TO: Operations

FROM: Steve Webber

DATE: October 5, 1973

SUBJECT: Reconfiguration of Memories, Processors and Bulk Store

This document gives instructions for doing dynamic reconfiguration of CPU's, memories and the Bulk Store. There are six commands which can be used to reconfigure and all of them are recognized by the Answering Service (without entering admin mode). In addition, any of the commands can be called from any SysDaemon or SysAdmin process in the event the Answering Service can not be used.

The processor and memory reconfiguration commands are all controlled by a reconfiguration "lock". If a reconfiguration attempt aborts for some reason this lock may be left ON. When this happens any further attempts to reconfigure will result in a message saying that reconfiguration is currently in progress. If this happens the following command may be executed:

        reconfigure$force_unlock

This will unlock the lock so that further reconfiguration attempts may be made. However, it is quite possible that the reconfiguration data bases are still in an inconsistent state (e.g. thinking a CPU was added when it really wasn't). If this is the case all reconfiguration requests for CPU's and memories should be delayed until after the next bootload.

Terminology and Switches

1. Each memory has one and only one "control CPU". This is
   the CPU that receives interrupts that are set in the
   memory. This is also the CPU that is pointed to by the
   EXECUTE INTERRUPT MASK ASSIGNMENT (EIMA) switch A on the
   memory. (Note that EIMA switches B, C, and D should always
   be OFF.)

2. Each CPU has one memory which is called its "control
   memory". This is the memory to which interrupts are sent
   which are directed to the CPU. Note that each CPU must
   have a memory associated with it --- there must be at
   least as many memories in the system as CPU's.

3. The PORT ENABLE switches on the CPU (and IOM and Bulk
   Store) determine which memories the CPU (or IOM or Bulk
   Store) may access. When reconfiguring, these switches
   control the effect of the other switches in the same row
   on the maintainance panel. It is imperative that all other
   switches (INITIALIZE ENABLE, ASSIGNMENT, INTERLACE,
   ADDRESS RANGE) in the same row as a given PORT ENABLE
   switch be correct before the PORT ENABLE switch is turned
   ON. In particular, if any switches in the same row are to
   be changed, the PORT ENABLE switch should be turned OFF.
   The IOM and Bulk Store switches work in the same way with
   the additional restriction that on the Bulk Store the PORT
   GROUP switch should also be set correctly before turning
   on the PORT ENABLE switch.

4. The INITIALIZE ENABLE switch on the CPU's, IOM's and Bulk

Store only come into play when the BOS (or T and D) system
is bootloaded. When this is done all of these switches
should be ON so that the initialize signal can be
reflected to all modules of the system being bootloaded.
After BOS is bootloaded all INITIALIZE ENABLE switches
should be turned OFF. The Bulk Store, however, should
never have its INITIALIZE ENABLE switches in the ON
position. The Bulk Store should be initialized manually
prior to bootload of either service or development system.
If either system is using the Bulk Store it need not (and
should not) be initialized.

5. The ASSIGNMENT switches on the CPU's, IOM's and Bulk Store
   specify in which order the memories are to be used. All
   assignment switches for a given system should be the same
   on all active modules in the system. (An active module is
   a CPU, an IOM or a Bulk Store.) Note that if it is
   necessary to change an ASSIGNMENT switch (e.g. to add a
   memory) on an active module, the PORT ENABLE switch
   corresponding to the ASSIGNMENT switch should first be set
   OFF.

6. The INTERLACE switches on the active modules should be set
   as indicated in any reconfiguration instructions. These
   switches reflect the state of the "external" interlace
   between two memory controllers. (The INTERLACE switches on
   the memories indicate "internal" interlace and should
   always be left in the ON position.)

7. The ADDRESS RANGE switches on the avtive modules should
   always be left in the FULL position. (At times of
   specific hardware problems FED may recommend changing
   these.)

8. The EXECUTE INTERRUPT MASK ASSIGNMENT (EIMA) switches on
   the memories control where interrupts sent to that memory
   will be directed. (These switches correspond to the
   CONTROL PROCESSOR switch on the 645 memories.) Only switch
   A should ever be in a position other than OFF.

9. The PORT CONTROL switches on the memories control which
   active modules have access to the memory. When in the PROG
   CONT position, the software can and does control this
   access by enabling only those active modules which are
   currently configured. (see later)

Adding Bulk Store (Paging Device)

        To   add   a   region   of   the   Bulk   Store   to   the   Multics
configuration the following command is used:

        addpag first count

where first is the (octal) number of the first Bulk Store record
to add and count is the (octal) number of records to add. The
block of records starting at first and going for count records
will be set up for use by the software. The hardware switches
enabling the given region of the Bulk Store must already have
been set up before this command is issued. (See MOSN-4.4.4.3 for
information about setting the Bulk Store configuration switches.)

Deleting Bulk Store (Paging Device)

To delete a region of the Bulk Store from the Multics configuration the following command is used:

delpag first count

"first" and "count" are the first record and size, respectively, of a region to be deconfigured. They should both be octal. This command tells the software not to use the specified region of the Bulk Store and hence after it is issued the region may be physically reconfigured if desired.

Adding a Processor

To add a processor to the system a well defined sequence of events must be followed. The script that follows is a sample output from a console session for adding a CPU:

|---|

addcpu b 6 b

Check that the following has been done (if not, do it in the following order):

```
    All memories: PORT CONTROL (port 6) set to PROG CONT
    cpu b must have the following switch settings:
      memory a:    Assignment 0    Interlace OFF    PORT ENABLE OFF
      memory b:    Assignment 1    Interlace OFF    PORT ENABLE OFF
      memory c:    Assignment 2    Interlace OFF    PORT ENABLE OFF
      CPU NO:   2
```

Have all the above been done?(yes/no and initials): yes shw

Remember to initialize and enable cpu b after changing the EIMA switch.

You will change EIMA switch / on memory b to port 6.
Wait until instructed to change them.   Are you ready? (yes/no): yes
Change switch(es) now.   Type 'done' when done: done

cpu b is now running.

|--|

In essence the following steps must be taken after the second
answer "yes":
   1. Change EIMA switch A on the control memory to the given
      port for the processor being added.
   2. Initialize the processor to be added (by pressing
      INITIALIZE & CLEAR).
   3. Set the PORT ENABLE switches ON (for all memories in the
      system) on the CPU to be added.
   4. Answer "done".

If the CPU fails to start running for some reason the software

will automatically instruct the operator to change the EIMA

switch back to its original position. Another attempt to add the

CPU can be issued when the command finishes its cleanup.

   A second CPU to be added to the system should be added _after_

the STARTUP command has been issued.

Deleting a Processor

   To remove a processor from the system the operator types

      delcpu N

where N is the name (e.g. "a", "b", etc.) of the processor to be

deleted. The program will instruct the operator to change EIMA

switches on all memories which are controlled by the CPU being

deleted. After the delcpu command has finished the processor

switches can be changed as desired --- the system software

insures that the processor can not access any of the service

memories. (It is not necessary to change the PORT CONTROL

switches on all memories for the processor being deleted.)

Adding a Memory

The following is a sample console output for adding a memory
to the system:

|--|

addmem c a

Make sure
IOM A, BULK STORE A, CPU A
 have the following switch settings:

        memory a: Assignment 0   Interlace OFF   PORT ENABLE ON
        memory b: Assignment 1   Interlace OFF   PORT ENABLE ON
        memory c: Assignment 2   Interlace OFF   PORT ENABLE ON

Perform the following (in the order given) on system controller
c.
     PORT CONTROL set to DISABLE for all ports.
     Initialize controller at its maintainance panel.
     Change EXECUTE INTERRUPT MASK ASSIGNMENT switch A to port 7.
     PORT CONTROL set to PROG CONT for ports 0, 2 and 7.

Have all the above been done?(yes/no and initials): yes shw

You have added memory controller c.

|--|

The argument to the addmem command is the memory name (e.g. "c")

as on the ROS configuration card. An additional argument can

optionally be provided to specify a particular CPU (e.g. "a") as

the control processor for the memory being added.

Removing a Memory

     To remove a memory from the system the following command

should be issued:

          delmem N

where N is the name of the memory as on the ROS configuration

card. Deleting a memory which is the control memory for some

processor will trigger code instructing the operator to change
the EIMA switch on a memory to be control memory for the CPU.
Note: the bootload (low-order) memory can not be deleted.

Notes

When the reconfiguration commands request the operator to
give a "yes/no" answer, then any answer other than "yes" will be
interpreted as "no" and will properly terminate the
reconfiguration request. When the operator is asked for a "yes"
or "done" answer, he must do as directed and answer appropriately
or risk crashing the system. If the error message "program error
-- notify programmer" ever occurs, DO NOT try to use any
reconfiguration command again until cleared by a programmer, or
until after the next bootload.

Note that when the configuration is changed dynamically, the
BOS cards for processors, memories and Bulk Store may no longer
reflect the true configuration. Therefore, before the next
bootload or salvage the hardware configuration and the BOS
configuration deck must be made consistent with each other (the
CONFIG U command in BOS should probably be used).

It is only possible to add Bulk Store to a system if the
system was informed about the ultimate possible size of the
paging device at system bootload time. This is indicated on the
page configuration card which specifies the maximum size of the
Bulk Store as well as optional regions which should not be used.

(See MOSM-4-3 for a description on the page configuration card.)

Only memories that are defined by BOS cards at bootload time can be added. The configuration card for a memory has a field to indicate if it is on-line or off-line at bootload time. All the memories actually being used at the time of the bootload must have an ON in the last field in the BOS configuration card. These memories should be the first in the configuration deck. Memories which might be added later have an OFF in this field. These should follow the other memory cards (i.e., these are the high order memories). Normally, in the service machine BOS deck there should be a memory card (either ON or OFF) for every memory in the installation.

The PORT CONTROL switches for all memory controllers running on the service system should be set in the PROG CONT position for <u>all</u> ports at all times. For the development system, all ports should be ENABLEd or DISABLEd to reflect the running development configuration. Note that after adding a memory unused ports are left in the OFF position. <u>After</u> the adding of the memory is completed the operator will put <u>ALL</u> PORT CONTROL switches in the PROG CONT position.

Returning to BOS

Under normal running conditions the EXECUTE button is used to force the system to enter BOS. This can be done in one of two ways:

1) Usually BOS will be entered by causing an execute fault. For
   this to occur, the EXECUTE SWITCHES switch must be _down_ when
   the EXECUTE button is pushed. The switch should normally be
   left in the down position. DO NOT put the processor in "step"
   when using the execute fault.

2) BOS can also be entered by executing the instruction pair
   located at location 4000 or 4002 in memory. In this case, the
   desired address (viz. 4000 or 4002) is set in switches 0-17
   and an execute double (XED) instruction is set in switches
   18-35 (the octal value of the switches should be either
   004000717200 or 004002717200). If the EXECUTE SWITCHES switch
   is in the up position when the EXECUTE buuton is pushed the
   XED will be executed: the processor should be in "step" when
   the button is pushed. BOS should be entered with an XED only
   in the cases noted below, or as a last resort.

BOS Configuration Deck

The order of the cards in the configuration deck is
important for cpu's and Memories. The first CPU card is for the
bootload CPU; the MEM cards must be in the order of increasing
address.

Crashes and Shutdowns

If the above instructions are followed, then whenever
Multics crashes and BOS is entered either by Multics or manually,
the normal dump procedures (FDUMP, etc.) should be followed. This
is independent of the number of processors running. It is not

necessary to explicitly dump the process that was executing on the non-bootload CPU. (It is still useful to copy down the PPR of the non-bootload CPU). If Multics will not go back to BOS for some reason, place both CPU's in "step" and _transfer_ back to BOS on the bootload CPU using 004000717200 in the switches and the EXECUTE SWITCHES switch in the _up_ position. If this succeeds, proceed normally. If this fails, try 004002717200. If this succeeds do not attempt an ESD; proceed to the salvager directly.

Before attempting an ESD, or a Salvage, all EIMA switches on all memories should be pointing at the bootload CPU.

Warning: With two processors, _NEVER_ put either processor in "step" (except in the case Multics will not reenter BOS). If either or both processors must be put in "step" (e.g., for work by the FE's) then enter BOS with the "bos" command on an initializer console; put both processors in "step" only while in BOS.