# PDP-11 Simulator Usage
# 01-Dec-2008

**COPYRIGHT NOTICE**

The following copyright notice applies to the SIMH source, binary, and documentation:

This memorandum documents the DEC PDP-11 simulator.

# 1  Simulator Files

To compile the PDP-11, you must define VM_PDP11 as part of the compilation command line.  If you want expanded file support, you must also define USE_INT64 and USE_ADDR64 as part of the compilation command line.

```
sim/        scp.h
            sim_console.h
            sim_defs.h
            sim_ether.h
            sim_fio.h
            sim_rev.h
            sim_sock.h
            sim_tape.h
            sim_timer.h
            sim_tmxr.h
            scp.c
            sim_console.c
            sim_ether.c
            sim_fio.c
            sim_sock.c
            sim_tape.c
            sim_timer.c
            sim_tmxr.c

sim/pdp11/  pdp11_cpumod.h
            pdp11_cr_dat.h
            pdp11_defs.h
            pdp11_mscp.h
            pdp11_uqssp.h
            pdp11_xq.h
            pdp11_xq_bootrom.h
            pdp11_cpu.c
            pdp11_cpumod.c
            pdp11_cr.c
            pdp11_dc.c
            pdp11_dl.c
            pdp11_dz.c
            pdp11_fp.c
            pdp11_hk.c
            pdp11_ke.c
            pdp11_kg.c
            pdp11_io.c
            pdp11_lp.c
            pdp11_pclk.c
            pdp11_pt.c
            pdp11_rc.c
            pdp11_rf.c
            pdp11_rh.c
            pdp11_rk.c
            pdp11_rl.c
            pdp11_rp.c
            pdp11_rq.c
```

```
pdp11_rx.c
pdp11_ry.c
pdp11_stddev.c
pdp11_sys.c
pdp11_ta.c
pdp11_tc.c
pdp11_tm.c
pdp11_tq.c
pdp11_ts.c
pdp11_tu.c
pdp11_vh.c
pdp11_xq.c
pdp11_xu.c
```

# 2  PDP-11 Features

The PDP-11 simulator is configured as follows:

```
device name(s)     simulates

CPU                PDP-11 CPU with 256KB of memory
PTR,PTP            PC11 paper tape reader/punch
TTI,TTO            DL11 console terminal
CR                 CR11/CD11 card reader
LPT                LP11 line printer
CLK                KW11-L line frequency clock
PCLK               KW11-P programmable clock
DCI,DCO            DC11 additional serial lines (up to 16)
DLI,DLO            KL11/DL11 additional serial lines (up to 16)
DZ                 DZ11 8-line terminal multiplexer (up to 4)
VH                 DHU11/DHQ11 8-line terminal multiplexer (up to 4)
RK                 RK11/RK05 cartridge disk controller with eight drives
HK                 RK611/RK06,RK07 cartridge disk controller with eight
                   Drives
RC                 RC11 fixed head disk
RF                 RF11/RS11 fixed head disk
RL                 RL11(RLV12)/RL01,RL02 cartridge disk controller with
                   four drives
RH                 RH11/RH70 Massbus adapter (up to 2)
RP                 RP04/05/06/07, RM02/03/05/80 Massbus disks with eight
                   drives
RQ                 RQDX3/UDA50 MSCP controller with four drives
RQB                second RQDX3/UDA50 MSCP controller with four drives
RQC                third RQDX3/UDA50 MSCP controller with four drives
RQD                fourth RQDX3/UDA50 MSCP controller with four drives
RX                 RX11/RX01 floppy disk controller with two drives
RY                 RX211/RX01 floppy disk controller with two drives
TA                 TA11/TU60 cassette controller with two drives
TC                 TC11/TU56 DECtape controller with eight drives
TM                 TM11/TU10 magnetic tape controller with eight drives
TS                 TS11/TSV05 magnetic tape controller with one drive
TQ                 TQK50/TU81 TMSCP magnetic tape controller with four
                   drives
TU                 TM02/TM03 magnetic tape formatter with eight drives
XQ                 DELQA/DEQNA Qbus Ethernet controller
```

```
XQB                     second DELQA/DEQNA Qbus Ethernet controller
XU                      DELUA/DEUNA Unibus Ethernet controller
XUB                     Second DELUA/DEUNA Unibus Ethernet controller
KE                      KE11A extended arithmetic option
KG                      KG11A communications arithmetic option
```

The DZ, VH, DCI/DCO, DLI/DLO, RK, HK, RC, RF, RL, RP, RQ, RQB, RQC, RQD, RX, RY, TA, TC, TM, TS, TQ, XQ, XQB, XU, XUB, KE, and KG devices can be set DISABLED.  DCI/DCO, DLI/DLO, RC, RF, RQB, RQC, RQD, RY, TA, TS, VH, XQB, XU, XUB, KE, and KG are disabled by default.

The PDP-11 simulator implements several unique stop conditions:

- Abort during exception vector fetch, and register STOP_VEC is set
- Abort during exception stack push, and register STOP_SPA is set
- Trap condition 'n' occurs, and register STOP_TRAP<n> is set
- Wait state entered, and no I/O operations outstanding (i.e., no interrupt can ever occur)
- A simulated DECtape runs off the end of its reel, and flag STOP_OFFR is set

The LOAD command supports standard binary format tapes.  The DUMP command is not implemented.

## 2.1  CPU and System

### 2.1.1  CPU

The CPU options include CPU type, CPU instruction set options for the specified type, and the size of main memory.

```
SET CPU 11/03               set CPU type to 11/03
SET CPU 11/04               set CPU type to 11/04
SET CPU 11/05               set CPU type to 11/05
SET CPU 11/20               set CPU type to 11/20
SET CPU 11/23               set CPU type to 11/23
SET CPU 11/23+              set CPU type to 11/23+
SET CPU 11/24               set CPU type to 11/24
SET CPU 11/34               set CPU type to 11/34
SET CPU 11/40               set CPU type to 11/40
SET CPU 11/44               set CPU type to 11/44
SET CPU 11/45               set CPU type to 11/45
SET CPU 11/53               set CPU type to 11/53
SET CPU 11/60               set CPU type to 11/60
SET CPU 11/70               set CPU type to 11/70
SET CPU 11/73               set CPU type to 11/73
SET CPU 11/73B              set CPU type to 11/73B
SET CPU 11/83               set CPU type to 11/83
SET CPU 11/84               set CPU type to 11/84
set CPU 11/93               set CPU type to 11/93
set CPU 11/94               set CPU type to 11/94
SET CPU U18                 deprecated; same as 11/45
SET CPU URH11               deprecated; same as 11/84
SET CPU URH70               deprecated; same as 11/70
SET CPU Q22                 deprecated; same as 11/73
SET CPU NOEIS               disable EIS instructions
SET CPU EIS                 enable EIS instructions
SET CPU NOFIS               disable FIS instructions
SET CPU FIS                 enable FIS instructions
```

```
SET CPU NOFPP                   disable FPP instructions
SET CPU FPP                     enable FPP instructions
SET CPU NOCIS                   disable CIS instructions
SET CPU CIS                     enable CIS instructions
SET CPU 16K                     set memory size = 16KB
SET CPU 32K                     set memory size = 32KB
SET CPU 48K                     set memory size = 48KB
SET CPU 64K                     set memory size = 64KB
SET CPU 96K                     set memory size = 96KB
SET CPU 128K                    set memory size = 128KB
SET CPU 192K                    set memory size = 192KB
SET CPU 256K                    set memory size = 256KB
SET CPU 384K                    set memory size = 384KB
SET CPU 512K                    set memory size = 512KB
SET CPU 768K                    set memory size = 768KB
SET CPU 1024K (or 1M)           set memory size = 1024KB
SET CPU 2048K (or 2M)           set memory size = 2048KB
SET CPU 3072K (or 3M)           set memory size = 3072KB
SET CPU 4096K (or 4M)           set memory size = 4096KB
```

The CPU types and their capabilities are shown in the following table:

| type | bus | mem | MMU? | Umap? | EIS? | FIS? | FPP? | CIS? |
|------|-----|-----|------|-------|------|------|------|------|
| 11/03 | Q | 64K | no | no | std | opt | no | no |
| 11/04 | U | 64K | no | no | no | no | no | no |
| 11/05 | U | 64K | no | no | no | no | no | no |
| 11/20 | U | 64K | no | no | no | no | no | no |
| 11/23 | Q | 4M | std | no | std | no | opt | opt |
| 11/23+ | Q | 4M | std | no | std | no | opt | opt |
| 11/24 | U | 4M | std | std | std | no | opt | opt |
| 11/34 | U | 256K | std | no | std | no | opt | no |
| 11/40 | U | 256K | std | no | std | opt | no | no |
| 11/44 | U | 4M | std | std | std | no | opt | opt |
| 11/45 | U | 256K | std | no | std | no | opt | no |
| 11/53 | Q | 4M | std | no | std | no | std | opt |
| 11/60 | U | 256K | std | no | std | no | std | no |
| 11/70 | U | 4M | std | std | std | no | opt | no |
| 11/73 | Q | 4M | std | no | std | no | std | opt |
| 11/73B | Q | 4M | std | no | std | no | std | opt |
| 11/83 | Q | 4M | std | no | std | no | std | opt |
| 11/84 | U | 4M | std | std | std | no | std | opt |
| 11/93 | Q | 4M | std | no | std | no | std | opt |
| 11/94 | U | 4M | std | std | std | no | std | opt |

If a capability is standard, it cannot be disabled; if a capability is not included, it cannot be enabled.

The CPU implements a SHOW command to display the I/O address assignments:

```
SHOW CPU IOSPACE                show I/O space address assignments
```

If memory size is being reduced, and the memory being truncated contains non-zero data, the simulator asks for confirmation.  Data in the truncated portion of memory is lost.  Initial memory size is 256KB.  If memory size is increased to more than 256KB, or the bus structure is changed, the simulator disables peripherals that can't run in the current bus structure.

These switches are recognized when examining or depositing in CPU memory:

```
-v                              interpret address as virtual
-d                              if mem mgt enabled, force data space
-k                              if mem mgt enabled, force kernel mode
-s                              if mem mgt enabled, force supervisor mode
-u                              if mem mgt enabled, force user mode
-p                              if mem mgt enabled, force previous mode
```

CPU registers include the architectural state of the PDP-11 processor as well as the control registers for the interrupt system.

```
name              size        comments

PC                16          program counter
R0..R5            16          R0..R5, current register set
SP                16          stack pointer, current mode
R00..R05          16          R0..R5, register set 0
R10..R15          16          R0..R5, register set 1
KSP               16          kernel stack pointer
SSP               16          supervisor stack pointer
USP               16          user stack pointer
PSW               16          processor status word
  CM              2           current mode, PSW<15:14>
  PM              2           previous mode, PSW<13:12>
  RS              2           register set, PSW<11>
  IPL             3           interrupt priority level, PSW<7:5>
  T               1           trace bit, PSW<4>
  N               1           negative flag, PSW<3>
  Z               1           zero flag, PSW<2>
  V               1           overflow flag, PSW<1>
  C               1           carry flag, PSW<0>
PIRQ              16          programmed interrupt requests
STKLIM            16          stack limit
FAC0H..FAC5H      32          FAC0..FAC5, high 32 bits
FAC0L..FAC5L      32          FAC0..FAC5, low 32 bits
FPS               16          floating point status
FEA               16          floating exception address
FEC               4           floating exception code
MMR0 to 3         16          memory management registers 0 to 3
{K/S/U}{I/D}{PAR/PDR}{0..7}
                  16          memory management registers
IREQ[0:7]         32          interrupt pending flags, IPL 0 to 7
TRAPS             18          trap pending flags
WAIT              0           wait state flag
WAIT_ENABLE       0           wait state enable flag
STOP_TRAPS        18          stop on trap flags
STOP_VECA         1           stop on read abort in trap or interrupt
STOP_SPA          1           stop on stack abort in trap or interrupt
PCQ[0:63]         16          PC prior to last jump, branch, or interrupt;
                              Most recent PC change first
WRU               8           interrupt character
```

The CPU attempts to detect when the simulator is idle. When idle, the simulator does not use any resources on the host system. Idle detection is controlled by the SET IDLE and SET NOIDLE commands:

```
SET CPU IDLE                         enable idle detection
SET CPU NOIDLE                       disable idle detection
```

Idle detection is disabled by default.  The CPU is considered idle if a WAIT instruction is executed.  This will work for RSTS/E and RSX-11M+, but not for RT-11 or UNIX.

The CPU can maintain a history of the most recently executed instructions.  This is controlled by the `SET CPU HISTORY` and `SHOW CPU HISTORY` commands:

```
SET CPU HISTORY                  clear history buffer
SET CPU HISTORY=0                disable history
SET CPU HISTORY=n                enable history, length = n
SHOW CPU HISTORY                 print CPU history
SHOW CPU HISTORY=n               print first n entries of CPU history
```

The maximum length for the history is 262144 entries.

## 2.1.2  System Registers (SYSTEM)

The SYSTEM device implements registers that vary among CPU types:

```
name        models                  size  comments

SR          11/04, 11/05, 11/20,    16    switch register or
            11/23+, 11/34, 11/40,         configuration register
            11/44, 11/45, 11/60,
            11/70, 11/73B, 11/83,
            11/84, 11/93, 11/94
DR          11/04, 11/05, 11/20,    16    display register or
            1123+, 11/24, 11/34,          board LEDs
            11/70, 11/73B, 11/83,
            11/84, 11/93, 11/94
MEMERR      11/44, 11/60, 11/70,    16    memory error register
            11/53, 11/73, 11/73B,
            11/83, 11/84, 11/93,
            11/94
CCR         11/44, 11/60, 11/70,    16    cache control register
            11/53, 11/73, 11/73B,
            11/83, 11/84, 11/93,
            11/94
MAINT       11/23+, 11/44, 11/70,   16    maintenance register
            11/53, 11/73, 11/73B,
            11/83, 11/84, 11/93,
            11/94
HITMISS     11/44, 11/60, 11/70,    16    hit/miss register
            11/53, 11/73, 11/73B,
            11/83, 11/84, 11/93,
            11/94
CPUERR      11/24, 11/44, 11/70,    16    CPU error register
            11/53, 11/73, 11/73B,
            11/83, 11/84, 11/93,
            11/94
MBRK        11/45, 11/70            16    microbreak register
SYSID       11/70                   16    system ID (default = 1234 hex)
JCSR        11/53, 11/73B, 11/83,   16    board control/status
            11/84, 11/93, 11/94
```

```
JPCR          11/23+, 11/53, 11/73B,   16      page control register
              11/83, 11/84, 11/93,
              11/94
JASR          11/93, 11/94             16      additional status
UDCR          11/84, 11/94             16      Unibus map diag control
UDDR          11/84, 11/94             16      Unibus map diag data
UCSR          11/84, 11/94             16      Unibus map control/status
ULAST         11/24                    23      last Unibus map result
```

For the 11/83, 11/84, 11/93, and 11/94, the user can set the default value of the clock frequency:

```
SET SYSTEM JCLK_DEFAULT={LINE|50Hz|60HZ|800HZ}
```

The user can check the default value with the SHOW SYSTEM JCLK_DEFAULT command.

## *2.2  I/O Devices*

### 2.2.1  Unibus and Qbus DMA Devices

DMA peripherals function differently, depending on whether the CPU type supports the Unibus or the Qbus, and whether the Unibus supports 22b direct memory access (11/70 with RH70 controllers):

```
peripheral  11/70        all        Qbus
            +RH70        other
                         Unibus

CD          18b          18b        disabled
RC          18b          18b        disabled
RF          18b          18b        disabled
RK          18b          18b        disabled if mem > 256K
HK          18b          18b        disabled if mem > 256K
RL          18b          18b        22b RLV12
RP          22b          18b        22b third party
RQ          18b          18b        22b RQDX3
RY          18b          18b        disabled if mem > 256K
TC          18b          18b        disabled
TM          18b          18b        disabled if mem > 256K
TS          18b          18b        22b TSV05
TQ          18b          18b        22b TQK50
TU          22b          18b        22b third party
VH          18b          18b        22b DHQ11
XQ          disabled     disabled   22b DELQA
XU          18b          18b        disabled
```

Non-DMA peripherals work the same in all configurations.  Unibus-only peripherals are disabled in a Qbus configuration, and Qbus-only peripherals are disabled in a Unibus configuration.  In addition, Qbus DMA peripherals with only 18b addressing capability are disabled in a Qbus configuration with more than 256KB memory.

### 2.2.2  I/O Device Addressing

PDP-11 I/O space and vector space are not large enough to allow all possible devices to be configured simultaneously at fixed addresses.  Instead, many devices have floating addresses and vectors; that is, the assigned device address and vector depend on the presence of other devices in the configuration:

```
DZ11                all instances have floating addresses
DHU11/DHQ11         all instances have floating addresses
RL11                first instance has fixed address, rest floating
RX11/RX211          first instance has fixed address, rest floating
DEUNA/DELUA         first instance has fixed address, rest floating
MSCP disk           first instance has fixed address, rest floating
TMSCP tape          first instance has fixed address, rest floating
```

In addition, some devices with fixed I/O space addresses have floating vector addresses. DCI/DCO and DLI/DLO have floating vector addresses.

To maintain addressing consistency as the configuration changes, the simulator implements DEC's standard I/O address and vector autoconfiguration. This allows the user to enable or disable devices without needing to manage I/O addresses and vectors. For example, if RY is enabled while RX is present, RY is assigned an I/O address in the floating I/O space range; but if RX is disabled and then RY is enabled, RY is assigned the fixed "first instance" I/O address for floppy disks.

Autoconfiguration cannot solve address conflicts between devices with overlapping fixed addresses. For example, with default I/O page addressing, the PDP-11 can support either a TM11 or a TS11, but not both, since they use the same I/O addresses.

In addition to autoconfiguration, most devices support the SET <device> ADDRESS command, which allows the I/O page address of the device to be changed, and the SET <device> VECTOR command, which allows the vector of the device to be changed. Explicitly setting the I/O address of a device that normally uses autoconfiguration DISABLES autoconfiguration for that device and for the entire system. As a consequence, the user may have to manually configure all other autoconfigured devices, because the autoconfiguration algorithm no longer recognizes the explicitly configured device. A device can be reset to autoconfigure with the SET <device> AUTOCONFIGURE command. Autoconfiguration can be restored for the entire system with the SET CPU AUTOCONFIGURE command.

The current I/O map can be displayed with the SHOW CPU IOSPACE command. Addresses that have set by autoconfiguration are marked with an asterisk (*).

All devices support the SHOW <device> ADDRESS and SHOW <device> VECTOR commands, which display the device address and vector, respectively.

## *2.3  Programmed I/O Devices*

### 2.3.1  PC11 Paper Tape Reader (PTR)

The paper tape reader (PTR) reads data from a disk file. The POS register specifies the number of the next data item to be read. Thus, by changing POS, the user can backspace or advance the reader.

The paper tape reader implements these registers:

```
name            size        comments

BUF             8           last data item processed
CSR             16          control/status register
INT             1           interrupt pending flag
ERR             1           error flag (CSR<15>)
BUSY            1           busy flag (CSR<11>)
DONE            1           device done flag (CSR<7>)
IE              1           interrupt enable flag (CSR<6>)
POS             32          position in the input file
```

```
TIME             24           time from I/O initiation to interrupt
STOP_IOE          1           stop on I/O error
```

Error handling is as follows:

```
error           STOP_IOE     processed as

not attached      1          report error and stop
                  0          out of tape

end of file       1          report error and stop
                  0          out of tape

OS I/O error      x          report error and stop
```

## 2.3.2  PC11 Paper Tape Punch (PTP)

The paper tape punch (PTP) writes data to a disk file.  The POS register specifies the number of the next
data item to be written.  Thus, by changing POS, the user can backspace or advance the punch.

The paper tape punch implements these registers:

```
name             size         comments

BUF              8            last data item processed
CSR              16           control/status register
INT              1            interrupt pending flag
ERR              1            error flag (CSR<15>)
DONE             1            device done flag (CSR<7>)
IE               1            interrupt enable flag (CSR<6>)
POS              32           position in the output file
TIME             24           time from I/O initiation to interrupt
STOP_IOE          1           stop on I/O error
```

Error handling is as follows:

```
error           STOP_IOE     processed as

not attached      1          report error and stop
                  0          out of tape

OS I/O error      x          report error and stop
```

## 2.3.3  DL11 Terminal Input (TTI)

The terminal interfaces (TTI, TTO) can be set to one of three modes, 7P, 7B or 8B:

```
mode   input characters        output characters

UC    high-order bit cleare,   high order-bit cleared,
      lower case converted     lower case converted
      to upper case            to upper case
7P    high-order bit cleared   high-order bit cleared,
                               non-printing characters suppressed
7B    high-order bit cleared   high-order bit cleared
```

```
        8B    no changes              no changes
```

The default mode is 8B.

## 2.3.4  DL11 Terminal Output (TTO)

The terminal input (TTI) polls the console keyboard for input.  It implements these registers:

```
        name            size            comments

        BUF             8               last data item processed
        CSR             16              control/status register
        INT             1               interrupt pending flag
        ERR             1               error flag (CSR<15>)
        DONE            1               device done flag (CSR<7>)
        IE              1               interrupt enable flag (CSR<6>)
        POS             32              number of characters output
        TIME            24              input polling interval (if 0, the keyboard
                                        is polled synchronously with the line clock)
```

The terminal output (TTO) writes to the simulator console window.  It implements these registers:

```
        name            size            comments

        BUF             8               last data item processed
        CSR             16              control/status register
        INT             1               interrupt pending flag
        ERR             1               error flag (CSR<15>)
        DONE            1               device done flag (CSR<7>)
        IE              1               interrupt enable flag (CSR<6>)
        POS             32              number of characters input
        TIME            24              time from I/O initiation to interrupt
```

## 2.3.5  LP11 Line Printer (LPT)

The line printer (LPT) writes data to a disk file.  The POS register specifies the number of the next data item to be written.  Thus, by changing POS, the user can backspace or advance the printer.

The line printer implements these registers:

```
        name            size            comments

        BUF             8               last data item processed
        CSR             16              control/status register
        INT             1               interrupt pending flag
        ERR             1               error flag (CSR<15>)
        DONE            1               device done flag (CSR<7>)
        IE              1               interrupt enable flag (CSR<6>)
        POS             32              position in the output file
        TIME            24              time from I/O initiation to interrupt
        STOP_IOE        1               stop on I/O error
```

Error handling is as follows:

```
        error           STOP_IOE        processed as
```

```
not attached        1               report error and stop
                    0               out of paper

OS I/O error        x               report error and stop
```

## 2.3.6  KW11-L Line-Time Clock (CLK)

The line-time clock (CLK) frequency can be adjusted as follows:

```
SET CLK 60HZ                        set frequency to 60Hz
SET CLK 50HZ                        set frequency to 50Hz
```

The default is 60Hz.

The line-time clock implements these registers:

```
name                size            comments

CSR                 16              control/status register
INT                 1               interrupt pending flag
DONE                1               device done flag (CSR<7>)
IE                  1               interrupt enable flag (CSR<6>)
TIME                24              clock interval
```

The line-time clock autocalibrates; the clock interval is adjusted up or down so that the clock tracks actual elapsed time.

## 2.3.7  KW11-P Programmable Clock (PCLK)

The programmable clock (PCLK) line frequency can be adjusted as follows:

```
SET PCLK 60HZ                       set frequency to 60Hz
SET PCLK 50HZ                       set frequency to 50Hz
```

The default is 60Hz.

The programmable clock implements these registers:

```
name                size            comments

CSR                 16              control/status register
CSB                 16              count set buffer
CNT                 16              current count
INT                 1               interrupt pending flag
OVFL                1               overflow (error) flag (CSR<15>)
DONE                1               device done flag (CSR<7>)
IE                  1               interrupt enable flag (CSR<6>)
UPDN                1               up/down count mode (CSR<4>)
MODE                1               single/repeat mode (CSR<3>)
RUN                 1               clock run (CSR<0>)
TIME[0 to 3]             32         clock interval, rates 0 to 3
TPS[0 to 3]         32              ticks per second, rates 0 to 3
```

The programmable clock autocalibrates; the clock interval is adjusted up or down so that the clock tracks actual elapsed time. Operation at the highest clock rate (100Khz) is not recommended.  The programmable clock is disabled by default.

## 2.3.8  TA11/TA60 Cassette Tape (CT)

The TA11 is a programmed I/O controller supporting two cassette drives (0 and 1).  The TA11 can be used like a small magtape under RT11 and RSX-11M, and with the CAPS-11 operating system.  Cassettes are simulated as magnetic tapes with a fixed capacity (93,000 characters).  The tape format is always SimH standard.  The TA11 is disabled by default.

TA11 options include the ability to make units write enabled or write locked.

```
SET CTn LOCKED                  set unit n write locked
SET CTn WRITEENABLED            set unit n write enabled
```

Units can not be set ENABLED or DISABLED.  The TA11 does not support the BOOT command.

The TA11 controller implements these registers:

```
name                size        comments

TACS                16          control/status register
TAIDB               8           input data buffer
TAODB               8           output data buffer
INT                 1           interrupt request
ERR                 1           error flag
TR                  1           transfer request flag
IE                  1           interrupt enable flag
WRITE               1           TA60 write operation flag
BPTR                17          buffer pointer
BLNT                17          buffer length
STIME               24          operation start time
CTIME               24          character latency
STOP_IOE            1           stop on I/O errors flag
POS[0:1]            32          position, units 0-1
```

Error handling is as follows:

```
error               processed as

not attached        tape not ready; if STOP_IOE, stop

end of file         bad tape

OS I/O error        CRC error; if STOP_IOE, stop
```

## *2.4  Floppy Disk Drives*

## 2.4.1  RX11/RX01 Floppy Disk (RX)

RX11 options include the ability to set units write enabled or write locked:

```
SET RXn LOCKED                  set unit n write locked
SET RXn WRITEENABLED            set unit n write enabled
```

The RX11 supports the BOOT command.

The RX11 implements these registers:

```
name              size          comments

RXCS              12            status
RXDB              8             data buffer
RXES              8             error status
RXERR             8             error code
RXTA              8             current track
RXSA              8             current sector
STAPTR            3             controller state
BUFPTR            3             buffer pointer
INT               1             interrupt pending flag
ERR               1             error flag (CSR<15>)
TR                1             transfer ready flag (CSR<7>)
IE                1             interrupt enable flag (CSR<6>)
DONE              1             device done flag (CSR<5>)
CTIME             24            command completion time
STIME             24            seek time, per track
XTIME             24            transfer ready delay
STOP_IOE          1             stop on I/O error
SBUF[0:127]       8             sector buffer array
```

Error handling is as follows:

```
error           STOP_IOE        processed as

not attached    1               report error and stop
                0               disk not ready
```

RX01 data files are buffered in memory; therefore, end of file and OS I/O errors cannot occur.

## 2.4.2  RX211/RX02 Floppy Disk (RY)

RX211 options include the ability to set units write enabled or write locked, single or double density, or autosized:

```
SET RYn LOCKED                   set unit n write locked
SET RYn WRITEENABLED             set unit n write enabled
SET RYn SINGLE                   set unit n single density
SET RYn DOUBLE                   set unit n double density (default)
SET RYn AUTOSIZE                 set unit n to autosize at ATTACH
```

The RX211 supports the BOOT command.  The RX211 is disabled in a Qbus system with more than 256KB of memory.

The RX211 implements these registers:

```
name              size          comments

RYCS              16            status
RYBA              16            buffer address
RYWC              8             word count
```

```
        RYDB          16            data buffer
        RYES          12            error status
        RYERR         8             error code
        RYTA          8             current track
        RYSA          8             current sector
        STAPTR        4             controller state
        INT           1             interrupt pending flag
        ERR           1             error flag (CSR<15>)
        TR            1             transfer ready flag (CSR<7>)
        IE            1             interrupt enable flag (CSR<6>)
        DONE          1             device done flag (CSR<5>)
        CTIME         24            command completion time
        STIME         24            seek time, per track
        XTIME         24            transfer ready delay
        STOP_IOE      1             stop on I/O error
        SBUF[0:255]   8             sector buffer array
```

Error handling is as follows:

```
        error         STOP_IOE      processed as

        not attached  1             report error and stop
                      0             disk not ready
```

RX02 data files are buffered in memory; therefore, end of file and OS I/O errors cannot occur.

## 2.5  Cartridge Disk Drives

### 2.5.1  RK11/RK05 Cartridge Disk (RK)

RK11 options include the ability to make units write enabled or write locked:

```
        SET RKn LOCKED                set unit n write locked
        SET RKn WRITEENABLED          set unit n write enabled
```

Units can also be set ENABLED or DISABLED.  The RK11 supports the BOOT command.  The RK11 is disabled in a Qbus system with more than 256KB of memory.

The RK11 implements these registers:

```
        name          size          comments

        RKCS          16            control/status
        RKDA          16            disk address
        RKBA          16            memory address
        RKWC          16            word count
        RKDS          16            drive status
        RKER          16            error status
        INTQ          9             interrupt queue
        DRVN          3             number of last selected drive
        INT           1             interrupt pending flag
        ERR           1             error flag (CSR<15>)
        DONE          1             device done flag (CSR<7>)
        IE            1             interrupt enable flag (CSR<6>)
        INT           1             interrupt pending flag
```

```
STIME               24              seek time, per cylinder
RTIME               24              rotational delay
STOP_IOE            1               stop on I/O error
```

Error handling is as follows:

```
error           STOP_IOE        processed as

not attached        1               report error and stop
                    0               disk not ready

end of file         x               assume rest of disk is zero

OS I/O error        x               report error and stop
```

## 2.5.2  RK611/RK06,RK07 Cartridge Disk (HK)

RK611 options include the ability to set units write enabled or write locked, to set the drive type to RK06, RK07, or autosize, and to write a DEC standard 044 compliant bad block table on the last track:

```
SET HKn LOCKED              set unit n write locked
SET HKn WRITEENABLED        set unit n write enabled
SET HKn RK06                set type to RK06
SET HKn RK07                set type to RK07
SET HKn AUTOSIZE            set type based on file size at ATTACH
SET HKn BADBLOCK            write bad block table on last track
```

The type options can be used only when a unit is not attached to a file.  The bad block option can be used only when a unit is attached to a file.  Units can be set ENABLED or DISABLED.  The RK611 supports the BOOT command.  The RK611 is disabled in a Qbus system with more than 256KB of memory.

The RK611 implements these registers:

```
name                size            comments

HKCS1               16              control/status 1
HKWC                16              word count
HKBA                16              bus address
HKDA                16              desired surface, sector
HKCS2               16              control/status 2
HKDS[0:7]           16              drive status, drives 0 to 7
HKER[0:7]           16              drive errors, drives 0 to 7
HKDB[0:2]           16              data buffer silo
HKDC                16              desired cylinder
HKOF                8               offset
HKMR                16              maintenance register
HKSPR               16              spare register
INT                 1               interrupt pending flag
ERR                 1               error flag (CSR<15>)
DONE                1               device done flag (CSR1<7>)
IE                  1               interrupt enable flag (CSR1<6>)
STIME               24              seek time, per cylinder
RTIME               24              rotational delay
STOP_IOE            1               stop on I/O error
```

Error handling is as follows:

```
error          STOP_IOE        processed as

not attached   1               report error and stop
               0               disk not ready

end of file    x               assume rest of disk is zero

OS I/O error   x               report error and stop
```

## 2.5.3 RL11(RLV12)/RL01,RL02 Cartridge Disk (RL)

RL11 options include the ability to set units write enabled or write locked, to set the drive type to RL01,
RL02, or autosize, and to write a DEC standard 044 compliant bad block table on the last track:

```
SET RLn LOCKED              set unit n write locked
SET RLn WRITEENABLED        set unit n write enabled
SET RLn RL01                set type to RL01
SET RLn RL02                set type to RL02
SET RLn AUTOSIZE            set type based on file size at ATTACH
SET RLn BADBLOCK            write bad block table on last track
```

The type options can be used only when a unit is not attached to a file. The bad block option can be used
only when a unit is attached to a file. Units can be set ENABLED or DISABLED. The RL11 supports the
BOOT command. In a Unibus system, the RL behaves like an RL11 with 18b addressing; in a Qbus (Q22)
system, the RL behaves like the RLV12 with 22b addressing.

The RL11 implements these registers:

```
name               size    comments

RLCS               16      control/status
RLDA               16      disk address
RLBA               16      memory address
RLBAE              6       memory address extension (RLV12)
RLMP,RLMP1,RLMP2   16      multipurpose register queue
INT                1       interrupt pending flag
ERR                1       error flag (CSR<15>)
DONE               1       device done flag (CSR<7>)
IE                 1       interrupt enable flag (CSR<6>)
STIME              24      seek time, per cylinder
RTIME              24      rotational delay
STOP_IOE           1       stop on I/O error
```

Error handling is as follows:

```
error          STOP_IOE        processed as

not attached   1               report error and stop
               0               disk not ready

end of file    x               assume rest of disk is zero

OS I/O error   x               report error and stop
```

## 2.6 Massbus Subsystems

### 2.6.1 RH70/RH11 Massbus Adapters (RHA, RHB)

The RH70/RH11 Massbus adapters interface Massbus peripherals to the memory bus or Unibus of the CPU.  The simulator provides two Massbus adapters.  The first, RHA, is configured for the RP family of disk drives.  The second, RHB, is configured for the TU family of tape controllers.  By default, RHA is enabled and RHB is disabled.  In a Unibus system, the RH adapters implement 22b addressing for the 11/70 and 18b addressing for all other models.  In a Qbus system, the RH adapters always implement 22b addressing.

Each RH adapter implements these registers:

```
name                size        comments

CS1                 16          control/status register 1
WC                  16          word count
BA                  16          bus address
CS2                 16          control/status register 2
DB                  16          data buffer
BAE                 6           bus address extension
CS3                 16          control/status register 3
IFF                 1           transfer complete interrupt request flop
INT                 1           interrupt pending flag
SC                  1           special condition (CSR1<15>)
DONE                1           device done flag (CSR1<7>)
IE                  1           interrupt enable flag (CSR1<6>)
```

### 2.6.2 RP04/05/06/07, RM02/03/05/80 Disk Pack Drives (RP)

The RP controller implements the Massbus family of large disk drives.  RP options include the ability to set units write enabled or write locked, to set the drive type to one of six disk types or autosize, and to write a DEC standard 044 compliant bad block table on the last track:

```
SET RPn LOCKED              set unit n write locked
SET RPn WRITEENABLED        set unit n write enabled
SET RPn RM03                set type to RM03
SET RPn RM05                set type to RM05
SET RPn RM80                set type to RM80
SET RPn RP04                set type to RP04
SET RPn RP06                set type to RP06
SET RPn RP07                set type to RP07
SET RPn AUTOSIZE            set type based on file size at ATTACH
SET RPn BADBLOCK            write bad block table on last track
```

The type options can be used only when a unit is not attached to a file.  The bad block option can be used only when a unit is attached to a file.  Units can be set ENABLED or DISABLED.  The RP controller supports the BOOT command.

The RP controller implements the registers listed below.  Registers suffixed with [0:7] are replicated per drive.

```
name                size        comments

CS1[0:7]            16          current operation
```

```
DA[0:7]              16              desired surface, sector
DS[0:7]              16              drive status
ER1[0:7]             16              drive errors
OF[0:7]              16              offset
DC[0:7]              16              desired cylinder
ER2[0:7]             16              error status 2
ER3[0:7]             16              error status 3
EC1[0:7]             16              ECC syndrome 1
EC2[0:7]             16              ECC syndrome 2
MR[0:7]              16              maintenance register
MR2[0:7]             16              maintenance register 2 (RM only)
HR[0:7]              16              holding register (RM only)
STIME                24              seek time, per cylinder
RTIME                24              rotational delay
STOP_IOE             1               stop on I/O error
```

Error handling is as follows:

```
error           STOP_IOE        processed as

not attached    1               report error and stop
                0               disk not ready

end of file     x               assume rest of disk is zero

OS I/O error    x               report error and stop
```

## 2.6.3  TM02/TM03/TE16/TU45/TU77 Magnetic Tapes (TU)

The TU controller implements the Massbus family of 800/1600bpi magnetic tape drives.  TU options include the ability to select the formatter type (TM02 or TM03), to set the drive type to one of three drives (TE16, TU45, or TU77), and to set the drives write enabled or write locked.

```
SET TU TM02                     set controller type to TM02
SET TU TM03                     set controller type to TM03
SET TUn TE16                    set drive type to TE16
SET TUn TU45                    set drive type to TU45
SET TUn TU77                    set drive type to TU77
```

Magnetic tape units can be set to a specific reel capacity in MB, or to unlimited capacity:

```
SET TUn CAPAC=m                 set unit n capacity to m MB (0 = unlimited)
SHOW TUn CAPAC                  show unit n capacity in MB
```

Units can be set ENABLED or DISABLED.  The TU controller supports the BOOT command.

The TU controller implements the following registers:

```
name                size            comments

CS1                 6               current operation
FC                  16              frame count
FS                  16              formatter status
ER                  16              formatter errors
CC                  16              check character
MR                  16              maintenance register
```

```
TC                 16            tape control register
TIME               24            operation execution time
UST                17            unit status, drives 0 to 7
POS                32            position, drive 0 to 7
STOP_IOE           1             stop of I/O error
```

Error handling is as follows:

```
error              processed as

not attached       tape not ready; if STOP_IOE, stop

end of file        bad tape

OS I/O error       parity error; if STOP_IOE, stop
```

## 2.7  RQDX3/UDA50 MSCP Disk Controllers (RQ, RQB, RQC, RQD)

The simulator implements four MSCP disk controllers, RQ, RQB, RQC, RQD.  Initially, RQB, RQC, and RQD are disabled.  Each RQ controller simulates an RQDX3 MSCP disk controller with four disk drives.  RQ options include the ability to set units write enabled or write locked, and to set the drive type to one of many disk types:

```
SET RQn LOCKED              set unit n write locked
SET RQn WRITEENABLED        set unit n write enabled
SET RQn RX50                set type to RX50
SET RQn RX33                set type to RX33
SET RQn RD51                set type to RD51
SET RQn RD52                set type to RD52
SET RQn RD53                set type to RD53
SET RQn RD54                set type to RD54
SET RQn RD31                set type to RD31
SET RQn RA81                set type to RA81
SET RQn RA82                set type to RA82
set RQn RA71                set type to RA71
SET RQn RA72                set type to RA72
SET RQn RA90                set type to RA90
SET RQn RA92                set type to RA92
SET RQn RRD40               set type to RRD40 (CD ROM)
SET RQn RAUSER{=n}          set type to RA82 with n MB's
SET -L RQn RAUSER{=n}       set type to RA82 with n LBN's
```

The type options can be used only when a unit is not attached to a file.  RAUSER is a "user specified" disk; the user can specify the size of the disk in either MB (1000000 bytes) or logical block numbers (LBN's, 512 bytes each).  The minimum size is 5MB; the maximum size is 2GB without extended file support, 1TB with extended file support.

Units can be set ENABLED or DISABLED.  Each RQ controller supports the BOOT command.  In a Unibus system, an RQ supports 18b addressing and identifies itself as a UDA50.  In a Qbus system, an RQ supports 22b addressing and identifies itself as an RQDX3.

Each RQ controller implements the following special SHOW commands:

```
SHOW RQn TYPE               show drive type
SHOW RQ RINGS               show command and response rings
```

```
SHOW RQ FREEQ                      show packet free queue
SHOW RQ RESPQ                      show packet response queue
SHOW RQ UNITQ                      show unit queues
SHOW RQ ALL                        show all ring and queue state
SHOW RQn UNITQ                     show unit queues for unit n
```

Each RQ controller implements these registers:

```
name              size        comments

SA                16          status/address register
S1DAT             16          step 1 init host data
CQBA              22          command queue base address
CQLNT             8           command queue length
CQIDX             8           command queue index
RQBA              22          request queue base address
RQLNT             8           request queue length
RQIDX             8           request queue index
FREE              5           head of free packet list
RESP              5           head of response packet list
PBSY              5           number of busy packets
CFLGS             16          controller flags
CSTA              4           controller state
PERR              9           port error number
CRED              5           host credits
HAT               17          host available timer
HTMO              17          host timeout value
CPKT[0:3]         5           current packet, units 0 to 3
PKTQ[0:3]         5           packet queue, units 0 to 3
UFLG[0:3]         16          unit flags, units 0 to 3
INT               1           interrupt request
ITIME             1           response time for initialization steps
                              (except for step 4)
QTIME             24          response time for 'immediate' packets
XTIME             24          response time for data transfers
PKTS[33*32]       16          packet buffers, 33W each, 32 entries
```

Some DEC operating systems, notably RSX11M/M+, are very sensitive to the timing parameters. Changing the default values may cause M/M+ to crash on boot or to hang during operation.

Error handling is as follows:

```
error             processed as

not attached      disk not ready

end of file       assume rest of disk is zero

OS I/O error      report error and stop
```

## *2.8  Fixed Head Disks*

### 2.8.1  RC11 Fixed Head Disk (RC)

RC11 options include the ability to set the number of platters to a fixed value between 1 and 4, or to autosize the number of platters:

```
SET RC 1P                       one platter (256K)
SET RC 2P                       two platters (512K)
SET RC 3P                       three platters (768K)
SET RC 4P                       four platters (1024K)
SET RC AUTOSIZE                 autosized on ATTACH
```

The default is one platter. The RC11 does not support the BOOT command. The RC11 is disabled at startup and is automatically disabled in a Qbus system.

The RC11 is a DMA device. The entire transfer occurs in a single DMA transfer.

The RC11 implements these registers:

```
name               size        comments

RCLA               16          look ahead register
RCDA               16          current disk address
RCER               16          error register
RCCS               16          control/status
RCWC               16          word count
RCCA               16          current memory address
RCMN               16          maintenance register
RCDB               16          data buffer
RCWLK              32          write lock switches
INT                1           interrupt pending flag
ERR                1           device error flag
DONE               1           device done flag
IE                 1           interrupt enable flag
TIME               24          rotational delay, per word
STOP_IOE           1           stop on I/O error
```

Error handling is as follows:

```
error           STOP_IOE        processed as

not attached    1               report error and stop
                0               non-existent disk
```

## 2.8.2  RF11/RS11 Fixed Head Disk (RF)

RF11 options include the ability to set the number of platters to a fixed value between 1 and 8, or to autosize the number of platters:

```
SET RF 1P                       one platter (256K)
SET RF 2P                       two platters (512K)
SET RF 3P                       three platters (768K)
SET RF 4P                       four platters (1024K)
SET RF 5P                       five platters (1280K)
SET RF 6P                       six platters (1536K)
SET RF 7P                       seven platters (1792K)
SET RF 8P                       eight platters (2048K)
SET RF AUTOSIZE                 autosized on ATTACH
```

The default is one platter.  The RF11 supports the `BOOT` command.  The RF11 is disabled at startup and is automatically disabled in a Qbus system.

The RF11 implements these registers:

```
name              size          comments

RFCS              16            control/status
RFWC              16            word count
RFCMA             16            current memory address
RFDA              16            current disk address
RFDAE             16            disk address extension
RFDBR             16            data buffer
RFMR              16            maintenance register
RFWLK             32            write lock switches
INT               1             interrupt pending flag
ERR               1             device error flag
DONE              1             device done flag
IE                1             interrupt enable flag
TIME              24            rotational delay, per word
BURST             1             burst flag
STOP_IOE          1             stop on I/O error
```

The RF11 is a DMA device.  If BURST = 0, word transfers are scheduled individually; if BURST = 1, the entire transfer occurs in a single DMA transfer.

Error handling is as follows:

```
error           STOP_IOE        processed as

not attached    1               report error and stop
                0               non-existent disk
```

RF11 data files are buffered in memory; therefore, end of file and OS I/O errors cannot occur.

## 2.9  TC11/TU56 DECtape (DT)

DECtapes drives are numbered 1-8; in the simulator, drive 8 is unit 0.  DECtape options include the ability to make units write enabled or write locked.

```
SET DTn LOCKED                  set unit n write locked
SET DTn WRITEENABLED            set unit n write enabled
```

Units can be set `ENABLED` or `DISABLED`.  The TC11 supports the `BOOT` command.  The TC11 is automatically disabled in a Qbus system.

The TC11 supports supports PDP-8 format, PDP-11 format, and 18b format DECtape images.  `ATTACH` assumes the image is in PDP-11 format; the user can force other choices with switches:

```
-t                              PDP-8 format
-f                              18b format
-a                              autoselect based on file size
```

The DECtape controller is a data-only simulator; the timing and mark track, and block header and trailer, are not stored.  Thus, the WRITE TIMING AND MARK TRACK function is not supported; the READ ALL function always returns the hardware standard block header and trailer; and the WRITE ALL function dumps non-data words into the bit bucket.

The TC controller implements these registers:

```
name                size            comments

TCST                16              status register
TCCM                16              command register
TCWC                16              word count register
TCBA                16              bus address register
TCDT                16              data register
INT                 1               interrupt pending flag
ERR                 1               error flag
DONE                1               done flag
IE                  1               interrupt enable flag
CTIME               31              time to complete transport stop
LTIME               31              time between lines
DCTIME              31              time to decelerate to a full stop
SUBSTATE            2               read/write command substate
POS[0:7]            32              position, in lines, units 0 to 7
STATT[0:7]          31              unit state, units 0 to 7
STOP_OFFR           1               stop on off-reel error
```

It is critically important to maintain certain timing relationships among the DECtape parameters, or the DECtape simulator will fail to operate correctly.

-   LTIME must be at least 6
-   DCTIME needs to be at least 100 times LTIME

Acceleration time is set to 75% of deceleration time.

## 2.10 Magnetic Tape Controllers

### 2.10.1       TM11 Magnetic Tape (TM)

TM options include the ability to make units write enabled or write locked.

```
SET TMn LOCKED              set unit n write locked
SET TMn WRITEENABLED        set unit n write enabled
```

Magnetic tape units can be set to a specific reel capacity in MB, or to unlimited capacity:

```
SET TMn CAPAC=m             set unit n capacity to m MB (0 = unlimited)
SHOW TMn CAPAC              show unit n capacity in MB
```

Units can be set ENABLED or DISABLED.

The TM11 supports the BOOT command.  The bootstrap supports both original and DEC standard boot formats.  Originally, a tape bootstrap read and executed the first record on tape.  To allow for ANSI labels, the DEC standard bootstrap skipped the first record and read and executed the second.  The DEC standard is the default; to bootstrap an original format tape, use the command BOOT –O MTn.  The TM11 is automatically disabled in a Qbus system with more than 256KB of memory.

The TM controller implements these registers:

```
name                size        comments

MTS                 16          status
MTC                 16          command
MTCMA               16          memory address
MTBRC               16          byte/record count
INT                 1           interrupt pending flag
ERR                 1           error flag
DONE                1           device done flag
IE                  1           interrupt enable flag
STOP_IOE            1           stop on I/O error
TIME                24          delay
UST[0:7]            16          unit status, units 0 to 7
POS[0:7]            32          position, units 0 to 7
```

Error handling is as follows:

```
error               processed as

not attached        tape not ready; if STOP_IOE, stop

end of file         bad tape

OS I/O error        parity error; if STOP_IOE, stop
```

## 2.10.2     TS11/TSV05 Magnetic Tape (TS)

TS options include the ability to make the unit write enabled or write locked.

```
SET TS LOCKED                   set unit write locked
SET TS WRITEENABLED             set unit write enabled
```

The TS drive can be set to a specific reel capacity in MB, or to unlimited capacity:

```
SET TS0 CAPAC=m                 set capacity to m MB (0 = unlimited)
SHOW TS0 CAPAC                  show capacity in MB
```

The TS11 supports the BOOT command.  The bootstrap supports only DEC standard boot formats.  To allow for ANSI labels, the DEC standard bootstrap skipped the first record and read and executed the second.  In a Unibus system, the TS behaves like the TS11 and implements 18b addresses.  In a Qbus system, the TS behaves like the TSV05 and implements 22b addresses.

The TS controller implements these registers:

```
name                size        comments

TSSR                16          status register
TSBA                16          bus address register
TSDBX               16          data buffer extension register
CHDR                16          command packet header
CADL                16          command packet low address or count
CADH                16          command packet high address
CLNT                16          command packet length
```

```
         MHDR                16              message packet header
         MRFC                16              message packet residual frame count
         MXS0                16              message packet extended status 0
         MXS1                16              message packet extended status 1
         MXS2                16              message packet extended status 2
         MXS3                16              message packet extended status 3
         MXS4                16              message packet extended status 4
         WADL                16              write char packet low address
         WADH                16              write char packet high address
         WLNT                16              write char packet length
         WOPT                16              write char packet options
         WXOPT               16              write char packet extended options
         ATTN                1               attention message pending
         BOOT                1               boot request pending
         OWNC                1               if set, tape owns command buffer
         OWNM                1               if set, tape owns message buffer
         TIME                24              delay
         POS                 32              position
```

Error handling is as follows:

```
         error               processed as

         not attached        tape not ready

         end of file         bad tape

         OS I/O error        fatal tape error
```

## 2.10.3    TQK50 TMSCP Disk Controller (TQ)

The TQ controller simulates the TQK50 TMSCP disk controller.  TQ options include the ability to set units
write enabled or write locked, and to specify the controller type and tape length:

```
         SET TQn LOCKED                 set unit n write locked
         SET TQn WRITEENABLED           set unit n write enabled
         SET TQ TK50                    set controller type to TK50
         SET TQ TK70                    set controller type to TK70
         SET TQ TU81                    set controller type to TU81
         SET TQ TKUSER{=n}              set controller type to TK50 with
                                        tape capacity of n MB
```

User-specified capacity must be between 50 and 2000 MB.

Regardless of the controller type, individual units can be set to a specific reel capacity in MB, or to unlimited
capacity:

```
         SET TQn CAPAC=m                set unit n capacity to m MB (0 = unlimited)
         SHOW TQn CAPAC                 show unit n capacity in MB
```

The TQ controller supports the BOOT command.   In a Unibus system, the TQ supports 18b addressing.  In a
Qbus system, the TQ supports 22b addressing.

The TQ controller implements the following special SHOW commands:

```
         SHOW TQ TYPE                   show controller type
```

```
        SHOW TQ RINGS                       show command and response rings
        SHOW TQ FREEQ                       show packet free queue
        SHOW TQ RESPQ                       show packet response queue
        SHOW TQ UNITQ                       show unit queues
        SHOW TQ ALL                         show all ring and queue state
        SHOW TQn UNITQ                      show unit queues for unit n
```

The TQ controller implements these registers:

```
        name                size            comments

        SA                  16              status/address register
        S1DAT               16              step 1 init host data
        CQBA                22              command queue base address
        CQLNT               8               command queue length
        CQIDX               8               command queue index
        RQBA                22              request queue base address
        RQLNT               8               request queue length
        RQIDX               8               request queue index
        FREE                5               head of free packet list
        RESP                5               head of response packet list
        PBSY                5               number of busy packets
        CFLGS               16              controller flags
        CSTA                4               controller state
        PERR                9               port error number
        CRED                5               host credits
        HAT                 17              host available timer
        HTMO                17              host timeout value
        CPKT[0:3]           5               current packet, units 0 to 3
        PKTQ[0:3]           5               packet queue, units 0 to 3
        UFLG[0:3]           16              unit flags, units 0 to 3
        POS[0:3]            32              tape position, units 0 to 3
        OBJP[0:3]           32              object position, units 0 to 3
        INT                 1               interrupt request
        ITIME               1               response time for initialization steps
                                            (except for step 4)
        QTIME               24              response time for 'immediate' packets
        XTIME               24              response time for data transfers
        PKTS[33*32]         16              packet buffers, 33W each, 32 entries
```

Some DEC operating systems, notably RSX11M/M+, are very sensitive to the timing parameters. Changing the default values may cause M/M+ to crash on boot or to hang during operation.

Error handling is as follows:

```
        error               processed as

        not attached        tape not ready

        end of file         end of medium

        OS I/O error        fatal tape error
```

## 2.11 Communications Devices

### 2.11.1          DC11 Additional Terminal Interfaces (DCI/DCO)

For very early system programs, the PDP-11 simulator supports up to sixteen additional DC11 terminal interfaces.  The additional terminals consist of two independent devices, DCI and DCO.  The entire set is modeled as a terminal multiplexer, with DCI as the master controller.  The additional terminals perform input and output through Telnet sessions connected to a user-specified port.  The number of lines is specified with a `SET` command:

```
    SET DCIX LINES=n               set number of additional lines to n [1-16]
```

The `ATTACH` command specifies the port to be used:

```
    ATTACH DCIX <port>             set up listening port
```

where port is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities.  The additional terminals are disabled by default.

The additional terminals can be set to one of four modes: UC, 7P, 7B, or 8B.

```
    mode   input characters          output characters

    UC     lower case converted      lower case converted to upper case,
            to upper case,           high-order bit cleared,
           high-order bit cleared    non-printing characters suppressed
    7P     high-order bit cleared    high-order bit cleared,
                                     non-printing characters suppressed
    7B     high-order bit cleared    high-order bit cleared
    8B     no changes                no changes
```

The default mode is 7P.  In addition, each line can be configured to behave as though it was attached to a dataset, or hardwired to a terminal:

```
    SET DCOn DATASET               simulate attachment to a dataset (modem)
    SET DCOn NODATASET             simulate direct attachment to a terminal
```

Finally, each line supports output logging.  The `SET DCOn LOG` command enables logging on a line:

```
    SET DCOn LOG=filename          log output of line n to filename
```

The `SET DCOn NOLOG` command disables logging and closes the open log file, if any.

Once DCI is attached and the simulator is running, the terminals listen for connections on the specified port.  They assume that the incoming connections are Telnet connections.  The connections remain open until disconnected either by the Telnet client, a `SET DCI DISCONNECT` command, or a `DETACH DCI` command.

Other special commands:

```
    SHOW DCI CONNECTIONS           show current connections
    SHOW DCI STATISTICS            show statistics for active connections
    SET DCOn DISCONNECT            disconnects the specified line.
```

The input device  (DCI) implements these registers:

```
name                size        comments

CSR[0:15]           16          input control/stats register, lines 0 to 15
BUF[0:15]           16          input buffer, lines 0 to 15
IREQ                16          interrupt requests, lines 0 to 15
```

The output device (DCO) implements these registers:

```
name                size        comments

CSR[0:15]           16          input control/stats register, lines 0 to 15
BUF[0:15]           8           input buffer, lines 0 to 15
IREQ                16          interrupt requests, lines 0 to 15
TIME[0:15]          31          time from I/O initiation to interrupt,
                                lines 0 to 15
```

The additional terminals do not support save and restore.  All open connections are lost when the simulator shuts down or DCI is detached.

## 2.11.2    KL11/DL11 Additional Terminal Interfaces (DLI/DLO)

The PDP-11 simulator supports up to sixteen additional KL11/DL11 terminal interfaces.  The additional terminals consist of two independent devices, DLI and DLO.  The entire set is modeled as a terminal multiplexer, with DLI as the master controller.  The additional terminals perform input and output through Telnet sessions connected to a user-specified port.  The number of lines is specified with a SET command:

```
SET DLI LINES=n                 set number of additional lines to n [1–16]
```

The ATTACH command specifies the port to be used:

```
ATTACH DLI <port>       set up listening port
```

where port is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities.  The additional terminals are disabled by default.

The additional terminals can be set to one of four modes: UC, 7P, 7B, or 8B.

```
mode   input characters         output characters

UC     lower case converted     lower case converted to upper case,
        to upper case,          high-order bit cleared,
       high-order bit cleared   non-printing characters suppressed
7P     high-order bit cleared   high-order bit cleared,
                                non-printing characters suppressed
7B     high-order bit cleared   high-order bit cleared
8B     no changes               no changes
```

The default mode is 7P.  .  In addition, each line can be configured to behave as though it was attached to a dataset, or hardwired to a terminal:

```
SET DLOn DATASET                simulate attachment to a dataset (modem)
SET DLOn NODATASET              simulate direct attachment to a terminal
```

Finally, each line supports output logging.  The SET DLOn LOG command enables logging on a line:

```
        SET DLOn LOG=filename          log output of line n to filename
```

The `SET DLOn NOLOG` command disables logging and closes the open log file, if any.

Once DLI is attached and the simulator is running, the terminals listen for connections on the specified port. They assume that the incoming connections are Telnet connections. The connections remain open until disconnected either by the Telnet client, a `SET DLI DISCONNECT` command, or a `DETACH DLI` command.

Other special commands:

```
        SHOW DLI CONNECTIONS           show current connections
        SHOW DLI STATISTICS            show statistics for active connections
        SET DLOn DISCONNECT            disconnects the specified line.
```

The input device  (DLI) implements these registers:

```
        name              size         comments

        CSR[0:15]         16           input control/stats register, lines 0 to 15
        BUF[0:15]         16           input buffer, lines 0 to 15
        IREQ              16           receive interrupt requests, lines 0 to 15
        DSI               16           dataset interrupt requests, lines 0 to 15
```

The output device (DLO) implements these registers:

```
        name              size         comments

        CSR[0:15]         16           input control/stats register, lines 0 to 15
        BUF[0:15]         8            input buffer, lines 0 to 15
        IREQ              16           interrupt requests, lines 0 to 15
        TIME[0:15]        31           time from I/O initiation to interrupt,
                                       lines 0 to 15
```

The additional terminals do not support save and restore.  All open connections are lost when the simulator shuts down or DLO is detached.

## 2.11.3        DZ11 Terminal Multiplexer (DZ)

The DZ11 is an 8-line terminal multiplexer.  Up to 4 DZ11's (32 lines) are supported.  The number of lines can be changed with the command

```
        SET DZ LINES=n                 set line count to n
```

The line count must be a multiple of 8, with a maximum of 32.

The DZ11 supports three character processing modes, 7P, 7B, and 8B:

```
        mode   input characters        output characters

        7P     high-order bit cleared  high-order bit cleared,
                                       non-printing characters suppressed
        7B     high-order bit cleared  high-order bit cleared
        8B     no changes              no changes
```

The default is 8B.

The DZ11 supports logging on a per-line basis.  The command

```
SET DZ LOG=line=filename
```

enables logging for the specified line to the indicated file.  The command

```
SET DZ NOLOG=line
```

disables logging for the specified line and closes any open log file.  Finally, the command

```
SHOW DZ LOG
```

displays logging information for all DZ lines.

The terminal lines perform input and output through Telnet sessions connected to a user-specified port.  The ATTACH command specifies the port to be used:

```
ATTACH {-am} DZ <port>        set up listening port
```

where port is a decimal number between 1 and 65535 that is not being used for other TCP/IP activities.  The optional switch -m turns on the DZ11's modem controls; the optional switch -a turns on active disconnects (disconnect session if computer clears Data Terminal Ready).  Without modem control, the DZ behaves as though terminals were directly connected; disconnecting the Telnet session does not cause any operating system-visible change in line status.

Once the DZ is attached and the simulator is running, the DZ will listen for connections on the specified port.  It assumes that the incoming connections are Telnet connections.  The connection remains open until disconnected by the simulated program, the Telnet client, a SET DZ DISCONNECT command, or a DETACH DZ command.

Other special DZ commands:

```
SHOW DZ CONNECTIONS         show current connections
SHOW DZ STATISTICS          show statistics for active connections
SET DZ DISCONNECT=linenumber  disconnects the specified line.
```

The DZ11 implements these registers:

```
name                size        comments

CSR[0:3]            16          control/status register, boards 0 to 3
RBUF[0:3]           16          receive buffer, boards 0 to 3
LPR[0:3]            16          line parameter register, boards 0 to 3
TCR[0:3]            16          transmission control register, boards 0 to 3
MSR[0:3]            16          modem status register, boards 0 to 3
TDR[0:3]            16          transmit data register, boards 0 to 3
SAENB[0:3]          1           silo alarm enabled, boards 0 to 3
RXINT               4           receive interrupts, boards 3..0
TXINT               4           transmit interrupts, boards 3..0
MDMTCL              1           modem control enabled
AUTODS              1           autodisconnect enabled
```

The DZ11 does not support save and restore.  All open connections are lost when the simulator shuts down or the DZ is detached.

## 2.11.4 DHQ11 Terminal Multiplexer (VH)

The DHQ11 is an 8-line terminal multiplexer for Qbus systems.  Up to 4 DHQ11's are supported.

The DHQ11 is a programmable asynchronous terminal multiplexer.  It has two programming modes: DHV11 and DHU11.  The register sets are compatible with these devices.  For transmission, the DHQ11 can be used in either DMA or programmed I/O mode.  For reception, there is a 256-entry FIFO for received characters, dataset status changes, and diagnostic information, and a programmable input interrupt timer (in DHU mode).  The device supports 16-, 18-, and 22-bit addressing.  The DHQ11 can be programmed to filter and/or handle XON/XOFF characters independently of the processor.  The DHQ11 supports programmable bit width (between 5 and 8) for the input and output of characters.

The DHQ11 has a rocker switch for determining the programming mode.  By default, the DHV11 mode is selected, though DHU11 mode is recommended for applications that can support it.  The VH controller may be adjusted on a per controller basis as follows:

```
    SET VHn DHU                    use the DHU programming mode and registers
    SET VHn DHV                    use the DHV programming mode and registers
```

DMA output is supported.  In a real DHQ11, DMA is not initiated immediately upon receipt of TX.DMA.START but is dependent upon some internal processes.  The VH controller mimics this behavior by default.  It may be desirable to alter this and start immediately, though this may not be compatible with all operating systems and diagnostics.  You can change the behavior of the VH controller as follows:

```
    SET VHn NORMAL                 use normal DMA procedures
    SET VHn FASTDMA                set DMA to initiate immediately
```

The terminal lines perform input and output through Telnet sessions connected to a user-specified port.  The ATTACH command specifies the port to be used:

```
    ATTACH VH <port>               set up listening port
```

where port is a decimal number between 1 and 65535 that is not  being used for other TCP/IP activities.  This port is the point of entry for al lines on all VH controllers.

Modem and auto-disconnect support may be set on an individual controller basis.  The SET MODEM command directs the controller to report modem status changes to the computer.  The SET HANGUP command turns on active disconnects (disconnect session if computer clears Data Terminal Ready).

```
    SET VHn [NO]MODEM              disable/enable modem control
    SET VHn [NO]HANGUP             disable/enable disconnect on DTR drop
```

Once the VH is attached and the simulator is running, the VH will listen for connections on the specified port.  It assumes that the incoming connections are Telnet connections.  The connection remains open until disconnected by the simulated program, the Telnet client, a SET VH DISCONNECT command, or a DETACH VH command.

Other special VH commands:

```
    SHOW VH CONNECTIONS            show current connections
    SHOW VH STATISTICS            show statistics for active connections
    SET VH DISCONNECT=linenumber  disconnects the specified line.
```

The DHQ11 implements these registers, though not all can be examined from SCP:

```
        name            size            comments

        CSR[0:3]        16              control/status register, boards 0 to 3
        RBUF[0:3]       16              receive buffer, boards 0 to 3
        LPR[0:3]        16              line parameter register, boards 0 to 3
        RXINT           4               receive interrupts, boards 3..0
        TXINT           4               transmit interrupts, boards 3..0
```

[more to be described...]

The DHQ11 does not support save and restore.  All open connections are lost when the simulator shuts down or the VH is detached.

## 2.12 Ethernet Controllers

### 2.12.1        DELQA/DEQNA Qbus Ethernet Controllers (XQ, XQB)

The simulator implements two DELQA/DEQNA Qbus Ethernet controllers (XQ, XQB).  Initially, XQ is enabled, and XQB is disabled.  Options allow control of the MAC address, the controller mode, and the sanity timer.

```
        SET  XQ MAC=<mac-address>       ex. 08-00-2B-AA-BB-CC
        SHOW XQ MAC
```

These commands are used to change or display the MAC address.  <mac-address> is a valid ethernet MAC, delimited by dashes or periods.  The controller defaults to 08-00-2B-AA-BB-CC, which should be sufficient if there is only one SIMH controller on your LAN.  Two cards with the same MAC address will see each other's packets, resulting in a serious mess.

```
        SET  XQ TYPE={DEQNA|[DELQA]}
        SHOW XQ TYPE
```

These commands are used to change or display the controller mode. DELQA mode is better and faster but may not be usable by older or non-DEC OS's.  Also, be aware that DEQNA mode is not supported by many modern OS's. The DEQNA-LOCK mode of the DELQA card is emulated by setting the the controller to DEQNA -- there is no need for a separate mode.  DEQNA-LOCK mode behaves exactly like a DEQNA, except for the operation of the VAR and MOP processing.

```
        SET  XQ SANITY={ON|[OFF]}
        SHOW XQ SANITY
```

These commands change or display the INITIALIZATION sanity timer (DEQNA jumper W3/DELQA switch S4).  The INITIALIZATION sanity timer has a default timeout of 4 minutes, and cannot be turned off, just reset.  The normal sanity timer can be set by operating system software regardless of the state of this switch.  Note that only the DEQNA (or the DELQA in DEQNA-LOCK mode (=DEQNA)) supports the sanity timer -- it is ignored by a DELQA in Normal mode, which uses switch S4 for a different purpose.

```
        SET  XQ POLL={DEFAULT|4..2500}
        SHOW XQ POLL
```

These commands change or display the service polling timer.  The polling timer is calibrated to run the service thread 200 times per second.  This value can be changed to accomodate particular system requirements for more (or less) frequent polling.

```
        SHOW XQ STATS
```

This command will display the accumulated statistics for the simulated Ethernet controller.

To access the network, the simulated Ethernet controller must be attached to a real Ethernet interface:

```
ATTACH XQ0 {ethX|<device_name>}              ex. eth0 or /dev/era0
SHOW XQ ETH
```

where X in 'ethX' is the number of the Ethernet controller to attach, or the real device name.  The X number is system dependant.  If you only have one Ethernet controller, the number will probably be 0. To find out what your system thinks the Ethernet numbers are, use the SHOW XQ ETH command.  The device list can be quite cryptic, depending on the host system, but is probably better than guessing. If you do not attach the device, the controller will behave as though the Ethernet cable were unplugged.

XQ and XQB have the following registers:

```
name             size          comments

SA0              16            station address word 0
SA1              16            station address word 1
SA2              16            station address word 2
SA3              16            station address word 3
SA4              16            station address word 4
SA5              16            station address word 5
RBDL             32            receive buffer descriptor list
XBDL             32            trans(X)mit buffer descriptor list
CSR              16            control status register
VAR              16            vector address register
INT              1             interrupt request flag
```

One final note: because of its asynchronous nature, the XQ controller is not limited to the ~1.5Mbit/sec of the real DEQNA/DELQA controllers, nor the 10Mbit/sec of a standard Ethernet.  Attach it to a Fast Ethernet (100 Mbit/sec) card, and "Feel the Power!" :-)

### 2.12.2          DELUA/DEUNA Unibus Ethernet Controllers (XU, XUB)

The simulator implements two DELUA/DEUNA Unibus Ethernet controllers (XU, XUB).  Its operation is analogous to the DELQA/DEQNA controller.

## 2.13 CR11/CD11 Card Reader (CR)

The card reader (CR) implements a single controller (either the CR11 or the CD11) and card reader (e.g., Documation M200, GDI Model 100) by reading a file and presenting lines or cards to the simulator.  Card decks may be represented by plain text ASCII files, card image files, or column binary files.  The CR11 controller is also compatible with the CM11-F, CME11, and CMS11.

Card image files are a file format designed by Douglas W. Jones at the University of Iowa to support the interchange of card deck data.  These files have a much richer information carrying capacity than plain ASCII files.  Card Image files can contain such interchange information as card-stock color, corner cuts, special artwork, as well as the binary punch data representing all 12 columns.  Complete details on the format, as well as sample code, are available at Prof. Jones's site: http://www.cs.uiowa.edu/~jones/cards/.

The card reader can be configured to support either of the two controllers supported by DEC:

```
        SET CR CR11            set controller type to CR11
        SET CR CD11            set controller type to CD11
```

The controller type must be set before attaching a virtual card deck to the device.  You may NOT change controller type once a file is attached.

 The primary differences are summarized in the table below.  By default, CR11 simulation is selected.

```
                       CR11              CD11
BR                     6                 4
registers              4                 3
data transfer          BR                DMA
card rate              200-600           1000-1200
hopper cap.            <= 1000           1000-2250
cards                  Mark-sense &      punched only
                       punched
```

Examples of the CR11 include the M8290 and M8291 (CMS11).  All card readers use a common vector at 0230 and CSR at 177160.  Even though the CR11 is normally configured as a BR6 device, it is configured for BR4 in this simulation.

The card reader supports ASCII, card image, and column binary format card "decks."  When reading plain ASCII files, lines longer than 80 characters are silently truncated.  Card image support is included for 80 column Hollerith, 82 column Hollerith (silently ignoring columns 0 and 81), and 40 column Hollerith (mark-sense) cards.  Column binary supports 80 column card images only.  All files are attached read-only (as if the -R switch were given).

```
        ATTACH -A CR <file>                    file is ASCII text
        ATTACH -B CR <file>                    file is column binary
        ATTACH -I CR <file>                    file is card image format
```

If no flags are given, the file extension is evaluated.  If the filename ends in .TXT, the file is treated as ASCII text.  If the filename ends in .CBN, the file is treated as column binary.  Otherwise, the CR driver looks for a card image header.  If a correct header is found the file is treated as card image format, otherwise it is treated as ASCII text.

The correct character translation MUST be set if a plain text file is to be used for card deck input.  The correct translation SHOULD be set to allow correct ASCII debugging of a card image or column binary input deck.  Depending upon the operating system in use, how it was generated, and how the card data will be read and used, the translation must be set correctly so that the proper character set is used by the driver.  Use the following command to explicitly set the correct translation:

```
        SET TRANSLATION={DEFAULT|026|026FTN|029|EBCDIC}
```

This command should be given after a deck is attached to the simulator.  The mappings above are completely described at http://www.cs.uiowa.edu/~jones/cards/codes.html.  Note that DEC typically used 029 or 026FTN mappings.

DEC operating systems used a variety of methods to determine the end of a deck (recognizing that 'hopper empty' does not necessarily mean the end of a deck.  Below is a summary of the various operating system conventions for signaling end of deck:

```
        RT-11:      12-11-0-1-6-7-8-9 punch in column 1


        RSTS/E:     12-11-0-1 or 12-11-0-1-6-7-8-9 punch in column 1
```

```
        RSX:            12-11-0-1-6-7-8-9 punch

        VMS:            12-11-0-1-6-7-8-9 punch in first 8 columns

        TOPS:           12-11-0-1 or 12-11-0-1-6-7-8-9 punch in column 1
```

Using the AUTOEOF setting, the card reader can be set to automatically generate an EOF card consisting of the 12-11-0-1-6-7-8-9 punch in columns 1-8.  When set to CD11 mode, this switch also enables automatic setting of the EOF bit in the controller after the EOF card has been processed.  [The CR11 does not have a similar capability.]  By default AUTOEOF is enabled.

```
        SET CR AUTOEOF
        SET CR NOAUTOEOF
```

The default card reader rate for the CR11 is 285 cpm, while the default rate for the CD11 is 1000 cpm.  The reader rate can be set to its default value or to anywhere in the range 200..1200 cpm.  This rate may be changed while the unit is attached.

```
        SET CR RATE={DEFAULT|200..1200}
```

It is standard operating procedure for operators to load a card deck and press the momentary action RESET button to clear any error conditions and alert the processor that a deck is available to read.  Use the following command to simulate pressing the card reader RESET button,

```
        SET CR RESET
```

Another common control of physical card readers is the STOP button.  An operator could use this button to finish the read operation for the current card and terminate reading a deck early.  Use the following command to simulate pressing the card reader STOP button.

```
        SET CR STOP
```

The simulator does not support the BOOT command.  The simulator does not stop on file I/O errors.  Instead the controller signals a reader check to the CPU.

The CR controller implements these registers:

```
        name            size            comments

        BUF             8               ASCII value of last column processed
        CRS             16              CR11 status register
        CRB1            16              CR11 12-bit Hollerith character
        CRB2            16              CR11 8-bit compressed character
        CRM             16              CR11 maintenance register
        CDST            16              CD11 control/status register
        CDCC            16              CD11 column count
        CDBA            16              CD11 current bus address
        CDDB            16              CD11 data buffer, 2nd status
        BLOWER          2               blower state value
        INT             1               interrupt pending flag
        ERR             1               error flag (CRS<15>)
        IE              1               interrupt enable flag (CRS<6>)
        POS             32              file position - do not alter
        TIME            24              delay time between columns
```

The CD11 simulation includes the Rev. J modification to make the CDDB act as a second status register during non-data transfer periods.

## 2.14 Arithmetic Options

### 2.14.1        KE11A Extended Arithmetic Option (KE)

The KE11A extended arithmetic option (KE) provides multiply, divide, normalization, and multi-bit shift capability on Unibus PDP-11's that lack the EIS instruction set.  In practice, it was only sold with the PDP-11/20.  The KE is disabled by default.

The KE implements these registers:

```
name            size        comments

AC              16          accumulator
MQ              16          multiplier-quotient
SC              6           shift count
SR              8           status register
```

### 2.14.2        KG11A Communications Arithmetic Option (KG)

The KG11-A is a programmed I/O, non-interrupting, dedicated arithmetic processor for the Unibus.  The device is used to compute the block check character (BCC) over a block of data, typically in data communication applications.  The KG11 can compute three different Cyclic Redundancy Check (CRC) polynomials (CRC-16, CRC-12, CRC-CCITT) and two Longitudinal Redundancy Checks (LRC, Exclusive-OR; LRC-8, LRC-16).  Up to eight units may be contiguously present in a single machine and are all located at fixed addresses.  This simulation implements all functionality of the device including the ability to single step computation of the BCC.   The KG is disabled by default.

The KG11 supports the following options:

```
SET KG UNITS=n              set the number of units [0-8]
SET KG DEBUG={opt,opt…}     set the debugging options
                            REG - any time a register is touched
                            POLY - any time the polynomial is changed
                            CYCLE - each cycle computing the polynomial
```

The KG11 implements the following registers, replicated for each unit:

```
name            size        comments

SR[0:7]         16          control and status register; R/W
BCC[0:7]        16          result block check character; R/O
DR[0:7]         16          input data register; W/O
PULSCNT[0:7]    16          polynomial cycle stage
```

# 3  Symbolic Display and Input

The PDP-11 simulator implements symbolic display and input.  Display is controlled by command line switches:

```
        -a              display as ASCII character
        -c              display as two packed ASCII characters
        -m              display instruction mnemonics
```

Input parsing is controlled by the first character typed in or by command line switches:

```
        ' or -a         ASCII character
        " or -c         two packed ASCII characters
        alphabetic      instruction mnemonic
        numeric         octal number
```

Instruction input uses standard PDP-11 assembler syntax.  There are sixteen instruction classes:

```
class               operands                examples                comments

no operands         none                    HALT, RESET
3b literal          literal [0 to 7]        SPL
6b literal          literal [0-077]         MARK
8b literal          literal [0-0377]        EMT, TRAP
register            register                RTS
sop                 specifier               SWAB, CLR, ASL
reg-sop             register, specifier     JSR, XOR, MUL
fop                 flt specifier           ABSf, NEGf
ac-fop              flt reg, flt specifier  LDf, MULf
ac-sop              flt reg, specifier      LDEXP, STEXP
ac-moded sop        flt reg, specifier      LDCif, STCfi
dop                 specifier, specifier    MOV, ADD, BIC
cond branch         address                 BR, BCC, BNE
sob                 register, address       SOB
cc clear            cc clear instructions   CLC, CLV, CLZ, CLN      combinable
cc set              cc set instructions     SEC, SEV, SEZ, SEN      combinable
```

For floating point opcodes, F and D variants, and I and L variants, may be specified regardless of the state of FPS.

The syntax for specifiers is as follows:

```
syntax              specifier   displacement        comments

Rn                  0n          -
Fn                  0n          -                   only in flt reg classes
(Rn)                1n          -
@(Rn)               7n          0                   equivalent to @0(Rn)
(Rn)+               2n          -
@(Rn)+              3n          -
-(Rn)               4n          -
@-(Rn)              5n          -
{+/-}d(Rn)          6n          {+/-}d
@{+/-}d(Rn)         7n          {+/-}d
#n                  27          n
@#n                 37          n
.+/-n               67          +/-n - 4
@.+/-n              77          +/-n - 4
{+/-}n              67          {+/-}n - PC - 4    if on disk, 37 and n
@{+/-}n             77          {+/-}n - PC - 4    if on disk, invalid
```