MCR10216
Detect Two CPUs on One SCU Port in Config Deck
Revision 1.0

Eric Swenson

November 24, 2026

1 Problem Description

Ticket #256 reports that if you author two CPUs, one specified as "on" and the other specified as "off", but specifying the same SCU port, such as shown below:

```
cpu a 7 on dps8 70. 8.
cpu b 7 off dps8 70. 8.
```

the system will not warn you, and adding the "off" cpu later will hang the system.

1.1 Cause

`scas_init` (`bound_temp_1`), which processes the CPU cards at initialization, checks for duplicates only among CPUs on port expanders (two CPUs on the same subport of an expanded port, or a port used both expanded and not). Two CPUs on the same ordinary port are not detected.

2 Proposed Fix

The proposed fix is to add an additional check for all CPUs specified on each the SCU ports, for duplicates:

For a CPU not on an expander, `scas_init` now also checks the CPUs already processed:

```
          do i = 0 to 7;
                if scs$processor_data (i).offline
                then if scs$processor_data (i).controller_port = cport
                     then call syserr (CRASH, "scas_init: CPU ^a and CPU ^a both
                          configured for SCU port ^d.",
                              tag_letter (tag), tag_letter (i), cport);
          end;
```

This is the same test, and the same severity, as the existing check for a duplicated expander subport: the system returns to BCE, where the deck can be corrected. Message documentation is

added.

The compare_ascii comparison of the current and proposed scas_init.pl1 is below;

```
compare_ascii >ldd>hard>source>bound_temp_1.s.archive::scas_init.pl1 ==

A237              else if scs$expanded_ports (cport)
Changed by B to:
B237              else do;
B238                   if scs$expanded_ports (cport)


Inserted in B:
B241                   do i = 0 to 7;                          /* Two CPUs on one port:
    adding the second   */
B242                      if scs$processor_data (i).offline    /*  one hangs the system
    (ticket 256). */
B243                      then if scs$processor_data (i).controller_port = cport
B244                           then call syserr (CRASH, "scas_init: CPU ^a and CPU ^a both
    configured for SCU port ^d.",
B245                                      tag_letter (tag), tag_letter (i), cport);
B246                   end;
B247                   end;
Preceding:
A240              pdata.controller_port = cport;               /* Get it in the processor
    data. */


Inserted in B:
B928        scas_init: CPU X and CPU Y both configured for SCU port P.
B929
B930        S:     $crash
B931
B932        T:     $init
B933
B934        M:     The CPU CONFIG cards for CPU's X and Y both indicate that
B935        they are connected to SCU port P, which is not an expanded port.
B936        Only one CPU can be connected to such a port; adding the second
B937        would hang the system.
B938
B939        A:     Correct the configuration deck and reboot.
B940
B941        Message:
Preceding:
A920        scas_init: MEM X is an old SCU with port expander

Comparison finished: 3 differences, 24 lines.
```

3 Documentation

New message documented at the end of `scas_init.pl1`. See above in `compare_ascii` comparison.

4 Test

1. Boot with the ticket's deck: the system crashes during initialization with "CPU B and CPU A both configured for SCU port 7." and returns to BCE.
2. Boot with distinct ports, and with CPUs on an expander: unchanged.

5 Files Changed

`bound_temp_1.s.archive`	`scas_init.pl1`
`>sl1`	`bound_temp_1` (hardcore; new system tape)

6 Version History

2026-11-24	1.0	Initial version of the MCR.