return to Kolsky

Poughkeepsie, New York
December 19, 1955

PROJECT STRETCH
FILE MEMO #17

COMPANY CONFIDENTIAL

Note: The reader is advised to read Stretch Memo #16 before this memo. A knowledge of the registers used in the Transform Command is assumed in this memo.

SUBJECT: Editing, Part II
By: E.M. Boehm, J.E. Griffith

Execution of a Transform command initiates a series of memory references for "table words". "Table words" control the execution of the transformation. There are multiple types of table words; and in general, a table word tells where the data for the output record is to be taken from, how much information is to be put in the output record, the position of the next control character in the input record, and what action should be taken next. The various forms of table words are listed in Figure #1.

Data for the output record can be obtained from the table word, the input record or from memory. Data entering the record from a table word is sent through an intermediate register (W register), and may be algebraically or logically added to earlier data in the W register or be inserted directly into the W register. When a table word is form #1. 4. or 11, data is algebraically added to the contents of the W register. Logical addition to the contents of the W register is specified by form #2, 5, and 12. If form #3, 6, 13, 28, 29 or 30 is executed, data is loaded into the W register, and any previous data in that register is lost. A direct or indirect reference may be made to memory for data to enter the output record. Direct reference is called for in forms 7, 8, and 17 while forms 9, 10 and 18 specify an indirect reference. The address of a memory reference for data may be modified by the contents of the control accumulator. No address modification is permitted when the memory reference is for a table word or for a word containing the address of data.

In the Transform execution word, a control bit specifies whether address modification is possible while the instruction is being executed. If indexing is to be used, the index register whose contents are to be loaded into the control accumulator, is specified in the execution word. It is not necessary for every data reference to be index, and a control bit in forms 7, 10, 17, and 18 is used to call for indexing (C bit in table word). A C bit in the table word is effective only if the Transform execution word has specified indexing and an index register. When address modification is called for, the index register number in the control accumulator is compared with the index number in the execution word. If the two numbers do not match, a memory cycle is captured to load the proper index register into the control accumulator before the data reference is made. Control table words #34 and 35 enable the index register number to be changed while transformation is in process. In these two forms the new index number replaces the index number in the execution word.

Data from a memory location enters the output record through the W register and erases any information in that register. In addition, data may be taken into the output record from the input record held in the I register (forms 14-16). After 60 bits of information have been inserted into the O register from the W register, a memory cycle is captured to write the contents of the O register into the memory location specified by L (O) in the D register. The O register is set to a one when its contents are stored. L (O) in the D register is increased by one. An input word in the I register may be rotated continuously or be replaced by the next word in the input record after the 60 bits have been utilized. Operation of the I register is under the control of the execution word. When automatic reloading is specified, a memory cycle will be captured after 60 bits have been used, to load the contents of the memory location, specified by L (I) in the D register, into the I register. L (I) in the D register will be increased by 1.

Field N in a table word is used to control the amount of data taken from the selected source into the output record. In forms 1-6 and 11-13, all 24 bits from the table word (the Z field) enter the W register, and N determines how many of the bits are sent from the W register into the O register. In form 28-30, the number of bits to be taken from Z in the table word are specified by the R field. N is used in form 28 to determine how many times the character defined by R enters the O register. A character in the Z word will be entered into the O register until L (O) reaches a specified value with forms 29 and 30. Field X in #29 specifies the desired value of L (O). With #30, X gives the memory location which contains the desired value of L (O). Data taken from memory or from the input record enters the W register and goes immediately to the O register. Field N specifies how many bits are to enter the O register. If N is larger than 60 or the number of bits in the I register, a memory reference is made for the next memory location (goes to W register) or word in the input record (goes to I register). A maximum of 511 bits may be specified by a table word. When information is taken from the input record M is used to specify the number of bits that are to be shifted in the input record before the bits commence to enter the output record. When bits are brought from memory by an indirect reference, a maximum of 7 bits may be ignored before starting to enter the output record. The ignored bits are specified by field P in form #9, 10 and 18.

After data has been shifted from the I register to the output record, the bits following the shifted data are available for transformation control. Bits shifted into the output record cannot be used to control transformation. The execution word of a Transform instruction specifies the size of a character in the input record. Control of the input record through field M is given in terms of characters, not bits. Forms which take information from the input record to the output record use M to determine the number of characters ignored before sending data to the output record. When data is sent to the output re ord from memory or the table word. M specifies how many characters are to be shifted before getting the next table word. Forms 19-27 causes the contents of the I register to be shifted a specified number of times (19-21) or until L (I) in the D register reaches a set amount with 22-24, L(I) is to reach the value given in Y of the table word. Y in forms 25-27 is the location in memory that contains the desired value for L (I). To increase the flexibility of the Transform command, 32-34 permit the size of a character in the I register to be changed to the amount specified by B. A character may range in size from 1 to 15 bits.

A fourth value included in the table word is an indication of the next operation to be carried out. The Transform command may proceed or can be halted. Look up is ended by several of the table word forms or by having N in the execution word reach zero. Each time a character in the I register is used, N is reduced by one. When N reaches zero the address in the execution word is made zero. Transfer of control cannot be executed as part of the transformation. Provision for I/O break-in which immediately captures the program counter bars the inclusion of transfer. Control transfer is facilitated by using the address part of the execution word to hold the desired transfer address. After the transformation has ended, the contents of the execution word address can be tested. If the field is blank, the calculator proceeds to the next instruction. If the field is not blank, control is transferred to the address specified in the field.

When the transform instruction is to be continued, the word form tells whether the reference address for the next table word is to be altered. In the execution word, a reference origin is given which may be used throughout the entire instruction or which may be changed at will. If the reference address is to be changed, the new address is given in the table, word. A reference address for a table word may be modified by the character in the I word. When the execution word or a table word specifies an unconditional reference, the table origin address is used as the address of the table word. When a conditional reference is callef for, the origin address is added to the character in the I register whose size is specified in the execution word or by table word for 31-33. The execution word specifies whetherthis character should be taken from the high or low order position of the I register.

The number of table word forms will be increased at a later date to increase the flexibility of the Transform instruction.

E. M. Boehm
J. E. Griffith

1. (Z)+C (W) → W, Table origin = Ot, N = no. of bits from W→0, M = no. of characters shifted in I
   a) Next lookup = Ot + character in I
   b) Next lookup = Ot
2. (Z)v C(W) → W, Table origin = Ot, N = no. of bits from W→0, M = no. of characters shifted in I
   a) Next lookup = Ot + character in I
   b) Next lookup = Ot
3. (Z)→W, Table origin = Ot, N = no. of bits from W→0, M = no. of characters shifted in I
   a) Next lookup = Ot + character in I
   b) Next lookup = Ot
4. (Z)+C(W)→W, End lookup with transfer address = X, N = no. of bits from W→0, M = no. of characters shifted in I.
5. (Z) v C(W)→W, End lookup with transfer address = X, N = no. of bits from W→0, M = no. of characters shifted in I
6. (Z)→W, End lookup with transfer address = X, N = no. of bits from W→0, M = no. of characters shifted in I
7. C(Y) → 0, Table origin = Ot, N = no. of bits from loc. y→0, M = no. of characters shifted in I
   a) Next lookup = Ot + character in I   c = 1, modify y by contents of control Acc.
   b) Next lookup = Ot   c = 0, do not index by contents of control Acc.
8. C (Y) → 0, End lookup with transfer address = X, N = no. of bits from loc. y → 0, M = no. of characters shifted in I
9. C (C(Y)) → 0, Table origin = Ot, N = no. of bits from loc C(Y) → 0, M = no. of characters shifted in I
   a) Next lookup = Ot + character in I   p = bits shifted in loc. C(Y) before entering 0.
   b) Next lookup = Ot
10. C (C(Y))→0, End lookup with transfer address = X, N = no. bits from loc C (Y)→C, M = no. characters shifted in I
    p = bits shifted in loc. C(Y) before entering 0.
11. (Z)+C(W)→W, Leave Table origin unchanged, N = no. bits from W→0, M = no. characters shifted in I
    a) Next lookup = Ot+1 + character in I
    b) Next lookup = Ot+1
12. (Z) v C(W) → W, Leave Table origin unchanged, N = no. bits from W→0, M = no. characters shifted in I
    a) Next lookup = Ot-1 + character in I
    b) Next lookup = Ot-1
13. (Z)→W, Leave Table origin unchanged, N = no. bits from W → 0, M = no. of characters shifted in I
    a) Next lookup = Ot-1 + character in I
    b) Next lookup = Ot-1
14. C(I) → 0, Table origin O_2, N = no. bits from I → 0, M = no. characters shifted in I before entering 0.
    a) Next lookup = Ot + character in I
    b) Next lookup = Ot
15. C (I)→0, End lookup with address = X, N = no. bits from I→0, M = no. characters shifted in I before entering 0.
16. C (I) →0, Leave table origin unchanged, N = no. bits from I→0, M = no. characters shifted in I before entering 0.
    a) Next lookup = Ot-1 character in I
    b) Next lookup = Ot-1
17. C(Y) → 0, Leave lookup origin unchanged, N = no. of bits from loc y → 0, M = no. characters shifted in I
    a) Next lookup = Ot-1 character in I
    b) Next lookup = Ot-1

Form no.	Control								
1.	a) 10 b) 11	M (7 bits)	N (7 bits)			Z (24 bits)		Ot (18 bits)	
2.	a) 100 b) 101	M (7 bits)	N (7 bits)			Z (24 bits)		Ot (18 bits)	
3.	a) 110 b) 111	M (7 bits)	N (7 bits)			Z (24 bits)		Ot (18 bits)	
4.	1000	M (7 bits)	N (7 bits)			Z (24 bits)		X (18 bits)	
5.	1001	M (7 bits)	N (7 bits)			Z (24 bits)		X (18 bits)	
6.	1010	M (7 bits)	N (7 bits)			Z (24 bits)		X (18 bits)	
7.	a) 1100 b) 1100	M (7 bits)	N (9 bits)	c	001 010	Y (18 bits)		Ot (18 bits)	
8.	1100	M (7)	N (9)	c	011	Y (18)		X (18)	
9.	a) 1110 b) 1111	M (7)	N (9)	c	P (3)	Y (18)		Ot (18)	
10.	1101	M (7)	N (9)	c	P (3)	Y (18)		Ot (18)	
11.	a) 0000 b) 0001	M (7)	N (7)			Z (24)			00000001
12.	a) 0000 b) 0001	M (7)	N (7)			Z (24)			00000010
13.	a) 0000 b) 0001	M (7)	N (7)			Z (24)			00000011
14.	a) 0000 b) 0001	M (7)	N (9)	-		Ot (18)			00000[illegible]
15.	0000	M (7)	N (9)	-		X (18)			00000111
16.	a) 0000 b) 0001	M (7)	N (9)	-					00001000
17.	a) 0000 b) 0001	M (7)	N (9)	c		Y (18)			00000100

18\. C (C(Y))→0, Leave lookup origin unchanged, N= no. of bits from loc. C(Y)→0, M= no. characters shifted in I, P= no. bits shifted in loc. C (Y) before entering 0
a) Next lookup = Ot-1 + character in I
b) Next lookup = Ot-1

19\. Shift I register, M= no. characters to shift in I, Ot = table origin
a) Next lookup = Ot + character in I
b) Next lookup = Ot

20\. Shift I register, M = no. characters to shift in I, End lookup with transfer Address = X

21\. Shift I register, M = no. characters to shift in I, Leave table origin unchanged
a) Next lookup = Ot-1 + character in I
b) Next lookup = Ot-1

22\. Shift I register until L (I) = X, Ot = table origin
a) Next lookup = Ot + character in I
b) Next lookup = Ot

23\. Shift I register until L (I) = X, End operation with transfer address = X

24\. Shift I register until L (I) = Y, Leave table origin unchanged
a) Next lookup = Ot-1 + character in I
b) Next lookup = Ot-1

25\. Shift I register until L(I) = C(Y). Ot = new table origin
a) Next lookup = Ot + character in I
b) Next lookup = Ot

26\. Shift I register until L (I) = C (Y). End operation with transfer address = X

27\. Shift I register until L (I) = C (Y). Leave table origin unchanged
a) Next lookup = Ot-1 + character in I
b) Next lookup = Ot-1

28\. Z→0, R = no. bits of Z to use, N = no. time R bits shifted into 0, M = no. characters in I to shift, End operation

29\. Z→0, R = no. bits of Z to use, shift R bits until L (0) = X, End Operation

30\. Z→0, R = no. bits of Z to use, shift until L (0) = C (X), End operation

31\. Alter size of character size in I register to M
Ot = new table origin
a) Next lookup = Ot + character in I
b) Next lookup = Ot

32\. Change character size of I register to B, new table origin = Ot
a) Next lookup = Ot + character in I
b) Next lookup = Ot

33\. Change character size of I reg to B, use same origin
a) Next lookup = Ot-1 + character in I
b) Next lookup = Ot-1

34\. Change index tag to equal H, new table origin = Ot
a) Next lookup = Ot + character in I
b) Next lookup = Ot

35\. Change index tag to equal H, use same origin
a) Next lookup = Ot-1 + character in I
b) Next lookup = Ot-1

Note:
Table forms where data is brought from a memory location may be indexed under the control of the C bit.

Form Control

No.									
18.	a) 0000 b) 0001	M (7)	N (9)	c P (3)	Y (18)				00000101
19.	a) 0000 b) 0001	M (7)			Ot (18)				00001001
20.	0000	M (7)			X (18)				00001010
21.	a) 0000 b) 0001	M (7)							00001011
22.	a) 0000 b) 0001		Y (18)	c	Ot (18)				00001100
23.	0000		Y (18)	c	X (18)				00001101
24.	a) 0000 b) 0001		Y (18)	c					00001110
25.	a) 0000 b) 0001		Y (18)	c	Ot (18)				00001111
26.	0000		Y (18)	c	X (18)				00010000
27.	a) 0000 b) 0001		Y (18)	c					00010001
28.	0000	M (7)	N (7)		Z (24)	R (5)			00010010
29.	0000		X.16 (14)		Z (24)	R (5)	c	X (4)	00010011
30.	0000		X.16 (14)		Z (24)	R (5)	c	X (4)	00010100
31.	a) 0000 b) 0001	M (7)			Ot (18)				00010101
32.	a) 0000 b) 0001	B (4)			Ot (18)				00010110
33.	a) 0000 b) 0001	B (4)							00010111
34.	a) 0000 b) 0001	H (8)			Ot (18)				00011000
35.	a) 0000 b) 0001	H (8)							00011000

\- Legend -
C (-) Contents of -
→ goes into
\+ algebraic add
∨ logical add
= equals
Ot-1 previous value of reference address