Griffith

November 5, 1956
H. C. Montgomery

Delta Computer Memo No. 16
Subject: Editing for Printing

Summary

A general problem in printing is defined and a system described for solving the problem. A typical editing problem is solved, illustrating the use of the new system.

Given.

A body of data is stored in consecutive memory locations in the order in which they are to be printed. These data are in the code used internally by the machine *, but are compressed so that punctuation and spacing are needed, and they may contain characters which we should like to have deleted or replaced with some other given characters.

The Problem is to operate on these data in a manner such that the results of the operation when presented to the printer will cause the information to be printed in a desired form.

Assumptions

1. The data are operated on from left to right in the ordinary sense of reading.

2. All arithmetic operations on the data have been completed previously.

3. The lengths of the print line and subfields within the line are known.

4. For a given character of information data, we know the subfield within the print line to which the character belongs.

Analysis

We separate the given problem into the following specific problems:

1. Generating and inserting blanks in specified positions.

2. Generating and inserting arbitrary characters at desired places.

3. Detecting the existence of and substituting other characters for non-significant zeros.

4. Deleting characters from the data to be edited.

* This restriction may be removed to permit an arbitrary code's being edited if one is willing to add to the complexity of the machine by giving it the facility for recognizing arbitrary bit configurations when operating in the non-significant zero mode and under control of certain other special instructions.

The reasons for listing separately problems which are logically similar is that they are treated differently. They are treated differently because the frequency of their occurrence in general editing practice varies.

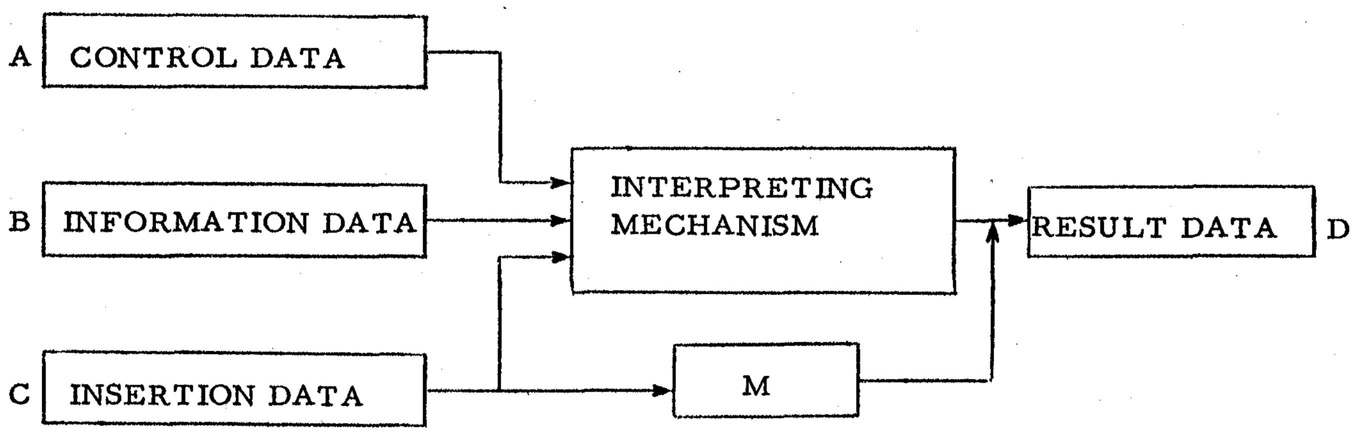
A Solution

The distinguishing feature of the system we shall describe is embodied in the concept connoted by the term "streaming" as applied to the transfer of data within a computer. The difference between this and the normal method of data transfer among registers is essentially a higher degree of automatic performance of the function of supplying the logical registers with the data they use, and in carrying away the results.

In this system all communication between the registers involved and memory takes place automatically in whole words once the procedure begins. That is, those registers which receive data from memory are automatically given the next word when their present contents are exhausted; and the register which sends data to memory automatically sends its contents to consecutive memory locations when it becomes full. In both cases the memory locations used for the data are consecutively addressed and preserve the order of the data as they will appear in the print line. We shall think of memory location addresses as being in ascending order according to the sequence in which they are used, although the opposite convention would work as well.

Hence, in what follows, when a stream is mentioned, the data identified with the stream will be understood to flow through its appointed registers without the programmer's having to provide for communication between these registers and memory on a word-for-word basis.

Consider the system -



The Registers

A. Through Register A passes the control stream which goes in two bit bytes to the interpreting mechanism which interprets these bytes according to the following code:

11 means put the next byte from B into the next byte position of D; leave C unchanged.
10 means put the next byte from C into the next byte position of D; leave B unchanged.
01 means take the next byte from C and interpret it as a special instruction.
00 means skip the next byte position of D; leave B and C unchanged.

At the end of each sub-cycle of editing operations another two bit control byte from A is automatically sent to the interpreting mechanism to instruct what operations are to be performed during the next sub-cycle.

B. Register B gets the data to be edited from memory a word at a time, and exactly as it appears in memory. Upon command from the control stream via the interpreting mechanism, these data flow a byte at a time into D.

C. Through C flows the stream of punctuation characters which is constructed beforehand to present these characters in the order of their use. This stream contains all those characters to be inserted except the blank; and the multiplicity of a character in the insertion stream for a given print line equals the multiplicity of the character's appearance in the line, with the exception of those characters inserted by a special instruction called in by the control byte 01.

This stream also contains the above-mentioned special instructions, examples of which will be given below.

D. Into D flows a byte at a time the data in its final output form. If the code used is not such that the registers are cleared to blanks, then D is wired so that it is automatically filled with blanks immediately following those times when it has sent its contents to memory.

M. M is an eight bit capacity register which operates only when specifically commanded to do so by a special instruction. Its main responsibility here is to act as the source for whatever character is to be substituted for non-significant zeros and commas. As will be seen, it also serves a number of other useful purposes.

A property common to all of the registers we have described is that they are capable of operating in modes using different byte lengths in order to use with facility a number of different codes. When a particular code is agreed upon, the most suitable byte length will be known so that when the instruction is given to initiate the editing procedure, one will also provide the instruction which tells the registers involved how many bits they are to use for each byte.

A consideration of the proposed code for the control stream, the specific problems listed under the analysis above, and a reflection on some other editing functions one may wish to perform suggest that among the library of special instructions to be defined the following ones will prove convenient:

Special Instruction One (S 11)

Put the next byte from C into M. Now continue executing the commands from the control stream; but before permitting any characters to enter D examine them and replace all zeros and commas with the contents of M. Revert to the normal procedure when the first significant character is found.

Special Instruction Two (S 12)

Put the next byte C into M. Now continue executing the commands from the control stream, suppressing all zeros and commas until a significant character is encountered. Insert the character in M immediately before this first significant character and then revert to the normal procedure.

Special Instruction Three (S 13)

Omit the next byte in B; do not advance C or D.

Special Instruction Four (S 14)

Put the next byte from C into M. Before executing the next control instruction advance D the number of bytes shown in M.

Example

Suppose we have in memory the data

36594WOOLENbTROUSERSb3303800175000067100

and suppose we want to print it in the form

ITEM NO.	DESCRIPTION	SIZE		QU	UNIT PRICE	EXTENSION
36594	WOOLENbTROUSERSb	33		38	17.50	$671.00
	s	s	sbbbbbbbbsb		sbb	sbbbb

The control stream and insertion stream which are required are given below. Directly under the code characters are given the familiar characters to which they correspond.

Control stream:

```
11 11 11 11 11 00 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11
3  6  5  9  4  b  W  0  0  L  E  N  b  T  R  O  U  S  E  R  S

11 00 11 11 00 01 00 01 11 11 11 00 01 11 10 11 11 11 10 11 11
b  b  3  3  b     b     b  3  8  b     b  b  b  1  7  .  5  0

00 01 11 10 11 11 11 10 11 11 11 10 11 11
b     b  b  b  b  b  b $6  7  1  .  5  0
```

Insertion stream:

SI4 8 SI1 b SI1 b , SI2 $, , .