HARVEST REPORT #3

Subject: Counting in Memory

By: S. W. Dunwell
W. A. Hunt

Date: September 6, 1956

September 6, 1956

HARVEST REPORT #3

Subject: Counting in Memory

There are numerous applications which require extensive counting followed by some form of statistical evaluation of the counts. It will be a general practice to use memory in lieu of counters for this purpose. In the following pages, a memory word or part of a memory word used as a counter will be called a counter.

Source of Data

The data which will control counting will usually consist of one or several streams of characters feeding from a corresponding number of continuous stream registers into the logical unit, which will arrange the characters as an address for counting. The process by which addresses will be assembled is described in Machine Specification Report #2.

Speed of Operation

The continuous stream registers and the logical unit will provide the addresses at which the counts are to be made at approximately the rate of one each 0.2 microsecond. This is currently assumed to be the highest rate at which data can be transmitted between major units in the machine, such as between the logical unit and a memory.

It is assumed that counting can proceed at this 0.2 microsecond rate in association with either the fast or the large memories of the machine, with the restriction that the flow of information must be interrupted whenever a new count is called for on the part of a memory which has not completed a previous counting operation. The extent to which such interruptions will occur will depend on whether the fast or large memory is being used and upon the number of memory units in the class.

Number of Counters

For most applications, as many counters are required as there

are characters in the alphabet, or some power of this number. In practice, the limitations upon addressing imposed by the direct assembly of characters bits as memory addresses will ordinarily make the number of counters allotted some multiple of a power of two. Thus a count of single decimal digits will occupy 16 counter spaces, regardless of the fact that only 10 possibilities exist. A count of pairs of digits will involve 16^2 counters, etc.

Splitting of Memory Words

Since in most cases the total count accumulated will be very small, provision will be made for carrying more than one count in a single memory word. It will be possible to arrange the memory to contain 1, 2, 4, 8, 16, or 32 counts, with the maximum size of the total determined by the number of counts in the word. If the word size is 64, rather than 60, provision will also be made for 64 single bit counts.

Memory Cycle

The counting process will be performed as a part of the basic memory cycle. The process will be similar to that for normal readout from memory with the exception that the amount returned to memory will be one greater than that which was read out. It is expected that this process will not increase the time required for the operation of either the fast or the large memory.

Counts Greater Than One

While the usual practice will be to add one when counting, it may be equally possible to add any power of two. From the circuit standpoint this would be accomplished by introducing a one at a higher order position of the field being used as a counter.

Overflow

When a memory word is split to contain two or more counts, provision must be made to signal if the section of the memory word has been filled. The signal will indicate that the counter capacity provided is not sufficient for the data under examination, and can be used to call in a subroutine to perform an appropriate modification of the program.

Indicating Existence

There are certain instances in which it is desired to indicate the existence of an item, or the coexistence of two items with the same counter address. In this case, the count can only be zero (indicating nonexistence) or one (indicating existence). If, during a memory reference for this purpose, a zero is encountered it will be increased to one. If a one is encountered, it will be allowed to remain a one and will not be increased.

Statistical Evaluation

In most cases, the counting operation will either be paralleled or followed by a statistical evaluation of the count to determine whether or not a statistical threshold has been exceeded. In many cases, the sum of the squares of the counts is sufficient for this evaluation. Such a sum can be obtained by accumulating a sum of the totals in the counter fields as the counting process proceeds. From a practical standpoint, it makes very little difference whether the total which is accumulated at each step includes or excludes the count being made at the time. To accumulate a sum of products in this way, the count in memory is read out to an accumulator on each counting cycle. The operation very closely parallels that of accumulating a sum of the amounts in a series of memory words, the only difference being that the words are increased by one as each reference is made.

When a more complex evaluation is required, it will sometimes become desirable to perform it concurrently with the counting of the next block of data. In this case, two areas in memory will be set aside for counting, with one being used for counting and the other for evaluation at any given time.

Resetting Memory

At the completion of the statistical evaluation, it is ordinarily necessary to reset all of the memory words serving as counters as a single unit. This must be done as rapidly as possible, since the counting of the next block of data cannot commence until after the entire memory area has been cleared. In order to do this rapidly, special provision must be made in the circuits of the machine to reset specified areas of memory in this way.

For a given machine, several areas of varying size might be specified. The exact size of these areas cannot be determined until more is known about application requirements.

For the purpose of application analysis, it can be assumed that the time required to reset a block of memory will be approximately twice the time required for a full memory cycle, that is 1.0 microsecond for the fast memory or 4.0 microseconds for the large memory.

Machine Specification Report #4 discusses the design problems involved in the development of a block reset feature for memory.

Sum of Items

Nearly all statistical operations require a sum of the items counted to establish a threshold value for evaluation. Since this count must be made in association with each memory reference, it is clear that it should be performed by a transistor counter or a part of the accumulator, rather than by a separate field in memory. The count would include only proper items, and will specifically exclude garbles, spaces, and such other special characters as may be identified as improper for counting by the logical unit or addressing mechanism.

Thinly Populated Areas

It will commonly occur that the count is to be made within a large area which is very thinly populated. For example, one might wish to count a thousand 5-digit groups, which implies a space of the size 16^5. In cases of this kind, it will be undesirable to assign memory space for each possible counter unless it is mandatory to maintain the highest possible speed. As an alternative, the usual practice will be to sort the items instead of counting them, and to combine like control numbers when they are encountered in the sort. This form of counting properly falls under the subject of sorting and will be discussed more fully in that context.

Design

When several counts are being made within each word, the problem

arises of directing the count of one to the proper place within the word. This can be done by locating a one in the desired position of an otherwise unused register, such as possibly the C register, and causing the resulting word to be added to memory. If this cannot readily be done, or if doing so causes undue traffic over the data bus, it may be necessary instead to send the address of the one directly to the memory unit and to provide the necessary decoding and distribution system in each memory. The problem of handling counts greater than 1 is similar, except that it implies some form of control over the value being entered beyond the control which selects the counter address. The wisdom of such a control is not clear at this time.

Since the counter being increased must send out to the accumulator to obtain a sum of squares, the field can at the same time be compared with a series of 1's the same length as the counters to determine whether the total (after the count) has reached its maximum limit. If so, a signal must be given to a selector to permit the program to be interrupted for a special overflow routine. A more general solution would be to set up 1's in a register in the positions corresponding to the high order position of each counter. The existence of a 1 in any of these positions of the word being read out would signal an overflow. The maximum count obtainable with this system would be less, but in association with entry of the 1 from the C register, it would constitute a relatively universal system in which the counters could individually vary in size under program control.

W. A. Hunt S. W. Dunwell

jh