The following is the way I normally used to clear memory.
Apparently, the variant is not being interpreted properly.
The 2nd and 3rd pages show the actual instruction and variant table.

25 140 77 110 110
A77 0000001
A74 0000000
A70 0000001

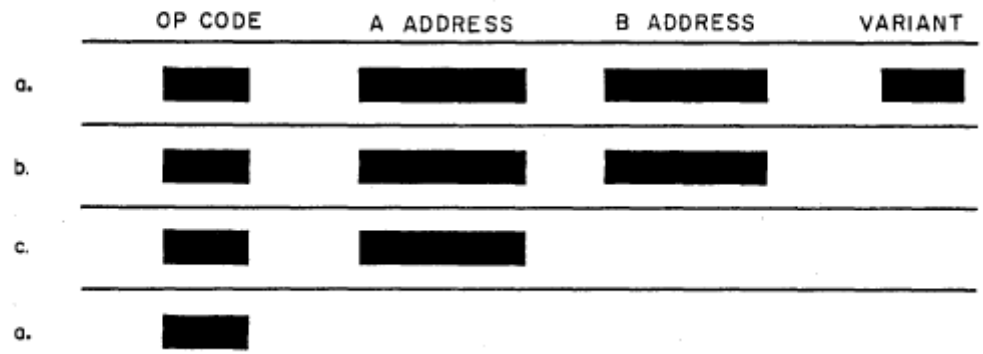
Here is the same instruction typed in differently (3 character mode).

25 110 00 00 00 00 00 01 77 140
A77 000001 (or 0000001 depending on the memory size)
Run
Memory clears to “25”

| | |
|---|---|
| 25 | The character that memory is cleared. In this example an octal “25” which is an “E” |
| 110 | Extended move Instruction |
| 00 00 00 | “A” register (3 character mode) |
| 00 00 01 | “B” register (3 character mode) |
| 77 | Variant for the move instruction. Move A to B, terminate when the first word mark is found in the A field. In this case, the “25” is moved from location 0 to location 1, both counters increment, no word mark is seen in location 1 so location 1 is moved to location 2, etc. |
| 140 | The move instruction needs a word mark to complete the instruction. |

| EXM | EXTENDED MOVE | $10_8$ |
|---|---|---|

FORMAT




FUNCTION

Format a: The contents of the A field are moved to the B field in the manner specified by the variant character (see Table 8-16). The programmer specifies how the move operation is to be performed by selecting the desired conditions from the table and encoding the resulting two octal digits as the variant character of the instruction.

Format b: The contents of the A field are moved to the B field in the manner specified by the variant character of a previous instruction (see Table 8-16).

Format c: The contents of the A field are moved to the field specified by the contents of the B-address register (BAR) in the manner specified by the variant character of a previous instruction (see Table 8-16).

Format d: The contents of the field specified by the contents of the A-address register (AAR) are moved to the field specified by the contents of BAR in the manner specified by the variant character of a previous instruction (see Table 8-16).

A variant of 77 will be seen as a “3X”

| Variant Character (Octal) | Condition |
|---|---|
| X1 | Move A-field data bits to corresponding bit positions in B field. |
| X2 | Move A-field word-mark bits to corresponding bit positions in B field. |
| X3 | Move A-field data and word-mark bits to corresponding bit positions in B field. |
| X4 | Move A-field item-mark bits to corresponding bit positions in B field. |
| X5 | Move A-field data and item-mark bits to corresponding bit positions in B field. |
| X6 | Move A-field word-mark and item-mark bits to corresponding bit positions in B field. |
| X7 | Move A-field data, word-mark and item-mark bits to corresponding bit positions in B field. |
| 0X | Move one character from A to B. The A- and B-address registers are decremented by one. |
| 1X | Move one character from A to B. The A- and B-address registers are incremented by one. |
| 2X | Move characters from right to left (A and B addresses specify rightmost characters in operand fields). Terminate the operation when the first A-field word mark is sensed. |
| 3X | Move characters from left to right (A and B addresses specify leftmost characters in operand fields). Terminate the operation when the first A-field word mark is sensed. |