Applied
Microsystems
Corporation

NetROM™ 500 Series

Hardware Interface Reference

May 1997

Trademarks

Contents

Preface

Chapter 1 Optimizing TTL Bus Design

Chapter 2 Optimizing PC Board Design

Chapter 3
NetROM's Cable Termination

Chapter 4
Using Mass Terminated Connectors

Chapter 5
Using the Command Status Connector

Chapter 6
Writing to NetROM Memory

Chapter 7
Circuit Design Recommendations

Preface

Introduction

NetROM is a powerful communications and ROM-emulation tool for use in embedded-systems design. Although NetROM's most important function is communication between the development host and the target system, NetROM allows design engineers to substitute NetROM for the target's ROM during software design. This simplifies and shortens a project's software development. For example, you can use NetROM's RAM instead of the time-consuming process of burning and reburning PROMs during design development and testing. When you are satisfied that the design is functional, disconnect NetROM and create a permanent ROM to replace the temporary NetROM circuit.

For NetROM to work most effectively as a design tool, engineers should consider several design elements to reduce or eliminate potential circuit problems.

This document describes some aspects of the design process in which designers can avoid design pitfalls. Most circuits in the field, perhaps 99%, function satisfactorily without any of the problems discussed here. However, we have found these design recommendations will optimize the performance of many circuits, not just the problem ones.

Conventions

This manual uses the following conventions:

- Target device refers to the printed circuit board to which the NetROM is temporarily attached.
- ROM is used to indicate ROMs, PROMs, and Flash ROMs, except in instances when differentiating between them is important.
- Book titles, emphasized words, command names, and keywords are in *italics*.
- Command parameters (both alpha and numeric) are in **boldface.**
- Computer programs are in `constant-spaced font.`
- Environment variable names are in "quotation marks."
- Items that are completely optional are enclosed in [square braces].
- Items that are mutually exclusive are separated by a vertical bar |.
- Mutually exclusive items, one of which is mandatory, are enclosed in {braces}.

Warnings, cautions, notes

Warning

Warning messages appear before procedures and alert you to the danger of personal injury which may result unless certain precautions are observed.

Caution

Caution messages appear before procedures and indicate that damage may be done to the emulator or to your target system unless certain steps are observed.

Note

Notes indicate important information for the proper operation and installation of your emulator.

Support services

Applied Microsystems provides a full range of support services. NetROM is covered by a 90-day warranty. Additional support agreements are available to provide additional services.

If you have trouble installing or using the product, consult your manuals to verify that you are following the correct procedures.

If the problem persists, call Customer Support. Customers outside the United States should contact their sales representative or local Applied Microsystems office. When you contact Customer Support, have ASI number available.

Telephone

800-ASK-4AMC (800-275-4262)

Internet address

If you have access to the Internet, you can contact Applied Microsystems Customer Support using the following address:

support@amc.com

You can also browse the Applied Microsystems World Wide Web page using the following URL:

http://www.amc.com

See the Applied web page for NetROM Application Notes.

FAX

(425) 883-3049

Chapter 1

Optimizing TTL Bus Design

This chapter discusses potential problems occurring in TTL bus design and recommends ways to optimize bus design.

Keep the ROM TTL bus on the PC board

Ideally, the bus should stay on the target's PC board and not go through a connector. If the target device contains more than one board, good design practice suggests that the bus stay on one board not go from board to board. Connectors and their connecting cables tend to add stray capacitance and inductance, which effects the performance of the bus.

Keep bus length short

The longer the bus the more inductance and capacitance. Independent empirical studies show that about 10 inches is the longest a bus should be without considering the analog effects on bus performance.

Minimize tees (stubs) and keep tee length short

A tee is formed on a PC board when the designer places a junction in the etch to carry a signal to two locations, rather than to a single destination. Typically, the etch or wires should go from point to point (serial fashion) without any intersections. Ideally, there are no tees on the bus. In practice, however, small stubs of one-quarter inch or less are acceptable. Longer stubs of one inch or more often create unnecessary problems.

Basically, stubs are undesirable because they cause reflections going back into the driver. A receiver can reflect energy back to the driver through both paths. With no stub, there is only one source of reflections back to the driver.

Reflections are undesirable because they effect the wavefront. A perfect waveform goes from a logical 1 to a logical 0 state, or vice versa, in the 3-to-6 nanosecond range. A tee adversely affects the sharp edges of the wavefront because of the reflections.

Figure 1-1 shows the effects a stub can have on a buffer. The signals going into (A) the buffer and out from the buffer show sharp delineated wavefronts when there is no stub, with the output waveform looking very similar to the input. When you add a tee in the circuit, not only does the wavefront input to the buffer (B) become very ragged, but reflections can cause the output to switch falsely (C), thus sending an errant low-going signal down the line.

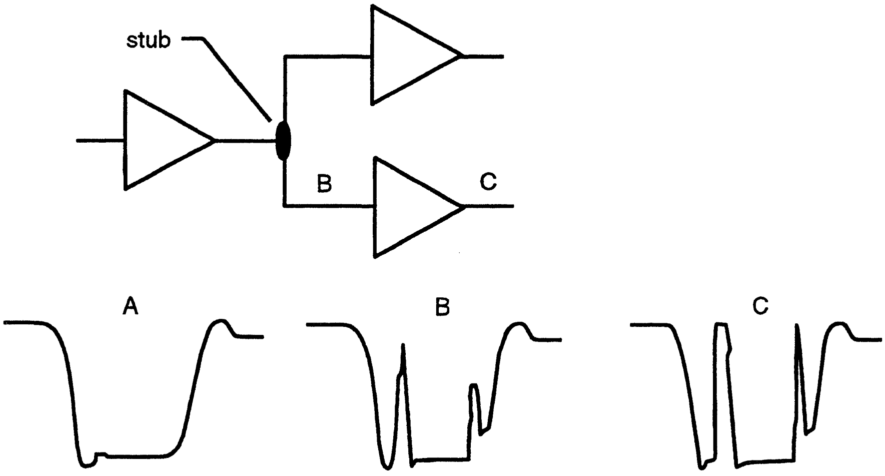

Figure 1-1 Effect of a tee (stub) on a buffer

If you must use tees, use as few as possible, and for each tee, keep the length as short as possible.

Daisy chain bus elements

Connect bus nodes (circuit devices such as buffers, drivers, receivers or any other ICs) in serial daisy-chain fashion. That is, directly connect the output of one node to the input of the next node in a chain. Although this technique may take extra board-layout design time initially, the extra effort will keep unwanted reflections and circuit noise to a minimum. If there are many nodes to be connected to a bus, you may want to reduce the number of etches by adding tees; however, as discussed above, adding tees can increase the noise level and unreliability of a circuit, and is not recommended.

Chapter 2
Optimizing PC Board Design

General design considerations

The following sections discuss factors to consider when designing boards of any size.

Multilayer PC board construction

Using multilayer construction for PC boards is one way to tighten the circuit parameters and reduce potential analog problems on the board. Multilayer design is important especially on boards that have long buses. It can also benefit boards with short buses. In multilayer design, there is a Vcc plane, a ground plane, and one or more signal planes. The embedded power planes offer distributed capacitance and controlled impedance for the signal etches.

Using NetROM's interface connector

NetROM provides two 100-pin connections to emulation pods 0 through 3. Depending on the device being emulated, different types of emulation cables are available which connect to the pod connectors. The emulation cables differ in the cable configuration, whether active components are included, and in the target connection mechanism. The cables connect to the target:

- By using plugs that connect to a target's existing memory DIP, PLCC sockets; or
- By using either a 50-pin or a 60-pin header that plug into a special mating connector that is designed into the target board. Note, NetROM 500 series active cables support 60-pin connectors and NetROM 500 series passive cables support 50-pin connectors; or

- By using adapters that are soldered to the target.

Existing memory DIP or PLCC sockets

The memory-socket connection is the easiest to connect. Typically, when a target is designed, PLCC or DIP sockets are used for memory devices and ROMs are plugged into the sockets. Applied Microsystems supplies cables with plugs for these sockets.

A drawback to this approach is that the sockets were not designed for external cables, and the pin connections could become unstable and the long cables can fall out of the sockets or make intermittent connections. Another drawback is potentially inserting the plugs backwards into the sockets, which can damage NetROM, the target, or both.

Surface mount connections

Surface mount adapters or sockets can be soldered to the device pads on the target circuit board. Applied Microsystems supplies cables with surface mount adapters and sockets.

A drawback to solder-down adapters is that the connection is fragile and if the adapter is disturbed, the connection could break, lifting the pads from the target. In addition, when the design is complete, the adapter or socket must be replaced by the component, making it difficult to reconnect NetROM in the future.

50-pin cables

We recommend, if possible, that one 50-pin connector be designed on the board for each pod to be used, with a maximum of four connectors per NetROM box. This gives the cabling system a robust mechanical advantage so if the cable is accidentally tapped or the target is moved, there will not be an intermittent problem caused by a marginal socket or soldered connection. When design is complete and NetROM is disconnected, the connectors remain on the board, making it easy to reconnect NetROM in the future. Production units can have this connector removed to save the connector cost.

If you wish to design matching connectors into your target's PC board, use the following connectors:

Yamaichi 50-pin connectors

Right-angle	Part number NFP-50A-0132
Straight	Part number NFP-50A-0134

See Chapter 4 for pin assignments.

60-pin cables

We recommend, if possible, that one 60-pin connector be designed on the board for each pod to be used, with a maximum of two connectors per NetROM. This gives the cabling system a robust mechanical advantage so if the cable is accidentally tapped or the target is moved, there will not be an intermittent problem caused by a marginal socket or soldered connection.When design is complete and NetROM is disconnected, the connectors remain on the board, making it easy to reconnect NetROM in the future. Production units can have this connector removed to save the connector cost.

If you wish to design matching connectors into your target's PC board, use the following connectors:

3M 60-pin connectors

Right-angle	Part number 81060-500303
Straight	Part number 81060-600303

See Chapter 4 for pin assignments.

Design considerations for boards with long TTL buses

Design considerations for long TTL buses are different than those for short buses. Here are two examples:

- ❑ The longer etch adds extra capacitance. Think of the etch as a capacitor that causes the signal to have slower rise and fall times. On a short bus, the rise and fall times have sharper edges with less undershoot or overshoot.
- ❑ Additional sockets add capacitance because there are more sockets on long buses. Also the IC socket pins connecting to the trace increase capacitance. Moreover, a driver or receiver in the IC itself has some capacitance effects. Because the etch leaves the controlled impedance of the PC board and goes up to a socket or some mechanical device, controlled impedance is lost, which may cause problems.

For long buses there are ways to clean up the signals and mitigate any adverse effects. When long-bus board designs are optimized, not only will the board function better and more efficiently, but such a design will facilitate NetROM's working with the board. The following paragraphs contain additional recommendations.

Divide buses into branches and add buffers

Instead of having 10 to 12 nodes (drivers or receivers) or more, divide the bus into two sections, each with five to six nodes (or more). Branch the bus close to its origin, and put a buffer on each of the shorter buses. Figure 2-1 shows how one trace or one line on a long bus can be divided into two smaller lines, each driven by its own buffer.

About 10 to 12 nodes is the practical limit for most buses. Most buses - control, address, and data types - can have buffers added to clean up the signals.

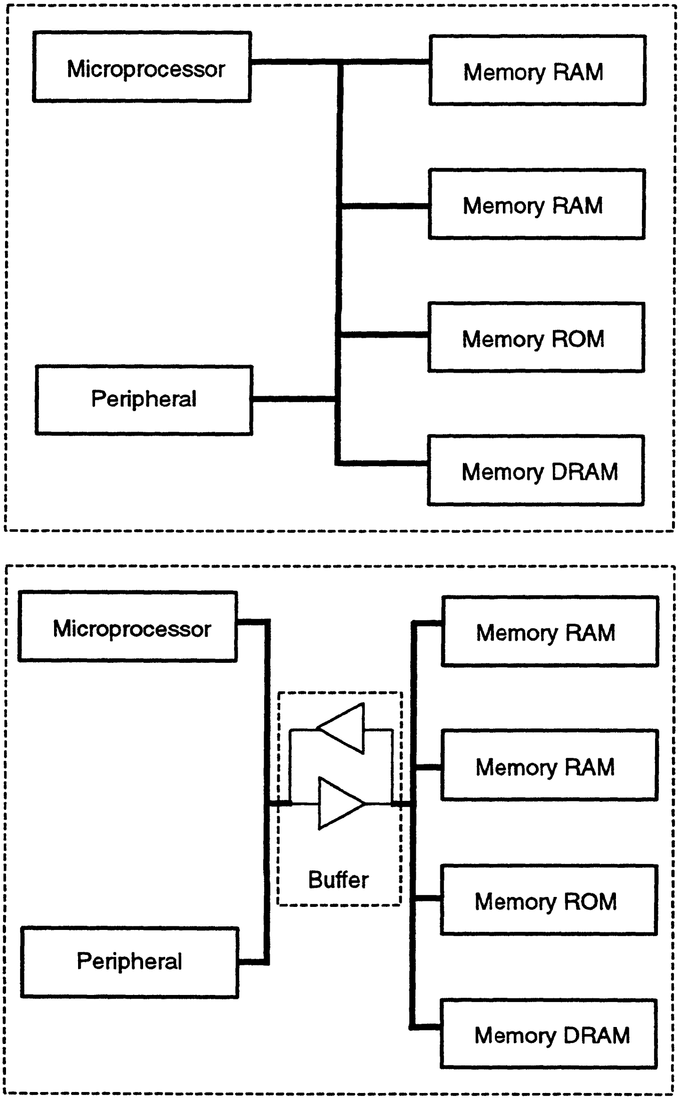

Figure 2-1 Dividing a long bus and adding buffers

However, a buffer will add some propagation delay, thus slowing the circuit. The beneficial effect is a buffer will reduce capacitance and reflections inherent with longer buses, and, therefore, increase circuit speed. So the added propagation delay through the buffers is mitigated by the fact that there is less capacitance.

Note

Empirical evidences shows that circuitry on two shorter buses runs better than circuitry run on one longer bus with no buffers.

You will want to add buffers in cases where there are many loads. Just adding a 12-inch cable segment will degrade the signal. An example of a suitable buffer is the one that NetROM uses as an input buffer, which is an IDT (Integrated Device Technology) 74FCT162244TPV.

Terminate at both ends of the bus

For long bi-directional data buses, we recommend terminating the bus at both ends, rather than just at one end. Use either parallel or Thevenin termination.

Thevenin termination on both ends provides resistors, which are tied to VCC and ground that will absorb any noise or adverse reflections on the wavefront. If, for example, one of the mid-bus drivers is active, the signal will travel to both ends of the bus. Thevenin termination also balances the impedance of drivers and receivers, thereby reducing adverse reflections and noise from the bus. Figure 2-2 shows a long bus with dual Thevenin termination.

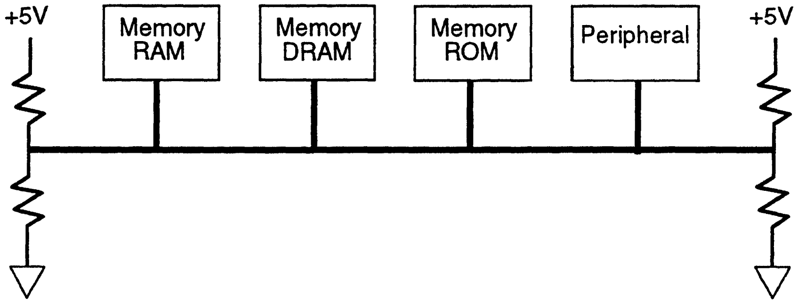

Figure 2-2 A long bus with dual Thevenin termination

Connect NetROM at the end of the bus

Because the NetROM cables can be plugged either into existing ROM sockets or into connectors added to the target's PC board, it is beneficial during the design phase to place the intended sockets or connectors at one end of the TTL bus. In any event, try to avoid connecting NetROM to any tee or stub in the middle of a bus.

Chapter 3

NetROM's Cable Termination

A NetROM cable termination circuit consists of these two elements:

- Receiver
- RC termination

Input receivers

NetROM also uses a special receiver in each circuit, which is an IDT (Integrated Device Technology) 74FCT162244TPV. The receiver connects in series between the signal line and the emulation RAM in NetROM. The receiver has typically 200 millivolts of hysteresis.

RC termination

NetROM has an RC termination circuit on each line of a bus. This consists of a nominal 68 ohm pull-up resistor placed between the signal line and NetROM's VCC (+5.0 Volts) and a 100 pf capacitor to ground.

Figure 3-1 illustrates the termination circuit inside NetROM. The figure is for one line of a bus. Because there are four pod connections and 31 bus lines (20 address, 8 data, and 3 control) per pod, there are 124 separate termination circuits within NetROM, one circuit for each line.

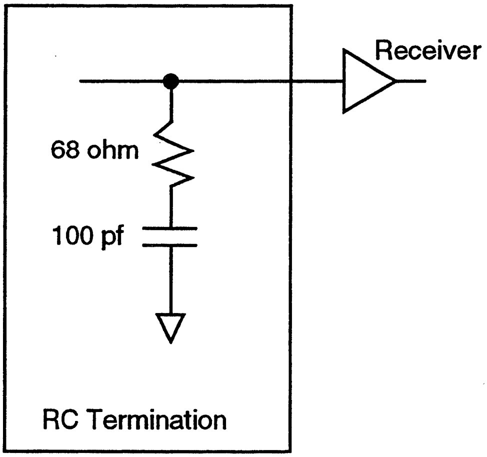

Figure 3-1 NetROM's internal termination circuit

Chapter 4
Using Mass Terminated Connectors

If you are designing your target from conception, you may want to add one, two, three, or four dedicated NetROM interface connectors to your target board for flat ribbon cables. This type of connection provides more reliability than possible by plugging connectors directly into memory sockets on your board. Refer to "Using NetROM's interface connector" on page 2-1.

Contents **Page**

Interface connectors

50-conductor ribbon cable

Each 50-conductor cable has independent control, address, and data buses that can be used for ROM emulation. Figure 4-1 shows the pin assignments on the cable connector. Table 4-1 shows the pin assignments for the signals.

Table 4-1 Pin assignments for 50-pin target cable connections

Pin	NetROM Signal	Pin	NetROM Signal
1	no connection	26	ground
2	A0	27	no connection
3	A1	28	no connection
4	A2	29	no connection
5	A3	30	no connection
6	ground	31	ground
7	A4	32	OE*
8	A5	33	Target Vcc
9	A6	34	CE*
10	A7	35	ground
11	ground	36	WE*
12	A8	37	no connection
13	A9	38	no connection
14	A10	39	no connection
15	A11	40	ground
16	ground	41	D7
17	A12	42	D6
18	A13	43	D5
19	A14	44	D4
20	A15	45	ground
21	ground	46	D3
22	A16	47	D2
23	A17	48	D1
24	A18	49	D0
25	A19	50	ground

* indicates a low-true state

60-conductor ribbon cable

Each 60-conductor cable has independent control, address, and data buses that can be used for ROM emulation. Figure 4-1 shows the pin assignments on the cable connector. Table 4-1 shows the pin assignments for the signals.

Table 4-2 Pin assignments for 60-pin target cable connections

Pin	NetROM Signal	Pin	NetROM Signal
1	ground	31	A18
2	CE0*	32	A19
3	OE*	33	A20
4	WE*	34	A21
5	Target Vcc	35	ground
6	BYTE*	36	ground
7	no connection	37	ground
8	no connection	38	ground
9	ground	39	ground
10	ground	40	ground
11	ground	41	ground
12	ground	42	ground
13	A0	43	CE2*
14	A1	44	CE1*
15	A2	45	D0
16	A3	46	D1
17	A4	47	D2
18	A5	48	D3
19	A6	49	D4
20	A7	50	D5
21	A8	51	D6
22	A9	52	D7
23	A10	53	D8

Table 4-2 Pin assignments for 60-pin target cable connections

Pin	NetROM Signal	Pin	NetROM Signal
24	A11	54	D9
25	A12	55	D10
26	A13	56	D11
27	A14	57	D12
28	A15	58	D13
29	A16	59	D14
30	A17	60	D15

* indicates a low-true state

Control lines

The output enable (OE*), chip enable (CE*), and write enable (WE*) lines are driven just like the output enable, chip enable, and write enable of a PROM device. Each line is low-true.

For 60-pin cables, if your device has only one chip enable, connect it to CE0*. Leave CE1* and CE2* unconnected.

Address lines

A0 to A21 are the address inputs to NetROM's RAM.

Data lines

D0 to D15 are bi-directional data lines. Some type of termination on the target board should be added to each data line on long-bus designs.

Grounds

The ground pins should be connected to the target's ground plane.

Target Vcc

NetROM uses the target's Vcc to switch on the pod LEDs on the NetROM front panel and to determine when to electrically connect to the target connection.

Byte enable (BYTE*)

Byte enable should be used as the equivalent pin as found on some flash devices. It controls whether the interface responds in 8-bit or 16-bit mode. When active (BYTE*=0) the interface is byte wide, where data is on D0-D7 and D15 then becomes the lowest order address. If the device being emulated does not have a BYTE* pin, leave the BYTE* pin unconnected.

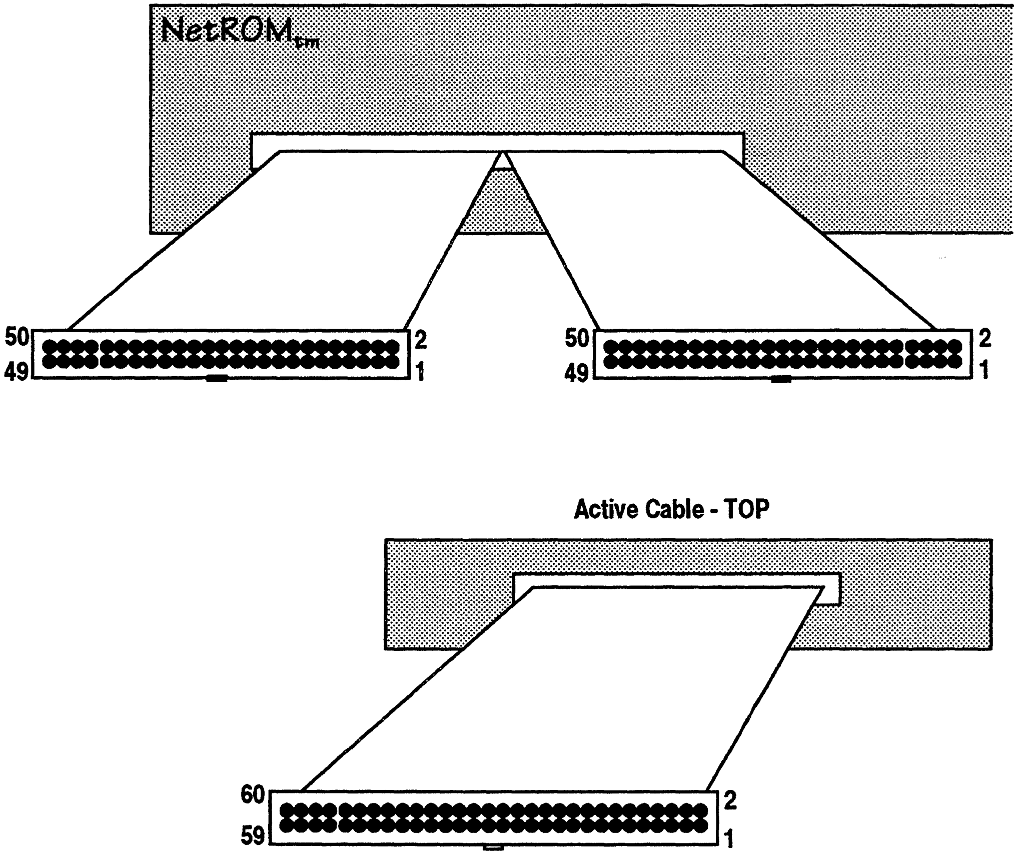

Figure 4-1 Cable connector pinouts

Target to NetROM connections

The device you are emulating determines how you should connect the target and NetROM signals. The following sections list the target-to-NetROM connections for supported devices. The device lists may be incomplete. See the Applied Microsystems World Wide Web page (http://www.amc.com) for current devices.

8-bit devices less than or equal to 1 MB

Devices that are less than or equal to 1 MB in size can use a NetROM passive cable with a 50-pin header. Table 4-3 applies to the following devices:

AMD Am27C64
AMD Am27C010-P
AMD Am27H010
AMD Am27C020-P
AMD Am27C128
AMD Am27C256
AMD Am27C512
AMD Am28F256-P
AMD Am28F512-P
AMD Am28F010-P
AMD Am28F010A-P

AMD Am28F020-P
AMD Am28F020A-P
AMD Am29F010-P
AMD Am29F020
AMD Am29F040
AMD Am29F080
AMD Am29LV004B

Intel 27C010
Intel 27C020
Intel 27C256
Intel 27C512
Intel 28F010
Intel 28F020
Intel 28F256
Intel 28F512
Intel 28F001B
Intel 28F004
Intel 28F008

Atmel AT48F080
SGS-Thomson M28V841
SGS-Thomson M28F841
Mitsubishi M5M27C101
NEC uPD27C1001AD
NEC uPD27C2001AD
Texas Instruments TMS27C128
Texas Instruments TMS27C256

Table 4-3 Target-to-NetROM connections

Target Bus Signal	NetROM Signal 50-pin header
OE*	OE*
CE*	CE*
WE*	WE*
Vcc	Target Vcc
A0	A0
A1	A1
A2	A2
...	...
A19	A19
A20	no connection
A21	no connection
A22	no connection
A23	no connection
D0	D0
D1	D1
D2	D2
...	...
D7	D7

8-bit and 16-bit devices less than or equal to 4 MB

No BYTE* pin

Devices that are less than or equal to 4 MB in size which do not use the BYTE* pin can use NetROM active cables with 60-pin headers. Table 4-4 applies to the following devices:

AMD Am27C040

AMD Am27C1024

AMD Am27C2048

AMD Am27C4096

Intel 27C040

Intel 27C210

Intel 27C220

Atmel AT29C1024 16-bit)

Atmel AT49F8192 (16-bit)

Mitsubishi M5M27C202

NEC uPD27C1024

SGS-Thomson 27C4002

SGS-Thomson M28V161

Texas Instruments TMS27C240

Texas Instruments TMS27PC240

Toshiba TC574000

Table 4-4 Target-to-NetROM connections (no BYTE* pin)

Target Bus Signal	NetROM Signal (60-pin header)	NetROM Signal (60-pin header)	NetROM Signal (2 60-pin headers)
	8-bit device	16-bit device	32-bit device
OE*	OE*	OE*	OE*
CE0*	CE0*	CE0*	CE0*
CE1*	no connection	CE1*[1]	CE1*[1]
CE2*	no connection	no connection	CE2*[1]
WE*	WE*	WE*	WE*
Vcc	Target Vcc	Target Vcc	Target Vcc
A0	A0	no connection	no connection
A1	A1	A0	no connection
A2	A2	A1	A0
A3	A3	A2	A1
...	...	...	...
A19	A19	A18	A17

Table 4-4 Target-to-NetROM connections (no BYTE* pin)

Target Bus Signal	NetROM Signal (60-pin header)	NetROM Signal (60-pin header)	NetROM Signal (2 60-pin headers)
	8-bit device	**16-bit device**	**32-bit device**
A20	A20	A19	A18
A21	A21	A20	A19
A22	no connection	A21	A20
A23	no connection	no connection	A21
D0	D0	D0	D0
D1	D1	D1	D1
D2	D2	D2	D2
...	...	...	...
D7	D7	D7	D7
D8	no connection	D8	D8
...	no connection	...	...
D15	no connection	D15	D15

1. If CE1* and CE2* are not used on the device being emulated, leave the pins unconnected at the 60-pin header.

BYTE* pin

Devices that are less than or equal to 4 MB in size which use the BYTE* pin can use NetROM active cables with 60-pin headers. Table 4-5 applies to the following devices:

AMD Am27C400
AMD Am29F100
AMD Am29F200
AMD Am29F400
AMD Am29LC800T
AMD Am29F016

Intel 27C400
Intel 28F200
Intel 28F400
Intel 28F800
Intel 28F016
Intel 28F032

Atmel AT27C8192
Hitachi HN624316
OKI MSM27C16227B
Mitsubishi M5M28F102
Sharp LH534000

Note

When used in 8-bit mode, connect AMD Am29F016, Intel 28F016, and Intel 28F032 according to Table 4-4.

Table 4-5 Target-to-NetROM connections (BYTE* pin)

Target Bus Signal	NetROM Signal (60-pin header)	NetROM Signal (60-pin header)	NetROM Signal (2 60-pin headers)
	8-bit device	**16-bit device**	**32-bit device**
OE*	OE*	OE*	OE*
CE0*	CE0*	CE0*	CE0*
CE1*	no connection	CE1*[1]	CE1*[1]
CE2*	no connection	no connection	CE2*[1]
WE*	WE*	WE*	WE*
Vcc	Target Vcc	Target Vcc	Target Vcc
BYTE*[2]	BYTE* (low)	BYTE* (high)	BYTE* (high)
A0	D15[2]	no connection	no connection
A1	A0	A0	no connection
A2	A1	A1	A0

Table 4-5 Target-to-NetROM connections (BYTE* pin)

Target Bus Signal	NetROM Signal (60-pin header)	NetROM Signal (60-pin header)	NetROM Signal (2 60-pin headers)
	8-bit device	16-bit device	32-bit device
A3	A2	A2	A1
...	...	...	...
A19	A18	A18	A17
A20	A19	A19	A18
A21	A20	A20	A19
A22	A21	A21	A20
A23	no connection	A22	A21
D0	D0	D0	D0
D1	D1	D1	D1
D2	D2	D2	D2
...	...	..	...
D7	D7	D7	D7
D8	no connection	D8	D8
...	...	...	...
D15	no connection	D15	D15

1. If CE1* and CE2* are not used on the device being emulated, leave the pins unconnected at the 60-pin header.
2. When BYTE* is asserted, D15 becomes NetROM's lowest order address bit.

Plugging cables into target connectors

Before applying power, take care to plug the cables correctly into the target interface connectors.

Caution ⚠

Plugging a cable in backwards can cause the target power supply to reverse drive the NetROM power supply. This would connect NetROM's plus and minus supplies to the target's minus and plus supplies creating a direct short circuit. The target or NetROM or both could be damaged.

On some sockets, ground and power pins are in opposite corners of the socket, so switching the cable switches the ground and power pins. Unfortunately, we cannot design against installing the cables backwards.

Pin 1 is clearly marked on both sockets and cables. Pin 1 on the cables has a red or blue identifier marking; pin 1 on DIP sockets is marked with a "1" or an arrow; and PLCC sockets have a little key. This key on the PLCC socket makes it difficult to put the PLCC plug in backwards.

Cable status LEDs

There are four status LEDs on the NetROM front panel; one LED for each cable. When you plug the cable correctly into the target, the corresponding NetROM green LED lights. If the cable is plugged in backwards, the LED will not light.

Chapter 5
Using the Command Status Connector

In addition to the ROM emulation connectors (described in Chapter 4), NetROM has a special 20-pin command status connector on its front panel. Use of this connector is optional.

When this connector is attached to a target the command status feature permits the target to send status signals and control signals to NetROM. It provides an external write line that, when driven low, allows the target to write to the emulation ROM (RAM) within NetROM. You can also connect to the write and write signals using a jumper cable. See page 3-29 and page 3-31 for instructions.

The part number of the 20-pin mating command status connector is 3M 3421. Figure 5-1 shows how the NetROM pins are labeled. Table 5-1 shows the pin assignments for the connector.

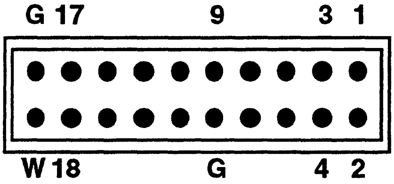

Figure 5-1 Command status connector pin labels

Table 5-1 Pin assignments for command status connector

Pin	ID	Assignment	Pin	ID	Assignment
1	R	TAR_CTL0*	11		TAR_STS0*
2	1	TAR_CTL1*	12		TAR_STS1*
3	2	TAR_CTL2*	13		TAR_STS2*
4	3	TAR_CTL3*	14		TAR_STS3*
5	4	TAR_CTL4*	15		TAR_STS4*
6	5	TAR_CTL5*	16		TAR_STS5*
7	6	TAR_CTL6*	17		TAR_STS6*
8	7	TAR_CTL7*	18		TAR_STS7*
9		no connection	19		ground
10		ground	20	W	TAR_WR*

Notes

Pin 1 (R) is usually used for reset.

TAR_CTL# is a Target Control Line.

TAR_STS# is a Target Status Line.

TAR_WR is a Target Write Line.

* indicates a low-true state.

Signals

There are eight target control signal lines, TAR_CTL0* through TAR_CTL7*. NetROM's open collector driver drives these lines. Each line must be terminated through a nominal 1.5-kohm resistor to VCC (+5.0 V) for proper operation.

There are also eight target status signal lines, TAR_STS0* through TAR_STS7*. NetROM reads the TTL signals on these lines off the cable.

The external write line, TAR_WR*, when driven low, allows data to be written to NetROM's RAM.

There are two ground lines, pins 10 and 19, which are to be connected to the target's ground plane.

NetROM also reads status lines. The eight status lines, TAR_STS0* through TAR_STS7*, constitute a bus on which the target can send status data to NetROM.

Application

When used together, the status lines, which carry signals from the target to NetROM, and the control lines, which carry signals from NetROM to the target, can communicate bi-directionally between the two devices. You can connect these lines to different parts of your circuit as needed.

- ❑ Connect TAR_CTL0* to your reset logic. When you are programming and are logged into NetROM, you can send out a command to make this line go assertive (low-true) and non-assertive (high-false) to reset your target. That may be easier than going to the target to press a button or turn the target's power off and on, etc.
- ❑ Connect TAR_CTL1* to the non-maskable interrupt (NMI) line of your processor, if you have one. That allows you to cause an interrupt or exception to the process that is going on in the target.
- ❑ There are two ways to have the target modify (write) NetROM's memory.

  If NetROM is connected to a connector or memory device that allows writing, such as a Flash ROM or RAM, then NetROM can pick up the write signal from the existing socket, and through the existing cables using the ROM emulation Connectors. Refer to Chapter 4.

However, in the case of a ROM on your target, there is no write line, because you cannot write to a read-only memory. So you need some way to get a write attribute to NetROM. Use pin 20, the target write line, TAR_WR*, to do this. Some applications require the target to write to NetROM and others do not. Use of this line depends on your application.

Use the *set tgtctl* command (see page 5-51 of the NetROM *User's Manual*) to specify a high true or low true state on any of the target control lines. Display the control signals with *di tgtctl* (see page 5-71 of the *NetROM User's Manual*); display the status signals with *di tgtstatus* (see page 5-72 of the *NetROM User's Manual*).

Chapter 6
Writing to NetROM Memory

Typically, a target writes to NetROM to perform some input-output function not designed into the target. For example, a very small target board that connects to an appliance most likely would not have a console or a monitor or a keyboard.

Console emulation

One way to emulate a console is to use a read-address console. The read-address console does not require the target to write to NetROM. Another way to emulate a console is to have the target write to NetROM through the pod connectors or the command status connectors:

Pod connectors.

If the target's memory device is a Flash ROM, RAM or PROM, then the write pin (WE or W) of that device can be used. This is the method most often used with NetROM if the read-address console is not used. If the target's memory device is a ROM, then there is no write line. In that case, if NetROM cables are connected through 50-pin or 60-pin installed connectors on the target board, the write line in these cables can be used.

Command status connectors.

If the target's memory device is a ROM, there is no write-enable pin at the ROM socket; however, you can use the external write line of the command status connector (refer to Chapter 5). This write-enable feature can be used in one of two ways:

- A 20-pin connector can be added to the target board and connected to the NetROM command status connector via a cable.
- For temporary use, a single wire can be attached to the target write line, pin 20 of the command status connector, and the other end of the wire, with a probe clip, can be

attached to the write-enable connection (wherever it is) on the target.

The following rules apply to all NetROM write cycles, either directly through the pod, or via the external write line:

- NetROM write cycle is similar to a static-ram write cycle.
- There is no algorithm required for writes, such as with Flash devices that use algorithms such as AAAA or 5555.
- The output enable (OE) pin on the pod connector must not be asserted while the target is driving data for the write cycle. NetROM has been designed to emulate PROMs. In a PROM device, if you assert output enable to tell the PROM that you want to read data, and if you also assert the read/write line in a write mode, you are telling the PROM that you want to write to it. NetROM will look at those signals and emulate a PROM. In that case, output enable will take precedence and there will be a read cycle instead of a write cycle. So when you are writing to NetROM, you must have the output enable control signal false (high) during the write cycle.

Timing

When writing to NetROM, certain timing considerations must be met. Figure 6-1 shows the read timing cycle. Figure 6-2 shows timing diagrams for two controlled-write cycles. The upper half of the figure is a read/write control cycle and the lower half is a chip-enable control cycle. See the tables following the figures for the values when using active and passive cables.

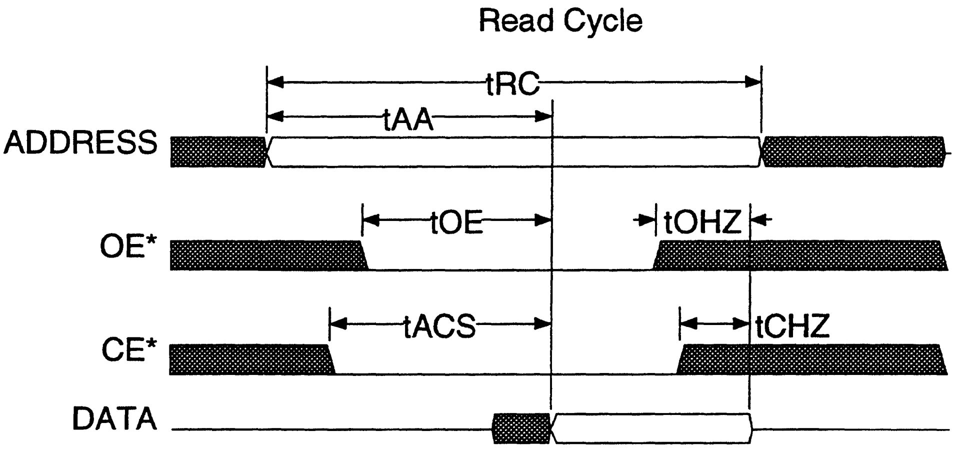

Figure 6-1 Read cycle timing diagram

Table 6-1 Read cycle - worst case

Symbol	Parameter	Passive Cable (nsec) (minimum)	Active Cable (nsec) (minimum)
tRC	Read Cycle Time	40	52
tAA	Address Access Time	40	52
tOE	Output Enable to Output Valid	55	67
tOHZ	Output Disable to Output High-Z	29	12
tACS	Chip Select Access Time	60	37
tCHZ	Chip Deselect to Output High-Z	29	12

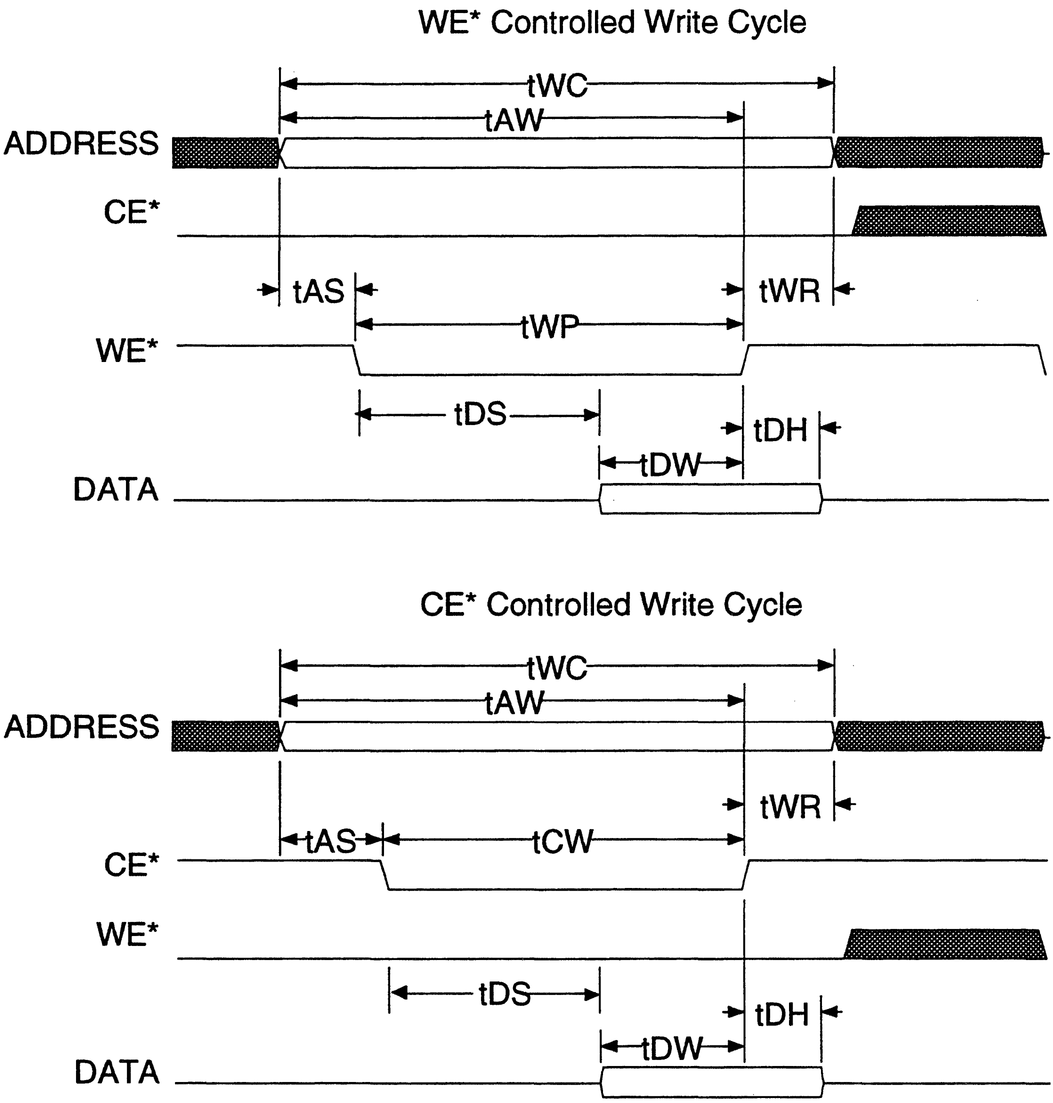

Figure 6-2 Write cycle timing diagram

Table 6-2 Write cycle - worst case

Symbol	Parameter	Passive Cable (nsec) (minimum)	Active Cable (nsec) (minimum)
tWC	Write Cycle Time	70	70
tAW	Address Valid to End of Write	13	13
tAS	Address Setup Time	0	0
tWP	Write Pulse Width	38	38
tCW	Chip Select to End of Write	38	38
tWR	Write Recovery Time	25	25
tDW	Data Valid to End of Write	9	9
tDH	Data Hold Time	25	25
tDS	Data Valid from Start of Write	20 (maximum)	20 (maximum)

Chapter 7
Circuit Design Recommendations

Summary of recommendations

The following list summarizes the circuit design recommendations made in this document:

- Keep the ROM TTL bus length short. Maximum length should be 10 inches before using termination techniques. (Chapter 1)
- Do not use tees (stubs) if you can avoid them. If you must use them, keep the tee length to under a quarter-inch. Keep the number of tees to a minimum. (Chapter 1)
- Use a single driver and a single receiver on a bus whenever possible; otherwise, keep the number to a minimum to reduce noise and reflections. (Chapter 1)
- Connect the output of one node (driver or receiver) directly to the input of another node. (Chapter 1)
- Design dedicated connectors on the target board to ensure more reliable connections than using adapters to plug the NetROM cables into PLCC or DIP memory sockets. For 50-pin headers, on the target use a Yamaichi connector, part number NFP-50A-0132 (right-angle) or part number NFP-50A-0134 (straight). For 60-pin headers, on the target use a 3M connector, part number 81060-500303 (right-angle) or part number 81060-600303 (straight). (Chapter 2)
- Use buffers to clean up noisy signals. If there are more than eight to ten nodes on a bus, add buffers to the bus. An example would be the buffer that NetROM uses as an input buffer, which is an IDT (Integrated Device Technology) 74FCT162244TPV. (Chapter 2)
- Use multilayer PC boards to reduce analog problems and provide more uniform impedance on the bus. Multilayer construction is especially useful on long-bus boards.

- ❑ Divide long buses into branches and add buffers to reduce noise, reflections and other analog problems. (Chapter 2)
- ❑ Terminate at both ends of a long bus to reduce reflections and increase performance. (Chapter 2)
- ❑ Connect NetROM at one end of the bus and not in its middle to reduce reflections. (Chapter 2)
- ❑ Use the command status connector to provide control functions to a target. Also use the command status write line when writing to NetROM if there is no write enable in the target's circuitry. (Chapter 5, Chapter 6)

Suggested reading

- ❑ *1993 IDT High-Speed CMOS Logic Design Guide*, Integrated Device Technology, Santa Clara, California, 1993
- ❑ Pace, Charles. "Terminate Bus Lines to Avoid Overshoot and Ringing." *Electronic Design News*, September 17, 1987
- ❑ Quinnell, Richard. "High-Speed Bus Interfaces." *Electronic Design News*, September 30, 1993
- ❑ Sterner, Rudy. "Guide to Designing with High-Speed CMOS ICs." *Electronic Design*, September 12, 1991

Index