

J. E. Griffith

May 25, 1957

FILE MEMO

SUBJECT: Clear Memory and Lock Memory

BY: W. Buchholz

For Harvest, it is planned to "clear" a block of memory by a single CLEAR MEMORY instruction. The specific scheme, devised to avoid special reset windings linking all cores in all planes, is to add another bit plane with set windings linking only the cores in that plane.

Assuming a block of 1024 words, CLEAR MEMORY sets the corresponding 1024 clear bits to one. The next time a reference is made to one of the memory locations in that block, the previous contents are ignored. If the operation is reading from memory, an all zero word is sent out in place of the previous contents. If the operation is writing, OR-ing, or counting in memory, the incoming word is simply stored, replacing the previous contents; at the same time the clear bit for that word only is reset to zero. The memory behaves as if the contents had actually been cleared when the CLEAR MEMORY instruction is given, even though the data bits are not changed immediately.

The clear bit thus has the meaning "write only". An analogous operation would be to provide a second special bit plane indicating "read only". The purpose of this plane is to "lock" a block of memory and prevent its contents from being altered.

A LOCK MEMORY instruction sets all lock bits for the desired memory block to one. Any read-out operation for that block of memory will not be affected. But any attempt to write into a memory location, for which the lock bit is on, will be suppressed. The original memory contents will be restored and a program interrupt will take place to signal an illegitimate operation. An UNLOCK MEMORY instruction is needed to reset the lock bits in a block to its normal zero state, permitting normal operations for that memory block.

A simpler scheme would provide lock bits in a single flip-flop for the entire memory unit or in a small group of flip-flops, one for each block of memory. The choice of scheme depends on the size of the block to be locked in one operation. It is wasteful of memory to make the block too large.

May 25, 1957

Locking memory provides a measure of protection against unauthorized interference between programs, particularly during the debugging stage. While a program is debugged, all memory areas not assigned to this program would be locked out. It would also be desirable to lock out memory areas occupied by supervisory programs.

To avoid unauthorized clearing and unlocking of memory areas, the assembly and loading programs could search all operating programs for CLEAR and UNLOCK instructions. While this is not complete protection, since any program could create such instructions after entry, it is much less likely that this will happen accidentally than that an indexing operation will accidentally alter a given memory area. It is also much easier to censor operation codes than to censor effective memory references resulting from complex indexing.

W. B. ...
5-25-57

WB/pkb

cc: J. F. Hanifin (2)