September 21, 1956

PROJECT STRETCH
LINK COMPUTER MEMO NO. 8

Subject: Storing Part Words in Memory without Destroying the Remainder

By: W. A. Hunt

The problem of partial word extraction from memory is solved in the serial computer by bringing the entire word from memory and doing the extraction at the arithmetic register level. This works fine for reading because the word is restored in memory by the write cycle of memory. However, the insertion of a part of a word of data into a memory location is not so readily facilitated. There are several ways in which it may be done. An attempt will be made to outline some of these below.

1. If the programmer decides to insert information in a word such that part of the information in the word is not destroyed, a special bit could be used to indicate this condition. The store instruction would have the address at which the information would be stored, and this address could be sent to memory and cause the memory read-out cycle to read into the C register. The information to be inserted would then be read directly into the C register in the proper position, destroying only the old information where the new was written in.

   This has the disadvantage that before the logic operation can proceed a memory reference must be made to obtain the data at the address in which the insertion is to be made. This may or may not be a penalty in time since the data from which the information was made had to be originally obtained by a memory reference at the start.

2. A second method is to write a mask at the same time that the data is written into the C register. This mask could be used at either the C register or at the memory level to cause the data to be inserted into the memory word and stored.

The method which I have in mind for #2 is to write a pattern word at the same time at which the data word is formed. This could be under control of a special bit in the store instruction. This pattern word would be either all ones or all zeros, probably all ones, in the locations in which the data word is being formed and would be all zeros in the part of the data word where the information was not to be destroyed in the memory locations. This control store word would be formed at the same time as the data word. An additional shift matrix would probably be required to form this control word. If the store instruction was marked, this control word would be used to combine the data in the memory location with the new data by masking a portion of the memory word and then ORing together the new data with the original data.

3. A third method could be to send the original control information, controlling the forming of the data word to the memory at which the information is to be stored and using it to control the entering of information into the data word. This seems to me needlessly complicated and expensive.

Methods 1 and 2 above require the same amount of transmission over the data buses. That is, they require that either the original data word is brought down to the computer, or the control word be sent to memory. In addition to this transfer, the data word to be stored must be transferred to memory also. That is, two transfers over the data bus for each insertion type of storage. The third method requires that either additional control information be sent to memory via either the address bus or additional lines, or that the data bus be used to send control information.

The second method described has an attractive feature in that the control mask is automatically formed with each data word and it is used or not used according to whether or not the store instructions call for this type of insertion. If it is practical to do the insertion at the memory, the effective storage time need not be appreciably slowed unless the two data transmissions slow the operation.

W. A. Hunt

WAH/jh

W. A. Hunt