MCR10206
scr_util$assign_mask assigns both free SCU masks (duplicate mask assignment after ISOLTS)
Revision 1.0

Eric Swenson

November 11, 2026

1 Problem Description

After ISOLTS has tested a processor, rebooting Multics without resetting the system controllers can fail in BCE with

```
scas_init: MEM B has duplicate mask assignments to CPU A
```

Seen on the DPS8M emulator (2026-11-02): `test cpu b` under ISOLTS, `quit`, then `shut` and `boot star` in the same emulator session. The controller holding the test processor's memory (MEM B) had both of its execute-interrupt masks assigned to CPU A's port.

`configure_test_cpu` needs a mask on the controller being tested that is assigned to an online processor (the "mask CPU"). If no mask of that controller is assigned to one, but a mask is free (4MW SCU), it calls `scr_util$assign_mask` to assign a free mask to the first online processor. It then saves the controller's configuration data (`scs$cfg_data_save = scs$cfg_data (iscu)`). During the test, `scr_util$isolts_scu_p1` sets mask A to the mask CPU and mask B to the test CPU; at the end `isolts_scu_p3` writes the saved configuration back to the controller.

`assign_mask` loops over the two masks of a 4MW SCU and assigns every free mask it finds: the loop has no exit after the first assignment. On a controller with no masks assigned (the usual case for a controller with no IOM connected, such as MEM B here), both masks are assigned to the mask CPU, in `scs$controller_data` (`eima_data`) and in `scs$cfg_data`. The saved copy has both, and `isolts_scu_p3` writes both to the controller. The controller keeps that until it is cleared, so the next bootload's `init_scu` (via `scas_init`) finds two masks assigned to one processor and stops.

Controller state observed with the emulator's `show scu1 state` (CPU A on port 7, CPU B on port 6):

When	Mask A	Mask B	Port 6
After bootload	off	off	disabled
During the test (`*options?`)	7	6	enabled
After `quit`	7	7	disabled

The running system is not otherwise affected: a later `test cpu` in the same bootload finds mask A assigned to CPU A and does not call `assign_mask` again. Only the next bootload fails. On hardware, the duplicate assignment stays until the controller is initialized.

2 Proposed Fix

`assign_mask` returns after assigning the first free mask:

```
            else
            scr_cfg2.mask_b_assign = mask_assignment;
            return;                      /* Assign only one mask. Without this,
                                            both free masks went to the target,
                                            so the next bootload's init_scu
                                            found duplicate masks. */
         end;
```

After an ISOLTS test the controller then has mask A assigned to the mask CPU and mask B free. `init_scu` accepts this (one mask per processor, every assigned mask on a configured processor), and a later `test cpu` uses mask A without assigning again.

`assign_mask` is called only by `configure_test_cpu`. `unassign_mask` (not called anywhere) already handles two masks assigned to one port and is unchanged.

The `compare_ascii` comparison of the original and proposed `scr_util.pl1` follows:

```
compare_ascii >ldd>hard>source>bound_wired_1.s.archive::scr_util.pl1 ==

Inserted in B:
B8
B9       /****^ HISTORY COMMENTS:
B10        1) change(2026-11-11,Swenson), approve(2026-11-11,MCR10206):
B11           Fix to assign only one mask and return as soon as it has done so.
B12                                                    END HISTORY COMMENTS */
B13
Preceding:
A8       /* SCR_UTIL - Procedure to Manipulate System Controller Registers and Data.


Inserted in B:
B270                     return;                              /* Assign only one mask.
    Without this,
B271                                                             both free masks went to
    the target,
B272                                                             so the next bootload's
    init_scu
B273                                                             found duplicate masks. */
Preceding:
A264                  end;

Comparison finished: 2 differences, 10 lines.
```

Workaround until installed: initialize the system controllers (on the emulator, restart it) before rebooting after an ISOLTS session.

3 Documentation

None.

4 Test

On the DPS8M emulator, with a system tape that includes the new `bound_wired_1`, two processors (CPU A port 7, CPU B port 6, set up for ISOLTS) and MEM B with no masks assigned at bootload:

1. Boot. `show scu1 state` at the emulator prompt: both masks off.
2. Log in, `isolts`, `test cpu b`: the test environment is set up as before. At `*options?`, `show scu1 state`: mask A = 7, mask B = 6, port 6 enabled.
3. Run a page (`tst700`), then `quit`. `show scu1 state`: mask A = 7, mask B off (before the fix: both 7), port 6 disabled.
4. `test cpu b` again in the same bootload: set up as before (mask A found, no new assignment); `quit`; masks unchanged.
5. `shut`, then `boot star` without restarting the emulator: Multics boots (before the fix: "scas_init: MEM B has duplicate mask assignments to CPU A"). Then `test cpu b` once more.
6. A normal bootload and clean shutdown with no ISOLTS use.

5 Files Changed

`bound_wired_1.s.archive`	`scr_util.pl1`

The change is in the hardcore (supervisor) library, so it needs a new system tape.

6 Version History

2026-11-03	1.0	Initial version of the MCR.