MPRO-06

HAP REPORT #1

Subject: Harvest Assembly Program - HAP

By: Raymond W. Southworth

Date: August 11, 1959

HARVEST ASSEMBLY PROGRAM

Introduction

A tentative set of formats for the Stream Byte-by-Byte instruction and its set-up words has been drawn up. The objectives have been to keep the formats as simple as possible, to achieve consistency among them, and to make them compatible with STRAP.

In general, the set-up words are considered to consist of numeric or programmer-symbolized fields such as limit of the statistical counter; and various stimulus and action codes, such as those for the statistical accumulator. It has seemed desirable to separate these two functions in coding, even though the code bits may appear in the same set-up words. The result has been to break up the block of nine set-up words into about 20 pseudo-operation codes. These are then preceded by another pseudo-op which specifies whether a full or a partial set-up is being written.

The list of mnemonics for the stimuli and actions has not been completed. Those used in the examples following the operation formats are only for illustrations.

1. **Stream Byte-by-Byte**

SBBB (Connections), $\frac{1}{3}$ (stimulus), relative address

EX: SBBB (P-G, Q-C, C-R), SKA (2P)

Notes:

1. The connections may be given either in terms of the system symbols for the actual units connected or the numbers of the open gates.
2. The number of adjustments following the SBBB instruction will be determined by the assembler and inserted in the assembled instruction.
3. If the relative address field is left blank, it is assumed that the skip will be to the SBBB instruction itself, ie., resume streaming. Skip to instructions beyond the range of the allowable relative address must be taken care of by the programmer by including a branch instruction.

2. Adjustments

ADJ $\frac{1}{3}$ (stimulus), action 1, action 2, . . . , SK$\frac{1}{3}$, relative address

EX: ADJ 1 (KB=0, F=LUM), REP (P,Q), SKIP (U), RST (T)

Notes:

1. The adjustments are numbered. They may be written in any order, but they will be assembled as numbered.
2. An action which has more than one operand, such as REP (P,Q) may be written either as above or as REP (P), REP (Q).
3. Since the action fields are identified by the op code, they may be written in any order.
4. A blank address for the skip instruction is assumed to mean a skip to the instruction following the adjustments. Omission of both the operation (SKS) and the address is assumed to mean a NOP, i.e., resume streaming.

3. **Set-up**

SETUP

PART SETUP

Notes:

1. SETUP is a pseudo-op to be used whenever it is desired to write a full set-up of nine words. It will reserve the block, set all words to zero, and then OR in the specified fields. It is not necessary to write the following set-up words in any particular order.
2. PART SETUP is used whenever it is desired to write less than the full nine words. The following codes will be examined, and a block extending from the first non-zero word to the last non-zero word in assembled form will be reserved. Unused fields within this block will be set to zero.

4. Match Definition

WMAT (character, connections)

XMAT (character, connections, R)

YMAT (character, connections, R)

ZMAT (character, connections)

EX: XMAT ((8)377, P.Q)

YMAT ((8)21, N(PVQVU), R)

Notes:

1. The match character may be given in either octal or Hollerith by means of an entry-mode.
2. The connections as used to cause a stimulus for adjustments, counters, and group size are indicated by a period between units when logical AND is desired and by a V for logical OR. The connections are assumed to be only of the OR type when considering the stimulus for the swallowing of bytes, no matter how they are written.
3. An N preceding the connections indicates a signal on no-match.
4. If R is included, the match is only on the rightmost bit. Otherwise it is on the full byte.

5. Match Swallowing Action

IF (M/N/Z OR), action

EX: IF (XOR), OMIT ALL

Notes:

1. The stimulus for the action in this instruction is a signal from the match unit specified, assuming that the inputs to the match unit are OR'ed together.

6. SC Definition

SC (limit, value)

EX: SC (1000)

7. SA Definition

SA (threshold, value)

EX: SA (5000, 1000)

8. SC and SA Action

IF (stimulus), action (s)

EX: IF (XVY), SC + 1, SARST

Notes:

1. These instructions may be broken down into stimuli and actions for counting and reset in the two units. If the same stimulus applies to more than one action, however, they may be written together, as above.

9. Logic Unit Definition

$$G\ (\text{Modulus},\ \overset{D}{S},\ \overset{CP}{0}\ )$$

EX: G (150)

Notes:

1. The modulus as assembled will be the necessary machine value, e.g., 75 in the above case.
2. If neither Double nor Single column operation is specified, S will be assumed.
3. If CP is not specified, there will be no carries propogated.

10. Logic Unit Operation

CG XXXX	Connect, where XXXX is used as in STRAP.
EQTG (P, 0)	Equals test
EQTG (P, NB)	
EQTG (0,P)	
EQTG (NB,P)	
EQTG (0,Q)	
EQTG (NB,Q)	
GETG (P-Q,0)	Greater than or equals test
GETG (P-Q, NB)	
LETG (Q-P, 0)	Less than or equals test
LETG (Q-P, NB)	
MODG (P-Q)	P-Q, mod M
MODG (Q-P)	Q-P, mod M
MODG (P + Q)	P + Q, mod M (code 30)
MODMG (P + Q)	P + Q, mod M (code 31)

Notes:

1. The connections to the logic unit may be P, SC, or U, and Q. Thus in all the above P may be replaced, if desired, by SC or U if one of those is being usdd in place of P.

2. In each of the test functions, the comparison is between the P and Q bytes, always in that order. If the test is satisfied, the output is the first byte given within parentheses; if it is not satisfied, it is the second byte.

11. F Unit Definition

F (group size, limit)

EX: F (1 Q, 2)

12. F Unit Operation

IF (stimulus), action

EX: IF (KK, LM), S0

Notes:

1. The stimulus in this case is written as the bit or bits that are to be tested. A one in any of them causes the F counter to advance according to the indicated action.

13. Index Starting Addresses and H Addresses

SP, base address, index table address
SQ, base address, index table address
SR, base address, index table address

14. Table Assembly Unit

T (Mode, RPL), TBA, K, T1 (S1, N1, I1,), T2 (S2, N2, I2)

Notes:

1. TBA = table base address
2. K = Cell size
3. RPL if present indicates replacement of TBA by new address
4. S = initial offset
5. N = number of bytes
6. I = increment

15. Table Extract Unit

U (Mode, MDM), N, I, BM_u

Notes:

1. MDM = memory distributor mode
2. BM_u = byte mask

16. Index Words

PX (Mode, EC, FF, NR, RI, SS, $\begin{matrix} M \\ R \\ MR \\ 0 \end{matrix}$), N, I, BM_p, offset, RBL

Notes:

1. If any of the one-bit codes, such as EC, is not specified the opposite case, CC here, will be assumed. These codes need not be in any particular order.

2. For an HL bit, the code PXH should be used. No byte mask need be included.

3. For a branch level word, the code PXBL should be used. The branch address is then written in place of the byte mask.

4. The programmer fields: N, I, etc. must be in the order shown. Null fields are indicated by a zero; field drop-out is from the right.