CUT once.'

Connect RESET

The RESET line must be connected to guarantee proper target operation. The reset signal is automatically asserted during each emulator access. This prevents the target from "running stupid" while we change its code space. The reset signal is released at the completion of the transfer, allowing the target to restart with the new code image. We also use the reset line to prevent the target from attempting to execute code before the emulator recognizes that the target has sufficient voltage to run.

FlexROM III provides two reset outputs. RESET is an open emitter, ACTIVE HIGH output. /RESET is an open collector ACTIVE LOW output. Both signals are pulled to the in-active state via very weak resisters. Connect the appropriate output to your target's RESET circuitry.

FlexROM III's reset signal should be connected to the target's reset circuitry at the most "upstream" point available. This is usually an RC combination that senses the target's power up or a manual reset switch. Sometimes this circuitry is connected directly to the CPU reset input. In other targets, it will be connected to the input to a reset chip, power management chip, a watchdog timer chip or a Schmitt trigger gate.

DO NOT CONNECT *FlexROM* III's RESET OUTPUT ACROSS A TOTEM-POLE OR BIPOLAR GATE, OR DIRECTLY TO GROUND OR 5VOLTS. THIS MAY RESULT IN DAMAGE TO *FlexROM* III OR THE TARGET.

If your target has a reset button, determining where to connect the reset line is easy. Use a DVM, SCOPE or logic probe to measure the voltage of both sides of the switch. They should be different (one HIGH and one GROUND). Push and hold the button and take the two measurements again. Both should now be the same (HIGH or GROUND). Notice that one side of the switch remained the same, the other side changed states. Connect the reset lead to the side that CHANGED STATES. The level at this point indicates which POLARITY (RESET or /RESET) we should use. Ground indicates active LOW; use /RESET. HIGH indicates active high; use RESET.

The next easiest connection point is an RC reset network. If you find a resister and capacitor in series between power and ground with the common point connected to the CPU or a monitor chip marked something like reset or clear, you probably found it. Connect the reset line at the RC juncture. If the resister is connect to ground and the capacitor is connected to power, use the RESET signal. If the resister is connected to power and the capacitor is connected to ground, use the /RESET signal. In some targets, the resistor or the capacitor may be missing but these instructions would still apply if you imagine the missing component is internal to the processor or monitor chip. If you are not sure, ask 'sparkie' (a technical term for the hardware engineer that designed the board).

Call our technical support if you are unsure about which output to use or where to connect to your target.

Apply Power

The emulator can be powered from the target or via an optional external power supply. If your target operates at 5Volts and can supply the extra emulator current (100ma typical/300ma max) to the memory socket, then you can use target power. If your target operates at less than 5V or is incapable of supply the extra emulator current, then you must use an external power supply. External power supplies are available from TechTools or your local dealer. The external supply must provide a regulated 5VDC at 300ma or greater.

The emulator status LED operates differently based on whether you are using target power or external power. If you are powering the emulator with an external power supply, refer to the section "Using External Power". Otherwise, refer to the section below, "Using Target Power".

Using Target Power

Turn on your target power supply. The emulator status LED should glow a STEADY GREEN. If not, refer to the troubleshooting tips below to correct the problem before continuing.

LED Glows RED or does not turn on:

Immediately turn off target power and correct the problem before reapplying power!

Your target does not have enough power to operate the emulator, power is not reaching the emulator or REVERSE voltage is reaching the emulator.

- Verify all target cable connections
- Verify the target cable is plugged in correctly (pin 1 orientation)
- Verify that any adapters you might be using are the correct pin-outs.
- Once you are SURE the emulator is not being subjected to reverse voltage, verify the target provides at least 4.5V to the memory socket WHILE THE EMULATOR IS PLUGGED IN. Connect an external power supply to the emulator if necessary.

LED Flashes GREEN:

You have enough power, but you failed to connect the reset line properly. When the emulator is first powered up, it asserts RESET to keep the target from running until it receives a download. The flashing GREEN LED indicates target activity.

- Verify the RESET line is connected
- Verify your selection of the RESET attachment point
- Verify your selection of RESET polarity.

Using External Power

Connect the external power supply but leave the target turned off. The emulator status LED should glow a STEADY RED, indicating the presence of emulator power but not target power. If not, disconnect the emulator power and refer to the troubleshooting tips below to correct the problem before continuing.

- Verify the external power supply is properly connected and plugged into a powered socket.
- Verify the external power supply is providing a regulated 5VDC at 300ma or more.
- Verify the external power supply is wired correctly (center is positive, ring is ground).
- Disconnect the target cable and re-apply emulator power. If the LED now GLOWS RED, the target wiring, adapters or cabling are incorrect.

Turn on target power. The emulator status LED should change from a steady RED GLOW to a steady GREEN GLOW. If not, refer to the troubleshooting tips below before continuing.

LED Turns OFF:

Something is VERY wrong with the target wiring, cables or adapters.

> Immediately turn off target power and then emulator power. Correct the problem before reapplying power!

LED STAYS RED:

The emulator did not sense at least 3V from the target power pins.

- Verify all target cable connections
- Verify the target cable is plugged in correctly (pin 1 orientation)
- Verify that any adapters you might be using are the correct pin-outs.
- Verify your ACM is properly configured.

LED Flashes GREEN:

The emulator detected the target's power, but the reset line is not properly connected. When the emulator is first powered up, it asserts RESET to keep the target from running until it has received a download. The flashing GREEN LED indicates target activity.

- Verify the RESET line is connected
- Verify your selection of the RESET attachment point
- Verify your selection of RESET polarity.

Connect the Download Cable

Select an unused **PRINTER** port on your PC. Remove any "dongles", security keys, printer switch boxes or extension cables from the port. Plug the supplied download cable directly into the selected printer port. Plug the other end of the cable into *FlexROM* III's "IN" port.

PLUGGING *FlexROM* III INTO A SERIAL PORT or an ETHERNET CABLE COULD DAMAGE THE EMULATOR!

Connect Daisy-chain Cables

Multiple units are daisy-chained by connecting the "OUT" connector from one unit to the "IN" connector of the next unit. Additional units are added in the same manner. Use the supplied 12" daisy-chain cable to interconnect the units.

When daisy-chained together, the emulators automatically determine their position in the chain, making it unnecessary to configure their addresses. However, it is important that you understand how their addresses are determined.

The LAST EMULATOR in the chain (the one without a cable plugged into its "OUT" connector) is always emulator #1. The next emulator in the chain is #2 and so forth. The emulator closest to the PC will always be the highest numbered emulator.

Run self-test

Running self-test immediately after installation gives you a solid baseline and helps you to find and correct any errors early on. Self-test will verify proper installation of all drivers, proper operation of the parallel port and all *FlexROM* III functions except the target interface buffers.

You will find 16 and 32bit console mode programs in directories below your install directory. The 16bit self test is included in case you plan to use the 16bit loader on a DOS machine. The 32bit self-test should be used if you are planning to use the Windows Quickloader or the 32bit console mode loader. We recommend using the 32bit self-test if you are running under Windows 95 or above. You can also invoke the 32 bit self-test program from within Quickloader.

If you are using multiple *FlexROM* IIIs, we recommend that you test them collectively to ensure reliable operation in the final configuration.

Type "RTTEST32" to run a full speed test over printer port 0x378. You can change the printer port by using a /P parameter. For example, if your emulator is connected to a printer port at 0x3bc, you would enter "RTTEST32 /P3BC". The most common printer port addresses (in order of popularity) are 378, 278 and 3bc. Type "RTTEST32 ?" to see all command line options.

If any of these tests fail, refer to the troubleshooting section for assistance. If the troubleshooting section does not resolve the problem, contact our technical support people for help.

Load your program

Your system should be functional and ready to go to work at this point. You can use the command line loader (RTLOAD16/RTLOAD32) or the Windows GUI loader (Quickloader) to load and verify files.

Start with a file that is known to work in this target. The binary image of a functioning EPROM is the best file to start with. Most EPROM programmers can read an EPROM and save its contents into a raw binary file. If binary is not an option, have the programmer store the file in Intel HEX or Motorola 'S' format.

This establishes a solid base-line. You can always come back to this file to verify everything is functioning (except perhaps your last code revision).

If you choose to use the command-line program, run it with a '?' parameter to see a simple help screen. You can also refer to the section below on RTLOAD16/RTLOAD32.

If you choose to use the Quickloader software, the setup should be pretty self-evident. Additionally, you can refer to the on-line help under the HELP menu.

If your system is not running at this point, check the troubleshooting section for suggestions. If that does not resolve the problem, PLEASE CONTACT OUR TECHNICAL SUPPORT at

support@tech-tools.com or (972) 272-9392.

We are confident that we can solve the problem in a timely manner!

Software

FlexROM III ships with several programs. The command line driven programs document their valid parameters if you invoke them with a '?' for a parameter. For example, to see the valid options for the loader program, type "RTLOAD32 ?". All programs except the Quickloader are batchable command-line style programs. The following are brief descriptions of the functions of each program.

QuickLoader

Quickloader (qloader.exe) is a 32bit Windows application. It operates under all 32bit versions of Windows including NT and 2000. Quickloader operates in 2 modes; one-click and fully interactive. In one-click mode, Quickloader performs automatic download and verify of one or more files via a single mouse click on a sys-tray icon. In fully interactive mode, Quickloader provides a full GUI interface to allow you to view and edit the files, control individual transfers and configure the project. Additionally, you can monitor the voltages, reset status and target activity from the hardware monitor window.

Extensive on-line help is included with Quickloader. The Quickloader help file can be accessed from the HELP menu or by double-clicking on the 'QL4HELP.HLP' file.

Command-line programs

We provide all of our command-line programs in both 16bit and 32bit versions. The 16bit versions run under DOS and all versions of Windows except NT and 2000. The 32bit versions uses our ring-0 drivers to access the low-level printer port registers. They work under ALL 32 bit versions of Windows (including NT and 2000.)

Each version shares the same base name but adds a '16' suffix for 16bit programs or a '32' suffix for 32bit versions. For example, RTLOAD16.EXE is the 16bit version and RTLOAD32.EXE is the 32bit version.

When you run the Quickloader installation, the 16 and 32 bit versions are installed into separate subdirectories. When you run the install on an NT or 2000 system, the 16bit versions are NOT installed because they cannot be used.

In the descriptions below, we will often refer to a program by its base name. For example, we would say: "RTRST allows you to set, clear or pulse the target reset." This applies to BOTH versions of RTRST (RTRST16.exe and RTRST32.exe). To actually run the program, you have to add the appropriate suffix (16 or 32). We will also give example usages for one version or the other but not both. They are identical.

If we give an example like "RTLOAD16……", you can freely substitute "RTLOAD32…." if you are using the 32bit tools.

RTLOAD16, RTLOAD32

RTLOAD loads binary and HEX files into one or more emulators. It can also perform a VERIFY-ONLY if you suspect that the emulator contents have changed.

Run either program with a '?' to see a help screen on the command line options. Notice that all parameters except ROMSIZE (/R) are optional and have logical defaults. You only need to specify the parameters you would like to change from their defaults.

EXAMPLE LOAD PARAMETERS:

RTLOAD32 file.bin /R32	Loads FILE.BIN into emulator #1 at port 0x378. Configure emulator to emulate a 32Kx8 (256Kbit) device
RTLOAD32 file.bin /p3bc /R128	Loads FILE.BIN into emulator #1 at port 0x3bc. Configure emulator to emulate a 128Kx8 (1 Mbit) device
RTLOAD32 file.bin /e2 /R1024	Loads FILE.BIN into 2 emulators, starting with #1. Configure emulators to emulate 1024x8 (8Mbit) devices
RTLOAD32 file.bin /e4 /f2 /R8	Loads FILE.BIN into 4 emulators, starting at #2. (2,3,4,5) Configure emulator to emulates 8Kx8 (64Kbit) devices
RTLOAD32 file.hex /d0 /h /R512	Loads FILE.HEX into emulator #1. Source file is in HEX. Configure emulator to emulate a 512Kx8 (4Mbit) device.

During the load and/or verify, the STATUS LEDs of all involved emulators will flash RED as they are accessed. At the completion of the transfer, target reset will be released. At this point, the STATUS LEDs will indicate the target status (RED = no target power, GREEN = target power good, FLASHING GREEN = target accessing the emulator).

The *FlexROM* III loader is very flexible, allowing any combination of emulators to be loaded in almost any manner desired. The loader can load to individual units or automatically perform file splitting between 2 or more units. When the loader splits a file between multiple emulators, it will always place the **FIRST BYTE** of the file into the **LOWEST NUMBERED EMULATOR** involved in the transfer. Successive bytes are loaded into successive emulators. The following examples demonstrate this:

EXAMPLE 1

If you have two 32Kbyte emulators daisy-chained together, and wish to split a file between them, you would specify: "RTLOAD32 filename /R64 /e2 /f1". The /f1 is optional since the default is to start with emulator #1. The /e2 parameter tells the loader that two emulators are involved in this transfer.

The loader will place the EVEN bytes (0,2,4..) of the file into emulator #1 (the LAST emulator in the chain). The ODD bytes (1,3,5..) will be placed into emulator #2. Note that you control which emulator holds the ODD bytes and which one holds the EVEN bytes by the order in which you daisy-chain them.

EXAMPLE 2

If you have four emulators daisy-chained together, an "/E4 " parameter would cause the loader to perform a 32 bit split between all four emulators. Bytes 0,4,8.. would be placed in the lowest numbered emulator (#1 by default). Bytes 1,5,9.. would land in the next higher numbered emulator. Bytes 2,6,10.. would be end up in the next emulator. Finally bytes 3,7,11… would be found in the highest numbered emulator involved in the transfer Again the byte ordering is determined by the emulator daisy-chain order.

RTTEST16, RTTEST32

These programs verify the reliability of the printer port and system wide communication with the emulators. They also perform a very complete self-test of all attached emulators. These programs test all emulator functions except the actual target interface buffers and target interface cable/connection.

RTTRIG16/RTTRIG32

RTTRIG16 and RTTRIG32 allow you to set the trigger value in any specific emulator.

RTSNAP16/RTSNAP32

RTSNAP continuously displays the value of the status pins, capture flag, trigger flag and snap-shot register from any specified emulator. It can display these on a single line or can be instructed to scroll the samples up the screen so that you can see the last couple of dozen samples. Pressing the space bar will toggle a freeze of the display. Pressing any other key will exit the program.

RTRST16/RTRST32

RTRST asserts, releases or pulses the target reset pins on all attached emulators simultaneously.

RTDUMP16/RTDUMP32

These programs allow you to view a HEX dump of 256 bytes of the emulator's memory. This can be useful for verifying that your download landed where you expected it to within the emulator's memory space.

RTCNTL16/RTCNTL32

These programs allow you to set/clear the 4 user control lines. You specify a single HEX digit that represents the state of the 4 lines.

Utilities

The CDROM contains several command line utilities. The loaders provide all HEX conversions and file manipulations needed so separate utilities are not really needed. They are provided to assist in manually manipulating data files if desired. The utilities operate under DOS and ALL versions of Windows.

Enhancements

Memory Retention

FlexROM III uses SRAMs to store the code image. If you wish to cycle the target power to see the target execute from power up, you will need to connect an external power supply to the emulator to maintain the SRAM contents during the power outage. The emulator will retain its contents as long as EITHER target power is applied OR external power is supplied to its power-in connector. The external power must be a regulated 5VDC at 300ma or more.

Faster Downloads - Turn Off Verify

After you have used the emulator for a while and gained confidence that it is loading reliably, you can add a /V0 parameter to tell the loader to skip the verify operation. This will reduce your load times by over 75%! *FlexROM* III is optimized for the fastest possible downloads. Therefore, the data stream FROM the HOST TO *FlexROM* III is much faster than the reverse direction.

If you ever wish to verify the CURRENT contents of *FlexROM* III's memory against a file, simple do a load with a "/V2" option (verify ONLY).

Increase 16bit Transfer Speed

The 16bit loader (RTLOAD16.EXE) is capable of over-running some printer ports. To allow proper operation with ALL printer ports, the 16bit loader includes a SPEED parameter. Once basic functionality is established, you may choose to speed-up your 16bit downloads. The download speed defaults to its slowest setting to insure compatibility with all printer ports. The self-test program does a reliability test at a specified speed setting. Re-run the FR3TEST program with a faster speed setting (lower /Dx number). If the reliability test shows that the system is reliable at this speed, you will be able to use this same parameter on the program.

Error Detection and Correction

If you add a "/C" parameter to the load command, the loader and *Flex*ROM II will generate and verify checksums on each packet. In addition, if an error is ever detected, the packet will be automatically re-sent. This adds about a 5% overhead to the transfers.

This is much more efficient than doing a complete verify of the SRAM contents against the load file and still provides a high degree of confidence that the download is valid.

Hardware Instrumentation

Voltage Monitor/Measurements

FlexROM III constantly monitors the target's voltage as well as its own. If the target's voltage drops below about 2.8V or if the emulator's internal voltage drops below 3.0 Volts, then the emulator will isolate itself from the target and assert a target reset.

In addition to monitoring the voltages, FlexROM III reports them to Quickloader so that the user can see the current target and emulator power status. Quickloader presents this information in its Hardware Monitor window.

The command-line routines also verify that the emulator has sufficient power to operate reliably before sending additional commands to the emulator. If the emulator reports insufficient power, the programs will alert you to the problem.

Trigger

FlexROM III incorporates an address match trigger circuit. This feature generates a pulse on the TRIGGER output (on the feature connector) each time the target accesses a specified memory location. You can specify the match value with Quickloader, RTTRIG16 or RTTRIG32. In addition to pulsing the trigger pin, an internal flag is set to hold the event until *FlexROM* III's status register is polled. The status register can be polled through Quickloader, RTSNAP16 or RTSNAP32.

The TRIGGER output can be used to trigger a Logic Analyzer, SCOPE or logic probe. It could even be used to trigger some external hardware.

Even if you have a Logic Analyzer, you will find this feature very useful. It is always ready and available without having to connect 20 or more address lines. Since it is always there (even when somebody "BORROWS" the logic analyzer), you will tend to make use of it.

When combined with a SCOPE, it provides a fully qualified Logic Analyzer type trigger on address matches to the SCOPE. When combined with a Logic Analyzer, it eliminates the need to connect all those signals to the address bus. This can increase the Logic Analyzer's trigger capabilities, free up all those extra probes to look at other things and shorten your setup times.

The trigger circuit can even help you diagnose hardware problems during the first bring-up of a new board. Simple set a trigger at the initial jump vector. Reset the target and see what happens. If you never receive a trigger, then the processor is not running at all (check power, decodes, oscillators, etc. …) or the chip select decoding is wrong. If you get continuous triggers, it is being reset over and over. This could be due to bad op-code fetches (bad data lines..), run-away watch-dogs, faulty reset circuit, etc. If you get the one and only one trigger expected, move the trigger to MAIN() and see if you get there, and so on.

Another trick involves using simple test programs or even data patterns (NOPS, JUMP-HEREs, JUMP to 0s) to provide predictable stimulus. For example, if you filled the memory with NOPs, most processors will walk through memory. Since all memory cells contain the same data, we should get the same results, regardless of stuck address lines or other address problems. If the processor does NOT increment through memory, it is probably fetching something other than NOPs . This points us to a bad data path. If it does walk through memory, the data path is reasonably healthy so we look at the address path. We can set a trigger at each power of 2 (address 0,1,2,4,8,16….). until we find the bad address line. Remember that a 16 bit address bus would require at most 16 trigger settings!

Snap-Shot

*Flex*ROM III incorporates a feature called Address Capture (or SNAP-SHOT). *Flex*ROM III hardware captures the value of the address lines during each valid target access. This data can be viewed with Quickloader, RTSNAP16 or RTSNAP32. The capture register can be interrogated without using any arbitration or worrying about interfering with the target.

While it does not maintain a history of every target cycle (like a logic analyzer or TRACE), it can provide considerable insight into your target's activities, once you understand exactly what it is giving to you. It is NOT a simple polled view of the current state of the address lines. It holds the address of the last VALID access to the emulator from the target. A quick glance at the current capture value can tell you if the program has "left the road", or has gotten stuck in a tight loop somewhere. If you have a map file handy, you can even tell where it is stuck!

Have you ever forgotten to clear an interrupt flag and found yourself being endlessly interrupted? This would be easy to spot with the capture register. Of course if you see addresses in the 0xC000 range and you know your code stops at 0x8000, you instantly know that all is not well.

Some customers are using this feature to help reverse-engineer the data table formats of automotive EPROMs, video game graphics, etc..

Control Lines

*Flex*ROM III provides 4 user control lines. These are TTL outputs on the feature connector that can be individually set high or low. You can control these with RTCNTL16, RTCNTL32 or from within Quickloader. The control lines are tri-stated whenever the target is powered down and will never drive higher than the target power supply.

Status Lines

*Flex*ROM III provides 4 user input pins. These pins are sampled at the beginning of each valid target access to the emulator. In addition, whenever a target access generates a trigger, the status pin levels at that instant are held until the status register is read. The status register can be read through RTSNAP16, RTSNAP32 or Quickloader.

Arbitration

*Flex*ROM III provides hardware based arbitration support. This hardware works with your target's hardware to affect arbitration, allowing you to read and write the emulator's memory while the target is accessing that same memory. This may be useful for updating data tables, strings or constants on-the-fly. The full-screen editor will even allow you to look into memory to see if the target left any messages there for you.

Regardless of arbitration settings, FlexROM III will ALWAYS grant access to the HOST whenever the target is powered down or being held in reset.

*Flex*ROM III supports several different arbitration methods.

Request/Grant

In Request/Grant arbitration, *Flex*ROM II will assert a /REQUEST before each access to the memory space. The *Flex*ROM II hardware arbiter will then wait until the target asserts a /GRANT signal before generating the memory cycle. At the completion of the access, *Flex*ROM II will release the /REQUEST signal and return control of the memory space back to the target.

*Flex*ROM III does not actually access the target's bus, but requesting access (and waiting for a grant) insures that the target will not attempt to interrupt our access to our memory.

These arbitration signals (/REQUEST and /GRANT) could be connected to your target's BUS request/grant, DMA request/grant or similar control lines. Of course many micro-controllers do not support any type of bus mastering or DMA operations.

Note that this is hardware based arbitration and very efficient. *Flex*ROM III will release /REQUEST within 300ns after /GRANT is asserted.

Ready

In Ready Arbitration, *Flex*ROM III inserts wait states into target cycles to affect arbitration. NOTE that this is radically different than simply slowing down your target with wait-states. *Flex*ROM III will insert wait-states ONLY DURING COLLISIONS. Your target runs at full speed except for during the brief moment in which the target attempts to access *Flex*ROM III's memory while it is being accessed by the *Flex*ROM II itself.

This is a hardware based arbitration method. The HOST posts a request to read or write to the emulator's memory space. *Flex*ROM III hardware watches the target's accesses and waits for the end of a target access before starting its own access. This minimizes the chances of a collision (minimizing the impact of arbitration). Once *Flex*ROM III sees a target access complete, it isolates its SRAM from the target and starts its own access. If the target starts an access before the *Flex*ROM III has completed its access, *Flex*ROM III asserts the /READY signal. This signal is held active until *Flex*ROM III has completed its access, re-enabled the target side buffers and the memory has had time to complete the current target access. At this point, /READY is released and the HOST is notified that the access completed.

The /READY line should be connected to your target's wait-state circuitry.

Cycle Interleaving

This technique depends on the emulator's internal CYCLE time plus its target ACCESS time being less than the target's CYCLE time. Since *Flex*ROM III avoids interrupting target accesses, and uses hardware based arbitration, it is possible to use CYCLE Interleaving if your target's memory CYCLE time is greater than 300ns. Many micro-controllers and some microprocessors meet this criterion.

To use this method, configure *Flex*ROM II for READY mode arbitration, but leave the /READY line disconnected.

Cycle Paralleling

This technique depends on you being able to find a signal in your target that guarantees the HOST exclusive access to the emulator. This is often much easier than it sounds. For example, most processors cannot fetch from their code space at the same time they are reading or writing their data space. In such systems, you may be able to use the DATA WRITE line or the SRAM chip select signal to indicate to the emulator that it is OK for the HOST to access the emulator's memory. The idea is that as long as the target is busy writing to its DATA space, it won't attempt to access its code space and therefore could not collide with the HOST access.

Other candidate signals might be IO READ, IO WRITE, or other peripheral chip selects. If your target offers none of these, you might be able to create a valid signal. You could instrument your firmware to periodically toggle an unused pin or access an unused but decoded memory or IO space to generate a stimulus that allows the emulator to gain access.

To use this mode, configure the emulator for REQUEST/FEGRANT or REQUEST/REGRANT and attach the selected signal to the emulator's GRANT line. Do not connect the REQUEST line to anything. The target is GRANTING us access every time it accesses the SRAM in this case.

The emulator will always wait for the START of a grant signal before initiating an access. The start is defined by a FALLING edge or a RISING edge of the signal, depending on which arbitration mode you selected (REGRANT or FEGRANT).

The time from the START of this GRANT signal to the next time the target may need data from the emulator must be at least 300ns for this to work.

"SLAM IT"

This is not true arbitration, but is worth mentioning. In some cases, arbitration is not really required, as long as the emulator does not tie up the bus too long. A good example of this is look-up data tables.

We have several customers using this technique to do on-the-fly modifications to Engine Controller ROMs, wave-form synthesizer ROMs, graphics ROMs, character generator ROMs, etc.

If your application uses the EPROM or FLASH device to hold data rather than code, and it can tolerate an occasional bad fetch (wrong data), it may be acceptable to simply allow *Flex*ROM III to collide with the target during the access. Note that *Flex*ROM III will not hold the target's bus or corrupt anything on the target's bus. It simply isolates itself from the target's address and data bus during its access, causing the target to fetch garbage during the collision. Each *Flex*ROM III access will isolate it from the target for approximately 300ns.

You have two options when using this method. You can configure *Flex*ROM III for CYCLE Interleaving (ready mode, but do not connect the ready line). This has the advantage of minimizing the probability of a

collision since *Flex*ROM III will wait until it sees a valid target cycle end before starting its cycle. Also, we will only collide if the target cycle time is less than 300ns. The down-side is that if the target stops accessing the emulator's memory space or accesses it infrequently, arbitration will time out. The other option is to configure *Flex*ROM III for REQUEST/FE-GRANT arbitration and then short the REQUEST line to the GRANT line; ensuring that *Flex*ROM III always receives an immediate grant. This method will always allow *Flex*ROM III access to the memory, but will blindly disrupt any current target access.

RESET

Of course this is not arbitration, but it is always worth asking "Do I really need to arbitrate at all?" If the changes that you will make to the EPROM/FLASH will require re-starting the target to take affect (like code changes), arbitration is not required (or desired). Simply let *Flex*ROM III assert reset during the modifications. *Flex*ROM III will then be guaranteed access to the emulation memory and the target will be automatically re-started after the changes.

Configurable Timing Parameters

FlexROM III gives you control over some of its timing parameters. If you are not a hardware type then these options may not make much sense. Don't worry about it. The default settings will almost always work. These options are provided to enhance the emulator's robustness in particularly noisy targets or targets that violate some of our specs. Even in those rare cases, you should be able to make reasonably well-informed selections based on the descriptions below and minimal help from your hardware engineer.

Throughout this section, we will be talking about control signals and timing requirements. To keep things readable, we need to agree on some acronyms. The follow acronyms are commonly used when talking about memory devices:

Acronym	Description	Notes
CS	Chip Select	Enables the device when driven low
OE	Output Enable	Enables the devices output buffers if CS is active
WT	Write Enable	Enables a WRITE cycle if the CS is active
BHE	Byte High Enable	Selects D8-15 for any valid cycles
BLE	Byte Low Enable	Selects D0-7 for any valid cycles
Tacc	Address Access Time	Time from address stable to valid data out
Tcs	Chip Select Access Time	Time from CS enabled to valid data out
Toe	Output Enable Access Time	Time from OE enabled to valid data out
Twp	Write Pulse Width	Time WT line is held active during a write cycle
Tcr	Cycle Recovery Time	Time between the end of a cycle and the beginning of the next. A cycle ends when CS goes inactive or both OE and WT go inactive. A cycle begins when CS is active AND either OE or WT goes active.

NOTE: It is somewhat of a misnomer to talk about a device's ACCESS TIME. There really is not a single access time spec. The device will provide data only after meeting ALL access time specs. Generally, when we say a device has a '70ns access time' we are referring the to Tacc spec which is usually equal to the Tcs spec.

Fast CS Access

Normally, FlexROM III drives its SRAM CS (chip select) lines with a buffered version of your target's CS line and provides stable data 45ns after the last transition on address or after CS goes active – whichever is later.

Enabling FAST CS ACCESS causes FlexROM III to force its SRAM CS lines always active. FlexROM III's output buffers are still gated with your target CS so the emulator continues to honor all target control signals. This reduces the emulator's Tcs access time from 45ns to 12ns.

Many targets will decode their CS signal off of address lines. This makes the Tcs the determining factor in setting its ACCESS time requirements. The Tacc might be 45ns but due to decoding delays, the Tcs could be 30 – 40ns. In this situation, the FAST CS option would allow our 45ns emulator to function properly in a target that normally requires a 30-40ns emulator.

You normally would not turn this option on unless your target has a Tcs requirement of < 45ns and its addresses are stable before CS is activated.

Turning on this option will result in higher operating current requirements and could increase system noise in targets that have extended periods of address line float.

Fast OE Access

Normally, FlexROM III drives its SRAM OE (output enable) lines with a buffered version of your target's OE line and provides stable data 45ns after the last transition on address or after CS goes active, or 30ns after OE goes active – whichever is later.

Enabling FAST OE ACCESS causes FlexROM III to force its SRAM OE lines always active. FlexROM III's output buffers are still gated with your target OE so the emulator continues to honor all target control signals. This reduces the emulator's OE access time from 30ns to 12ns.

Turning on this option will result in slightly higher operating current requirements and could increase system noise in some targets during target writes to the emulator.

You normally would not turn on this option unless your target has a Toe timing specification of < 30ns. However, it generally does not hurt to do so.

It is acceptable to use FAST CS and FAST OE at the same time.

Filtering

FlexROM III provides the ability to filter noise out of the target's control signals before using them to control the SRAMs or to detect the end of valid cycles (for arbitration, snap-shot & trigger). Any noise on a control signal while it is active can corrupt a target write or cut into the read margins. Filtering out this noise greatly improves emulator performance in noisy environments.

The up side to filtering is a more robust emulator. The trade-off (there's ALWAYS a trade-off) is that the greater the filtering, the longer it takes to determine that the cycle really ended. For example, if we are filtering out pulses shorter than 20ns, the true end of cycle must obviously last somewhat longer than 20ns or we will filter it out as noise. Even if end-of-cycle gets filtered out, the target can usually continue to function properly. The end-of-cycle detection is used to update our SNAP-SHOT register, the trigger circuit and to terminate target write cycles.

The default settings enable the control line filters and sets them to minimum filtering. This is a reasonable default that allows full speed access on most targets. In fact, most targets could increase the filtering to MAX and still meet timing.

The only time you may have to turn off filtering is if:
- The target does writes to the emulator OR you are doing READY mode arbitrated accesses.
- AND the target has very short cycle recovery times
- AND the target has fast access time (45ns) requirements

If you are not doing arbitrated accesses to the emulator and the target does not write to the emulator, you can usually add as much filtering as you want. In this situation, too much filtering on a target with short recovery times would result in the SNAP-SHOT & trigger circuits failing to update, but the target would function correctly as long the LATCHING options (see next section) are left at their defaults.

Latching

FlexROM III always latches the address lines during a target write. This enables it to enforce proper address hold times to the SRAM despite varying path delays or sloppy target timing. By default, they are only held stable while the target CS and WT signals are BOTH active.

In some cases, it might be desirable to latch the addresses on reads as well as writes or to latch them as soon as CS goes active. To avoid latching noise, it is sometimes desirable to delay latching until some time after the control signal. Most system noise occurs immediately after control signal transitions. Delaying the actual freezing allows the signals to settle.

FlexROM III gives you the ability to select the source that triggers the latch and how long AFTER the trigger it will wait before actually freezing the signals.

The latching options and possible reasons you would want to use them are listed below.

Latch on CS active

This triggers the latch when CS goes active, irrespective of WT or OE. You might choose this setting if your system has a very narrow WT active signal. Latching the addresses much earlier in the cycle may make it possible to operate in a target that violates our MNIMUM Twp spec of 25ns.

It may also allow us to operate properly in a target that generates noise on the address lines while the write cycle is in process. Once a write has started, it is illegal to change the address lines. Any noise on the address lines during the write will corrupt one or more memory locations. Some targets will have stable addresses just before the WT goes active, but generate noise right after or during the WT active. Latching earlier in the cycle makes us immune to these violations.

DO NOT USE THIS SETTING IF YOUR TARGET PERMANENTLY GROUNDS (ENABLES) THE CS SIGNAL.

You also would NOT want to use this setting if your target does burst reads to the emulator. This occurs when the target asserts CS and OE and then does a sequence of ADDRESS – controlled reads without releasing either CS or OE between each access. In this case, you could use the 'LATCH on CS and NOT OE' setting.

Latch on CS and WT active (default)

This triggers the latch when we see BOTH CS active and WT active. This is the default setting. It is appropriate in most cases.

Latch on CS and either WT or OE

This latches the addresses when CS is active AND EITHER WT or OE are active. This might be appropriate if noise on the address lines is causing access time violations. 'LATCH on CS ACTIVE' is another option if your target does not have its CS line grounded and it does not do burst-reads.

Latch on CS and NOT OE

This setting triggers the latch when CS goes active, but then RELEASES the address lines if it sees OE go active. This allows us to do early latching if desired but still support targets that do burst-reads.

Latch Delay Enable

This option specifies IF (and how much) we delay actually freezing the address lines after a signal triggers latching. Most address line noise occurs immediately after control signals activate or when the emulator begins driving data back to the target. Being able to delay latching the address until after the noise settles down makes it possible for us to be immune to it.

The default setting is Latch Delay is ENABLED and set to its minimum delay setting.

Latch BHE/BLE Enable

This option specifies that we should freeze the Byte Enable signals as well as the addresses. It is on by default. Most targets do not drive BLE or BHE directly. The ACM-8 derives them from A0. The ACM-16 forces them both active. The only time you might need to turn this option OFF is if you are using the ACM-C module and your target needs to change the Byte Enable lines without release CS or the WT or OE signals. This is conceivable but highly unlikely.

Troubleshooting

The distribution software contains test programs called RTTEST32.EXE and RTTEST16. They verify the port address, its reliable at the selected transfer speed and the emulator's functionality as well as proper installation of the ring-0 drivers. The first step in troubleshooting is to run this program and make special note of any problems it reports.

Regardless of the symptom, verify the following:

1. There are no "dongles", security keys, switch boxes or other devices between the *FlexROM* III download cable and the printer port.
2. Try each available printer port configuration. Some ports work best in STANDARD, AT, NIBBLE or COMPATIBLE mode. Others may provide more reliable high-speed operation when set to ECP, EPP, PS2, Bi-directional or HIGH-SPEED mode.
3. The target is powered up or an external power supply is connected to the emulators
4. All download, daisy-chain and target interface cables are tight and fully inserted.
5. The selected printer port is not being serviced by a print spooler or some type of TSR.
6. If the PC is a laptop or a "GREEN" PC, verify that the printer port is not disabled, powered down or configured for any type of power savings mode.

In all cases, try "booting" clean from a DOS diskette to see if the symptoms disappear. If the 16bit loader and self-test programs work under this condition, chances are good that some other program is interfering with our access to the printer port.

DLL not found or DRIVER not found

Drivers not properly installed

The 32bit software versions require our ring-0 drivers. These are automatically installed into the correct directories and added to the windows registry when you use our SETUP.EXE program. You must run SETUP.EXE install the files onto a computer. Just COPYING the files will not properly register them with Windows.

Under NT and 2000, you must have ADMINISTRATIVE/Supervisor PRIVILEGES to properly install the drivers and update the registry.

Failed to re-boot after installation

The drivers will not become active until after you re-boot. Sorry, as much as we hate the 'reboot windows for this to take affect' syndrome, its just the way it works.

Printer Port Not Found Message

Specified the wrong port address

The software could not find a printer port at the specified address. Try another address. The command line programs use a /Pxxx parameter to select the port address. Quickloader uses pull-down selections. The most common addresses are 378, 278 and 3BC. Also verify that the attached emulators are powered up. Un-powered emulators could prevent the software from seeing the port itself.

Using the wrong software

Under NT and Windows 2000 Professional, you MUST use the 32bit software. These include Quickloader and all command line programs that end in '32'. These programs all use our ring-0 drivers to access the printer port hardware. Win95, Win98 and ME users may user either the 16bit or 32bit versions of these programs. Win3.1 and DOS users must use the 16bit versions. We do not provide a 16bit GUI application for DOS or Win3.1.

No Emulator(s) Found

Specified an existing but incorrect printer port.

This error message indicates that the specified port was found, but no emulators were found at that port. The emulators might be plugged into a different port. Change the port selection to specify the correct port or move the download cable to the specified port. The most common printer port addresses are 378, 278 and 3BC. If you are using the command-line loaders, use the /Pxxx parameter to specify the port.

Using wrong download cable or backshell

The FlexROM III download cable is compatible with the EconoROM III and the older FLexROM II product but is NOT compatible with EconoROM II or earlier FlexROMs. Use the cable provided with FlexROM III.

Transfer speed set too high for this port (16bit loaders only).

The 16bit software is capable of over-running some printer ports. You may need to reduce the speed to achieve reliable operation in these situations. The RTTEST16 program verifies transfer reliability. Test the desired transfer speed reliability with "RTTEST16 /dx" (where x is 0,1,2,3 or 4). /d0 is the fastest setting.

If you are daisy-chaining several emulators, their combined load can slow down the printer port rise times, resulting in a need to slow down the transfer speed to avoid over-running the port signals. Run the RTTEST16 program with the emulators daisy-chained together to properly reproduce the eventual working environment.

One or more Emulators not powered up.

FlexROM III draws its operating power from the target's EPROM socket or an optional external power supply. The target must be powered up or the external power supply connected during all load or verify operations. If multiple emulators are daisy-chained, they must ALL be powered up to talk to ANY of them.

Target power is noisy or Voltage too low.

FlexROM III draws more current than the EPROM it is replacing. If the target is a very low power system, or is already pushing its power supply limits, it may not have enough extra current to power the emulator, causing power fluctuations or noise.

If the target voltage (at the memory socket) is less than 4.5V, then an external power supply MUST be connected to the emulator.

NOTE: The power may look good while the target is in reset but become noisy or droop once the target starts running.

If the emulator is being powered from the target, ensure that the memory socket the emulator is plugged into has sufficient by-pass capacitance (a good idea for EPROMs as well!).

Fails Verify

Target Power noisy or Voltage too low.

Noise problems are very dynamic. It is possible that we could identify the emulators but be unable load and verify properly. *FlexROM* III draws more current than the EPROM it is replacing. If the target is a very low power system, or is already pushing its power supply limits, it may not have enough current to spare to power the emulator, causing power fluctuations or noise. An external power supply would solve this.

Transfer speed set too high for this port (16bit loaders only).

The 16bit software is capable of over-running some printer ports. You may need to reduce the speed to achieve reliable operation in these situations. The RTTEST16 program verifies transfer reliability. Test the desired transfer speed reliability with "RTTEST16 /dx" (where x is 0,1,2,3 or 4). /d0 is the fastest setting.

If you are daisy-chaining several emulators, their combined load can slow down the printer port rise times, resulting in a need to slow down the transfer speed to avoid over-running the port signals. Run the FR3TEST program with the emulators daisy-chained together to properly reproduce the eventual working environment.

A print spooler or TSR is interfering with our port accesses

Try "booting" clean from a DOS diskette to see if things stabilize. If this corrects the problem, suspect that some other program (like a print spooler) is interfering with our access to the port.

Printer port set to an incompatible mode

If your printer port is configurable through BIOS settings, try each configuration. Some ports will perform better in STANDARD, AT, NIBBLE or COMPATIBLE mode. Others will allow higher transfer speeds when configured for ECP, EPP, PS2, HIGH-SPEED or similar settings. This is rarely a problem.

Verifies, but target does not run

Target not being reset or not being released from reset

If the wrong reset output is being used or it is connected to the wrong spot on the target, it may not be resetting the target during the download. We would get a good download, but the target was fetching garbage during the transfer. Some processors will HALT, others will simply execute the random garbage they received. In either event, the processor will not "know" to start over and execute the new code.

Try pushing your target's reset button if it has one. If it starts to run, the reset line is connected to the wrong place on the target.

Try removing the reset line. If the target starts running, you probably selected the wrong reset polarity.

If the emulator is running from an external power supply, try removing the reset line and cycling TARGET power. If the target now comes up and runs, the reset signal connection was wrong. While you could do this after each load, we highly recommend correcting the reset problem to make resets automatic and to prevent the target from running garbage during the download. The reset line also ensures proper power-up sequencing by preventing the target from fetching code before the emulator has had a chance to enable its target-side buffers.

Emulator device size set wrong.

The device SIZE configuration affects how many target address lines are enabled. The file will download and verify correctly as long as the file is less than or equal the SIZE setting and the maximum emulator size. If the size is set larger than the device you are emulating, some extra address lines will be enabled. The target may be driving these lines high or NOT AT ALL. This will either warp the addressing or cause random address changes. Ensure you are configuring the SIZE to match the device you are emulating.

Wrong PLCC or other adapters used.

Any adapter used between the target cable and the target socket itself has the potential of introducing errors. In particular, PLCC footprints vary by EPROM size, model and manufacturer. In general, small EPROMs (<1Mbit) generally have one pin out and larger EPROMs have a different pin out. Most FLASH chips are pinned out like large EPROMs. However, SOME small FLASH are wired like SMALL EPROMs. In general, large devices (>=1Mbit) use a 32 pin DIP to 32 pin PLCC adapter. Smaller devices usually need a 28 pin DIP to 32 pin PLCC adapter but MIGHT need the 32 pin version. Give us a call if you are not sure.

Wrong file format specified

The files you wish to download might be in a HEX format or they might be a raw, binary image. The loaders need to be 'told' whether they are in HEX or BINARY format. If you are not sure about the file's format, "TYPE" it to the screen. If you see random printable, non-printable and graphics characters, it is BINARY. If you see well formatted HEX characters, it is a HEX file. Intel HEX files start each line with a ":". Motorola 'S' files start each line with an "S". Tektronix HEX files start with a "%" or a "/". All of our loaders automatically recognizes and converts 8 variations of these HEX formats. However, even HEX files are a FORM of binary so you must tell the loaders whether to blindly load them as binary or assume they are in HEX format.

Remember, your Resume' will load and verify, but most likely will not execute properly!

Incorrect HEX conversion

If your source file is in HEX, you may need to specify an offset to get a proper conversion. If an offset is required, it is usually the starting address of the EPROM. You can use RTDUMP16, RTDUMP32 or Quickloader to see where the conversion placed the binary data within the device space.

Failed to split the file between emulators

If you want the command line loader to spilt the file between multiple emulators, you need to specify a /Ex parameter to tell the loader how many emulators to split the file between. In Quickloader, you must specify the 'number of emulators'.

File loaded into the wrong emulator(s)

If you have 2 or more emulators daisy-chained, you may have loaded the file into the wrong unit(s). Watch the STATUS LEDs to see which units where actually selected for the transfer. The loaders default to emulator #1. To select a different emulator, specify a /Fx parameter. This tells the load the FIRST emulator involved in the transfer.

Incorrect Byte Order

If you are loading into multiple emulators with automatic splitting, remember that the loader will load the **FIRST byte in the file into the lowest numbered emulator** involved in the transfer. If the emulators are plugged into the wrong target sockets, you can move them to the correct sockets or you can change the order in which the emulators are daisy-chained.

Target Power noisy or Voltage too low.

Noise problems are very dynamic. It is possible that we could load and verify OK, but as soon as we release reset and let the target start running, system noise increases and trashes the system. Low system voltage could behave in a similar manner. It is possible that we could load and verify with a target voltage that is well under spec, but the target will not execute properly. *FlexROM* III draws more current than the EPROM it is replacing. The target may be able to supply enough current during downloads, but may not be able to hold the voltage up when it starts running, causing power fluctuations or noise.

Emulator too slow

FlexROM IIIs are rated at 45ns. The emulator must be at least as fast as the access times required by the target (smaller numbers are faster). If the target's access requirements are not documented, you can get a general idea of its requirements by looking at the EPROM's rating. The EPROM's speed is not an absolute indicator because many EPROMs run faster than they are marked; particularly at full voltage and room temperature. We have seen many systems that appear to run fine, even though their access times routinely violate the EPROM's rating. Fortunately, *FlexROM* III's speed rating (45ns) is faster than most ERPOMs used in embedded systems.

If your target's access time specs require a faster emulator, look at FAST OE and FAST CS options described in the 'Configurable Timing Parameters" section. One or both of these options may allow you to relax the target's access timing requirements and bring them within the emulator's specs.

Noisy Control Signals

Try enabling control line filtering or increasing the amount of filtering. See the "Configurable Timing Parameters" section for details.

Noisy Address Lines

Look at the Latching Options described in the "Configurable Timing Parameters" section.

Bad Code

Remember, the one variable that is always changing in this set up is the code itself! If you keep a baseline code image on hand that is known to work, you could always load that image into the emulator as a sort of sanity check.

Bad Target Hardware

Try running the target from an EPROM that contains known good code to see if the hardware functions without the emulator. This is always a good sanity check if things just stopped working and the 'Bad Code' suggestions above did not help.

Arbitration Time-outs

Arbitration time-outs are reported when *Flex*ROM III was unable to complete a memory access within about 65 us. This can only happen if arbitration has been enabled. If you did NOT intend to do arbitrated accesses, remove the "/A" parameter from the loader command line or turn on the "RESET during DOWNLOADS" option in Quickloader. If you truly meant to do arbitrated accesses to the memory space while the target is using that same space, verify the following:

If you are using READY arbitration, the emulator MUST see target activity to properly arbitrate with it. If the target is not accessing the memory space emulated by this emulator, it will not see activity and therefore will not be able to arbitrate. The emulator must see ACTIVITY (both start of accesses and end of accesses). Also note the /READY line uses an open collector driver and provides a very weak pull-up resistor. If the target does not provide a pull-up at the connection point, you may need to add a 1K - 10K resistor between the connection point and your target's VCC to improve rise times.

If you are using REQUEST/GRANT arbitration, verify that the target arbiter is working, both /RESET and /GRANT are properly connected and that the target arbiter expects an active LOW /REQUEST and generates an active LOW /GRANT (if using FEGRANT) or an active HIGH GRANT(if using REGRANT). Also note that /REQUEST is an open collector driver to allow multiple emulators arbitrate in parallel. If your target does not provide a pull-up resistor at the attachment point, you may need to add one (around 10K should do).

If you are using CYCLE INTERLEAVING arbitration, verify that you set the arbitration option to READY mode and left the ready line disconnected. Also verify that the target cycle time is at least 300ns. This is measured from the END of one cycle (CS, OE or WRITE going INACTIVE) to data stable requirement in the next cycle.

If you are using Cycle Paralleling, verify that the selected 'grant' signal truly indicates that the target will not expect data from the emulator for at least 300ns. You could verify this with a 2 channel scope. Trigger one channel on the selected 'grant' signal. Connect the 2nd channel to the emulator's chip select line. If the two NEVER go active at the same time, or if there is always at least 300ns between the start of the 'grant' signal and the start of the emulator chip select, then it should work.

Works but VERY slowly under NT/2000

You are using the wrong software. The 16bit software will sometimes work on these systems, but through a virtualization mechanism that is VERY slow. Our 32bit software uses ring-0 drivers to access the hardware at about 2.5Mbit/sec. Use Quickloader or RTLOAD32.EXE. We no longer install the 16bit software when you run SETUP on an NT machine so you normally cannot make this mistake.

Checksum Errors

Checksum errors indicate a communications problem. This may be due to bad download cable connections, a slow rise-time printer port, noise on the download cable, noise on the target power, etc. You might try a slower download speed (increase the /d parameter). Also verify that the download cable is not lying across a power transformer, your monitor, an arc-welder, etc. Other sources of problems could be switch-boxes with dirty contacts or long download cable extensions.

If these tips did not solve the problem you are experiencing, please contact our technical support people at

support@tech-tools.com
http://www.tech-tools.com
Voice: (972) 272-9392
FAX: (972) 494-5815

FAQs

Can I use FlexROM III on a 3V target?

YES, but you will need an external power supply. Connect 5VDC at 300ma or greater to the emulator's external power connector. External power supplies are available form TechTools.

What is the lowest target voltage supported?

FlexROM III was designed to support 3.0V to 5.0V targets. However, it CAN support lower voltage targets within limits. Three things set the lower limit; Logic INPUT thresholds, Vout HIGH from the emulator, and the target isolation threshold.

- Logic INPUT Thresholds. *FlexROM* III will accept a Vin HIGH minimum of 1.5Volts. Your target MUST produce at least 1.5V for logic HIGHs. Most logic down to about 2V can meet this requirement.
- Vout HIGH levels: Your target must tolerate logic HIGHs from the emulator data lines of 3.2Volts. This depends entirely on your target design.
- Target Isolation Threshold: To allow safe target power cycling,, *FlexROM* III constantly monitors the target's voltage. When it detects that the target voltage has dropped below a fixed threshold, it assumes the target is powering down so it turns off its output buffers to avoid leaking current into the target. It also asserts reset and turns the STATUS LED RED. If the emulator did NOT provide this safety feature, it could start reverse biasing devices on the target, potentially causing damage. Even if the target were not damaged, the current leakage could prevent the target from properly powering down. This could prevent the reset circuitry from operating. This cut-off threshold is approximate 2.7 to 2.8V.

Bottom line: You CAN emulate lower voltage targets as long as you can tolerate 3.2V on your data bus, your target provides logic highs of at least 1.5V and our low voltage cut-off circuit does not kick-in.

How accurate are FlexROM III's voltage readings?

The readings are typically accurate to +- 250mv.

Can FlexROM III emulate FLASH devices?

The emulator supports FLASH pin-outs and accepts target writes. However, it does NOT emulate any of the special FLASH algorithms. For example, it does not support the special sequences that lock/unlock pages of memory or any kind of busy indicators. It operates at full speed and looks like SRAM to the target.

Do I _REALLY_ need to connect the reset line?

This seems to be the biggest hurdle for first-time users. While the reset line may not be essential in some specific cases, it is ALWAYS a good idea and OFTEN a requirement for consistent, reliable operation. There is a lot to be said for consistency during debug! It is well worth the minimal time necessary to figure out how to connect the reset line. It serves several functions:

- It keeps the target from running garbage immediately after the emulator is powered-up. The reset will be automatically released after the first code download.
- It keeps the target from running garbage DURING a download or verify.
- It ensures the target gets a clean reset and starts running the new code after a download.
- It ensures proper power-up synchronization when the emulator is being powered from external power and the target power is cycled.
- It makes the entire download/verify/run operation fully automatic and consistent.

Can FlexROM III emulate DIP/PLCC/TSOP… packages?

Yes. TechTools stocks low cost PLCC adapters and all emulators ship with DIP cables. We find that DIP and PLCC packages cover about 95% of our customers' needs. For other packages, we refer you to the standard adapter companies. They can provide adapters for all common EPROM, SRAM and FLASH devices.

Can I use a longer target cable?

Maybe. We use 'soft" drivers and 6" cables to minimize our impact on your target. This combination has worked very successfully for us for over 10 years. Lengthening the cable increases the probability of both cross-talk and line reflections.(depending on your target drivers' rise-times). Longer cables also add capacitance to the bus. If your target access times are considerably longer than ours (45ns), and/or your drivers are fairly soft, you may be able to tolerate a considerable cable length increase. On the other hand, if you need every last nanosecond of our access time and your bus has very fast rise-time drivers, you should probably stick with our proven solution.

Can I use a longer Download cable?

Yes. You can use up to 25' of download cable between the PC and the first emulator. However, we recommend that you use the 12" daisy-chain cable provide to link multiple emulators together. This provides a more 'lumped load' at the end of the line.

If you replace the flat cable, make sure the cable is wired 1-to-1. Most data cable is wired 1-to-1. Most voice cable is wired reversed (1-to-8). If you extend the download cable by inserting a DB25M-DB25F cable, best results are obtained with the typical round PC printer port cable extensions. These typically common all of the grounds on the port, creating better signal integrity. The worst choice (in this situation) is flat ribbon cable because only one ground ends up being used.

Appendix I Specifications

FlexROM III Timing Specifications:

The following specifications assume:
- All Filtering has been disabled.
- The Latching options have been left at the default setting of (CS and WT active).
- Latch delay has been disabled.

Enabling Filtering and/or Latch delaying adds 10-40ns to the Tcr-2. Tcr-3 and Tcr-4 specs.
Enabling Latch on (CS) or Latch on (CS and NOT OE) with MINIMUM delay selected reduces Twp to 20ns. Higher delay settings could result in longer Trdc & Twtc requirements.

Read Cycles:

Symbol	Name	Description	Value
Trdc	Read Cycle	Time for a complete read cycle	45ns (max)
Tacc	Address Access	Time from address stable to data valid	45ns (max)
Tcs	CS Access Fast-CS enabled Fast-CS disabled	Time from CS active to data valid	 12 (max) 45ns (max)
Toe	OE Access Fast-OE enabled Fast-OE disabled	Time from OE active to data valid	 12ns (max) 30ns (max)
Todo	OE to data disable	Time from rising edge OE to data float	0 – 6ns
Tcod	CS to data disable	Time from rising edge CS to data float	0 – 12ns
Tcr-1	Cycle Recovery time, valid data	Minimum recovery after a read	0ns (min)

NOTE: Data will be valid after ALL access time specs have been meet; Tacc after address stable, Tcs after CS active and Toe after OE active.

Write Cycles (WT controlled/CS controlled):

Twtc	Write Cycle	Time for a complete write cycle	45ns (max)
Tas	Address Setup time	Address stable to falling edge WT/CS	0ns (min)
Tah	Address Hold time	Address held after rising edge WT/CS	0ns (min)
Tds	Data Setup time	Data stable to rising edge WT/CS	10ns (min)
Tdh	Data Hold time	Data held after rising edge WT/CS	0ns (min)
Twp	WT pulse width	Minimum WT active pulse width	25ns (min)
Todw	WT to data disable	Time from falling edge WT to data float (if CS & OE were active)	0 – 12ns
Tcr-2	Cycle recovery time, writes	Minimum recovery time after a write	20ns (min)

Snap-shot and Trigger functions

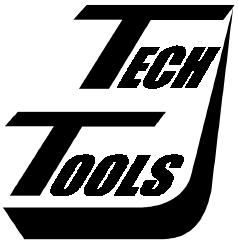
Tcr-3	Cycle Recovery time, reads	Minimum recovery time after a read	10ns (min)
Tcr-4	Cycle recovery time, writes	Minimum recovery time after a write	25ns (min)
Top	OE pulse width	Minimum OE active pulse	25ns (min)

NOTE: Violation of the Snap-shot/Trigger specs will not cause target side failures unless their specs are being violated as well.

Power Requirements

Parameter	Minimum	Typical	Max
Volts	4.75 VDC	----	5.25 VDC
Current	---	100ma	300ma

TechTools reserves the right to change these specifications at any time without notice. Also, since the ACM modules and memory options contain active circuitry that is involved in establishing these parameters, future ACM modules or model sizes may affect these numbers.



TechTools
PO Box 462101
Garland, TX 75046
(972) 272-9392 FAX: (972) 494-5814
Sales@tech-tools.com Support@tech-tools.com
www.tech-tools.com

