Published: 04/25/68
(Supersedes: BG.5.00, 06/07/67,
BG.5.00, 06/06/66)

Identification

Structure of the Core Map
P. G. Neumann, M. R. Wagner, and R. C. Daley

Purpose

This section describes in detail the internal structure of the core map. The core map is a wired down data base which describes the status of all core memory available to the Multics system and is manipulated exclusively by the modules of core control.

Introduction

The core map is maintained in the wired-down data segment cormap and is logically divided into three data bases, a 1024-word block map, a 64-word block map and a core map header. The 1024-word block map is an array containing an entry for each 1024-word block of core available to the Multics system and is constructed so that each entry may be directly referenced by using the 1024-word core block address (high order 14 bits of a 24 bit address) as an index into the array. During Multics initialization some of the available 1024-word blocks are split into 16 contiguous 64-word blocks to make up the 64-word block map. When this is done, the 1024-word block map entry is flagged with a special code and an index to the 16 contiguous entries in the 64-word block map is placed in the corresponding 1024-word block map entry.

The 64-word block map contains 16 contiguous entries for each 1024-word block which has been split into 16 contiguous 64-word blocks. A single entry in the 64-word map corresponding to a 64-word block may also be referenced directly by the absolute core location of that block in the following way. First, the corresponding 1024-word map entry is referenced by using the high-order 14 bits of the memory address as an index into the 1024-word map. If this entry indicates that the 1024-word block has been split, the entry will also contain an index to the first of the 16 contiguous entries in the 64-word map which correspond to this 1024-word block. Second, the 64-word map entry is then referenced by adding this index with the next 4 bits (bits 15-18) of the memory address to form the desired index into the 64-word map.

Both the 1024-word map and the 64-word map are contained in a single array. The first N entries in the array correspond to the available 1024-word blocks in a system configured for N*1024 words of memory. The 64-word map begins at the entry indexed by N+1 and continues through N+M, where M is the number of 64-word blocks in the 64-word map.

Each entry in the 1024 and 64-word block maps are labeled with a 3-bit type code to indicate the current status of the corresponding block of core. The following list summarizes these type codes and their meanings.

Type 0	is used in the 1024-word map to indicate a 1024-word block of core which is unavailable to the system (e.g. part of a bad core stack).
Type 1	(free) is used to indicate a free block of core which is available for assignment.
Type 2	(removable) is used to indicate a block of core which is currently assigned to a page of a segment which may be removed from core whenever it becomes necessary (see removal strategy in BG.5.02).
Type 3	(wired) is used to indicate a block of core which is currently assigned to a page or page table which may not be removed from core unless a specific request is made to do so or unless the status is changed to removable (type 2).
Type 4	(permanent wired) is used to indicate a block of core which has been assigned to part of the wired-down hardcore supervisor and may not be unassigned.
Type 5	(temporary) is used only during system initialization to indicate pages and page tables of temporary segments loaded by interim_1_segfault in loading collection 2. (Consult the BL sections of this manual for a discussion of initialization and temporary segments.)
Type 6	is currently unassigned.
Type 7	is used to indicate a 1024-word block which has been split into 16 contiguous 64-word blocks.

For ease of manipulation, block map entries are threaded into circular lists in which each entry contains an index to its successor and to its predecessor. Separate lists are maintained for core blocks of types 1-5 and are called respectively, the free list, the removal list, the wired list, the permanent list and the temporary list. Two independent sets of these lists are maintained, one for 1024-word blocks and the other for 64-word blocks. No lists are maintained for core blocks of types 0, 6 and 7 in the core map.

The core map header contains the index of the first entry for each of the above lists and a count of the number entries in each list. The header also contains a pointer to the array containing the 1024-word and 64-word block maps, indices into that array defining the first and last entry in the 64-word block map and two counts (see below) to help in evaluating the performance of the core management module.

Structure of the Core Map Header

The following PL/I declaration gives the structure of the core map header. The individual items are described in detail below.

```
    dcl 1 cm based (pc),

        2 lp (0:15),

          3 first fixed bin (12),

          3 total fixed bin (12),

        2 bmptr ptr,

        2 start_64 fixed bin (12),

        2 end_64 fixed bin (12),

        2 notfound fixed bin (12),

        2 retry fixed bin (12),

        2 map ptr;
```

lp-	is an array containing information about each of the various threaded lists contained in the core map. The type code (0-7) is used as the index to find information for lists in the 1024-word map, while the type code +8 (8-15) is used for the 64-word map. For example, lp(1) defines the 1024-word free list and lp(9) defines the 64-word free list. For each list, an index (first) to the first entry in the threaded list and a count in the threaded list and a count (total) of the number of entries in the list are maintained.
bmptr-	is a pointer to the block map array which contains the 1024-word and 64-word block maps (see below).
start 64-	is an index to the first entry in the 64-word block map. (start_64 -1 is the index to the last in the 1024-word map.)
end_64-	is an index to the (fictitious) entry after the last entry in the 64-word map. Both start_64 and end_64 are used only when initializing the 64-word map and are left in the core map only for debugging purposes.
retry-	is a count of the number of times that a 1024-word block was used to satisfy a request for a 64-word block and is also intended as a metering device.
map-	is a dummy item used to determine the base location of the block map array. The pointer to the block map array is obtained in the following PL/I statement. pc→cm.bmptr = addr (pc→cm.map);

Structure of the Block Map

The following PL/I declaration gives the structure of the block map array which contains both the 1024-word and 64-word block maps. The individual items are described in detail below.

```
     dcl 1 ml (262144) based (pc→cm.bmptr),

          2 point bit (12),

          2 back bit (12),

          2 type bit (3),

          2 initial bit (1),

          2 pageno bit (8),

          2 sstrp bit (18),

          2 loc bit (18);
```

point-	is the index of the next entry in the threaded list (if any) of which this entry is a member. In the case where the entry is a 1024-word block which has been split (type=7) into 16 64-word blocks, this item is reinterpreted as an index to the first of the 16 contiguous entries in the 64-word map which correspond to this 1024-word map block.
back-	is the index of the preceding entry in the threaded list (if any).
type-	is the type code (0-7) indicating the current status of this block of core.
initial-	is used by the core management removal algorithm (see BG.5.02) to distinguish the initial usage of a page from its subsequent usage.
pageno-	is the page number (if any) of the page for which the block of core has been assigned (used only where applicable).
sstrp-	is a relative pointer to the SST entry defining the segment to which this page has been assigned. This item applies only when the core block is assigned as a page and type is removable (2) or wired (3).
loc-	is the high order 18 bits of a 24-bit absolute memory address defining the first location of the core block (low order 6 bits assumed zero).