Published: 11/03/67

Identification

Major Module Configuration Table Initialization for Initial Multics
J. H. Saltzer

Purpose

Multics can operate with a wide variety of 645 system hardware cabling arrangements and switch settings. (Restrictions on possible configurations are detailed in section BC.1.01.) The source of the operating system's information about the configuration is the Major Module Configuration Table (MMCT) which can describe any allowable configuration. Section BL.5.02 defines the MMCT. The Major Module Configuration Table is used primarily during system initialization, to aid in construction of specialized tables of the system such as the core map and the system communication segment.

Early in the initialization of the system, the Major Module Configuration Table must be set up to describe the particular hardware configuration which is available. In order to do this set up, the complete description of the configuration of the system must be somehow communicated to a MMCT initializer which proceeds to build the MMCT.

For Initial Multics, a simplified MMCT initializer is used. The main characteristic of this simplified program is that it assumes that it knows the hardware cabling arrangements and most of the switch settings of the particular installation and requires as further input only the statement of presence or absence of each major module. This final input is provided by setting for each major module one of the "processor data" switches before system initialization.

The precise list of assumptions made by this program is given below. It is essential to note that the Initial Multics MMCT initializer program is the only Multics procedure which makes these assumptions. All other Multics procedures, whether operating during initialization or afterwards, must obtain their knowledge of the system configuration by consulting the Major Module Configuration Table, or data bases derived from it. Put another way, any change in the list of assumptions below should result in a change to the single program mmct_init; no other Multics procedure should be affected, except possibly bootload programs which execute before mmct_init is called.

Calling sequence:

mmct_init is an EPL procedure, called by:

```
                call mmct_init;
```

without arguments. In lieu of arguments, mmct_init assumes that the following variables of the segment initialization_constants (described in Section BL.3.03) have already been set by the caller:

```
     tag.port(i)     /* cpu tag - port number table      */
     switches        /* bootload processor data switches */
     cpu_base        /* bootload processor base address  */
     gioc_base       /* bootload gioc base address       */
     gioc_port       /* port number of bootload gioc     */
     cpu_tag         /* cpu tag of bootload cpu          */
```

mmct_init, on the basis of these constants and the assumptions below, constructs the segment "mmct" in the format described in BK.4.04, and then returns.

Configuration assumptions.

The following assumptions are made by the Initial Multics MMCT setup routine for the Project MAC and BTL 645 configurations:*

*Note: Two important considerations have played a part in selecting the assumptions listed here:

a. It should be possible to operate on either the Project MAC or BTL 645 system with the same Multics System Tape, at least initially.

b. Whenever possible, there should be "standard Multics settings" for 645 system configuration switches. These standard settings should not change when a major module is removed from the system for repair.

1. The cabling among major modules is exactly that described in the current version of MSPM section BC.1.02.

2. All interrupt cell patch boards are set in accordance with the assignments shown in the current version of MSPM Section BC.1.04, and the "rule of 32" is in force.

3. The presence or absence of each major module is established by reading the processor data switches, as follows:

switch	module
1-20	(not used)
21	CPU A
22	CPU B
23	GIOC A
24	GIOC B
25	drum
26	prototype clock A
27	prototype clock B
28	System Controller E
29	
30	System Controller F
31	
32	System Controller G
33	
34	System Controller H
35	

For each of these modules, switch *down* (bit value 0) means the module is absent, switch *up* (bit value 1) means the module is present.

For each system controller, a pair of switches is used, with meaning.

00	64K controller absent
01	64K controller present
10	128K controller absent
11	128K controller present

4. Memory Address assignments are made according to the following rule: System Controllers with 128K of memory are all assigned lower addresses than those with 64K of memory. Controllers of like size are ordered alphabetically with lower addresses in the lower-lettered controllers. Addresses are assigned starting at location zero and without gaps. Initially, if the controller is absent (switch setting 00 or 10) no addresses are assigned to it. Thus, initially, the address space is contiguous and starts at location 0. This technique is chosen because GECOS requires a contiguous, zero-based address space and initially there will be frequent switches between GECOS and Multics. The specified address assignment need not be changed to perform the transition.

When GECOS dependence is lessened, controllers should be assigned the same address ranges whether present or absent, since Multics can withstand addressing "holes" and constant address assignments result in permanent, unchanging, standard Multics port selection switch settings in the CPU's, GIOC, and drum.

5. Prototype Clock addressing assignments are fixed as follows:

clock	address
A	640K
B	768K

These high addresses are used to allow for the possibility that controllers E, F, G, and H all have 128K block assignments. This address assignment results in the following permanent port selection switch settings in each CPU:

port	selection bits
A	101
B	110

6. Clock A is the primary clock, if present, and clock B is backup, unless clock A is not available.

7. If two CPU's are present in the configuration, they have the same base address.

8. The drum base address is at location 2K relative to the bootload GIOC base address.

9. If a second GIOC is present in the configuration, and one processor is in use, the second GIOC base address is at location 1K relative to the first GIOC base address.

10. If two CPU's are present in the configuration, the bootload GIOC base address is located in some system controller; that system controller directs interrupts to the bootload CPU.

11. If a second GIOC is present in the configuration, and two processors are in use, the second GIOC base address is at location -1K relative to the highest address in the collection of system controllers which interrupt the second CPU.

12. If four identical-size system controllers are available, 4-way interlace is used. Otherwise, 2-way interlace is used where possible, namely if either of the pairs EF or GH are present and of the same size.