IBM Customer Engineering

Manual of Instruction

IBM 1410-1401 Compatibility

223-2597

IBM

Customer Engineering Manual of Instruction

1410 - 1401 Compatibility

© 1961, 1963 by International Business Machines Corporation

MINOR REVISION (January 1963)

This edition, Form 223-2597, is a revision of the section entitled "1401 Compatibility" (pages 333 through 411) of *Customer Engineering Manual of Instruction, 1410 Data Processing System,* Form 225-6549-1. Principal changes in this edition are:

PAGE	SUBJECT
13	Indexing
55	I-O Error Conditions

Address comments regarding the content of this publication to: IBM Corporation, Dept 296, PO Box 390, Poughkeepsie, N. Y.

Contents

The basic design of the IBM 1410 Data Processing System permits the running of many programs as originally written for IBM 1401 Data Processing System. The compatibility switch on the CE panel of the IBM 1415 Console enables a 1410 system to utilize programs written for all models of the 1401 (A, B, C, D, E, and F). This manual assumes a knowledge of the basic 1410 Data Processing System.

One major difference occurs when the IBM 1410 system operates in the 1401 mode. This is the use of the three-position addresses. The digit portion of the address locates storage positions 000 through 999. The zones over the units and hundreds positions are assigned binary values to increase the number of addresses to 15,999. The A-zone over the hundreds position has a digit value of 1 or 1,000. An address T33 (T is A, 2, 1 bits) locates storage position 01333. The digit values of the other zones are:

B = 2	.....	B = 8
A = 1	.....	A = 4
Hund	Tens	Units

The address to locate position 15,999 is I 9 I (I is A, B, 8, 1 bits).

Some 1401 operations require the addition of two addresses. To handle these operations on a 1410 system requires the addition of the zone adder and the auxiliary binary adder.

Zone and Auxiliary Binary Adders

The zone adder combines the binary 1 and 2 bits while the auxiliary binary adder combines the 4 and 8 bits. For example:

AUX BIN ADDR	ZONE ADDER
	8421
Add A —	0111 = 7
B —	0110 = 6
	0001
Carry	11
	1101 = 13

Notice that each position must be able to carry into the next position. On the data flow diagram (Figure 1) the line from the zone adder to the auxiliary binary adder is the zone adder carry. In adding two addresses together, there can be a carry from the hundreds position digits that are added in the adder. For example:

BINARY	8421	DECIMAL
		B
		A A
A-field	0111	821 = 07821
		A A
B-field	0101	380 = 05380
	0010	101
Carry	1 11	1
	1	B
		A A
	1101	201 = 13201

On the data flow diagram, the line from the adder to the zone adder is the adder carry.

Operation (Figure 2)

When the zones are available to the zone adder, the A zones combine and the B zones combine. A carry from the A zones feeds to the B zones. A carry into the zone adder combines with the A and B zones to control the output of the zone adder. The output of the zone added is the two zones, plus a zone check bit that maintains odd parity. Two B zones or a carry into a B zone causes a zone adder carry that feeds to the auxiliary binary adder. The auxiliary binary adder combines the units-positions zones and the zone-adder carry. The output of the zone adder feeds to the assembly unit, and the output of the auxiliary binary adder feeds to the compatibility translator BCD to two-of-five. Zone adder and aux binary adder outputs are available as soon as the inputs are conditioned. No zone assembly controls are used. (See adder zone inputs to assembly on Systems 15. 50. 05, 06, 07.)

Instruction Read-Out (Figure 3)

Instruction read-out of a 1401 instruction uses the zone and auxiliary binary to convert the three-position addresses to the five-position addresses that are used by the 1410 to address storage by the 1410.

The instruction word is reverse scanned (+1) with the op code as the first character read out of storage. The op code must contain a word mark (WM) to be set into the op register. As in the 1410 mode, a failure to have a WM prevents the next I-cycle and causes an instruction check. The character in the op register is decoded to control the remainder of instruction read-out (Figure 4).

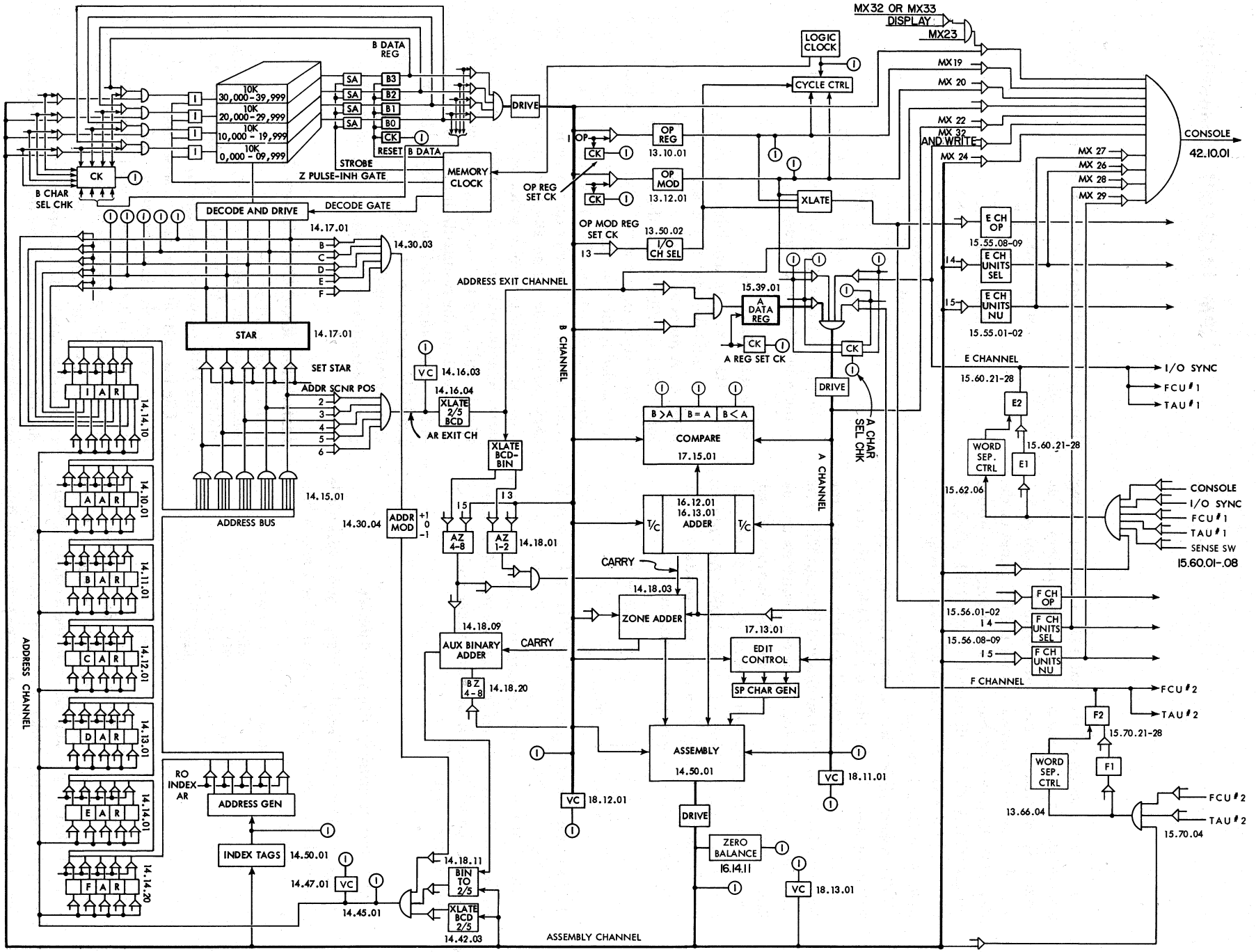
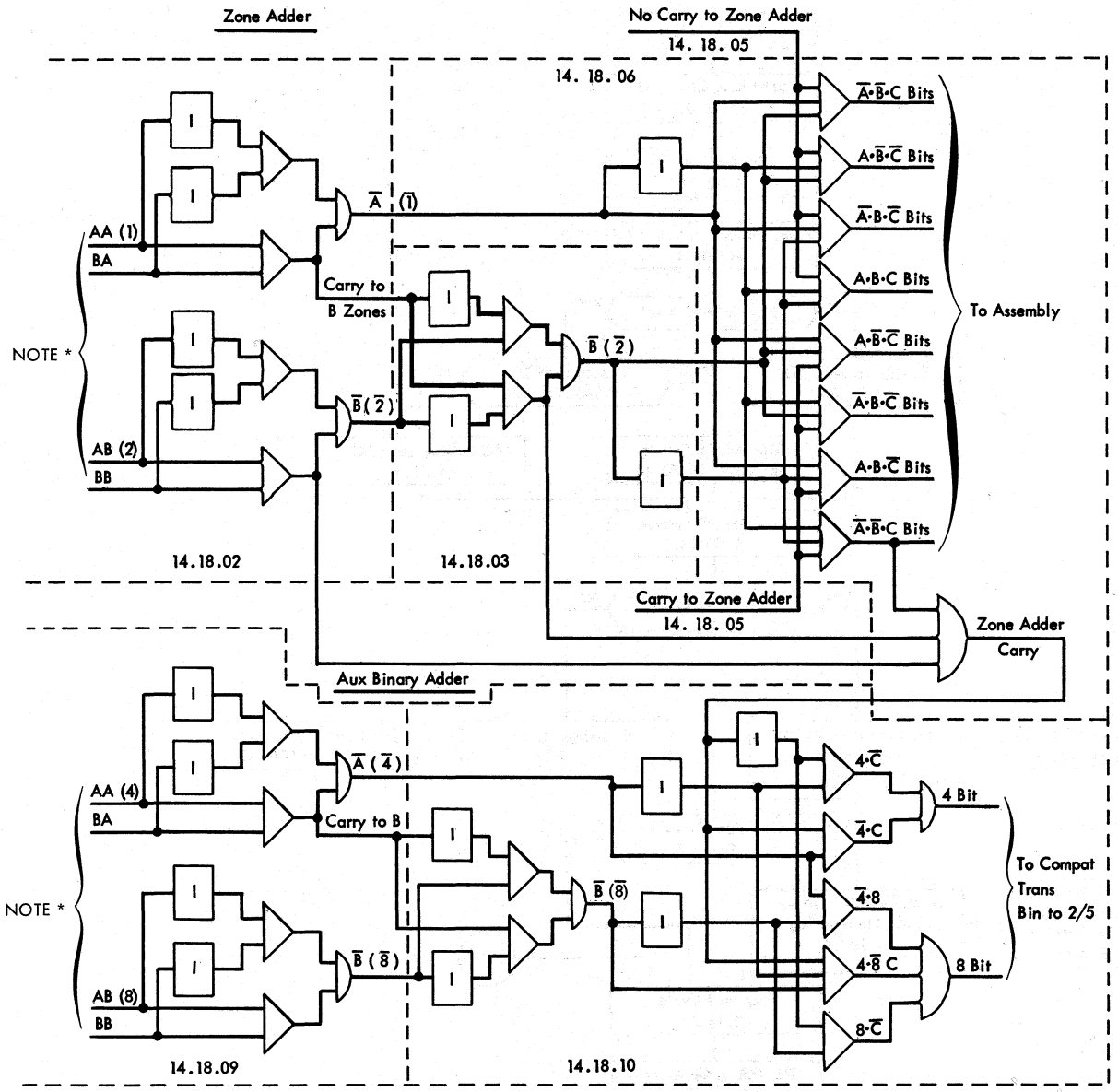


Figure 1 IBM 1410 Data Flow



* Zone Adder may receive inputs from the following:
1 - A channel A and B zones.
2 - B channel A and B zones.
3 - Binary A reg 1 and 2 latches.
4 - Binary A reg 4 and 8 latches.
5 - Carry to zone adder.

Aux Binary Adder may receive inputs from:
1 - Binary A register 4 and 8 latches.
2 - Binary B register 4 and 8 latches.
3 - Zone Adder carry.
For this figure, inputs are A and B zones from A- and B-ch.

Figure 2. Zone and Auxiliary Binary Adders

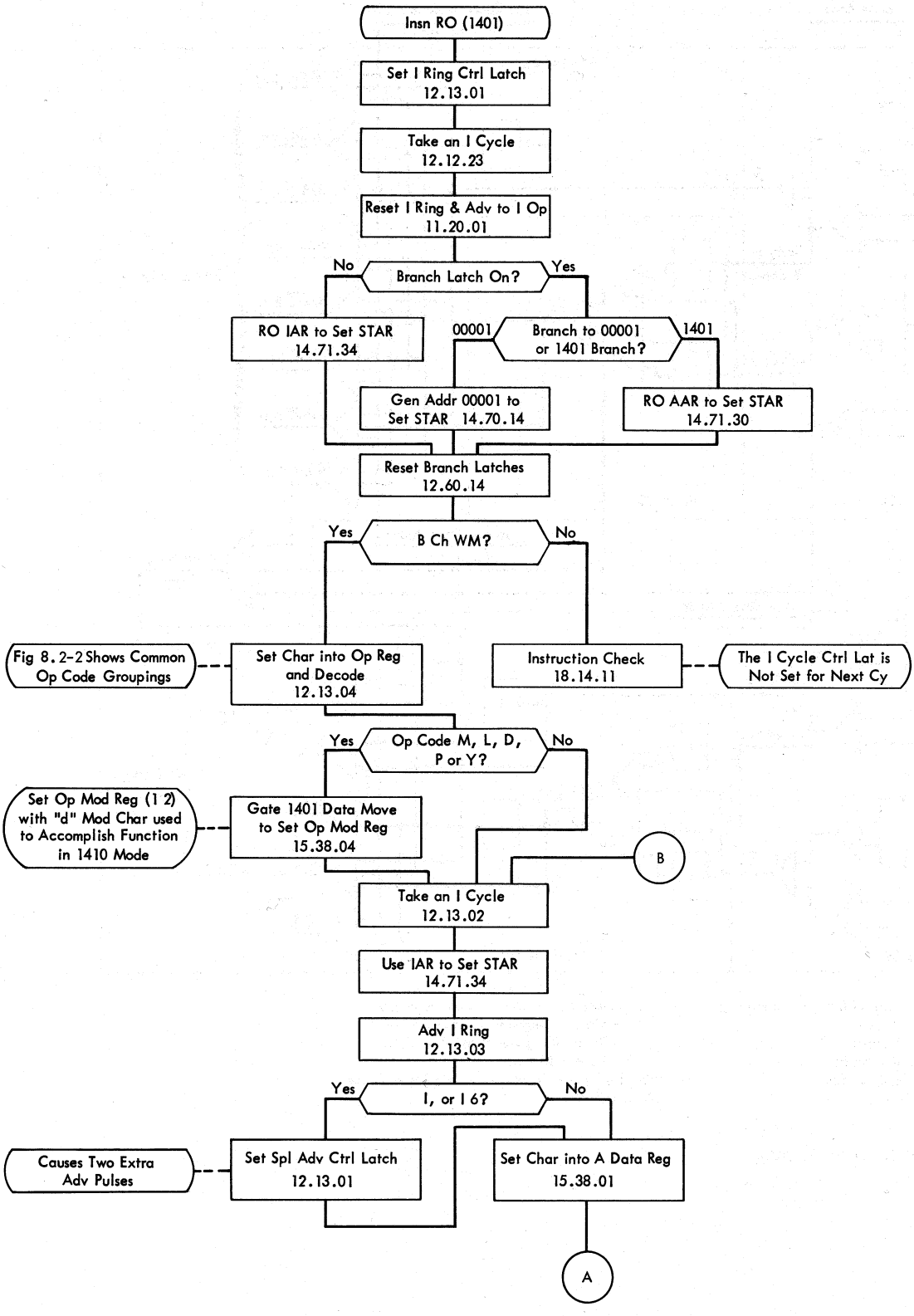


Figure 3A. Instruction Read-Out (IBM 1401 Mode)

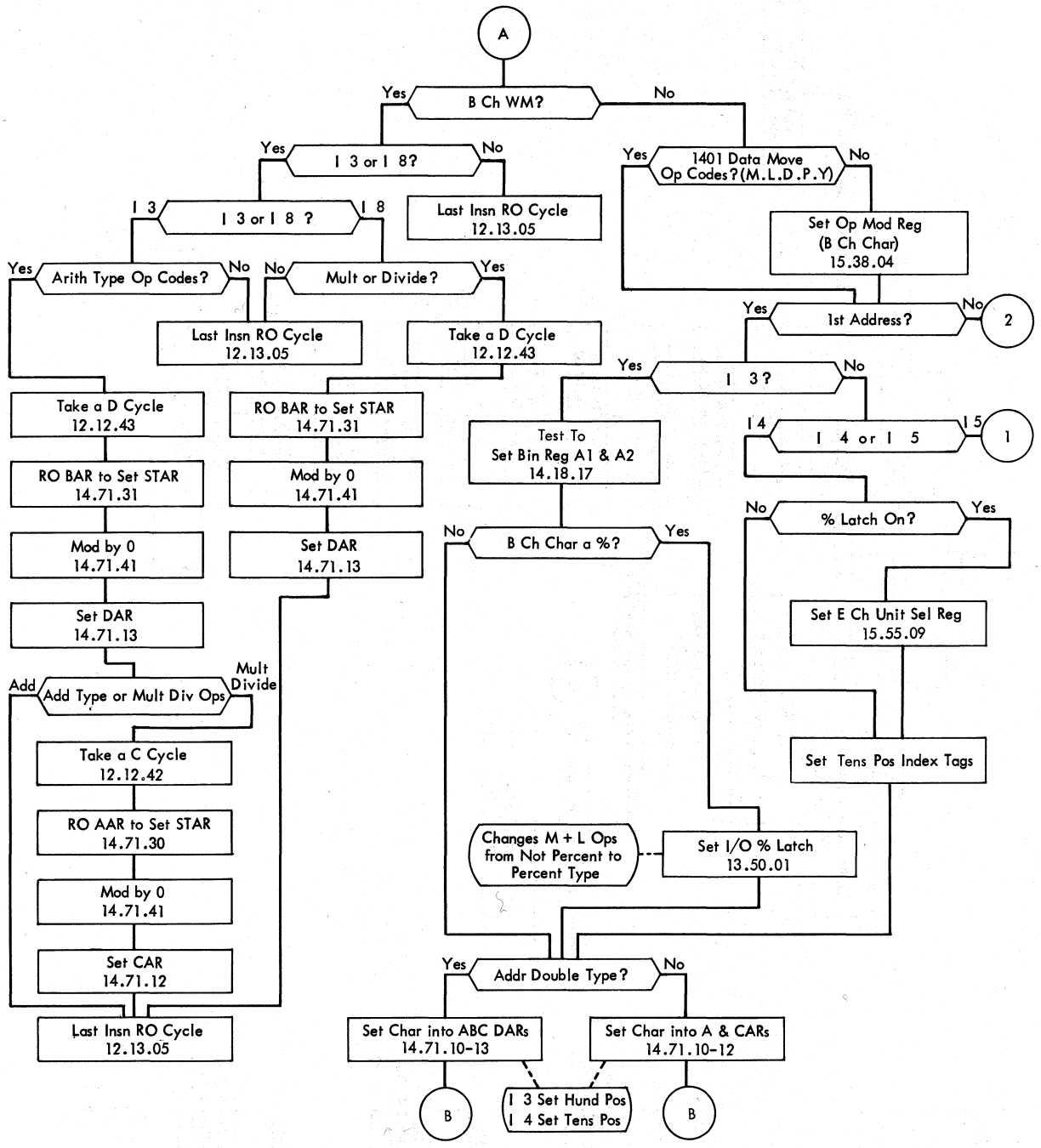


Figure 3B. Instruction Read-Out (IBM 1401 Mode)

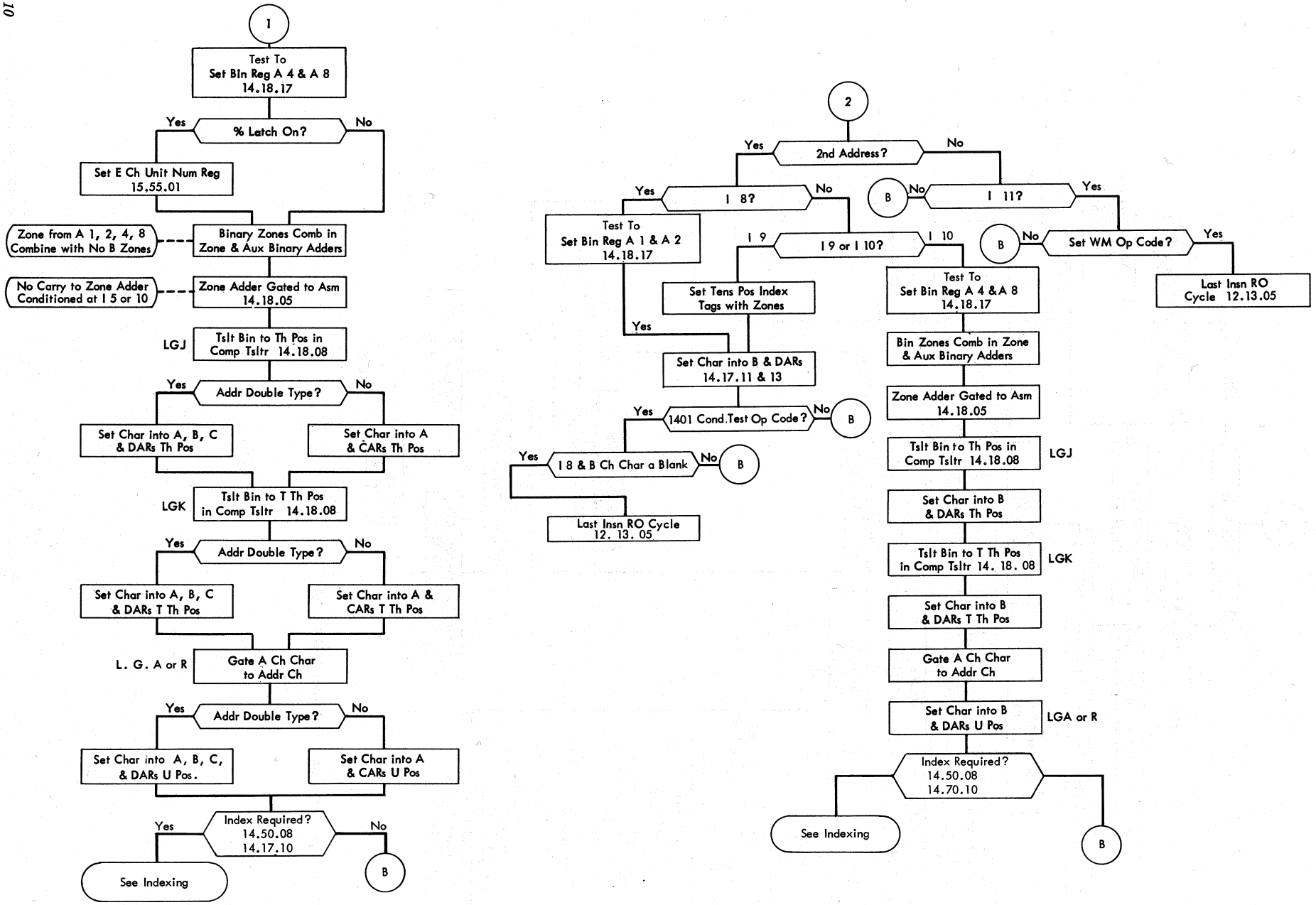


Figure 3C. Instruction Read-Out (IBM 1401 Mode)

1401 OP CODES

COMMON OP CODE GROUPING	?	!	A	S	@	%	E	Z	C	W	V	/	•	,	¤	U	#	D	P	Y	B	H	Q	1-7	8	9	M	L	K	F	N
PERCENT TYPE OP CODES																X											%	%			
NOT PERCENT TYPE OP CODES	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X	X	BS/BL						$\overline{\%}$	$\overline{\%}$			
ADDR DOUBLE OP CODES	X	X	X	X								X		X	X						BS										
NOT ADDR DOUBLE OP CODES					X	X	X	X	X	X	X		X			X		X	X	X	BL	X	X				%/$\overline{\%}$	%/$\overline{\%}$	X	X	
1 ADDR PLUS MOD OP CODES																X					BS										
2 ADDR NO MOD OP CODES	X	X	X	X	X	X	X	X	X			X		X	X																
2 ADDR PLUS MOD OP CODES										X	X							X	X	X	BL						%/$\overline{\%}$	%/$\overline{\%}$			
TWO ADDRESS OP CODES	X	X	X	X	X	X	X	X	X	X	X	X		X	X			X	X	X	BL						%/$\overline{\%}$	%/$\overline{\%}$			
ADDR TYPE OP CODES	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	BS/BL						%/$\overline{\%}$	%/$\overline{\%}$			
2 CHAR ONLY OP CODES																													X	X	
C CYCLE OP CODES					X	X																									
NO C OR D CYCLE OP CODES							X	X	X	X	X	X	X	X	X			X	X	X	BS/BL						$\overline{\%}$	$\overline{\%}$			
NO D CYCLE AT I RING 6 OPS	X	X	X	X			X	X	X	X	X	X	X	X	X			X	X	X	BL						$\overline{\%}$	$\overline{\%}$			
NO INDEX ON 1ST ADDR OPS																X											%	%			
RESET TYPE OP CODES	X	X																													
ADD OR SUBT OP CODES			X	X																											
MPY OR DIV OP CODES					X	X																									
ADD TYPE OP CODES	X	X	X	X																											
ARITH TYPE OP CODES	X	X	X	X	X	X																									
E OR Z OP CODES							X	X																							
COMPARE TYPE OP CODES									X												BL										
BRANCH TYPE OP CODES										X	X	X	X								BS/BL										
NO BRANCH OP CODES	X	X	X	X	X	X	X	X	X					X	X	X		X	X	X							%/$\overline{\%}$	%/$\overline{\%}$	X	X	
WORD MARK OP CODES														X	X																
M OR L OP CODES																											$\overline{\%}$	$\overline{\%}$			
1401 STORAGE AR OP CODES																						X	X								
1401 NO OP LIROC																									X	X					X
1401 CARD OR PRINT																								X							
1ST SCAN FIRST OP CODES	X	X	X	X	X	X	X	X	X	X	X	X		X	X		X				BL	X	X								
A CYCLE FIRST OP CODES	X	X	X	X	X	X	X	X	X					X	X		X	X	X	X							$\overline{\%}$	$\overline{\%}$			
STD A CYCLE OP CODES	X	X	X	X	X	X	X	X	X								X	X	X	X							$\overline{\%}$	$\overline{\%}$			
B CYCLE FIRST OP CODES										X	X	X									BL										
A REG TO A CH ON B CYCLE OPS	X	X	X	X	X	X	X	X	X			X	X	X	X		X	X	X	X		X	X				$\overline{\%}$	$\overline{\%}$			
OP MOD TO A CH ON B CYCLE OPS										X	X					X					BS/BL								X	X	
LOAD MEM ON B CYCLE OP CODES	X	X	X	X	X	X	X	X									X	X	X	X							$\overline{\%}$	$\overline{\%}$			
REGEN MEM ON B CYCLE OP CODES									X	X	X		X								BS/BL								X	X	
STOP AT F ON B CYCLE OP CODES										X	X	X	X								BS										
STOP AT J ON B CYCLE OP CODES	X	X			X	X	X	X									X	X	X	X							$\overline{\%}$	$\overline{\%}$			
READ OUT B AR ON SCAN B CY OPS							X	X	X	X	X	X	X	X	X			X	X	X	BS/BL						$\overline{\%}$	$\overline{\%}$			
READ OUT A AR ON A CYCLE OPS	X	X	X	X			X	X	X					X	X		X	X	X	X		X	X				$\overline{\%}$	$\overline{\%}$			

NOTE

SYMBOL	MEANING	TYPICAL FORMAT
BS	B SHORT	B (I) d
BL	B LONG	B (I) (B) d
M%	M PERCENT	M (%UX) (B) d
M$\overline{\%}$	M NOT PERCENT	M (A) (B)
L%	L PERCENT	L (%UX) (B) d
L$\overline{\%}$	L NOT PERCENT	L (A) (B)

Figure 4. IBM 1410 Common Op-Code Grouping, IBM 1401 Compatibility

The I-ring advances to I1 during the second I-cycle that sets the "spl adv ctrl" latch to advance the I-ring to I3. Every character read out during instruction read-out sets into the A-data register.

Op codes M, L, D, P, and Y are 1401 data move op codes. When one of these is decoded in the op decode, it causes the op modifier register to set with the d-modifier character that is used by the corresponding 1410 data move op code.

For example, a 1401 D-op code moves the numeric portion of a single A-field character to the B-field. At I-ring-2-time (LGC of the second I-cycle) this op code sets the op modifier register with a 1 bit only. All M- or L-op codes are assumed to be 1401 data move op codes at I2, and they set the op modifier register accordingly. At I3, if these op codes are I-O op codes, the percent latch sets and the op mod register resets with R or W modifier at 111. For all op codes except 1401 data move operations, the op modifier register sets with every character except the one at the last instruction RO cycle.

The zones of the character that reads out at I3 time are stored in the auxiliary binary registers A1 and A2. The digit is gated to the address channel where it is set into the A and C address registers or the A, B, C, and D address registers (address-double type).

The I-ring advances to I4, and the next character reads out. Any zones over this character set into the tens position index tag latches. The digits are stored in the address registers.

The ring is now advanced to I5 where the zones (over the units position of the address) are set into the binary register A4 and A8 latches. The zones from binary registers A and A2 are gated to the zone adder where they combine with no-zones on the B side of the adder. At the same time, the zones from binary registers A4 and A8 combine in the auxiliary binary adder.

The output from the zone adder is gated through the assembly to the 1401 compatibility translator where (along with the output of the auxiliary binary adder) it is translated into the thousands and ten-thousands positions of the address at LGJ and LGK, respectively (Figure 5).

The I-ring advances to I6 where the "spl adv ctrl" latch again sets to advance the ring to I8. The second address reads out from I8 to I10.

Instruction read-out ends when the B-channel WM over the next op code is sensed. The set-WM op code and 1401 condition test op code end the instructions read-out at I11 and I8, respectively, without a word mark to the right of the instruction word.

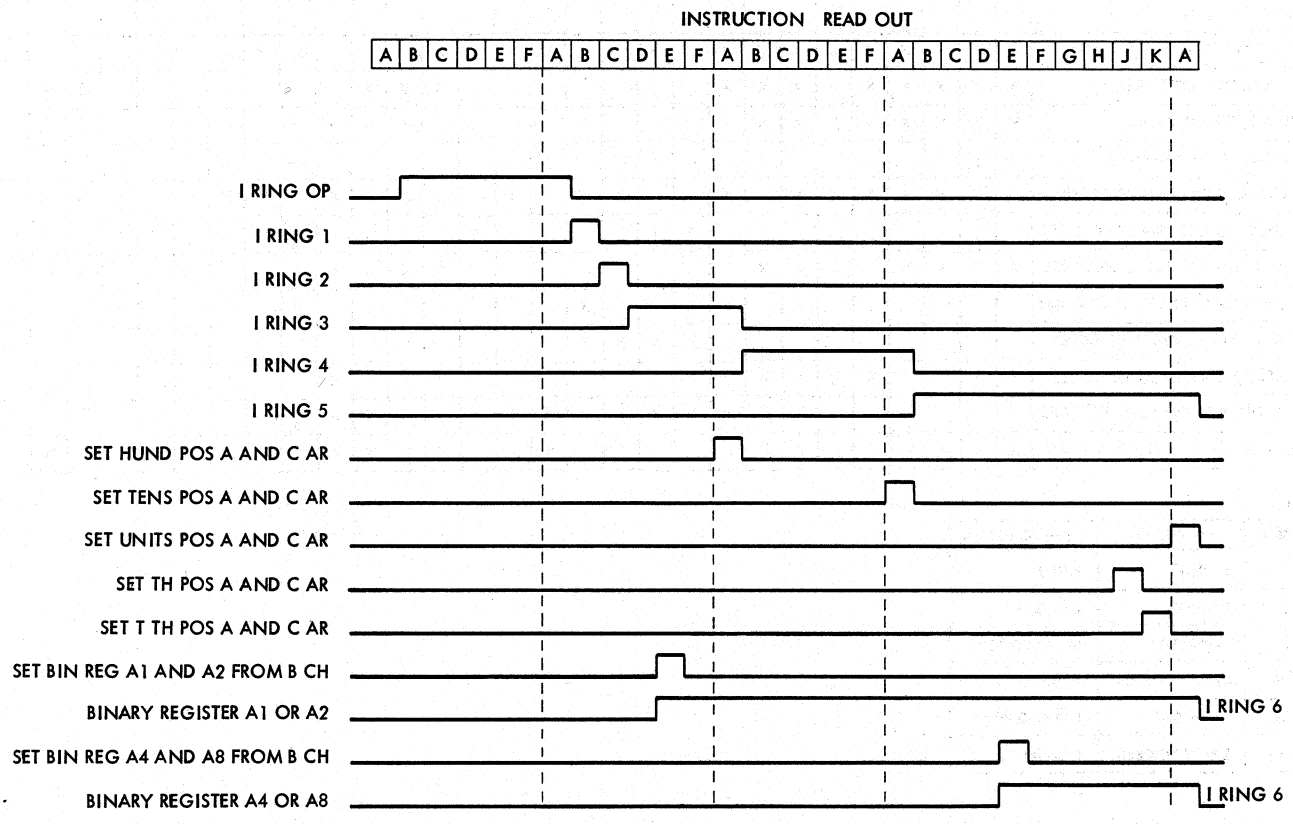


Figure 5. Zone Adder Timing, Instruction Read-Out

Indexing

Indexing in the 1401 mode is the same as in the 1410. However, selection of index registers is limited in the 1401 mode because of the use of zone bits over the units and hundreds positions of the three-character 1401 address that represent the thousands and ten-thousands position of the 1410 address. Thus, only the A and B bits over the tens position of the 1401 address character can be used to select the index register. The A and B bits provide only four combinations. These four combinations select index registers as indicated:

INDEX TAG	REGISTER	ADDRESS
None	—	—
A	13	00087-89
B	14	00092-94
A and B	15	00097-99

The following example represents typical 1401 instruction-word format with index bits:

```
          A A       A A
          B B       B   B
Op        (1 1 1)   (3 3 3)  d
```

This becomes:

```
(0 3 1 1 1)        (1 1 3 3 3) d
Indexed by         Indexed by
contents of        contents of
00097-99           00087-89
```

Instruction read-out begins in the usual manner with the read-out of the op code from memory. The operation continues as previously explained until I-ring-4 time when the read-out of the tens position detects the index tags. The ten-position index-tag latches set at I-ring-4, LG-A time. This in turn sets the X-cycle control, and starts the A-ring at I-ring-5, LG-F time.

When the 1410 operates in the 1401 mode, both A- and B-index tags are forced in the hundreds position. This, in combination with the ten-position index tags, selects index registers 13, 14 or 15.

At the end of I-ring-5 time, the A-address has read from memory and converted into a five-digit-numeric address in the AAR and CAR. Instruction read-out now temporarily halts to permit indexing of the A-address. The I-ring-5 line remains up until the end of the indexing operation. I6 then turns on to continue the instruction read-out.

The index tags determine which index register the address generator selects. The address generator sets the STAR. Each position of the index field reads out in sequence to the B-channel starting with the units. This operates the same as 1410 indexing.

The following sequence of events, shown in Figures 6 and 7, takes place:

1. Read out units position of index register. At A-ring-1, LG-A time, the address generator sets the STAR to read out the units position of the index register. The zone-bit index tags in the tens position of the A address determine the address to be generated. The read-out of memory is placed on the B channel for entry into the adder. Typical contents of an index register in 1401 mode are:

```
A   A
    B
4 2 2
```

The zone bits over the units and hundreds position provide the two high-order positions of the index register contents.

2. Set binary register B4 and B8. The zone bits over the units position also read out to the B channel and set in the B4 and B8 binary register at A-ring-2, LG-E time. These zone bits are added later in the auxiliary binary adder to the zone bits over the units position of the address of the instruction that is being indexed. These zone bits (A4, A8) are still in the A-binary register because the register is not reset until I-ring-6 time of instruction read-out.

3. Reset AAR. The AAR is reset at A-ring-2, LG-F time because the indexed instruction reads into the AAR. The CAR is not reset, but retains the numeric values of the unindexed instruction.

4. Read out units position instruction character to A channel. The units character of the CAR (contains the A address) reads out, translates from two-of-five to BCD, and sets in the A-data register at LG-D time of the first X cycle.

5. Add units position of the index word to the units position of the instruction word. The adder is now used to add the numerical portion of the units character of the instruction address from the A-data register to the numerical portion of the units character of the index word (from memory but now on the B channel). The sum is placed in the AAR units position at A-ring-2, LG-H time.

6. Add tens-position characters. At A-ring-3 time the tens-position character from the CAR adds to the tens-position character from the index register. The result sets in the AAR at A-ring-3, LG-H time.

7. A-ring cycling. Only three A-ring times, in addition to A1, are required to index the three-character 1401 program address. The index register and instruction zone bits must add in the zone adder and auxiliary binary adder so that the thousands and ten-thousands characters can set in the AAR. To provide time for this, the A-ring-4 is extended to include LG-J and K.

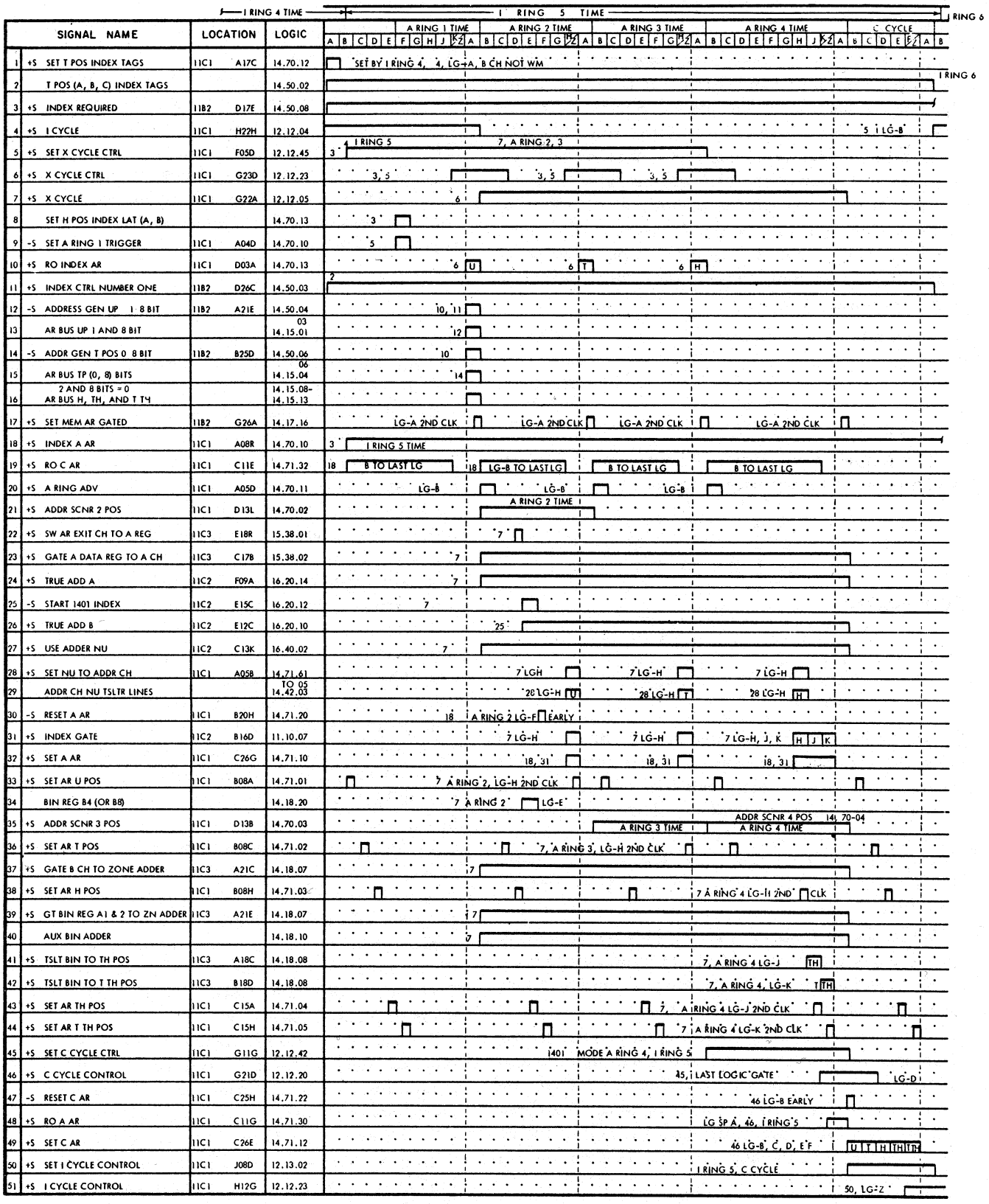


Figure 6. 1401 Mode Indexing Sequence

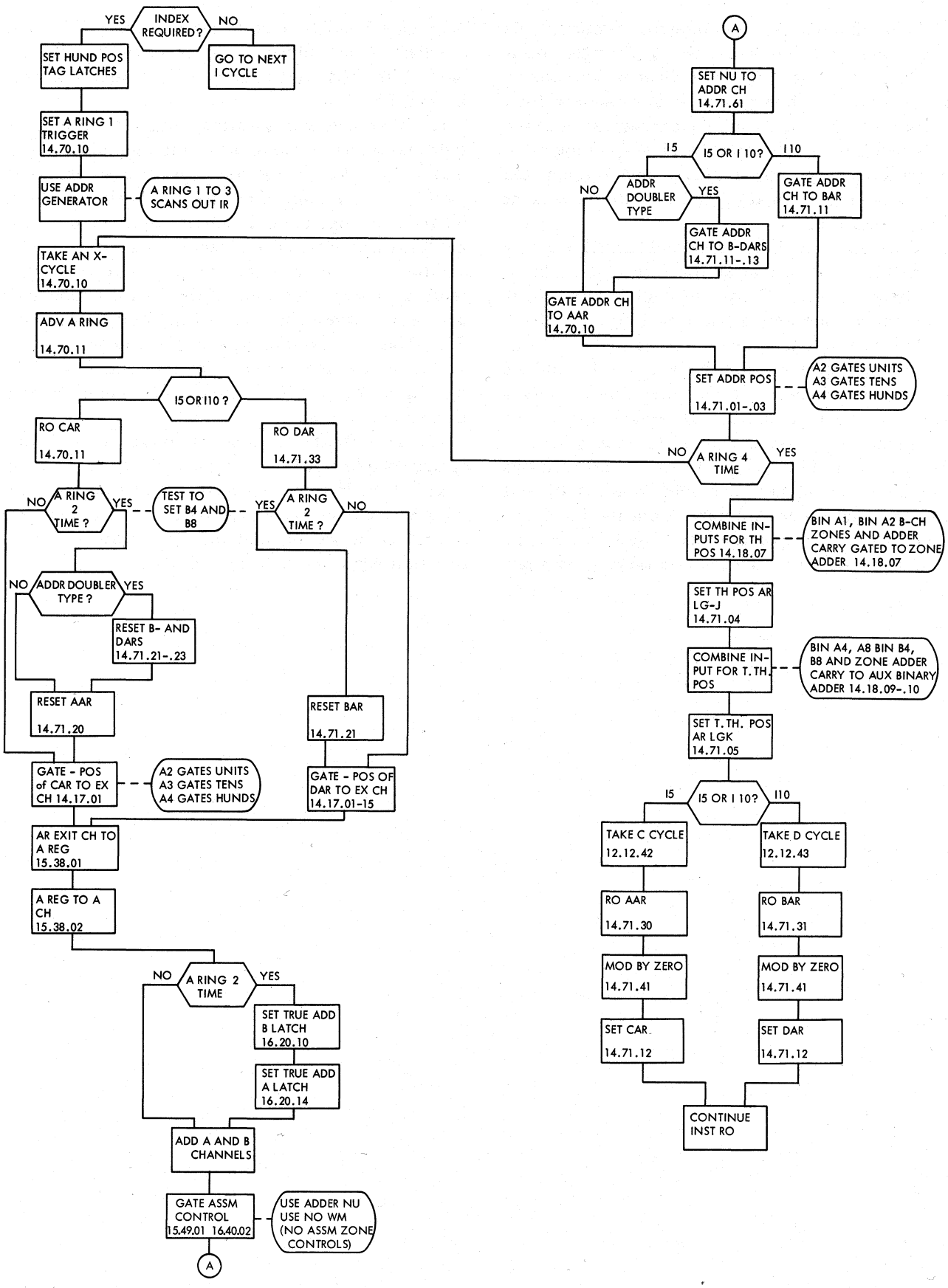


Figure 7. 1401 Mode Indexing

8. Add hundreds-position numeric characters. At A-ring-4 time, the hundreds-position character reads from the index register to the B channel. Also during A-ring-4 time, the hundreds-position character from the CAR reads into the A-data register, and then to the A channel. The result of the hundreds-position addition passes through the assembly area through the BCD-to-two-of-five translator, and sets in the AAR at A-ring-4, LG-H time.

9. Enter zone bits in the zone and auxiliary binary adders. The zone bits from the instruction address are still in the A1, A2, A4, and A8 binary-register latches where they were set during instruction readout. The zone bits from the index-register units-position character were previously set in the B4 and B8 binary-register latches. The zone bits from the index-register hundreds position (B1 and B2), via the B channel, are set directly into the zone adder.

10. Add zone bits. The instruction-word zone-bits A1 and A2 add to the index-word zone-bits B1 and B2 in the zone adder. Any carry from the hundreds-position CPU adder operation also enters in the zone adder. The result of the addition transfers to assembly. Any carry from the zone adder addition adds in the auxiliary binary adder to the instruction-word zone-bits A4 and A8, and to the index-register zone-bits, B4 and B8.

11. Translate zone-addition output to two-of-five code and set AAR. The result of the zone-adder operation is brought via the assembly area to the compatibility translator. The result of the auxiliary binary-adder operation is also brought to the compatibility translator. The binary results of the zone-bit addition are translated to the two-of-five code. The result represents the value of the indexed thousands and ten-thousands characters of the instruction word. The thousands position sets in the AAR at A-ring-4, LG-J time. The ten-thousands position sets in the AAR at K time. As previously noted, the hundreds position sets at LG-H time.

12. Transfer the indexed AAR to the CAR. Indexing is now complete. A C cycle is taken to read out the indexed A address of the instruction from the AAR. This is modified by zero and set in the CAR. I-cycle control is turned on and the B portion of the instruction begins with I-ring-6 time.

The 1401 D, P, Y, M ($\overline{\%}$) and L ($\overline{\%}$) op codes perform nearly the same function as the 1410 D-op code. There is this significant variation: the 1410 uses different op modifiers to perform different types of move operations; the 1401 instruction requires a different op code to obtain variations of the basic move operation.

By comparing the 1410 D-op code and modifications with the 1401 D, P, Y, M and L operations, it can be seen that in order to duplicate the operation in 1401 mode, it is necessary to generate an appropriate op modifier. This occurs at I2 time.

The op modifier is generated for the various move and load operations in accordance with the following table:

1401 INSTRUCTION	d-CHARACTER TO BE GENERATED		BITS
Move numeric	D (A) (B)	1	1, $\overline{2}$, $\overline{4}$, $\overline{8}$, $\overline{A}$, $\overline{B}$, $\overline{C}$
Move zones	Y (A) (B)	2	$\overline{1}$, 2, $\overline{4}$, $\overline{8}$, $\overline{A}$, $\overline{B}$, $\overline{C}$
Move character to record, or group-mark with a word-mark	P (A) (B)		1, 2, $\overline{4}$, 8, A, B, $\overline{C}$
Move characters to A or B WM	M (A) (B)	C	1, 2, $\overline{4}$, $\overline{8}$, A, B, C
Load characters to A-field WM	L(A) (B), L (A)	X	1, 2, 4, $\overline{8}$, A, $\overline{B}$, C

The A and B bits, along with the 8 bit, determine when the data transfer should end.

OP MODIFIER BITS	STOP AT
$\overline{8}$, A, $\overline{B}$	1st WM—A-field
$\overline{8}$, A, B	1st WM—A or B-field
8, A, B	1st RM or GM, WM, A-field
$\overline{8}$, $\overline{A}$, $\overline{B}$	After one position

The 8 bit determines the direction of the scan:

1. 8 bit: Reverse scan.
2. Not 8 bit: Forward scan.

The 1, 2, and 4 bits of the op modifier determine which portion of the A-field characters transfers:

OP MODIFIER BITS	USE
1	A-field—num
Not 1	B-field—num
2	A-field—zones
Not 2	B-field—zones
4	A-field WM
Not 4	B-field WM

Move Numeric (D)

The 1401 D-op code moves the numeric portion (8-4-2-1) of the single character in the A address to the B address. The zone portions (AB) and word marks are not disturbed at either address.

Op-Modifier Function

The 1401 D-op does not have a modifier. The proper 1410 modifier must be set in the op modifier register to get the desired transfer. The following 1410 functions must be performed:

1. Transfer A-field numeric (op modifier bit 1).
2. Retain B-field WM (op modifier bit $\overline{4}$).
3. Retain B-field zone (op modifier bit $\overline{2}$).
4. End data transfer after one positon (op modifier bit $\overline{8}$ $\overline{A}$ $\overline{B}$).

Operation (Figure 8)

The op modifier register sets to 1 $\overline{2}$ $\overline{4}$ $\overline{8}$ $\overline{A}$ $\overline{B}$ $\overline{C}$ to cause the 1410 to function as indicated in the preceding section. Cycling of the 1410 occurs as with a normal 1410 D-op. The same op-code-grouping lines come up. During the A cycle the A-field character reads out of storage and sets into the A-data register. During the following B cycle the B-field character reads out to the B channel. The 1-bit op-modifier character brings up assembly controls to gate the A-field numeric from the A-channel, and to gate the B-field zone and WM from the B channel to the core storage B-field address. The $\overline{8}$ $\overline{A}$ $\overline{B}$ op-modifier character ends the operation after one A cycle and one B cycle.

Move Zone

The 1401 Y-op code moves the zone positions (AB) from the A address to the B address. The numeric portion of each character is not disturbed at either address.

Op-Modifier Function

The Y-op code does not exist in the 1410. Therefore, a D-move op with the proper modifier accomplishes the

1401 op Y function in the 1410. The following 1410 functions must be performed. The indicated op-modifier bits must be set.

1410 FUNCTIONS	OP MODIFIER BITS
Transfer A-field zones	Set Bit 2
Retain B-field WM	Set Bit Not 4
Retain B-field numeric	Set Bit Not 1
End data transfer after one position	Set Bits $\overline{8}$ $\overline{A}$ $\overline{B}$

Operation (Figure 8)

The op modifier register sets to 1 $\overline{2}$ $\overline{4}$ $\overline{8}$ $\overline{A}$ $\overline{B}$ $\overline{C}$ to cause the 1410 to function as indicated in the preceding paragraph. The operation continues the same as the move numeric 1401 op code. The only difference is the setting of the op modifier register and the assembly unit controls.

Circuits

Set op-modifier register to $\overline{1}$ $\overline{2}$ $\overline{4}$ $\overline{8}$ $\overline{A}$ $\overline{B}$ $\overline{C}$. This operates the same as the move numeric op code. See the section "Move Numeric (D)." The only difference is that the 2-op-modifier register bit is set instead of the 1 bit.

Move Characters to Record or Group Mark/Word Mark

The 1401 P-op code moves an entire record from one core storage area to another. The A and B addresses specify the high-order position of each area of storage. Transmission starts at the high-order address of the A field and continues until a record mark (A82 bits) or a group mark with a word mark (BA 8 4 2 1 WM) is sensed in the A field. The record mark or group mark with a word mark transfers to the B field. Word marks in the B field remain unchanged. A-field word marks do not transmit.

The P-op code does not exist with the 1410. Therefore, a D-move op with the proper modifier accomplishes the 1401 op P-function with the 1410.

Op-Modifier Function

The 1401 P-op does not have an op modifier. Therefore, the proper 1410 D-op code modifier must be set in the op modifier register to get the desired transfer of data. The following 1410 functions must be performed:

1410 FUNCTIONS	OP-MODIFIER BITS
Transfer A-field numeric	Bit 1
Transfer A-field zones	Bit 2
Retain B-field WM	Bit not 4
Transfer high to low-order	Bit 8
Stop operation with 1st RM or GM · WM A field	8 A B

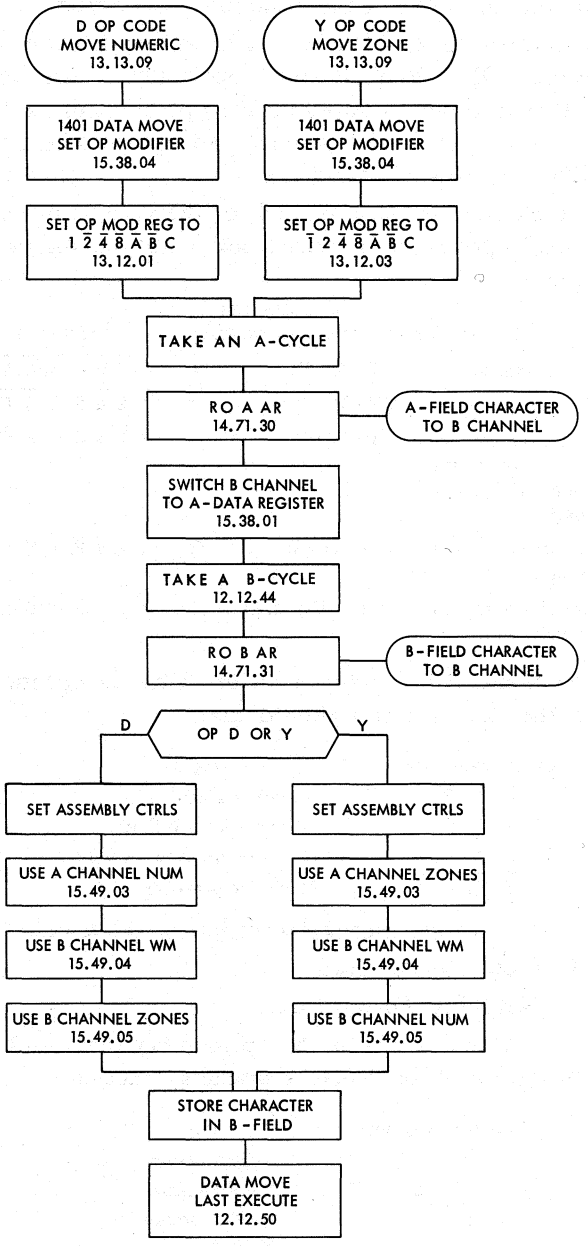


Figure 8. 1401 Mode, Move Numeric and Move Zone

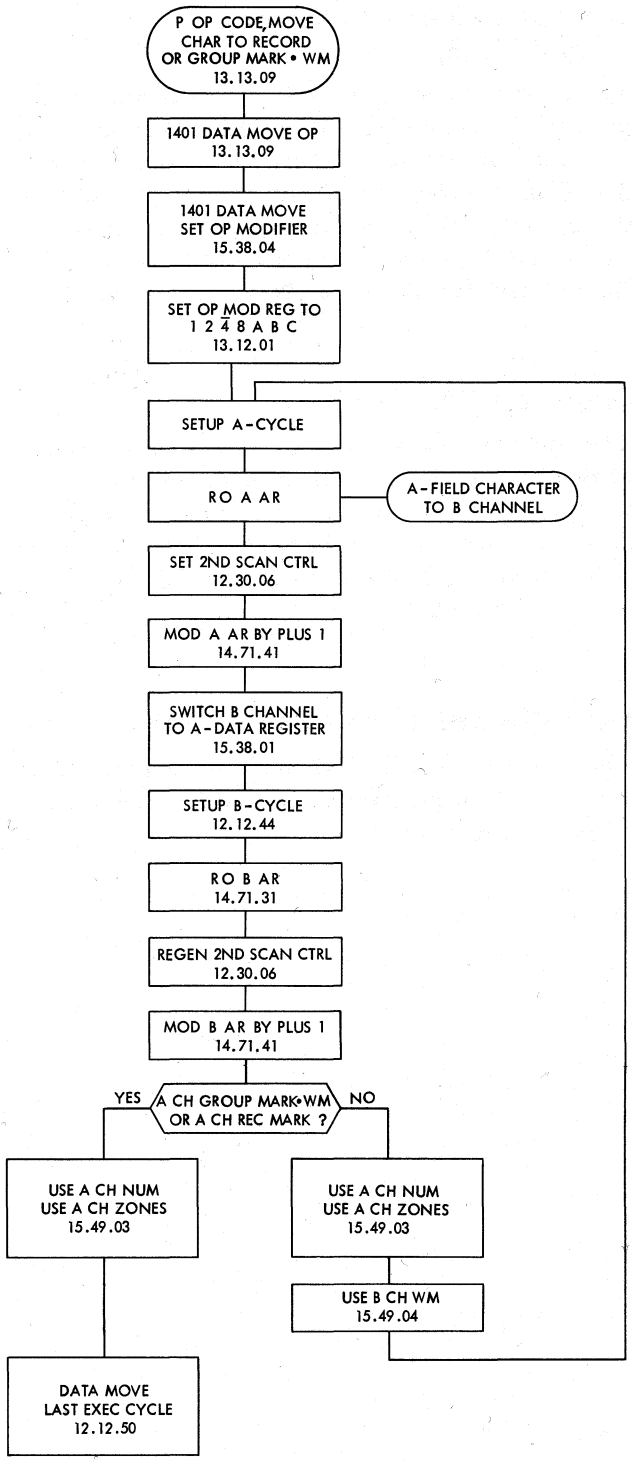


Figure 9. 1401 Mode, Move Character to RM or GM WM

Operation (Figure 9)

The op modifier register sets to 1 2 $\overline{4}$ 8 A B $\overline{C}$ to cause the 1410 to function as indicated in the preceding paragraph. After this is done, alternate A and B cycles perform the operation. Details of this op code are covered in *IBM Customer Engineering Instruction-Reference, 1410 Data Processing System* (Form 223-2599).

The 1- and 2-bit lines from the op modifier register bring up the assembly controls that gate to storage in the B-field location the desired portions of the A- and B-channel characters with the proper parity. The next character of the A field reads out on the following A cycle, and the operation continues by alternate A and B cycles, until a WM reads out and combines with the $\overline{8}$ A B bits from the op modifier register to end the operation.

Move Characters to A or B Word Mark M (A) (B), M (A)

The M-op code moves data in storage from the A-field address to the B-field address. The first word mark encountered in either field stops the operation. In the case of a single address instruction, the B address is taken from the BAR and does not have to be written or interpreted as part of the instruction. The first WM encountered stops the operation as in the "2 address instruction."

Op-Modifier Function

The 1401 M-op does not have a modifier. The appropriate 1410 modifier must be set in the op-modifier register to get the desired transfer. The following 1410 functions must be performed:

1410 FUNCTIONS	OP-MODIFIER BITS
Transfer A-field numeric	Bit 1
Transfer A-field zones	Bit 2
Retain B-field WM	Bit 4
Set first scan	Bit 8
End operation at first WM A or B field	Bit 8 A B

Operation

During the I2 cycle the op modifier register sets to 1 2 $\overline{4}$ $\overline{8}$ A B C to cause the proper 1410 functions as indicated in "Op-Modifier Function." Also, the controls set up for an A cycle. The first character of the A field reads out during the A cycle, and sets into the A-data register. The next cycle is a B cycle, during which the first character of the B field reads out and is placed on the B channel.

Load Characters to A-Word Mark L (A) (B), L (A)

The L-op code moves the data and the word mark in storage from the A-field address to the B-field address. The A-field word mark stops the operation. B-field word marks clear unless the B field is larger than the A field.

Op-Modifier Function

The 1401 L-op does not have a modifier. The appropriate 1410 modifier must be set in the op-modifier register to effect the desired transfer. The following 1410 functions must be performed:

1410 FUNCTIONS	OP-MODIFIER BITS
Transfer A-field numeric	Bit 1
Transfer A-field zones	Bit 2
Transfer A-field word mark	Bit 4

Set the first scan-and-end-operation at first A-field WM: Op mod $\overline{8}$ A $\overline{B}$ bit.

Operation

During the I2 cycle the op modifier register sets to 1 2 4 $\overline{8}$ $\overline{A}$ $\overline{B}$ C to cause the proper 1410 functions as indicated in "op modifier function." Also, the controls set for an A cycle, and the first character of the A field reads out and sets in the A-data register. The next cycle is a B cycle, during which the first character of the B field reads out and is placed on the B channel. The 1-, 2-, and 4-bit lines from the op-modifier register bring up the assembly controls that gate to storage, in the B-field location, the numerical, zone and WM of the A-channel characters that have the proper parity. B-field WM, if any, clears from storage.

The next character of the A field reads out on the following A cycle, and the operation continues by alternate A and B cycles until an A-channel WM combines with the $\overline{8}$ A $\overline{B}$ bits from the op modifier register to end the operation.

Move Characters and Suppress Zeros Z (A) (B)

The function, operation and circuits of the execute phase of Z (A) (B) in 1401 mode are the same as the standard 1410 operation. NOTE: on Systems 13.13.03, Z-op decode is common to both operations. Details of this op code, and the following two op codes are covered in *IBM Customer Engineering Instruction Reference, 1410 Data Processing System* (Form 223-2599).

Compare C (A) (B)

The function, operation, and circuits of the execute phase of C (A) (B) in 1401 mode are the same as the standard 1410 operation except for set of high-low-equal that does not occur in the 1401 mode for single-character C-ops. C-op decode (13. 13. 03) is common to both operations.

Move Characters and Edit E (A) (B)

The edit operation in 1401 mode is acomplished exactly as in 1410 program execution. E-op decode (13. 13. 02) is common.

Add and Subtract

These op codes operate the same as in 1410 mode except for the sign, zones and overflow controls.

Sign

If the signs of the fields and type of op code require a true-add, the zones from the units position of the B field are used as the sign of the result.

If a complement-add is required, the sign of the B field is used as the sign of the result but it is inserted in the standard form (B bit only, minus; AB bits, plus). If a recomplement cycle is required, the sign of the B field is inverted to a standard sign.

Zones

All zones in the B field are removed, except the units position sign or high-order position on true-add.

Overflow

An adder carry during the B cycle, in which a B-channel word mark is sensed, indicates an overflow condition. This sets the arithmetic overflow latch, which is testable, by a branch instruction.

The adder carry is also gated to the zone adder, where is it combined with any A- or B-channel zones. The zone adder output is gated through the assembly and is stored in the high-order position of the B field. For example:

```
                    v
A-field             8 0 2
                    v
B-field             2 9 1
                    v
                    A
Arith Overflow <--  0 9 3
```

If the B field high-order position contains A and B bits prior to an overflow, the arithmetic overflow latch will be set, but there will be no zone bits over the result in the B field.

Zero and Add, Zero and Subtract

The adder is not used for these op codes. The numeric portion of the A channel is gated through assembly. The sign of the A field is used on zero and add, and is inverted on zero and subtract. No other zones are used. Zeros are inserted on the assembly channel for all extra, high-order positions of the B field.

Multiply and Divide

These op codes operate the same as in 1410 mode. The arithmetic overflow latch sets instead of the divide overflow latch.

Branch (B) B (I) b

The B-op code with an I address and a blank d-modifier or no modifier is an unconditional branch. This means that the normal program sequence is interrupted and continued at any desired point, as specified by the I address, without testing for specific conditions.

Operation (Figure 10)

When a branch is required, the CPU takes a B cycle. During the B cycle, the IAR reads out to the STAR through the modifier to the BAR. A modify-by-zero condition exists in the modifier because the no-scan latch is set.

The 1401 branch latch sets to cause the AAR to read out to STAR for the I-op cycle of the instruction read-out.

This is the start of normal instruction read-out for the first branch routine instruction.

Branch, if Indicator On B (I) d

Op-Code Function

The d-character modifier specifies the condition to be tested. If the indicator tested is off, the next instruction in sequence is taken. If the indicator tested is on, the program branches to the I address.

Figure 11 illustrates the d-modifiers and the indicators they test.

Operation (Figure 10)

At I-ring-8 time, during instruction read-out, the op-modifier register sets with the op-modifier character and is decoded.

The op-modifier decode compares with the indicators and at I-ring-9 time (last instruction RO cycle) either the branch or no-branch condition is brought up. "No-branch" causes the last execute cycle that initiates the instruction read-out of the NSI. The 1401 branch latch causes the extra B cycle, as covered in the section "Branch (B)."

Branch, if Character Is Equal B (I) (B) d

Op-Code Function

This instruction code causes the single character at the B address to compare to the d-character. If the bit configuration is the same, the program branches to the I-address. If it is not the same, the program continues in sequence. The d-character can be any combination of the six BCD code bits (A B 8421).

Operation (Figure 12)

At I-ring-11 time, during instruction read-out, the op-modifier register sets with the op-modifier character. At I-ring-12 time, the first scan-control latch sets and a B cycle starts. The character at the B-field address reads to the B channel. The op-modifier register is gated to the A channel and a comparison is made between A and B channels in the compare unit. If the equal latch sets, the program branches to the I address in the AAR. In all other cases the program continues to the NSI.

Branch, if Bit Equal W (I) (B) d

Op-Code Function

This instruction compares the character at the B address, bit by bit, with the d-character. If any bit in the character at the B address matches any bit in the configuration of the d-character, the program branches to the I address. The d-character can contain any character or any combination of bits that can exist in a single position of core storage. WM and C bits are not compared.

Operation (Figure 12)

At I-ring-11 time, during instruction read-out, the op-modifier register sets with the op-modifier character. At I-ring-12 time, the first scan-control latch sets and a B cycle starts. The character at the B-field address reads to the B channel. The op modifier register is gated to the A-channel and the bits compare. Any bit on both channels starts a branch.

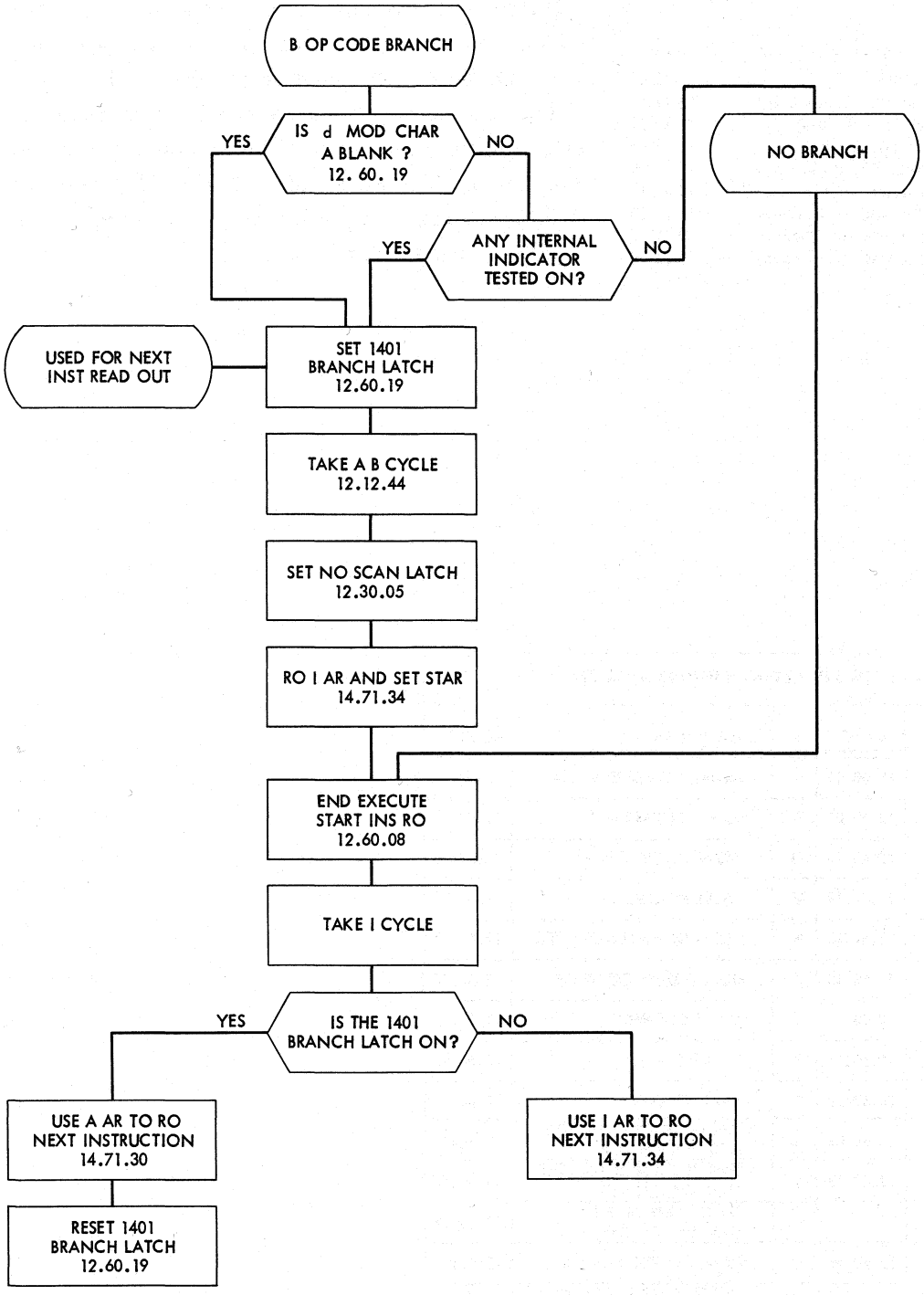


Figure 10. Branch Unconditional and Branch if Indicator On

Branch, if WM and/or Zone V (I) (B) d

Op-Code Function

This op code tests the character that is located in the B address for the condition specified by the d-character, and branches to the I address, if the condition is met. The d-modifiers and the conditions they specify are as follows:

d-CHARACTER	BITS	CONTROL	LOGIC
1	1	Word mark	13.12.17
2	2	No zone (no A, no B bit)	13.12.17
B	B A 2	12-zone (A · B bits)	13.12.11
K	B 2	11-zone (B, A bit)	13.12.16
S	A 2	Zero-zone (A, no B bit)	13.12.10
3	2 1	Either a WM or no-zone	13.12.17
C	B A 2 1	Either a WM or 12-zone	13.12.15
L	B 2 1	Either a WM or 11-zone	13.12.16
T	A 2 1	Either a WM or zero-zone	13.12.10

Operation (Figure 13)

This operation duplicates the B-op code until the character designated by the BAR reads to the B channel. At this point, the bits on the B channel are tested for the configuration specified by the d-modifier. The d-modifier was gated to the A channel.

If the conditions imposed by the d-modifier (see the table in the preceding paragraph) are met, a branch starts. Otherwise, the program proceeds with the NSI.

The test of the B-address character is made by switching the result of zone bit compare ($=$ or $\neq$) with the op modifier 1 and/or 2 bits. The choice of d-characters makes it possible to test for several combinations of WM and zone bits.

CHARACTER AT d FOR B (I) d BRANCH IF INDICATOR ON					
d	BRANCH ON	LOGIC	d	BRANCH ON	LOGIC
bl	UNCONDITIONAL	12.60.01	S	EQUAL COMPARE B = A	12.60.01
9	CARRIAGE CHANNEL #9	12.60.01	T	LOW COMPARE B<A	12.60.01
A	"LAST CARD" SWITCH	12.60.18	U	HIGH COMPARE B>A	12.60.01
B	SENSE SWITCH B	12.60.18	V	VALIDITY CHECK	12.60.17
C	SENSE SWITCH C	12.60.18	W	FILE WRONG LENGTH REC	12.60.17
D	SENSE SWITCH D	12.60.18	X	FILE ADDRESS COMPARE	12.60.17
E	SENSE SWITCH E	12.60.18	Y	ANY FILE CHECK	12.60.17
F	SENSE SWITCH F	12.60.18	Z	OVERFLOW	12.60.02
G	SENSE SWITCH G	12.60.18	*	INQUIRY CLEAR	12.60.17
K	END OF REEL	12.60.05	?	READER ERROR IF I/O CHK STOP SWITCH OFF	12.60.18
L	TAPE ERROR	12.60.17	!	PUNCH ERROR IF I/O CHK STOP SWITCH OFF	12.60.17
N	ACCESS INOPERABLE		‡	PRINTER ERROR IF I/O CHK STOP SWITCH OFF	12.60.17
P	PRINTER BUSY	12.60.17	@	CARRIAGE CHANNEL #12	12.60.01
Q	INQUIRY REQUEST	12.60.02	%	PROCESSING CHECK WITH PROCESS CHK SWITCH OFF	12.60.18
R	CARRIAGE BUSY	12.60.15	/	UNEQUAL COMPARE B ≠ A	12.60.01

Figure 11. Character At d for Branch If Indicator On

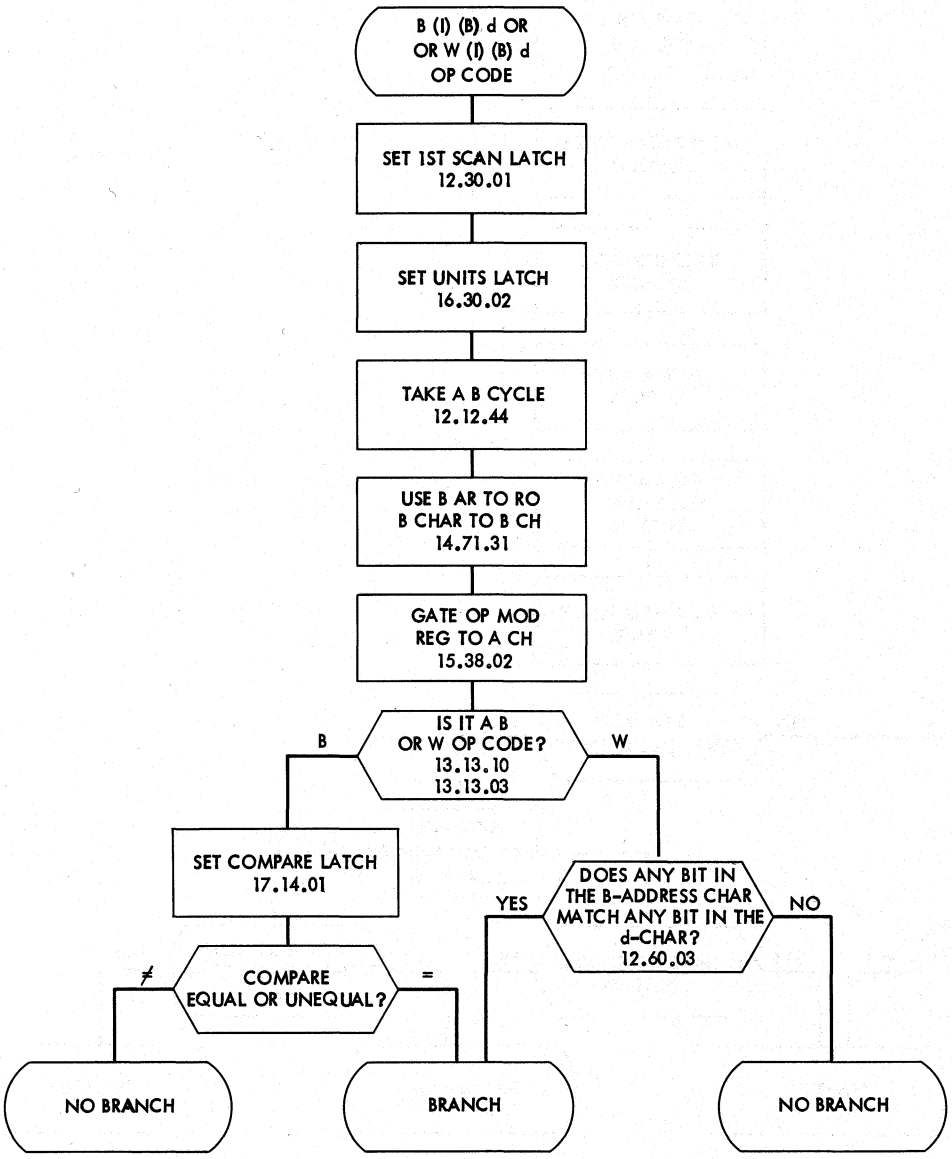


Figure 12. Branch if Character is Equal; Branch if Bit is Equal

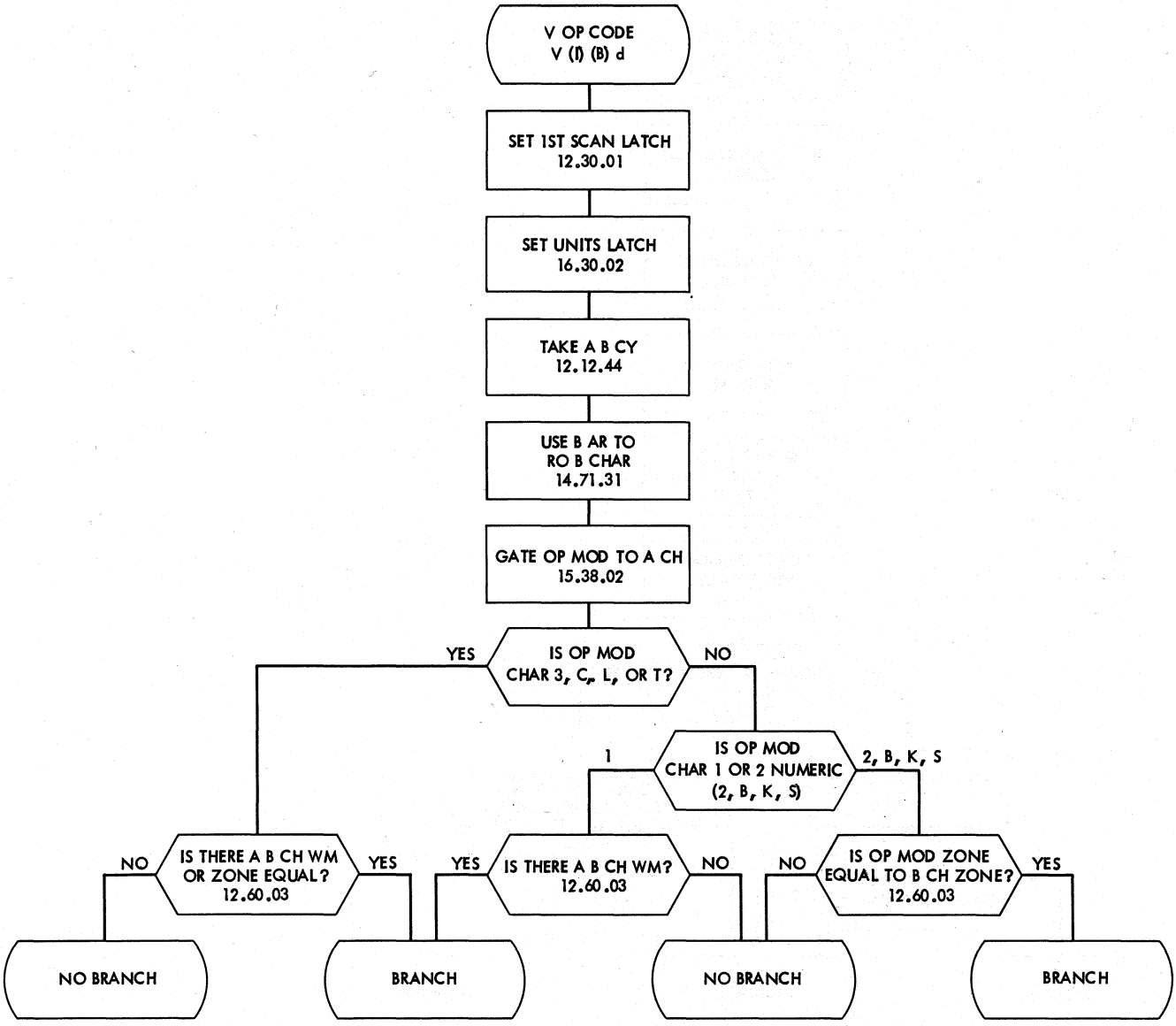


Figure 13. Branch if WM and/or Zone

Store A—Address Register

Op-Code Function

The 1401 Q-op code stores the contents of the A-address register from the previous operation in the three-position field that has its units position defined by the A address of the store-A-address-register instruction. Word marks in the A-field are retained. The instruction format is: Q (AAA).

Operation (Figure 14)

The A address of the Q-op code reads into the AAR on the instruction read-out cycle. The AAR, however, contains the address that must be stored. Therefore, the A address left from the previous operation must move to the BAR before the new A address reads in. This is done during a B cycle that occurs just after the Q-op is detected at I-op time. After the B cycle, the I ring advances, and the (AAA) address reads into the AAR.

The BAR contains a five-digit numeric 1410 address. In 1401 mode, a three-digit address with appropriate zones over the units and hundreds positions must be stored. The units, tens and hundreds positions of the BAR are stored on A cycles during following steps of the A ring. The zones over the units and hundreds that represent thousands and ten-thousands position values must be stored at the same time as the numeric portion of the character. To do this, the thousands and ten-thousands positions of the BAR read out to the binary register during two C cycles before the storage operation (Figure 15).

During the storage operation, at A-ring-2 time, the units position of the BAR reads out via the address exit channel to the A-data register, to the A channel, and to assembly. At the same time, the A4 and A8 binary registers are gated out to the zone adder. The zone adder output is gated to the assembly area to combine with the numeric bits from the A channel. The resulting character reads into storage in the location addressed by the AAR.

At A-ring-3 time, the tens position numeric bit reads out of the BAR, via the A channel, to storage. No zones are involved. At A-ring-4 time, the hundreds position numeric bit reads out of the BAR to assembly the same as the units position. The binary-register A1 and A2 outputs are gated through the zone adder to assembly to form the zone portion of the character to be stored.

Store B—Address Register

Op-Code Function

The 1401 H-op code stores the contents of the B-address register from the previous operation in the three-position field that has its units position defined by the A address of the store-B-address-register instruction. Word marks in the A field are retained. The instruction format is: H (AAA).

Operation (Figure 14)

The (AAA) address reads into the AAR in the usual manner. Because the address to be stored is already in the BAR, no transfer from the AAR to the BAR is needed. Thus, the B cycle after I-op time is not necessary. The operation proceeds in the same manner as the Q-op code.

Modify Address

Op-Code Function

The # (AAA) (BBB) instruction adds the three-character field specified by the A address to the B-address field. The three numeric positions and zones over the units and hundreds positions of each field added, and the three-position result (including appropriate zones), are stored in the B field. Word marks are ignored and remain unchanged in both fields.

Operation (Figure 16)

Three sets of A and B cycles add the address from the B field to the modifier in the A field. The modified address is then stored back in the B field. The zone bits over the B-field units and hundreds positions must be added to the zone bits over the modifier. The problem is illustrated in the following example:

		A = 1 B = 2 B = 8
Contents of B address to be modified		9 6 5
		B = 2
Contents of A address modifier		0 0 0
		A = 1 A = 4 B = 8
	Sum	9 6 5
In 1410 numerical form	11965	
	+ 2000	
	13965	

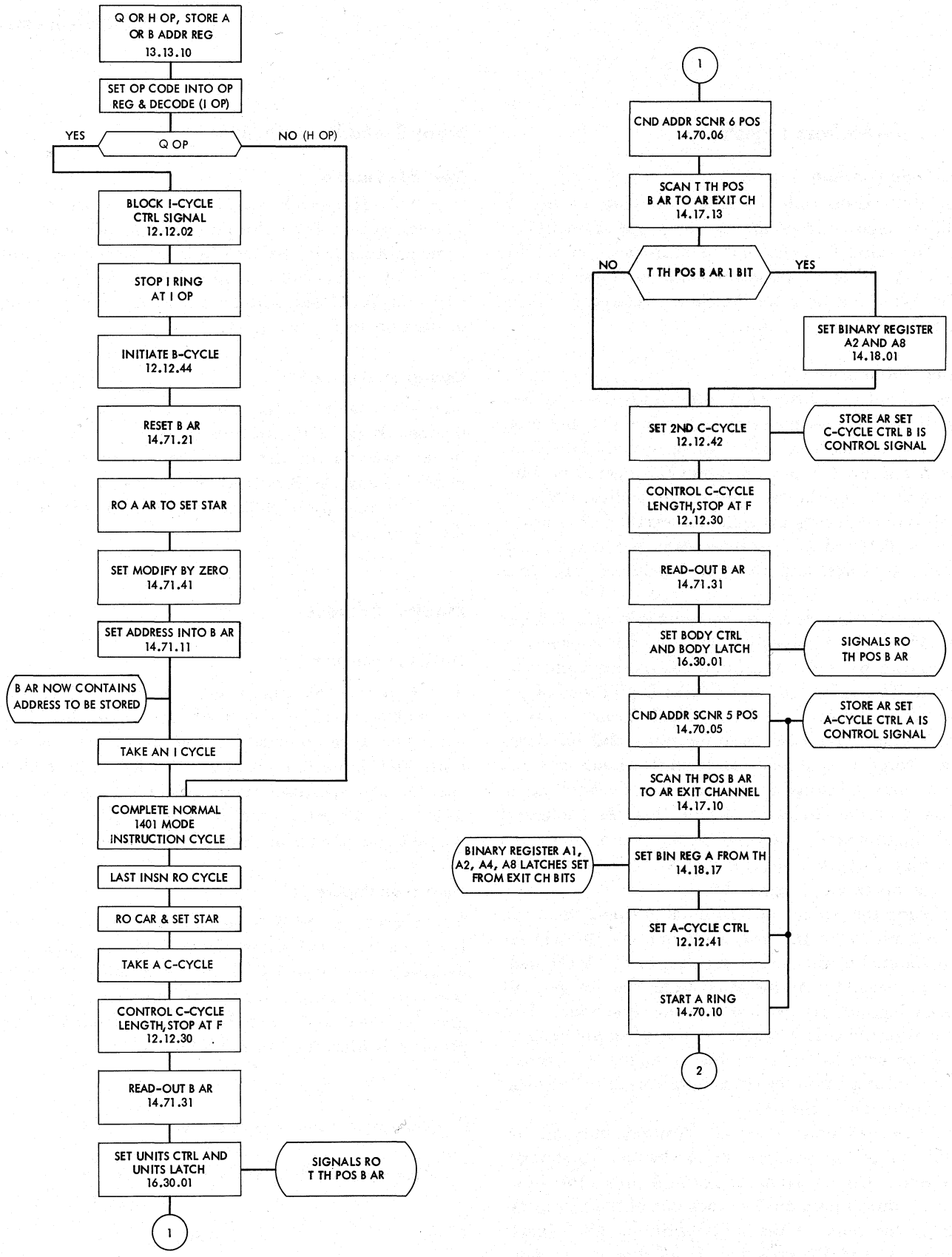


Figure 14A. IBM 1401 Mode Store AAR or BAR

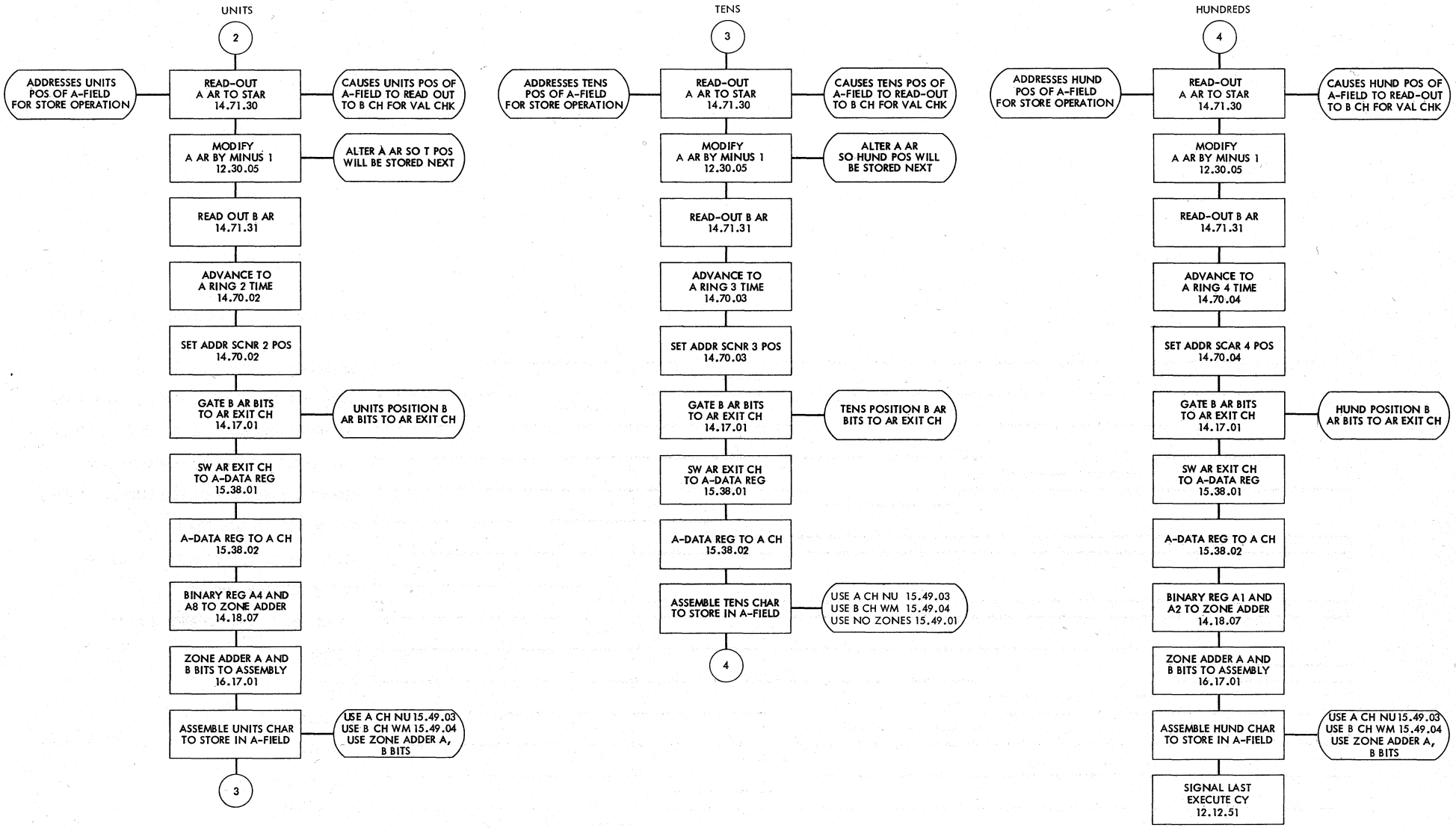


Figure 14B. IBM 1401 Mode Store AAR or BAR

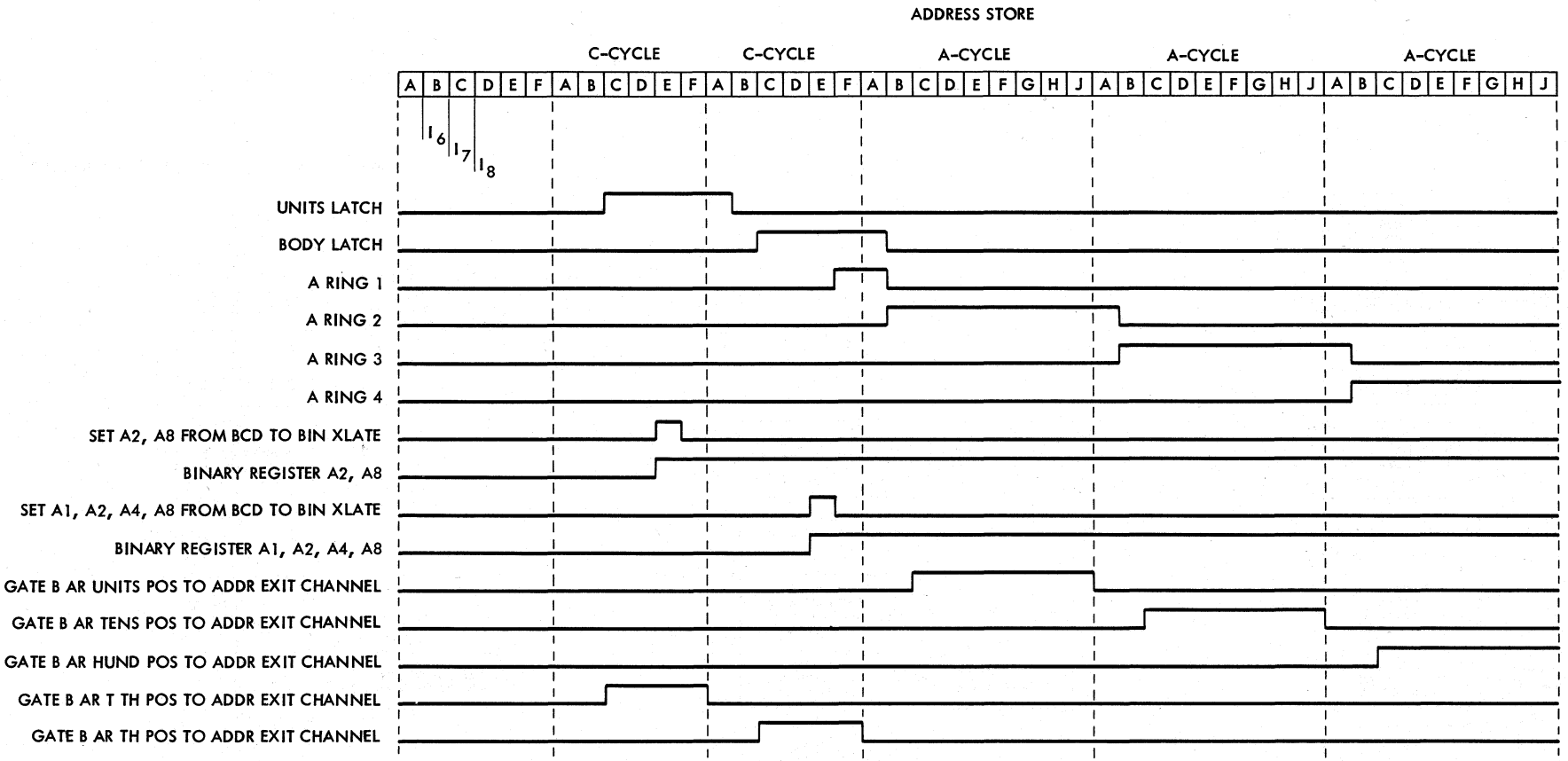


Figure 15. 1401 Mode Address Store Sequence

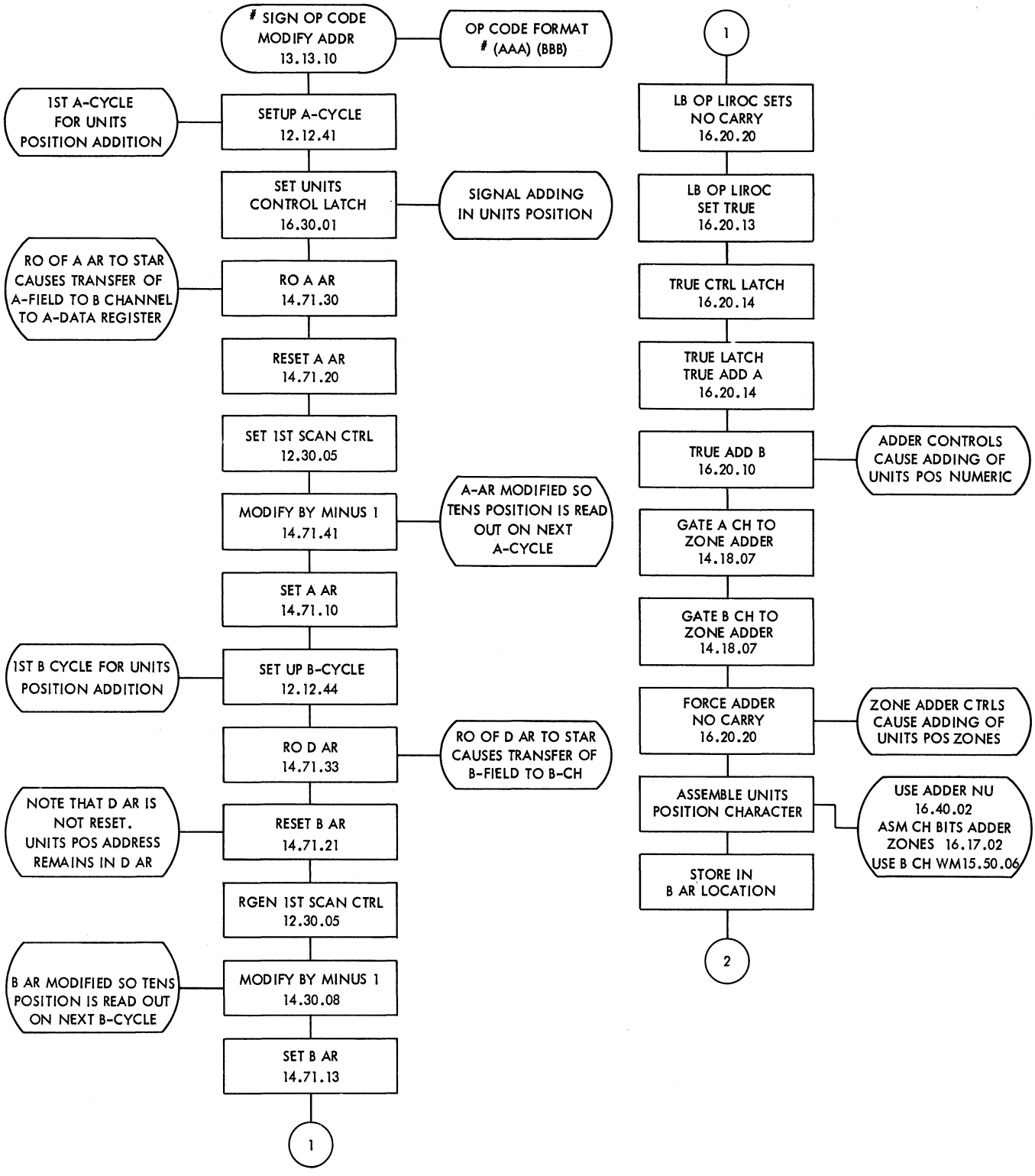


Figure 16A. 1401 Mode Modify Address

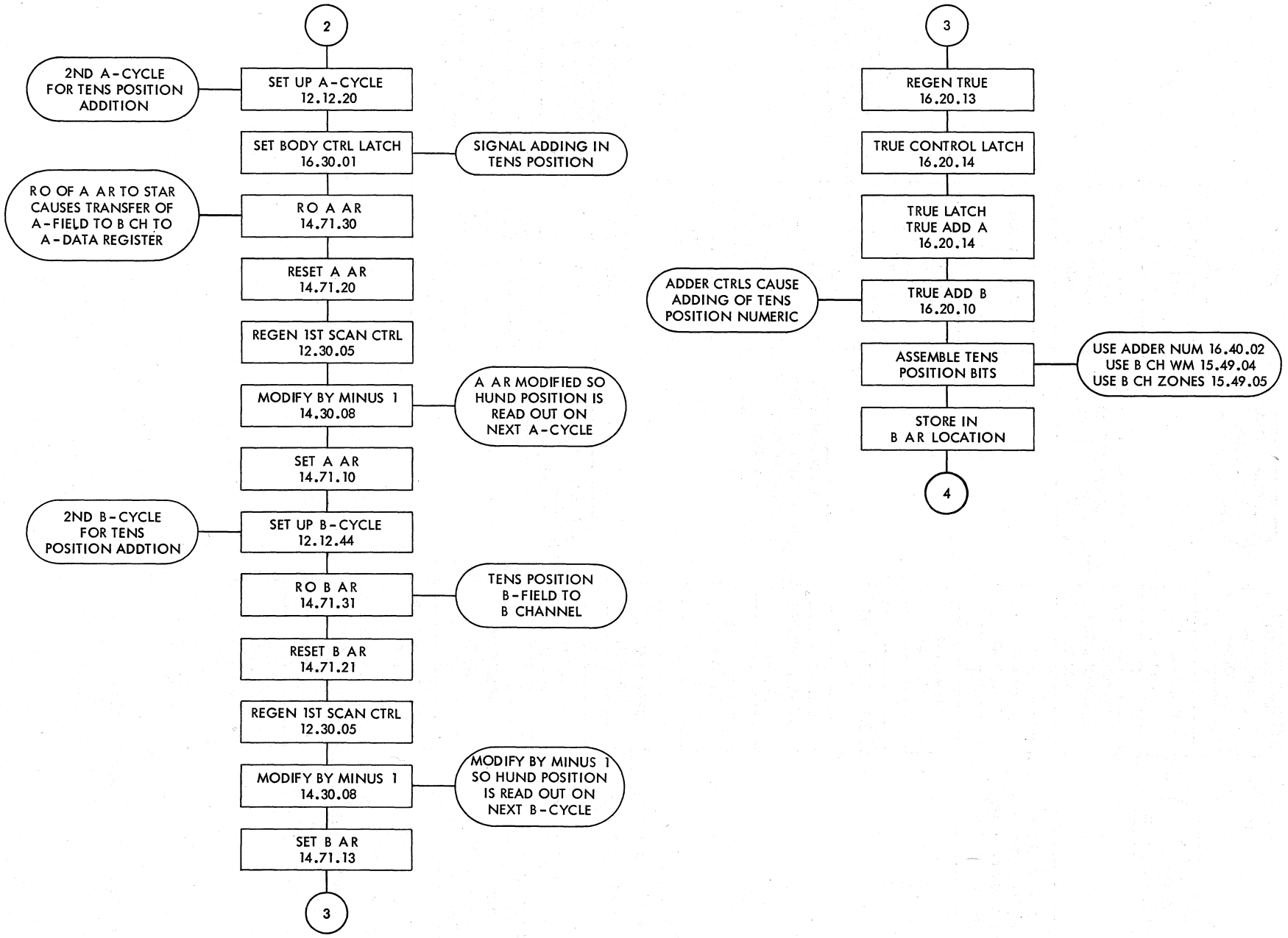


Figure 16B. 1401 Mode Modify Address

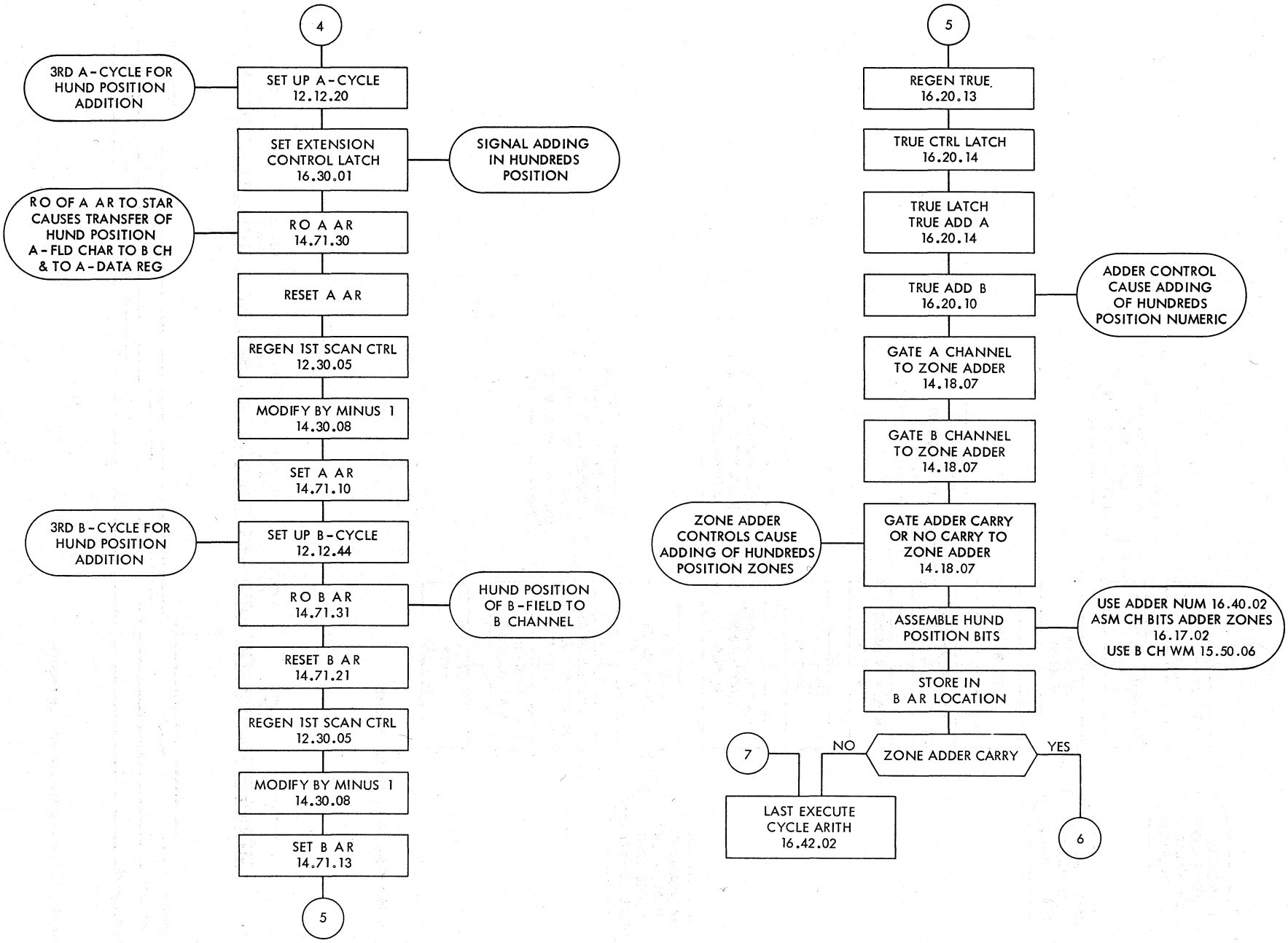


Figure 16C. 1401 Mode Modify Address

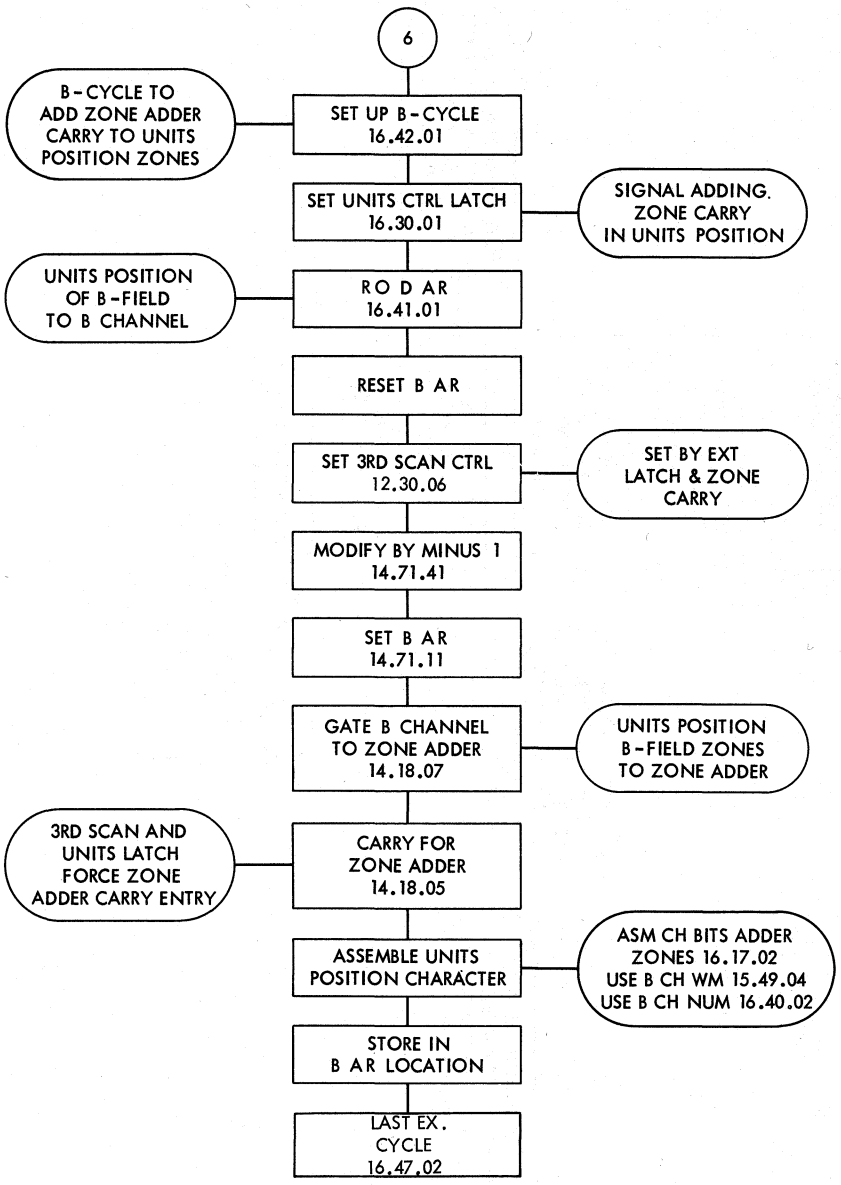


Figure 16D. 1401 Mode Modify Address

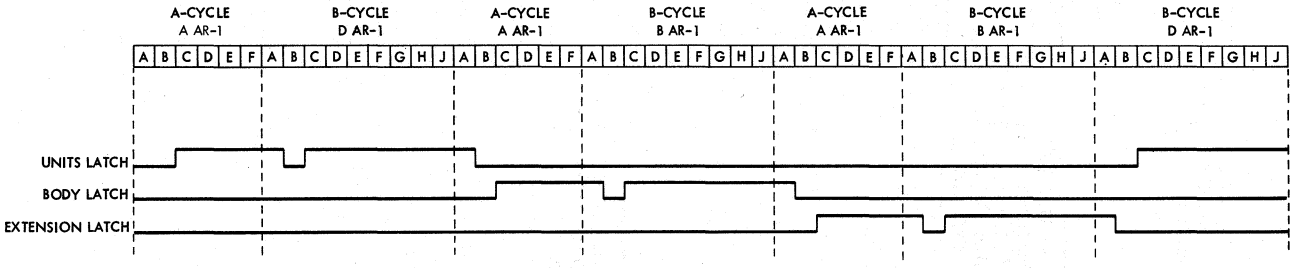


Figure 17. 1401 Mode Modify Address Sequence

The following sequence of events takes place (Figure 17).

1. First A and B cycles. The units-position 5 and 0 digits add in the 1410 adder. The result switches to the assembly channel. The B-zone bit on the B channel adds to no-bit in the zone adder. The resulting B bit is gated to the zone assembly by a forced adder no-carry.

2. Second A and B cycles. The tens-position 6 and 0 digits add in the 1410 adder. The result switches to the assembly channel. Any zone bits (index tags) over the 6 in the B field are gated from the B channel to assembly. No zone adder operation is involved.

3. Third A and B cycles. The hundreds-position 9 and 0 digits add. The result switches to the assembly channel. The A and B zone bits on the B channel add to the B zone bit on the A channel in the zone adder. The resulting A bit is gated to the zone assembly by the adder no-carry. In addition, a zone-adder-carry results. This must be added to the zones in the units position. A B cycle is taken to do this.

4. Special B cycle, caused by zone-adder-carry. On the first add cycle (B-cycle porition, the DAR reads out to address storage, but does not reset. Instead, the BAR resets and is modified, and the DAR remains unchanged, with the units-position B address intact. This address is used on the special B cycle to read out the units position of the B field to the B channel. The previously stored B bit then adds to the zone-adder-carry to result in the A and B bits. These are gated to the assembly channel along with the B-channel numeric bits.

5. Signal last-execute cycle. The last-execute cycle signal develops after the third add cycle, if no zone-adder-carry occurs. If a zone-adder-carry takes place, the last-execute cycle signals after the special B cycle caused by the zone-adder-carry.

The units, body, and extension latches signal which position is added:

LATCH	POSITION ADDED
Units	Units position
Body	Tens position
Extension	Hundreds position

No Operation (N)

Op-Code Function

The N-op code performs no operation. It can substitute for the op code of any instruction to make that instruction ineffective.

Operation (Figure 3)

When the op registers detect a "no-op" op code, they allow the I ring to advance and I cycles to be taken until the next op code, as defined by a word mark, reads out of storage. The word mark conditions "last insn RO cycle," "last I-cycle," and "last execute cycle" to start the next instruction read-out operation.

Halt (.)

Op-Code Function

The . op code stops the machine and starts a stop print-out operation. Pressing the start key starts the program at the next instruction in sequence.

Operation (Figure 18)

The last instruction read-out cycle (determined by the presence of a B-channel WM bit at I-ring-1 time) causes a simultaneous last-execute cycle. This sets the stop latch, stops the logic clock, and starts a stop print-out operation.

Halt and Branch op . (I)

Operation (Figure 18)

The I address reads into the AAR. A B cycle then starts. During the B cycle the IAR reads out to the STAR, and through the modifier (set to modify-by-zero condition) to the BAR. The last-execute cycle stops the logic clock and starts the stop print-out.

When the start key is pressed, the 1401 branch latch sets to read out the AAR to STAR for the I-op cycle of instruction read-out.

Set Word Mark, (A) (B)

Op-Code Function

The , op code causes a word mark to set at each address specified in the instruction. The data at each address are undisturbed. If this instruction is given with one address (A address), a word mark is set at the A address only. If the address instruction is indexed, a word mark sets at the locations specified by the indexed A address. However, if this instruction is given with no A address (no-address instruction), word marks are set at the locations specified by the addresses in both the A- and B-address registers (contents from previous operation).

Operation

One A and one B cycle are required to execute this operation. During A cycle, the A-address character reads from storage to the B channel. It is then gated through the assembly and back into storage, accompanied by a WM bit. Also, at this point, a check bit is added or removed to maintain parity.

During B cycle the character at the B address reads from storage and is treated similarly. If only one address is specified, the A-data address is stored in both the A- and B-address registers during instruction RO. Therefore, when the execute phase begins for a single address instruction, the A-data address is in both the AAR and BAR. Thus a WM is set in the same location twice, once during A cycle, once during B cycle.

If this instruction is given with no address, WM's are set in A- and B-address locations specified in the previous instruction. The operation is the same as a two-address instruction.

Clear Word Mark□(A) (B)

Op-Code Function

The □ op code causes word marks cleared from the locations specified by the A and B addresses of the instruction. The data at each address are undisturbed. If this instruction is given with one address, a word mark clears at the A-address only. If the instruction is indexed, a word mark clears at the location specified by the indexed address. However, if this instruction is given with no addresses, word marks clear at the locations specified by the addresses in both A- and B-address registers from the previous operation.

Operation

This operation is executed in much the same manner as set word mark. One A and one B cycle are taken. The character to be treated is handled in the same manner. The same op-code grouping lines are activated. The significant difference is that the WM bit is removed (instead of being set) as the character is gated through assembly to storage.

This is accomplished by controlling the assembly to "use no WM."

Clear Storage / (A)

Op-Code Function

The / op code causes as many as 100 positions of core storage to be cleared of data and word marks. Clearing starts at the A address and is forward-scanned to the nearest hundreds position. The cleared area sets to blanks.

Operation (Figure 19)

The operation is executed by a series of B cycles, during which the assembly is controlled to "use no WM's," "Zones," or "Numerics." Clearing starts with the location specified by the A address, and continues to the nearest even-hundreds position. The "Mod by Minus-one" condition that is still on at logic gate D-time recognizes the even-hundreds address. This indicates a borrow-one-from-hundreds position. It sets the even-hundreds latch, and ends the execute phase.

When this instruction is given without an address, the contents of the B-address register from the previous operation are used as the A-address.

Clear Storage and Branch / (I) (B)

Op-Code Function

This instruction performs the same operation as / (A), except that the clearing starts at the B address. The I address specifies the location of the next instruction. This is an unconditional branch.

Operation (Figure 19)

As described for the clear storage operation, "clear storage and branch" is executed by a series of B cycles. Because the last instruction RO cycle occurs at I-ring-11, the 1401 branch latch is set on. When the even-hundreds address sets ON and is gated with the 1401 branch latch on, "no scan control" sets and another B cycle is taken to store the address in the IAR. The next instruction reads out, starting from the address specified by the AAR.

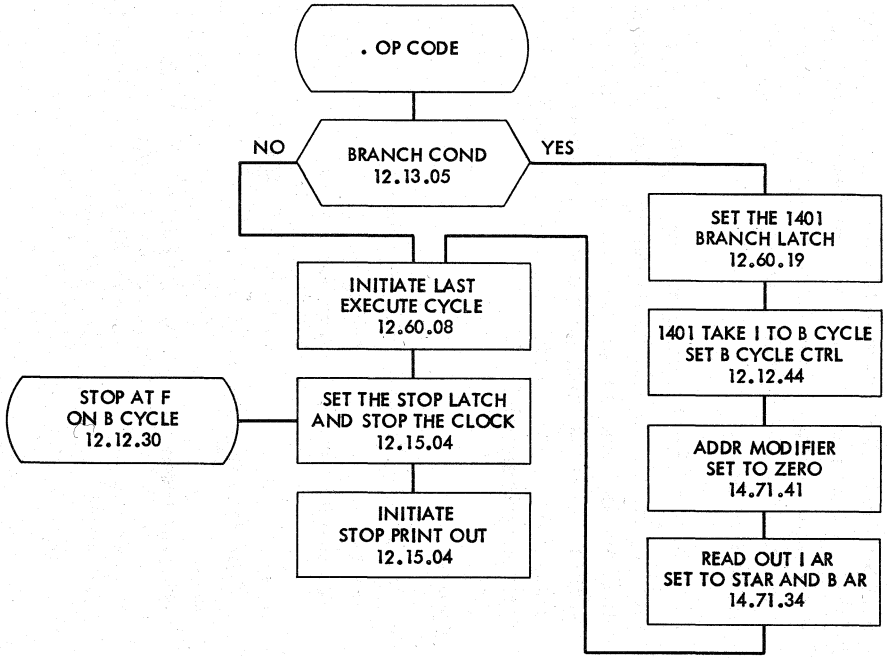


Figure 18. Halt, Halt and Branch

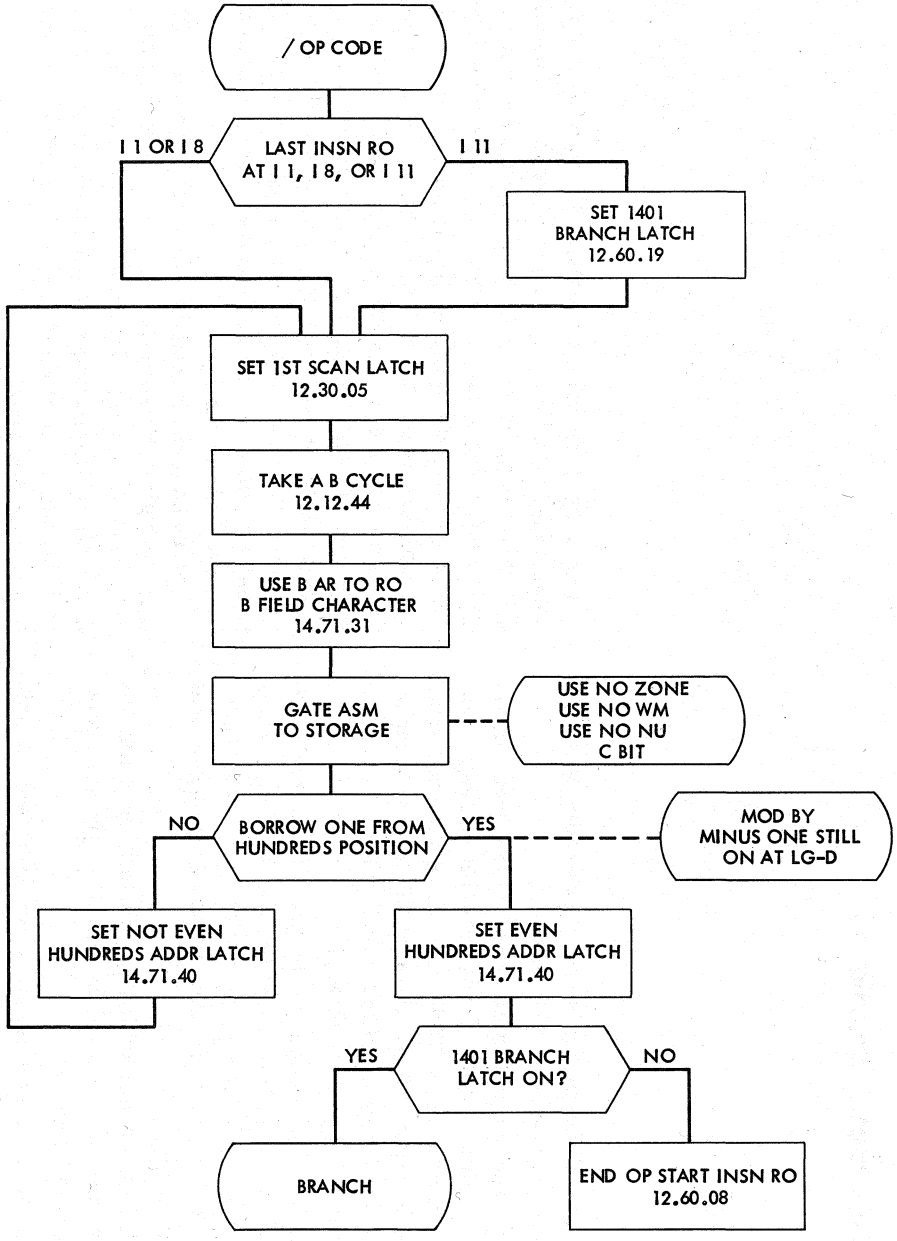


Figure 19. Clear Op Code

Four major differences arise when processing IBM 1401 I-O instructions on the IBM 1410:

1. Specific input-output areas are assigned in the CPU core storage unit. Cards read into storage locations 00001 through 00080. Cards are punched from storage locations 00101 through 00180. Information prints from storage locations 00201 00332. No. GM/WM is required to define the I-O area.

2. A 1401 instruction can be given to address more than one I-O unit. For example, the 1401 op code 7 causes the printer to print a line, the card reader to read a card, and the card punch to punch a card (in that order).

3. A 1401 I-O instruction can include an I address to cause an unconditional branch after the I-O operation is completed.

4. If an I-O error occurs or if an addressed I-O unit is not ready or is busy, the CPU stops and waits either until the operator corrects the not-ready or error condition, or until the addressed unit becomes not busy. This means that the 1410 I-O channel status indicators are never set when operating in the 1401 mode. I-O interlock is never set in 1401 mode.

Read a Card

Op-Code Function (1)

The 1401 read-a-card op code 1 causes all 80 columns of information to read into core storage locations 00001 to 00080.

Operation (Figures 20, 21 and 22)

A reader operation in 1401 mode causes the contents of the 80-position read-storage unit in the synchronizer to transfer to the core storage unit in the CPU. The starting address, 00001, is generated in the CPU. 1401 instructions cause the 1410 to operate in the move mode; there is no translation of word-separator characters. Word marks that previously were set in 00001 through 00080 read out on the B-channel, combine in assembly with the characters from read storage, and read back into the CPU core storage unit.

The 1 op code brings up the 1401 card-print ops signal to indicate that one or more of the input-output operations is to be performed. At last instruction RO cycle time, a 4-position I-O scan-ring scans the op register 2 (print), op register 1 (read), and op register 4 (punch) lines in sequence. If the tested op register output is not up, the ring advances at the next 1st-clock-pulse time to test the next op register output in sequence. Since the op register 1 line is up, the scan-ring advance stops, and the read trigger (13.70.01) remains on until the record from read storage transfers into CPU core storage.

The E-channel is selected automatically by the 1401 read operation. Because the 1401 does not have status indicators, E-channel status indicators are not set before starting transmission. If the reader is not ready or busy, the CPU stops and waits until the I-O device is ready or not busy.

The E-channel unoverlap-in-process latch sets so the transmission takes place via the E-channel in the unoverlap mode. The address 00001 is generated and gated to the BAR. The transfer of data takes place in the usual manner, and the BAR is updated on each B cycle. After the transfer, the next card reads into reader storage.

When the buffer, end-of-transfer signal returns to the CPU, the E-channel "status sample B delay" comes ON. This, in turn, advances the scanning to the end position provided that no punch operation is indicated. Input-output controls turn OFF, and a last-execute cycle signals.

Circuits

1. Recognize the card-print operation:

SIGNAL	CONTROL	LOGIC
1401 card print ops	(not) Ctrl reg disable 1401 mode Op crdr, not A, not B, not 8	13.13.11

2. Scan for print or read or punch:

SIGNAL	CONTROL	LOGIC
1401 print trigger	1401 card print ops Last insn RO cycle, LG-E	13.70.01
Not 1401 card or prtr mode	(not) Op reg 4 bit (not) Read trigger (not) Punch trigger	13.70.03

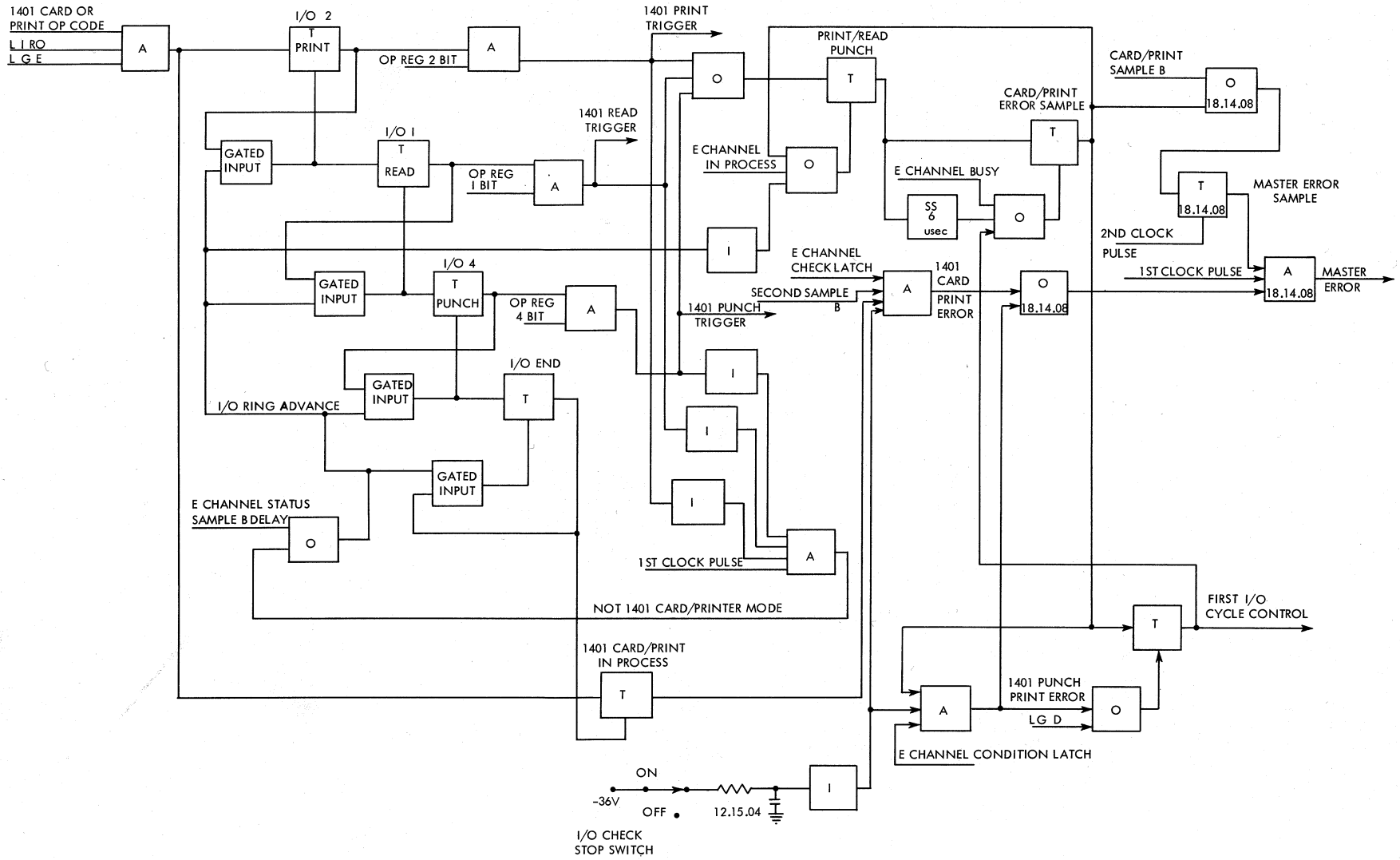


Figure 20. 1401 Mode Input-Output Ring, Buffer and Error Controls

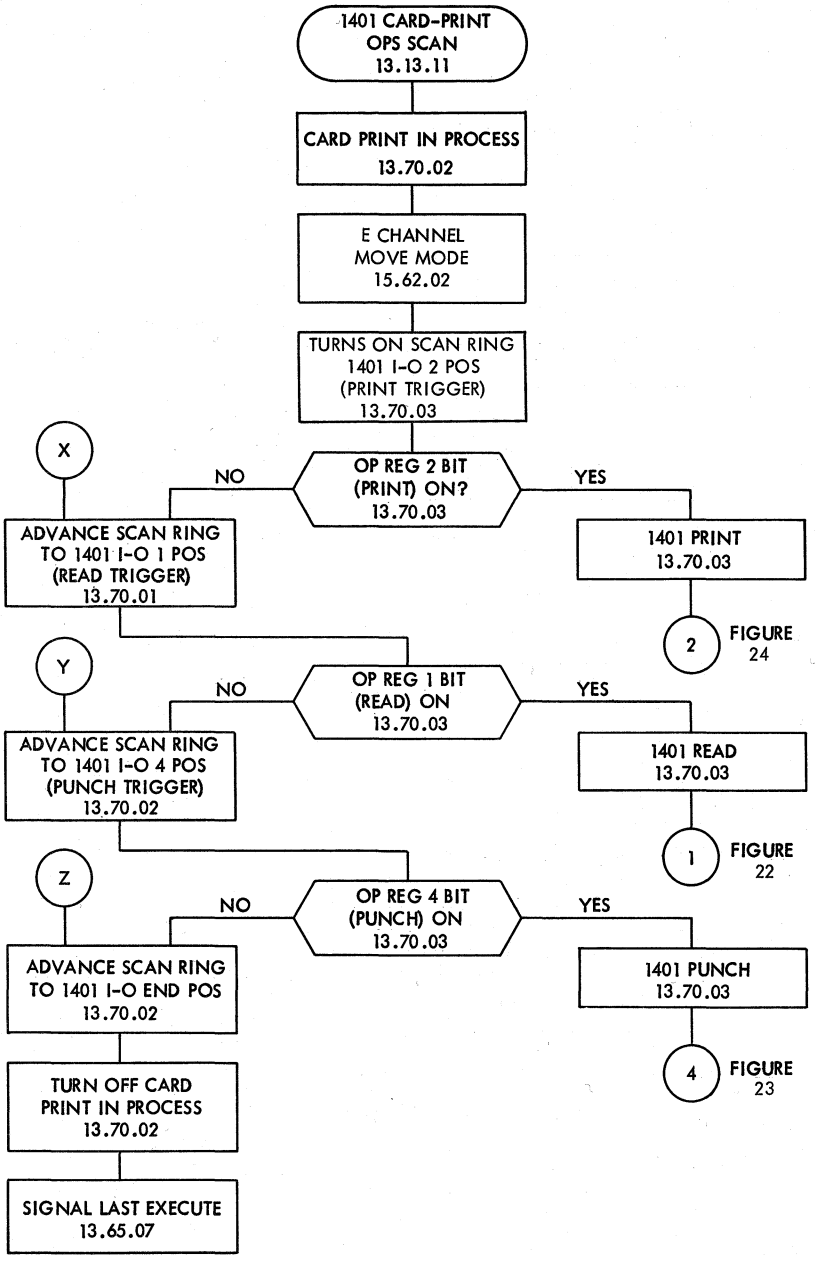


Figure 21. 1401 Card-Print Ops Scan

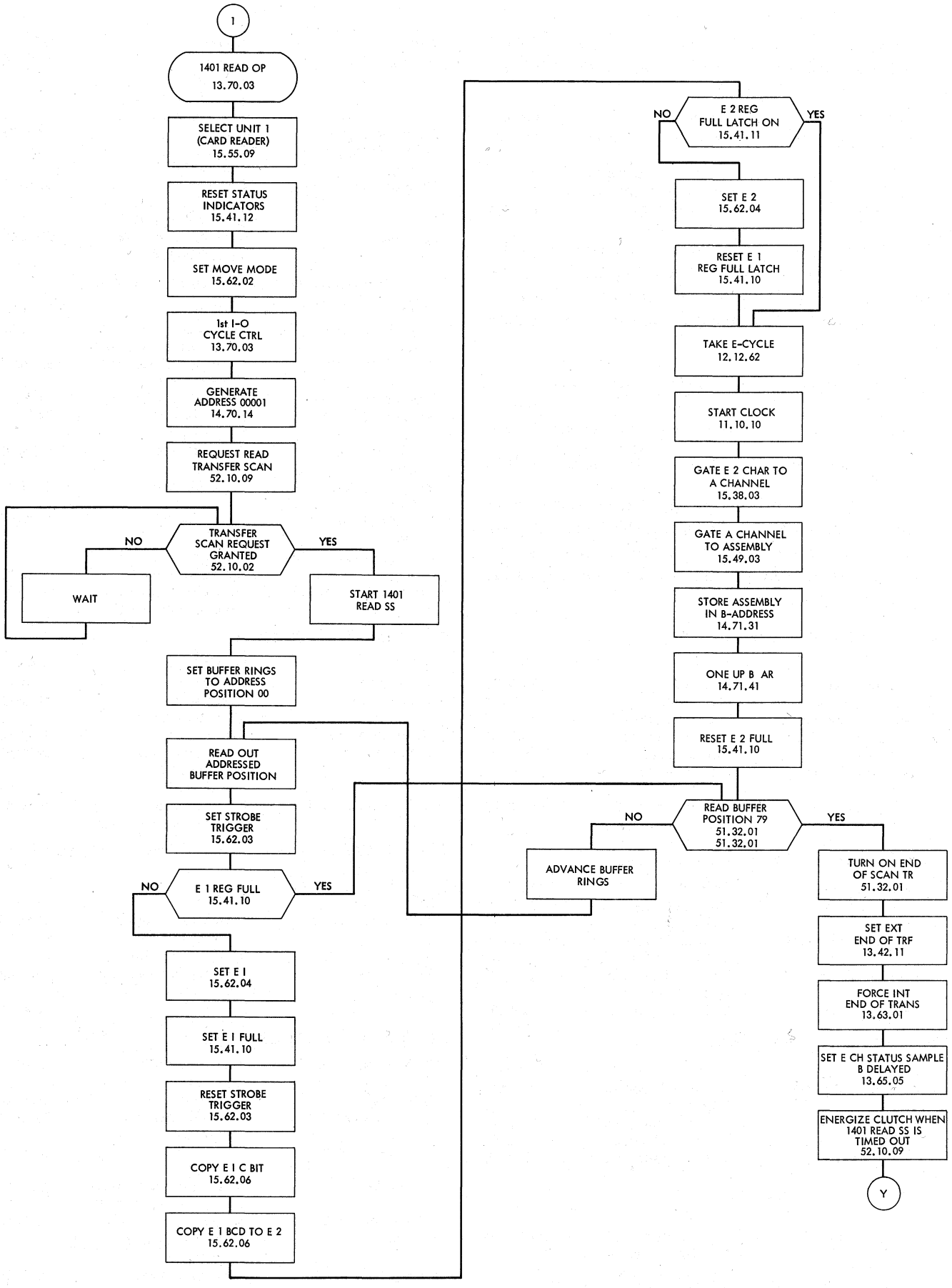


Figure 22. 1401 Mode Read Op

Because this is not a print operation ("op reg 2-bit" is not on), the scan ring advances to the 1401 I-O 1-position (read trigger).

SIGNAL	CONTROL	LOGIC
1401 I-O ring advance	(not) 1401 card or prtr	13.70.01
1401 read trigger	1401 I-O ring advance 1401 print trigger	13.70.01
1401 Read	1401 read trigger Op reg 1 Bit	13.70.03

The 1401 read signal prevents the development of the "not 1401 card" or "Prtr Mode" signal. Thus, the scanning does not advance and the read trigger remains on until read storage transfer-scans to core storage.

3. Signal the card print-in-process (i.e., read):

SIGNAL	CONTROL	LOGIC
1401 set cd print-in-proc	1401 card or print op code Last insn RO cycle, LG-E	13.70.01
1401 card print-in-proc	1401 set cd print-in-proc	13.70.02

The 1401 card print-in-process trigger remains on until the scan ring advances to the 1401 I-O end position. On a read-only operation, this occurs immediately after read storage empties, because no punch operation takes place.

4. Select E channel and unit 1:

SIGNAL	CONTROL	LOGIC
E-ch select unit 1	1401 read	13.50.03
Unit 1 select to I-O	1401 read	13.50.03
E-ch select any buffer	E-ch select unit 1	13.60.03
E-ch in mode (latch)	1401 read trigger	15.62.01
E-ch input mode	E-ch in mode	15.62.01
E-ch input op to buffer	E-ch in mode	15.62.01
Gate I-O sync to E1	E-ch sel any buffer E-ch input mode	15.62.07

5. Set up the E channel for processing:

SIGNAL	CONTROL	LOGIC
Card print reset (trigger)	1401 read, 2nd clock pulse clamped	13.70.03
1st I-O cycle control	Card print reset (trigger) 2nd clock pulse clamped	13.70.03
E-ch unovlp-in-process	1401 card print-in-process 1st I-O cycle control	13.60.04
E-ch-in-process	E-ch unovlp-in-process	13.60.04

6. Generate address 00001:

SIGNAL	CONTROL	LOGIC
RO 00001 index addr	1st I-O cycle control 1401 read trigger	14.70.14
RO fixed addr	RO 00001 index addr	14.70.14
AR bus (00001)	RO fixed addr and RO 00001 index addr to	14.15.01 14.15.12

7. Request a read transfer scan:

SIGNAL	CONTROL	LOGIC
Ready-to-buffer	E-ch-in-process E-ch select any buffer E-ch input mode	13.70.04
Ready	Ready-to-buffer	51.40.04
Rd trans req	Ready Rd ready (not) Reader busy Select unit 1	52.10.09
Rd priority gate	Rd trans req	52.10.07
Rd pr req trigger (on)	Rd priority gate Time 080-090	52.10.01
Gate on rd scan tr	Rd pr req tr (not) Same scan (not) PT scan req	52.10.01
Rd scan	Gate on rd scan tr Time 030-040	52.10.02

8. Read out the addressed buffer position:

SIGNAL	CONTROL	LOGIC
Time Pl 1 latch (on)	Read scan (not) Single cycle mode	51.30.02
Time Pl 2 latch (on)	Time pl 1 latch (on) Time 000-010	51.30.02
Timing pulse gate	Time pl 2 latch (on)	51.30.02
Rd pulse L latch (on)	Time 000-010 Not clock 2	51.30.05
Rd pulse L latch (off)	Time 040-050	51.30.05
Rd pulse S latch (on)	Time 010-020	51.30.05
Rd pulse S latch (off)	Time 040-050	51.30.05
Read pulse 1 010-040	Timing pulse gate Rd pulse S latch	51.30.05
Read pulse 2 005-040	Rd pulse L latch Timing pulse gate	51.30.05

9. Set the strobe trigger:

SIGNAL	CONTROL	LOGIC
Trans scan	Rd tfr scan	51.40.40
Strobe latch	Trans scan Time 020-030	51.40.43
I-O to CPU trans	Rd trf scan	51.40.40
Buffer strobe	I-O to CPU trans Time 080-090 Strobe latch	51.40.43
E-ch strobe trigger	Buffer strobe E-ch select any buffer	15.62.03

10. Set the E1 register, set the E1 full latch, and reset the strobe trigger:

SIGNAL	CONTROL	LOGIC
Set E1 reg	E-ch strobe trigger E-ch input mode (not) E1 reg full	15.62.04
E-ch strobe trigger (off)	E-ch input mode Set E1 reg	15.62.03
E1 full	Set E1	15.41.10

11. Transfer E1 to E2:

E-ch move mode	1401 card print-in-process	15.62.02
E-ch reset	Card print reset	15.41.12
Copy E1 BCD to E2	(not) E1 reg word separator	15.62.06
E2 reg full latch (off)	E-ch reset	15.41.10
Set E2 reg	E1 reg full latch (on) E2 reg full latch (off)	15.62.04
E2 reg full latch (on)	Set E2 reg (not) E1 reg word separator	15.41.10
E1 reg full latch (off)	E-ch input mode Set E2 reg	15.41.10

12. Signal that an E cycle is required:

SIGNAL	CONTROL	LOGIC
E-cycle required	E-ch input mode E-ch-in-process E2 reg full	12.12.62
E-cycle ctrl	E-cycle required, 2nd clock pulse LG-Z, (not) F-cycle required	12.12.66
E-cycle	E-cycle ctrl, LG-B or S	12.12.66

13. Control the clock:

SIGNAL	CONTROL	LOGIC
Comp disable cycle	E-ch unovlp-in-process	12.12.60

This holds the clock off, until "E-cycle-required" comes on.

SIGNAL	CONTROL	LOGIC
Logic gate A (unclamped)	E-cycle required E-ch unovlp-in-process	11.10.10

14. Gate the E2 character to the A channel:

SIGNAL	CONTROL	LOGIC
Gate E2 data reg to A-ch	E-ch input mode E-cycle	15.38.03

15. Gate A channel to assembly:

SIGNAL	CONTROL	LOGIC
Use A-ch zones	In cy GM · WM ctrl A-ch valid or ast switch (off) Odd-parity cycle	15.49.03
Use A-ch nu	In cy GM · WM ctrl A-ch valid or ast switch (off) Odd-parity cycle	15.49.03
Use B-ch WM	Input cycle Move cycle	15.49.04

16. Store the assembly channel in the B address:

SIGNAL	CONTROL	LOGIC
B or E or F-cycle ctrl	E-cycle ctrl	12.12.02
B or E or F-cycle	B-cycle	12.12.07
RO BAR	E-cycle ctrl E-ch unovlp-in-process	14.71.31
Reset BAR	B or E or F-cycle ctrl LG-Early B	14.71.21
Set BAR	B or E or F-cycle ctrl, LG-B or C B or E or F-cycle, LG-D or E or F	14.71.11
Addr mod set to plus one	E-cycle ctrl	14.71.41

17. Energize the read clutch:

SIGNAL	CONTROL	LOGIC
1401 read latch	Rd trans req 1401 mode	52.10.08
Rd feed gate (latch)	1401 read latch, End of SS timing	52.10.08
Read feed (trigger)	Rd feed gate, Proc feed rd	52.10.09
Rd clutch	Read feed	52.10.09

18. End the external transfer, and develop sample pulses:

SIGNAL	CONTROL	LOGIC
E-ch ext end-of-transfer	Buffer-end-of-transfer E-ch select any buffer 1st clock pulse	13.42.11
E-ch last input cycle	E-ch ext end-of-transfer (not) E1 reg full (not) E2 reg full E-ch cycle, E-ch input mode Logic gate C or T	13.63.02
E-ch int end-of-transfer	E-ch last input cycle LG-F or W	13.63.01
E-ch int end-of-trf delayed	E-ch last input cycle Logic gate Z	13.63.01
E-ch status sample B	E-ch int end-of-trf delayed E-ch ext end-of-transfer Logic gate Z, (not) 2nd clock pulse	13.65.05
E-ch second sample B	E-ch status sample B (not) 2nd clock pulse	13.65.05
E-ch status sample B delayed	E-ch second sample B (not) 2nd clock pulse	13.65.05

19. Advance the scan ring to the end:

SIGNAL	CONTROL	LOGIC
1401 I-O ring advance	E-ch status sample B delayed	13.70.01
1401 I-O 4-pos (punch)	1401 read trigger 1401 I-O ring advance	13.70.02
(not) 1401 card or prtr mode	(not) read, (not) punch, (not) print	13.70.03
1401 I-O ring advance	(not) 1401 card or prtr mode	13,70.01
1401 I-O end pos	1401 punch trigger 1401 I-O ring advance	13.70.02
1401 card print-in-proc (off)	1401 I-O ring advance 1401 I-O end	13.70.02

20. Signal last execute cycle:

SIGNAL	CONTROL	LOGIC
Last execute cycle I-O	1401 I-O end (not) 1401 branch latch	13.65.07
Last execute cycle	Last execute cycle I-O	12.12.51

Punch a Card

Op-Code Function (4)

The 1401 punch-a-card op code (4) punches the information in the 1410 core storage locations 00101 to 00180 into a card. Word marks are not disturbed by the punch-out.

Operation (Figures 20, 21 and 23)

The 4 op code brings up 1401 card print ops to allow the 1401 I-O scan ring to advance. The scan-ring print trigger turns on, but since there is not an op register 2 bit, the ring advances to the read-trigger position.

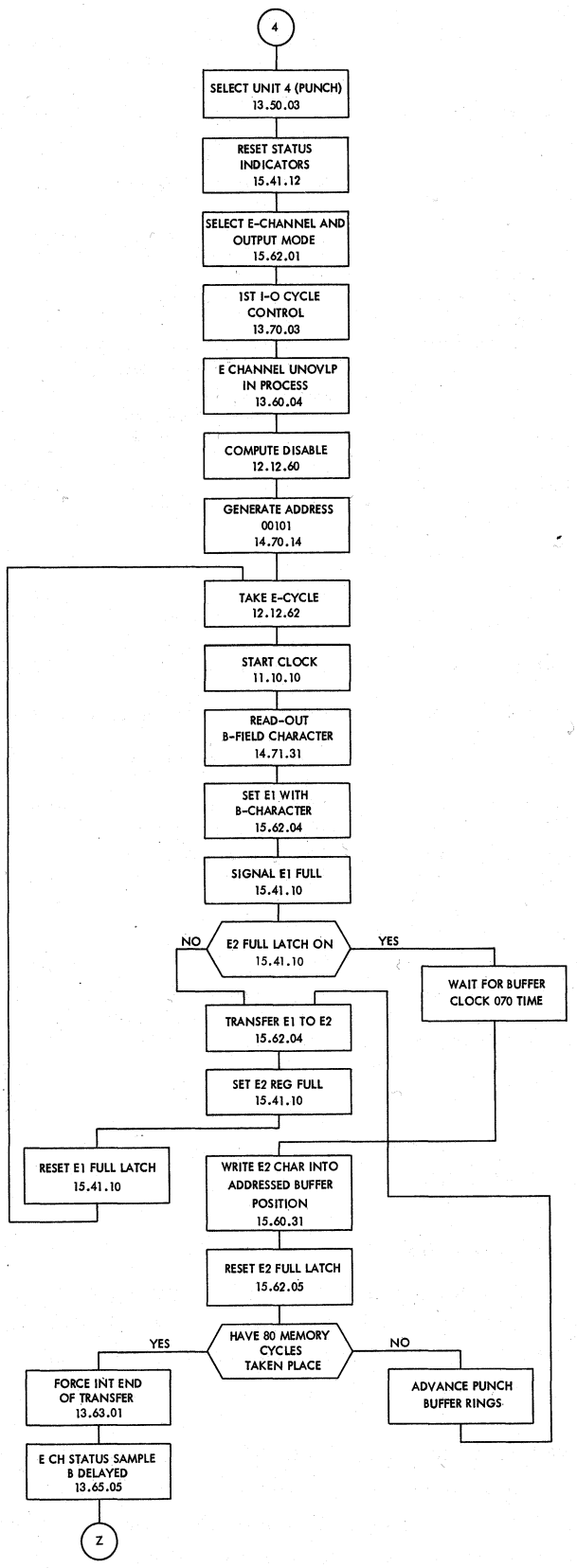


Figure 23. 1401 Mode Punch Op

There is no 1 bit in the op register, so that the ring advances to the punch-trigger position. The 4 bit in the op register combines with the punch trigger on output to signal a punch operation. A punch transfer scan is requested, and the transfer takes place over the E channel in the unoverlapped, move mode. Eighty characters transfer to punch storage. When the transfer is complete, the punch is set in motion to punch the record, and the CPU is released for further proccessing.

Circuits

1. Recognize the card-print operation:

SIGNAL	CONTROL	LOGIC
1401 card print opr	(not) Ctrl reg disable 1401 mode Op dcdr not A, not B, not 8	13.13.11

2. Scan for print or read or punch:

SIGNAL	CONTROL	LOGIC
1401 print trigger	1401 card print ops Last insn RO cycle Logic gate F	13.70.01
Not 1401 card or prtr mode	(not) 1401 punch (not) 1401 read (not) 1401 print	13.70.03
1401 I-O ring advance	(not) 1401 card or prtr mode	13.70.01

Because this is not a print operation (op register 2 bit is not on), the scan ring advances to the 1401 I-O 1 position (read trigger). Because this is also not a 1401 read operation (not op register 1 bit), the ring advances again to the 1401 I-O 4 position (punch trigger).

SIGNAL	CONTROL	LOGIC
1401 punch	1401 punch trigger Op reg 4-bit	13.70.03

The 1401 punch signal prevents the development of the "not 1401 card" or "prtr mode" signal. The 1401 I-O ring advance signal is not developed, so that the punch trigger remains on until the punch transfer is complete.

3. Signal card print-in-process:

SIGNAL	CONTROL	LOGIC
1401 set cd print-in-proc	1401 card or print op code Last insn RO cycle, LG-E	13.70.01
1401 card print-in-proc	1401 set cd print-in-proc	13.70.02

The 1401 card print-in-process trigger remains on until the punch buffer is filled.

4. Select E channel and unit 4:

SIGNAL	CONTROL	LOGIC
E-ch select unit 4	1401 punch	13.50.03
Unit 4 select to I-O	E-ch select unit 4	13.50.03
E-ch select any buffer	E-ch select unit 4	13.60.03
E-ch out mode (trigger)	1401 punch trigger	15.62.01
E-ch output op to buffer	E-ch out mode	15.62.01
E-ch output mode	E-ch out mode	15.62.01
Gate asm ch to E1 input	E-ch output mode (not) Control reg disable	15.62.07
E-ch move mode	1401 card print in process	15.62.02

5. Set up the E channel for processing:

SIGNAL	CONTROL	LOGIC
Card print reset (trigger)	1401 punch 2nd clock pulse clamped	13.70.03
1st I-O cycle control	Card print reset 2nd clock pulse clamped	13.70.03
E-ch unovlp-in-process	1401 card print-in-process 1st I-O cycle control	13.60.04
E-ch-in-process	E-ch unovlp-in-process	13.60.04

6. Generate address 00101:

SIGNAL	CONTROL	LOGIC
RO 00101 index addr	1st I-O cycle control 1401 punch trigger	14.70.14
RO-fixed addr	RO 00101 index addr	14.70.14
AR bus (00101)	RO fixed addr and RO 00101 index addr to	14.15.01 14.15.12

7. Request an E cycle:

SIGNAL	CONTROL	LOGIC
E-cycle required	E-ch output mode (not) E-ch int end-of-transfer E1 reg full latch (off) (not) E-cycle, any last gate	12.12.62
(Start logic clock)	E-cycle-required E-ch unovlp-in-process	11.10.10
E-cycle ctrl	E-cycle required 2nd clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12.12.66
E-cycle	E-cycle ctrl Logic gate B or S	12.12.66

8. Read out the B address to the B channel:

SIGNAL	CONTROL	LOGIC
B or E or F-cycle ctrl	E-cycle ctrl	12.12.02
B or E or F-cycle	E-cycle	12.12.07
RO BAR	E-cycle ctrl E-ch unovlp-in-process	14.71.31
Reset BAR	B or E or F-cycle ctrl LG-early B	14.71.21
Set BAR	B or E or F-cycle ctrl Logic gate B or C	14.71.11
Addr mod set to plus one	E-cycle ctrl	14.71.41

9. Gate the B-channel character to the E1 input:

SIGNAL	CONTROL	LOGIC
Output cycle	E-ch output mode, E-cycle	13.60.06
Output field cycle	E-cycle (not) E-ch select unit 2	15.49.04
Use B-ch nu	Output cycle Output field cycle	15.49.05
E-ch select odd parity unit	E-ch select unit 4	13.60.03
Odd parity cycle	E-cycle E-ch select odd parity unit	13.60.02
Use B-ch zones	Output cycle Output field cycle Odd parity cycle	15.49.05
Gate asm ch to E1 input	E-ch output mode (not) Control reg disable	15.62.07

10. Set E1 data register:

SIGNAL	CONTROL	LOGIC
Set E1 reg	1401 card print-in-proc (not) E-ch 2nd addr trf	15.62.04
E1 reg full	Set E1 reg	15.41.10

11. Transfer E1 to E2:

SIGNAL	CONTROL	LOGIC
E-ch move mode	1401 card print-in-process	15.62.02
E-ch reset	Card print reset	15.41.12
E2 reg full latch (off)	E-ch reset	15.41.10
Set E2 reg	E1 reg full latch (on) E2 reg full latch (off)	15.62.04
E2 reg full latch (on)	E-ch output mode Set E2 reg	15.41.10
E1 reg full latch (off)	(not) Set E1 reg	15.41.10

When the E1 register full latch is turned off, "E-cycle required" is brought up to cause another character to read out and be gated to the E1 register.

12. Request a punch-transfer scan:

SIGNAL	CONTROL	LOGIC
Ready to buffer	E-ch-in-process E-ch select any buffer E2 reg full	13.70.04
Ready	Ready to buffer	51.40.04
AC set pch trans req trig	Pch ready Ready (not) Pch busy	52.10.14
Pch trans req	Select unit 4 AC set pch trans req trig	52.10.15
Pch priority gate	Pch trans req	52.10.15
Pch pr req (trigger)	Pch priority gate Time 080-090	52.10.01
Gate on pch scan TR	Pch pr req Same scan (not) Inh rd priority req	52.10.01
Pch scan	Gate on pch scan TR Time 030-040	52.10.02
Pch trans scan	Pch ready Pch trans req Pch scan	52.10.06
Trans scan	Pch trans scan	51.40.40

13. Write E2 character into the addressed buffer position:

SIGNAL	CONTROL	LOGIC
CPU to I-O Sync bit	E2 reg bit	15.60.31-34
CPU to I-O bit	CPU to I-O sync bit	51.40.10
Inh gate bit	CPU to I-O bit	51.16.03
Write pulse	070-100	51.30.05

14. Set the strobe trigger:

SIGNAL	CONTROL	LOGIC
Trans scan	Pch trans scan	51.40.40
Strobe latch	Trans scan Time 020-030	51.40.43
CPU to I-O trans	Pch trans scan	51.40.40
Buffer strobe	Strobe latch CPU to I-O trans Time 100-000	51.40.43
Reset E2 full latch	E2 reg full E-ch strobe trigger E-ch output mode	15.62.05
E2 reg full latch (off)	Reset E2 full latch	15.41.10
E-ch strobe trigger (off)	Reset E2 full latch E-ch output mode	15.62.03

15. After 80 memory cycles, stop scanning the buffer and develop the external end-of-transfer:

SIGNAL	CONTROL	LOGIC
End of scan	Tens ring AC set Tens ring 7	51.32.01
End-of-transfer, 1-latch (on)	Trans scan	51.40.12
End-of-transfer, 2-latch (on)	End-of-transfer, 1-latch (on) (not) Trans scan Time 090-100	51.40.12
End-of-trans	End-of-transfer, 2-latch (on)	51.40.12
Buffer end-of-trans	End-of-trans	51.40.12
E-ch ext end-of-transfer	Buffer end of trans E-ch select any buffer 1st clock pulse	13.42.11
E-ch int end-of-transfer	E-ch ext end-of-transfer E-ch output mode E1 reg full E2 reg full 2nd clock pulse	13.63.01
E-ch int end-of-trf delayed	E-ch int end-of-transfer Logic gate Z	13.65.05
E-ch status sample B	E-ch int end-of-trf delayed E-ch ext end-of-transfer (not) E-ch status sample Logic gate Z, (not) 2nd clock pulse	13.65.05
E-ch second sample B	E-ch status sample B (not) 2nd clock pulse	13.65.05
E-ch status sample B delay	E-ch 2nd sample B (not) 2nd clock pulse	13.65.05

16. Energize the punch clutch:

SIGNAL	CONTROL	LOGIC
Correct trans to buffer	E-ch status sample B-delay 1401 mode E-ch select any buffer	13.70.04
Go	Correct trans to buffer	51.40.04
Punch feed gate	Go Select unit 4	52.10.14
Punch feed	Proc pch Pch feed gate	52.10.15

17. Advance scan ring to the end position:

SIGNAL	CONTROL	LOGIC
1401 I-O ring advance	E-ch status sample B-delayed	13.70.01
1401 I-O end pos	1401 punch trigger 1401 I-O ring advance	13.70.02
1401 card print-in-proc (off)	1401 I-O ring advance 1401 I-O end	13.70.02

18. Turn off the unoverlap-in-process latch and bring up the last execute cycle:

SIGNAL	CONTROL	LOGIC
(not) E-ch unovlp-in-process	E-ch status sample B-delay	13.60.04
Last execute cycle I-O	1401 I-O end (not) 1401 branch latch	13.65.07
Last execute cycle	Last execute cycle I-O	12.12.51

Print a Line

Op-Code Function (2)

The print op code causes the 132 (or 100) characters in CPU core storage to transfer to print storage. After a correct transfer, the printer is signalled to begin printing the information just transferred.

Operation (Figures 20, 21 and 24)

No check is made in the 1401 mode for an I-O interlock condition. The status indicators are reset, and have no effect on the operation. The 1401 print op code is recognized in the same manner as described for read and punch. The E channel and unit 2 are selected. The transfer of data takes place in the unoverlapped mode. The address 00201 is developed so that core storage locations 00201 through 00332 (or 00300) can read out in sequence to print storage. E cycles are taken as required to keep the E1 and E2 registers full. The BAR is modified by plus one on each E cycle. Each time E2 is filled, the character is stored in the next sequential print storage position. When the print storage is filled, an E-channel "external-end-of-transfer" signal develops. This results in a "go" signal which starts the printing operation. It also results in advancing the scan ring to read position to test for an op-register 1 condition.

Circuits

1. Recognize a card-print operation:

SIGNAL	CONTROL	LOGIC
1401 card print ops	(not) Ctrl reg disable 1401 mode Op dcdr not A, not B, not 8	13.13.11

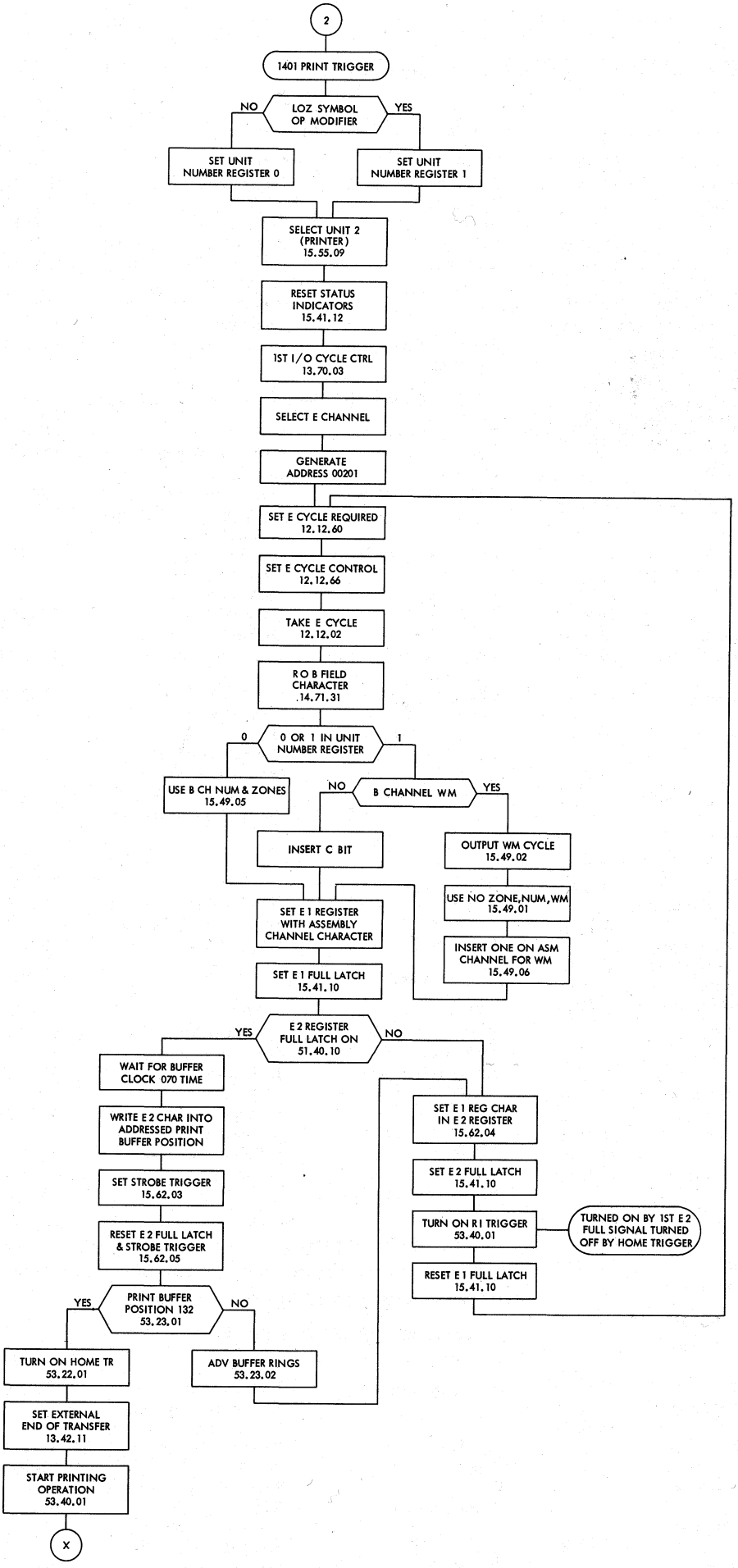


Figure 24. 1401 Mode Print Op

2. Scan for print or read or punch:

SIGNAL	CONTROL	LOGIC
1401 print trigger	1401 card print ops Last insn RO cycle Logic gate F	13.70.01
1401 print	1401 print trigger Op reg 2-bit	13.70.03

The 1401 print signal blocks the drive signal for the scan ring. Thus, the 1401 print trigger remains on until print storage is filled from core storage.

3. Signal card print-in-process:

SIGNAL	CONTROL	LOGIC
1401 set cd print-in-proc	1401 card or print op code Last insn RO cycle Logic gate E	13.70.01
1401 card print-in-proc	1401 set cd print-in-proc	13.70.02

4. Select E channel and unit 2:

SIGNAL	CONTROL	LOGIC
E-ch select unit 2	1401 print	13.50.03
Unit 2, select to I-O	E-ch select unit 2	13.50.03
E-ch select any buffer	E-ch select unit 2	13.60.03
E-ch out mode (trigger)	1401 print trigger	15.62.01
E-ch output op to buffer	E-ch out mode	15.62.01
E-ch output mode	E-ch out mode (not) Control reg disable	15.62.01

5. Set up the E channel for processing:

SIGNAL	CONTROL	LOGIC
Card print reset	1401 print 2nd clock pulse clamped	13.70.03
1st I-O cycle control	Card print reset 2nd clock pulse clamped	13.70.03
E-ch unovlp-in-process	1401 card print-in-process 1st I-O cycle control	13.60.04
E-ch in-process	E-ch unovlp-in-process	13.60.04

6. Generate address 00201:

SIGNAL	CONTROL	LOGIC
RO 00201 index addr	1st I-O cycle control 1401 print trigger	14.70.14
RO fixed addr	RO 0020 index addr	14.70.14
AR bus (00201)	RO fixed addr and RO 00201 index addr to	14.15.01 14.15.12

7. Request an E cycle to fill E1:

SIGNAL	CONTROL	LOGIC
E-cycle required	E-ch output mode (not) E-ch int end-of-transfer E1 reg full latch (off) (not) E-cycle any last gate	12.12.62
(start logic clock)	E-cycle required E-ch unovlp-in-process	11.10.10
E-cycle ctrl	E-cycle required 2nd-clock pulse Logic gate Z (not) E-ch int end-of-transfer (not) F-cycle required	12.12.66
E-cycle	E-cycle ctrl Logic gate B or S	12.12.66

8. Read out the B address to the B channel:

SIGNAL	CONTROL	LOGIC
B or E or F-cycle ctrl	E-cycle ctrl	12.12.02
B or E or F-cycle	B-cycle	12.12.07
RO BAR	E-cycle ctrl E-ch unovlp-in-process	14.71.31
Reset BAR	B or E or F-cycle ctrl LG early B	14.71.21
Set BAR	B or E or F-cycle ctrl Logic gate B or C	14.71.11
Addr mod set to plus-one	E-cycle ctrl	14.71.41

9. Gate B-channel character to E1 input:

SIGNAL	CONTROL	LOGIC
Output cycle	E-ch output mode E-cycle	13.60.06
E-ch unit number 0	1401 print trigger (not) Loz symbol op modifier	15.55.04
Output field cycle	E-cycle E-ch select unit 2 E-ch number 0	15.49.04
Use B-ch nu	Output cycle Output field cycle	15.49.05
E-ch select odd parity unit	E-ch select unit 2	13.60.03
Odd parity cycle	E-cycle E-ch select odd parity unit	13.60.02
Use B-ch zones	Output-cycle Output field cycle Odd parity cycle	15.49.05
Gate asm ch to E1 input	E-ch output mode (not) Control reg disable	15.62.07

10. Set the E1 data register:

SIGNAL	CONTROL	LOGIC
Set E1 reg	1401 card print-in-proc (not) E-ch 2nd-addr-trf	15.62.04
E1 reg full	Set E1 reg	15.41.10

11. Transfer E1 to E2:

SIGNAL	CONTROL	LOGIC
E-ch move mode	1401 card print-in-process	15.62.02
E-ch reset	Card print reset	15.41.12
E2 ref full latch (off)	E-ch reset	15.41.10
Set E2 reg	E1 reg-full latch (on) E2 reg-full latch (off)	15.62.04
E2 reg full latch (on)	E-ch output mode Set E2 reg	15.41.10
E1 reg full latch (off)	(not) Set E1 reg	15.41.10

When the E1-register full latch is turned off, "E-cycle required" is brought up to cause another character to be read out and gated to the E1 register.

12. Request a print transfer scan:

SIGNAL	CONTROL	LOGIC
Ready to buffer	E-ch in-process E-ch select any buffer E2 reg full	13.70.04
Ready	Ready to buffer	51.40.04
Read in	Ready Ready for CPU transfer	53.40.01
Print transfer	Read in (not) CE mode (not) C-bit insert	53.11.05
Trans scan	Print transfer	51.40.40

13. Write the E2 character into the addressed print-buffer position:

SIGNAL	CONTROL	LOGIC
CPU to I-O sync bit	E2 reg bit	15.60.31-34
CPU to I-O bit	CPU to I-O sync bit	51.40.10
CPU transfer	Print trans	53.11.05
Gated CPU input bit	CPU transfer CPU to I-O bit	53.11.01
Print data input bit	Gated CPU input bit	53.11.03
Inhibit gate bit	Print data input bit	53.10.05
Inhibit pulse 1, 2	Inhibit pulse Read in or scan (not) Delay	53.45.01
bit inh dr	Inhibit gate bit Inhibit pulse 1, 2	53.10.06
Write pulse	Write pulse from I-O Read in or scan (not) Delay	53.45.01

14. Send the strobe pulse to the CPU to reset the E2 full latch:

SIGNAL	CONTROL	LOGIC
Trans scan	Print trans	51.40.40
Strobe latch	Trans scan Time 020-030	51.40.43
CPU to I-O trans	Print trans	51.40.40
Buffer strobe	Strobe latch CPU to I-O trans Time 100-000	51.40.43
Reset E2 full latch	E2 reg full E-ch strobe trigger E-ch output mode	15.62.05
E2 reg full latch (off)	Reset E2 full latch	15.41.10
E-ch strobe trigger (off)	Reset E2 full latch E-ch output mode	15.62.03

15. After 132 memory cycles, turn on the home trigger:

SIGNAL	CONTROL	LOGIC
Turn off rings latch (on)	Read in	53.23.01
Gate on home trigger	Turn off rings latch (on) Threes-ring 3 Fives-ring 5 Tens-ring 3	53.23.01
Threes-ring advance	Read in Ring advance (time 010-020)	53.23.02
Read in (off)	Gate on home trigger Ring advance (time 010-020)	53.40.01
Home trigger	Threes-ring advance Gate on home trigger	53.22.01

16. Signal the external end-of-transfer, and develop the E channel status sample B pulse:

SIGNAL	CONTROL	LOGIC
End-of-transfer 1 latch	Trans scan	51.40.12
(not) print trans	Read in (off)	53.11.05
(not) trans scan	(not) Print trans	51.40.40
End-of-transfer 2 latch	End-of-transfer 1 (not) Trans scan Time 090-100	51.40.12
Buffer end-of-transfer	End-of-transfer 2	51.40.12
E-ch ext end-of-transfer	Buffer end-of-transfer E-ch select any buffer 1st clock pulse	13.42.11
E-ch int end-of-transfer	E-ch ext end-of-transfer E-ch output mode E1 reg full E2 reg full	13.63.01
E-ch int end-of-trf delayed	E-ch int end-of-transfer Logic gate Z	13.65.05
E-ch status sample B	E-ch int end-of-trf delayed (not) 2nd clock pulse E-ch ext end-of-transfer Logic gate Z	13.65.05
E-ch second sample B	E-ch status sample B	13.65.05
E-ch status sample B-delay	(not) 2nd clock pulse E-ch second sample B (not) 2nd clock pulse	13.65.05

17. Start the printing operation:

SIGNAL	CONTROL	LOGIC
Correct transfer to buffer	E-ch status sample B-delay 1401 mode E-ch select any buffer	13.70.04
Go	Correct transfer to buffer	51.40.04
Scan call	Go Ready for print or forms Select unit 2 (not) Scan call	53.40.01

18. Set last execute cycle: The E-channel status-sample B-delayed signal advances the scan ring through the read and punch positions to turn on 1401 I-O end.

SIGNAL	CONTROL	LOGIC
E-ch unovlp-in-process (off)	E-ch status sample B-delay	13.60.04
Last execute cycle I-O	1401 I-O end (not) 1401 branch latch	13.65.07
Last execute cycle	Last execute cycle I-O	12.12.51

Write Word Marks

Op-Code Function (2⌑)

The word marks in the print area of core storage print as 1's in the corresponding print positions. The lozenge symbol op modifier translates the word marks to 1's and suppresses the printing of all zone, WM or numeric characters (except 1's for word marks).

Operation (Figures 20, 21 and 24)

This op code functions in the same manner as the write-a-line op code. B-channel characters are gated through assembly and are analyzed for WM's. A 1 bit is inserted on the assembly channel each time a WM is sensed. All characters without word marks are converted to valid blanks (C bit only).

Circuits

1. Develop an output WM cycle:

SIGNAL	CONTROL	LOGIC
Loz symbol op modifier	Op mod reg C-bit 4, not 2, not 1, op mod B A 8 op mod	13.21.21
E-ch unit number 1	Loz symbol op mod 1401 print trigger	15.55.04
Output WM cycle	E-ch select unit 2 E-cycle E-ch unit number 1	15.49.02

2. Insert one bit for each character containing a word mark:

SIGNAL	CONTROL	LOGIC
Assembly ch nu one insert	Output WM cycle B-ch WM bit Output cycle	15.49.06

3. Insert a C bit for each character containing no word mark:

SIGNAL	CONTROL	LOGIC
Use no numerics	Output WM cycle B-ch not WM bit Output cycle	15.49.01
Use no zones	Output WM cycle Output cycle	15.49.01
Use no WM	Output cycle Move cycle	15.49.01

"Use no numerics" and "use no zones" combine with "B-ch not WM bit" to produce a C bit.

Combination Card-Print Instructions

Op-Code Function

Two or three I-O op code functions can be combined into one instruction. In any case, the printer takes priority over the reader, and the reader over the punch. The combination op codes are as follows:

COMBINATION OP CODES	OP CODE	OP REG BITS
Write and read	3	12
Read and punch	5	14
Write and punch	6	24
Write, read, and punch	7	124

Operation (Figures 20, 21, 22, 23 and 24)

In each of the cases, the transfer between CPU core storage and the I-O storage unit takes place in the priority order print, read, punch. As soon as one transfer is complete, the corresponding unit is set in motion, and a test is made of the op register bits to determine whether the next operation in priority order is to be performed. When a transfer-to-print-storage is complete, a test is made of the op register 1-bit line (read). If it is up, a transfer of read storage to CPU core storage is made. If the op register 1 line is down, an immediate test is made of the op register 4 line to determine whether a punch operation should take place. If no op register 4 bit exists, an I-O end signal results. This in turn causes a last execute cycle. The testing of the op register bit output is made by the 4-position, 1401 I-O scan ring. This scans each op register bit in turn to determine which operations should take place. The positions of the ring test the op register outputs as follows:

POSITION	NAME OF TRIGGER	OP REG BIT TESTED	LOGIC
1	Print trigger	2	13.70.01
2	Read trigger	1	13.70.01
3	Punch trigger	4	13.70.02
4	1401 I-O end	–	13.70.02

The ring advances by a clock pulse if the op register bit line tested is not up. "E-ch status sample B delayed" advances the ring at the end of the data transfer if the op register bit line tested is on.

Select Stacker

Op-Code Function

This instruction selects the card that was just read or punched into the stacker pocket specified by the op modifier.

d-CHARACTER	FEED	STACKER
1	Read	1
2	Read	8/2
4	Punch	4
8	Punch	8/2

Operation (Figure 25)

The select stacker op code switches the op-modifier character to the A-data register during the I cycle in which it appears on the B channel. The A-data register contents then transfer via E1 and E2 to the CPU to I-O bit lines. The forms stacker "go" signal develops and is switched with stacker select to develop a signal that samples the CPU-to-I-O bit lines (op modifier contents) to turn on either the read-stacker-select or the punch-stacker-select latches. Actual selection of the card in the punch feed does not occur until the punch busy goes off after the previous punching operation.

Circuits

Op-code grouping lines are:

1401 no exe cy branch ops
2-char only op codes
Op mod to A-ch on B-cycle ops
Regen mem on B-cycle ops
No branch

1. Develop the stacker select signals:

SIGNAL	CONTROL	LOGIC
1401 stacker select op code	Op reg 1401 C-bit Op dcdr B, not A, not 8 Op dcdr 2, not 4, not 1	13.13.11
Stacker sel op code	1401 stacker sel op code	13.13.08
Stacker select to buffer	Stacker sel op code	13.70.04

2. Reset the status indicators:

SIGNAL	CONTROL	LOGIC
For K E-ch reset	2-char-only op codes Logic gate C I-ring 1	15.41.12
E-ch reset This causes the reset of:	For K E-ch reset	15.41.12
	E-ch data check	12.62.04
	E-ch not ready	12.62.01
	E-ch condition	12.62.04
	E-ch busy	12.62.02
	E-ch no transfer latch	13.72.04
	E-ch wrong-length record	13.63.03
	E-ch correct-length record	13.63.03

3. Set the op modifier character in the A-data register:

SIGNAL	CONTROL	LOGIC
Sw B-ch to A-reg	I-cycle Logic gate D 2nd clock pulse	15.38.01

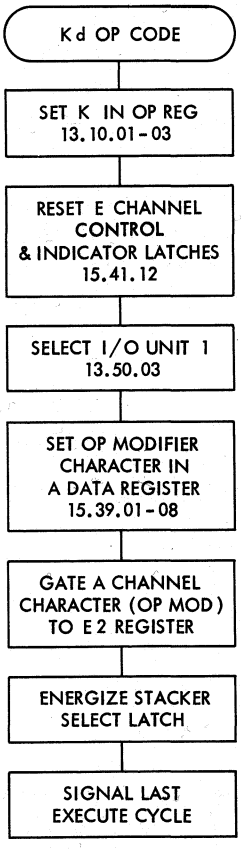


Figure 25. 1401 Mode Stacker Select Op

4. Gate the A-channel character (op modifier) to the E2 register:

SIGNAL	CONTROL	LOGIC
A-ch bit	Sw B-ch to A-reg	15.39.01-08
Use A-ch nu	I-cycle	15.49.03
Use A-ch zone	I-cycle	15.49.03
Asm ch bit	Use A-ch nu	15.40.01-10
Gate asm ch to E1	I-cycle and not CR disable	15.62.07
Set E1 reg	2-char only op code A-ch not WM bit I-cycle	15.62.04
Copy E1 BCD to E2	(not) E1 reg word separator	15.62.06
Set E2 reg	E1 reg full (not) E2 reg full	15.62.04
E-ch—bit	Set E2 reg	15.60.31-34

5. Select the stacker magnet according to the E-channel (op modifier) character:

SIGNAL	CONTROL	LOGIC
Forms stacker go	2-char only op codes Last insn RO cycle (not) E-ch not ready (not) E-ch busy (not) E-ch no transfer latch	13.70.04
Stacker select Rd stk sel and Pch stk sel	Stack select to buffer 1401 mode Stacker select Forms stacker go	51.40.01 51.40.50
Rd stk one (or two) latch	Rd stk sel CPU to I-O bit 1 (or 2)	52.13.02
Stack four (or eight)	Pch stk sel CPU to I-O bit 4 (or 8)	52.13.01

The AC set for the stack 4 or 8 read unit occurs when the select unit 4 line is up and punch busy goes off. This assures that the card is punched and ready for selection into the stacker pocket when the select magnet is energized.

6. Signal the last execute cycle:

SIGNAL	CONTROL	LOGIC
Last execute cycle I-O	2-char only op codes 1401 mode (not) 1401 branch latch Last insn RO cycle E-ch any status (on)	13.65.07

Carriage Control

Op-Code and Op-Mod Functions

The F-op code sends the op-modifier character to the carriage circuitry to control forms skipping and spacing. The op-modifier character causes the tape-controlled carriage to take a single, double, or triple space, or to skip to the next hole in the designated tape channel.

Operation (Figure 26)

The forms control operation transfers the op-modifier character to the A-data register during the I cycle in which it appears on the B channel. The A-data register contents then switch via E1 and E2 to the CPU I-O bit lines. The forms-stacker "go" signal then develops, and is switched with forms control to develop a CCC register gate signal that samples the CPU to I-O bit lines to set the CCC register.

Circuits

Op-code grouping lines are:

2-char only op codes
1401 no exe cy branch ops
Op-mod to A-ch on B-cy ops
Regen mem on B-cy op codes

1. Set up controls for forms op:

SIGNAL	CONTROL	LOGIC
1401 forms ctrl	Op dcdr B, A, not 8	13.13.11
Op code	Op dcdr 42 not 1 Op reg 1401 C-bit	
Forms ctrl op code	1401 forms ctrl op code	13.13.08
Forms ctrl to buffer	Forms ctrl op code	13.70.04
Forms control	Forms ctrl to buffer	53.50.04

2. Reset the status indicators:

SIGNAL	CONTROL	LOGIC
For K E-ch reset	2-char only op codes Logic gate C I-ring 1	15.41.12
E-ch reset This causes the reset of:	For K E-ch reset	15.41.12
	C-ch data check	12.62.04
	E-ch not ready	12.62.01
	C-ch condition	12.62.04
	E-ch busy	12.62.02
	E-ch no transfer latch	13.72.04
	C-ch wrong-length record	13.63.03
	E-ch correct-length record	13.63.03

3. Set the op modifier character in the A-data register:

SIGNAL	CONTROL	LOGIC
Sw B-ch to A-reg	I-cycle Logic gate D 2nd clock pulse	15.38.01

4. Gate the A-channel character (op modifier) to E2 register:

SIGNAL	CONTROL	LOGIC
A-ch bit	Sw B-ch to A-reg	15.39.01-08
Use A-ch nu	I-cycle	15.49.03
Use A-ch zone	I-cycle (not) 1401 and I-ring 5 or 10 time	15.49.03
Asm ch bit	Use A-ch nu Use A-ch zones	15.50.01-10
Gate asm ch to E1	I-cycle and not CR disable	15.62.07
Set E1 reg	2-char only op code A-ch not WM bit I-cycle	15.62.04
Copy E1 BCD to E2	(not) E1 reg word separator	15.62.06
Set E2 reg	E1 reg full (not) E2 reg full	15.62.04
E-ch bit	Set E2 reg	15.60.31-34

5. Gate the E2 register to the carriage-control character register:

SIGNAL	CONTROL	LOGIC
Forms stacker go	2-char only op codes Last insn RO cycle (not) E-ch not ready (not) E-ch busy (not) E-ch no transfer latch	13.70.04
Forms and stacker go	Forms stacker go	51.40.50
CCC reg gate	Ready for print or forms Forms control (not Forms busy status	53.50.03
CPU to I-O —bit	Ed reg bit	15.60.31-34
CCC reg—bit	CCC reg gate CPU to I bit	53.50.01-02

6. Reset the CCC register: The CCC register resets in the usual manner, when the ss unit (53.55.03) times out at the end of carriage movement. The reset of the CCC register prevents repetition of the same operation.

The CCC register also resets at the beginning of each forms-control operation. The reset latch (53.50.03) controls the timing of this reset to occur between the time the forms-control line comes on and the time the forms and stacker-go line comes on. This reset allows the programmer to change the CCC register contents when a space-after-print or a skip-after-print has previously been set up. This change in the CCC register would occur while printing is taking place.

7. Develop magnet impulses for spacing and skipping: This occurs in the same manner as described in *IBM Customer Engineering Instruction-Reference, 1414 Input-Output Synchronizer, Models 3, 4, 5, 6, and 8* (Form 223-2590).

8. Signal the last execute cycle:

SIGNAL	CONTROL	LOGIC
Last execute cycle I-O	2-char only op codes 1401 mode (not) 1401 branch latch Last insn RO cycle E-ch any status (on)	13.65.07

Card-Print-Branch Instructions

Op-Code Function

Any of the following 1401 input-output op codes cause a branch to the I address (read into the 1410 AAR) after the input or output data transfers.

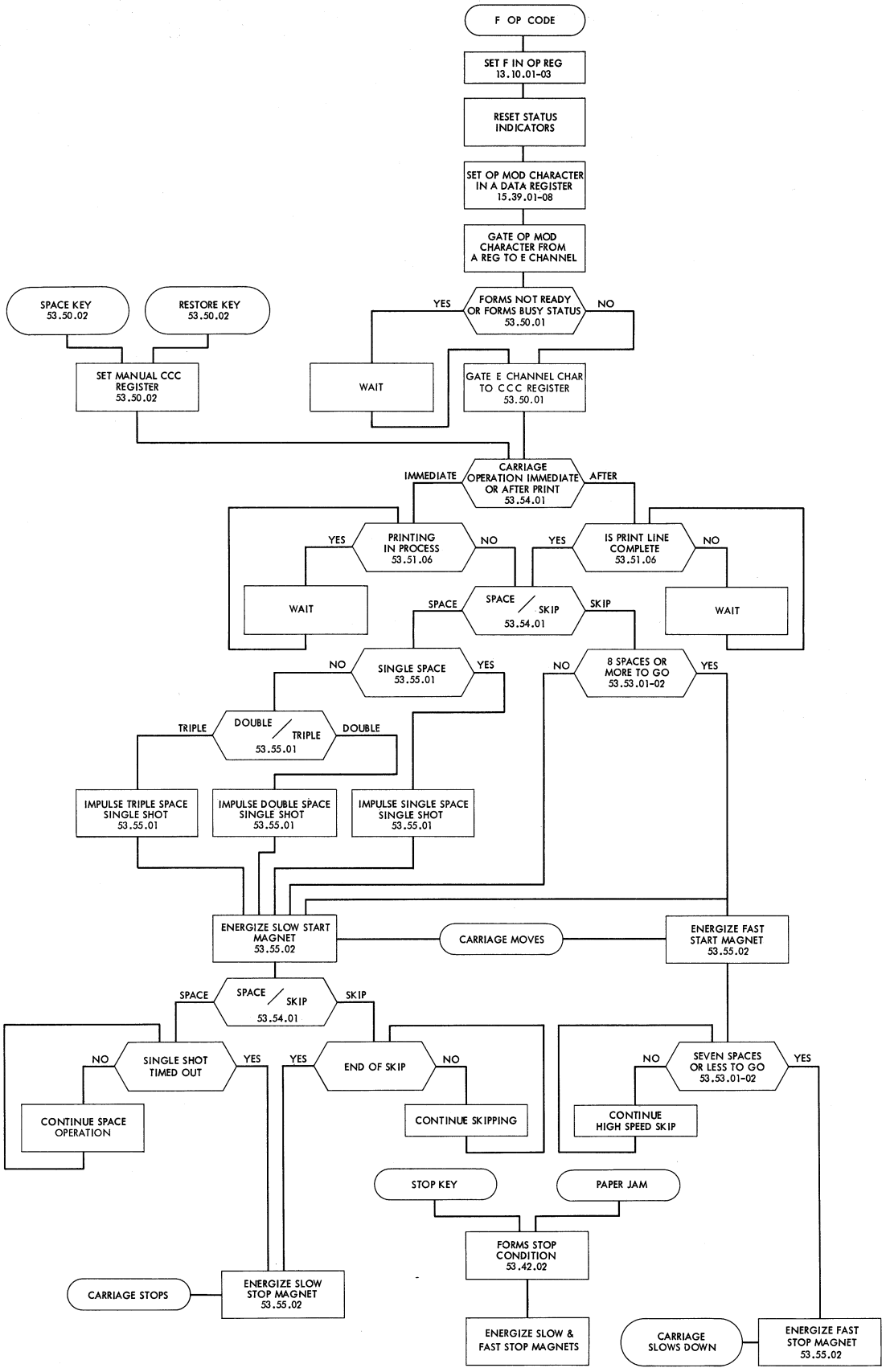


Figure 26. 1401 Mode Carriage Control Op

1401 I-O OP CODES	OP	I-ADDR	OP MOD
Read and branch	1	xxx	
Punch and branch	4	xxx	
Write and branch	2	xxx	
Write WM and branch	2	xxx	Loz
Read, punch, and branch	5	xxx	
Write, punch, and branch	6	xxx	
Write, read, punch, and branch	7	xxx	

Operation

Whenever the I ring runs through 8 time, as it does with all of the codes in the preceding paragraph, the 1401 branch latch turns on. The operation then continues in the normal manner. The 1401 branch latch prevents the usual last-execute cycle, when 1401 I-O end comes on. Instead, the 1401 branch latch switches with 1401 I-O end to cause a B cycle. During this B cycle the IAR (contains the address of the next instruction in sequence) reads out, is modified by zero, and returns to the BAR. The programmer can use the next following instruction to store the BAR, if it is needed later in the program. A last-execute cycle now occurs, and the next instruction is taken from the address in the AAR.

Circuits

1. Recognize the input-output branch op:

SIGNAL	CONTROL	LOGIC
1401 I-ring-8 branch ops	1401 card or prints ops	13.13.11
1401 branch latch	1401-I-ring-8 branch ops I-ring-8 Last logic gate	12.60.19

2. Prevent normal last-execute cycle I-O. The 1401 branch latch (13. 65. 07) prevents the usual last-execute cycle I-O, when 1401 I-O end comes on.

3. Transfer IAR to BAR after I-O transfers are complete:

SIGNAL	CONTROL	LOGIC
1401 I-O set branch cnds	1401 I-O end 1401 branch latch	13.70.02
B-cycle ctrl This causes a B-cycle in which "set BAR" takes place.	1401 I-O set branch cnds	12.12.21
No scan ctrl This causes "modify by zero."	1401 I-O set branch cnds	12.30.03
RO IAR The IAR reads out to be modified by zero, and then returns to the BAR.	No scan ctrl B-cycle ctrl LG spec A	14.71.34

4. Signal last execute cycle:

SIGNAL	CONTROL	LOGIC
Last-execute cycle br cnds	No scan B-cycle	12.60.08
Last-execute cycle	Last-execute cycle br cnds	12.12.51

5. Read out AAR on first I cycle:

SIGNAL	CONTROL	LOGIC
RO AAR	1401 branch latch 1401 mode I-cycle ctrl I-ring ctrl	14.71.30

Stacker Select or Control Carriage and Branch

The stacker select and branch or control carriage and branch instructions operate in the same manner as control carriage or select stacker except that the next instruction is taken from the specified I address. The instruction format is either K (xxx) d, or F (xxx) d.

Operation

If the I ring goes to I9 (as it does when a modifier is included), the 1401 branch latch sets. At the same time, a 1401 take-I-to-B-cycle is forced. This causes an immediate B cycle with no scan, during which the last instruction address that was used transfers from IAR to BAR for possible future use by the programmer. A last-execute cycle is forced during the B cycle. During the resulting I cycle, the 1401 branch latch causes the AAR contents to be placed in STAR. Both stack select and carriage control operations occur as described in previous sections.

Circuits

Refer to previous sections for the basic circuits for either op code. The following additional circuits are developed to cause the branch to the I address (read into the 1410 AAR).

1. Recognize the stacker, select or control carriage and branch operation:

SIGNAL	CONTROL	LOGIC
1401 I-ring-9 branch ops	1401 stacker select op code or 1401 forms ctrl	13.13.11
1401 branch latch	1401 I-ring-9 branch ops I-ring-9 time Last logic gate	12.60.19

2. Transfer IAR to BAR:

SIGNAL	CONTROL	LOGIC
1401 take I to B-cycle	1401 no exe cy branch ops 1401 I-ring-9 branch ops I-ring-9 Last insn RO cycle	12.60.19
Set B-cycle ctrl br ops	1401 take I to B-cycle	12.60.04
Set B-cycle ctrl	Set B-cycle ctrl br ops	12.12.44
B-cycle ctrl latch	Set B-cycle ctrl Last logic gate	12.12.21
Set no scan ctrl	1401 take I to B-cycle	12.30.05
RO IAR	No scan ctrl B-cycle ctrl Logic gate special A	14.71.34

The IAR reads out to be modified by zero and then returns to the BAR during the B cycle so that it can later be stored by the programmer, if required.

3. Signal last-execute cycle:

SIGNAL	CONTROL	LOGIC
Last-execute cycle br ends	No scan B-cycle	12.60.08
Last-execute cycle	Last-execute cycle br ends	12.12.51

4. Read out AAR on first I cycle:

SIGNAL	CONTROL	LOGIC
RO AAR	1401 branch latch 1401 mode I-cycle ctrl I-ring ctrl	14.71.30

Start Read Feed and Start Punch Feed

Op-Code Function (8, 9)

Both of these op-codes are used on the 1401 to avoid interlock time. This provides more program time. Because the 1410 last-execute cycle occurs as soon as the I-O storage unit fills, no interlock time is involved. Thus, neither of these codes has any function on the 1410, and they are handled the same as "no-op."

Operation

Logic Systems 13.13.11 indicates the manner that codes 8 and 9 develop to cause a "no-op" signal. See the section "No Operation (N)" for a description of the no-op function.

I-O Error Conditions

I-O error checking in 1401 mode requires significant changes in some related circuits. The E-channel resets shown in Figure 28 are developed only in 1401 mode.

Two additional inputs are active in the reset CPU circuits developed on 13.70.04. See Figure 29. Some normal 1410 inputs to these circuits are blocked in 1401 mode.

Status latches are not set for testing by branch instructions but E-channel data check and E-channel condition latches are set and used in the following manner. See Figures 27 and 30. A read data check is used for an example. If a data check is detected and the 1401 I-O check stop switch is on, master error is developed. This holds the clock in 1 CP. E-channel reset is not developed so that the data check and read indicators remain on. If the 1401 I-O check stop switch is off and a data check is detected, the data check latch remains on only from E-channel status B until status sample B delay.

1401 Read Error (1401 I-O Check Stop Switch On)

SIGNAL	CONTROL	LOGIC
E-ch check bus	Buffer error E-ch sel any buffer	12.62.03
1401 read error latch	E-ch check bus 1401 read trigger E-ch status sample B	13.65.01
Master error sample	1401 card print sample B	18.14.08
E-ch data check latch	E-ch check bus Status sample B	12.62.04
1401 card print error	E-ch check latch 1401 card print in process 1401 I-O check stop switch on E-ch second sample B	13.70.03
Master error	1401 card print error Master error sample 1 CP	18.14.08

The error card will have stacked when the system stops. If the check stop switch is off, the following occurs. Assume a B (I) ? instruction.

SIGNAL	CONTROL	LOGIC
1401 branch condition	1401 read error ? op mod	12.60.18
Reset reader error latch	Condition test branch op code No scan ? op mod LLG	12.61.12

Reader End of File

In the 1410, reader EOF gives buffer condition which sets the E-channel condition status latch at status sample A of the next read op code. In 1401 mode, condition is not set for the reader. No status sample A is developed for 1401 card print op codes. EOF is tested for by the 1401 condition test op code B (I) A. Execution of this branch instruction is shown in Figure 31.

1401 Punch Errors

Errors on a 1401 mode punch transfer scan will bring up the E-channel data check bus. Errors occurring during the 1402 punch cycle will bring up E-channel condition bus. The two types of errors are covered separately.

Punch transfer error (E-Ch data check):

SIGNAL	CONTROL	LOGIC
Punch transfer check latch	Punch scan Punch transfer request Error 1	52.12.01
Buffer error	Punch transfer check Sel unit 4	51.40.21
E-ch check bus	Buffer error E-ch sel any buffer	12.62.03
E-ch data check latch	E-ch check bus Status sample B	12.62.04
Master error sample	Card print sample B	18.14.08
1401 card print error	E-ch check latch E-ch second sample B I-O check stop switch on 1401 card print in process	13.70.03
Master error	Master error sample 1401 card print error	18.14.08

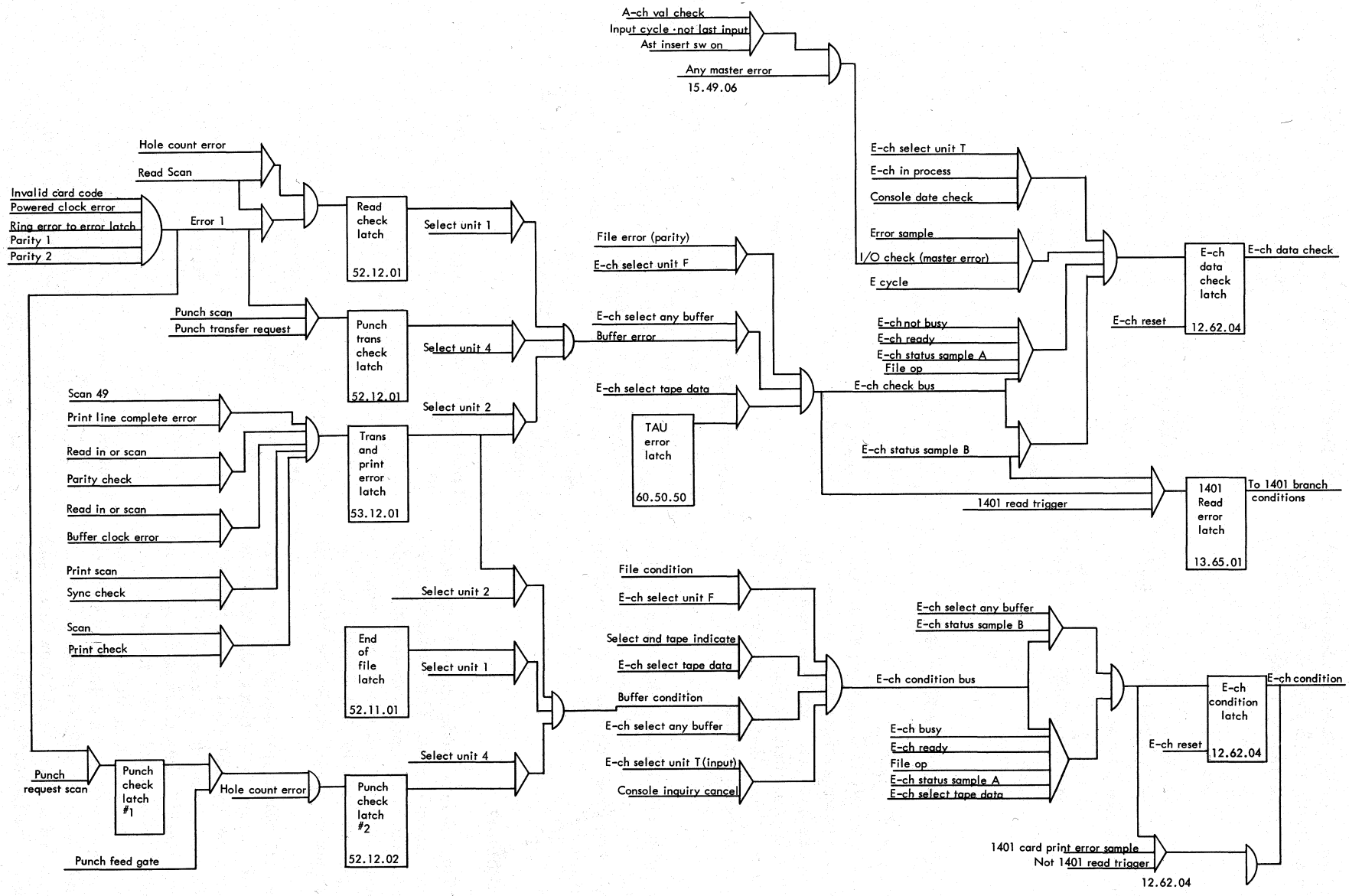


Figure 27. Data Check and Condition Status Latches

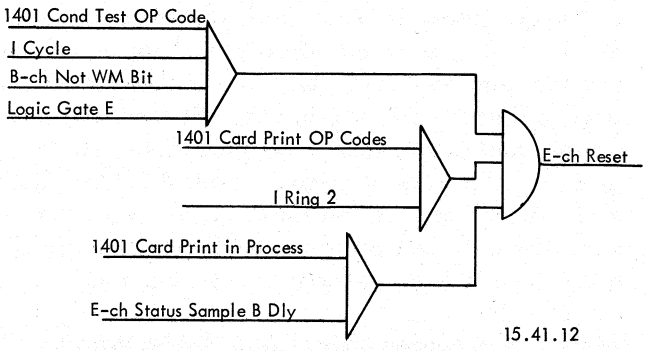


Figure 28. 1401 Mode E-Channel Resets

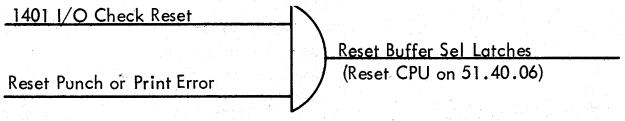


Figure 29. 1401 Mode Reset Buffer Select Latches

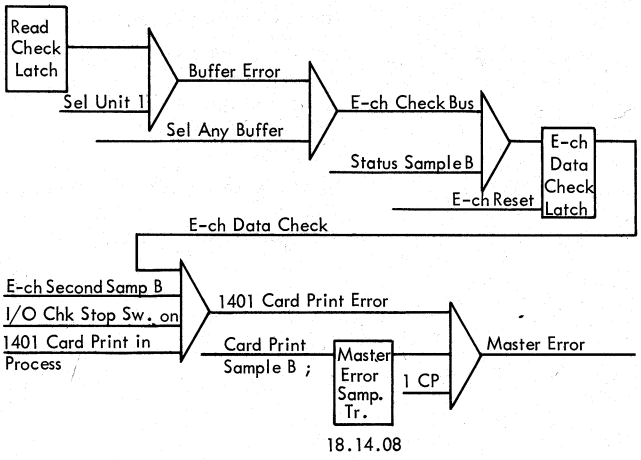


Figure 30. 1401 Mode Read Error

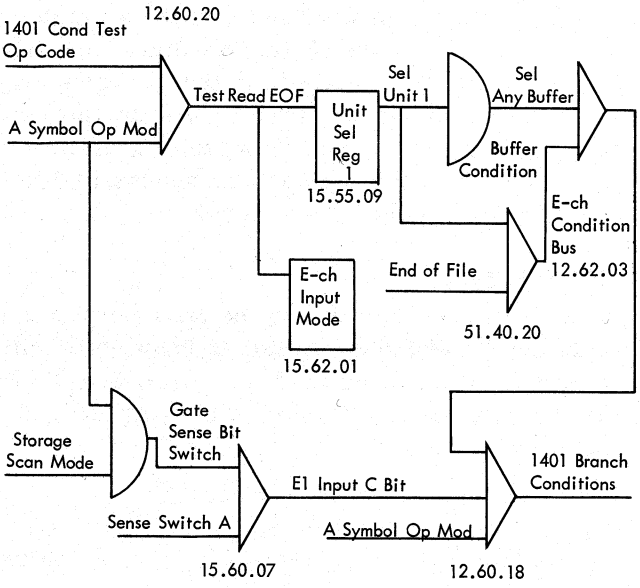


Figure 31. 1401 Mode Test Reader EOF

The system will stop with punch and data check indicators on. Punch cycle was taken for this instruction. If the I-O check stop switch is off, the system will not stop. In this case, program may contain a branch on punch error instruction. This would generate a reset to the punch transfer check latch by bringing up "reset buffer sel latches" on 13. 70. 04.

Errors on 1402 punch cycles (E-Ch condition):

SIGNAL	CONTROL	LOGIC
Punch check latch 1	Error 1 Punch request scan	52.12.02
Punch check latch 2	Hole count error or Punch check latch 1 on previous punch cycle	52.12.02
Buffer condition	Punch check latch 2 Sel unit 4 (second 4 instruction after instruction read out for error card)	51.40.20
E-ch condition bus	Buffer condition Sel any buffer	12.62.03
E-ch condition latch	1401 card print error sample (not) 1401 read trigger E-ch condition bus	12.62.04
Master error sample	1401 card print error sample	18.14.08
1401 punch print error	E-ch check latch E-ch second sample B I-O check stop switch on 1401 card print in process	13.70.03
Master error	1401 punch print error Master error sample 1 CP	18.14.08

The CPU stops with write and data check indicators on. A card did not punch for this instruction. The error card has been stacked. If the I-O check stop switch is off, a branch on punch error instruction will cause "reset punch or print error" to be generated on 12. 61. 12. This will bring up "reset buffer sel latches" which will reset punch check latch 2.

1401 Print Errors

Print errors may occur during the print transfer scan or during the 1403 print operation. Errors during the transfer scan will bring up buffer error. Errors during the 1403 print operation will bring up buffer condition.

Print transfer errors (E-Ch data check):

SIGNAL	CONTROL	LOGIC
Transfer and print error latch	See 53.12.01	53.12.01
Buffer error	Transfer and print error latch Sel unit 2	51.40.21
E-ch check bus	Buffer error E-ch sel any buffer	12.62.03
E-ch data check latch	E-ch check bus E-ch status sample B	12.62.04
Master error sample	Card print sample B	18.14.08
1401 card print error	E-ch data check latch E-ch second sample B I-O check stop switch on 1401 card print in process	13.70.03
Master error	1401 card print error Master error sample 1 CP	18.14.08

The CPU stops with data check and write indicators on. An error line prints. If I-O check stop is off, the program continues and a branch will take place on a B (I) ≠ instruction. The branch will cause the print transfer and check latch to be reset. Note that "reset buffer select latches" at second sample B is blocked on a 1401 punch or print operation. If this were not so, a transfer scan error could not be detected later in the program on a B (I) ≠ or B (I) ! instruction.

1403 print operation errors (E-Ch condition):

SIGNAL	CONTROL	LOGIC
Transfer and print error latch	See 53.12.01	53.12.01
Print error status	Transfer and print error latch Sel unit 2 (this will be at next print instruction)	53.42.02
Buffer condition	Print error status	51.40.20
E-ch condition bus	Buffer condition E-ch sel any buffer	12.62.03
E-ch condition latch	1401 card print error sample E-ch condition bus (not) 1401 read trigger	12.62.04
1401 punch print error	1401 I-O check stop switch on E-ch condition latch 1401 card print error sample	13.70.03
Master error	Master error sample 1 CP 1401 punch print error	18.14.08

COMMENT SHEET

IBM 1410 - 1401 COMPATIBILITY

CUSTOMER ENGINEERING MANUAL OF INSTRUCTION. FORM 223-2597

FROM

NAME _____

OFFICE NO. _____

FOLD

CHECK ONE OF THE COMMENTS AND EXPLAIN IN THE SPACE PROVIDED

FOLD

☐ SUGGESTED ADDITION (PAGE , TIMING CHART, DRAWING, PROCEDURE, ETC.)

☐ SUGGESTED DELETION (PAGE)

☐ ERROR (PAGE)

EXPLANATION

CUT ALONG LINE

FOLD

FOLD

NO POSTAGE NECESSARY IF MAILED IN U.S.A.

FOLD ON TWO LINES, STAPLE, AND MAIL

STAPLE STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 81

POUGHKEEPSIE, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
P.O. BOX 390
POUGHKEEPSIE, N.Y.

ATTN: CE MANUALS, DEPARTMENT 296

CUT ALONG LINE

FOLD FOLD

1/63:2M-FW-64

STAPLE STAPLE

IBM 1410 Printed in U.S.A. 223-2597

IBM

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, New York