MNEMONIC DEBUGGER (PROM)

Commands:

```
+ = not in DNx60
- = DNx60 only
* = DN3000 only
# = DNx60 and DN3000 only

+ A <location>                          Access location
* AR                                    Access Control Register
- AS                                    Display current ASID
  B <location>                          Breakpoint
+ C <start> <end> <target>              Copy Memory
+ CA <start>                            CALL Subroutine
# CB <location>                         Clear breakpoints
+ D <start> <end> <items/line>          Dump Memory
+ DI <type><unit> <log vol>             Define Disk
+ DL                                    Down-line Loader
- DR                                    Dump registers
* DU                                    Dump system
  EX <filename>                         Load and Execute File
  EY <filename>                         Load and Execute File
                                         with trap after load
+ F <start> <end> <word>                Fill Memory
* FO                                    Force load
+ G <location>                          Jump to Location
# H                                     Help
# IC                                    Enable/disable/show
                                        Instruction cache
+ LD                                    Lists SAUn Directories
+ LO <filename>                         Load File
+ M                                     Map Address Space
+ P                                     Unmap Address Space
# PV                                    PA-to-VA
+ RE                                    Reset System
- RR                                    Access region registers
+ S <start> <end> <value> <mask>        Search Memory
+ SH <0-3>                              Spindown Winchester
  SK                                    Select keyboard
# SS                                    Single step
+ TE                                    Run boot PROM diagnostics
+ V <start> <end> <target>              Verify Memory
# VP                                    VA-to-PA
# XD                                    XON/XOFF disable
# XE                                    XON/XOFF enable
```

Commands valid in DNx60 CPU only:

```
DC          Enable/disable/show data cache
GB          Go Back to CPIO environment, halt CPU
FP          Access Floating Point Registers
```

Commands valid in DNx60 CPIO only:
Those marked with an asterisk require the micro exec to be loaded and/or running.

```
 UC          Clock step CPU
*US          Micro step  <cnt <uaddr>>
*UT          Micro trace <cnt <uaddr>>
 UH          Halt the CPU and CPIO becomes master
 UR          Run micro code <from uaddr>
 UX          Reset micro machine
*UU          Micro trap to the micro executive
*UI          IPL micro machine at uaddr 0
 UP          Display or set micro pc
 UA          Load microcode file set
 UL          Load one microcode file by name
 UF          Set or clear microcode loaded flag
*MG          Start the CPU <at macro addr> and CPIO
               becomes slave
*MR          Start the CPU <at macro addr> and CPIO
               dies
*MS          Macro step the CPU <cnt <addr>>
*MT          Macro trace the CPU <cnt <addr>>
 MX          Load and execute program using CPIO
 GF          Go Forward to CPU with reset
               exception, halt CPIO
 FR          Fill wcs with freeze micro instructions
*LP          Set loop mode for uexec commands
 MM          Set CPIO in master mode
 SM          Set CPIO in slave mode
 DM          Set CPIO in dead mode
 MO          Display CPIO master/slave mode
 DS          Dump CPU state
```

```
A [<size_spec>] <location>[<base_spec>]

Accesses <location> and prints address and contents
according to <size_spec> and <base_spec>.

AR <ctl_reg>

Access certain system control registers.

   ctl_reg:
      TC   = MMU translation control
      RP   = MMU root pointer
      DFC  = CPU Destination function code
      SFC  = CPU Source function code
      CACR = CPU Cache control
      CAAR = CPU Cache address
```

```
B [<location>]
```

Sets/clears the breakpoint at the location specified. Breakpoint is not inserted until G command. Previous instruction is reinstalled on breakpoint entry or vector entry. For older (reverse-mapped) systems, only one breakpoint may be set at a time, and it must be cleared or moved before continuing after breaking.

```
C <start> <end> <target>
```

Copies memory defined by the bounds <start> to <end> onto memory starting at <target> through <target>+<end>-<start>.

```
CA <start>
```

Calls the subroutine which starts at <start>. All registers saved from the last entry except A0 are restored immediately prior to the call.

```
CB [<location>]
```

Clears the breakpoint at <location>, or all the breakpoints set, it you omit the argument. An error message appears if you attempt to clear breakpoints that are not set.

```
D [<size_spec>] <start> <end>

<items_per_line>[<base_spec>]
```

Dumps memory defined by the bounds <start> to <end> onto the terminal printing address followed by specified <items_per_line>. The default is one per line. The <size_spec> controls the item-size to be dumped: byte, word, long, instruction.

```
DI <W>|<F>|<N> [<nn>]|<S> <0-3> <1-10>
```

Disk defines the boot device: Winchester, Floppy, Node (nn), Storage Module unit 0-3, and logical volume 1-10. Defaults are: W, 0, 1.

```
DL
```

Transfers control to the down-line loader.

```
DS
```

Dump the CPU state. Valid only from the CPIO environment. It is useful if micromachine freezes (UPCxxx).

DU

This command initiates the system dump procedure. Requires operating system to have been booted (to provide dump address and routine). Precede with the device specification (DI N <nodeid>, .DI F or DI C), default is DI F. Will dump to multiple floppies.

EX <filename>

Execute restores the named file from the "SAU" directory of the boot device and transfers control to it. After the restore is complete, the LOW, HIGH, and START addresses are displayed.

EY <filename>

Just like EX except:

- Just before passing control to the program, the MD will trap, and you can patch the program. Type "G", "G *+2" to continue.
- If you are executing AEGIS, AEGIS will trap again after establishing the OS mapping and before calling OS_$INIT. AEGIS can then be patched or examined using the virtual addresses AEGIS.MAP in the appropriate SAU directory. Type "G", "G *+2" to continue bringing up AEGIS.

NOTE: Older boot PROMs may not support the EY command.

F <start> <end> [<word>]

Fills memory defined by bounds <start> to <end> with a word value <word>.

FO

This command initiates a force load sequence.

G [<location>]

Jumps to <location> after inserting breakpoint (if any), restoring all registers and SR.

GB

Go back (return control) to the boot processor. This is valid only in the main processor environment.

GF

GF start the main processor with a reset exception. You must have loaded microcode before you execute GF.

H

Calls HELP files.

IC <on> <off>

Enables or disables the instruction cache. LD

List directory displays the contents of the "SAU" directory of the boot device.

LO <filename>

Load restores the named file from the "SAU" directory of the boot device. The LOW, HIGH, and START addresses are displayed.

LP

LP sets the loop bit on any microexecutive command issued until the machine is reset. With the loop bit set, the commands operate in loop mode.

M

Maps address space and enables mmu. Memory is rearranged as shown under Address Space in Chapters 7 thorough 12.

MG [<adr>]

MG starts the main processor at the address you specify, or at the current main processor program counter (PC), if you do not give an argument. The boot processor goes into slave mode.

MR [<adr>]

Starts the main processor at the address you specify, or at the current main processor program counter (PC), if you do not give an argument. The boot processor halts.

MS [<Count> [<adr>]]

Macro steps once or for the number of counts, showing the program counter at the end of the steps. A macrostep executes one instruction from memory, on the main processor.

MT [<Count> [<adr>]]

Macro traces for the number of counts you specify, or until you hit a key. Prints the program counter at each step.

MX <filename>

Loads and executes the program <filename> on the boot processor. <filename> is a program in the SAU4 directory on the volume defined as the disk device by the DI command.

P

Turns off the mapping. MMU is assumed at FFB400.

PV <PA> <addr>

Convert physical address to all virtual addresses by searching the page tables starting at current region registers or use region registers at ADDR. PA must be long-word aligned.

RE

Reset executes the RESET instruction. If entered while running on CPU B, a second RESET instruction is executed for CPU A. The debugger will initialize and wait for terminal input. This command also enables the POWER-OFF key.

RR <region register number (0-31, decimal, 0-1F Hex>

Accesses the region register you specify in the argument, and displays its contents.

S [<size_spec>] <start> <end> <value> [<mask>][<base_spec>]

Searches memory defined by bounds <start> to <end> for <value> through optional <mask>. If <mask> is not specified it defaults to $FFFFFFFF. The <size_spec> controls the item-size to be searched: byte, word or long.

SH <0-3>

Shuts down the Winchester unit and acknowledges outstanding interrupts. This command also enables the POWER-OFF key.

SK

Forces the MD to enter its keyboard polling loop without requiring a REset.

TE

Forces execution of boot PROM diagnostics.

UA

Loads the microcode loading program (ULOAD) and the entire set of microcode files needed to boot the operating system.

UC [<count>]

Executes a clock cycle, or the number of cycles you specify in the <count> field, on the main processor. The boot processor enters slave mode.

UF [-OFF]

UF sets or clears the "microcode loaded" flag in the PROM. When the flag is set, the PROM does not load microcode when enabling the main processor. UH

Halts (disables) the main processor and puts the boot processor into master mode. The boot processor now has bus mastership and control of the system.

UI

Resets the main processor and runs it from microlocation zero. This command starts the micro executive (if it is loaded).

UL <filename> -N

Reads one microcode data file from the SAU4 directory and the micro-loading program (ULOAD) from the boot device, and loads the microcode. By default, if the microcode file is a WCS file, UL fills the WCS with freezes. Use the -N option when you are loading two WCS files separately.

UP <uadr>

Displays or sets the microcode program counter. If the main processor is running, you cannot read or set the microcode program counter.

UR [<uadr>]

Runs the microcode on the main processor, starting at the micro address you specify. The boot processor enters slave mode. If you do not specify an address, UR uses the current microcode program counter.

US [<Count> [<uadr>]]

Micro steps once or for the number of counts you give, showing the micro program counter at the end of the steps. If you specify <uadr>, the program starts at the micro address you specify.

UT [<Count> [<uadr>]]

Micro traces for the number of counts you specify or until you hit a key. The micro program counter is displayed at each step. If you specify a micro address, the trace starts at the micro address you specify.

UU

Causes a micro trap in the main processor, and displays the microcode program counter. You can use this command to stop the main processor.

UX

Resets the main processor and sets the micro program counter to 0.

V [<size_spec>] <start> <end> <target>[<base_spec>]

Verifies equality of two memory areas defined by <start> to <end> and <target> to <target>+<end>-<start>. If a discrepancy is found, the address in the first area and the contents of each are printed in the appropriate format.

VP <VA> <addr>

Convert virtual address to physical address using current region registers at ADDR. VA must be long-word aligned.

XD

Disables the X-On protocol used for communication with dumb terminals.

XE

Enables the X-On protocol used for communication with dumb terminals.

Command Formats:

```
<command>[<size_spec>] [<parameter_list>][<base_spec>]

<command> ::= A|B|C|D|DL|F|G|S|V|<empty>

<size_spec> ::= :I|:B|:W|:L

<parameter_list> ::= <parameter> ... [up to four]

<parameter> ::= <num_exp>|Dn|An|Rn|CCR|SR|
                {An}|<num_exp>(An)|<num_exp>{<index_spec>}|
                <num_exp>(An,<index_spec>)

<num_exp> ::= <num>|*|<num_exp>+<num>|<num_exp>-<num>|
              <num>_exp>x<num>

<num> ::= <simple_number|$<simple_number>|
          <base>$<simple_number>|-<num>|<quoted_string>

<base> ::= <simple_number>

<quoted_string> ::= '<letter> ... <letter>' [up to four]

<index_spec> ::= An.W|Dn.W|An.L|Dn.L

<base_spec> ::= :O|:D|:H|:A
```

To reference an address stored in a relocation register, prefix the register name with a zero.

Semantics:

:I ::= instr-sized items, output in mnemonic format.

:B ::= byte-sized items, output in numeric format.

:W ::= word-sized items, output in numeric format.

:L ::= longword-sized items, output in numeric format.

Parameters are evaluated to a memory location or to an MD saved register, [e.g. Dn, An] or to a location computed from a saved register [num(An)]. Up to four parameters may be required. Unspecified parameters are set to zero.

:O ::= numbers and immediate constants printed in octal.

:D ::= numbers and immediate constants printed in dec.

:H ::= numbers and immediate constants printed in hex.

:A ::= numbers and immediate constants printed in ASCII.

All numeric input defaults to hexadecimal. $num implies hexadecimal. <base>$num implies base is <base> [8$777 is

octal, 2$1001 is binary]. <base_spec> and <size_spec> may be specified anywhere in the command line as well as anywhere in A command input (except in quoted strings]. All addresses and offsets are printed in hexadecimal regardless of <base_spec>.

CRASH ANALYSIS

Most fatal errors recognized by AEGIS are reported by the crash_system routine, which prints a status code (see Chapter 4, Error Codes and Messages), the address of the ECB for the failing routine, and the ID (PID) of the current process. AEGIS then executes a TRAP instruction, causing entry to the PROM mnemonic debugger with an "S" code (see MNEMONIC DEBUGGER ERROR CODES in Chapter 4). A dump can then be taken as described below.

If the system appears hung, or to stop the machine at any time for debugging, make sure the NORMAL/SERVICE switch is in the SERVICE position and type CTRL/<RETURN>, to pass control to the mnemonic debugger (MD). To continue, type:

```
> G
> G *+2 Then CTRL/<F>.
```

If this fails, press the RESET switch. (You should first verify that the hang is not a temporary one caused by a network failure.)

Notes on DNx60 Crash Status

DNx60's are microcoded machines. They may crash as described above with a CRASH SYSTEM message, ending up in MD but still running the micromachine (the CPU - MD prompt is '>') or they may occasionally freeze, (i.e. the micromachine halts), ending up in MD but running the CPIO processor (MD prompt is '%'). When a freeze occurs, the micro PC is printed by CPIO:

```
UPC: xxx
```

You may still take a dump from CPIO exactly as described above. A freeze is a more radical failure than a normal crash and usually indicates a hardware failure in the micromachine.