

MVME162
MVME162LX
MVME162FX
Installation and
Configuration Guide

101428-010

Contents

Preface

Package Contents	v
System Requirements	vi
Hardware	vi
Software	vi
Vital Statistics	vii
Board Specifications	vii
On-Board Devices	viii
Supported Microtec Components	viii
Target RAM Size — Minimum Requirement	viii
Notational Conventions	ix
Related Publications	ix
Questions and Suggestions	x

1 Establishing the Spectra Connection

Creating Boot PROMs	1-1
Installing Boot PROMs Into the Target Hardware	1-1
Cabling	1-2
Configuring Ethernet or Serial Interfaces	1-2
Ethernet Connection	1-2
Assigning the Board an IP Address	1-2
Serial Connection	1-3
Updating /etc/remote	1-3
Updating \$SPECTRA/host/etc/connconf	1-4
Starting serial_server	1-5
Connecting to the Target With XSH	1-6
Ethernet	1-6
Serial	1-6
Special Notes for Serial Ports	1-7

2 Configuration Information

Software Configuration	2-1
Memory Map	2-1
Default File	2-3
Bridge	2-3
Console	2-3
Device Driver Configuration Parameters	2-3
Timer	2-3

Figure 2-1.	Memory Map	2-2
Figure 2-2.	MVME162 Board Configuration	2-9
Figure 2-3.	MVME162LX Board Configuration	2-12
Figure 2-4.	MVME162FX Board Configuration	2-15

Table P-1.	Hardware Requirements	vi
Table P-2.	MVME162/LX/FX Board Specifications	vii
Table P-3.	MVME162/LX/FX On-Board Devices	viii
Table P-4.	Notational Conventions	ix
Table 2-1.	Timer 1 Device Driver Configuration Parameters	2-3
Table 2-2.	Timer 2 Device Driver Configuration Parameters	2-4
Table 2-3.	Serial 1 Device Driver Configuration Parameters	2-4
Table 2-4.	Serial 2 Device Driver Configuration Parameters	2-5
Table 2-5.	Ethernet Device Driver Configuration Parameters	2-6
Table 2-6.	VME Device Driver Configuration Parameters	2-6
Table 2-7.	Xconfig Variables	2-7
Table 2-8.	MVME162 PROM Parameters	2-7
Table 2-9.	MVME162 Factory Default Jumper Settings	2-8
Table 2-10.	MVME162LX PROM Parameters	2-10
Table 2-11.	MVME162LX Factory Default Jumper Settings	2-10
Table 2-12.	MVME162FX PROM Parameters	2-13
Table 2-13.	MVME162FX Factory Default Jumper Settings	2-13

B

- Board
 - configuration
 - MVME162 2-7
 - MVME162FX 2-13
 - MVME162LX 2-10
 - layout
 - MVME162 2-9
 - MVME162FX 2-15
 - MVME162LX 2-12
 - specifications vii
- Boot PROMs
 - creating 1-1
 - installing into target hardware 1-1
- boot.hex.tmp file 2-17
- Bridge 1-2, 2-3

C

- Cables 1-2, 2-25
- Configuration parameters
 - Ethernet 2-6
 - serial 2-4
 - timer device driver 2-3
 - VME 2-6
- Configuration, Flash/PROMs 2-26
- connconf file 1-4
- Connection
 - Ethernet 1-2, 1-6
 - serial 1-3, 1-6
 - target 1-6
- Console 2-3

D

- Default file 2-3
- devcnfg.c file 1-7, 2-18
- Device driver configuration parameters 2-3
- Devices, on-board viii
- DRAM parity checking 2-27

E

- /etc/remote file 1-3
- Ethernet
 - address failure 2-22
 - assigning the address 1-2, 2-23
 - boards without 2-27
 - connection 1-2, 1-6

F

- File, default 2-3
- Files
 - boot.hex.tmp 2-17
 - connconf 1-4
 - devcnfg.c 1-7, 2-18
 - /etc/remote 1-3
 - mo162.def 2-1, 2-3
 - mo162ram.def 2-17
- Flash/PROM configuration 2-26

H

- Hardware requirements vi
- Hardware setup
 - board layout
 - MVME162 2-9
 - MVME162FX 2-15
 - MVME162LX 2-12
 - cables 2-25
 - jumper settings
 - MVME162 2-8
 - MVME162FX 2-13
 - MVME162LX 2-10
 - PROMs
 - MVME162 2-7
 - MVME162FX 2-13
 - MVME162LX 2-10

Preface

This guide describes how to install the MVME162/LX/FX Board Support Package (BSP) for use with the Spectra development environment on SunOS, Solaris, and HP-UX versions of UNIX, and Windows NT.

Spectra BSP Installation and Configuration Guides do not supply technical information about a target board beyond what may be needed to run the Spectra development environment on properly configured hardware. Consult the board manufacturer's documentation provided with your target board for details about issues such as serial communication, power lines, memory modules, placement in a card cage, switch settings, daughterboards, port configurations, and start-up procedures.

If you need to set up the target board in an unconventional manner to suit your application, you should investigate the consequences for hardware and software.

Package Contents

Your BSP contains a CD-ROM, one or more Spectra boot PROMs, and this installation guide.

Establishing the Spectra Connection 1

This chapter provides information about the procedures you need to perform to successfully start using your board support package (BSP).

Creating Boot PROMs

Your BSP may include one or more Spectra boot PROMs containing a bootstrap program and communication software for your target board.

If boot PROMs are not supplied, or if you wish to make new boot PROMs, use Xconfig to create the boot image using the command line:

```
xconfig boot.def mo162.def microtec.def
```

For more information on creating boot PROMs, see the Microtec *Board Support Package (BSP) Developer's Guide and Reference*.

Installing Boot PROMs Into the Target Hardware

Set the jumper settings and install the PROMs as described in the section *Hardware Setup* in Chapter 2, *Configuration Information*. Where necessary, also consult the board manufacturer's documentation.

Install the board in the backplane (if any) and apply power.

Configuration Information 2

This chapter provides configuration information for the MVME162, MVME162LX, and MVME162FX boards.

Software Configuration

This section describes the memory map, default files, device driver configuration parameters, and Xconfig variables.

Memory Map

The following memory map (Figure 2-1) uses default **mo162** boot PROMs. The map is defined in **mo162.def**. If any inconsistencies exist, **mo162.def** supersedes this map. This map includes shared memory addresses. If your application does not use shared memory, use Xconfig to configure the mapping of your own system.

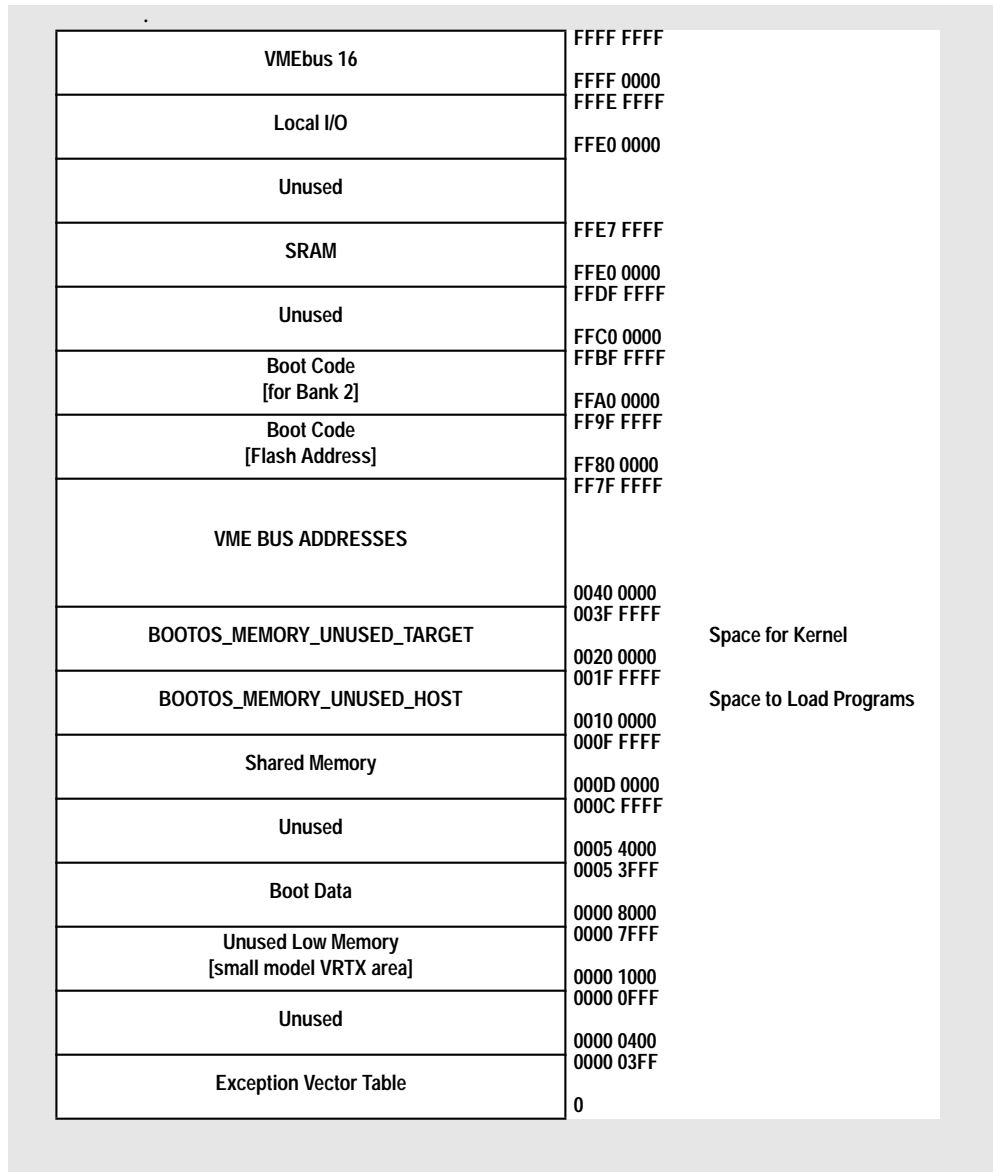


Figure 2-1. Memory Map

Default File

Use the **mo162.def** default file to configure the system for the bridge in boot PROMs.

Bridge

The *logio* device to be used as a bridge is **logio_ether_1_id** (MVME 712 or Front Panel / Ethernet).

Console

By default, the console is **logio_serial_1_id** (MVME162 Front Panel Serial Port 1 / Console).

Device Driver Configuration Parameters

This section describes the timer, serial, Ethernet, and VME device driver configuration parameters.

Timer

Table 2-1 and Table 2-2 list the timer device driver configuration parameters for the MVME162/LX/FX board.

Table 2-1. Timer 1 Device Driver Configuration Parameters

Component	Parameter
ID	logio_timer_1_id
Name	Motorola Peripheral Chip Controller (PCC)
Port	timer 1
Module name	mopcc162
Interface	timer_1 interface
Vector	0x59
Default interrupt rate	10 ms

Table 2-2. Timer 2 Device Driver Configuration Parameters

Component	Parameter
ID	logio_timer_2_id
Name	Motorola Peripheral Chip Controller (PCC)
Port	timer 2
Module name	mopcc162
Interface	timer_1 interface
Vector	0x58
Default interrupt rate	10 ms

Serial

Table 2-3 and Table 2-4 list the serial device driver configuration parameters for the MVME162/LX/FX board.

Table 2-3. Serial 1 Device Driver Configuration Parameters

Component	Parameter
ID	logio_serial_1_id
Location	MVME 162 Front Panel, Serial Port 1 / Console
Name	Zilog Z85230 SCC Serial Communication Controller
Port	A
Module name	zi8530
Interface	serial_2 interface
Vector	Tx — 0x78, Rx — 0x7C, RxError — 0x7E
Packet/tty	tty
Baud	19200
Bits	8
Parity	None

(cont.)

Table 2-3. Serial 1 Device Driver Configuration Parameters (cont.)

Component	Parameter
Stop bits	1
Ctrl_port	0xFFF45005
Data_port	0xFFF45007

Table 2-4. Serial 2 Device Driver Configuration Parameters

Component	Parameter
ID	logio_serial_2_id
Location	MVME 162 Front Panel, Serial Port 2
Name	Zilog Z85230 SCC Serial Communication Controller
Port	B
Module name	zi8530
Interface	serial_2 interface
Vector	Tx — 0x70, Rx — 0x74, RxError — 0x76
Packet/tty	packet
Baud	19200
Bits	8
Parity	None
Stop bits	1
Ctrl_port	0xFFF45001
Data_port	0xFFF45003

Ethernet

Table 2-5 lists the Ethernet device driver configuration parameters for the MVME162/LX/FX board.

Table 2-5. Ethernet Device Driver Configuration Parameters

Component	Parameter
ID	logio_ether_1_id
Location	MVME 712 Transition Module or Front Panel / Ethernet
Name	AMD 7990 Ethernet Controller
Module name	am7990
Interface	ether_1 interface
Vector	0x57
Rx Buffers	64
Tx Buffers	12

VME

Table 2-6 lists the VME device driver configuration parameters for the MVME162/LX/FX board.

Table 2-6. VME Device Driver Configuration Parameters

Component	Parameter
ID	logio_shmem_1_id
Location	P1 & P2 VME connector
Name	VMEchip2
Module name	mo162sm
Interface	shmem_1 interface
Vector	0x62

Xconfig Variables

Table 2-7 lists the Xconfig variables for the MVME162/LX/FX board.

Table 2-7. Xconfig Variables

Variable	Default Value	Description
board.target	m68040	68040 target
board.name	mo162	MVME162/LX/FX
board.boot.code		Start of CODE section either in ROM or RAM
board.boot.data		Start of BSS section in RAM

Hardware Setup

This section describes hardware setup for the MVME162, MVME162LX, and MVME162FX boards.

MVME162 Board Configuration

This section describes the PROMs, switch and jumper settings, and board layout for the MVME162 board.

PROMs

Table 2-8 lists the parameters for PROMs used with the MVME162 board.

Table 2-8. MVME162 PROM Parameters

Component	Parameter
U47	JEDEC 32-pin PLCC capable of 4 Mbit densities organized as a 512Kbit x 8 device
PROM type	27C010
Speed	155 ns or faster Slower PROMs may work but have not been tested.

Jumper Settings

Use the factory default settings for the MVME162 board listed below in Table 2-9.

Table 2-9. MVME162 Factory Default Jumper Settings

Jumper Location	Jumper Pins	Description
J1	1-2	System Controller
J11	open	Serial Port 1 / Internal Clock
J12	open	Serial Port 2 / Internal Clock
J20	1-3, 2-4	VMEbus +5V Standby
J21	1-2	user-definable
J22	1-2	Connected
	3-4	user-definable
	5-6	user-definable
	7-8	user-definable
	9-10	Open = PROM, Connected = Flash
	11-12	user-definable
	13-14	user-definable
	15-16	user-definable

Note

If these settings are different from the board manufacturer's manual, the board manufacturer's manual takes precedence.

Board Layout

Figure 2-2 shows the board layout for the MVME162.

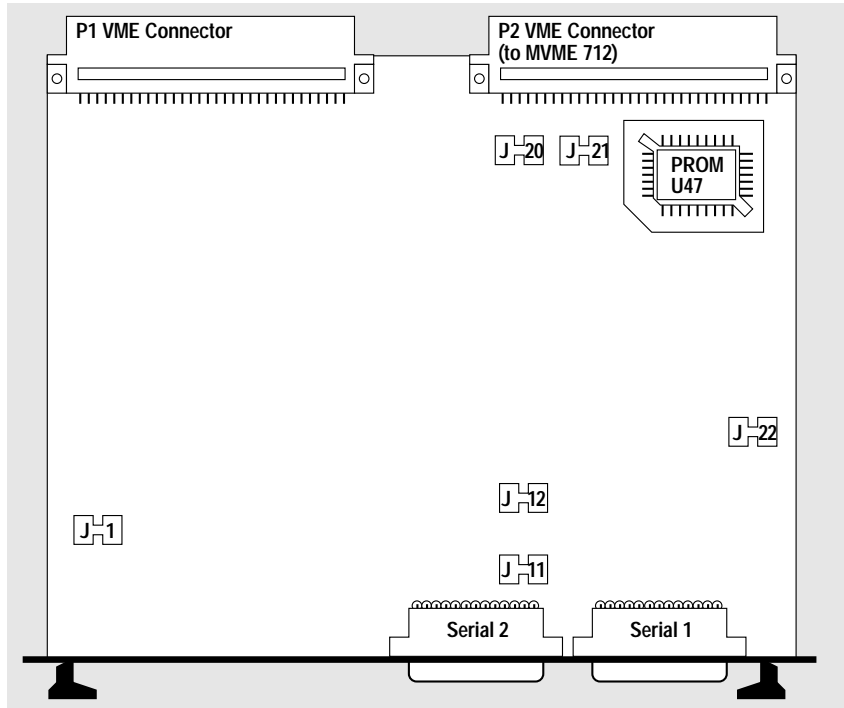


Figure 2-2. MVME162 Board Configuration

MVME162LX Board Configuration

This section describes the PROMs, switch and jumper settings, and board layout for the MVME162LX board.

PROMs

Table 2-10 lists the parameters for PROMs used with the MVME162LX board.

Table 2-10. MVME162LX PROM Parameters

Component	Parameter
U24, U23, U22, U21	32-pin DIP capable of 8 Mbit densities organized as 128 Kbit x 8 or 256 Kbit x 8 or 512 Kbit x 8 or 1 Mbit x 8 device
PROM type	27C010
Speed	155 ns or faster Slower PROMs may work but have not been tested.

Jumper Settings

Use the factory default settings for the MVME162LX board listed below in Table 2-11.

Table 2-11. MVME162LX Factory Default Jumper Settings

Jumper Location	Jumper Pins	Description
J1	1-2	System Controller
J11	1-2	user-definable
	3-4	user-definable
	5-6	user-definable
	7-8	Open = PROM, Connected = Flash
	9-10	user-definable
	11-12	user-definable
	13-14	user-definable

(cont.)

Table 2-11. MVME162LX Factory Default Jumper Settings (cont.)

Jumper Location	Jumper Pins	Description
J12	15-16	Connected
		Defines EPROM / Flash configuration:
	3-4, 5-6, 9-11, 10-12	128Kbit x 8 EPROMs
	3-4, 9-11, 10-12	256Kbit x 8 EPROMs
	5-6, 8-10, 9-11	512Kbit x 8 EPROMs
	7-9, 8-10	1 Mbit x 8 EPROMs
J13	1-2, 7-9, 8-10	1 Mbit x 8 (Flash disabled)
	1-3, 2-4	SRAM backup power source; Primary and secondary source VMEbus +5V Standby
J14	1-2	on-board SCSI bus terminator enabled

Note

If these settings are different from the board manufacturer's manual, the manual takes precedence.

Board Layout

Figure 2-3 shows the board layout for the MVME162LX.

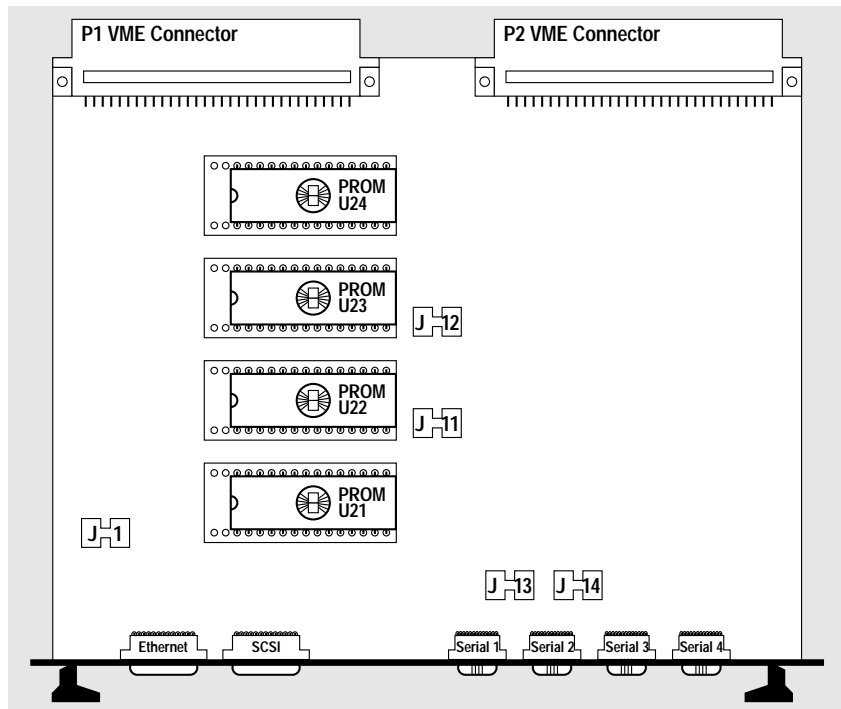


Figure 2-3. MVME162LX Board Configuration

MVME162FX Board Configuration

This section describes the PROMs, switch and jumper settings, and board layout for the MVME162FX board.

PROMs

Table 2-12 lists the parameters for PROMs used with the MVME162FX board.

Table 2-12. MVME162FX PROM Parameters

Component	Parameter
U47	JEDEC 32-pin PLCC capable of 4 Mbit densities organized as a 512Kbit x 8 device
PROM type	27C010
Speed	155 ns or faster Slower PROMs may work but have not been tested.

Jumper Settings

Use the factory default settings for the MVME162FX board listed below in Table 2-13.

Table 2-13. MVME162FX Factory Default Jumper Settings

Jumper Location	Jumper Pins	Description
J1	1-2	System Controller
J11	open	Serial Port 1 / Internal Clock
J12	open	Serial Port 2 / Internal Clock
J20	1-3, 2-4	VMEbus +5V Standby
J21	2-3	4 Mbit PROM
J22	1-2	Connected
	3-4	user-definable
	5-6	user-definable

(cont.)

Table 2-13. MVME162FX Factory Default Jumper Settings (cont.)

Jumper Location	Jumper Pins	Description
	7-8	user-definable
	9-10	Open = PROM, Connected = Flash
	11-12	user-definable
	13-14	user-definable
	15-16	user-definable
J23	open	not used
J24	1-2	8 Mhz IP Bus Clock

Note

If these settings are different from the board manufacturer's manual, the manual takes precedence.

Board Layout

Figure 2-4 shows the board layout for the MVME162FX.

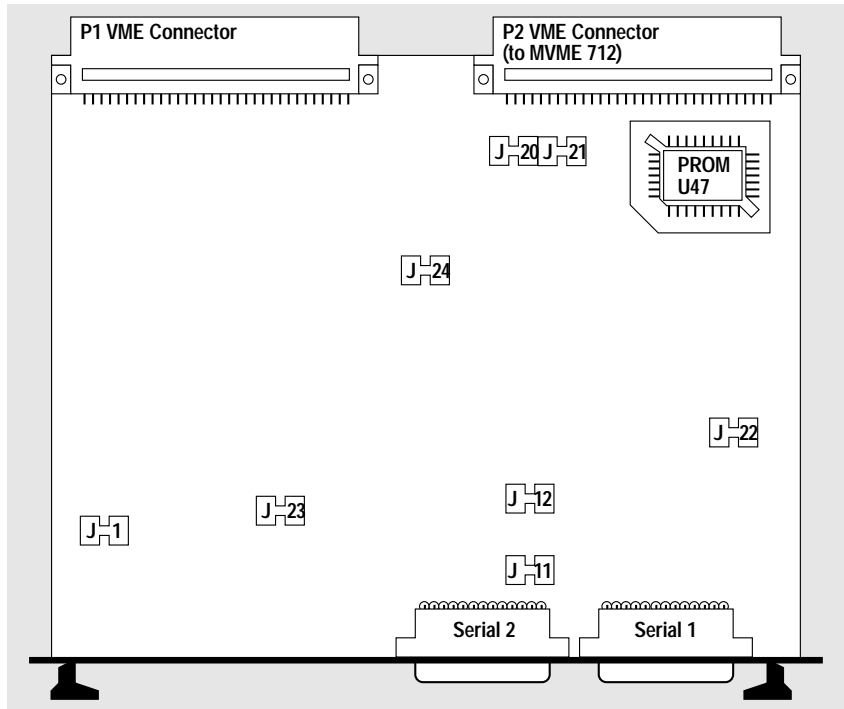


Figure 2-4. MVME162FX Board Configuration

Supplementary Notes

Booting Directly From Spectra PROMs

The quickest method of booting the board is directly from Spectra boot PROMs. Use the file **mo162.def** to generate boot PROMs and make certain J22 pins 9-10 are open.

MVME162Bug ROMBoot and Spectra Bridge

The Spectra bridge is also a Motorola ROMBoot compliant module. The start location of this module is at offset 0x18 from the start of the boot code starting address. To access the Motorola diagnostic routines prior to booting Spectra:

1. Create **boot.hex** using **mo162romboot.hex**. This creates an image identical to **mo162.def** except that the code address has been modified from 0xff800000 to 0xffa00000.
2. Burn this image into a PROM and install the PROM into the board.
3. Using MVME162Bug, enter the following parameters:

```
ROM Boot Enable [Y/N]           = Y?  
ROM Boot at power-up only [Y/N] = N?  
ROM Boot Enable search of VMEbus [Y/N] = N?  
ROM Boot Abort Delay            = 5?  
ROM Boot Direct Starting Address = FFA00018?  
ROM Boot Direct Ending Address  = FFA000F0?
```

The abort delay may be reduced to speed up MVME162 ROMBoot time. The starting address must be specified in this case as FFA00018, which is 0x18 from the beginning of the boot code image. The value of FFA000F0 was used to limit the boot search time in case of an error.

The user may abort the Spectra boot by pressing the Abort button on the front panel or by pressing the Break key. If the ROMBoot sequence is halted by either the Abort or Break method, the sequence can be initiated by using the command:

```
162-Bug>rb;v
```

Downloading Bridge Into RAM Using MVME162Bug

You can use the MVME162Bug monitor to download the bridge into the target memory when PROMs are not available.

Serial Download

Create a file, **mo162ram.def**:

```
@include(mo162.def)
board.boot.code:20000
board.boot.data:35000

# memory map

board.memory:code,data,shmem,host,target,
             flash,rom,sram,io,vme16

board.memory.code.address: 0x20000
board.memory.code.size: 0x15000
board.memory.code.type: BOOTOS_MEMORY_SHARED|
                        BOOTOS_MEMORY_READONLY|
                        BOOTOS_MEMORY_COPYBACK

board.memory.data.address: 0x35000
board.memory.data.size: 0x40000

board.memory.flash.address: 0xff800000
board.memory.rom.address: 0xffa00000
```

Use Xconfig to generate the file **boot.hex.tmp** image in the current directory:

```
xconfig bootconf.def mo162ram.def microtec.def
```

From the current directory, start a **tip** session to the MVME162:

```
Copyright Motorola Inc. 1988 - 1992, All Rights Reserved

MVME 162 Debugger/Diagnostics Release Version 1.1 - 12/18/92
COLD Start

Local Memory Found =00400000 (&4194304)

MPU Clock Speed =25Mhz

162-Bug>lo 0
~>Local File Name? boot.hex.tmp

3565 lines transferred in 2 minutes 46 seconds
!

162-Bug>
162-Bug>
```

Press the Return key to display the prompt:

```
162-Bug>go 2000c
Effective address: 002000C
```

If MVME162Bug is at 9600 baud, and the Spectra console is at 19200 baud, garbled characters will be displayed. Allow three sequences of characters to be displayed before exiting **tip**. In some cases, **tip** will lock the serial port if it attempts to exit while garbled data is being transmitted to the port. Exit **tip** and then restart it at 19200. Any reset from the XSH prompt will display the correct console characters. You can change the default baud rate from 19200 to 9600 by modifying the entry in **devcnfg.c**.

Booting Directly From tftpboot

There are two methods for booting directly from **tftpboot**:

- using the boot shell method without modifying the PROMs
- creating PROMs

The following sections detail each of these methods.

Using the Boot Shell Method Without Modifying PROMs

Create a **mo162** boot image where the boot code is in RAM. For example, create a file **user.def** that contains the following:

```
@include (boot.def)
@include (mo162.def)
@include (microtec.def)

board.memory: code,data,host,target

board.boot.code: 100000
board.boot.data: 115000

board.memory.code.address: 0x100000
board.memory.code.size: 0x15000
board.memory.code.type: BOOTOS_MEMORY_SHARED|
    BOOTOS_MEMORY_READONLY|
    BOOTOS_MEMORY_COPYBACK

board.memory.data.address: 0x115000
board.memory.data.size: 0x40000

board.memory.host.address: 0x155000
board.memory.host.size: 0xab000

board.memory.target.address: 0x200000
board.memory.target.size: 0x200000
```

Use Xconfig to create the boot image:

```
xconfig user.def
```

Use the **s2s** command to specify the entry point of the starting code:

```
s2s -e 0x10000c boot.hex mo162.boot
```

Copy the **mo162.boot** file to your server's **tftpboot** directory. To boot directly from **tftpboot**, change the following in the Spectra boot PROM shell:

```
Spectra Boot version 4.1
Copyright (c) 1992-1996 Microtec
Warm reset
Type any key within 2 seconds to get shell prompt
boot> printenv
VERBOSITY=3
NODENAME=default
NAMESERVER=default
BOARD=DEV_BOARD
BRIDGE=DEV_ETHER_1
TIMER=DEV_TIMER_1
CONSOLE=DEV_SERIAL_1
TFTP_DEVICE=DEV_ETHER_1
BOOT_ORDER=shell:rarp:xtrace
```

Set your boot order by typing:

```
boot>setenv BOOT_ORDER shell:tftp
```

Set your target IP address by typing:

```
boot>setenv TFTP_DEVICE_IP 149.147.5.33
```

Set your host IP address by typing:

```
boot>setenv TFTP_HOST_IP 149.147.5.44
```

Set the **tftpboot** filename (the default is the target IP address):

```
boot>setenv TFTP_FILE mo162.boot
>exit
```

Reset the board. You should see something resembling the following:

```
Attempting boot via shell
Spectra Boot version 4.1
Copyright (c) 1992-1996 Microtec
Warm reset
Type any key within 2 seconds to get shell prompt
Timeout, exiting shell
Attempting boot via tftp
tftp client IP address: 149.147.5.33 95930521
tftp server IP address: 9593052c
Attempting TFTP load.
Server is 9593052c
TFTP...
TFTP file is mol62.boot
TFTP server is Hex 9593052c
Entry = 100000c Addr = 100000c
Entry at 100000c
RET FOR TFTP = 0 Code at 100000c
Attempting boot via shell
Spectra Boot version 4.1
Copyright (c) 1992-1996 Microtec
Cold reset
Type any key within 2 seconds to get shell prompt
Timeout, exiting shell
Attempting boot via rarp
ethernet address is 8:0:3e:22:43:22
IP address is 149.147.5.33
RARP server is 8:0:20:7:bd:20 192.9.200.135
Attempting boot via xtrace
```

Creating PROMs

To create PROMs that will boot directly from **tftpboot** by default, create a file **user.def** that contains the following:

```
@include (boot.def)
@include (mo162.def)
@include (microtec.def)

# for tftpboot:

boot.methods: shell, tftp
boot.env.boot_order: shell, tftp

# add enviroment variables:
boot.env.variables: verbosity,nodename,nameserver,
    tftp_device_ip,tftp_host_ip,tftp_file

# Change to own target ip address
boot.env.tftp_device_ip.value: 149.147.5.33

# Change to own host ip address
boot.env.tftp_host_ip.value: 149.147.5.44

# Change tftpboot file name
boot.env.tftp_file.value: mo162.hex

board.boot.code: 8000
board.boot.data: 1d000

board.memory.cold.address: 0x8000
board.memory.cold.size: 0x15000
board.memory.cold.type: BOOTOS_MEMORY_SHARED|
    BOOTOS_MEMORY_READONLY|
    BOOTOS_MEMORY_COPYBACK

board.memory.data.address: 0x1d000
board.memory.data.size: 0x40000
```

Use Xconfig to create the S-record **boot.hex** file for the PROMs:

```
xconfig user.def
```

Use the **s2s** command to specify the entry point of the starting code:

```
s2s -e 0x800c boot.hex mo162.hex
```

Ethernet Address Failure

If the RAM chip backup battery fails, the board will lose its Ethernet address and revert to the default of 08:00:3E:20:00:00 or 08:00:3E:2F:FF:FF. The correct Ethernet address should be 08:00:3E:2X:XX:XX, where X:XX:XX represents the last five digits of the Ethernet address.

To correct the failed Ethernet address:

1. Identify the Ethernet address failure.
2. Enter the boot shell.
3. Set the Ethernet address and exit the boot shell.
4. Reset the board.

Identifying the Ethernet Address Failure

The following sequence shows a board with the incorrect Ethernet address 08:00:3E:2F:FF:FF:

```
Attempting boot via shell
Spectra Boot version 4.1
Copyright (c) 1992-1996 Microtec
Warm reset
Type any key within 2 seconds to get shell prompt
Timeout, exiting shell
Attempting boot via rarp
Ethernet address is 8:0:3e:2f:ff:ff
```

Entering the Boot Shell

Reset the board and press a key at the console prompt to enter the shell:

```
Attempting boot via shell
Spectra Boot version 4.1
Copyright (c) 1992-1996 Microtec
Warm reset
Type any key within 2 seconds to get shell prompt
boot>
```

Setting the Ethernet Address and Exiting the Boot Shell

At the boot prompt, enter the correct Ethernet address for the board and exit:

```
boot> setenv ETHER_ADDR 08:00:3E:2X:XX:XX
boot> exit
```

For example:

```
boot> setenv ETHER_ADDR 08:00:3e:20:18:47
boot> exit
```

Wait for a message resembling the following:

```
Attempting boot via rarp
Ethernet address is 8:0:3e:20:18:47
```

After the RARP program runs, the new board address will be set.

Resetting the Board

Once the new Ethernet address is displayed, press the reset button on the board to reinitialize the Ethernet chip for the new address.

A message similar to the following is displayed:

```
Attempting boot via rarp
Ethernet address is 8:0:3e:20:18:47
Attempting boot via shell
Spectra Boot version 4.1
Copyright (c) 1992-1996 Microtec
Warm reset
Type any key within 2 seconds to get shell prompt
Timeout, exiting shell
Attempting boot via rarp
Ethernet address is 8:0:3e:20:18:47
IP address is 138.121.2.171
RARP server is 0:0:8e:6:3:43 138.121.2.248
Attempting boot via xtrace
```

If the battery-backed RAM is operational, this permanently sets the Ethernet address.

Using MVME162Bug to Set the Ethernet Address

You can also use the Motorola 162-Bug PROMs to correct the Ethernet address. Consult Motorola's MVME162 board documentation for more information. Look for the Ethernet address on a sticker attached to the P2 connector.

Cables

For the MVME162 and MVME162FX boards, configure a serial cable (3-wire communication) as follows:

<u>Host</u>		<u>Target</u>
xmit 2	<----->	2 rcv
rcv 3	<----->	3 xmit
gnd 7	<----->	7 gnd

Connect serial cables from either the front panel of the board or use a 712 transition module, but not both. Use Port B only with a serial interface module installed at connector J10 on the MVME162 board.

For the MVME162LX board, configure a cable as follows:

<u>Host</u>		<u>Target</u>
<u>RS232</u>		<u>RJ45 Jack</u>
xmit 2	<----->	4 xmit
rcv 3	<----->	5 rcv
gnd 7	<----->	3,6 gnd

Connect cables in the front panel of the board.

For more information about cabling, consult Motorola's MVME162, MVME162LX, and MVME162FX board documentation.

Configurations Tested

MVME162, MVME162LX, and MVME162FX boards may be ordered in several different configurations. This BSP was developed with the intention of using a single boot image (PROM) for any MVME162, MVME162LX, or MVME162FX board, so the BSP boot image should run on any version.

The following configurations have been tested:

- MVME162-12 (MC68LC040), 4MB DRAM, 512KB SRAM
- MVME162-223 (MC68040), 4MB DRAM, 512K SRAM
- MVME162-523 (MC68040), 8MB DRAM, 512K SRAM

Note

For boards using standard RAM, use the start-up code, **crt0.s**.

For boards using ECC RAM, use the start-up code **crt0.ecc.dram.s**.

Timers

A watchdog timer is not used. Both Timer1 and Timer2 from MCC are configured to generate interrupts every 10 milliseconds. Timer1 is used by Xtrace. Timer2 is unused. Interrupts from Timer2 are disabled at the MC chip. Timer3, Timer4, and two additional timers in the VMEchip2 are not programmed.

Flash/PROM Configuration

Using jumper J22, you can interchange Flash and PROM addresses. The default configuration file is written for Flash at 0xff800000 and EPROM at 0xffa00000. Even if you install the PROMs, when you power up control is transferred to MVME162Bug.

To transfer control to Xtrace, use command **go ffa0000c**. If you remove jumper J22 (GP13), the PROM memory map is selected. Change **board.boot.code** to ff800000. For more information about using jumper J22 and the memory map, consult Motorola's MVME162 board documentation.

VMEchip2 and Boot Code

The start-up code, **crt0.s**, performs minimal initialization of the VMEchip2 (if the chip is present). Depending on the application, VMEchip2 initialization should be customized. If DRAM is present, the DRAM base address is 0, and the SRAM base

address is 0xffe00000. If there is no DRAM, the SRAM base address is configured to 0. Consequently, you can use the same PROMs regardless of the memory configuration.

Boards Without Ethernet

The default bridge is Ethernet. If Ethernet is not present, use the serial bridge. This release does not support auto configuration, which would automatically reconfigure the bridge to **serial_2** if no Ethernet was found. If Ethernet is not present, create new PROMs using **serial_2** as the bridge. Remove **ether_1 device** from the devices list to avoid crashing the board when it attempts to initialize the nonexistent Ethernet facility.

DRAM Parity Checking

On the MVME162 board, use jumper J22 (GP17) to select DRAM parity checking. Installing the jumper disables DRAM parity. If you leave the jumper open, DRAM is cleared and parity checking is enabled when you power up the board.

Cabling

If a console connection is provided or the bridge is serial, use a serial cable to connect the target and the host. For details, see the section *Cables* in Chapter 2, *Configuration Information*.

Configuring Ethernet or Serial Interfaces

Chapter 2, *Configuration Information*, provides details of the serial and Ethernet interfaces.

The **Bridge** for this target is either:

- **logio_ether_xx_id** (detail; see the section *Ethernet Connection*)
- or
- **logio_serial_xx_id** (see the section *Serial Connection*)

Ethernet Connection

Assign the board an Ethernet address.

Some boards store the Ethernet address in a nonvolatile or battery backed-up RAM area. This address may require configuration. For instructions on how to configure the Ethernet address, see the section, *Supplementary Notes* in Chapter 2, *Configuration Information*.

Assigning the Board an IP Address

If the target board does not have an IP address (this will be the case for new boards), then you or your system administrator must assign one to the board. Consult the network and system administration documentation provided by the workstation vendor for information on this procedure.

Serial Connection

Use the **serial_server** program to communicate with the target using a serial packet interface.

To use the **serial_server** program, perform the following steps:

1. Update the file **/etc/remote**.
2. Update the file **\$SPECTRA/host/etc/connconf**.
3. Start the **serial_server** program.

Updating /etc/remote

See Chapter 2, *Configuration Information*, to determine the baud rate, parity, stop bits, and number of bits for the bridge device.

Generally, these values are:

- Baud: 19200 (9600 on slower boards)
- Parity: none
- Stop bits: 1
- Bits: 8

Edit the file **/etc/remote** to create an entry with the above communication parameters.

Example

In the following example, entries are created for **/dev/ttya** and **/dev/ttyb** (for baud rates of 4800, 9600, 19200, and 38400). The entry name is listed first; its parameters follow on a separate line. The entry name can be anything, but should be descriptive. For instance, the entry name for **/dev/ttya** at 4800 baud is **mo162a4800**.

```
mo162a4800:\
    :dv=/dev/ttya:br#4800:el=^C^S^Q^U^D:ie=%$:oe=^D:
mo162a9600:\
    :dv=/dev/ttya:br#9600:el=^C^S^Q^U^D:ie=%$:oe=^D:
mo162a19200:\
    :dv=/dev/ttya:br#19200:el=^C^S^Q^U^D:ie=%$:oe=^D:
mo162a38400:\
    :dv=/dev/ttya:br#38400:el=^C^S^Q^U^D:ie=%$:oe=^D:
mo162b4800:\
    :dv=/dev/ttyb:br#4800:el=^C^S^Q^U^D:ie=%$:oe=^D:
mo162b9600:\
    :dv=/dev/ttyb:br#9600:el=^C^S^Q^U^D:ie=%$:oe=^D:
mo162b19200:\
    :dv=/dev/ttyb:br#19200:el=^C^S^Q^U^D:ie=%$:oe=^D:
mo162b38400:\
    :dv=/dev/ttyb:br#38400:el=^C^S^Q^U^D:ie=%$:oe=^D:
```

The parameters are named to reflect the real **tty** channel on the workstation.

Updating \$SPECTRA/host/etc/connconf

For each **/etc/remote** entry, create a logical name to be used by XSH as follows:

target_name entry_name host_name port_number baud_rate

<i>target_name</i>	The name you will use when executing serial_server on the host. serial_server is executed on the host for a serial packet-based bridge to the host machine from the target.
<i>entry_name</i>	The name of the specific entry mapped to the <i>target_name</i> in the /etc/remote file. This file sets the parameters for the connection.
<i>host_name</i>	The workstation with a physical serial connection to the target.

Example

The following example shows the notation used by Microtec for a workstation called **sun29**:

mo162a48	mo162a4800	sun29	2000	4800
mo162b48	mo162b4800	sun29	2001	4800
mo162a96	mo162a9600	sun29	2002	9600
mo162b96	mo162b9600	sun29	2003	9600
mo162a19	mo162a19200	sun29	2004	19200
mo162b19	mo162b19200	sun29	2005	19200
mo162a38	mo162a38400	sun29	2006	38400
mo162b38	mo162b38400	sun29	2007	38400

In the first line of the above example, `mo162a48`, the target name for starting **serial_server**, is mapped to the serial parameter `mo162a4800`, as defined by the **/etc/remote** *entry_name* on the host machine `sun29`. The Xtrace Protocol will use UDP port number 2000 to communicate to the target `mo360aa48`, and a baud rate of 4800 will be used to transmit the serial packets to and from the target.

Note

All board names in the **connconf** file must be unique. Do not use the same board name under NIS and in the **connconf** file.

Starting serial_server

Connect a serial cable from the workstation to the target board. Start **serial_server** for the corresponding host port and baud rate.

For example, if the host port on workstation **sun29** is **/dev/ttyb**, and if the serial bridge ID is configured for a baud rate of 9600, invoking:

```
serial_server mo162b96 &
```

lets the `serial_server` program communicate with the target **mo162b96** (assuming the **connconf** and **/etc/remote** files contain entries matching the previous examples).

Connecting to the Target With XSH

The following examples assume a board with an Ethernet name of **foo.eng.mri.com** or **serial_server** name of **mo162b19**.

Ethernet

```
xsh
Spectra Cross-Development Shell; XSH 4.6B
Copyright (C) 1991-1996 Microtec

>connect foo.eng.mri.com
foo.eng.mri.com connected (non-os mode)
foo.eng.mri.com>
Cold reset on target foo.eng.mri.com
FF80AED4 2F02          MOVE.L      D2, -(SP)
foo.eng.mri.com>
```

Serial

```
xsh -t mo162b19
Spectra Cross-Development Shell; XSH 4.6B
Copyright (C) 1991-1996 Microtec

mo162b19 connected (non-os mode)
mo162b19>
Cold reset on target mo162b19
0005A720 9421FFc0stwu          1,0xffffffffc0(1)
mo162b19>
```

Special Notes for Serial Ports

Configure a serial port for either **serial_packet_device** or **serial_tty_device** by manually changing the configuration in the **devcnfg.c** file. There is no Xconfig option for the BSP. The default for the BSP is listed in subsequent sections.

A serial port configured as **serial_packet_device** can only be used as a bridge. **vconsole** output can also be directed to this port if a hardware timer provides a tick. A **serial_packet_device** does not accept **tty** output directly since that output is not in packet form.

A serial port configured as **serial_tty_device** can only be used for **tty** input/output. This port cannot operate as an Xtrace bridge.

For Microtec BSPs in general:

```
Serial Port #1: tty,      9600
Serial Port #2: packet  19200
Serial Port #3: tty,      9600
.
.
.
Serial Port #n: tty,      9600
```

System Requirements

This section lists hardware and software requirements for the MVME162/LX/FX board.

Hardware

Table P-1 lists hardware requirements for the MVME162/LX/FX BSP.

Table P-1. Hardware Requirements

Item	Description
Host	Sun-4 workstation running SunOS version 4.1.3 (or later versions) or Solaris 2.4 (or later versions) in SunOS binary compatibility mode
	HP 700 workstation running HP-UX 9.0 (or later versions)
	PC-compatible system running Windows NT 4.0 (or later versions)
Target	Motorola MVME162, MVME162LX, or MVME162FX boards (power must be supplied on VMEbus P2 for the MVME162 and MVME162FX)

Software

Before you install this BSP, you must install your Spectra cross-development environment software and the Microtec compiler toolkit specified in the *Release Notes*.

Vital Statistics

This section lists board specifications, on-board devices, supported Microtec components, and minimum target RAM size for the MVME162/LX/FX board.

Board Specifications

Table P-2 lists board specifications for the MVME162/LX/FX board.

Table P-2. MVME162/LX/FX Board Specifications

Board Item	Description
Board name	Motorola MVME162
CPU type	MC68040 or MC68LC040
Clock frequency	Variable, 25MHz or 33MHz
Floating-point unit	Included in CPU in boards using MC68040
Memory Configuration	Variable, minimum 1MB
RAM	32-bit DRAM with parity
EPROM	4 Mbit (512 Kbit x 8 EPROM)
NVRAM	8 KB RAM/clock with battery backup
FLASH	1MB
SRAM	Variable 512KB/1MB/2MB

On-Board Devices

Table P-3 lists the on-board devices found on the MVME162/LX/FX board.

Table P-3. MVME162/LX/FX On-Board Devices

Devices	Description
Timers	Six 32-bit tick timers, watchdog timer
Serial I/O	Two to four channels on Z85230 serial communication controller
Ethernet	Intel i82596
SCSI	NCR 53C710 SCSI I/O controller
MMU	Available
VME	VMEchip2

Supported Microtec Components

This BSP supports the following components:

- IFX (I/O and File Executive)
- Remote procedure calls
- RTL (Run-Time Library)
- SNX (STREAMS and TCP/IP Networking Executive)
- Spectra Backplane
- VRTXsa Run-Time Kernel
- VRTX32 Run-Time Kernel
- XRAY Pro debug suite
- XRAY debugger
- Xpert Profiler

Target RAM Size — Minimum Requirement

40 KB (with tuning, Xtrace only)

Notational Conventions

This guide uses the notational conventions shown in Table P-4 (unless otherwise noted).

Table P-4. Notational Conventions

Symbol	Name	Usage
{ }	Curly Braces	Enclose a list from which you must choose an item.
[]	Square Brackets	Enclose optional items.
...	Ellipsis	Indicates that you may repeat the preceding item zero or more times.
	Vertical Bar	Separates alternative items in a list.
	Punctuation	Punctuation such as commas (,) and colons (:) must be entered as shown.
	Typewriter Font	Represents code or user input in interactive examples.
	<i>Italics</i>	Represents a descriptive item that should be replaced with an actual item.
	Bold	Represents elements that need to stand out from the main body of text.

Related Publications

Refer to the following publications for further information about Microtec products:

- *Getting Started (UNIX Hosts).*
- *Getting Started (Windows Hosts).*
- *Spectra Backplane Concepts.*
- *Board Support Package (BSP) Developer's Guide and Reference.*
- *Debug Shell (XSH) User's Guide and Reference.*
- *Configuration Tool (Xconfig) User's Guide and Reference.*
- *I/O and File Executive (IFX) Programmer's Guide and Reference.*

-
- *STREAMS and TCP/IP Networking Executive (SNX) and SNMP Programmer's Guide and Reference.*
 - *Run-Time Library (RTL) Programmer's Guide and Reference.*

Questions and Suggestions

Microtec is committed to providing its customers with quality software development and RTOS tools and support services. Our commitment continues beyond your purchase of the product throughout your development life cycle.

If you have questions or suggestions regarding this product, please contact your Microtec support representative. Contact numbers are listed on the back cover of this document.

J

Jumper settings

- MVME162 2-8
- MVME162FX 2-13
- MVME162LX 2-10

L

- logio_ether_1_id 1-2, 2-3
- logio_serial_1_id 1-2, 2-3

M

- Memory map 2-1
- mo162.def file 2-1, 2-3
- mo162ram.def file 2-17
- MVME162
 - board configuration 2-7
 - board layout 2-9
 - jumper settings 2-8
 - PROMs 2-7
- MVME162Bug 2-17, 2-24
- MVME162FX
 - board configuration 2-13
 - board layout 2-15
 - jumper settings 2-13
 - PROMs 2-13
- MVME162LX
 - board configuration 2-10
 - board layout 2-12
 - jumper settings 2-10
 - PROMs 2-10

N

- Notational conventions ix

O

- On-board devices viii

P

- PROM specifications
 - MVME162 2-7

- MVME162FX 2-13
- MVME162LX 2-10

Q

- Questions x

S

- Serial connection 1-3, 1-6
- Serial ports, configuring
 - serial_packet_device 1-7
 - serial_tty_device 1-7
- serial_server 1-3, 1-5
- Software configuration
 - bridge 2-3
 - console 2-3
 - default file 2-3
 - device driver parameters 2-3
 - Ethernet 2-6
 - memory map 2-1
 - serial 2-4
 - VME 2-6
- Software requirements vi
- Specifications, board vii
- Suggestions x
- System requirements
 - hardware vi
 - software vi

T

- Target connection 1-6
- Timer 2-26
- tip 2-18

V

- vconsole 1-7
- VMEchip2 2-26

X

- Xconfig
 - creating boot PROMs 1-1
 - creating boot.hex.tmp 2-17

Index

variables 2-7
XSH
target connection 1-6





Serial	2-4
Ethernet	2-6
VME	2-6
Xconfig Variables	2-7
Hardware Setup	2-7
MVME162 Board Configuration	2-7
PROMs	2-7
Jumper Settings	2-8
Board Layout	2-9
MVME162LX Board Configuration	2-10
PROMs	2-10
Jumper Settings	2-10
Board Layout	2-12
MVME162FX Board Configuration	2-13
PROMs	2-13
Jumper Settings	2-13
Board Layout	2-15
Supplementary Notes	2-16
Booting Directly From Spectra PROMs	2-16
MVME162Bug ROMBoot and Spectra Bridge	2-16
Downloading Bridge Into RAM Using MVME162Bug	2-17
Serial Download	2-17
Booting Directly From tftpboot	2-19
Using the Boot Shell Method Without Modifying PROMs	2-19
Creating PROMs	2-22
Ethernet Address Failure	2-23
Identifying the Ethernet Address Failure	2-23
Entering the Boot Shell	2-23
Setting the Ethernet Address and Exiting the Boot Shell	2-23
Resetting the Board	2-24
Using MVME162Bug to Set the Ethernet Address	2-24
Cables	2-25
Configurations Tested	2-26
Timers	2-26
Flash/PROM Configuration	2-26
VMEchip2 and Boot Code	2-26
Boards Without Ethernet	2-27
DRAM Parity Checking	2-27
Index	Index-1

Figures

Figure 2-1.	Memory Map	2-2
Figure 2-2.	MVME162 Board Configuration	2-9
Figure 2-3.	MVME162LX Board Configuration	2-12
Figure 2-4.	MVME162FX Board Configuration	2-15

Tables

Table P-1.	Hardware Requirements	vi
Table P-2.	MVME162/LX/FX Board Specifications	vii
Table P-3.	MVME162/LX/FX On-Board Devices	viii
Table P-4.	Notational Conventions	ix
Table 2-1.	Timer 1 Device Driver Configuration Parameters	2-3
Table 2-2.	Timer 2 Device Driver Configuration Parameters	2-4
Table 2-3.	Serial 1 Device Driver Configuration Parameters	2-4
Table 2-4.	Serial 2 Device Driver Configuration Parameters	2-5
Table 2-5.	Ethernet Device Driver Configuration Parameters	2-6
Table 2-6.	VME Device Driver Configuration Parameters	2-6
Table 2-7.	Xconfig Variables	2-7
Table 2-8.	MVME162 PROM Parameters	2-7
Table 2-9.	MVME162 Factory Default Jumper Settings	2-8
Table 2-10.	MVME162LX PROM Parameters	2-10
Table 2-11.	MVME162LX Factory Default Jumper Settings	2-10
Table 2-12.	MVME162FX PROM Parameters	2-13
Table 2-13.	MVME162FX Factory Default Jumper Settings	2-13

TRADEMARKS

ARTX/ADA Realtime Executive[®], Microtec[®], the Microtec logo, Nanokernel[®], RTscope[®], RTsource[®], Spectra[®], VRTX[®], VRTX32[®], VRTXvelocity[®], XRAY[®], Xtrace[®], and Xtrace Protocol[®] are registered trademarks of Microtec.

BSPBuilder[™], FastStart[™], IFX[™], KernelBuilder[™], KernelIntegrator[™], *logio*[™], SNX[™], Source Explorer[™], the Spectra logo, Target Manager[™], TNX[™], ToolBuilder[™], Virtual Target[™], VRTXmc[™], VRTX/OS[™], VRTXsa[™], Xconfig[™], Xpert[™], Xpert Profiler[™], XRAY In-Circuit Debugger[™], XRAY In-Circuit Debugger Monitor[™], and XSH[™] are trademarks of Microtec.

Other product names mentioned are trademarks or registered trademarks of their respective companies.

RESTRICTED RIGHTS LEGEND

U.S. Government Restricted Rights. This product and related documentation have been developed entirely at private expense and are commercial computer software provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. Government or a U.S. Government subcontractor is subject to the restrictions set forth in the license agreement provided with the product pursuant to DFARS 227.7202-3(a) or as set forth in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19, as applicable.

Microtec
880 Ridder Park Dr.
San Jose, CA 95131

Copyright © 1987-1997 Microtec. All rights reserved. No part of this publication may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical or otherwise, without prior written permission of Microtec.

00

Revision History

REV.	REVISION HISTORY	DATE	APPD.
-001	Updated for Spectra 3.0.5	3/93	C.S.
-002	Updated for Spectra 3.0.6	5/93	
-003	Updated for Spectra 3.0.7	9/93	
-004	Updated for Spectra 3.1.2	4/94	
-005	Updated for Spectra 3.1B (HP 700)	10/94	P.J.
-006	Updated for Spectra 3.C (Sun-4 SunOS/Solaris)	12/94	P.J.
-007	Updated for Spectra 3.CAB (Sun-4 SunOS/Solaris)	2/95	P.J.
-008	Removed vinstall references	1/96	M.G.
-009	Updated for Spectra 4.0 (added support for MVME162LX and MVME162FX)	12/96	M.G.
-010	Updated for Windows NT	6/97	M.G.

Revision History
