Logical Coding

(Flow charting) by Mrs. Jean J. Bartik
V.H. (I don't know)
Miss Francis E. Snyder

For some time, various groups working with digital computers have discussed the type of picture to make to represent the logical operations necessary to solve a problem. This picture has been called at various times flow chart, problem block diagram, problem set-up, and so forth, and different groups have evolved pictures of different forms. No difficulty arises in interpreting a picture as long as it is kept within one group; however, when two or more groups are interchanging their pictures, the interpretation becomes difficult. Therefore, it seems feasible that some standard should be adopted for groups that are working together. The Applications Group has agreed on the following standard which is based on the flow chart described by von Neumann and Goldstine in "Planning and Coding of Problems for an Electronic Computing Instrument."

LOGICAL CODING

A flow chart makes use of the following symbols

$\longrightarrow$ = direction of flow.

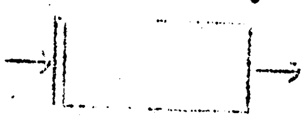
 = an operation box.

An operation box contains the operation or operations to be performed on data in the computer, exclusive of comparisons.

Example:

$$\rightarrow \boxed{x + y = z \rightarrow A_0} \rightarrow$$

This means that x is added to y to produce z, which is stored in memory location A_0. Here A merely specifies some block of memory locations in the computer, and A_0 is not given a specific memory location except in relation to other numbers stored in section A. A_0 means the first memory location in Section A. This generality is to the programmer's advantage, for he usually doesnot know what memory locations to assign until the whole problem has been planned. The generality is also useful if the routine is to be inserted in different problems, and the memory locations vary from problem to problem. If the routine is not to be used elsewhere, specific memory locations can be assigned after the flow chart has been drawn and checked. It is much easier to change a memory location on the flow chart than in the coded problem, for the insertion or deletion of a number in the coded problem may mean offsetting all of the following instructions or numbers by one.

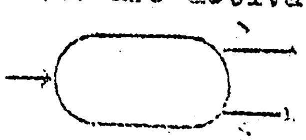
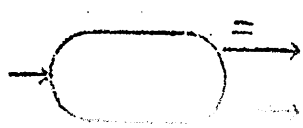
 = a substitution box.

A substitution box contains the operation or operations required to prepare the control before entering a routine or to reset the control after completing a routine.

Example: Extending the example above, assume that there are a series of x's and y's stored in the memory to be added to produce z's. The z's will be stored in section A of the memory. For this example, assume that one x will be added to one y to produce one z which is stored in one memory location of section A.

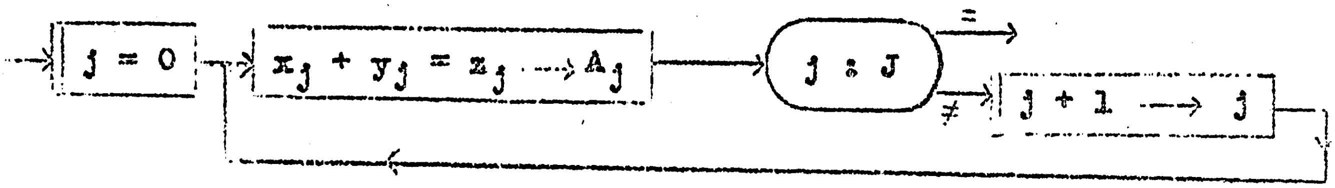
$$\rightarrow \boxed{j = 0} \rightarrow \boxed{x_j + y_j = z_j \rightarrow A_j} \rightarrow$$

This means that x_0 is added to y_0 to produce z_0 which is stored in memory location A_0. A record should be kept of the sectional storage of x and y. "j" is used as a control to choose which memory locations are activated.

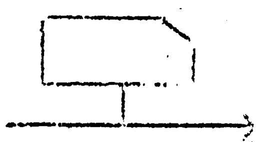
 or  = a decision box

A decision box contains the two quantities on which the decision is based through the use of the test, T, or equality, Q, orders. The decision consists of choosing one of two paths depending on the relative size of the two quantities designated in the decision box. Decisions can be made either on control or data. In the decision box, the quantity stored in the accumulator is written first and the quantity in the L register second. The two are separated by a colon, :.

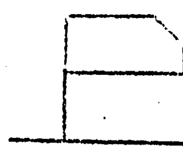
Example: Extending the example above assume that one wishes to add all of the x's and y's to produce all of the z's which will be stored in the A section of the memory.



This means that the series of x's and y's beginning with x_0 and y_0 and ending with x_J and y_J are added to produce $z_0 \ldots z_J$, which are stored in memory locations $A_0 \ldots A_J$. Here "j" may or may not be stored as a counter. Probably in this case it would not be a counter. It is probably one of the instructions used to perform one of the operations specified in the operation box. "J" must be of the same form as "j". Note that when one is added to "j" or when "j" is set to zero, three instructions must be changed: (1) the instruction that adds x to the accumulator; (2) the instruction that adds y to the accumulator; (3) the instruction that clears z to A.



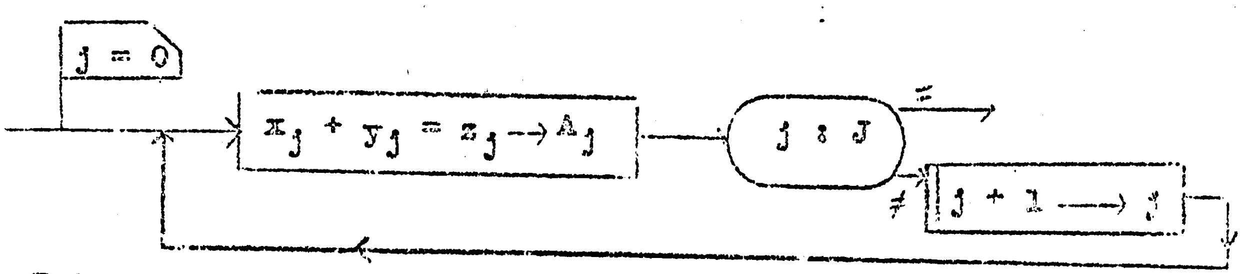
or



= an assertion box.

An assertion box is used as a signpost or reminder. It does not indicate that any operations are to be done in the computer. It is merely a statement that at this time, this is so. It is helpful in stating the necessary status of control and counters upon entering a routine.

Example: Take the last example and draw the flow chart using an assertion box for the first substitution box.

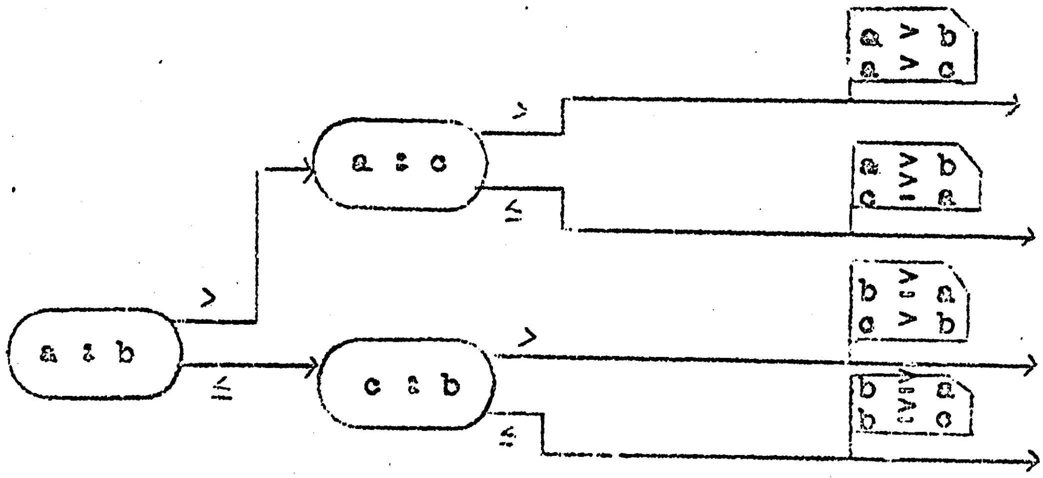


This means that the initial state of the instructions was such that "j" = "0". Here "j" does not have to be set to "0" since it is

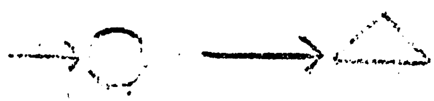
LOGICAL CODING

The assertion box is also used to state what has been proved after a series of decisionshave been made through the use of the T or Q orders.

Example: Suppose that it is desirable to determine which is the largest number contained in three quantities, a, b, and c.



Here the assertion boxes state what was learned from the preceding sequence of operations.

Connectors: 

A connector represented by a circle, indicates a connection which can be found on the immediate flow chart.

A connector represented by a triangle, indicates a connection which can be found on some other flow chart or page and must designate to which flow chart or page the reference is made.

1. Remote Connector. Remote connector take the place of drawing long lines across a flow chart. When a connector is merely the continuation of the flow chart, a Greek letter is inserted in the circle → α. Since most people prefer to read from left to right or from top to bottom the remote connector can be used to get back to the left or top after reaching the right or the bottom of the chart. A remote connector containing a Greek letter does not mean a computer operation.

A remote connector may indicate a routine which can be entered from several points through a conditional or unconditional transfer of control. When such is the case, either a number or a letter is inserted in the circle → 1, → P . It has been suggested to reserve certain letters for routine common to all programs

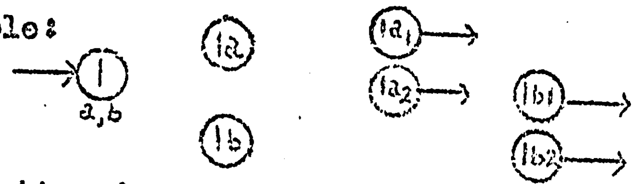
→ P Print out routine

→ S Sentinel routine

LOGICAL CODING

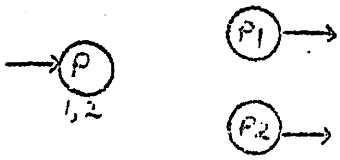
2. Variable Connector. A variable connector indicates that the out-put from a routine is not static, but is variable. The output is dependent upon the input to the routine. In general a variable connector is a number or a letter or a group of letters. When a number is used the first variation is a small letter; the second variation is a number, and any other variations alternate a letter with a number.

Example:



When a letter is used the alternation is a number followed by a letter.

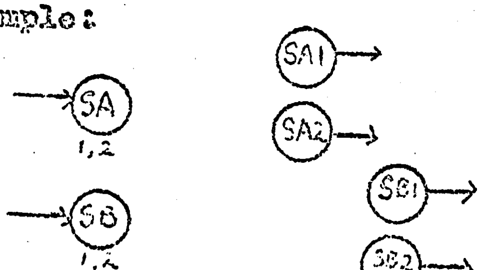
Example:



This example would take care of an output print routine where two output tapes are used as alternates.

When a subroutine contains the same logic as another subroutine in the problem but does not necessarily commence from the same point or exit to the same point a double capital letter system may be used.

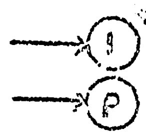
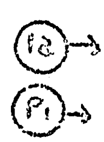
Example:



This example would take care of testing for end of two input tapes A and B where the logical procedure is the same but where the actual coding is done in duplicate because the routine commences from different conditions.

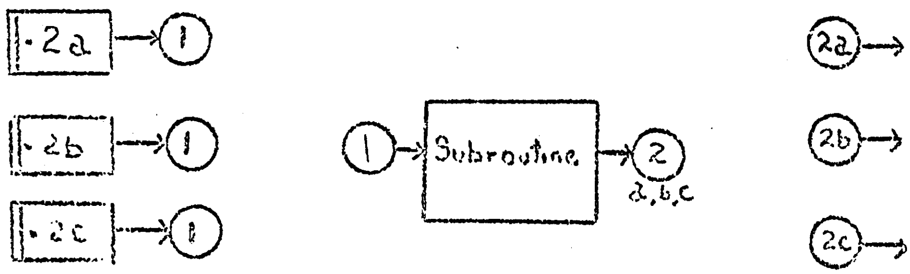
The variations are listed under the connector as numbers or letters.

The outlet of a variable connector is changed in a substitution box thus:

-1a means that  will now go to 
-P1 " " " " " "

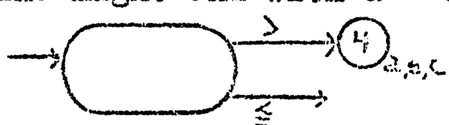
LOGICAL CODING

Example: Suppose that a subroutine is entered from three different points in the main routine. Each time the subroutine is done a new sequence is initiated in the main routine.

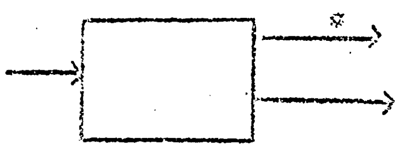


This means that the exit point of the subroutine has been set before entering the subroutine from one of the three different channels. This output could be set by transferring a stored constant to the output position of the subroutine in the form of a transfer of control instruction. It could be set also by the special instruction that makes an unconditional transfor instruction from the setting of the control counter plus one (Rm)

The subroutine might end with a T or a Q instruction.



Here the appropriate T instruction could be transferred into the subroutine or the appropriate memory location could be inserted into a constant T instruction.



= automatic overflow indication.

Since the Univac makes an automatic decision when an overflow occurs in the accumulator, it can not be programmed in a decision box. The addition or subtraction that caused the overflow would occur in an operation box. The unstarred output indicates the direction of flow if an overflow occurs. Since the computer goes to a fixed position 0000, if an overflow occurs, it is necessary to set the 0000 position to transfer to the appropriate routine. This should be done through a substitution box that precedes the operation box that gives rise to an accumulator overflow.

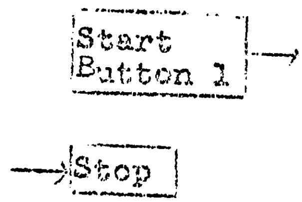
The general specification of memory locations has been discussed above. At times it is desirable to operate on the contents of these memory locations. To differentiate between the two, a parenthesis, ( ), is used to mean ' contents of'.

A_0 means the first memory location in section A.

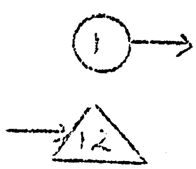
(A_0) means the contents of the first memory location in Section A.

LOGICAL CODING

All complete problems start with the initial start button which brings in 60 words of instruction from a tape and ends with a stop instruction. It has been proposed to show these operations on the flow chart as:



When a flow chart represents a subroutine the input and output are connectors.



input

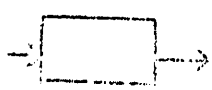
output

After the flow chart has been completed, it is desirable to tie it up with the coded instructions that accomplish the solution of the problem. This is done by labeling the boxes with the starting memory location taken from the coding sheets.

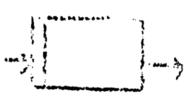
Variations of and additions to the flow chart described above may seem feasible as more people use it.

Summary:

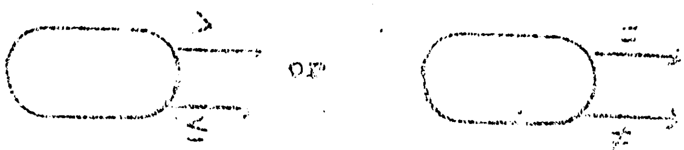
= direction of flow.



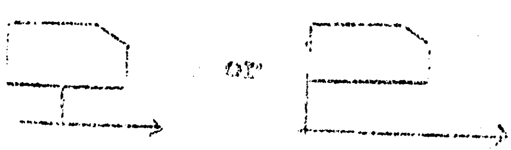
= an operation box.



= a substitution box



= a decision box



= an assertion box.

LOGICAL CODING

$\longrightarrow (\alpha)$ = a remote connector

If n = 2, $\longrightarrow (1)_{1,1}$ = a variable connector

$\boxed{\text{Start Button}\ 1} \longrightarrow$ = input

A number is inserted in the Start Button Box to denote which tape contains the instruction

$\longrightarrow \boxed{\text{Stop}}$ = end

( ) = contents of.

$\xrightarrow{*}$ = accumulator overflow indication.

The English alphabet is used to provide symbols for the control and data in the computer.

The Greek alphabet is used to label remote connectors.

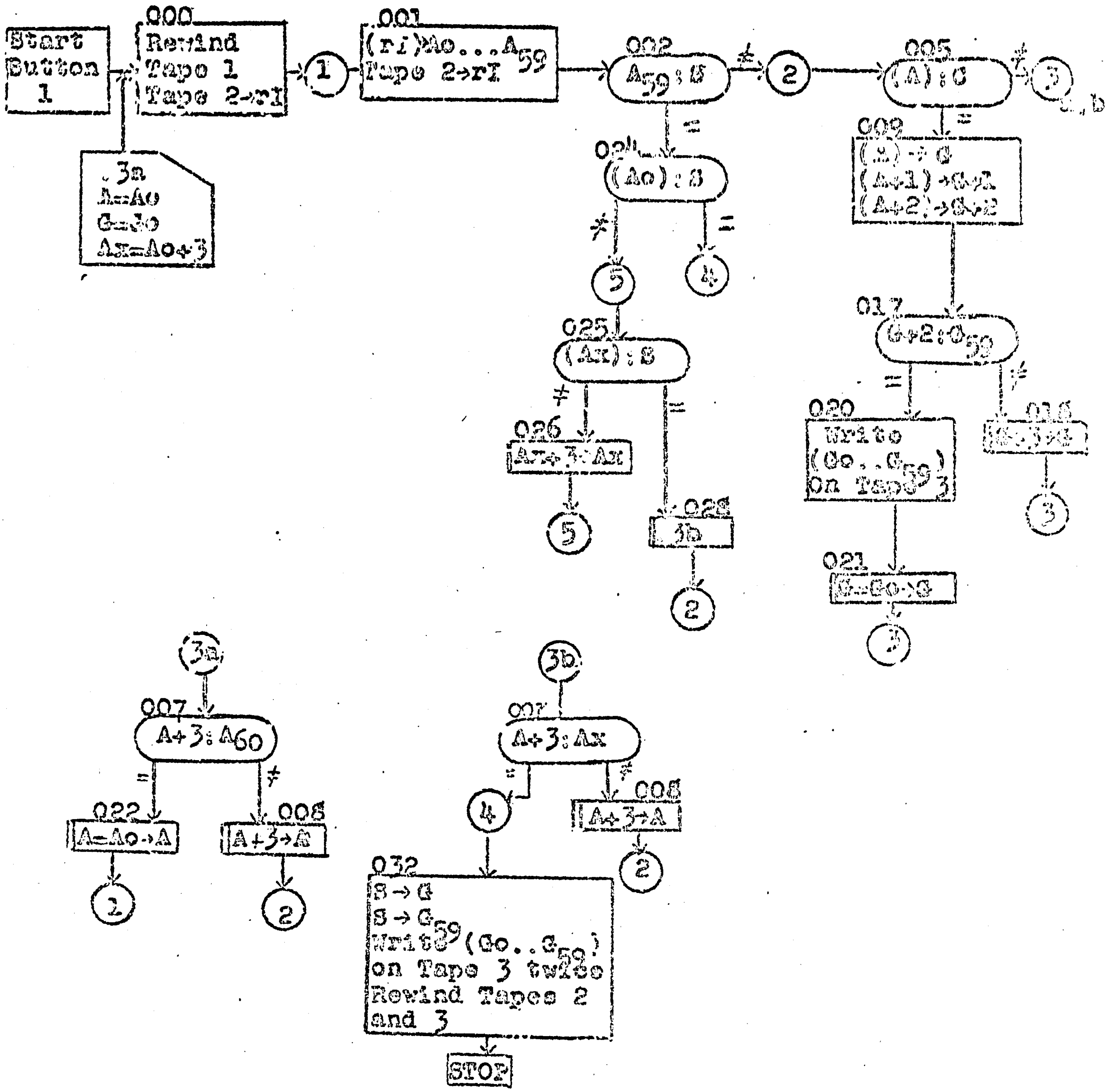
Memory location numbers are used to correlate the flow chart with the coded instructions.

Arabic numerals have miscellaneous uses, such as subscripts, superscripts or connectors

JJB
VH
FES
7/1/49
sr 7/5/49

Code C-10
6/20/49

SELECTION OF 3 WORD ITEMS ON 2 KEY DIGITS TO BE PLACED ON A SEPARATE TAPE



LEGEND

Ao...A59	Stores input block
Go...G59	Stores output block
G	Current position in output block
A	Current position in input block
S	End of tape sentinel
C	Code of key digits being isolated
Ax	Location of last item on input tape
Tape 1	Contains instructions
Tape 2	Contains input data
Tape 3	Contains output data

Selection of 3-Word Items on 2 Key digits
To be placed on a Separate Tape

Eg: (one month out of 12)

Memory: 0-52 instruction, 120-179 input data, 180-239 output da

000	81000		Rewind tape 1	
		12000	Input data to rI	
001	32120		Data to 120...179 = A_0...A59	1
		B 179	(A59)	
002	L 046		S	
		Q 024	Compare for end of input tape (A59):S	
003	K 000			
		F 042	Digit extractor	
004	E(120)			
		L 038		2
005	00000			
		Q 009	Compare (A) and C	
006	B 004			
		A 039	A+3	3
007	L 048			
		Q 022	Compare A+3: A60 for end of input block →	3a
#007	L 048	Q 032	Compare A+3: A_x for end of all input data →	4 3b
008	C 004		A+3 to A	
		U 004		
009	F 043		Instruction Extractor	
		B 004		
010	E 049			
		H 013	Puts A in instruction B000 C(G)	
011	A 040			
		H 014	Prepares instruction (A+1) to G+1	
012	X 000			
		C 015	Prepares instruction (A+2) to G+2	
013	[B(120)			
		C(180)]	(A) to G	
014	[B 121			
		C(181)]		
015	[B(122)			
		C(182)]		
016	E 015			
		L 051		
017	F 042		Digit extractor	
		Q 020	Compare G+2 and G59 for end of output block	
018	A 041			
		C 049	G+3 to G	
019	00000			
		U 006	Return to main sequence →	3
020	53180		Write (G_0...G59) to tape no. 5	
		B 050		
021	C 049		G = Go to G	
		U 006		

Selection of 3 Word Item

```
022     B 047
                  C 004      A = A0 to A                                  2a
023     00000
                  U 001
-------------------------------------------------------------------------------
024     B 120
                  Q 032      Compare A0 to S
-------------------------------------------------------------------------------
025     B(123)               Ax
                  Q 028      Compare Ax to S                              5
026     B 025
                  A 039
027     C 025                Ax + 3 --> Ax
                  U 025      --> 5
-------------------------------------------------------------------------------
028     B 025                Ax
                  F 043      Instruction extractor
029     E 048
                  C 048      E (Ax) Lo38 to o48
030     E 052
                  C 007      3b to 3
031     F 042
                  U 004      --> 2
-------------------------------------------------------------------------------
032     B 045                                                             [illegible]
                  F 044
033     E 049                Build up instruction to position end of output
                  C 035      tape sentinel
034     A 046
                  S2000      Rewind tape 2
035   [ H 239
                  C(180) ]   Write sentinel after last output item and end of
                             block
036     53180                Write last block of information on tape 3
                  53180      Write one additional block; (2nd sentinel block)
037     S3000                Rewind Tape 3
                  90000      Stop
-------------------------------------------------------------------------------
038                          C code of key digits to be isolated
039     000003    000000
040     000001    000001
041     000000    000001
042     Digit extractor
043     110000    111111
044     000000    111111
045     H 239     000000
046                          S = End of tape sentinel
047     E 120     L 038
048    (E 180     L 038)
049     B 000     C(180)
050     B 000     C 180
051     B 000     C 239
052     L 048     Q 032
```

coded: F.E.S. 6/21/49
checked [illegible] 6/21/49