ERA 1103

TIMING SEQUENCES

Appendix B
of
PX 71209

Remington Rand
ENGINEERING RESEARCH ASSOCIATES DIVISION

1902 WEST MINNEHAHA AVE. ST. PAUL W4 MINNESOTA

PX71209

APPENDIX B

TIMING SEQUENCES

1. GENERAL

Appendix B lists the exact operational sequences which are performed in the execution of the instructions. The format relates the discrete operations of each sequence in such manner that the time of occurrence of each operation is clearly delineated. Two distinct sets of sequences are presented: Command Timing Sequences and Subcommand Timing Sequences.

2. COMMAND TIMING SEQUENCES

The Command Timing Sequences concern those signals, called "commands", which are generated within the Command Timing Circuits (CTC) as a result of combining translated Operation Codes with timing signals (MAIN PULSES or MP's) generated by the Main Pulse Distributor. The resultant commands effect the principal steps in the execution of the instructions with regard to time, and, in addition, initiate necessary subsequences which are governed by subcommands.

The Command Timing sequences are arranged in a numerical order according to the octal values of the Operation Codes. A sheet listing the so-called "Instruction Reference Commands" precedes the first program instruction (Transmit Positive). The Instruction Reference Commands conclude the non-repeated execution of each instruction and are generated by MP6 and MP7; since the majority of the instructions are so terminated, the redundancy of concluding each instruction sheet with these commands is avoided.

The format employed for each instruction sheet contains the title of the instruction followed by its abbreviation, the octal operation code, a description of the function performed, and a tabular list of the commands. The tables have four columns, the contents of which are explained below:

a. MP. - This column lists the MAIN PULSES used in the generation of the commands. All commands listed between horizontal lines are generated simultaneously by the MP in this column with the exception of indented commands which will be explained in subparagraph b. below.

b. COMMAND. - This column lists the commands generated by the MP's. Each command produced directly by a MAIN PULSE is set to the left. Many commands, however, automatically and simultaneously produce one or more subservient commands; such subservient commands are indented and immediately follow the MP-produced command with which they are associated.

c. SOURCE. - The SOURCE column contains abbreviations which refer to the block diagrams in Volume 7 concerning the Command Timing Circuits. These block diagrams are subdivided into lesser portions each associated with a particular portion of the Control Section. For instance, the abbreviation "CTC-SCC" refers to the Storage Class Control portion of the Command Timing Circuits diagram

PX 71209

which shows diagrammatically the development of the particular command by combining the translated Operation Code with a MAIN PULSE.

d. DESTINATION. - The DEST. Column shows the circuit to which the command is directed. In some cases the desired result is effected directly by the command. In other cases, the command initiates a subsequence which is controlled by subcommands in the circuit referred to in the DEST. Column; these subsequences are listed in Table B-1.

3. SUBCOMMAND TIMING SEQUENCES

Many of the commands from CTC initiate subsequences that are controlled by subcommands. During the time interval occupied by major portion of the subsequence, the Main Pulse Distributor is stopped. The stopping of MPD is effected by a variety of signals: these are:

a. WAIT INTERNAL REFERENCE
b. WAIT RSC
c. WAIT MT READ
d. WAIT MT WRITE
e. STOP
f. TEST LOCKOUT (in conjunction with an EXTERNAL LOCKOUT ENABLE)

At the conclusion of the subsequence (or in some cases, before the conclusion of the subsequence) a RESUME signal is sent to the Pulse Distributor Control (PDC) which allows the next MP to be issued.

The subsequences are timed by CONTROLLED CLOCK PULSES or pulses from CPD or EPD rather than by MAIN PULSES within CTC as is the case in the main instruction sequences. Two of the Subcommand Timing Sequences are not initiated by CTC Commands; these are: the MDAC Locating Sequence and the ESAC Regeneration Sequence. These subsequences are repeated continuously until a storage reference to MD or ES temporarily interrupts the repetition of the corresponding subsequence.

The Subcommand Timing Sequences together with their respective initiating commands are shown in tabular form in Table B-1. Not included in Table B-1 is the Subcommand Timing Sequence for Unassigned Addresses but is included in the SCC subsequences. The computer will be stopped if a storage reference to an unassigned address is included in an instruction.

PX 71209

TABLE B-1.

COMMANDS WHICH PRODUCE SUB-COMMAND TIMING SEQUENCES USED IN EXECUTING INSTRUCTIONS

WHEN THE FOLLOWING COMMANDS APPEAR IN COMMAND TIMING SEQUENCES,				THEY PRODUCE THESE SUB-COMMAND TIMING SEQUENCES
MP	COMMAND	SOURCE	DEST.	
	Initiate Read	CTC-SCC	SCC	SCC INITIATE READ SEQUENCE (SAR) → MDAC READ SEQUENCE; ESAC READ SEQUENCE; *ARAC READ A or Q SEQUENCE
	Initiate Write (0-35)	CTC-SCC	SCC	SCC INITIATE WRITE (0-35) SEQ (SAR) → MDAC WRITE (0-35) SEQUENCE; ESAC WRITE (0-35) SEQUENCE; *ARAC WRITE A or Q SEQUENCES
	Initiate Write (0-14)	CTC-SCC	SCC	SCC INITIATE WRITE (0-14) SEQ (SAR) → MDAC WRITE (0-14) SEQUENCE; ESAC WRITE (0-14) SEQUENCE
	Initiate Write (15-29)	CTC-SCC	SCC	Same as for (0-14) except that bits (15-29) are involved
	Initiate Read jMT	CTC-MT	MT	MT READING SEQUENCE
	Initiate Write jMT	CTC-MT	MT	MT WRITING SEQUENCE
	Initiate Advance jMT	CTC-MT	MT	MT POSITIONING (ADVANCE) SEQUENCE
	Initiate Back jMT	CTC-MT	MT	MT POSITIONING (BACK) SEQUENCE
	Initiate Print	CTC-OUT	TWR	*TYPEWRITER SEQUENCE
	Initiate High Speed Punch	CTC-OUT	HPR	HIGH SPEED PUNCH SEQUENCE
	Add X to A Subtract X from A Split Add X to A Split Subtract X from A Subtract 1 from A Initiate Logical	CTC-ASC	ASC	ASC {ADD X TO A / SUBTRACT X FROM A / SPLIT ADD X TO A / SPLIT SUBTRACT X FROM A / SUBTRACT 1 FROM A / INITIATE LOGICAL} SEQUENCE
	Initiate Shift A Initiate Shift Q	CTC-SKC	SKC	ASC/SKC {SHIFT A / SHIFT Q} SEQUENCE
	Initiate Scale Factor Initiate Multiply Initiate Divide	CTC-ASC	ASC	ASC/SKC {SCALE FACTOR / MULTIPLY / DIVIDE} SEQUENCE
	Initiate Repeat	CTC-RSC	RSC	RSC INITIATE REPEAT SEQUENCE
	Initiate End Test	CTC-RSC	RSC	RSC END TEST (NO JUMP) SEQUENCE
	Initiate Jump Terminate (Initiate End Test)	CTC-RSC	RSC	RSC END TEST (WITH JUMP) SEQUENCE

*There are several types of these sequences. The particular ARAC READ or WRITE SEQUENCE that is executed depends on the instruction being executed; the particular TYPEWRITER SEQUENCE that is carried out depends on what is transmitted to TWR from X.

PX 71209

LOCATION OF TABLES

PX 71209

Loading Instruction: ASSEMBLE DATA (00, 02, or 04)

OPERATION CODE: 00 02 04

Left circular shift (Q) by six places, thus leaving the six lower order stages of the Q-Register clear for receiving the next six bits of information from the PT reader. Remove the loading instruction from MCR: then wait for the next feed pulse.

	COMMAND	SOURCE	DEST.
Feed Pulse	Start Controlled Clock Pulses	Input	CRC
LDP	Shift (Q) Left 1	CTC-AR	Q
LDP	Shift (Q) Left 1	CTC-AR	Q
LDP	Shift (Q) Left 1	CTC-AR	Q
LDP	Shift (Q) Left 1	CTC-AR	Q
LDP	Shift (Q) Left 1	CTC-AR	Q
LDP	Shift (Q) Left 1	CTC-AR	Q
LP 6	Set MPD to 1	MPD	MPD
MP 6	Clear PCR	CTC-PCR	PCR
MP 6	Wait Operation Step (Stop Controlled Clock Pulses)	MPD	CRC

PX 71209

Loading
Instruction: ENTER DATA (01)

OPERATION CODE: 01

Enter the word that has been assembled in the Q-Register at the address indicated by PAK. Then wait for the next feed pulse.

	COMMAND	SOURCE	DEST.
Feed Pulse	Start Controlled Clock Pulses	Input	CRC
MP 1	Set SAR to Q Address	CTC-SAR	SAR
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
MP 2	Clear Q	CTC-AR	Q
MP 3	- - - -		
MP 4	- - - -		
MP 5	Transmit PAK to SAR	CTC-PAK	PAK
	Advance PAK	CTC-PAK	PAK
	Initiate Write (0-35) (see note)	CTC-SCC	SCC
LP 6	Set MPD to 1	MPD	MPD
MP 6	Clear PCR	CTC-PCR	PCR
MP 6	Wait Operation Step (Stop Controlled Clock Pulses)	MPD	CRC

Note: No wait is generated with the INITIATE WRITE command. Thus, assemble data operations may continue during the storage reference.

PX 71209

Loading
Instruction: INSERT ADDRESS (03) OPERATION CODE: 03

Set PAK to the address which has been assembled in the Q-Register. Then wait for the next feed pulse.

	COMMAND	SOURCE	DEST.
Feed Pulse	Start Controlled Clock Pulses	Input	CRC
MP 1	Set SAR to Q Address	CTC-SAR	SAR
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
MP 2	Clear Q	CTC-AR	Q
	Transmit (X) to PCR (Initiate 2 us delay)	CTC-AR	X/CRC
MP 3	Transmit VAK to SAR	CTC-PCR	VAK
	Clear PAK	CTC-PAK	PAK
MP 4	Transmit SAR to PAK	CTC-SAR	SAR
MP 5	- - - -		
LP 6	Set MPD to 1	MPD	MPD
MP 6	Clear PCR	CTC-PCR	PCR
MP 6	Wait Operation Step (Stop Controlled Clock Pulses)	MPD	CRC

PX 71209

Loading
Instruction: CHECK ADDRESS (05)

OPERATION CODE: 05

Compare the address in PAK with the address which has been assembled in the Q-Register. If the two addresses are the same, advance the load check counter and continue loading. If the two addresses differ, generate a Fault Stop at the beginning of the next assemble data loading operation.

Feed Pulse	COMMAND		SOURCE	DEST.
Feed Pulse	Start Controlled Clock Pulses		Input	CRC
MP 1	Set SAR to Q Address		CTC-SAR	SAR
	Initiate Read		CTC-SCC	SCC
	Clear X		CTC-AR	X
	Wait Internal Reference		CTC-PDC	PDC
MP 2	Clear Q		CTC-AR	Q
	Transmit (X) to PCR (Initiate 2 us delay)		CTC-AR	X/CRC
MP 3	Transmit VAK to SAR		CTC-PCR	VAK
	Complement PAK - Set Load Check FF		CTC-PAK	PAK/Fault
MP 4	Transmit SAR to PAK		CTC-SAR	SAR
MP 5	(Transmit PAK to SAR)		CTC-PAK	PAK
	Advance PAK		CTC-PAK	PAK
	If PAK ≠ 40000	If PAK = 40000		
	- - - -	PAK End Carry	PAK	Fault
	- - - - (Load Check FF remains set)	Clear Load Ck FF	Fault	Fault
	- - - -	Adv Load Ck Counter	Fault	SC Pan
LP 6	Set MPD to 1	Set MPD to 1	MPD	MPD
MP 6	Clear PCR	Clear PCR	CTC-PCR	PCR
MP 6	Wait Operation Step	Wait Operation Step	MPD	CRC

Next assemble data operation

MP 1	Test Load Fault	- - - - (continue through	MPD	Fault
	Set Load Fault FF	- - - - operation)	Fault	Fault

PX 71209

Instruction Reference Commands

The non-repeated execution of every program instruction is concluded as MPD advances through MP 6 and MP 7. Commands issued on these MP's extract the next instruction from storage and prepare the computer for the execution. These Instruction Reference Commands are not listed under the separate instruction headings, but they are understood to conclude each non-repeated execution of every program instruction. The foregoing does not apply to the Repeat (RPjnw) instruction which is concluded differently.

	COMMAND	SOURCE	DEST.
MP 6	Clear PCR	CTC-PCR	PCR
MP 6	(Clear RSC)	CTC-RSC	RSC
OP 6	Transmit PAK to SAR	CTC-PAK	PAK
	Advance PAK	CTC-PAK	PAK
OP 6	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
MP 7	Clear SAR	CTC-SAR	SAR
MP 7	Transmit (X) to PCR (Initiate 2 us delay)	CTC-AR	X/CRC

PX 71209

Instruction: TRANSMIT POSITIVE (TPuv) OPERATION CODE: 11

Replace (v) with (u)

MP	COMMAND	SOURCE	DEST
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
5	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

OPERATION CODE: 12

Instruction: TRANSMIT MAGNITUDE (TMuv)

Replace (V) with the absolute magnitude of (u).

MP	COMMAND		SOURCE	DEST.
0	Clear X		CTC-AR	X
	Transmit UAK to SAR		CTC-PCR	UAK
	Initiate Read		CTC-SCC	SCC
	Clear X		CTC-AR	X
	Wait Internal Reference		CTC-PDC	PDC
1	- - - -			
	If (X) is positive	If (X) is negative		
5	- - - -	Complement (X)	CTC-AR	X
	Transmit VAK to SAR	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Ref.	Wait Internal Ref.	CTC-PDC	PDC

Note: MP 1 provides additional time for MCT to detect the sign of (X) and respond.

PX 71209

OPERATION CODE: 13

Instruction: TRANSMIT NEGATIVE (TNuv)

Replace (v) with the complement of (u)

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
5	Complement (X)	CTC-AR	X
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

Instruction: INTERPRET (IP--)

OPERATION CODE: 14

Let Y represent the address from which CI was obtained. Replace the right-hand 15 bits of (F_1) with the quantity Y plus 1. Then take (F_2) as the next instruction.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit PAK to SAR	CTC-PAK	PAK
	Advance PAK	CTC-PAK	PAK
1	Transmit SAR to X	CTC-SAR	SAR
	Set SAR to F_1	CTC-SAR	SAR
	Clear PAK	CTC-PAK	PAK
	(Clear RSC)	CTC-RSC	RSC
	Initiate Write (0-14)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC
2	Transmit PAK to SAR	CTC-PAK	PAK
	Advance PAK (see note)	CTC-PAK	PAK
5	- - -	- - -	- - -

Note: At this point PAK contains 00001 which is Fixed Address F_2. The NI will then be taken from F_2.

PX 71209

OPERATION CODE: 15

Instruction: TRANSMIT U-ADDRESS (TUuv)

Replace the 15 bits of (v), designated by v_{15} through v_{29}, with the corresponding bits of (u), leaving the remaining 21 bits of (v) undisturbed.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
5	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (15-29)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

Note: The Accumulator and the Q-Register are not acceptable "v" execution addresses for this instruction.

PX 71209

OPERATION CODE: 16

Instruction: TRANSMIT V-ADDRESS (TV uv)

Replace the right-hand 15 bits of (v), designated by v_0 through v_{14}, with the corresponding bits of (u), leaving the remaining 21 bits of (v) undisturbed.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
5	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-14)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

Note: The Accumulator and the Q-Register are not acceptable "v" execution addresses for this instruction.

Instruction: EXTERNAL FUNCTION (EF-v)

OPERATION CODE: 17

Select a unit of external equipment and perform the function designated by (v).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Test Input-Output Lockouts	MPD	PDC
5	Transmit (X) to IOB (Set Select F.F)	CTC-IO	IOB
	Initiate Lockout IOB Write	CTC-IO	PDC

PX 71209

Instruction: REPLACE ADD (RAuv) OPERATION CODE: 21

Form in A the sum of D(u) and D(v). Then replace (u) with (A_R).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
2	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
3	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
4	Clear X	CTC-AR	X
5	Transmit (A_R) to X	CTC-AR	A
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

Instruction: REPLACE SUBTRACT (RSuv) OPERATION CODE: 23

Form in A the difference D(u) minus D(v). Then replace (u) with (A_R).

MP	COMMAND	SOURCE	DEST
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
2	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
3	Subtract (X) from (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
4	Clear X	CTC-AR	X
5	Transmit (A_R) to X	CTC-AR	A
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

Instruction: CONTROLLED COMPLEMENT (CCuv) OPERATION CODE: 27

Replace (A_R) with (u) leaving (A_L) undisturbed. Then complement those bits of (A_R) that correspond to ones in (v). Then replace (u) with (A_R).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A (see note)	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Complement (X)	CTC-AR	X
2	Transmit (X') to A_R	CTC-AR	A
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
3	Complement (X)	CTC-AR	X
4	Transmit (X') to A_R	CTC-AR	A
	Clear X	CTC-AR	X
5	Transmit (A_R) to X	CTC-AR	A
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

Note: The presence of the MCT 27 operation code enable blocks the CLEAR A_L signal in ARAC. Thus, as a result of the INITIATE CLEAR A command, only A_R is cleared.

PX 71209

OPERATION CODE: 31

Instruction: SPLIT POSITIVE ENTRY (SPuk)

Form S(u) in A. Then left circular shift (A) by k places. (The value of k must not exceed seven bits, i.e., bits V_7 through V_{14} must contain zeros).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Transmit VAK to SAR	CTC-PCR	VAK
	Split Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
5	Initiate Shift (A)	CTC-SKC	SKC
	Wait Internal Reference (see note)	CTC-PDC	PDC

Note: No wait is generated if "k" is zero.

PX 71209

OPERATION CODE: 32

Instruction: SPLIT ADD (SAuk)

Add S(u) to (A). Then left circular shift (A) by k places. (The value of k must not exceed seven bits, i.e., bits V_7 through V_{14} must contain zeros).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Transmit VAK to SAR	CTC-PCR	VAK
	Split Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
5	Initiate Shift (A)	CTC-SKC	SKC
	Wait Internal Reference (see note)	CTC-PDC	PDC

Note: No wait is generated if "k" is zero.

PX 71209

OPERATION CODE: 33

Instruction: SPLIT NEGATIVE ENTRY (S$\bar{N}$uk)

Form in A the complement of S(u). Then left circular shift (A) by k places. (The value of k must not exceed seven bits, i.e., bits V_7 through V_{14} must contain zeros).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Transmit VAK to SAR	CTC-PCR	VAK
	Split Subtract (X) from (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
5	Initiate Shift (A)	CTC-SKC	SKC
	Wait Internal Reference (see note)	CTC-PDC	PDC

Note: No wait is generated if "k" is zero.

PX 71209

OPERATION CODE: 34

Instruction: SPLIT SUBTRACT (SSuk)

Subtract S(u) from (A). Then left circular shift (A) by k places. (The value of k must not exceed seven bits, i.e., bits V_7 through V_{14} must contain zeros).

MP	COMMAND	SOURCE	DIST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Transmit VAK to SAR	CTC-PCR	VAK
	Split Subtract (X) from (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
5	Initiate Shift (A)	CTC-SKC	SKC
	Wait Internal Reference (see note)	CTC-PDC	PDC

Note: No wait is generated if "k" is zero.

OPERATION CODE: 35

Instruction: ADD AND TRANSMIT (ATuv)

Add D(u) to (A). Then replace (v) with (A_R).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
2	Clear X	CTC-AR	X
5	Transmit (A_R) to X	CTC-AR	A
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

OPERATION CODE: 36

Instruction: SUBTRACT AND TRANSMIT (STuv)

Subtract D(u) from (A). Then replace (v) with (A_R).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Subtract (X) from (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
2	Clear X	CTC-AR	X
5	Transmit (A_R) to X	CTC-AR	A
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

OPERATION CODE: 37

Instruction: RETURN JUMP (RJuv)

Let y represent the address from which CI was obtained. Replace the right-hand 15 bits of (u) with the quantity y plus 1. Then take (v) as NI.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit PAK to SAR	CTC-PAK	PAK
	Advance PAK	CTC-PAK	PAK
1	Transmit SAR to X	CTC-SAR	SAR
	Clear SAR	CTC-SAR	SAR
2	- - - -		
3	Transmit VAK to SAR	CTC-PCR	VAK
	Clear PAK	CTC-PAK	PAK
	(Clear RSC)	CTC-RSC	RSC
4	Transmit SAR to PAK	CTC-SAR	SAR
	Clear SAR	CTC-SAR	SAR
5	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Write (0-14)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

Note: MP 2 used for delay only

PX 71209

OPERATION CODE: 41

Instruction: INDEX JUMP (IJuv)

Form in A the difference D(u) minus 1. Then if $A_{71} = 1$, continue the present sequence of instructions; if $A_{71} = 0$, replace (u) with (A_R) and take (v) as NI.

MP	COMMAND		SOURCE	DEST.
0	Clear X		CTC-AR	X
	Initiate Clear A		CTC-ARAC	ARAC
	Transmit UAK to SAR		CTC-PCR	UAK
	Initiate Read		CTC-SCC	SCC
	Clear X		CTC-AR	X
	Wait Internal Reference		CTC-PDC	PDC
1	Add (X) to (A)		CTC-ASC	ASC
	Wait Internal Reference		CTC-PDC	PDC
2	Transmit VAK to SAR		CTC-PCR	VAK
	Subtract 1 from (A)		CTC-ASC	ASC
	Wait Internal Reference		CTC-PDC	PDC
	If (A) is positive	If (A) is negative		
3	Clear PAK	- - - -	CTC-PAK	PAK
	(Clear RSC)	- - - -	CTC-RSC	RSC
4	Transmit SAR to PAK	- - - -	CTC-SAR	SAR
	Clear SAR	Clear SAR	CTC-SAR	SAR
	Clear X	Clear X	CTC-AR	X
5	Transmit (A_R) to X	Transmit (A_R) to X	CTC-AR	A
	Transmit UAK to SAR	- - - -	CTC-PCR	UAK
	Initiate Write (0-35)	- - - -	CTC-SCC	SCC
	Wait Internal Ref.	- - - -	CTC-PDC	PDC

PX 71209

OPERATION CODE: 42

Instruction: THRESHOLD JUMP (TJuv), not repeated

Subtract (u) from (A). If A_{71} is then 1, take (v) as the next instruction; if A_{71} is then 0, continue the present sequence of instructions. Then, in either case, restore (A) to its initial state.

MP	COMMAND		SOURCE	DEST.
0	Clear X		CTC-AR	X
	Transmit UAK to SAR		CTC-PCR	UAK
	Initiate Read		CTC-SCC	SCC
	Clear X		CTC-AR	X
	Wait Internal Reference		CTC-PDC	PDC
1	Subtract (X) from (A)		CTC-ASC	ASC
	Wait Internal Reference		CTC-PDC	PDC
2	- - - -			
	If (A) is positive	If (A) is negative		
3	Complement (X).	Complement (X)	CTC-AR	X
	- - - -	Clear PAK	CTC-PAK	PAK
	- - - -	Transmit VAK to SAR	CTC-PCR	VAK
5	- - - -	Transmit SAR to PAK	CTC-SAR	SAR
	- - - -	Clear SAR	CTC-SAR	SAR
	Add (X) to (A)	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Ref.	Wait Internal Ref.	CTC-PDC	PDC

Note: MP 2 provides additional time for MCT to detect the sign of (A) and respond.

PX 71209

OPERATION CODE: 42

Instruction: THRESHOLD JUMP (TJuv), repeated

Subtract (u) from (A). If A_{71} is then 1, replace (Q) with jn-r and take (v) as the next instruction; if A_{71} is then 0, continue with the present sequence of instructions. Then, in either case, restore (A) to its initial state (see note 1).

MP	COMMAND		SOURCE	DEST.
0	Clear X		CTC-AR	X
	Transmit UAK to SAR		CTC-PCR	UAK
	Initiate Read		CTC-SCC	SCC
	Clear X		CTC-AR	X
	Wait Internal Reference		CTC-PDC	PDC
1	Subtract (X) from (A)		CTC-ASC	ASC
	Wait Internal Reference		CTC-PDC	PDC
2	- - - -			
	If (A) is positive	If (A) is negative		
3	Complement (X)	Complement (X)	CTC-AR	X
	- - - -	Complement PAK	CTC-PAK	PAK
5	Add (X) to (A)	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Ref.	Wait Internal Ref.	CTC-PDC	PDC
	- - - -	Initiate Jump Terminate	CTC-RSC	RSC
	- - - -	Transmit PAK to SAR	CTC-PAK	PAK
	- - - -	(Advance PAK)	CTC-PAK	PAK
	- - - -	Clear MCR	CTC-PCR	MCR
	Initiate End Test	(Initiate End Test)	CTC-RSC	RSC
	Advance PAK	(Advance PAK)	CTC-PAK	PAK
	Wait RSC	Wait RSC (see jump termination)	CTC-PDC	PDC

Note: 1. This instruction is preceded by instruction 75jnw which leaves the complement of "jn" in PAK for controlling the execution of this instruction.
2. MP 2 provides additional time for MCT to detect the sign of (A) and respond.

PX 71209

OPERATION CODE: 43

Instruction: EQUALITY JUMP (EJuv), not repeated

Subtract (u) from (A). If (A) is then zero, take (v) as the next instruction; if (A) is then not zero, continue the present sequence. Then in either case, restore (A) to its initial state.

MP	COMMAND		SOURCE	DEST.
0	Clear X		CTC-AR	X
	Transmit UAK to SAR		CTC-PCR	UAK
	Initiate Read		CTC-SCC	SCC
	Clear X		CTC-AR	X
	Wait Internal Reference		CTC-PDC	PDC
1	Subtract (X) from (A)		CTC-ASC	ASC
	Wait Internal Reference		CTC-PDC	PDC
2	Complement (X)		CTC-AR	X
	Subtract 1 from (A)		CTC-ASC	ASC
	Wait Internal Reference		CTC-PDC	PDC
	If (A) was zero	If (A) was not zero		
3	Transmit VAK to SAR	- - - -	CTC-PCR	VAK
	Clear PAK	- - - -	CTC-PAK	PAK
	Add (X) to (A)	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Ref.	Wait Internal Ref.	CTC-PDC	PDC
4	Transmit SAR to PAK	- - - -	CTC-SAR	SAR
	Clear SAR	- - - -	CTC-SAR	SAR
	Set X to 1	Set X to 1	CTC-AR	X
5	Add (X) to (A)	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Ref.	Wait Internal Ref.	CTC-PDC	PDC

PX 71209

OPERATION CODE: 43

Instruction: EQUALITY JUMP (EJuv), repeated

Subtract (u) from (A). If (A) is then zero, replace (Q) with jn-r and take (v) as the next instruction; if (A) is then not zero, repeat the execution. Then, in either case, restore (A) to its initial state (see note).

MP	COMMAND		SOURCE	DEST.
0	Clear X		CTC-AR	X
	Transmit UAK to SAR		CTC-PCR	UAK
	Initiate Read		CTC-SCC	SCC
	Clear X		CTC-AR	X
	Wait Internal Reference		CTC-PDC	PDC
1	Subtract (X) from (A)		CTC-ASC	ASC
	Wait Internal Reference		CTC-PDC	PDC
2	Complement (X)		CTC-AR	X
	Subtract 1 from (A)		CTC-ASC	ASC
	Wait Internal Reference		CTC-PDC	PDC
	If (A) was zero	If (A) was not zero		
3	Complement PAK	- - - -	CTC-PAK	PAK
	Add (X) to (A)	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	Wait Internal Ref.	CTC-PDC	PDC
4	Set X to 1	Set X to 1	CTC-AR	X
5	Add (X) to (A)	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	Wait Internal Ref.	CTC-PDC	PDC
	Initiate Jump Terminate	- - - -	CTC-RSC	RSC
	Transmit PAK to SAR	- - - -	CTC-PAK	PAK
	(Advance PAK)	- - - -	CTC-PAK	PAK
	Clear MCR	- - - -	CTC-PCR	MCR
	(Initiate End Test)	Initiate End Test	CTC-RSC	RSC
	(Advance PAK)	Advance PAK	CTC-PAK	PAK
	Wait RSC (see jump termination)	Wait RSC	CTC-PDC	PDC

NOTE: This instruction is preceded by instruction 75jnw which leaves the complement of "jn" in PAK for controlling the execution of this instruction.

PX 71209

OPERATION CODE: 44

Instruction: Q-JUMP (QJuv)

If $Q_{35} = 1$, take (u) as NI; if $Q_{35} = 0$, take (v) as NI. Then, in either case, left circular shift (Q) by one place.

MP	COMMAND		SOURCE	DEST.
	If (Q) is positive	If (Q) is negative		
0	Clear X	Clear X	CTC-AR	X
	Clear PAK	Clear PAK	CTC-PAK	PAK
	(Clear RSC)	(Clear RSC)	CTC-RSC	RSC
	Transmit VAK to SAR	Transmit UAK to SAR	CTC-PCR	VAK/UAK
5	Transmit SAR to PAK		CTC-SAR	SAR
	Clear SAR		CTC-SAR	SAR
	Shift (Q) Left 1		CTC-AR	Q

PX 71209

Remington Rand
Engineering Research Associates Division

OPERATION CODE: 45

Instruction: MANUALLY SELECTIVE JUMP (MJjv)

If the number j is 0, take (v) as NI. If j is 1, 2, or 3, and the correspondingly numbered MJ selecting switch is set to "jump", take (v) as NI; if this switch is not set to "jump", continue the present sequence.

MP	COMMAND: For jump	COMMAND: For no jump	SOURCE	DEST.
0	Clear X	Clear X	CTC-AR	X
	Transmit VAK to SAR	Transmit VAK to SAR	CTC-PCR	VAK
	Clear PAK	- - - -	CTC-PAK	PAK
	(Clear RSC)	- - - -	CTC-RSC	RSC
5	Transmit SAR to PAK	- - - -	CTC-SAR	SAR
	Clear SAR	Clear SAR	CTC-SAR	SAR

OPERATION CODE: 46

Instruction: SIGN JUMP (SJuv)

If $A_{71} = 1$, take (u) as NI. If $A_{71} = 0$, take (v) as NI.

MP	COMMAND		SOURCE	DEST.
	If (A) is positive	If (A) is negative		
0	Clear X	Clear X	CTC-AR	X
	Clear PAK	Clear PAK	CTC-PAK	PAK
	(Clear RSC)	(Clear RSC)	CTC-RSC	RSC
	Transmit VAK to SAR	Transmit UAK to SAR	CTC-PCR	VAK/UAK
5	Transmit SAR to PAK		CTC-SAR	SAR
	Clear SAR		CTC-SAR	SAR

PX 71209

OPERATION CODE: 47

Instruction: ZERO JUMP (ZJuv)

If (A) is not zero, take (u) as NI: if (A) is zero, take (v) as NI.

<table>
<tr><th>MP</th><th colspan="2">COMMAND</th><th>SOURCE</th><th>DEST.</th></tr>
<tr><td>0</td><td colspan="2">Clear X
Clear PAK
(Clear RSC)
Subtract 1 from (A)
Wait Internal Reference</td><td>CTC-AR
CTC-PAK
CTC-RSC
CTC-ASC
CTC-PDC</td><td>X
PAK
RSC
ASC
PDC</td></tr>
<tr><td rowspan="2">1</td><td>If (A) was zero</td><td>If (A) was not zero</td><td></td><td></td></tr>
<tr><td>Transmit VAK to SAR
Set X to 1</td><td>Transmit UAK to SAR
Set X to 1</td><td>CTC-PCR
CTC-AR</td><td>VAK/UAK
X</td></tr>
<tr><td>5</td><td colspan="2">Transmit SAR to PAK
Clear SAR
Add (X) to (A)
Wait Internal Reference</td><td>CTC-SAR
CTC-SAR
CTC-ASC
CTC-PDC</td><td>SAR
SAR
ASC
PDC</td></tr>
</table>

PX 71209

Instruction: Q-CONTROLLED TRANSMIT (QTuv)

OPERATION CODE: 51

Form in A the number L(Q)(u). Then replace (v) by (A_R).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	AR
	Wait Internal Reference	CTC-PDC	PDC
1	Transmit (Q') to X'	CTC-AR	Q
2	Split add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
5	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

OPERATION CODE: 52

Instruction: Q-CONTROLLED ADD (QAuv)

Add to (A) the number L(Q)(u). Then replace (v) by (A_R).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Transmit (Q') to X'	CTC-AR	Q
2	Split Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
3	Clear X	CTC-AR	X
5	Transmit (A_R) to X	CTC-AR	A
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

OPERATION CODE: 53

Instruction: Q-CONTROLLED SUBSTITUTE (QSuv)

Form in A the quantity L(Q)(u) plus L (Q') (v). Then replace (v) with (A_R). The effect is to replace selected bits of (v) with the corresponding bits of (u) in those places corresponding to 1's in Q. The final (v) is the same as the final (A_R).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Initiate Logical (see note)	CTC-ASC	ASC
	Extend Arithmetic Sequence	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
2	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
3	Initiate Logical (see note)	CTC-ASC	ASC
	Extend Arithmetic Sequence	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
5	Transmit (A_R) to X	CTC-AR	A
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

Note: At the conclusion of the first logical operation (Q) is complemented in preparation for the second logical operation. At the conclusion of the second logical operation (Q') is complemented, thus restoring (Q) to its initial value.

PX 71209

OPERATION CODE: 54

Instruction: LEFT SHIFT IN A (LAuk)

Replace (A) with D(u). Then left circular shift (A) by k places. Then replace (u) with (A_R). If u = a, the first step is omitted, so that the initial content of A is shifted. (The value of k must not exceed seven bits, i.e., bits V_7 through V_{14} must contain zeros, if SAR at MP5 is to contain the original u-address).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Transmit VAK to SAR	CTC-PCR	VAK
	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
2	Clear X	CTC-AR	X
	Initiate Shift (A)	CTC-SKC	SKC
	Wait Internal Reference (see note)	CTC-PDC	PDC
5	Transmit (A_R) to X	CTC-AR	A
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

Note: No wait is generated if "k" is zero.

PX 71209

OPERATION CODE: 55

Instruction: LEFT SHIFT IN Q (LQuk)

Replace (Q) with (u). Then left circular shift (Q) by k places. Then replace (u) with (Q). (The value of k must not exceed seven bits, i.e., bits V_7 through V_{14} must contain zeros, **if SAR at MP5 is to contain the original u-address**).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Clear Q	CTC-AR	Q
2	Transmit (X) to Q	CTC-AR	X
	Clear X	CTC-AR	X
	Transmit VAK to SAR	CTC-PCR	VAK
3	Complement (X)	CTC-AR	X
	Initiate Shift (Q)	CTC-SKC	SKC
	Wait Internal Reference (see note)	CTC-PDC	PDC
5	Transmit (Q') to X'	CTC-AR	Q
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

Note: No wait is generated if "k" is zero.

PX 71209

OPERATION CODE: 56

Instruction: MANUALLY SELECTIVE STOP (MSjv)

If j = 0, stop the computer operation and provide suitable indication. If j is 1, 2, or 3 and the correspondingly numbered MS selecting switch is set to "stop", stop the computer operation and provide suitable indication. Whether or not a stop occurs, (v) is NI.

MP	COMMAND		SOURCE	DEST.
0	Clear X		CTC-AR	X
	Clear PAK		CTC-PAK	PAK
	(Clear RSC)		CTC-RSC	RSC
	Transmit VAK to SAR		CTC-PCR	VAK
	For Stop	For no Stop		
5	Transmit SAR to PAK	Transmit SAR to PAK	CTC-SAR	SAR
	Clear SAR	Clear SAR	CTC-SAR	SAR
	Stop Clock	- - - -	CTC-Stop	Stop
	Stop	- - - -	Stop	PDC/CRC

PX 71209

OPERATION CODE: 57

Instruction: FINAL STOP (FS--)

Stop computer operation and provide suitable indication.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Stop Clock	CTC-Stop	Stop
	Stop	Stop	PDC/CRC

Note: To resume operation following a final stop, new selections must be made.

OPERATION CODE: 61

Instruction: PRINT (PR-v)

Replace (TWR) with the right-hand 6 bits of (v). Cause the typewriter to perform the operation corresponding to the 6-bit code.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Test Lockout	MPD	PDC
2	Initiate Print (see note)	CTC-OUT	TWR
	Initiate Lockout TWC	CTC-OUT	PDC
3	- - - -		
4	- - - -		
5	- - - -		

Note: On the INITIATE PRINT command (X_0 through X_5) is transferred to TWR.

PX 71209

OPERATION CODE: 63

Instruction: PUNCH (PUjv)

Replace (HPR) with the right-hand 6-bits of (v). Cause the punch to respond to (HPR). If j = 0, omit seventh level hole; if j = 1 include seventh level hole.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Test Lockout	MPD	PDC
2	Initiate High-Speed Punch (see note)	CTC-OUT	HPR
	Initiate Lockout HPC	CTC-OUT	PDC
3	- - - -		
4	- - - -		
5	- - - -		

Note: On the INITIATE HIGH-SPEED PUNCH command (X_0 through X_5) and UAK_{12} is transferred to HPR.

PX 71209

OPERATION CODE: 64

Instruction: READ MAGNETIC TAPE (RMjnv)

Read n blocks from MT unit j (running forward) to 32n consecutive addresses in ES starting with v.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
1	Transmit SAR to X	CTC-SAR	SAR
	Clear Q	CTC-AR	Q
	Test Lockout	MPD	PDC
2	Clear MT "j" Counters	CTC-MT	jMT
3	Transmit (X) to MT "j" BK	CTC-MT	X
4	Clear SAR	CTC-SAR	SAR
5	Initiate Lockout MT "j"	CTC-MT	PDC
	Initiate Read MT "j"	CTC-MT	jMT
	Wait MT Read	CTC-PDC	PDC

Note: One block of magnetic tape consists of 32 words, each of 36 bits.

PX 71209

OPERATION CODE: 65

Instruction: WRITE MAGNETIC TAPE (WMjnv)

From 32n consecutive addresses in ES starting with v, write n blocks on MT unit j (running forward).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
1	Transmit SAR to X	CTC-SAR	SAR
	Test Lockout	MPD	PDC
2	Clear MT "j" Counters	CTC-MT	jMT
3	Transmit (X) to MT "j" BK	CTC-MT	X
4	Clear SAR	CTC-SAR	SAR
5	Initiate Lockout MT "j"	CTC-MT	PDC
	Initiate Write MT "j"	CTC-MT	jMT
	Wait MT Write	CTC-PDC	PDC

Note: One block of magnetic tape consists of 32 words, each of 36 bits.

PX 71209

OPERATION CODE: 66

Instruction: ADVANCE MAGNETIC TAPE (AMjn-)

Move the magnetic tape of MT unit j in the forward direction by n blocks.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
1	Transmit SAR to X	CTC-SAR	SAR
	Test Lockout	MPD	PDC
2	Clear MT "j" Counters	CTC-MT	jMT
3	Transmit (X) to MT "j" BK	CTC-MT	X
4	Clear SAR	CTC-SAR	SAR
5	Initiate Lockout MT "j"	CTC-MT	PDC
	Initiate Advance MT "j"	CTC-MT	jMT

Note: One block of magnetic tape consists of 32 words, each of 36 bits.

PX 71209

OPERATION CODE: 67

Instruction: BACK MAGNETIC TAPE (BMjn-)

Move the magnetic tape of MT unit j in the backward direction by n blocks.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCP	UAK
1	Transmit SAR to X	CTC-SAR	SAR
	Test Lockout	MPD	PDC
2	Clear MT "j" Counters	CTC-MT	jMT
3	Transmit (X) to MT "j" BK	CTC-MT	X
4	Clear SAR	CTC-SAR	SAR
5	Initiate Lockout MT "j"	CTC-MT	PDC
	Initiate Back MT "j"	CTC-MT	jMT

Note: One block of magnetic tape consists of 32 words, each of 36 bits.

PX 71209

OPERATION CODE: 71

Instruction: MULTIPLY (MPuv)

Form in A the 72-bit product of (u) and (v), leaving in Q the multiplier (u).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Clear Q	CTC-AR	Q
2	Transmit (X) to Q	CTC-AR	X
	Clear X	CTC-AR	X
3	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
5	Initiate Multiply	CTC-ASC	ASC
	Set SK to 36	CTC-SAR	SAR
	Extend Arithmetic Sequence	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

Remington Rand
Engineering Research Associates Division

OPERATION CODE: 72

Instruction: MULTIPLY ADD (MAuv)

Add to (A) the 72-bit product of (u) and (v), leaving in Q the multiplier (u).

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Clear Q	CTC-AR	Q
	Set SK to 36	CTC-SAR	SAR
2	Transmit (X) to Q	CTC-AR	X
	Clear X	CTC-AR	X
	Initiate Shift (A)	CTC-SKC	SKC
	Wait Internal Reference	CTC-PDC	PDC
3	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
5	Initiate Multiply	CTC-ASC	ASC
	Set SK to 36	CTC-SAR	SAR
	Extend Arithmetic Sequence	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

OPERATION CODE: 73

Instruction: DIVIDE (DVuv)

Divide the 72-bit number in (A) by (u), putting the quotient in Q and leaving in A a non-negative remainder, R. Then replace (v) by (Q). The quotient and remainder are defined by: $(A)_i = (u)\cdot(Q)+R$ where $0 \leq R < |(u)|$. Here $(A)_i$ denotes the initial contents of A.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Clear Q	CTC-AR	Q
	Set SK to 36	CTC-SAR	SAR
	Initiate Divide	CTC-ASC	ASC
	Extend Arithmetic Sequence	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
2	Clear X	CTC-AR	X
3	Complement X	CTC-AR	X
5	Transmit (Q') to X'	CTC-AR	Q
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-35)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

PX 71209

OPERATION CODE: 74

Instruction: SCALE FACTOR (SFuv)

Replace (A) with D(u). Then left circular shift (A) by 35 places. Then continue to shift (A) until $A_{34} \neq A_{35}$. Then replace the right hand 15 bits of (v) with the number of shifts, k, which would be necessary to return (A) to its original position. If (A) is all ones, or all zeros, k = 37. If u is a, (A) is left unchanged in the first step, instead of being replaced by $D(A_R)$.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Initiate Clear A	CTC-ARAC	ARAC
	Transmit UAK to SAR	CTC-PCR	UAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
1	Set SK to 36	CTC-SAR	SAR
	Add (X) to (A)	CTC-ASC	ASC
	Wait Internal Reference	CTC-PDC	PDC
2	Clear X	CTC-AR	X
	Initiate Shift (A)	CTC-SKC	SKC
	Wait Internal Reference	CTC-PDC	PDC
3	Initiate Scale Factor	CTC-ASC	ASC
	Set SK to 36	CTC-SAR	SAR
	Wait Internal Reference	CTC-PDC	PDC
4	Transmit SAR to X	CTC-SAR	SAR
	Clear SAR	CTC-SAR	SAR
5	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write (0-14)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC

Instruction: REPEAT (RPjnw) OPERATION CODE: 75

This instruction calls for the next instruction (NIuv) to be executed n times, its "u" and "v" addresses being modified or not according to the value of j. Afterwards the program is continued by the execution of the instruction stored at a fixed address F_1. (See notes on following pages.)

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit VAK to SAR	CTC-PCR	VAK
1	Transmit SAR to X	CTC-SAR	SAR
	Set SAR to F_1	CTC-SAR	SAR
	Initiate Write (0-14)	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC
2	Clear X	CTC-AR	X
	Transmit PAK to SAR	CTC-PAK	PAK
	Advance PAK	CTC-PAK	PAK
	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
3	Clear PAK	CTC-PAK	PAK
	Initiate 75 Sequence (Set 75 FF to "1", Clear Hold Repeat FF, and initiate 2 us delay)	CTC-RSC	RSC/CRC
	Transmit UAK to SAR	CTC-PCR	UAK
	Clear PCR	CTC-PCR	PCR
4	Transmit SAR to PAK	CTC-SAR	SAR
5	Initiate Repeat	CTC-RSC	RSC
	Complement PAK	CTC-PAK	PAK
	Transmit (X) to PCR (Initiate 2 us delay)	CTC-AR	X/CRC
	Wait RSC	CTC-PDC	PDC
7	Clear SAR	CTC-SAR	SAR

NOTES ON THE REPEAT INSTRUCTION

Because of the complexity of the Repeat instruction, the following paragraphs provide necessary, additional information on both the execution of the Repeat instruction and the execution of the repeated instruction.

THE REPEAT INSTRUCTION. - The Repeat instruction has the form 75jnw. The normal values of j, 0 through 3, determine the advance of the execution addresses of the repeated instruction. The code for j is as follows:

if j = 0, neither the u-address or the v-address of the repeated instruction is advanced;

if j = 1, the v-address of the repeated instruction is advanced after each execution;

if j = 2, the u-address of the repeated instruction is advanced after each execution;

if j = 3, both the u-address and the v-address of the repeated instruction are advanced after each execution.

It should be noted that the modification of the u-address and v-address is done in UAK and VAK respectively; therefore, the original form of the instruction in its memory location is unaltered for possible future use.

The value of n determines the number of times the repeated instruction is to be executed. The value of n can vary throughout the range $0 \leq n \leq 2^{12}-1$, or from 0 through 4095. If n is zero, the repeat sequence is terminated immediately, and the following instruction is not executed since the next instruction is taken from fixed address F_1; in this case, the Repeat instruction performs the same function as would a Manually Selective Jump (45jv) in which the value for j is zero.

The repeat termination address w replaces the 15 lower-order bits of the fixed address F_1 (either 00000 or 40001 depending on the setting of the F_1 switch on the Supervisory Control Panel). Normally the address w at F_1 is referred to at the end of the repeated executions to provide the next program instruction; in some cases, however, the repeated executions are terminated differently.

During the execution of the Repeat instruction the contents of PAK are replaced by jn, PAK is complemented, and then advanced by one. Then, the j portion of PAK (now complemented) is sent to RSC wherein the method of advancing UAK and VAK is determined. The n portion of PAK contains the complement of the original n plus one. It may be seen that (if PAK is advanced after each execution of the repeated instruction) after n executions the lower-order 12 stages of PAK will contain all zeros and a carry from PAK_{11} will be generated; it is this carry (the END REPEAT signal) that announces to RSC that the required number of executions has been completed and that a repeat termination is in order. Also, during the execution of the Repeat instruction, the 15-bit address w is stored at F_1. This is not used until the repeated executions are completed

PX 71209

at which time the contents of F_1 normally are taken as the next instruction (F_1 usually contains a Manually Selective Jump instruction, 45jv, in which j is zero. The v-address portion of 45jv is replaced by w during the execution of the Repeat instruction).

THE REPEATED INSTRUCTION. - Depending on the selection of the instruction to be repeated, a variety of results can be obtained. These can be divided into four cases:

CASE 1. If the Interpret (14--), Return Jump (37uv), Q-Jump (44uv), Sign Jump (46uv), Zero Jump (47uv), Manually Selective Stop (56jv), or Final Stop (57--) instruction is chosen, the repeat sequence is automatically terminated since either RSC is cleared or the clock is stopped at the end of the first execution. Thus, these instructions behave as if no Repeat instruction preceded them. The termination is by the Instruction Reference Commands on page B-9.

CASE 2. If either the Index Jump (41uv) or the Manually Selective Jump (45jv) is selected, the instruction will be executed n times unless a jump is called for. If a jump is encountered, RSC is cleared and the jump is executed (no count of the number of times the instruction was executed is retained). If no jump is encountered during the n executions, a Normal Termination of the repeat sequence is executed and the next instruction is taken from F_1.

CASE 3. If either the Threshold Jump (42uv) or the Equality Jump (43uv) is selected, the instruction can be repeated or not repeated depending on whether or not a jump is called for. If a jump is executed before n executions or on the nth execution, the repeat sequence is terminated by a special sequence called the Jump Termination. This sequence stores the quantity j(n-r) in the Q-Register and then performs the jump as prescribed. The contents of Q may then be referred to in the determination of the number of executions actually performed (by subtracting the n-r in Q from the original n). If n executions are performed and no jump condition is satisfied, the repeat sequence is terminated by the Normal Termination and the next instruction is taken from F_1.

CASE 4. All other instructions, not referred to in the three cases above, will be executed n times as specified. After n executions, the repeat sequence is terminated by the Normal Termination and the next instruction is taken from F_1.

CHANGED COMMANDS. - During each of the n executions of any repeated instruction (unless RSC has been cleared or a jump has been called for) some additional commands are produced on MP5, MP6 is omitted, and MP7 is drastically changed. The following short table shows these changes:

PX 71209

MP	COMMAND	SOURCE	DEST.
5	Initiate End Test	CTC-RSC	RSC
	Advance PAK	CTC-PAK	PAK
	Wait RSC	CTC-PDC	PDC
7	Clear SAR	CTC-SAR	SAR

The command INITIATE END TEST is used in RSC to initiate an RSC End Test (No Jump) Sequence as shown on page B-103. If n-r is not zero, MPD is set to 7 and UAK and VAK are advanced according to the value of j. On MP7, SAR is cleared in preparation for the succeeding execution of the repeated instruction. If n-r is zero, the Normal Termination is executed using both MP6 and MP7. If RSC has been cleared prior to MP5, the commands above are not issued but the normal Instruction Reference Commands conclude the execution, and the next instruction is taken from the address specified in VAK. If during the repeated execution of a Threshold Jump instruction or an Equality Jump instruction a jump is called for, MP5 produces the commands which initiate the End Test and advance PAK; however, the actual End Test is superseded by the Jump Termination sequence which is also initiated on MP5.

REPEAT TERMINATIONS. - As indicated above, all repeated sequences are not terminated alike. There are three possible terminations; these are discussed in the subparagraphs below:

NORMAL TERMINATION. - At the occurrence of MP5, an End Test is made during each repeated execution. This test determines whether or not the nth execution has just been concluded. If n executions have been completed, the Normal Termination sequence follows MP5 as shown in the table below:

	COMMAND	SOURCE	DEST.
MP6	Set SAR to Fixed Address F_1	CTC-SAR	SAR
MP6	Clear PCR	CTC-PCR	PCR
MP6	Clear RSC	CTC-RSC	RSC
OP6	Initiate Read	CTC-SCC	SCC
	Clear X	CTC-AR	X
	Wait Internal Reference	CTC-PDC	PDC
MP7	Clear SAR	CTC-SAR	SAR
MP7	Transmit (X) to PCR (Initiate 2us delay)	CTC-AR	X/CRC

PX 71209

This sequence sets SAR to the Fixed Address F_1 (00000 or 40001) wherein the address w replaced the v-address portion of the former contents. Normally, F_1 contains a 45jv (Manually Selective Jump) instruction (j is zero) which will cause a jump to the address specified by the w of the Repeat instruction, and the program will proceed from the instruction stored at address w (it should be noted that any jump instruction can be used provided that a jump does occur; if a jump does not occur, the contents of PAK will be unaltered giving rise to a subsequent erroneous program continuation).

JUMP TERMINATION. - The Jump Termination is used to conclude the Threshold Jump (42uv) and the Equality Jump (43uv) only if the threshold or equality conditions call for a jump operation before or just after n executions. The Jump Termination sequence is initiated in RSC by an INITIATE JUMP TERMINATE command from CTC. This command is produced by MP5 if the jump condition is satisfied; the INITIATE END TEST and the ADVANCE PAK commands are also produced on MP5 but are disregarded if the INITIATE JUMP TERMINATE command is produced. The repeat sequence is modified thereby as shown in the table, RSC END TEST (WITH JUMP) SEQUENCE, on page B-106. This sequence clears X and sets MPD to 1 in preparation for the Jump Termination sequence as shown in the table below:

MP	COMMAND	SOURCE	DEST.
1	Transmit SAR to X	CTC-SAR	SAR
	Clear SAR	CTC-SAR	SAR
	Clear PAK	CTC-PAK	PAK
	Clear Q	CTC-AR	Q
2	Transmit (X) to Q	CTC-AR	X
	Transmit VAK to SAR	CTC-PCR	VAK
3	- - - - -		
4	- - - - -		
5	Transmit SAR to PAK	CTC-SAR	SAR

This sequence places j(n-r) in Q (effected by the transmissions PAK to SAR, SAR to X, and (X) to Q; the transmission PAK to SAR occurs on MP5 of the last repeated execution of the instruction in which the threshold or equality condition was met). Then, the v-address of the repeated instruction is inserted in PAK (CLEAR PAK, TRANSMIT VAK TO SAR and TRANSMIT SAR TO PAK) from which the next instruction will be taken. The v-address of the repeated instruction may or may not be advanced depending on the selection of the j value in the Repeat instruction. The Jump Termination sequence is concluded by the normal Instruction Reference Commands as given on page B-9.

PX 71299

OTHER TERMINATIONS. - All stop or jump instructions (other than 42uv and 43uv) either stop the clock or clear RSC both of which terminate the repeat sequence immediately. These instructions are concluded by the Instruction Reference Commands on page B-9 as if no Repeat instruction preceded them.

ABNORMALITIES.

CASE 1. If the Fixed Address F_1 does not contain a jump instruction or if a jump is not effected in the execution of the instruction at F_1, the instruction following the instruction at F_1 will be taken from the unaltered PAK. The address therein will be the complement of the j of the Repeat instruction as modified by the PAK_{11} CARRY produced as n becomes zero. The PAK_{11} CARRY goes to PAK_{12} and PAK_{13} as well as to RSC as the END REPEAT signal. However, the carry to PAK_{14} is automatically blocked because of the nature of PAK, and, therefore, its contents will be unaltered by a carry from PAK_{13}. As a result, the following addresses will be produced for the various normal values of j:

j = 0, address will be 40000
j = 1, address will be 70000
j = 2, address will be 60000
j = 3, address will be 50000

CASE 2. Values of j in excess of 3, i.e., j equals 4, 5, 6, or 7, will set up an unterminated repeat sequence. In each instance, PAK_{14} will contain a zero after PAK is complemented. A zero in PAK_{14} blocks the carry from PAK_9 to PAK_{10} thus causing a closed loop in PAK_0 through PAK_9. Because this carry is blocked, there will be no carry from PAK_{11} which is the END REPEAT signal to RSC, and thus the repeat sequence cannot be terminated unless a jump or a stop results from the repeated instruction. As in the case of j being 0, 1, 2, or 3, the values 4, 5, 6, or 7 will advance the u-address and v-address in a corresponding manner. If j is 4, neither will be advanced; if j is 5 only the v-address will be advanced; if j is 6 only the u-address will be advanced; if j is 7 both addresses will be advanced. These abnormal values of j have a practical application in the TEST mode of operation; in manual reading to or writing from the Q-Register such an unterminated repeat sequence is used advantageously. If the u- or v-address is in the MD range the address can be advanced through the complete range 40000 through 77777 and the next advance will return the address to 40000 since there is no carry from the 13th stage to the 14th stage in either UAK or VAK. If the u- or v-address is in the rapid access storage range (00000 through 01777) or the Q-address or A-address, the address will be advanced only through 1024 addresses and then start over. This is due to the blocking of the carry between the 9th and 10th stages if the 14th stage contains a zero.

CASE 3. If a Repeat instruction is followed immediately by another Repeat instruction, the second will supersede the first and the address of the repeated instruction is determined by the address in PAK. This address is the complement of j(n-1) which remains in PAK as a residue of the first Repeat instruction.

PX 1209

OPERATION CODE: 76

Instruction: EXTERNAL READ (ERjv)

If j = 0, replace the right-hand 8-bits of (v) with (IOA); if j = 1, replace (v) with (IOB). If the external unit utilizes step-by-step operation, advance one step.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
1	Test Input-Output Lockouts	MPD	PDC
3	Transmit IOA (or IOB) to X	CTC-IO	IOA(IOB)
	Clear IOA (or IOB)	CTC-IO	IOA(IOB)
	Transmit VAK to SAR	CTC-PCR	VAK
	Initiate Write	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC
	Initiate Lockout IOA (or IOB) Read	CTC-IO	PDC

PX 71209

OPERATION CODE: 77

Instruction: EXTERNAL WRITE (EWjv)

If j = 0, replace (IOA) with the right-hand 8 bits of (v); if j = 1, replace (IOB) with (v). Cause the previously selected unit to respond to the information in IOA or IOB.

MP	COMMAND	SOURCE	DEST.
0	Clear X	CTC-AR	X
	Transmit VAK to SAP	CTC-PCR	VAK
	Initiate Read	CTC-SCC	SCC
	Wait Internal Reference	CTC-PDC	PDC
1	Test Input-Output Lockouts	MPD	PDC
5	Transmit (X) to IOA (or IOB)	CTC-IO	IOA(IOB)
	Set IOA Write FF (or IOB Write FF)	CTC-IO	IOA(IOB)
	Initiate Lockout IOA (or IOB) Write	CTC-IO	PDC

PX 71209

SUBCOMMAND TIMING SEQUENCES

The Subcommand Timing Sequences presented on the following pages generate minor sequences which complete major sequences generated by the Command Timing Circuits. Each Subcommand Timing Sequence is initiated either by a command from CTC or by a subcommand from another of the Subcommand Timing Sequences. As an example, the following is given: In the instruction Transmit Positive (11uv) on page B-10, the command INITIATE READ is produced on MPO. In the DEST. column, SCC is given as the destination. On page B-62 the SCC Initiate Read Sequences are given. According to the storage class represented by the u-address, one of the five sequences listed is selected. Assuming that an MD address is represented, the subcommands INITIATE READ MD and INITIATE MD REFERENCE have MDAC as a destination. The proper MDAC table is given on page B-67 wherein the actual MD reading sequence is accomplished. The MD RESUME signal, produced on CP-0 when an address coincidence is detected, goes to PDC allowing MP5 to be issued to CTC thus negating the WAIT INTERNAL REFERENCE produced on MPO as a result of the INITIATE READ signal. It may be seen in the foregoing that two Subcommand Timing Sequences were generated; one as a result of a CTC command and one as a result of an SCC subcommand.

It should be noted in using the following tables that the first table entries are produced as a result of and at the same time as the initiating command or subcommand. The succeeding entries are produced either by CONTROLLED CLOCK PULSES or by pulses from several of the minor pulse distributors. These statements apply especially to the ARAC and ASC subsequences wherein no clock pulse columns are given. In all cases, horizontal lines separate the timing periods.

SCC INITIATE READ SEQUENCES

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Read (ES Address)		Set Initiate Read to "1"	
	CCP	"1" From Initiate Read Initiate Read ES Clear Initiate Read to "0"	FAULT ESAC

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Read (MD Address)		Set Initiate Read to "1"	
	CCP	"1" From Initiate Read Initiate Read MD Initiate MD Reference Clear Initiate Read to "0"	FAULT MDAC MDAC

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Read (Q Address)		Set Initiate Read to "1"	
	CCP	"1" From Initiate Read Initiate Read Q Clear Initiate Read to "0"	FAULT ARAC

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Read (A Address)		Set Initiate Read to "1"	
	CCP	"1" From Initiate Read Initiate Read A Clear Initiate Read to "0"	FAULT ARAC

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Read (Unassigned Address)		Set Initiate Read to "1"	
	CCP	"1" From Initiate Read (Initiates SCC Fault) Clear Initiate Read to "0"	FAULT

PX 71209

SCC INITIATE WRITE (0-35) SEQUENCES

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Write (0-35) → (ES Address)		Set Initiate Write to "1"	
	CCP	"1" From Initiate Write Initiate Write ES Clear Initiate Write to "0"	FAULT ESAC
Initiate Write (0-35) → (MD Address)		Set Initiate Write to "1"	
	CCP	"1" From Initiate Write Initiate Write MD Initiate MD Reference Clear Initiate Write to "0"	FAULT MDAC MDAC
Initiate Write (0-35) → (Q Address)		Set Initiate Write to "1"	
	CCP	"1" From Initiate Write Initiate Write Q Clear Initiate Write to "0"	FAULT ARAC
Initiate Write (0-35) → (A Address)		Set Initiate Write to "1"	
	CCP	"1" From Initiate Write Initiate Write A Clear Initiate Write to "0"	FAULT ARAC
Initiate Write (0-35) → (Unassigned Address)		Set Initiate Write to "1"	
	CCP	"1" From Initiate Write (Initiates SCC Fault) Clear Initiate Write to "0"	FAULT

PX 71209

SCC INITIATE WRITE (0-14) SEQUENCES

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Write (0-14) →		Set IW (0-14) to "1"	
(ES Address)	CCP	"1" From IW (0-14) Initiate Write ES (0-14) Clear IW (0-14) to "0"	FAULT ESAC

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Write (0-14) →		Set IW (0-14) to "1"	
(MD Address)	CCP	"1" From IW (0-14) Initiate Write MD (0-14) Initiate MD Reference Clear IW (0-14) to "0"	FAULT MDAC MDAC

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Write (0-14) →		Set IW (0-14) to "1"	
(All Other Addresses)	CCP	"1" From IW (0-14) (Initiates SCC Fault) Clear IW (0-14) to "0"	FAULT

PX 71209

SCC INITIATE WRITE (15-29) SEQUENCES

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Write (15-29) →		Set IW (15-29) to "1"	
(ES Address)	CCP	"1" From IW (15-29) Initiate Write ES (15-29) Clear IW (15-29) to "0"	FAULT ESAC

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Write (15-29) →		Set IW (15-29) to "1"	
(MD Address)	CCP	"1" From IW (15-29) Initiate Write MD (15-29) Initiate MD Reference Clear IW (15-29)	FAULT MDAC MDAC

CTC COMMAND		SCC SUBCOMMAND	DESTINATION
Initiate Write (15-29) →		Set IW (15-29) to "1"	
(All Other Addresses)	CCP	"1" From IW (15-29) (Initiates SCC Fault) Clear IW (15-29)	FAULT

PX 71209

MDAC LOCATING SEQUENCE

Mark Pulse	Set Preset FF
CP-0	Preset AIK to 1's Set Coincidence Lockout FF Clear Preset FF Clear Advance AIK FF
CP-1	*
CP-3	

CP-0	Advance AIK AIK End Carry Clear Coincidence Lockout FF Clear Advance AIK FF
CP-1	*
CP-3	

CP-0	Advance AIK Clear Advance AIK FF
CP-1	*
CP-3	

AIK ≠ 4095 AIK = 4095

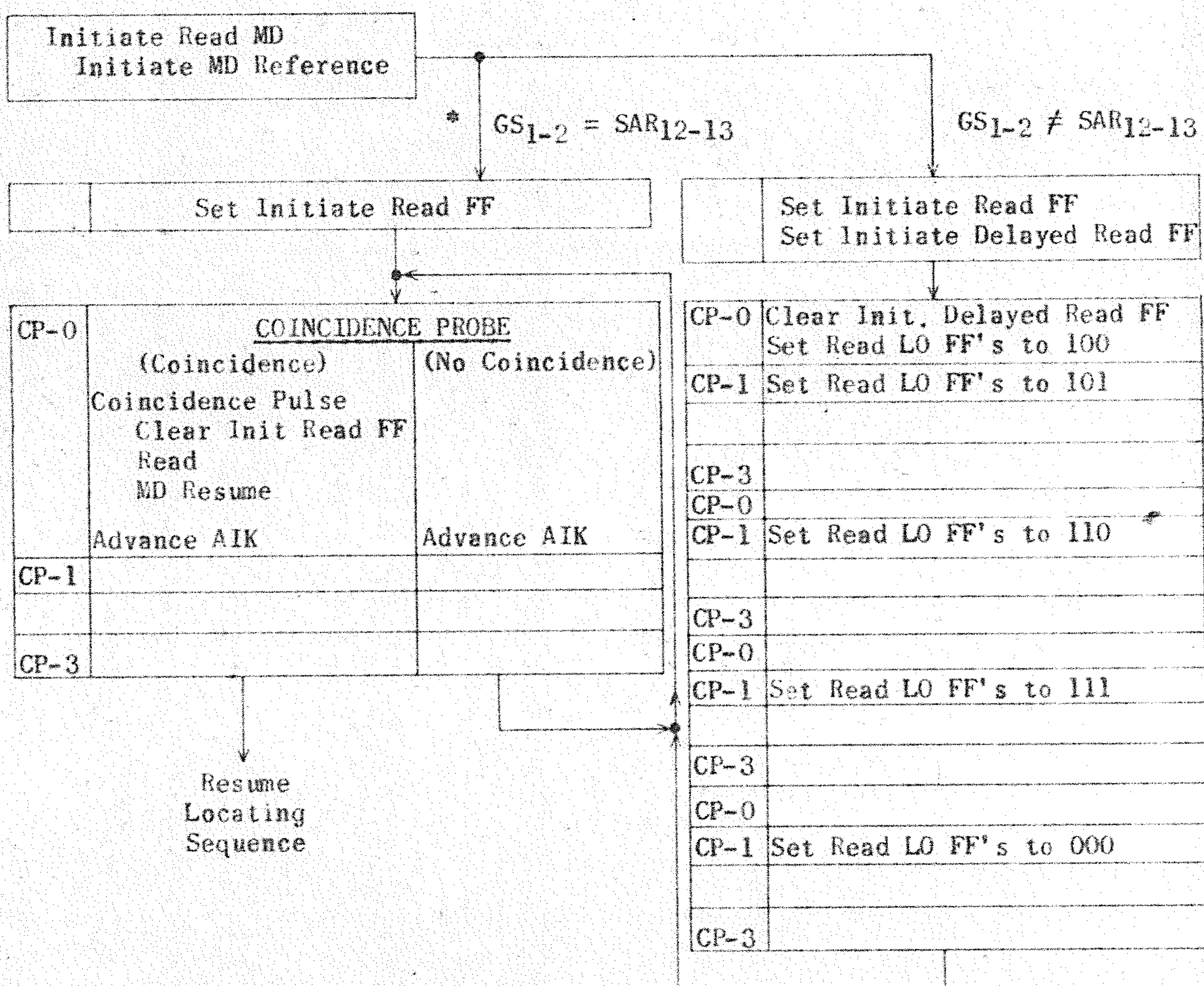
CP-0	Advance AIK AIK End Carry Clear Coincidence Lockout FF Clear Advance AIK FF
CP-1	*
CP-3	

* If the Read Lockout FF's are not all set to zero, CP1 will advance these FF's until their count is 000.

PX 71309

MDAC READ SEQUENCE

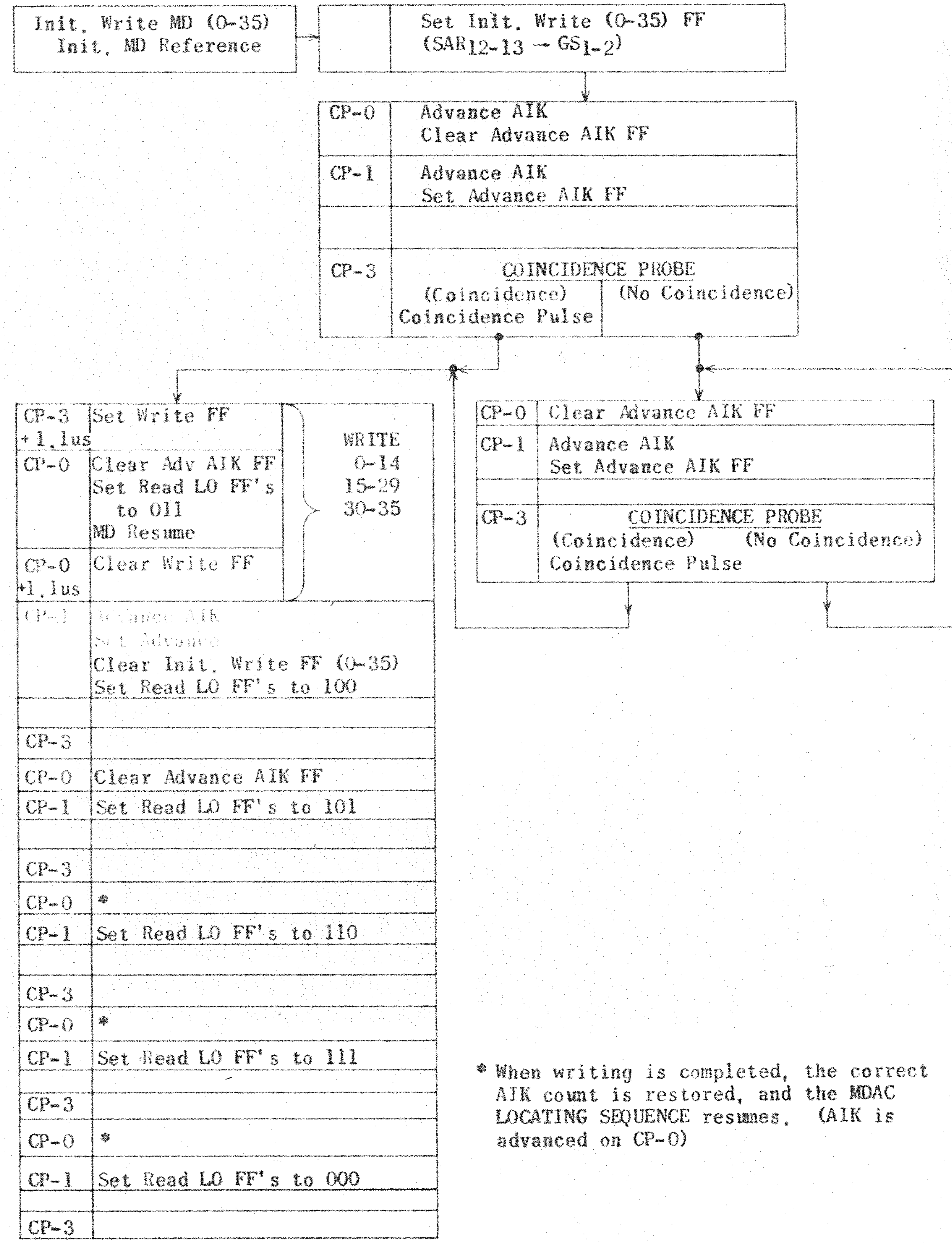
SCC SUBCOMMAND



* It should be noted that even though $GS_{1-2} = SAR_{12-13}$, if a read operation follows a write operation before approximately 40 usec have elapsed, all or a portion of the Read LO FF counting delay ($GS_{1-2} \neq SAR_{12-13}$) will be carried out until each Read LO FF is in its "0" state.

PX 71209

MDAC WRITE (0-35) SEQUENCE



* When writing is completed, the correct AIK count is restored, and the MDAC LOCATING SEQUENCE resumes. (AIK is advanced on CP-0)

PX 71209

MDAC WRITE (0-14) SEQUENCE

SCC SUBCOMMAND

SCC Subcommand		
Init. Write MD (0-14) Init. MD Reference	→	Set Init. Write (0-14) FF $(SAR_{12-13} \rightarrow GS_{1-2})$

CP-0	Advance AIK Clear Advance AIK FF
CP-1	Advance AIK Set Advance AIK FF
CP-3	COINCIDENCE PULSE (Coincidence) COINCIDENCE PULSE | (No Coincidence)

(Coincidence):

CP-3 +1.1us	Set Write FF	Write (0-14)
CP-0	Clear Advance AIK FF Set Read LO FF's to 011 MD Resume	
CP-0 +1.1us	Clear Write FF	
CP-1	Clear Init. Write(0-14) FF Set Read LO FF's to 100 Advance AIK Set Advance AIK FF	
CP-3		
CP-0	Clear Advance AIK FF	
CP-1	Set Read LO FF's to 101	
CP-3		
CP-0	*	
CP-1	Set Read LO FF's to 110	
CP-3		
CP-0	*	
CP-1	Set Read LO FF's to 111	
CP-3		
CP-0	*	
CP-1	Set Read LO FF's to 000	
CP-3		

(No Coincidence):

CP-0	Clear Advance AIK FF
CP-1	Advance AIK Set Advance AIK FF
CP-3	COINCIDENCE PULSE (Coincidence) COINCIDENCE PULSE | (No Coincidence)

* When writing is completed, the correct AIK count is restored, and the MDAC LOCATING SEQUENCE resumes, (AIK is advanced by CP-0)

PX 71209

MDAC WRITE (15-29) SEQUENCE

SCC SUBCOMMAND

Init. Write MD (15-29)
Init. MD Reference

→

	Set Init. Write (15-29) FF ($SAR_{12-13} \rightarrow GS_{1-2}$)
CP-0	Advance AIK Clear Advance AIK FF
CP-1	Advance AIK Set Advance AIK FF
CP-3	COINCIDENCE PROBE (Coincidence) COINCIDENCE PULSE | (No Coincidence)

(Coincidence) →

CP-3 +1.1us	Set Write FF	Write (15-29)
CP-0	Set Read LO FF's to 011 MD Resume	
CP-0 +1.1us	Clear Write FF	
CP-1	Clear Init. Write (15-29) FF Set Read LO FF's to 100 Advance AIK Set Advance AIK FF	
CP-3		
CP-0	Clear Advance AIK FF	
CP-1	Set Read LO FF's to 101	
CP-3		
CP-0	*	
CP-1	Set Read LO FF's to 110	
CP-3		
CP-0	*	
CP-1	Set Read LO FF's to 111	
CP-3		
CP-0	*	
CP-1	Set Read LO FF's to 000	
CP-3		

(No Coincidence) →

CP-0	Clear Adv. AIK FF
CP-1	Advance AIK Set Adv. AIK FF
CP-3	COINCIDENCE PROBE (Coincidence) COINCIDENCE PULSE | (No Coincidence)

(From this second COINCIDENCE PROBE, Coincidence goes to the Set Write FF step and No Coincidence goes back to CP-0 Clear Adv. AIK FF.)

* When writing is completed, the correct AIK count is restored, and the MDAC LOCATING SEQUENCE resumes. (AIK is advanced by CP-0)

PX 71209

ESAC REGENERATE SEQUENCE

SCC SUBCOMMAND	EP	ESAC SUBCOMMAND
(none) →	EP-0	Clear DD I FF Clear DD II FF Set Probe II FF Clear Dash I FF Clear ESO stages RK → DR Advance RK
	EP-2*	Set Probe I FF, begin Probe Set Dash II FF Initiate 1.1 us delay
	EP-2 +1.1 us	Restore(0-14) } Restore (15-29) } (ES → ESO) Restore (30-35) } Clear Probe II FF, end Probe Set DD I FF } Set DD II FF } begin Dash (ESO → ES) Set Dash I FF } Initiate 1.1 us delay
	EP-3	Clear Probe I FF
	EP-2 +2.2 us	Clear Dash II FF, end Dash

* Note: An extra clock-pulse period occurs between EP-0 and EP-2.

PX 71209

ESAC READ SEQUENCE

SCC SUBCOMMAND	EP	ESAC SUBCOMMAND
Init Read ES →		Set RD Sync FF Set Address Source FF
	EP-0	Clear DDI FF Clear DD II FF Set Probe II FF Clear Dash I FF Clear ESO stages SAR → DR Clear Address Source FF Clear RD Sync FF Set Read FF
	EP-2*	Set Probe 1 FF, begin Probe Set Dash II FF Initiate 1.1 us delay
	EP-2 +1.1 us	Restore(0-14) } Restore (15-29) } (ES → ESO) Restore (30-35) } Clear Probe II FF, end Probe Set DD I FF } Set DD II FF } begin Dash (ESO → ES) Set Dash I FF } Initiate 1.1 us delay
	EP-3	Read (ESO → X) Clear Probe I FF Clear Read FF ES Resume (to PDC)
	EP-2 +2.2 us	Clear Dash II FF, end Dash

*Note: An extra clock-pulse period occurs between EP-0 and EP-2.

PX 71209

ESAC WRITE (0-35) SEQUENCE

SCC SUBCOMMAND	EP	ESAC SUBCOMMAND
Init Write ES(0-35) →		Set WR Sync (0-35) FF Set Address Source FF
	EP-0	Clear DD I FF Clear DD II FF Set Probe II FF Clear Dash I FF Clear ESO stages SAR → DR Clear Address Source FF Set WR/REST (0-14) FF Set WR/REST (15-29) FF Set WR/REST (30-35) FF Clear WR Sync (0-35) FF
	EP-2*	Set Probe I FF, begin Probe Set Dash II FF Initiate 1.1 us delay
	EP-2 +1.1 us	Write (0-14), Write (15-29), Write (30-35) } (X → ESO) Clear Probe II FF, end Probe Set DD I FF, Set DD II FF, Set Dash I FF } begin Dash (ESO → ES) Initiate 1.1 us delay
	EP-3	Clear WR/REST (0-14) FF Clear WR/REST (15-29) FF Clear WR/REST (30-35) FF Clear Probe I FF ES Resume (to PDC)
	EP-2 +2.2 us	Clear Dash II FF, end Dash

* Note: An extra clock-pulse period occurs between EP-0 and EP-2.

PX 71209

ESAC WRITE (0-14) SEQUENCE

SCC SUBCOMMAND	EP	ESAC SUBCOMMAND
Init Write ES (0-14) →		Set WR Sync (0-14) FF Set Address Source FF
	EP-0	Clear DD I FF Clear DD II FF Set Probe II FF Clear Dash I FF Clear ESO stages SAR → DR Clear Address Source FF Set WR/REST (0-14) FF Clear WR Sync (0-14) FF
	EP-2 *	Set Probe I FF, begin Probe Set Dash II FF Initiate 1.1 us delay
	EP-2 +1.1 us	Write (0-14) (X → ESO 0-14) Restore (15-29) } (ES → ESO 15-35) Restore (30-35) Clear Probe II FF, end Probe Set DD I FF } Set DD II FF } begin Dash (ESO → ES) Set Dash I FF } Initiate 1.1 us delay
	EP-3	Clear WR/REST (0-14) FF Clear Probe I FF ES Resume (to PDC)
	EP-2 +2.2 us	Clear Dash II FF, end Dash

* Note: An extra clock-pulse period occurs between EP-0 and EP-2.

PX 71209

ESAC WRITE (15-29) SEQUENCE

SCC SUBCOMMAND	EP	ESAC SUBCOMMAND
Init Write (15-29) →		Set WR Sync (15-29) FF Set Address Source FF
	EP-0	Clear DD I FF Clear DD II FF Set Probe II FF Clear Dash I FF Clear ESO stages SAR → DR Clear Address Source FF Set WR/REST (15-29) FF Clear WR Sync (15-29) FF
	EP-2 *	Set Probe I FF, begin Probe Set Dash II FF Initiate 1.1 us delay
	EP-2 +1.1 us	Write (15-29) } (X → ESO 15-29) Restore (0-14), Restore (30-35) } (ES → ESO 0-14, 30-35) Clear Probe II FF, end Probe Set DD I FF, Set DD II FF, Set Dash I FF } begin Dash (ESO → ES) Initiate 1.1 us delay
	EP-3	Clear WR/REST (15-29) FF Clear Probe I FF ES Resume (to PDC)
	EP-2 +2.2 us	Clear Dash II FF, end Dash

* Note: An extra clock-pulse period occurs between EP-0 and EP-2.

PX 71209

ARAC READ SEQUENCE

INSTRUCTION 27

Q REFERENCE

CTC COMMAND - INITIATE CLEAR A

SCC SUBCOMMAND-
INITIATE READ Q

SCC SUBCOMMAND - INITIATE READ Q	CTC COMMAND - INITIATE CLEAR A
	Set CLEAR A FF
COMPLEMENT X Set READ Q FF	CLEAR A_R Clear CLEAR A FF
$Q' \rightarrow X'$ Clear READ Q FF ARAC RESUME Clear SAR	---- ---- ---- ----

A REFERENCE

CTC COMMAND - INITIATE CLEAR A

SCC SUBCOMMAND-
INITIATE READ A

SCC SUBCOMMAND - INITIATE READ A	CTC COMMAND - INITIATE CLEAR A
	Set CLEAR A FF
$A_R \rightarrow X$ ARAC RESUME Clear SAR	Clear CLEAR A FF Clear A_R

PX 71209

ARAC READ SEQUENCE

INSTRUCTION 41, 54, OR 74

Q REFERENCE

CTC COMMAND - INITIATE CLEAR A

SCC SUBCOMMAND-
INITIATE READ Q

	Set CLEAR A FF
COMPLEMENT X Set READ Q FF	CLEAR A_R CLEAR A_L Clear CLEAR A FF
Q' → X' Clear READ Q FF ARAC RESUME Clear SAR	- - - - - - - - - - - - - - - - - - - -

A REFERENCE

CTC COMMAND - INITIATE CLEAR A

SCC SUBCOMMAND-
INITIATE READ A

	Set CLEAR A FF
ARAC RESUME Clear SAR	Clear CLEAR A FF

PX 71209

ARAC READ SEQUENCE

INSTRUCTION 21, 23, 31, 33, 51, 53, OR 71

Q REFERENCE

CTC COMMAND - INITIATE CLEAR A

SCC SUBCOMMAND-
INITIATE READ Q

	Set CLEAR A FF
COMPLEMENT X Set READ Q FF	CLEAR A_R CLEAR A_L Clear CLEAR A FF
$Q' \rightarrow X'$ Clear READ Q FF ARAC RESUME Clear SAR	---- ---- ---- ----

A REFERENCE

CTC COMMAND - INITIATE CLEAR A

SCC SUBCOMMAND-
INITIATE READ A

	Set CLEAR A FF
$A_R \rightarrow X$ ARAC RESUME Clear SAR	CLEAR A_R CLEAR A_L Clear CLEAR A FF

PX 71200

ARAC READ SEQUENCE

ALL OTHER INSTRUCTIONS

SCC COMMAND - INITIATE READ Q OR INITIATE READ A

Q REF.	A REF.
COMPLEMENT X Set READ Q FF	$A_R \rightarrow X$ ARAC RESUME Clear SAR
$Q' \rightarrow A'$ Clear READ Q FF ARAC RESUME Clear SAR	---- ---- ---- ----

ARAC WRITE SEQUENCE

INSTRUCTION 11, 12, 13, 55, 73, OR 76

SCC COMMAND - INITIATE WRITE Q OR INITIATE WRITE A

Q REF.	A REF.
CLEAR Q Set WRITE A OR Q FF	Clear A_R and A_L Set WRITE A OR Q FF
X ⟶ Q Clear WRITE A OR Q FF ARAC RESUME Clear SAR	ADD X TO A * Clear WRITE A OR Q FF Clear SAR

* Note: The ADD X TO A signal to ASC eventually generates an ASC RESUME rather than an ARAC RESUME.

PX 71209

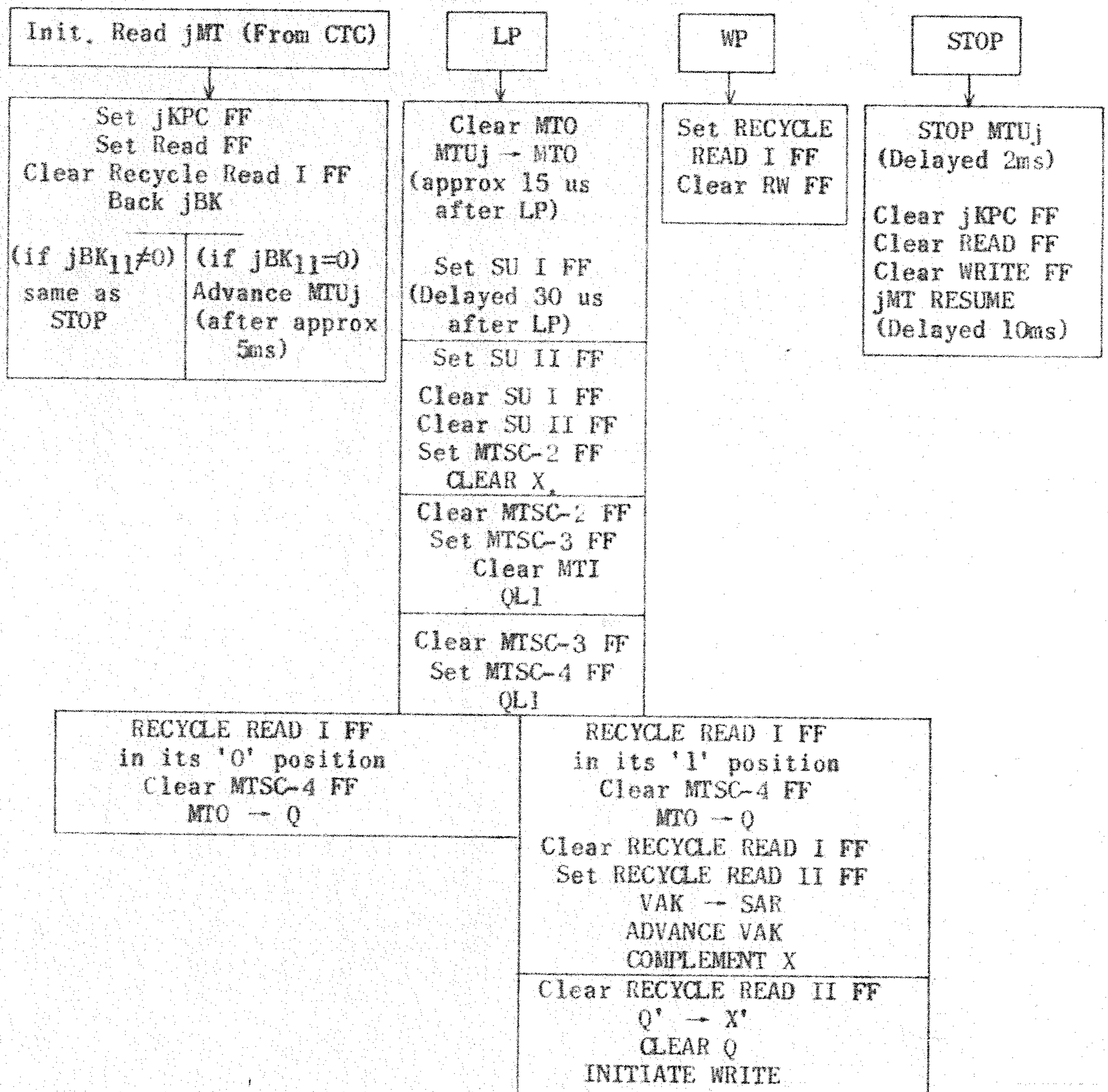
ARAC WRITE SEQUENCE

ALL OTHER INSTRUCTIONS

SCC COMMAND - INITIATE WRITE Q OR INITIATE WRITE A

Q REF.	A REF.
CLEAR Q Set WRITE A OR Q FF	ARAC RESUME Clear SAR
$X \rightarrow Q$ Clear WRITE A OR Q FF ARAC RESUME Clear SAR	---- ---- ---- ----

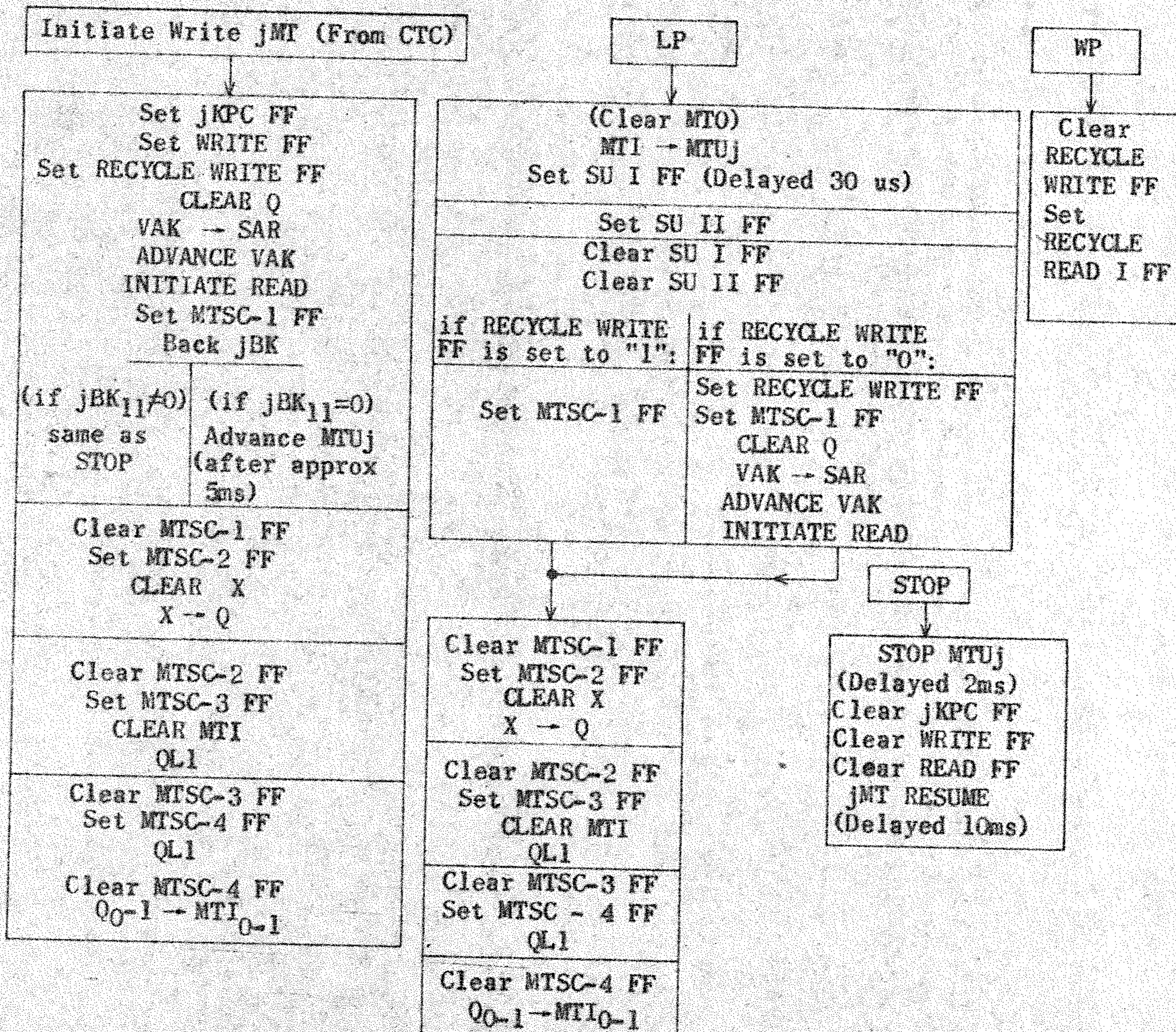
MT READING SEQUENCE



Note: Approximately 14ms after the pulse INITIATE READ jMT is received from CTC, a line pulse (LP) is generated by MTUj. Approximately 15us after the LP is produced, the transmission MTUj → MTO occurs. MTUj generates successive LP's at the rate of 4 1/2 kc. Along with each 18th LP, LK generates a word pulse (WP). Along with each 32nd WP, WK generates a block pulse (BP) which subtracts one count from the preset block count (BK). When BK = 0, the BP becomes a STOP pulse.

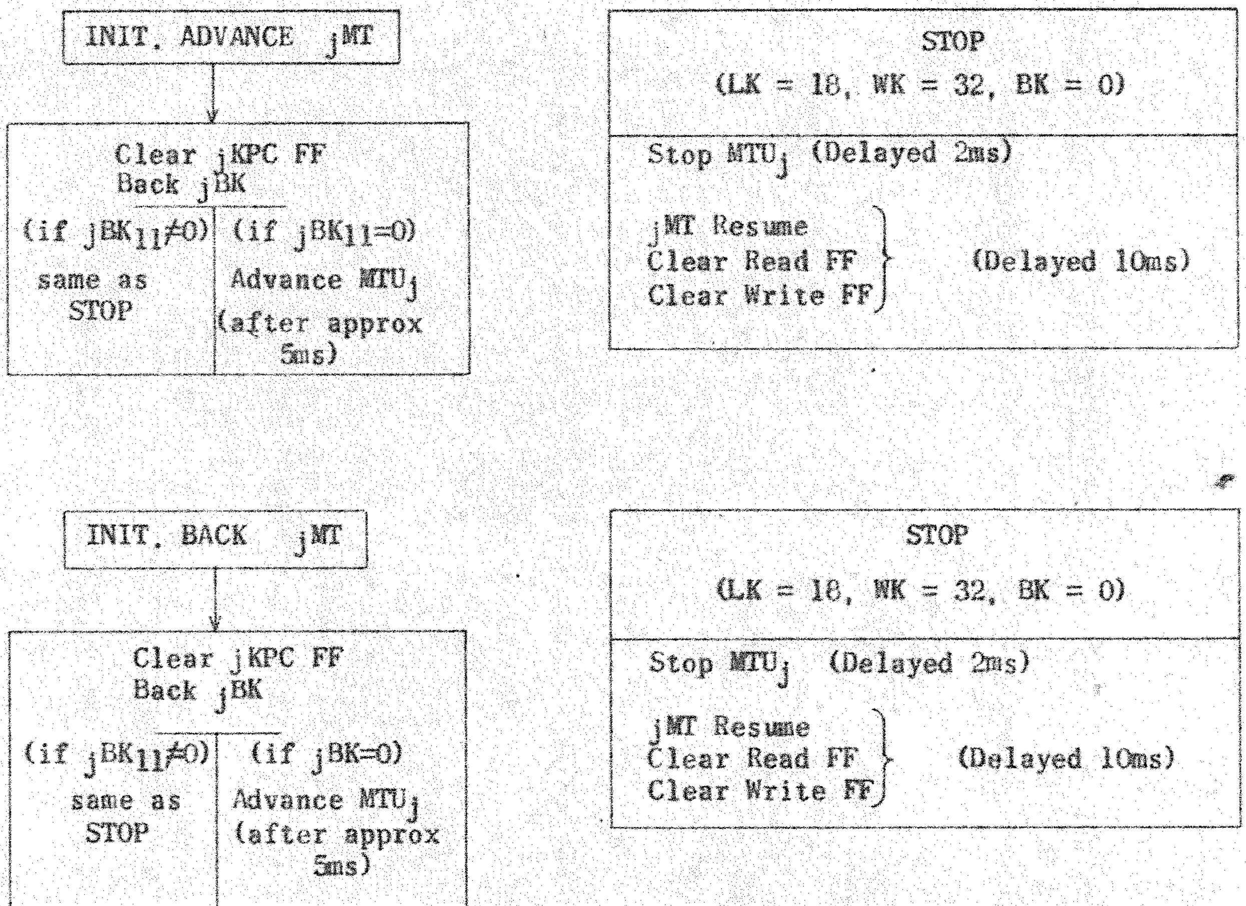
PX 71209

MT WRITING SEQUENCE



Note: Approximately 14ms after the pulse INITIATE WRITE jMT is received from CTC, a line pulse (LP) is generated by MTUj. MTUj generates successive LP's at the rate of 4 1/2 kc. Along with each 18th LP, LK generates a word pulse (WP). Along with each 32nd WP, WK generates a block pulse (BP) which subtracts one count from the preset block count (BK). When BK=0, the BP becomes a STOP pulse. (As before each gated LP clears MTO, however this is of no significance in the MT Writing operation.)

MT POSITIONING SEQUENCES



PX 71209

TYPEWRITER SEQUENCES

CTC COMMAND	EVENT	RESULTS
$X \rightarrow$ TWR & Initiate Print $\rightarrow$		$(X_0 - X_5) \rightarrow$ TWR Energizes: Appropriate TWR relays (K30064-K30069), and Reader Clutch (L_{RC}).
	TWR relays energized	Energizes appropriate impossible Print and Function Translator relays (K26007-K26013). Closes contacts in circuits of appropriate Code Translator Solenoids (L_{T1} - L_{T6})
	Reader Clutch energized	Initiates cycle in typewriter
	Typewriter cycle initiated	Reader Common Contacts close
	Reader Common Contacts closed	Energizes: Enable Impossible Print relay (K26019), Translator Clutch (L_{TC}), Code Translator Solenoids, and Character Acknowledge relay (K26003)
	Enable Impossible Print, and Impossible Print and Function Translator relays energized	Translates code held in TWR and detects 1 of 3 possible conditions: 1. No circuit through translator-Character Sequence is followed. See a. below. 2. Circuit through translator energizes Function relay (K26014)-Function Sequence is followed. See b below. 3. Circuit through translator energizes Impossible Print relay (K26006)-Impossible Print Sequence is followed. See c. below. NOTE: See d. below for typewriter code. Any code not listed is invalid and results in an Impossible Print Sequence

PX 71209

TYPEWRITER SEQUENCES (Cont'd)

a. Character Sequence
(All operations except Impossible Print, Carriage Return, and Tab.)

EVENT	RESULTS
Translator Clutch and Code Translator Solenoids energized	Causes typewriter to perform the appropriate operation.
Character Acknowledge relay energized	De-energizes Acknowledge I relay (K26004)
Acknowledge I relay de-energized	De-energizes Acknowledge II relay. Momentarily removes B+ from TWR relays, and Reader Clutch, causing them to de-energize, and extinguishes TWR thyratrons thus clearing TWR.
TWR relays de-energized	De-energizes: Code Translator relays, and impossible Print and Function Translator relays.
Reader Clutch de-energized	Prevents initiation of another typewriter cycle until another print command is received from CTC.
Reader Common Contacts open	De-energize: Translator Clutch, Character Acknowledge relay, and Enable Impossible Print Relay
Character Acknowledge relay de-energized	Energize Acknowledge I relay.
Acknowledge I relay energized	Energize Acknowledge II relay. Fires thyratron to produce TWC RESUME (to PDC).

PX 71209

TYPEWRITER SEQUENCES (Cont'd)

b. Function Sequence
(Carriage Return and Tab only)

EVENT	RESULTS
Function relay energized	Closes hold-in contact through Carriage Return Contact. Opens contact in TWC RESUME circuit.
Translator Clutch and Code Translator Solenoids energized	Causes typewriter to perform the appropriate operation
Character Acknowledge relay energized	De-energizes Acknowledge I relay (K26004).
Acknowledge I relay de-energized	De-energizes Acknowledge II relay. Momentarily removes B+ from TWR relays, and Reader Clutch causing them to de-energize, and extinguishes TWR thyratrons thus clearing TWR.
TWR relays de-energized	De-energizes: Code Translator Solenoids and Impossible Print and Function Translator relays.
Reader Clutch de-energized	Prevents initiation of another typewriter cycle until another print command is received from CTC.
Carriage Return Contacts open	De-energizes Function Relay.
Function relay de-energized	Closes contact in TWC RESUME circuit
Reader Common Contacts open	De-energize: Translator Clutch, Character Acknowledge relay, and Enable Impossible Print relay
Character Acknowledge relay de-energized	Closes contact in circuit of Acknowledge I relay.
Carriage Return Contacts close	Energizes Acknowledge I relay.
Acknowledge I relay energized	Energizes Acknowledge II relay. Fires thyratron to produce TWC RESUME (to PDC).

PX 71209

TYPEWRITER SEQUENCES (Cont'd)

c. Impossible Print Sequence

EVENT	RESULTS
Impossible Print relay energized	Closes hold-in contact for itself. Closes hold-in contact for Acknowledge I relay (K26004) thus holding it energized and preventing a clearing of TWR and a TWC RESUME. De-energizes Reader Clutch. Energizes Impossible Print Fault relay (K30062 in Fault Detector).
Impossible Print Fault relay energized	Causes "A" Fault in Main Equipment.
Reader Clutch de-energized	Prevents initiation of another typewriter cycle until another print command is received from CTC. (In addition the Impossible Print relay (K26006) must be de-energized by clearing the fault.)

TYPEWRITER SEQUENCE (Concl.)

d. Typewriter Code

LETTERS

UPPER CASE	LOWER CASE	OCTAL CODE
A	a	30
B	b	23
C	c	16
D	d	22
E	e	20
F	f	26
G	g	13
H	h	05
I	i	14
J	j	32
K	k	36
L	l	11
M	m	07
N	n	06
O	o	03
P	p	15
Q	q	35
R	r	12
S	s	24
T	t	01
U	u	34
V	v	17
W	w	31
X	x	27
Y	y	25
Z	z	21

NUMBERS

UPPER CASE	LOWER CASE	OCTAL CODE
^{1}	1	52
^{2}	2	74
^{3}	3	70
^{4}	4	64
^{5}	5	62
^{6}	6	66
^{7}	7	72
^{8}	8	60
^{9}	9	33
^{0}	0	37

OPERATIONS

TYPEWRITER OPERATION	OCTAL CODE
SPACE	04
SHIFT UP	47
SHIFT DOWN	57
BACK SPACE	61
CAR. RET.	45
TAB.	51
COLOR SHIFT	02

SIGNS

UPPER CASE	LOWER CASE	OCTAL CODE
-	-	56
·	=	44
/	+	54
(	,	46
)	.	42
_	|	50

EXPLANATION OF SIGNS

UPPER CASE	LOWER CASE
Superscript Minus	Hyphen or Minus
Multiply	Equals
Virgule	Plus
Open Parens	Comma
Close Parens	Period
Underline	Absolute

HIGH-SPEED PUNCH SEQUENCE

CTC COMMAND

X → HPR &
Init. High-Speed →
PUNCH

Event	Results
	Sets Init. HSP FF. X_0 -X_5 and UAK_{12} → HPR.
Next closing of punch sync. contacts when Init. HSP is set to "1"	Fires thyratrons, energizing Tape Level and Tape Feed solenoids, operating punch. Clears Init. HSP FF. Sets HPC Resume FF. Clears HPR.
Sync. contacts open	Extinguishes Tape Level and Tape Feed Solenoid relay pullers. Produces HPC Resume (to PDC).
HPC Resume	Clears HPC Resume FF.

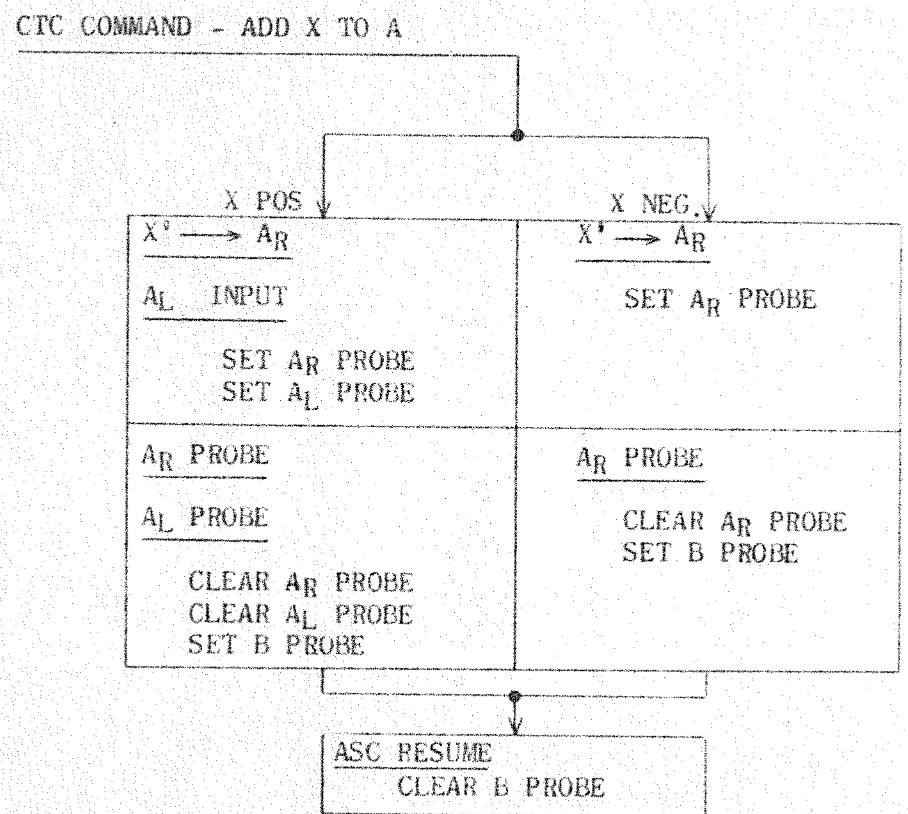
ARITHMETIC SEQUENCE CONTROL

TIMING SEQUENCES

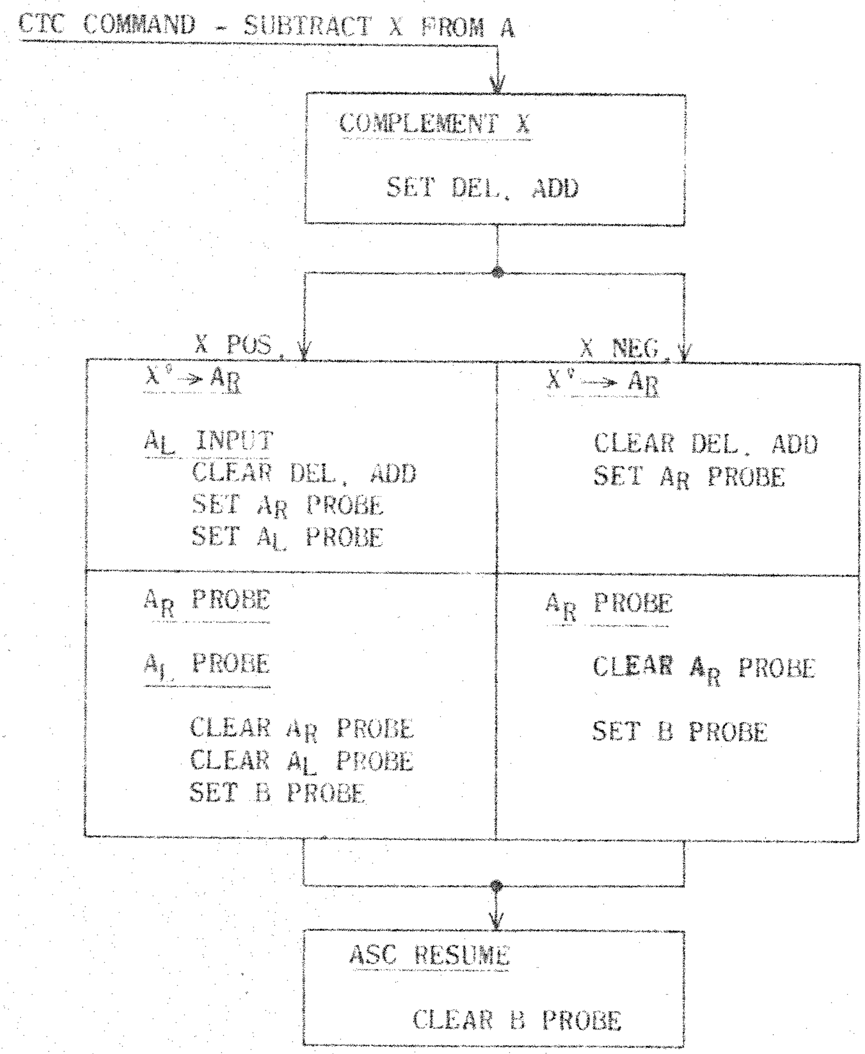
The following pages are slightly different in format from those previously encountered. In the following tabulations only signals are listed (their corresponding events are obvious in almost every case). In order to separate those subcommands which are issued from the ASC and those used solely within the ASC, the following format has been used. The subcommands set flush left and underlined are those which are issued by the ASC to different systems within the equipment. The subcommands which are indented and not underlined are used only within the ASC.

Those sequences involving subcommands which are generated in SKC and which are used to control some portion of the major ASC sequence, are called ASC/SKC sequences. An asterisk is used to identify SKC subcommands which effect some control wherever so doing facilitates an understanding of the ASC/SKC sequence.

ASC ADD X TO A SEQUENCE



ASC SUBTRACT X FROM A SEQUENCE



PX 71209

ASC SPLIT ADD X TO A SEQUENCE

CTC COMMAND - SPLIT ADD X TO A

$X^{9} \longrightarrow A_R$

A_L INPUT
- SET A_R PROBE
- SET A_L PROBE

A_R PROBE

A_L PROBE
- CLEAR A_R PROBE
- CLEAR A_L PROBE
- SET B PROBE

ASC RESUME
- CLEAR B PROBE

ASC SPLIT SUBTRACT X FROM A SEQUENCE

CTC COMMAND - SPLIT SUBTRACT X FROM A

COMPLEMENT X
- SET SPLIT SUBTRACT

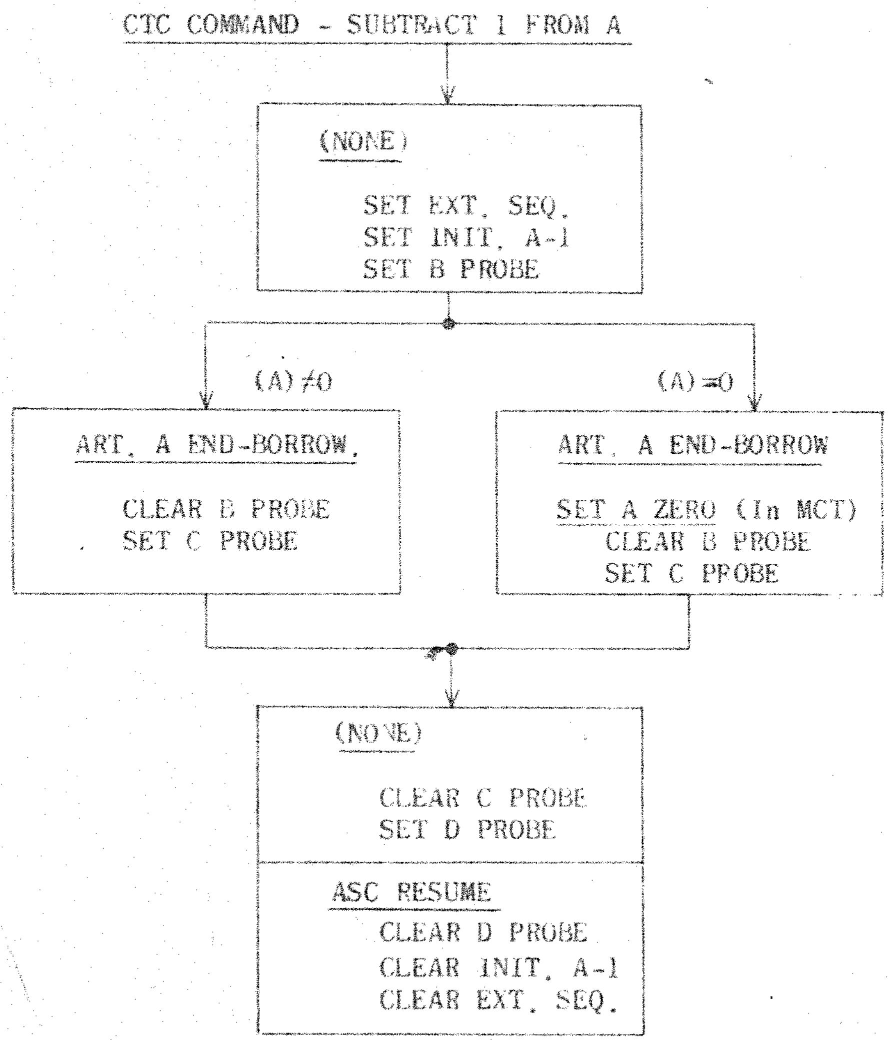
$X^{9} \longrightarrow A_R$
- CLEAR SPLIT SUBTRACT
- SET A_R PROBE

A_R PROBE
- CLEAR A_R PROBE
- SET B PROBE

ASC RESUME
- CLEAR B PROBE

PX 71209

ASC SUBTRACT 1 FROM A SEQUENCE



PX 71209

ASC LOGICAL SEQUENCE

CTC COMMANDS - INITIATE LOGICAL & EXT. ARITH. SEQUENCE

↓

$Q' \longrightarrow X'$
- SET EXT. SEQ.
- SET INIT. LOG.
- SET DEL. ADD

$X' \longrightarrow A_R$
A_L INPUT
- CLEAR DEL. ADD
- SET A_R PROBE
- SET A_L PROBE

A_R PROBE

A_L PROBE

CLEAR X
- CLEAR A_R PROBE
- CLEAR A_L PROBE
- SET REST. X
- SET B PROBE

COMPLEMENT X
- CLEAR B PROBE
- CLEAR REST. X
- SET C PROBE

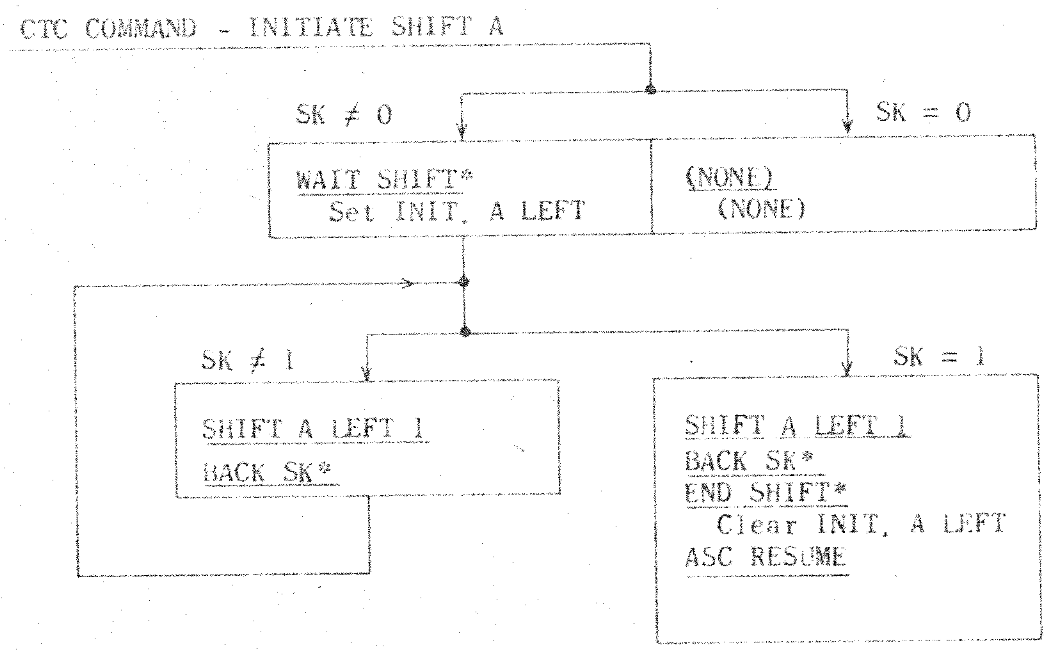
$Q' \longrightarrow X'$
CLEAR Q
- CLEAR C PROBE
- CLEAR EXT. SEQ.
- SET D PROBE

COMPLEMENT X
- CLEAR D PROBE
- SET E PROBE

$X \longrightarrow Q$
CLEAR X
ASC RESUME
- CLEAR E PROBE
- CLEAR INIT. LOG.

PX 71209

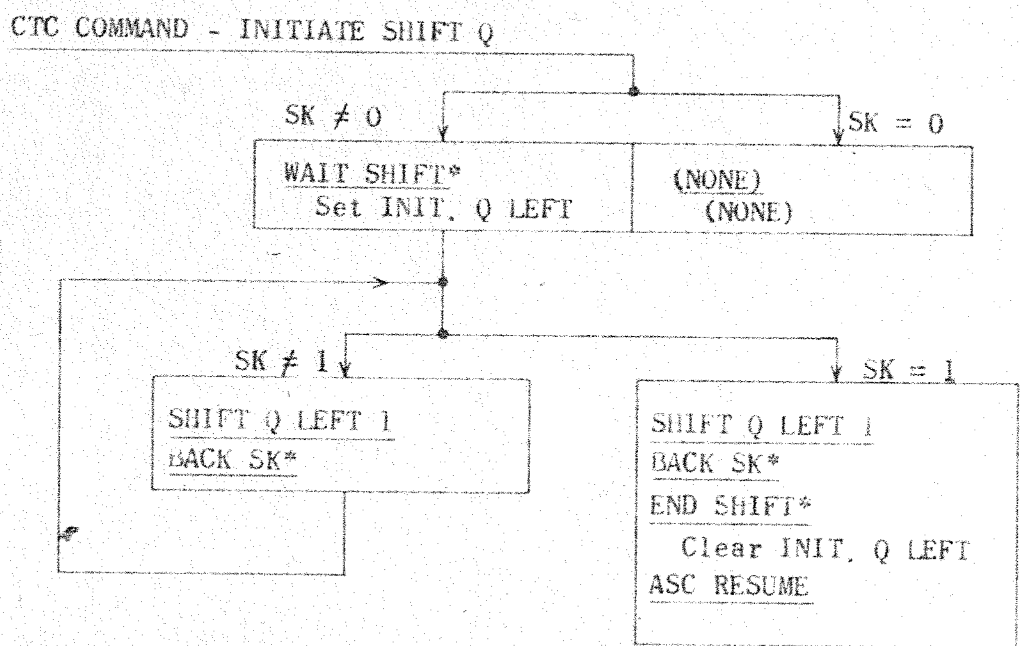
ASC/SKC SHIFT A SEQUENCE



* Signals produced in SKC

PX 71209

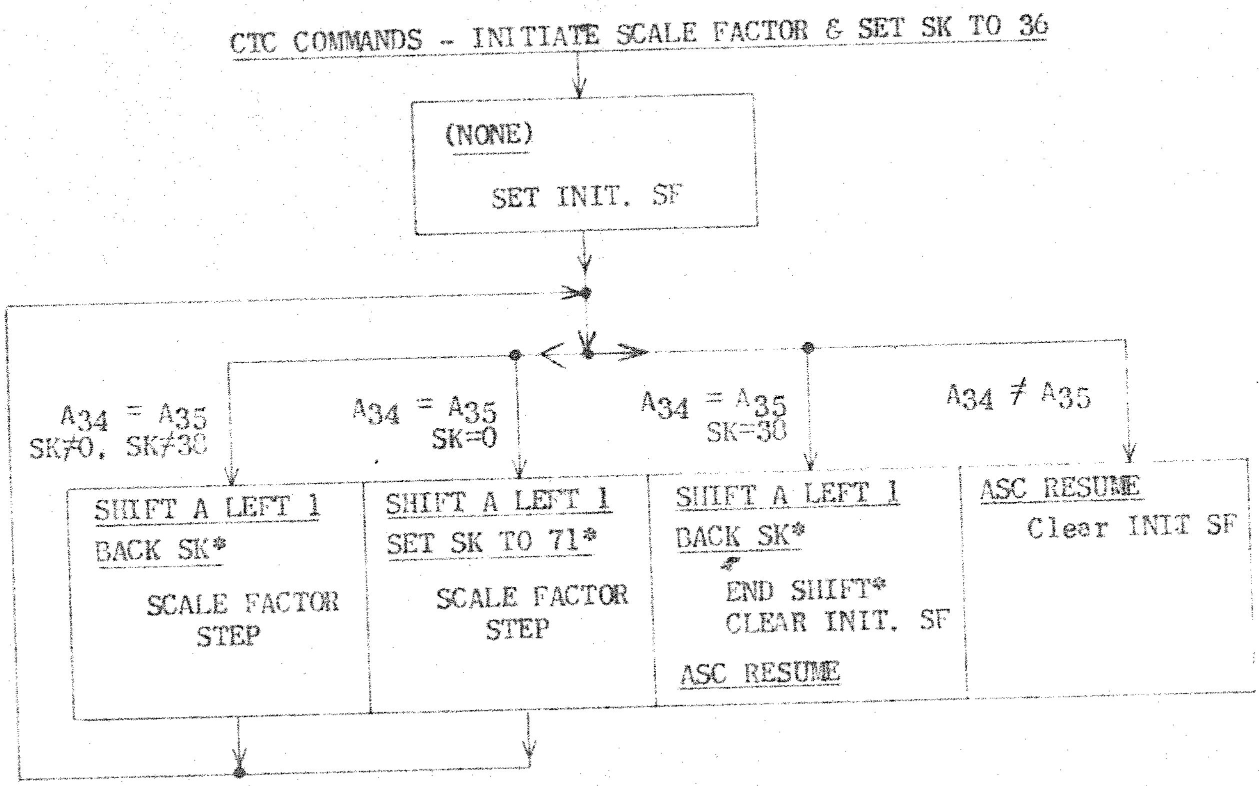
ASC/SKC SHIFT Q SEQUENCE



* Signals produced in SKC

PX 71209

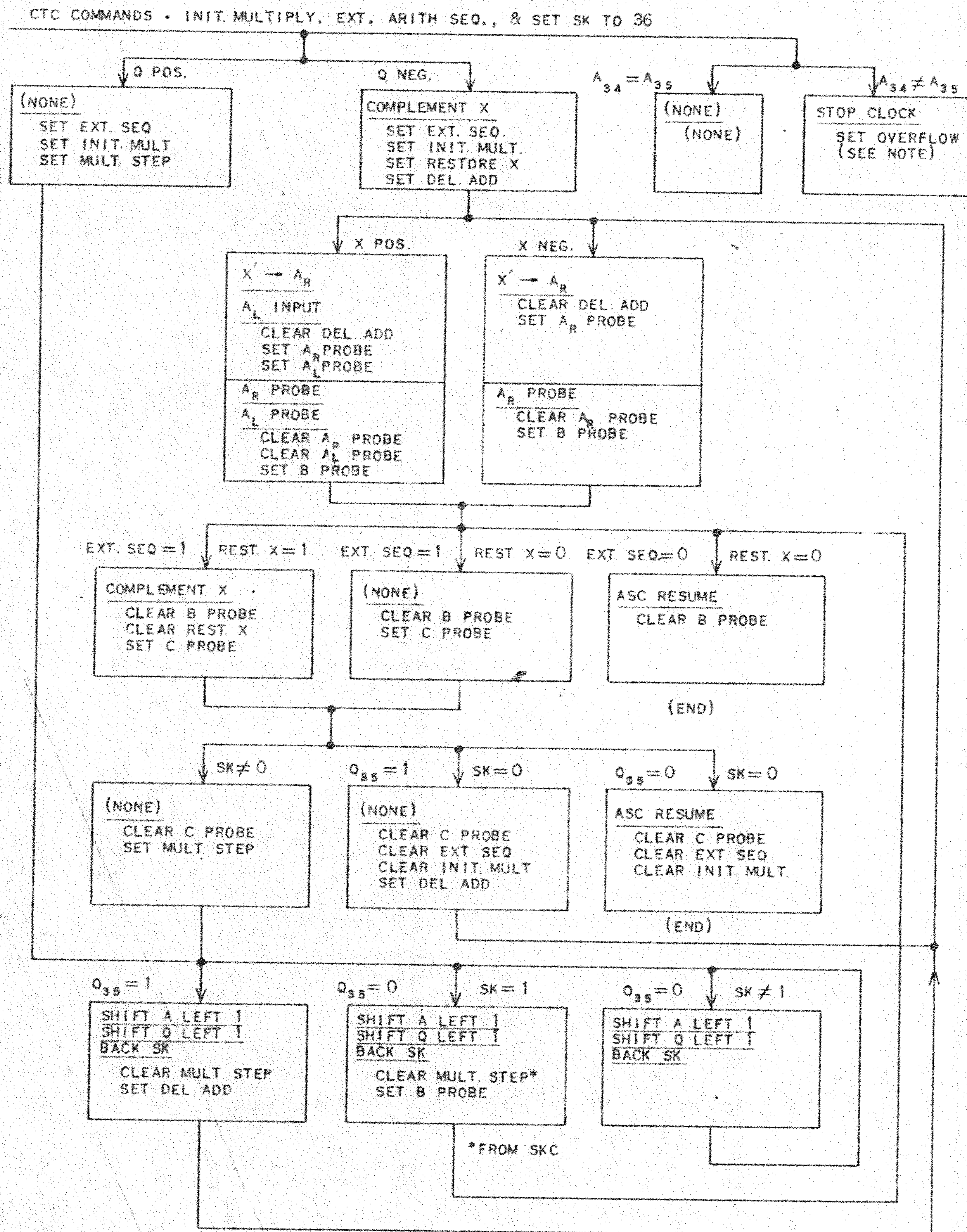
ASC/SKC SCALE FACTOR SEQUENCE



*Signals produced in SKC

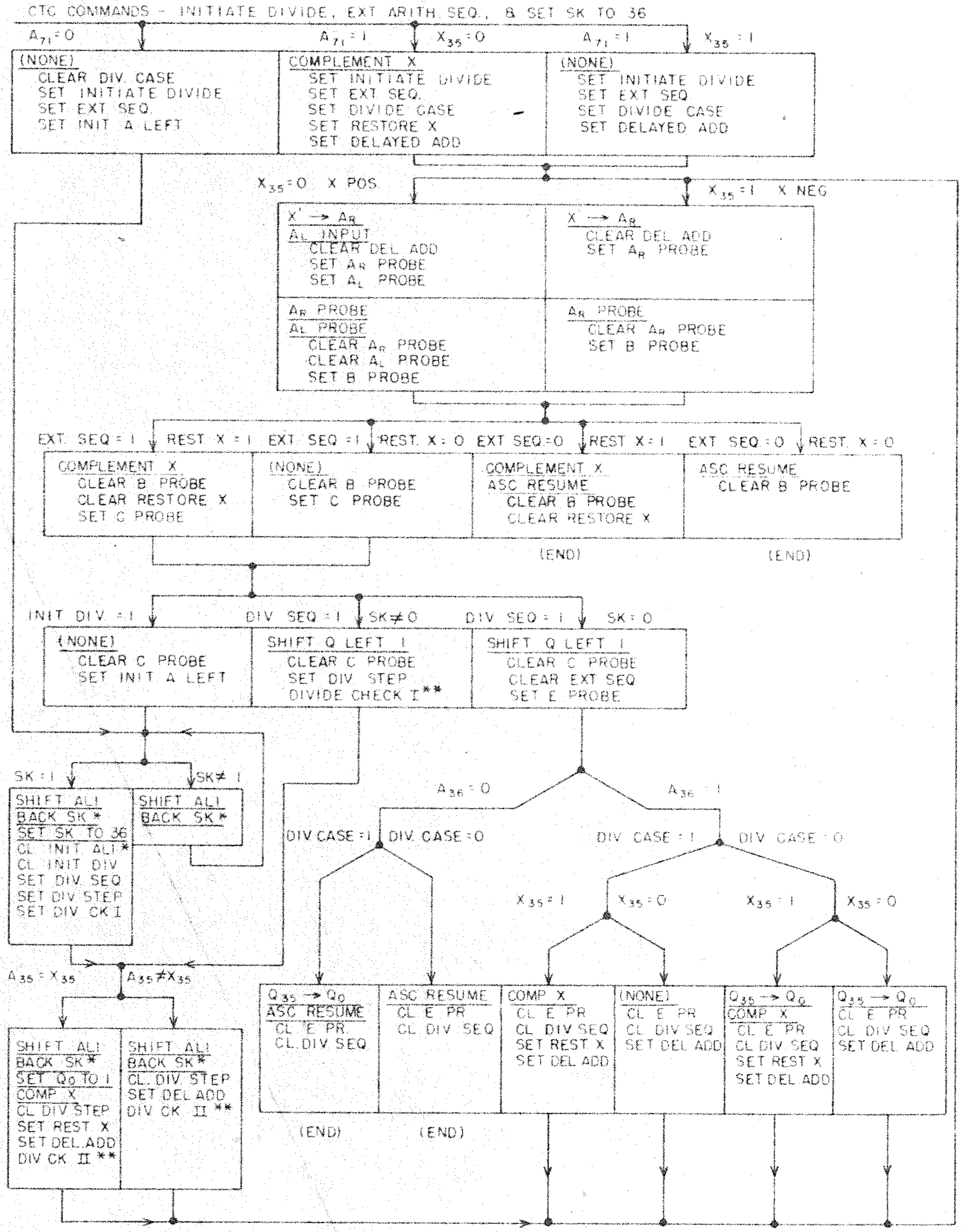
NOTE: Prior to the initiation of the Scale Factor Sequence the contents of A is shifted left 36 places. During the sequence a test is made to determine if $A_{34} \neq A_{35}$. If this condition is met the sequence ends. If this condition is not met the contents of A is shifted left and the test repeated. This testing and shifting continues until $A_{34} \neq A_{35}$ or, if this condition is never met, until all bits of A have been tested. At the end of the sequence SK contains the number of shifts necessary to return A to its original contents.

ASC/SKC MULTIPLY SEQUENCE



NOTE: SETTING THE OVERFLOW FLIP-FLOP PRODUCES AN A FAULT. CLEARING THE A FAULT CLEARS THE OVERFLOW FLIP-FLOP AND ALLOWS THE ASC MULTIPLY SEQUENCE TO PROCEED.

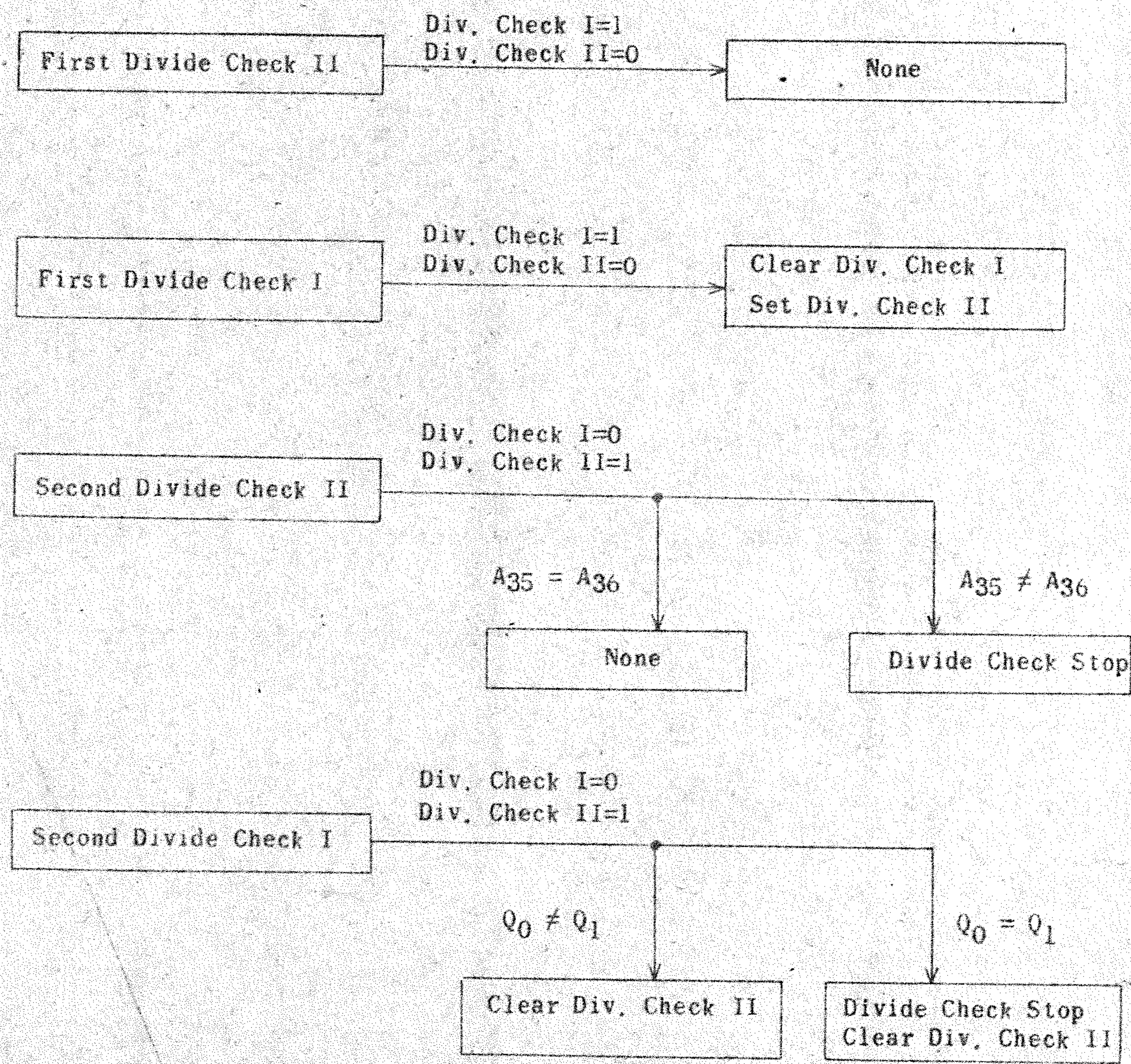
ASC/SKC DIVIDE SEQUENCE



*FROM SKC **DIVIDE CHECK PROBES MADE BEFORE SHIFTING. SEE NEXT PAGE.

PX 71209

ASC DIVIDE SEQUENCE, Continued

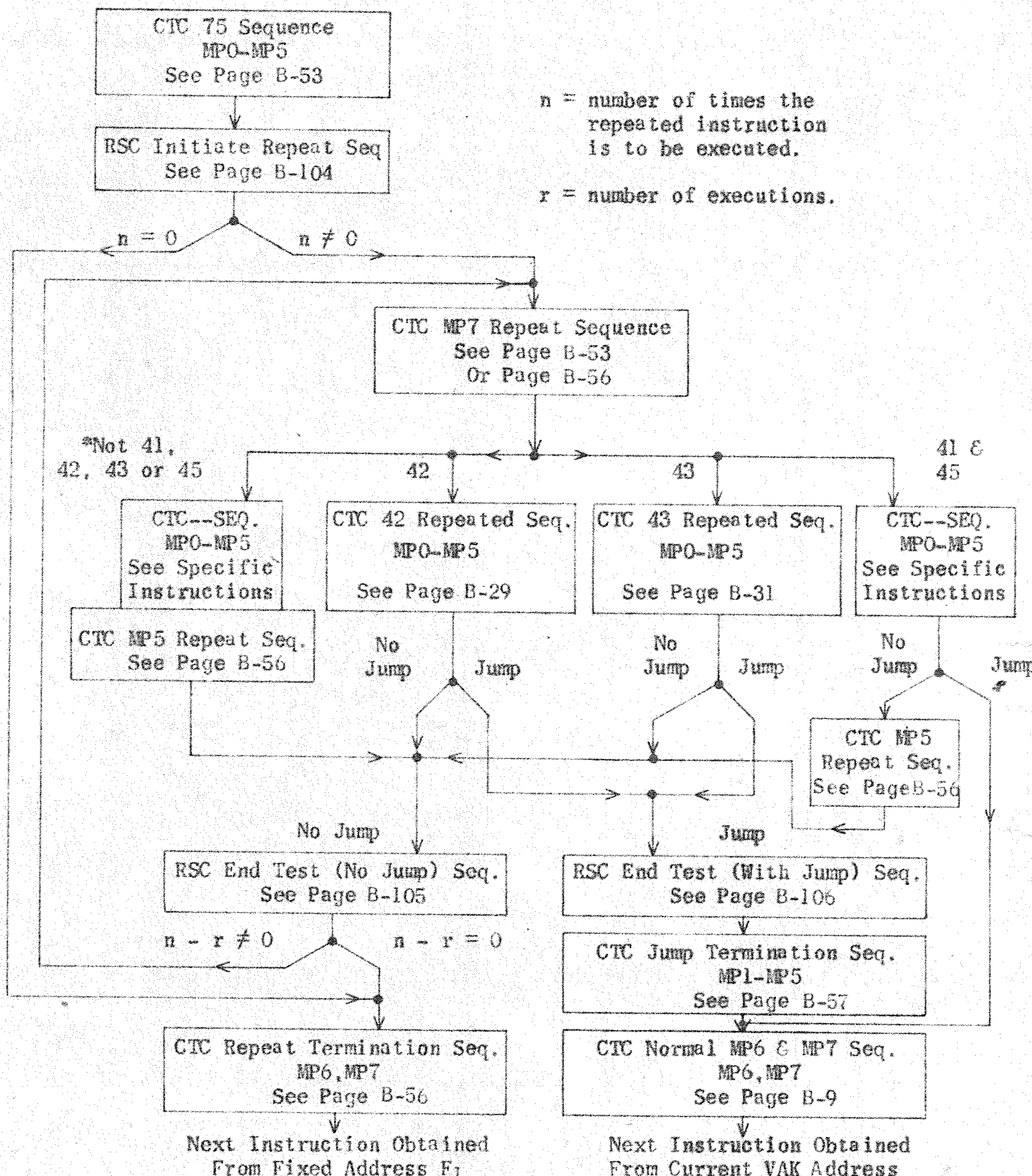


NOTES:

1. After the above division checks have been made the Divide Check. I & II flip-flops are both cleared. Divide check Probes I & II are produced but have no effect.
2. Divide Check Stop produces an A Fault Stop.

PX 71209

REPEAT SEQUENCES



*NOTE: Instructions 14, 37, 44, 46, 47, 56 & 57 cannot be repeated because RSC is cleared or the clock is stopped during the execution of these instructions. These instructions therefore, proceed as if no repeat preceded them.

RSC INITIATE REPEAT SEQUENCE

NOTE: During a CTC 75 sequence the 75 flip-flop is set to "1" and the Hold Repeat flip-flop is cleared to "0".

CTC COMMAND: Initiate Repeat →

	RSC SUBCOMMAND	DESTINATION
	Set Init. Repeat to "1" Set End Repeat to "1"	
CP	Advance PAK (End Repeat from PAK clears End Repeat to "0" if n=0.) Clear Init. Repeat to "0" Clear 75 FF to "0" Set Hold Repeat to "1" Set Init. Test to "1" Clear Adv. Address to "0"	PAK
CP	Clear Init. Test to "0" Set Delay Test to "1"	

	n≠0	n=0	
CP	Set MPD to 7		MPD
	Set Adv. Address to "1"		
	Clear Delay Test to "0"	Clear Delay Test to "0"	
	RSC Resume	RSC Resume	PDC

NOTE: Destinations are shown for those RSC signals which are sent to other sections of the computer.

PX 71209

RSC INITIATE REPEAT SEQUENCE

NOTE: During a CTC 75 sequence the 75 flip-flop is set to "1" and the Hold Repeat flip-flop is cleared to "0".

CTC COMMAND: Initiate Repeat → RSC SUBCOMMAND

	RSC SUBCOMMAND	DESTINATION
	Set Init. Repeat to "1"	
	Set End Repeat to "1"	
CP	Advance PAK (End Repeat from PAK clears End Repeat to "0" if n=0.)	PAK
	Clear Init. Repeat to "0"	
	Clear 75 FF to "0"	
	Set Hold Repeat to "1"	
	Set Init. Test to "1"	
	Clear Adv. Address to "0"	
CP	Clear Init. Test to "0"	
	Set Delay Test to "1"	

	n≠0	n=0	
CP	Set MPD to 7		MPD
	Set Adv. Address to "1"		
	Clear Delay Test to "0"	Clear Delay Test to "0"	
	RSC Resume	RSC Resume	PDC

NOTE: Destinations are shown for those RSC signals which are sent to other sections of the computer.

PX 71200

RSC END TEST (NO JUMP) SEQUENCE

NOTES: 1. This sequence is performed after a repeated instruction sequence if Initiate Jump Terminate is not received from CTC.

2. During MP5 of the repeated sequence, PAK is advanced. If n-r=0, End Repeat from PAK_{11} clears End Repeat to "0"

CTC COMMAND		RSC SUBCOMMAND	DESTINATION
Initiate End Test →		Set Initiate Test to "1"	
	CP	Set Delay Test to "1"	
		Clear Initiate Test to "0"	

	n-r≠0	n-r=0	
CP	Set MPD to 7		MPD
	Conditionally Advance UAK and VAK		PCR
	Clear Delay Test to "0"	Clear Delay Test to "0"	
	RSC Resume	RSC Resume	PDC

NOTE: Destinations are shown for those RSC signals which are sent to other sections of the computer.

PX 71109

RSC END TEST (WITH JUMP) SEQUENCE

NOTES: 1. This sequence is performed after a repeated instruction sequence if Initiate Jump Terminate is received from CTC.

2. During MP5 of the repeated sequence, PAK is advanced. If n-r=0, End Repeat from PAK_{11} clears End Repeat to "0".

CTC COMMANDS		RSC COMMAND	DESTINATION
Initiate Jump Terminate Initiate End Test	→	Set Jump Terminate to "1" Set Init. Test to "1" Clear Hold Repeat to "0"	
	CP	Set Delay Test to "1" Clear Init. Test to "0"	
	CP	Clear X	X
		Set MPD to 1	MPD
		Clear Delay Test to "0"	
		RSC Resume	PDC

NOTE: Destinations are shown for those RSC signals which are sent to other sections of the computer.

PX 71202