MCR10120
Fix Interpretation of DPS8/M ID PROM Fields
Revision 1.1

Eric Swenson

August 24, 2022

1 Problem Description

When Multics starts up and gets configured CPUs running, or when CPUs are added during Multics service, a message is added to the `syserr_log` of the form:

```
21:37:49 3539213 4 CPU A: Model #: DPS 8/SIM M0; Serial #:    2208; Ship date:
    19\004\024\031\000\014\377.
```

The reason is that the person who added support for these log messages interpreted the documentation on the DPS8 ID PROM to specify offsets to strings in the PROM in decimal, when, in fact they were intended to be in octal. When the dps8 simulator was written, it initialized the ID PROM such that the messages in the `syserr_log` read correctly, however, later, when trying to ensure the simulator allowed ISOLTS to run successfully, we discovered that the tests assumed octal offsets. In addition, we located the Honeywell *Multics Differences Manual DPS 8 70/M* manual, which confirmed the intent and the hardware implementation.

If the offsets in the Multics code that emits these messages are adjusted to use an octal interpretation, then the messages once again read correctly. For example:

```
17:01:29 3540447 4 CPU A: Model #: DPS 8/SIM M; Serial #: 9; Ship date: 220819.
```

The ticket for this issue is https://multics-trac.swenson.org/ticket/247

2 Analysis

The simulator was changed quite some time ago to make ISOLTS properly interpret the ID prom. To wit, ISOLTS now displays output like the following:

```
prom data indicates following system
bytes 0-33
   model      serial   date
   dps 8/sim m0        220820
```

```
bytes 34-40
processor     dps processor
   bcd options
   dps options          1
   8k cache             1
   mult options         1
   cpl/npl
   port speed=fast cm
   ou  speed=fast ou
   hexadecimal option
   rscr  option         1
bytes 50
gcos lockout bytes
   gcos3                1
   gcos8                1
   cp6                  1
   multics              1
```

Interestingly, the *Multics Differences Manual DPS 8 70/M* manual, on pages 32-33, includes the following table, which has an error in it:

Byte Location (Octal)	Contents
0-13	CPU Model Number (Byte 0 - Most Significant Byte)
13-25	CPU Serial Number (Byte 13 - Most Significant Byte)
26-33	Date-Ship Code (YYMMDD)

Note the overlap of the last byte of CPU Model Number and the first byte of CPU Serial Number. The correct value for the last byte offset for the CPU Model Number is 12. Thus, the table should have read:

Byte Location (Octal)	Contents
0-12	CPU Model Number (Byte 0 - Most Significant Byte)
13-25	CPU Serial Number (Byte 13 - Most Significant Byte)
26-33	Date-Ship Code (YYMMDD)

ISOLTS correctly handles the 11-byte CPU Model Number string.

There are two places in Multics where the ID PROM is read. They both use the procedure `privileged_mode_ut$read_id_prom` to read from the ID PROM. This procedure is declared as follows:

```
dcl privileged_mode_ut$read_id_prom entry (char (*) aligned, fixed bin);
```

The implementation of this ALM routine is to use the Multics argument descriptor for the first parameter, declared `char (*) aligned`, to determine the length of the supplied parameter string, and uses this length as the number of bytes to read from the ID PROM. The second parameter, declared `fixed bin` specifies the offset in the ID PROM.

A typical call to this procedure would look like this:

```
call privileged_mode_ut$read_id_prom (cpu_model, 0);
```

In that call, the `cpu_model` variable was originally declared:

```
dcl cpu_model           char (13) aligned;
```

The author of the code which added support for the ID PROM clearly looked at the spec, saw the 0-13, realized that it should have been 0-12, interpreted these values as decimal, and computed a length of 13 for the CPU Model. This turned out to be incorrect. In fact the original code had incorrect values for the lengths of each of the variables to receive values from ID PROM, and incorrect offsets for the CPU Serial Number and Date-Ship Code. This is all due to incorrectly interpreting the offsets as decimal, rather than octal.

3 Proposed Fix

The proposed fix is quite simple. There are two programs – `start_cpu.pl1` and `tc_init.pl1` that have identical declarations and code for reading and emitting to the `syserr_log` messages indicating the startup of DPS8/M CPUs. Both of these require the same fixes. The offsets to the values in the ID PROM for the CPU Serial Number and Data Ship Code must be adjusted to be the decimal equivalent of the octal interpretation of the specification. And the declarations for the string variables that receive the results from `privileged_mode_ut$read_id_prom` must be adjusted to be their correct lengths. The changes are as follows:

In `start_cpu.pl1`:

```
cpa [lpn start_cpu.pl1] ==

A75               cpu_model           char (13) aligned,      /* storage for cpu model
    number (from ID PROM) */
A76               cpu_serial          char (13) aligned,      /* storage for cpu serial
    number (from ID PROM) */
A77               cpu_ship_date       char (8) aligned;       /* storage for cpu ship
    date (from ID PROM) */
Changed by B to:
B75               cpu_model           char (11) aligned,      /* storage for cpu model
    number (from ID PROM) */
B76               cpu_serial          char (11) aligned,      /* storage for cpu serial
    number (from ID PROM) */
B77               cpu_ship_date       char (6) aligned;       /* storage for cpu ship
    date (from ID PROM) */


A558                          call privileged_mode_ut$read_id_prom (cpu_serial, 13);
A559                                                          /* get cpu serial # from ID
    PROM */
A560                          call privileged_mode_ut$read_id_prom (cpu_ship_date, 26);
A561                                                          /* get ship date from ID
    PROM */
Changed by B to:
B558                          call privileged_mode_ut$read_id_prom (cpu_serial, 11);
B559                                                          /* get cpu serial # from ID
    PROM, at offset 13o */
```

```
B560                          call privileged_mode_ut$read_id_prom (cpu_ship_date, 22);
B561                                                    /* get ship date from ID
    PROM, at offset 26o */

Comparison finished: 2 differences, 14 lines.
```

And in `tc_init.pl1`:

```
A77       dcl  cpu_model char (13) aligned;                   /* storage for cpu model
    number (from ID PROM) */
A78       dcl  cpu_serial char (13) aligned;                  /* storage for cpu serial
    number (from ID PROM) */
A79       dcl  cpu_ship_date char (8) aligned;                /* storage for cpu ship
    date (from ID PROM) */
Changed by B to:
B77       dcl  cpu_model char (11) aligned;                   /* storage for cpu model
    number (from ID PROM) */
B78       dcl  cpu_serial char (11) aligned;                  /* storage for cpu serial
    number (from ID PROM) */
B79       dcl  cpu_ship_date char (6) aligned;                /* storage for cpu ship
    date (from ID PROM) */


A564                      call privileged_mode_ut$read_id_prom (cpu_serial, 13);
A565                                                    /* get cpu serial # from ID
    PROM */
A566                      call privileged_mode_ut$read_id_prom (cpu_ship_date, 26);
A567                                                    /* get ship date from ID
    PROM */
Changed by B to:
B564                      call privileged_mode_ut$read_id_prom (cpu_serial, 11);
B565                                                    /* get cpu serial # from ID
    PROM, at offset 13o */
B566                      call privileged_mode_ut$read_id_prom (cpu_ship_date, 22);
B567                                                    /* get ship date from ID
    PROM, at offset 26o */

Comparison finished: 2 differences, 14 lines.
```

4 Testing

A Multics tape was created with the above fix. Once the system was started, the `print_syserr_log` command was issued with the following arguments:

```
psl -syserr -from -1hour -match CPU
```

The result was:

```
17:01:29 3540447 4 CPU A: Model #: DPS 8/SIM M; Serial #: 9; Ship date: 220819.
```

```
17:01:29 3540448 0 start_cpu: Added CPU B.
r 17:03 1.199 49
```

This indicates proper interpretation of the ID PROM fields of interest to `tc_init`. In addition, a DPS/M CPU was deleted and then added dynamically added, and again the `syserr_err` log was checked and seen to be correct. This indicates proper interpretation in `start_cpu`.

In addition, the GHM_Test system has been running with this fix as well with no issues.

Our CI/CD process for simulator changes runs a series of tests which begin with cold booting a Multics system, configuring a system, and then running a bunch of texts. It culminates with running ISOLTS. This process will be run (has been run once already) using a hardcore with the above fix. The output logs from the CI-KIT will be inspected to ensure that the `syserr` messages are correct.

5 Documentation

There are no documentation changes required by this change apart from comments in the code and history comments.

6 Version History

2022-08-22	1.0	Initial version of the MCR.
2022-08-24	1.0	Correct a couple typos.