THE National* 315

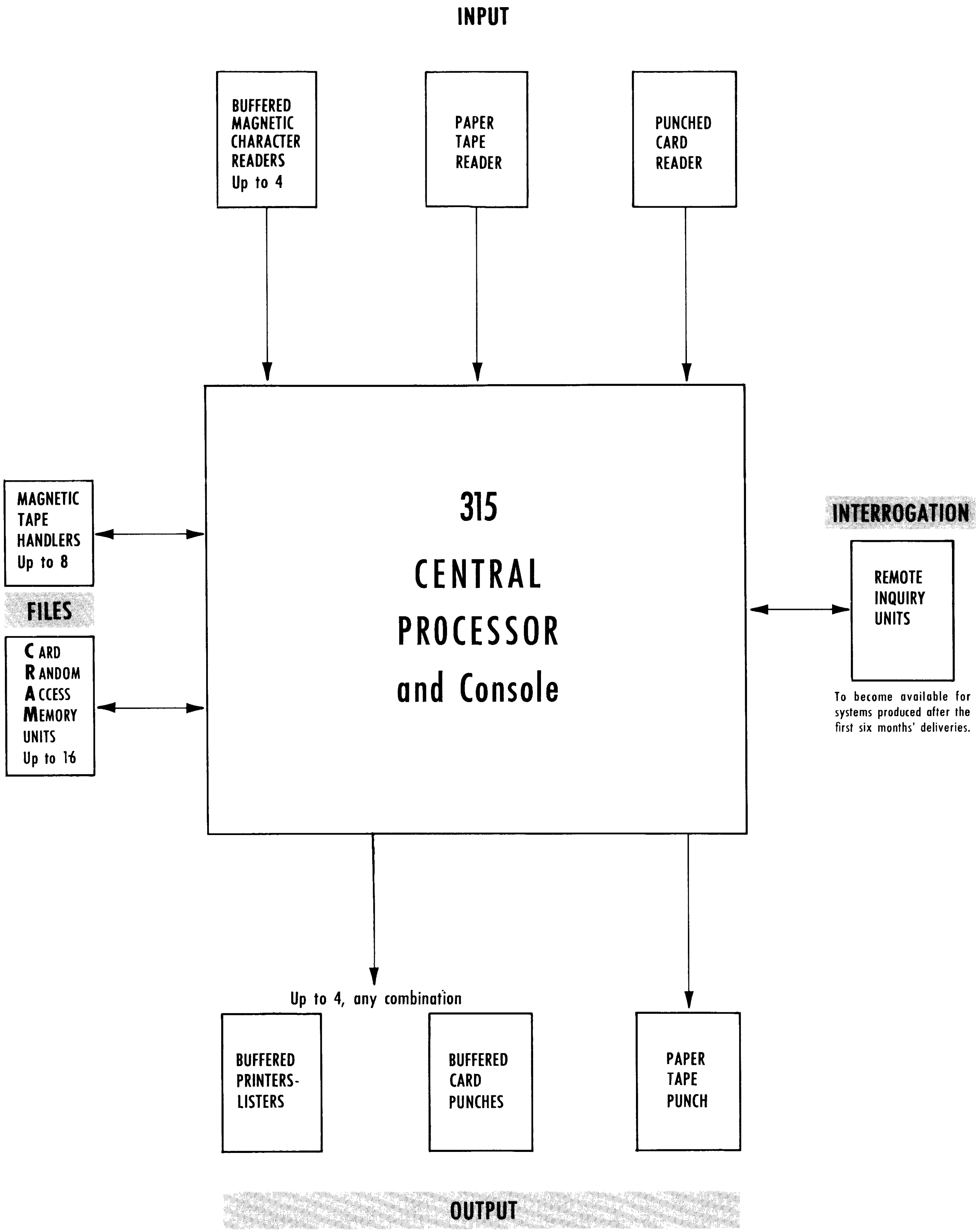
ELECTRONIC DATA PROCESSING SYSTEM

INPUT, OUTPUT, FILE OPERATIONS

A

THE NATIONAL CASH REGISTER COMPANY • DAYTON 9, OHIO

PRINTED IN U.S.A. F-7405 6-61 The NCR Co.



INPUT
BUFFERED MAGNETIC CHARACTER READERS Up to 4
PAPER TAPE READER
PUNCHED CARD READER
MAGNETIC TAPE HANDLERS Up to 8
FILES
CARD RANDOM ACCESS MEMORY UNITS Up to 16
315
CENTRAL PROCESSOR and Console
INTERROGATION
REMOTE INQUIRY UNITS
To become available for systems produced after the first six months' deliveries.
Up to 4, any combination
BUFFERED PRINTERS-LISTERS
BUFFERED CARD PUNCHES
PAPER TAPE PUNCH
OUTPUT

TABLE OF CONTENTS

*These instructions permit a "literal" to be named as "A" if desired.

INTRODUCTION

This publication, describing the input, output, and file operations of the NCR 315 Data Processing System, assumes complete familiarity with the previous publication which describes the internal operations of the processor. The operations for Inquiry Units and for Magnetic Character Sorter-Readers are the subject of another publication, and are not discussed here.

Processor communication with the "outside world" is performed in Alpha form (6 bits per character, 2 characters per slab) except for the Console Typewriter which accepts either Alphas or Digits for input-output. File information is recorded on magnetic tape and CRAM in "image alpha" form, whether it is actually Alpha or Digit.

Remember Except as specifically noted, none of these operations change the Sign flag, the Overflow flag, or the contents of the Accumulator.

PERIPHERAL UNITS

CONSOLE TYPEWRITER:

The Console includes an electric typewriter which permits modest amounts of information to be typed out by the program, or to be entered by the operator, in either Digit or Alpha form.

During normal operations the Console Typewriter will be used to create a Log of operations, and for occasional input of information such as Today's Date, etc.

CARD READER:

Two models of Card Reader are available, one operating at 400 cards per minute, the other at 2000 cards per minute. Both are serial readers, reading one card column at a time photoelectrically.

The 400 cpm reader stores an image of each column as a pattern of 12 bits in one slab of memory. There is sufficient time between columns for the processor program to translate the column images into the actual characters, and to do a certain amount of validity-checking (double-punch, blank-column, etc). Any kind of non-standard punching or binary punching may be read with the standard translation program, by storing appropriate tables in memory.

The 2000 cpm reader has automatic translation circuitry within it, and stores the actual characters in memory after checking for invalid configurations. The automatic translation may be by-passed if desired, to read non-standard or binary punching with programmed translation; the reader can switch from translate to non-translate and back any number of times within a card.

Both readers leave the processor free between columns, during the unread portion of each card (if less than the entire card is read), and between cards. They both exercise Demand as each new card reaches the reading station.

PAPER TAPE READER:

The Paper Tape Reader operates at 1000 characters per second, and will stop between characters at any time, under control of the processor program. It stores an image of each row of punching as a binary configuration in memory, and there is sufficient time between characters for the program to translate these images into the actual characters, and to do validity-checking.

Any code system whatever, up to 8 channels, will be accepted by the reader, and can be translated by the standard program, with appropriate tables.

LINE PRINTERS:

The processor can operate up to four printers or card punches, in any combination. Each printer is completely buffered, and performs all printing and paper movement under its own control, with automatic end-of-page detection. It has a 56-character font, and prints a 120-character line.

The printer has a 2-speed switch for printing at 680 alphanumeric (900 numeric) lines per minute, or at 380 alphanumeric (450 numeric) lines per minute. The lower speed provides superior print quality for photo-offset reproduction. The higher printing rates (at both speeds) for numeric information arise automatically out of the characteristics of the printer itself, and are obtained for any mixture of numeric and alphanumeric lines; no special programming or special adjustment of the printer is required.

When printing at either speed, paper movement over blank lines is at the rate of 14 inches per second (5040 lines per minute) regardless of the number of blank lines.

A triple-feed Lister attachment is available, and may be attached and removed by the operator. The printer then produces three separate 30-character-wide lists, side by side on three strips of paper, with independent paper movement provided by individual pin-feed tractors for each list.

After each line has been printed, the printer is READY and can exercise Demand, so that the processor program may immediately transmit the information for another line.

CARD PUNCHES:

The processor can operate up to four card punches or printers, in any combination. The punches may be IBM 523 (100 cards per minute) or IBM 544 (250 cards per minute). Each punch is completely buffered, and performs all punching, and plugboard-programmed editing and checking, under its own control.

Each Card Punch Buffer automatically translates the characters from the processor memory into standard card code for punching. There is a switch on the buffer, permitting the translation to be by-passed, in which case each column of the card is punched as an image of the bit-configuration in one slab of memory. This feature permits the output of cards with any kind of non-standard or binary punching.

When each card is completely punched, the punch is READY and can exercise Demand, so that the processor program may immediately transmit the information for another card.

PAPER TAPE PUNCH:

The paper tape punch operates at 110 characters per second, and can punch any code system whatever, up to 8 channels. The time between characters is available to the processor program for translation of the characters into the required hole-configurations, and for any other processing work.

MAGNETIC TAPE HANDLERS:

The processor may accommodate up to 8 handlers, each holding a reel of tape up to 2400 feet long. The tape moves at 120 inches per second, and three recording densities are available:

200 alphanumeric characters per inch
  24,000 characters per second (24 kc)
  Compatible with IBM tapes.

333 alphanumeric characters per inch
  40,000 characters per second (40 kc)
  60,000 digits per second

500 alphanumeric characters per inch
  60,000 characters per second (60 kc)
  90,000 digits per second

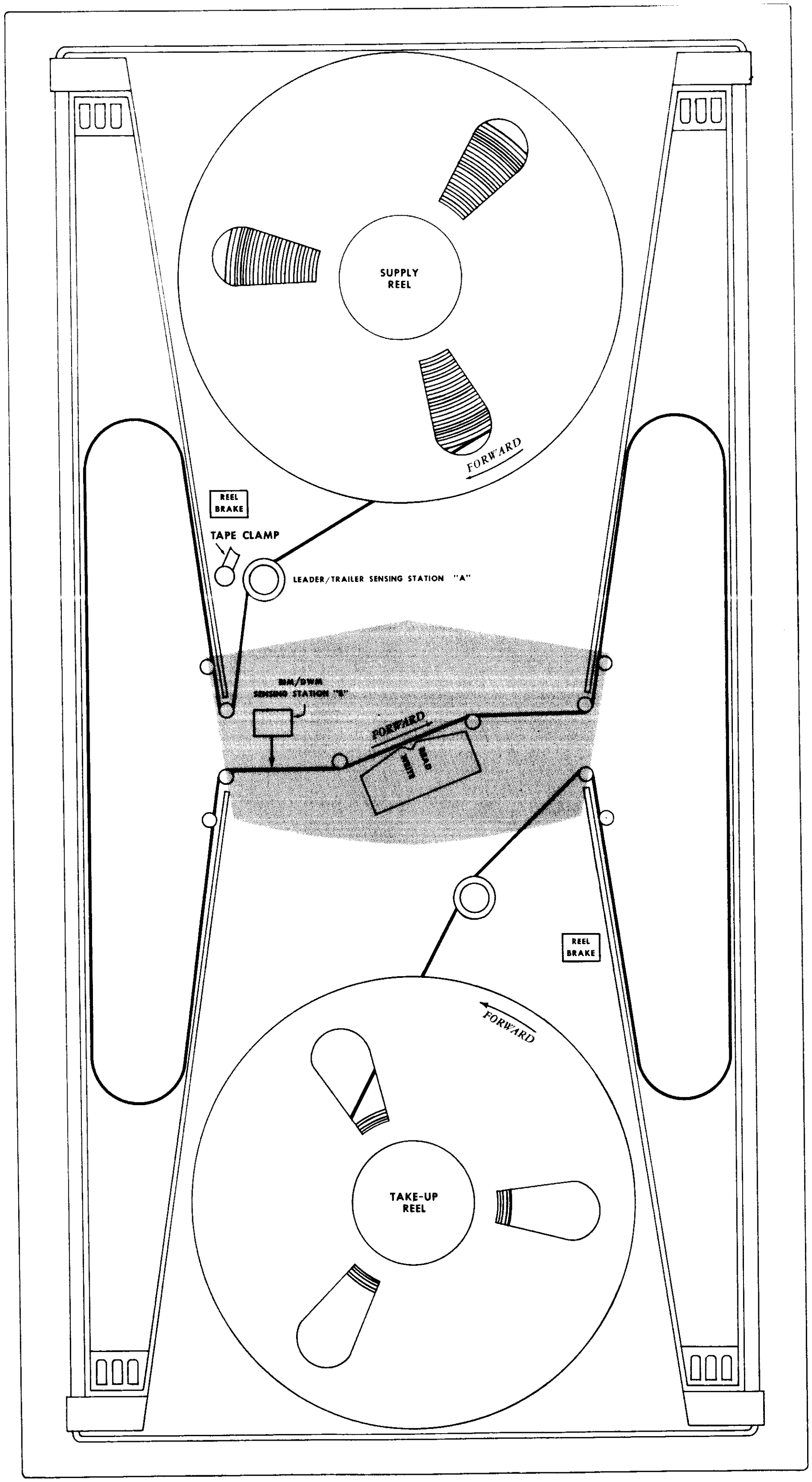
The Standard Handler can read and write tape at 24 and 40 kc. The High Speed Handler can read and write at all three speeds. Each handler has a switch to determine the density of writing; while reading, this switch must correspond to the density at which the tape was recorded.

Information is recorded on magnetic tape in "image alpha" form. Images of the left-hand and right-hand six bits of each slab are recorded as alpha characters on the tape. If the slab contains alpha information, these will be the actual characters; if the slab contains digit information, these will be pseudo-characters, each comprising $1^1/_2$ digits, which will recreate the image of the slab in memory when the tape is read. Therefore three digits take the same amount of tape, and the same read-write time, as two alphas, so that information transfer rates for numeric information are $1^1/_2$ times those for alpha-numeric information.

An information record on tape may contain 1 to 7,999 slabs (2 to 15,998 alphanumeric characters, 3 to 23,997 digits or any proportional mixture). Between records there is a three-quarter-inch *gap,* (length of blank tape). It is clearly advantageous to *block* records—to put as many accounts as possible into a single record—to minimize the number of gaps. The large number of index registers in the NCR 315, and their flexibility in use, make this procedure extremely convenient, whether the records are fixed or variable in length.

As the tape is being written, the processor automatically adds a parity-bit to each character, and records 7 bit-channels on the tape, Even-parity is established for 24-kc tapes, odd-parity for 40-kc and 60-kc tapes. At the end of the record, the processor automatically adds a parity-character to the record, establishing lengthwise parity along each channel. A reading head on the handler, immediately behind the writing head, automatically reads back the information just written, and checks the 2-dimensional parities; if there has been any error in writing the record, there is an automatic branch in the processor program, so that the tape may be backed up, and the record re-written. This branch actually goes into a master program called **STEP** (Standard Tape Executive Program), which makes several attempts to write the record, and if still unsuccessful, lays down a unique *skip* record and then writes the desired record on the tape. When the tape is later read, this *skip* record will cause another automatic branch into **STEP**, which will discard it and proceed with the processing.

STEP is an extremely sophisticated program, which is held in memory at all times when magnetic tapes are used, and which handles automatically *all* housekeeping chores connected with tapes. These chores include label-checking, error-correction, record counts and check sums, end-of-tape alternation of handlers, end-of-file detection, run-to-run supervisor, etc.



MAGNETIC TAPE TRANSPORT

When a tape is mounted on a handler, it follows the path shown in the diagram. The loops at the sides are vacuum chambers, and the grey area represents a cover which should be removed only by maintenance personnel. A leader is permanently fastened to the take-up reel, and when the tape is completely rewound, the tape-to-leader splice is between the tape clamp and the supply reel, with the leader threaded through the mechanism. In order to change tapes, the operator opens the door, closes the tape clamp, and disconnects the mechanical splice. He changes the reel, connects the splice, releases the clamp, and closes the door. Electronic interlocks prevent any use of the handler until the clamp is released and the door closed.

After the tape has been loaded, the first Read or Write instruction addressed to the handler causes the take-up reel to wind the leader, and the tape, until the Beginning-of-Information Marker (BIM) is detected at the BIM/DWM Sensing Station "B" and then reading or writing begins. When a Rewind instruction is addressed to the handler, the mechanism is reversed and the supply reel winds the tape until the BIM is detected at "B". The tape is now ready to be re-read, or re-written, if desired.

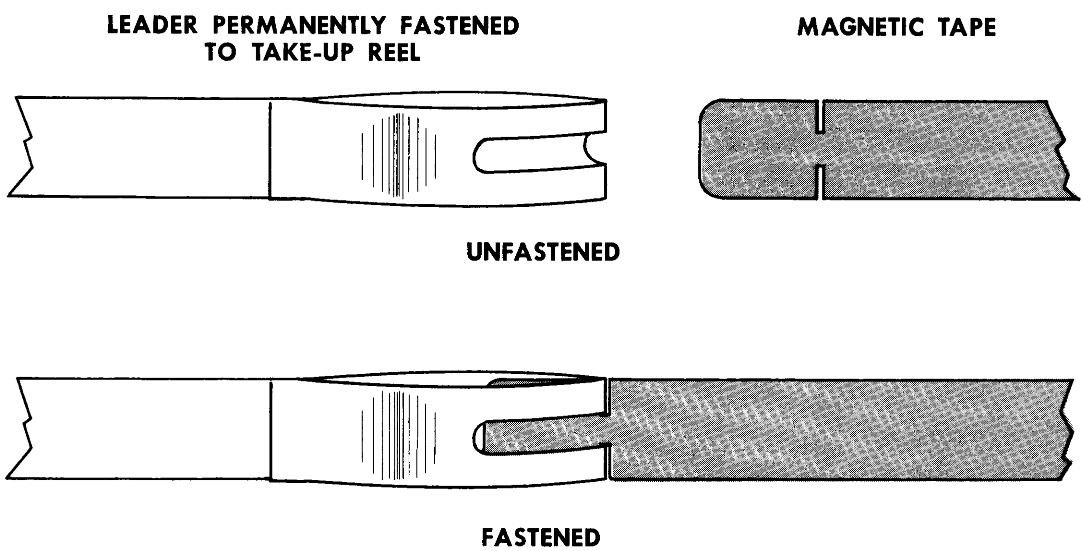
When recording tape, station "B" is alert for Destination Warning Marker (DWM) and upon detecting it, signals the processor that the physical end of the tape is approaching. The processor program should now write an end-of-tape record. The BIM and DWM are reflective spots on the back of the tape, 10 feet and 18 feet from the beginning and end, respectively, and station "B" detects them photo-electrically.

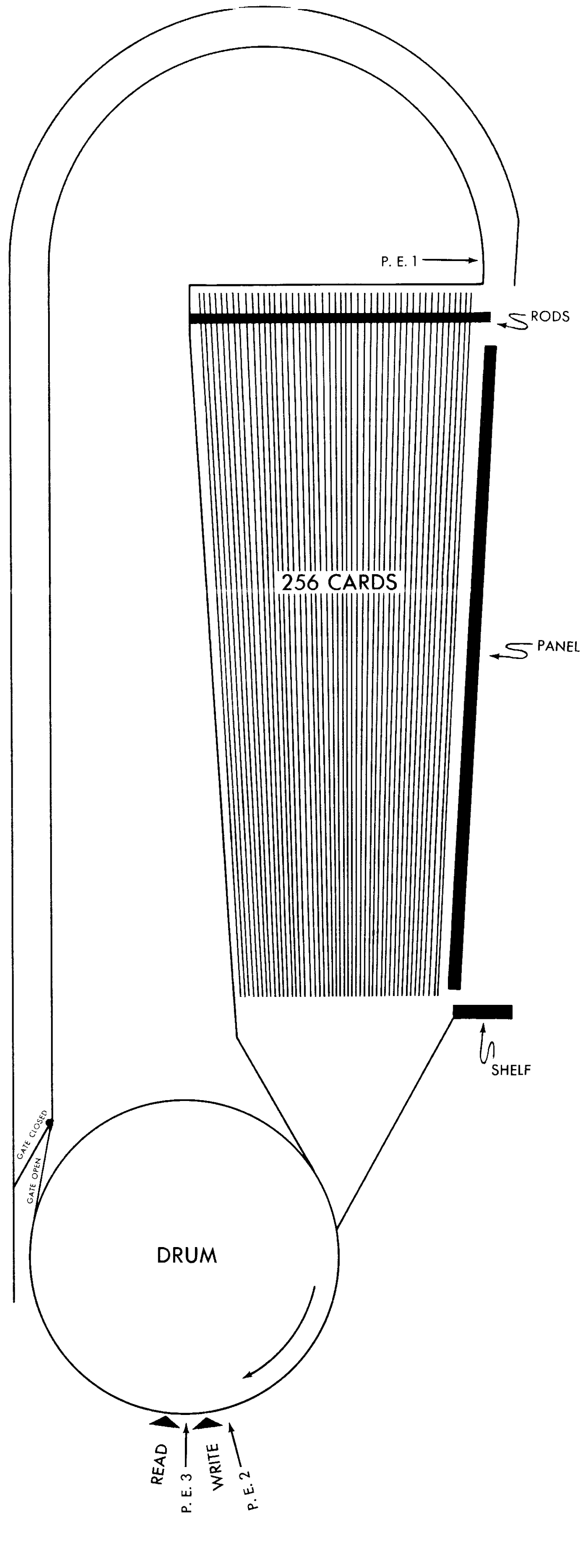
The leader and the trailer ("trailing leader") are made of electrically-conductive material, and are detected at the Leader-Trailer sensing station "A". If writing continues too long after the DWM, or if an attempt is made to read past the end of the recorded information, station "A" will detect the trailer before any physical damage can result, and the handler will go out of the "operate" status.

When a reel of tape is to be changed, after being rewound to the BIM, the operator presses the manual Rewind button on the handler, causing additional rewind until the leader reaches station "A" and passes a few inches beyond, in position to be clamped conveniently.

TAPE-TO-LEADER SPLICE

Actual Size





P. E. 1
RODS
256 CARDS
PANEL
SHELF
GATE CLOSED
GATE OPEN
DRUM
READ
P. E. 3
WRITE
P. E. 2

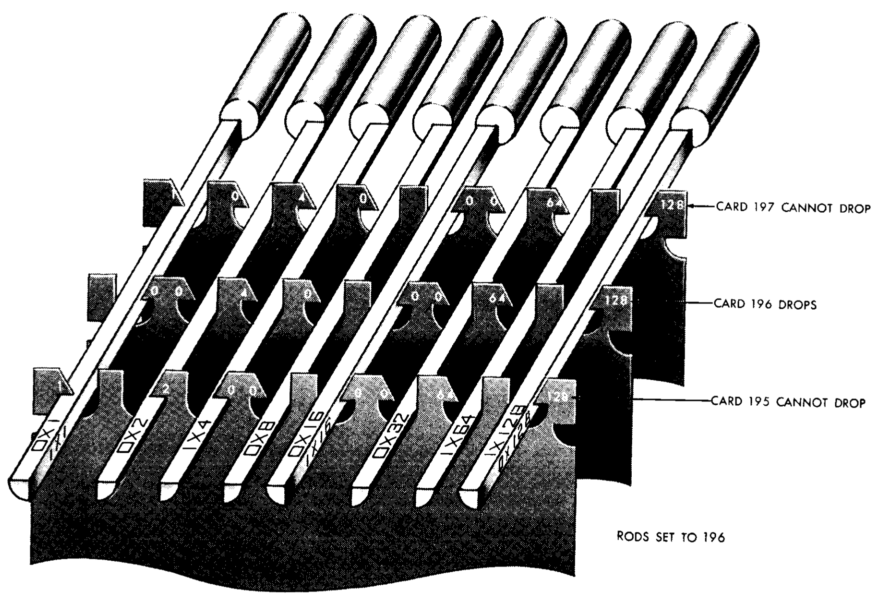
CARD RANDOM ACCESS MEMORY (CRAM):

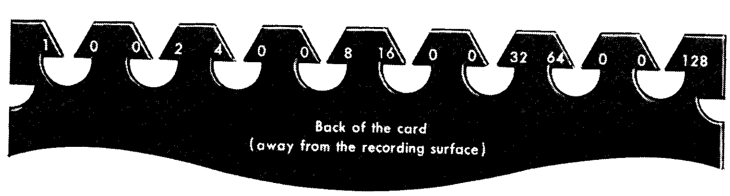
A CRAM unit holds a deck of 256 magnetic cards. When the processor program calls for a particular card, that card drops out of the chamber and wraps around a rotating drum, where it is held by air suction. The card now has the properties of a magnetic drum memory, in which the recording occupies $^2/_3$ of the drum circumference. One-third of each drum-revolution is free for updating the information so that it is ready for re-recording during the next revolution.

When the program has finished with this card the exit gate opens, the card is released from the drum, and it goes up through a chute back to the chamber. When the leading edge of the card reaches a photo-electric cell (P.E. 1 in the diagram opposite) the panel retracts to allow the card into the chamber, and the shelf comes forward to stop the card. Then the panel comes forward to put the card into the chamber and the shelf slides back, out of the way of the next card to be dropped.

Each card has eight notches across the top, cut in a pattern which is unique to that card, and the cards hang in the chamber from a row of eight selection rods. The diagram below shows the coding scheme for the notches, and the positions of the rods. When the processor calls for a particular card, the rods rotate into the combination of positions corresponding to that card. A pair of gating rods, which fit into notches at the sides of the cards, support the entire deck while the rods are coming into position; then they swing back, and the selected card drops out of the chamber.

This achieves a true random selection of cards, as each card is chosen by the selection rods regardless of its position within the deck. In fact, the physical sequence of the cards is itself random, since the last-used card is always replaced at the back of the deck.

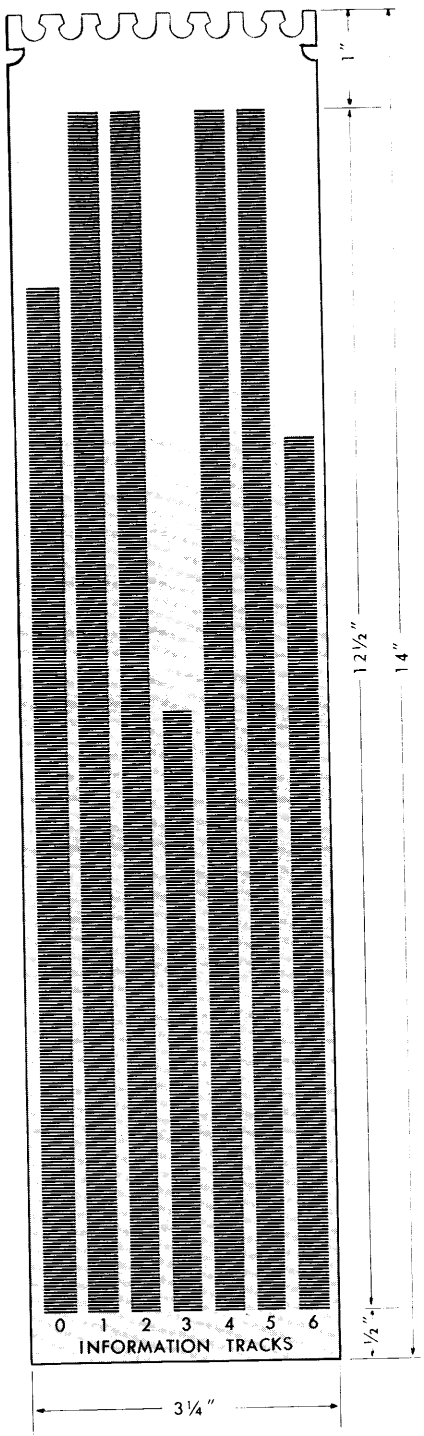




CODING OF NOTCHES

In order to encode a card, one point is clipped from each notch.

The card number is then the sum of the numbers on the remaining points.



One-half actual size

EACH TRACK CONTAINS UP TO 1550 SLABS

CAPACITY OF 1 CARD: OVER 32,500 DIGITS

CAPACITY OF 1 CARTRIDGE: OVER 8,300,000 DIGITS

While the card is on the drum it may be written or read in any of seven information-tracks. No switching time is involved in changing from one track to another, as the CRAM is provided with individual read-write heads for each track. The pattern of tracks on the card is shown in the diagram. Each information-track holds one record of information which may contain 1 to 1,550 slabs (2 to 3,100 alphanumeric characters, 3 to 4,650 digits or any proportional mixture). Recording is performed in "image alpha" form, in seven bit-channels per track, with character-parity and record-parity, as on magnetic tape, And as with magnetic tape, each track has a read head immediately behind the write head, so that all recording is automatically checked while it is being performed. Information transfer rate is 100 kc (100,000 alphanumeric characters or 150,000 digits per second).

A Read or Write operation always starts at the beginning of a track, and terminates as soon as information transfer is complete, even if the record is less than 1,550 slabs long. The CRAM also has the ability to execute a partial-read, which terminates at any desired point within the record. Termination of the operation at end of information, rather than at the end of the track, allows a greater portion of the drum-revolution to be shared with other processor operations.

A master program called **PACE** (PAckaged CRAM Executive), which is analagous to **STEP**, performs all housekeeping chores for CRAM. The scope of **PACE** is similiar to that of **STEP**.

When the processor calls for a CRAM to drop a particular card, the selection rods release that card and it starts to drop. From the moment the processor calls for the card to the moment the leading edge of the card reaches photo-electric cell P.E. 2 (see diagram on page 6) the CRAM is in a DROPPING status, and is inhibited from accepting a command to drop another card.

As soon as the leading edge of the card reaches P.E. 2 the DROPPING signal is turned off, and the LOADED signal goes on, indicating that a card is loaded on the drum. Only a portion of the card is actually in contact with the drum at this point, but reading or writing may now begin.

When a card is released from the drum, the CRAM is no longer LOADED, and it will not accept a Read or Write instruction.

The CRAM will accept a Read or Write instruction at any time a card is LOADED, but the instruction will be executed with minimum delay only while the leading edge of the card is between P.E. 2 and P.E. 3. Once the leading edge has passed P.E. 3, it is too late to read or write during this revolution, since the beginning of the track has passed the write head; and until the leading edge reaches P.E. 2, it is too early to read or write during the next revolution. When the leading edge is between the two P.E. cells, the card is in *minimum-access position* and can exercise Demand so that the processor program may immediately read or write.

A good deal of time-sharing is possible with the CRAM. As soon as a card is no longer DROPPING, the unit may be instructed to drop the *next* card, and then to read or write the card which has just been loaded (the *present* card). In this case, the unit is both DROPPING and LOADED at the same time. This situation will remain for one revolution of the card, and then the present card will automatically be released from the drum, and return to the chamber.

When processing CRAM files, the program will determine when it is about to perform the last Read or Write on a card, and will call for the next card before performing that last operation on the present card. There will be a significant interval between completion of that operation and the moment when the CRAM is LOADED with the next card; the processor will utilize that interval to perform other work, and as soon as the next card reaches P.E. 2 (the CRAM is LOADED) it will exercise Demand.

If an interval of 750 milli-seconds passes without any command being addressed to a CRAM, the card on the drum is released and returned to the chamber.

READY STATUS AND DEMAND INTERRUPT

As a general principal, if a peripheral unit is READY, and if its Unit Demand flag is set, it will transmit a *demand signal* to the processor; if the processor's Demand Permit flag is set, Demand interrupt will occur.

When interrupt occurs, the demand program must determine which of several peripheral units caused the interrupt. This is done by attempting to *select* each unit in turn, since the Select operation includes a test for READY. However, each unit has its individual characteristics, which are summarized in the table on the next page.

Note that the card reader does not have a Unit Demand flag. When reading cards there will usually be no occasion to have any other units demanding, since the timing of the card reader will govern other peripheral operations.

Priorities among competitively-demanding units are determined simply by the sequence of the SELECT commands in the program.

	READY		DEMAND SIGNAL Exercise Demand if Demand Permit flag on	
	ON	OFF	ON	OFF
PRINTER	Finished printing a line	PRNT instruction or out of paper	READY and Unit Demand on	READY off or Unit Demand off
CARD PUNCH	Finished punching a card or Could not punch a card	PNCH instruction / PNCH instruction takes branch	READY and Unit Demand on	READY off or Unit Demand off
CARD READER	Card leading edge reaches reading station	Card column 1 reaches reading station	READY on	READY off
CRAM	Card in minimum access position AND Unit Demand on; unless present card after SELC:DN	WCC or RCC instruction or card passed P.E. 3 or Unit Demand off	READY on (includes Unit Demand on)	SELC instruction or READY off

STUDENT'S NOTES

SELECT a CRAM

OP	V	L	X/Y	A/B
S E L C	(V)		X	A
			Y	J

Jump is to the instruction whose address is "J" + (contents of RY).
Link in R15.

SELC:DP **Select** a **CRAM**, **drop** a card.

The next Read or Write is intended for the **present** card.

Jump if a card is now DROPPING on this unit.

The present card remains available for one more Read or Write before it is released from the drum.

This unit will become READY (if its Unit Demand flag is set) next time the present card reaches minumum-access position.

If the next Read or Write is issued too late for the present card, *that* operation will take the "wrong card" jump.

Selecting a CRAM has no effect on its Unit Demand flag.

Selecting a CRAM turns off the Demand signal in that CRAM.

SELC:DN **Select** a **CRAM**, **drop** a card.

The next Read or Write is intended for the **next** card.

Jump if a card is now DROPPING on this unit.

If the next Read or Write is issued too soon for the next card, *that* operation will take the "not loaded" jump.

As soon as this instruction is given, the LOADED status is terminated, and the present card becomes inaccessible. The CRAM will become LOADED again only when the next card reaches the drum and, if the Unit Demand flag is on, it will become READY at the same time.

(**A**) is a 2-slab word:

CRAM No.	Card No.

CRAM numbers: 0- 7
10-17
only RH 5 bits are used

Card numbers: 000-255
expressed as an 8-bit binary number. A standard subroutine is provided for the decimal-to-binary conversion.

*SELC:T **Select** a **CRAM** and **test** for READY (card in minimum-access position, and Unit Demand set).

Jump if unit is READY.

(**A**) or "**A**" is CRAM number.

*SELC:R **Select** a **CRAM** and **release** present card after one more opportunity to Read or Write.

Jump if unit is not LOADED (no card on drum, or present card after SELC:DN).

(**A**) or "**A**" is CRAM number.

*SELECT other input-output devices

Op	V	L	X/Y	A/B
S E L P			X	A
			Y	J

Op	V	L	X/Y	A/B
S E L S			X	A
			Y	J

Op	V	L	X/Y	A/B
S E L Q			X	A
			Y	J

Jump is to the instruction whose address is "J" + (contents of RY).
Link in R15.

*SELP: Select a Printer or Card Punch.
(A) or "**A**" is unit number: 0-3
Only RH 2 bits are used.
Jump if unit is READY.

*SELS: Select a Sorter.
(A) or "**A**" is unit number: 0-3
Only RH 2 bits are used.
Jump if unit is READY.

*SELQ: Select an Inquiry buffer.
(A) or "**A**" is unit number.
Jump if unit is READY.

Selecting a unit has no effect on its Unit Demand flag.

These instructions do not turn off the Demand signal in the peripheral unit; the instructions which operate the units do turn it off.

*SET and CLEAR Unit Demand Flags

Op	V	L	X/Y	A/B
S E T U	(V)		X	A
			0 0	

Op	V	L	X/Y	A/B
C L R U	(V)		X	A
			0 0	

The second line of the instruction format is not used. It is suggested that zeros be entered in the Y column to indicate on a coding sheet that this is a double-stage instruction.

*SETU:C Set
*CLRU:C Clear } Unit Demand in CRAM.

(A) or "**A**" is unit number: 0-7
10-17
Only RH 5 bits are used.

*SETU:P Set
*CLRU:P Clear } Unit Demand in Printer or Card Punch.

(A) or "**A**" is unit number: 0-3
Only RH 2 bits are used.

*SETU:S Set
*CLRU:S Clear } Unit Demand in Sorter.

(A) or "**A**" is unit number: 0-3
Only RH 2 bits are used.

*SETU:Q Set
*CLRU:Q Clear } Unit Demand in Inquiry buffer.

(A) or "**A**" is unit number.

Setting or clearing Unit Demand has no effect upon Selection of units.

PUNCH PAPER TAPE:

Op	V	L	X/Y	A/B
P P T	(V)		X	A
				N

Punch N rows of paper tape from memory, starting with the LH end of the A-area.
N may be 000-999.

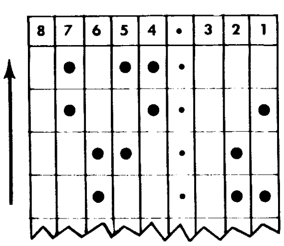
In each mode, an image of the bit-configuration in memory is punched into the tape, with holes corresponding to 1-bits. Channel 1 on the tape (see illustration below) corresponds to the RH bit of the character or the slab. Before punching, the program must translate each character into the bit-configuration used for that character in the paper tape code. Standard subroutines and macro-instructions are furnished for this purpose.

It is customary to translate and punch one character at a time, since there is sufficient time between characters to perform the table-lookup while the Punch is operating at its full speed of 110 characters per second.

When punching tape for communication between Processors, the Character mode is used and no translation is required, as the instruction will punch an exact image of the bit-configurations stored in memory. In that situation, N may be made as large as convenient.

PPT :C Each alpha **character** is punched into the RH 6 channels of a row on the tape. The Processor automatically punches *hole* or *no-hole* into the seventh channel to establish odd-parity (an odd number of holes in each row).

Thus the two slabs | H 9 | S L | would be punched as:

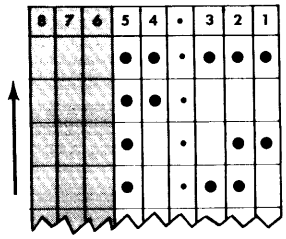


H (01 1000)
9 (00 1001)
S (11 0010)
L (10 0011)

The punching corresponds to QRST in tape-to-card conversion code (odd-parity). Before punching, the translation subroutine for this output code will have translated QRST into H9SL.

PPT :S The RH 8 bits of each **slab** are punched into the 8 channels of a row on the tape.

Thus the four slabs | 0 p | 0 H | 0 C | 0 F |
would be punched as:



0 p (00 0000 01 1111)
0 H (00 0000 01 1000)
0 C (00 0000 01 0011)
0 F (00 0000 01 0110)

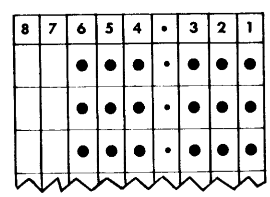
The punching corresponds to letter-shift, ABC in 5-channel telegraph code (no parity). Before punching, the translation subroutine for this output code will have translated ABC into 0p0H0C0F, and the translation would have included insertion of the required letter-shift.

NOTE: If, for any reason, the program issues a PPT too late to maintain full punching speed, the Punch will stop and wait. The only time lost is the waiting time.

If N = 0, these operations do nothing.

TAPE FEED:

When the Tape Feed button on the Punch is depressed, the Punch will continuously emit the following configuration until the button is released:



When punching in any given code, the program will initially punch a foot or two of the *run-in* configuration defined for that particular code, as a leader; and again at the end, as a trailer. After the tape has been punched, the operator may tear off the ends of the tape which contain the *tape feeds.*

When punching without translation, for communication between Processors, the *tape feed* itself will be used as the *run-in* configuration.

Special facilities have been provided in the Read Paper Tape instruction to accommodate the fact that this is an even-parity configuration.

READ PAPER TAPE

Op	V	L	X/Y	A/B
R P T	(V)		X	A
			Y	N

Two-way jump table starts in JY.
Link in J14.

Start the Reader (unless it is already running) and read N rows of paper tape into memory, starting at the LH end of the A-area.

N may be 000-999.

If the program issues another RPT instruction soon enough, the tape will continue moving at full speed. If the program does not issue another RPT instruction by the time the Reader reaches the next character, the tape will stop in position to read that character; this will cost an acceleration time when the tape is restarted.

In each mode, an image of the row is stored as a bit-configuration in memory, with 1-bits corresponding to holes. Channel 1 on the tape (see illustration under PPT) corresponds to the RH bit of the character or the slab. After reading, the program must translate each bit-configuration into the actual character. Standard subroutines and macro-instructions are furnished for this purpose.

It is customary to read and translate one character at a time, since there is sufficient time between characters to perform the table-lookup while the Reader is operating at its full speed of 1000 characters per second.

When tape has been punched without translation, for communication between Processors, the tape contains an image of the bit-patterns which were in memory at the time of punching; reading is done in the Character mode and no translation is required. In that situation, N may be made as large as convenient.

If N = 0, these operations do nothing.

JY: Parity error branch for RPT:C and RPT:CX.

JY + 1: Tape feed branch for RPT:CX if N = 1.

RPT :C The RH 6 channels of successive rows are read into successive **character**-positions in memory. Channel 7 is used by the Processor to verify odd-parity (an odd number of holes in each row) and is not stored in memory. Channel 8 is ignored.

Complete slabs are always stored. If N is an odd number, then a *zero* will be stored as the RH character of the last slab.

If any row on the tape contains an even number of holes, the Processor will store the character **x** in memory for that row. After all N rows have been read, the Processor will then take its next instruction from the address named in JY (parity error branch).

Tape feed (see PPT) will be ignored if it precedes data during any execution of the instruction. That is, *tape feeds* will be passed over without reading, and without counting, until some other configuration is found on the tape.

Tape feed is treated as a parity error if it occurs within the data.

RPT :CX This is an **extra** variation of the character mode, permitting use of *tape feed* as an end-of-item code. This special variation is identical with the Character mode, except when N = 1.

If N = 1, and if the first row on the tape is a *tape feed,* only that character is passed on the tape. Nothing is stored in memory, and the Processor takes its next instruction from the address named in (JY + 1).

RPT :S The 8 channels of each row on the tape are read into the RH 8 bit-positions of a **slab**, with the LH 4 bits of the slab set to 0 bits.

Tape feed is just another 8-bit configuration in this mode.

Y is not used, as this variation does not branch.

FEED a Punched Card and READ Columns
READ Columns from a Punched Card

400 card per minute reader

Op	V	L	X/Y	A/B
R C θ L	F		X	A
			Y	N

Op	V	L	X/Y	A/B
R C θ L			X	A
			Y	N

Jump is to address in JY.
Link in J14.

Punched cards are read serially, column by column, beginning with card-column 1. N columns of a card are read into memory, starting at the LH end of the A-area. If columns 81 and 82 have been punched, they are ignored.

N may be 000-159.

The image of each card column is stored as a bit-configuration in one slab of memory, with 1-bits corresponding to holes in the card. Card row 9 is stored as the RH bit of the slab; card row 12 is stored as the LH bit of the slab (see illustration on page 18). After reading, the program must translate each bit-configuration into the actual character. Standard subroutines and macro-instructions are furnished for this purpose.

It is customary to read and translate one column at a time, since there is sufficient time between card columns to perform the table-lookup while the card is passing.

When cards have been punched in the Direct mode, for communication between Processors, each card column contains an image of the bit-patterns which were in memory at the time of punching, and no translation is required when reading. In that situation, N may be made as large as convenient.

When the leading edge of each card reaches the reading station, the Card Reader will exercise Demand if the Processor's Demand Permit flag is on. Note that there is no Unit Demand flag in the Card Reader, and when reading cards, all other Unit Demand flags will usually be turned off.

RCθL:F Feed a card, read N columns.

If N=0, feed only; do not test for previous *missed column.*

If a FEED is issued after column 1 and before column 35 of the present card, then the Reader will remain in *continuous feed* and the next card will be fed with minimum interval between cards.

A FEED issued before column 1 can apply only to the present card. If this card has already been fed, the new FEED does nothing.

RCθL: Read N columns.

If N=0, this instruction does nothing.

JY:

Missed-column jump. Branch without execution. One or more columns of the present card have passed the reading station without being captured by an RCθL or RCθL:F instruction. The current instruction will not be executed, but will branch instead to the address named in JY.

Only the instruction RCθL:F with N=0 may be executed when the missed-column condition exists. This instruction will not branch; it *will* set the feed signal for the next card.

The missed-column condition is automatically reset when the trailing edge of the card passes the reading station. Thus reading may terminate when the desired portion of a card has been read, and resume with the next card without encountering the JY branch.

HOW TO USE THESE OPERATIONS:

The feeding and reading functions are, to some extent, independent of each other. When the Card Reader is at rest a FEED will, of course, cause a card to be fed past the reading station. However, once column 1 of this card has been read, the Reader will accept and store a signal to feed the next card as soon as this card has passed.

Thus, purely as an illustrative example, the following sequence would cause the entire contents of two cards to be read.

FEED and READ 1 column.

A card is fed; the Processor waits for column 1, reads it, and terminates the instruction.

The card continues to move past the reading station.

FEED and READ 159 columns.

The Reader accepts and stores the feed signal for the second card.

The Processor waits for column 2 of the first card, reads it, waits for column 3, reads it; and so on through column 80.

The Reader feeds the next card immediately.

The Processor waits for column 1, reads it; and so on through column 80. Then the instruction terminates.

However, this would be a very inefficient way to use the Processor, since all the time between columns, and between cards, would be wasted in waiting. A more efficient procedure, and the one actually used, is outlined:

- Feed and Read zero columns.
- Set Demand Permit flag.
- A Begin to execute some other program until the leading edge of the card reaches the reading station. The Reader will then exercise Demand.
- Read 1 column into working storage.
- Translate that column into a character, and store it in memory.
- Feed and read 1 column into working storage.
- Translate and store that column.
- Read 1 column into working storage.
- Translate and store that column.
- Repeat the previous two operations 77 more times, to complete the card.
- Set Demand Permit flag.
- Demand Link Return. Resume execution of the other program (point A in this outline).

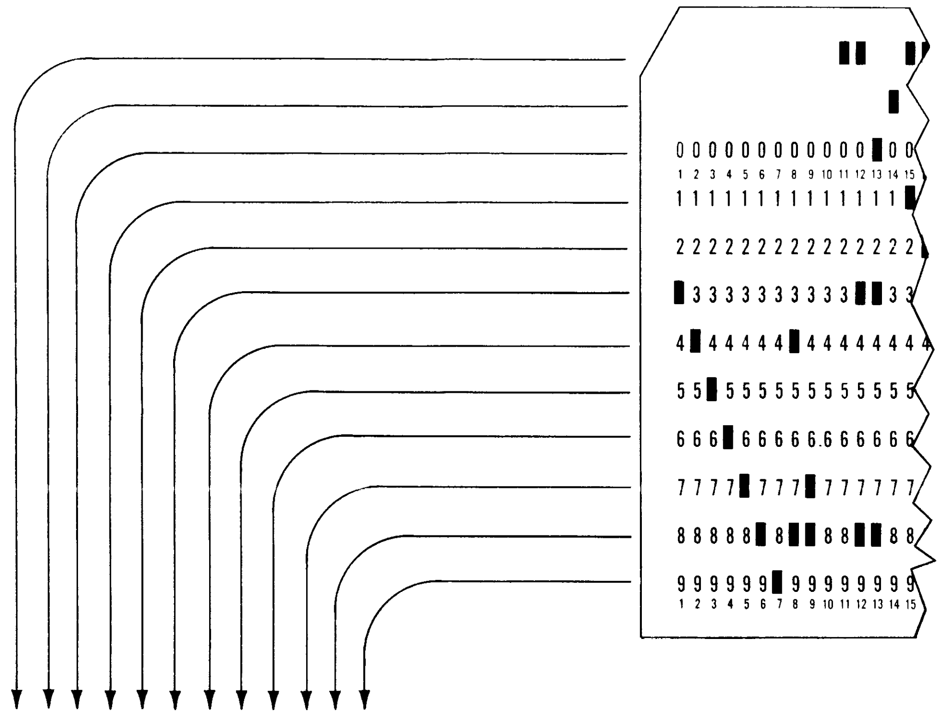
And so on for as many cards as will fit into the input area which the programmer has assigned.

Further time-sharing may be achieved if the input requires only the early columns of each card. The program stops the column-by-column reading after the last wanted column, and proceeds to other work.

The missed-column condition is set when the first unwanted column passes the reading station without being read. But the program never encounters this condition since there is no further attempt to read the current card, and the condition is automatically reset by the trailing edge of that card.

The leading edge of the next card causes the Reader to exercise Demand, informing the program that it is time to begin reading the columns of the next card.

The formation of the card-column image in memory is illustrated in the following example:



SUCCESSIVE SLABS OF MEMORY

Card Rows →	12	11	0	1	2	3	4	5	6	7	8	9	Card Columns
	0	0	0	0	0	1	0	0	0	0	0	0	1
	0	0	0	0	0	0	1	0	0	0	0	0	2
	0	0	0	0	0	0	0	1	0	0	0	0	3
	0	0	0	0	0	0	0	0	1	0	0	0	4
	0	0	0	0	0	0	0	0	0	1	0	0	5
	0	0	0	0	0	0	0	0	0	0	1	0	6
	0	0	0	0	0	0	0	0	0	0	0	1	7
	0	0	0	0	0	0	1	0	0	0	1	0	8
	0	0	0	0	0	0	0	0	0	1	1	0	9
	0	0	0	0	0	0	0	0	0	0	0	0	10
	1	0	0	0	0	0	0	0	0	0	0	0	11
	1	0	0	0	0	1	0	0	0	0	1	0	12
	0	0	1	0	0	1	0	0	0	0	1	0	13
	0	1	0	0	0	0	0	0	0	0	0	0	14
	1	0	0	1	0	0	0	0	0	0	0	0	15

Bit-patterns in memory

FEED a Punched Card and READ Columns
READ columns from a Punched Card

2000 card per minute reader
without **translation by the reader**

Op	V	L	X/Y	A/B
R C θ L	F		X	A
			Y	N

Op	V	L	X/Y	A/B
R C θ L			X	A
			Y	N

Jump is to address in JY.
Link in J14.

These two instructions work for the 2000 cpm reader exactly as they do for the 400 cpm reader, except that at the higher reading speed there is not time to perform any significant amount of processor work between columns.

FEED a Punched Card and READ Columns
READ Columns from a Punched Card

2000 card per minute reader
with **translation by the reader**

Op	V	L	X/Y	A/B
R C θ L	T F		X	A
			Y	N

Op	V	L	X/Y	A/B
R C θ L	T		X	A
			Y	N

Jump is to address in JY.
Link in J14.

When these two instructions are used with the 2000 cpm reader, they work just like RCθL:F and RCθL except:

- The reader translates each card column into a character.
- N is the number of **slabs** to be stored, two characters (two columns) to a slab.

If the reader detects a configuration in any column which does not correspond to one of the 64 characters it can translate, it will store the character **x** in the processor and set the Overflow flag.

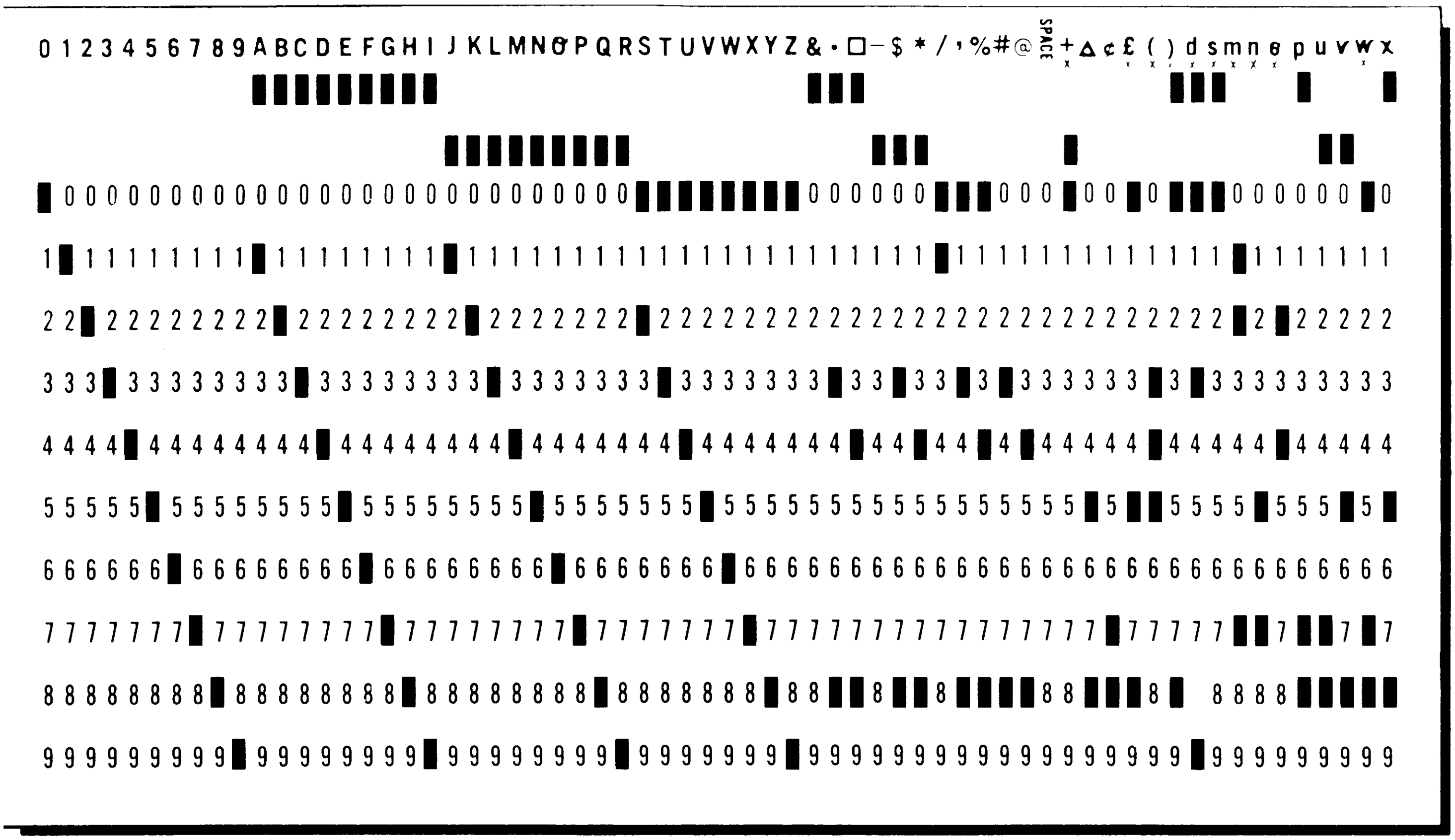
The program may switch from translate to non-translate and back as many times as desired within a card, if the card fields make it appropriate to do so. However, if an odd number of untranslated columns have been read before switching, the last untranslated column will appear a second time as a character.

Suppose the first half-dozen columns of a card contained **A B C D E F**, and that three columns were read untranslated, then two slabs with translation. Five slabs of memory would receive information; the first three would contain **A B C** as images of the columns, and the next two would contain **CD EF** as translated characters.

If these instructions are addressed to the 400 cpm reader, they will work just like RCθL:F and RCθL.

OVERFLOW: If, during the execution of either of these instructions, the reader found an unallowable configuration and substituted **x** for it.

PUNCHED CARD CODE



X WILL NOT PRINT ON I.B.M. 026 PRINTING KEY PUNCH

NOTE: The characters shown above, following "space", can be punched through a modification of the I.B.M. 026 Printing Key Punch, whereby these characters have been assigned to the number-shift position on the following keys:

KEY	**R E Q T A F D S G N H Z C V W X**
CHARACTER	**+ Δ ¢ £ ( ) d s m n θ p u v w x**

PUNCH a card

Op	V	L	X/Y	A/B
P N C H			X	A
			Y	S

S controls use of Selectors on punch plugboard.
Jump is to address in JY. Link in J14.

The instruction automatically addresses itself to whichever Punch was last selected by SELP. If a Printer was selected last, this instruction will be interpreted as PRNT and will operate the selected Printer.

The 40-slab or 80-slab A-area, containing the edited information to be punched, is transmitted to the selected Card Punch Buffer, together with the Selector control. This terminates the instruction within the Processor, and the program immediately resumes.

Under its own independent control, the Card Punch Buffer operates the Punch and punches one card. When it has finished this task, it reports READY.

The control panel on the Card Punch Buffer includes a 2-position switch, labelled TRANSLATE and DIRECT.

When the switch is in the T position, the Processor transmits 40 slabs (80 characters) and the card is punched in the *Character* mode, one alpha to a column, with each alpha automatically translated into conventional card-code. The translation is performed by an encoding matrix in the Card Punch Buffer, and does not affect the timing of the operation in any way.

When the switch is in the D position, the Processor transmits 80 slabs (160 characters) and the card is punched in the *slab* mode, one slab to a column, with each column punched as an exact image of the bit-pattern in the slab: holes correspond to 1-bits. The correspondence between bit-positions in memory, and rows in the card column, is the same as in RCØL; subsequent reading of the card without translation will duplicate memory as it was when the card was punched.

SELECTOR CONTROL:

Optional with the Model 523 punch, and standard with the Model 544 punch, are three pairs of selector hubs on the plugboard, marked 1, 2, 3. The 3 RH bits of S are regarded as correspondingly numbered:

Bits of S
x 3 2 1

and each bit-position controls the correspondingly-marked pair of selector hubs. Thus up to 8 different combinations may be programmed, corresponding to 8 plugboard operations.

JY:

This instruction will abort if the *previous* card on this Punch could not be punched. The information for that previous card remains in the buffer. Instead of executing *this* instruction, the Processor will take its next instruction from the address named in JY.

The following conditions will prevent punching of a card. Each of them causes the Punch to become READY, even though the card has not been punched.

- Power has gone off in this Punch.
  The next card cannot be punched, and the PNCH instruction after that will abort.
- Card misfeed or card jam.
  A misfed card cannot be punched at all; a jammed card cannot be punched completely. The next PNCH instruction will abort.
- Input hopper empty or output stacker full.
  The next card cannot be punched, and the PNCH instruction after that will abort.
- Plugboard programmed halt, such as detection of double-punch or blank-column.
  Card "1" contains a punching error;
  Card "2" is being punched while "1" is being checked. After "2" is completely punched, the Punch halts because of the error in "1".
  Card "3" cannot be punched.
  The PNCH instruction for card "4" will abort.
  The error-card "1" will be the last in the output stacker. Card "2" will be in the read-check station.

When a PNCH instruction aborts and takes the JY branch, the Punch becomes NOT READY, and remains in that state until the condition is corrected by the operator. He must clear the buffer, usually by punching its contents into a card; operator controls are provided for this purpose.

READY:

A Punch is READY when it has completed the punching of a card, or when it has failed to punch a card and has set the branch indicator.

It is NOT READY while it is punching a card, or after the Processor detects a branch condition and resets the branch indicator.

Ordinarily the program will test the READY state by means of SELP before issuing PNCH However, if PNCH is issued to a Punch which is NOT READY, the Processor will wait until the Punch becomes READY.

DEMAND INTERRUPT:

Whenever a Punch is READY, it will exercise Demand if its own Unit Demand flag, and the Processor's Demand Permit flag, are both on.

HALT and accept Console input

Op	V	L	X/Y	A/B
H A L T	(V)		X	A
			0 0	

The second line of the instruction format is not used. It is suggested that zeros be entered in the Y column to indicate on a coding sheet that this is a double-stage instruction.

Execution of the program is halted, and any information entered on the Console Keyboard will be stored in memory, starting at the LH end of the A-area. As many slabs as the operator desires may be entered.

HALT:D Store input in memory as Digits. If a non-digit character is entered, the Processor will ignore that character, and error-halt.

HALT:A Store input in memory as Alphas.

Before entering any information, or after pressing REST after some information has been entered, the operator may press the ALPHA or DIGIT button to change the mode of input, if desired.

Keyboard entries are held in a 1-slab input register, which is automatically putaway into successive slabs of memory each time it is filled with 2 Alphas or 3 Digits.

Execution of the program may be resumed by pressing the COMPUTE button on the keyboard. However, this button is inoperative unless the input register is empty. Pressing the REST button will clear the input register without putaway, and will permit use of the COMPUTE button, or any other Console operation.

ERROR HALT: If a non-Digit character is typed while in HALT:D. The character is not entered.

PRINT

Op	V	L	X/Y	A/B
P R N T			X	A
			Y	M F

M is Mode.
F is Vertical Format control.
JY is beginning of 2-way Jump Table.
Link in J14.

The instruction automatically addresses itself to whichever Printer was last selected by SELP. If a Card Punch was selected last, this instruction will be interpreted as PNCH and will operate the selected Card Punch.

The 60-slab A-area, containing the edited information to be printed, is transmitted to the selected Printer's Buffer together with the Mode and the Vertical Format Control. This terminates the instruction within the processor, and the program immediately resumes.

Under its own independent control, the Printer moves the paper as specified by M and F, then prints the line. When it has finished this task, it reports READY.

MODE:

	OUT-OF-PAPER JUMP (JY+1)	
	Print the line	Do not print the line
F is number of lines	0	2
F is recognition code	1	3

If M is 0 or 2, then F is the actual number of lines of paper to be moved before printing this line. If F is a non-decimal digit, then:

F	means
@	10
,	11
space	12
&	13
.	14
-	15

F=0 means do not move paper at all; this will cause this line to be overprinted on the previously printed line.

If M is 1 or 3, paper is moved until a punched configuration on the VFU Tape (see below) exactly matches the bit-configuration of F, where holes in the tape correspond to 1-bits of F. This line is then printed at the point on the paper opposite that configuration on the tape.

If F=0 the paper will stop at the first unpunched position on the VFU Tape.

If there is no VFU configuration which matches F, the Printer moves paper until the VFU "special" code (4 holes) is reached the *second* time. Paper movement then stops, and the Printer error-halts.

NON-DATA CHARACTERS:

The 8 characters m n ø p u v w x are called "non-data characters" since they are merely convenient names for their respective binary configurations, and will not appear in data. However, they may appear in programs, and when printing program listings the Printer will be called upon to print them.

When any non-data character is transmitted to the Printer, it will appear on the page as the corresponding capital letter, overprinted with a *plus* sign.

JY (end of page):

Paper movement for the *previous* line on this Printer reached or passed a "special" code (4 holes) on the VFU Tape, usually marking the end of the page. *This* line will not be accepted by the Printer, but the "special" indication will be cleared so that subsequent lines transmitted to the Printer will not encounter it. The Printer remains in the READY state.

JY+1 (out of paper):

Paper movement for the *previous* line on this Printer reached or passed a "special" code (4 holes) on the **last** sheet of paper. *This* line will be printed or not, according to the Mode of this instruction, but the branch is always taken, and the Printer becomes NOT READY. It remains in that state until the operator loads more paper.

VERTICAL FORMAT UNIT (VFU):

This unit, which is part of the Printer, holds a loop of paper tape into which 15 different code-configurations can be punched. The loop is the same length as the form being printed, and the punched codes exactly correspond to the lines which they designate.

As the paper moves vertically, the VFU, and its tape loop, move with it so that all paper movement is controlled by the VFU tape as specified by M and F in the instruction.

LISTER ATTACHMENT:

A Lister attachment is available, and may be attached or removed by the operator at any time. When the Lister is attached, three separate 30-column lists may be printed, with independent paper-advance for each of them.

The lists occupy, respectively:

	Columns in the print line	Slabs in the A-area	"Advance" values for F
Left	1 thru 30	0 thru 14	4
Middle	45 thru 74	22 thru 36	2
Right	91 thru 120	45 thru 59	1

When the Lister is attached, the Printer still accepts, stores, and prints the entire line. Therefore the programmer must see to it that all the character-positions between the lists are always *spaces.* At any point in the program it may be appropriate to print one, two, or all three lists; in the print line, the programmer fills with *spaces* the 15 words corresponding to any list which is not to be printed at the moment.

Attaching the Lister automatically disconnects the VFU. The only paper movement then is a 1-line advance, before printing, selectively for each list or combination of lists. The paper-advance is controlled by F, which is the sum (from 0 to 7) of the "advance" values of those lists which are to be advanced.

When the Lister is attached, the Mode of the instruction is not used. When the Printer reaches a point four inches from the physical end of the paper, the next line will not be printed, and *that* instruction will take the JY+1 branch.

READY:

A Printer is READY when it has advanced the paper and completed the printing of a line unless an "out of paper" condition exists.

Ordinarily the program will test the READY state by means of SELP before issuing PRNT. However, if PRNT is issued to a Printer which is NOT READY, the Processor will wait until the Printer becomes READY.

DEMAND INTERRUPT:

Whenever a Printer is READY, it will exercise Demand if its own Unit Demand Flag, and the Processor's Demand Permit Flag, are both on.

TYPE on Console Typewriter

Op	V	L	X/Y	A/B
T Y P E	(V)		X	A
				N

Type N slabs from memory, starting with the LH end of the A-area. N may be 000-999.

TYPE:D Type digits.

TYPE:A Type alphas.

There is no format control in these modes, and the Console Typewriter is not provided with an automatic carriage-return.

TYPE:AP Type alphas with programmed format.

Whenever the characters **w** and **x** appear in the A-word, the Processor automatically substitutes the format-control operations "tab" and "carriage return" respectively.

w becomes Tab

x becomes Carriage Return

If N=0, these operations do nothing.

READ MAGNETIC TAPE

Op	V	L	X/Y	A/B
R M T			X	A
			Y	I

JY is the beginning of a 6-way Jump Table used for all Tape operations on a given file. RMT uses only four of the jumps. Link in J14.

All or part of one record on magnetic tape is read into the A-area of memory. The information is automatically checked for 2-dimensional parity while the reading is being performed. Checking is for odd-parity at 60 kc and 40 kc, and for even-parity at 24 kc.

If the record contains an odd number of characters (as can occur with IBM-recorded tapes) a zero is automatically stored in memory to complete the last slab of the record.

I: Actual address (000-999) in memory of 3-slab Information Table, containing Handler-number and number of slabs to be read from the block.

I	I+1	I+2
Ha	← N →	

Ha: Handler-number: 0-7
only RH 3 bits are used

N: Number of slabs: 0000-7999
only RH 15 bits are used

Since the Handler-number is referenced in the instruction, there is no explicit operation to select the Handler.

If the record is more than N slabs long, the first N slabs of the record are stored in memory, while the rest of the record is used only for checking but is not stored in memory. The entire record is always checked, and therefore the time to perform a partial read is the same as to perform a complete read.

If the record is N slabs long, or less, the entire record is stored in memory.

If N=0, the tape does not move; the instruction merely tests for BUSY and LOCKOUT.

RECORD LENGTH:

If the records in a given file are not uniform in length, then by convention the record-length must appear as the first two slabs of the record.

INDEX:

Indexing a tape forward is accomplished by Reading with N = 1.

CRM:

Control Record Mark. Any 1-slab record serves as a CRM.

Conventions have been established for a number of CRMs, as listed. In most cases the programmer need be aware of these only to be sure that he never uses one of them for a CRM which he may record himself.

TT End of Tape
FF End of File
ƏƏ Skip Record follows
HH Hash-totals follow
CC Memory Dump follows

BRANCHES:

	READ	WRITE	BACK	WIND
JY	Read Error			
JY+1		Write Error		
JY+2	CRM			
JY+3		DWM		
JY+4	Busy Rewinding			
JY+5	Use Lockout or Write Lockout			

The complete Jump Table is shown, since the same table will normally be used for all operations on the same file.

Priority of branches:
- Use Lockout
- Busy
- Write Lockout
- Read or Write Error
- CRM or DWM (Destination Warning Marker)

If two branch conditions arise simultaneously, the Overflow flag (see below) is set.

Since all error-detection, end-of-tape alternation, etc., are handled by **STEP** (Standard Tape Executive Program), and a single Magnetic Tape Jump Table for the entire program is constructed by ***NEAT*** (National's Electronic Autocoding Technique), the programmer normally remains unaware of the branch conditions. He provides only for end-of-file CRM and for any special purposes for which he may choose to use CRMs.

READ ERROR:

When the Processor detects a read-error, it still reads the full N slabs or the entire record, whichever is less, into memory.

LOCKOUT:

One of the modes of REWIND will place a Use Lockout on the Handler. The Processor may not use this Handler again until the Use Lockout has been cleared by the operator, usually when he changes reels of tape.

R30: When the operation is complete, R30 contains the address of the first memory slab *following* the information read.

If N = 0, or if unable to read because of Busy or Lockout, (R30) remains unchanged.

OVERFLOW: If the Overflow flag is on, this instruction turns it off before the operation begins.

If the operation encounters a Read Error and a CRM simultaneously, it takes the Read Error branch and sets Overflow.

ERROR HALT: Attempting to execute a READ after a WRITE on the same tape. No tape movement.

Attempting to READ the Trailer (ie- past the physical end of the tape). No tape movement.

Attempting to READ blank tape. Tape runs to the Trailer, then error halts.

WRITE MAGNETIC TAPE

Op	V	L	X/Y	A/B
W M T			X	A
			Y	I

JY is the beginning of a 6-way Jump Table used for all Tape operations on a given file. WMT uses only four of the jumps. Link in J14.

One record is written on magnetic tape from the A-area of memory. The information has 2-dimensional parity automatically added to it, then it is immediately read back from the tape, and the parities checked. The operation uses odd-parity at 60 kc and 40 kc, and even-parity at 24 kc.

I: Actual address (000-999) in memory of 3-slab Information Table, containing Handler-number and number of slabs to be written as a record.

I	I+1	I+2
Ha	←	N →

Ha: Handler-number: 0-7
only RH 3 bits are used

N: Number of slabs: 0000-7999
only RH 15 bits are used

Since the Handler-number is referenced in the instruction, there is no explicit operation to select the Handler.

If N = 0, the tape does not move; the instruction merely tests for BUSY and LOCKOUT.

RECORD LENGTH:

If the records in a given file are not uniform in length, then by convention the record-length must appear as the first two slabs of the record.

CRM:

If N = 1, the resulting record is a CRM. See definitions and discussion under RMT.

DWM:

Destination Warning Marker. Approximately 18 feet from the physical end of the tape (the Trailer) the Handler will encounter the DWM, indicating that the end of the tape is approaching; it is safe to finish the current record, and to record (if desired) a few more records of control information, but no more file information should be written on this reel.

Once the tape passes the DWM, subsequent WMT instructions will not take this branch. If the programmer does any writing beyond the DWM, it is his responsibility not to write so much that he reaches the trailer.

The DWM is significant only when writing; it is ignored when reading.

BRANCHES:

(JY+1) Write Error.
(JY+3) DWM.
(JY+4) Handler busy rewinding.
(JY+5) Handler in Use Lockout or Write Lockout.

See complete table, and discussion, under RMT.

LOCKOUT:

One of the modes of REWIND will place a Use Lockout on the Handler. The Processor may not use this Handler again until the Use Lockout has been cleared by the operator, usually when he changes reels of tape.

Each Handler is normally in a Write Lockout state, making it impossible to destroy information on a tape by accidently recording over it. When it is determined that a given reel is to be recorded, the operator releases the Lockout on that Handler; as soon as the reel is recorded, rewound and changed, the Lockout is automatically established again.

OVERFLOW: If the Overflow flag is on, this instruction turns it off before the operation begins.

If the operation encounters a Write Error and a DWM simultaneously, it takes the Write Error branch and sets Overflow.

ERROR HALT: Attempting to WRITE when positioned on the Trailer. No tape movement.

BACKUP Magnetic Tape one Record

Op	V	L	X/Y	A/B
B A C K				
			Y	I

JY is the beginning of a 6-way Jump Table used for all Tape operations on a given file. BACK uses only two of the jumps. Link in J14.

The designated magnetic tape is moved backward one record, so that it is now positioned to re-read or re-write the last record which was read or written.

I: Actual address (000-999) in memory of 1-slab Information Table (first slab of 3-slab table used for RMT or WMT) containing Handler-number.

I	
	Ha

Ha: Handler-number: 0-7
only RH 3 bits are used

Since the Handler-number is referenced in the instruction, there is no explicit operation to select the Handler.

BRANCHES:

(JY+4) Handler busy rewinding.
(JY+5) Handler is in Use Lockout.
See complete table, and discussion, under RMT.

OVERFLOW: If the Overflow flag is on, this instruction turns it off before the operation begins.

This instruction never sets Overflow.

ERROR HALT: Tape positioned on Trailer, Leader, or BIM (Beginning of Information Marker).

REWIND Magnetic Tape

Op	V	L	X/Y	A/B
W I N D	(V)			
			Y	I

JY is the beginning of a 6-way Jump Table used for all Tape operations on a given file. WIND uses only two of the jumps. Link in J14.

WIND: Rewind the designated magnetic tape to the Beginning-of-Information Marker (BIM).

WIND:L Rewind, and set Use Lockout in the Handler.

I: Actual address (000-999) in memory of 1-slab Information Table (first slab of 3-slab table used for RMT or WMT) containing Handler-number.

I	
	Ha

Ha: Handler-number: 0-7
only RH 3 bits are used

Since the Handler-number is referenced in the instruction, there is no explicit operation to select the Handler.

BRANCHES:

(JY+4) Handler already busy rewinding.
(JY+5) Handler already in Use Lockout.
See complete table, and discussion, under RMT.

NOTE: If tape is already positioned on Leader or BIM, the operation does nothing.

OVERFLOW: If the Overflow flag is on, this instruction turns it off before the operation begins.

READ CRAM CARD

Op	V	L	X/Y	A/B
R C C			X	A
			Y	I

JY is the beginning of a 6-way Jump Table used for both Reading and Writing. RCC uses 4 of the jumps.
Link in J14.

The instruction automatically addresses itself to whichever CRAM was last selected by SELC, and reads one record from the card which is already on the drum.

Information is recorded on the card with one record in each of 7 tracks. Record length may be 1 to 1550 slabs.

All or part of one record on the card is read into the A-area of memory. The information is automatically checked for 2-dimensional parity while the reading is being performed.

I: Actual address (000-999) in memory of 3-slab Information Table containing track-number and number of slabs to be read from the record.

I	I+1	I+2
Tr	← N	→

Tr: Track-number: 0-6
only RH 3 bits are used

N: Number of slabs: 0000-1550
only RH 15 bits are used

If the record is more than N slabs long, the first N slabs of the record are stored in memory, while the rest of the record is discarded. See comments under PARTIAL READ.

If the record is N slabs long, or less, the entire record is read.

If Tr=7, or if N=0, the instruction does nothing.

If N is greater than 1550, the Processor error-halts.

BLOCK LENGTH:

If the records in a given file are not uniform in length, then by convention the record-length must appear as the first two slabs of the record.

CRM:

Control Record Mark. Any 1-slab record serves as a CRM. See remarks under Read Magnetic Tape for CRM conventions.

BRANCHES:

	READ	WRITE
JY	Read Error	
JY+1		Write Error
JY+2	CRM	
JY+3	Not Loaded	
JY+4	Wrong Card	
JY+5		Write Lockout

The complete Jump Table is shown, since the same table will be used for reading and writing.

Attempting to read a blank track (one which has never before been recorded) will cause the Processor to take the "wrong card" branch.

Priority of branches:
- Write Lockout
- Wrong Card
- Not Loaded
- Read or Write Error
- CRM

If two branch conditions arise simultaneously, the Overflow flag (see below) is set.

Since all error-detection and housekeeping are handled by **PACE** (PAckaged CRAM Executive), and a single CRAM Jump Table for the entire program is constructed by ***NEAT*** (National's Electronic Autocoding Technique), the programmer normally remains unaware of the branch conditions. He provides only for end-of-file CRM, and even that only when using CRAM Cards as a serial file.

PARTIAL READ:

If the record is more than N slabs long, reading terminates with the Nth slab, and the Processor immediately proceeds with the program without waiting for the rest of the record. In this case, while each character read into memory is checked for its parity bit, the record-parity character is not verified.

When writing a record which is expected to be subject to a Partial Read, it is customary to place an additional slab in the record at the point where the partial reading will terminate. This slab contains a programmed check-sum of that part of the record which precedes it, and the check-sum is verified after the partial read. Standard subroutines are provided for creating and verifying this check-sum.

READ ERROR:

When the Processor detects a read-error, it still reads the full N slabs or the entire record, whichever is less, into memory.

R30:	When the operation is complete, R30 contains the address of the first memory slab *following* the information read. If Tr=7, or N=0, or if unable to read because of NOT LOADED or WRONG CARD, (R30) remains unchanged.
OVERFLOW:	If the Overflow flag is on, this instruction turns it off before the operation begins. If the operation encounters a Read Error and a CRM simultaneously, it takes the Read Error branch and sets Overflow.

See next page for
WRITE CRAM CARD

WRITE CRAM CARD

Op	V	L	X/Y	A/B
W C C			X	A
			Y	I

JY is the beginning of a 6-way Jump Table used for both Reading and Writing. WCC uses 4 of the jumps.
Link in J14.

The instruction automatically addresses itself to whichever CRAM was last selected by SELC, and writes one block on the card which is already on the drum.

Information is recorded on the card with one record in each of 7 tracks. Record length may be 1 to 1550 slabs.

One record is written in the designated track from the A-area of memory. The information has 2-dimensional parity automatically added to it, and is immediately read back from the card, and the parities checked.

I: Actual address (000-999) in memory of 3-slab Information Table containing track-number and number of slabs to be written as the record.

I	I+1	I+2
Tr	N	

Tr: Track-number 0-6
only RH 3 bits are used

N: Number of slabs: 0000-1550
only RH 15 bits are used

After the record has been written, and the record-parity character recorded and checked, the Processor terminates the operation, and proceeds to the next instruction in the program. The CRAM, under its own independent control, then erases the rest of the track.

If Tr=7, or if N=0, the instruction does nothing.

If N is greater then 1550, the Processor error-halts.

RECORD LENGTH:

If the records in a given file are not uniform in length, then by convention the record-length must appear as the first two slabs of the record.

CRM:

If N=1, the resulting record is a CRM. See remarks under Read Magnetic Tape.

BRANCHES:

(JY+1) Write Error
(JY+3) Not Loaded
(JY+4) Wrong Card
(JY+5) Write Lockout

See complete table, and discussion, under RCC.

LOCKOUT:

Each CRAM is normally in a Write Lockout state, making it impossible to destroy information on a card by accidently recording over it. When it is determined that a deck of cards is permitted to have new information recorded on it, the operator releases the Lockout on that CRAM; as soon as the deck is removed, and a new deck is placed in the CRAM, the Lockout is automatically established again.

OVERFLOW: If the Overflow flag is on, this instruction turns it off before the operation begins.

This instruction never sets Overflow.