MVME166BUG/L2
July 1993

Programming MVME166 FLASH

This customer letter addresses the programming of the resident FLASH memory devices. It is possible that you may accidentally erase the FLASH memory which contains the MVME166BUG. This customer letter is a not an issue if you do not need to retain the MVME166BUG in FLASH memory.

The FLASH memory on the MVME166 encompasses the address range of $FF800000 through $FF8FFFFF, totaling 1 megabyte in size. The MVME166BUG is 1/2 megabyte (524,288 bytes) in size and resides in the address range of $FF800000 through $FF87FFFF. The remaining FLASH memory of address range $FF880000 through $FF8FFFFF is open to your application. This range also totals 1/2 megabyte in size.

Please read all procedures before any FLASH memory programming is performed.

PROCEDURE #1

The following procedure should be followed to program the FLASH memory with your application. This procedure would be used if the MVME166BUG is intact (resident) within the FLASH memory:

Procedure Steps:

1. Copy the MVME166BUG from the FLASH memory to local memory. For example:

   ```
   166-Bug>BM FF800000 FF87FFFF 100000;l
   Effective address: FF800000
   Effective address: FF87FFFF
   Effective address: 00100000
   166-Bug>
   ```

   This copies the MVME166BUG to local memory, address range of $00100000 through $0017FFFF.

2. Load your application into local memory, starting at address $00180000.

3. Now program the FLASH memory with both the MVME166BUG and your application.

```
166-Bug>PFLASH 100000:100000 FF800000
Source Starting/Ending Addresses      =00100000/001FFFFF
Destination Starting/Ending Addresses =FF800000/FF8FFFFF
Number of Effective Bytes             =00100000 (&1048576)

Program FLASH Memory (Y/N)? y
```

At this point, the system hangs up, and eventually the FAIL LED flashes, indicating that the programming is complete.

Note **FLASH memory (total of 1 megabyte) programming may take as long as 15 minutes to complete.**

PROCEDURE #2

The following procedure should be followed to program the FLASH memory from the MVME166 download PROM. The MVME166 download PROM is a subset of the MVME166BUG product and is also known as the BOOTBUG. The BOOTBUG contains enough functionality to enable the downloading of object code (by means of the VMEbus, serial port, SCSI bus, or the network) and reprogramming of the FLASH memory.

There is a jumper on the MVME166 board that controls the operation of the BOOTBUG. If the jumper at J3 pins 7 and 8 is in place (GPI3),then the BOOTBUG (which always executes at reset/power-up) unconditionally jumps to the debugger product contained in the FLASH memory. If this jumper is removed, execution continues in the (diminished functionality) BOOTBUG.

Before using some of the features of the MVME166 BOOTBUG, some parameters may need to be defined. For example, the SCSI ID, the Ethernet address, the clock speed of the board, or possibly the mapping of the VMEbus. You need to run the **SETUP** command to define these parameters before using any SCSI, VME, or Ethernet I/O.

This procedure would also be used if the MVME166BUG is not resident within the FLASH memory.

Procedure Steps:

1. Power down the board and remove from the chassis.
2. Remove the jumper at J3 pins 7 and 8.
3. Install the board and apply power.

4. Using the MVME166 BOOTBUG, download (e.g., **IOP**, **LO**, **NIOP**) the MVME166BUG product image into local memory.

   For example:

```
166-BBug>SETUP
MPU Clock Speed       = "????"? 3300
Ethernet Address      = 000000000000? 08003E2194D4 (from P2
                                                     label)
Local SCSI Identifier = "??"? 07
VME Slave Enable #1 [Y/N]                = N?
VME Slave Starting Address               = 00000000?
VME Slave Ending Address                 = 0000FFFF?
VME Slave Address Translation Address    = 00000000?
VME Slave Address Translation Select     = 00000000?
VME Slave Control                        = 0000?
VME Master Enable [Y/N]                  = Y?
VME Master Starting Address              = 40000000?
VME Master Ending Address                = 4FFFFFFF?
VME Master Address Translation Address   = 00000000?
VME Master Address Translation Select    = 00000000?
VME Master Control                       = 0D?
166-BBug>
```

   (Load streaming tape)

```
166-Bug>IOP
Controller LUN   =00? 0
Device LUN       =00? 40
Read/Write/Format=R? R
Memory Address   =0000E000? FFE00
File Number      =00000000? 0
Number of Blocks =0002? 802 (802 hexadecimal x 256 decimal
                            block size)
Flag Byte        =00? 1
Address Modifier =00? 0
166-Bug>
```

   The resultant local memory contents:

$000FFE00 to $000FFFFF	FLASH file header, 512 bytes in size
$00100000 to $0017FFFF	FLASH file object, 524,288 bytes in size

Once the **SETUP** command has been executed, you could also download from the network, for example:

```
166-BBug>NIOT
Controller LUN =00?
Device LUN     =00?
Node Control Memory Address =FFE10000?
Client IP Address      =0.0.0.0? 144.191.24.144
Server IP Address      =0.0.0.0? 144.191.24.252
Subnet IP Address Mask =255.255.255.0?
Broadcast IP Address   =255.255.255.255?
Gateway IP Address     =0.0.0.0?
BOOTP/RARP Request Retry       =00?
TFTP/ARP Request Retry         =00?
166-BBug>
```

```
166-BBug>NIOP
Controller LUN =00?
Device LUN     =00?
Get/Put        =G?
File Name     =? /riscy/fwdb/BUGDB/GET/obj/166b/98-w4104c01b
Memory Address =0000E000? FFE00
Length         =00000000?
Byte Offset    =00000000?

Bytes Received =&524800, Bytes Loaded =&524800
Bytes/Second   =&52480, Elapsed Time =10 Second(s)
166-BBug>
```

5. Now program the FLASH memory with the MVME166BUG product image, for example:

```
166-BBug>PFLASH 100000:80000 FF800000
Source Starting/Ending Addresses      =00100000/0017FFFF
Destination Starting/Ending Addresses =FF800000/FF87FFFF
Number of Effective Bytes             =00080000 (&524288)
```

```
Program FLASH Memory (Y/N)? y
Zeroing Block Number     =00 ($FF800000)
Zeroing Block Number     =01 ($FF800000)
Zeroing Block Number     =02 ($FF800000)
Zeroing Block Number     =03 ($FF800000)
Erasing Block Number     =00 ($FF800000)
Erasing Block Number     =01 ($FF800000)
Erasing Block Number     =02 ($FF800000)
Erasing Block Number     =03 ($FF800000)
Programming Block Number =00 ($FF800000)
Programming Block Number =01 ($FF800000)
Programming Block Number =02 ($FF800000)
Programming Block Number =03 ($FF800000)
FLASH Memory Programming Complete
166-BBug>
```

Note **FLASH memory (total of 1 megabyte) programming may take as long as 15 minutes to complete.**

6. Power down the board and remove from the chassis.
7. Install the jumper at J3 pins 7 and 8.
8. Install the board and apply power.

PROCEDURE #3

The following procedure would be used to update the MVME166 board with future revisions (e.g., upgrades, enhancements) of the MVME166BUG. This procedure also assumes that a user application is also resident within the FLASH memory.

Procedure Steps:

1. Copy the current contents of the FLASH memory to local memory. For example:

```
166-Bug>BM FF800000 FF8FFFFF 100000;l
Effective address: FF800000
Effective address: FF8FFFFF
Effective address: 00100000
166-Bug>
```

   This copies the MVME166BUG and any possible user application to local memory, address range of $00100000 to $001FFFFF.

2. Download (e.g., **IOP**, **LO**, **NIOP**) the updated version of the MVME166BUG into local memory, starting at address $000FFE00. The distribution medium contains a single file, exactly 524,800 (512 header + 524,288 object) bytes in size.

   For example:

   (Load streaming tape)

   ```
   166-Bug>IOP
   Controller LUN   =00? 0
   Device LUN       =00? 40
   Read/Write/Format=R? R
   Memory Address   =0000E000? FFE00
   File Number      =00000000? 0
   Number of Blocks =0002? 802 (802 hexadecimal x 256 decimal
                               block size)
   Flag Byte        =00? 1
   Address Modifier =00? 0
   166-Bug>
   ```

   The resultant local memory contents:

$000FFE00 to $000FFFFF	FLASH file header, 512 bytes in size
$00100000 to $0017FFFF	FLASH file object, 524,288 bytes in size
$00180000 to $001FFFFF	User application, 524,288 bytes in size

   Dependent on the board configuration (see **ENV** and **CNFG** commands), you could also download from the network, for example:

   ```
   166-Bug>NIOT
   Controller LUN =00?
   Device LUN     =00?
   Node Control Memory Address =FFE10000?
   Client IP Address      =0.0.0.0? 144.191.24.144
   Server IP Address      =0.0.0.0? 144.191.24.252
   Subnet IP Address Mask =255.255.255.0?
   Broadcast IP Address   =255.255.255.255?
   Gateway IP Address     =0.0.0.0?
   BOOTP/RARP Request Retry       =00?
   TFTP/ARP Request Retry         =00?
   166-Bug>
   ```

```
166-Bug>NIOP
Controller LUN =00?
Device LUN     =00?
Get/Put        =G?
File Name     =? /riscy/fwdb/BUGDB/GET/obj/166b/98-w4104c01b

Memory Address =0000E000? FFE00
Length         =00000000?
Byte Offset    =00000000?

Bytes Received =&524800, Bytes Loaded =&524800
Bytes/Second   =&52480, Elapsed Time =10 Second(s)
166-Bug>
```

3. Now program the FLASH memory with both the MVME166BUG and also possibly a user application.

```
166-Bug>PFLASH 100000:100000 FF800000;r
Source Starting/Ending Addresses      =00100000/001FFFFF
Destination Starting/Ending Addresses =FF800000/FF8FFFFF
Number of Effective Bytes             =00100000 (&1048576)

Program FLASH Memory (Y/N)? y
```

Control does not return here. The FLASH memory program operation was instructed to reset (**;r** command line option) upon FLASH memory programming. Upon completion, following the board reset, the updated MVME166BUG signs on.

Note

FLASH memory (total of 1 megabyte) programming may take as long as 15 minutes to complete.