NCR HARDWARE MANUAL

1965

315 Electronic Data Processing System

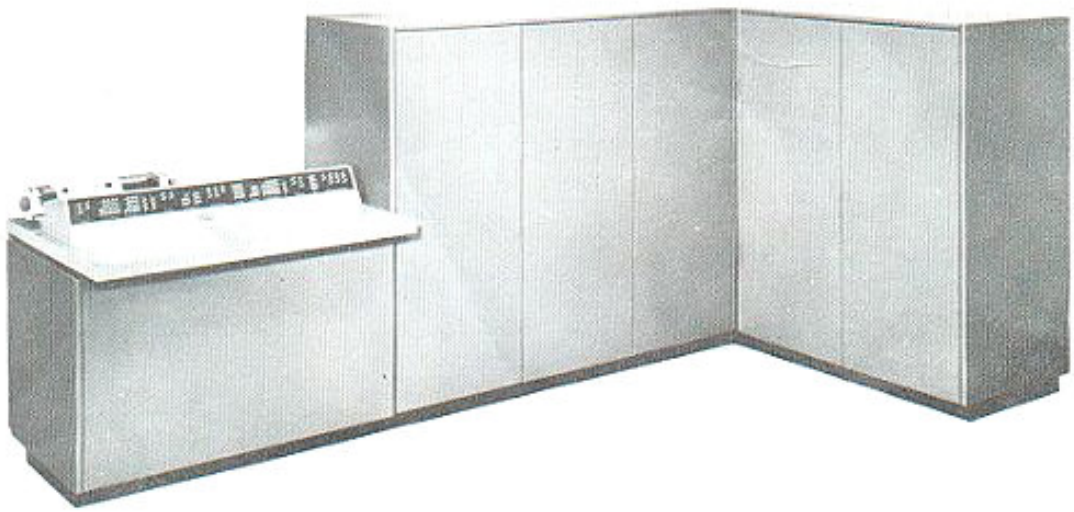
A Marketing Services Educational Publication

HARDWARE (Operating – Processor)

November 65 ST-5008-15

TITLE -- CENTRAL PROCESSOR (315, 315-100)

General Description



315 Data Processor

The 315 and 315-101 Central Processor contains the following elements:

- Central Console
- Control unit
- Arithmetic unit
- Memory section

Both numeric and alphabetic information may be handled by the Processor. The basic information unit is called a slab which consists of 12 bits plus a parity bit. The external Main Memory is available in many sizes as follows: 5,000, 10,000, 15,000, 20,000, 30,000 or 40,000 slabs.

The memory provides the storage for the program and is used for the temporary storage of data during computation and processing. The Processor executes instructions and controls the flow of data between memory and the associated peripheral units. No restriction is made on the assignment of memory locations for the program or data. Information is transferred to and from the memory by slab. The arithmetic unit also processes data by the complete slab.

The arithmetic unit called the Adder adds two 12-bit slabs in parallel. These two slabs may be added in one of the following 5 ways:

- Decimal Addition in groups of four bits, with carries between all bits and between the three groups of four bits.
- Decimal Subtraction in groups of four bits, with carries between all bits and between the groups of four bits.

- Binary Addition with carries over the full twelve bits.
- Binary Addition with carries over two groups of six bits, but no carry between the sixth and the seventh bits.
- Binary Subtraction with carries over the full twelve bits.

These are not necessarily direct command functions i.e., BADD, but are used by other commands such as compare, count, scan and register and address modification in any command.

An integral part of the Processor is a magnetic core memory referred to as the Auxiliary Memory. Containing special registers which are accessible by special commands and used for relative addressing procedures. Auxiliary memory consists of 64 registers that have 18 bits plus 1 parity bit; an accumulator of 8 slabs, each containing 12 bits plus parity.

In order to access 40,000 slabs of memory (00,000 to 39,999) an 18-bit address is required. This address consists of 4 full digits with 4 bits each, and an additional 2 bits for the most significant digit with a range of 0 through 5.

Instruction Format

The 315, 315-100 systems function primarily as single address processors. A few commands are capable of referring to two addresses at the same time for the purpose of transferring information from one main memory location to another. However, only one operand at a time is accessed by an arithmetic command.

For example, if the contents of memory address 100 are added to the contents of memory address 150, and the sum is stored in memory address 200, three commands must be executed.

- The LOAD command to load the contents of address 100 into the accumulator for temporary storage.
- The ADD command to add the contents of address 150 to the contents of the accumulator. The sum is retained in the accumulator.
- The STORE command to store the contents of the accumulator in address 200.

Addition of the two numbers is performed in the diode logic adder, but the accumulator provides temporary storage for the intermediate and final results.

NCR 315 CODE CHART

ZONE BITS	NUMERIC BITS															
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
00	0	1	2	3	4	5	6	7	8	9	@	,	☐	&	.	—
01	!	A	B	C	D	E	F	G	H	I	;	".	?	:	←	↑
10	+	J	K	L	M	N	©	P	Q	R	%	=	$	(	)	/
11	*	#	S	T	U	V	W	X	Y	Z	<	>	'	[	]	\

ABSOLUTE FORMAT	X xF C		A
	Y yQ G		B

PRINTER REPRESENTATION OF "NON-PRINTING" CHARACTERS	OVERPRINTED WITH A "+" SYMBOL							
	M	N	O	P	U	V	W	X
	?	:	←	↑	'	[	]	\

xF or yQ							
0	1	2	3	4	5	6	7
8	9	@	,	☐	&	.	—

X or Y	0	1	2	3	4	5	6	7	8	9	@	,	☐	&	.	—
REGISTER NUMBER	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31

*Trade Mark

Single Stage Instruction

Add Instruction ...

F	C	Q	G	OP CODE
1	3	-	-	03

Example ...

Add the contents of memory address 03 999 and 04 000 to the contents of the Accumulator and store result temporarily in the Accumulator.

IR 3 originally contained 03 000.

NEAT

OP V	L X	OPERAND A
ADD	02 03	999

ABSOLUTE

X OP	A
3 1 3	9 9 9

BCD [INTERNAL]

X × OP	A
0011 0 001 0011	1001 1001 1001

R# GR F C	A
IR 3 1 3	9 9 9

Double Stage Instruction

Count (CNT) Instruction...

F	C	Q	G	OP Code
1	0	0	0	10

Example...

Add 1 to IR7, then compare result with the value stored in memory locations 03 999 and 04 000.

IR3 originally contained 03 000

NEAT

OP	V	L	X	OPERAND A	B	Y
CNT			03	999	1	7

ABSOLUTE

X	OP	A	Y	B
3 , 1 , 0		9 , 9 , 9	7 , 0 , 0	0 , 0 , 1

BCD [INTERNAL]

X x	OP	A	Y y	Q	G	B
0011 , 0 , 001 , 0000		1001 , 1001 , 1001	0111 , 0 , 000 , 0000			0000 , 0000 , 0001

R#	GR	F	C	A	R#	GR	Q	G	B
IR 3 ,	1	, 0		9 , 9 , 9	IR 7 ,	0	, 0		0 , 0 , 1

NCR 315 INSTRUCTION GROUPS, MNEMONICS, CODES AND CORRESPONDING N-REGISTER SETTINGS

INSTRUCTION GROUP	OP/V MNEMONIC	CODE F C Q G	N-REG SETTING
LOAD	*LD:	1	0 1
STORE	*ST:	2	0 2
ADD	*ADD:	3	0 3
SUBTRACT	*SUB:	4	0 4
MULTIPLY	*MULT:	5	0 5
COMPARE	*COMP:	6	0 6
TEST	TEST: G	0 7	0 7
TEST	TEST: L	1 7	0 7
TEST	TEST: E	2 7	0 7
TEST	TEST: -	3 7	0 7
TEST	TEST: 0	4 7	0 7
TEST	JUMP:	5 7	0 7
TEST	TEST: D	6 7	0 7
TEST	TEST: T	7 7	0 7
TEST	**DIR	0 7	0 7
TEST	**SETF: +	1 7	0 7
TEST	**SETF: 0	2 7	0 7
TEST	**SETF: -	3 7	0 7
TEST	**SETF: D	6 7	0 7
TEST	**SETF: T	7 7	0 7
SHIFT	*SHFT: AR	0 8	1 0
SHIFT	*SHFT: DR	1 8	1 0
SHIFT	*SHFT: RR	2 8	1 0
SHIFT	*SHFT: DL	3 8	1 0
SHIFT	*SHFT: RC	4 8	1 0
SHIFT	*SHFT: LC	6 8	1 0
SHIFT	*SHFT: AL	7 8	1 0
ADD TO MEMORY	*ADD: M	9	1 1
BINARY ADD	*BADD:	@	1 2
DIVIDE	*DIV:	,	1 3
JUMP	TEST: LH	0 ∅	1 4
JUMP	TEST: RH	1 ∅	1 4
JUMP	SETF: LH	2 ∅	1 4
JUMP	SETF: RH	3 ∅	1 4
JUMP	CLRF: LH	4 ∅	1 4
JUMP	CLRF: RH	4 ∅	1 4

*R = 15 OR ≠ 15
**R = 15

INSTRUCTION GROUP	OP/V MNEMONIC	F C Q G	N-REG SETTING
JUMP	JUMP: I	6 ∅	1 4
JUMP	JUMP: IP	7 ∅	1 4
JUMP	**MLRA:	6 ∅	1 4
JUMP	**SKIP:	7 ∅	1 4
EDIT	EDIT:	&	1 5
SUPPRESS	SUPP:	.	1 6
COUNT	*CNT:	1 0	2 1
MODIFY	LD: R	2 0 0 0	2 2
MODIFY	LD: J	2 0 0 1	2 2
MODIFY	SLD: R	2 0 0 2	2 2
MODIFY	SLD: J	2 0 0 3	2 2
MODIFY	MOVE: RR	2 0 0 4	2 2
MODIFY	MOVE: JR	2 0 0 5	2 2
MODIFY	MOVE: RJ	2 0 0 6	2 2
MODIFY	MOVE: JJ	2 0 0 7	2 2
MODIFY	ST: R	2 0 0 8	2 2
MODIFY	ST: J	2 0 0 9	2 2
MODIFY	AUG: R	2 0 1 0	2 2
MODIFY	AUG: J	2 0 1 1	2 2
MODIFY	SAUG: R	2 0 1 2	2 2
MODIFY	SAUG: J	2 0 1 3	2 2
MOVE	MOVE: B	3 0 0	2 3
MOVE	MOVE: E	3 0 1	2 3
MOVE	**SPRD: B	3 0 0	2 3
MOVE	**SPRD: E	3 0 1	2 3
SCAN	SCND: G$\bar{V}$	4 0 $\bar{V}$ 1	2 4
SCAN	SCND: L$\bar{V}$	4 0 $\bar{V}$ 2	2 4
SCAN	SCND: E$\bar{V}$	4 0 $\bar{V}$ 4	2 4
SCAN	SCNA: G$\bar{N}$	4 0 $\bar{M}$ 9	2 4
SCAN	SCNA: L$\bar{N}$	4 0 $\bar{M}$ @	2 4
SCAN	SCNA: E$\bar{N}$	4 0 $\bar{M}$ ∅	2 4
PACK	LDAD:	5 0 0	2 5
PACK	LDAD: XR	5 0 1	2 5
PACK	LDAD: XL	5 0 2	2 5
PACK	LDAD: XB	5 0 3	2 5
		CONT'D	

$\bar{V}$ = 1-7
$\bar{N}$ = 1, 2, 3,
$\bar{M}$ = 1, 4, 5

NCR 315 INSTRUCTION GROUPS, MNEMONICS, CODES AND CORRESPONDING N-REGISTER SETTINGS.

INSTRUCTION GROUP	OP/V MNEMONIC	CODE F C Q G	N-REG SETTING
PUTAWAY	STDA;	6 0 0	2 6
PUTAWAY	PAST; XR	6 0 1	2 6
PUTAWAY	PAST; XL	6 0 2	2 6
PUTAWAY	PAST; XB	6 0 3	2 6
INTERROGATE	SELC; DN	7 0 0 0	2 7
INTERROGATE	SELC; DP	7 0 0 1	2 7
INTERROGATE	*SELC; R	7 0 0 2	2 7
INTERROGATE	*SELC; T	7 0 0 3	2 7
INTERROGATE	*TEST; SW	7 0 1 0	2 7
INTERROGATE	*SELP;	7 0 1 1	2 7
INTERROGATE	*SELS;	7 0 1 2	2 7
INTERROGATE	*SELQ;	7 0 1 3	2 7
INTERROGATE	*CLRU; C	7 0 2 0	2 7
INTERROGATE	*CLRU; P	7 0 2 1	2 7
INTERROGATE	*CLRU; S	7 0 2 2	2 7
INTERROGATE	*CLRU; Q	7 0 2 3	2 7
INTERROGATE	*SETU; C	7 0 3 0	2 7
INTERROGATE	*SETU; P	7 0 3 1	2 7
INTERROGATE	*SETU; S	7 0 3 2	2 7
INTERROGATE	*SETU; Q	7 0 3 3	2 7
PLACE	HALT; D	1 - 0	3 1
PLACE	HALT; A	1 - 1	3 1
PLACE	TYPE; D	1 - 2	3 1
PLACE	TYPE; A	1 - 3	3 1
PLACE	PPT; C	1 - 4	3 1
PLACE	PPT; S	1 - 5	3 1
PLACE	TYPE; AP	1 - 7	3 1
READER	RPT; S	2 - 1	3 2
READER	RPT; C	2 - 2	3 2
READER	RPT; CX	2 - 3	3 2
READER	RCOL; F	2 - 4	3 2
READER	RCOL;	2 - 5	3 2
READER	RCOL; TF	2 - 6	3 2
READER	RCOL; T	2 - 7	3 2

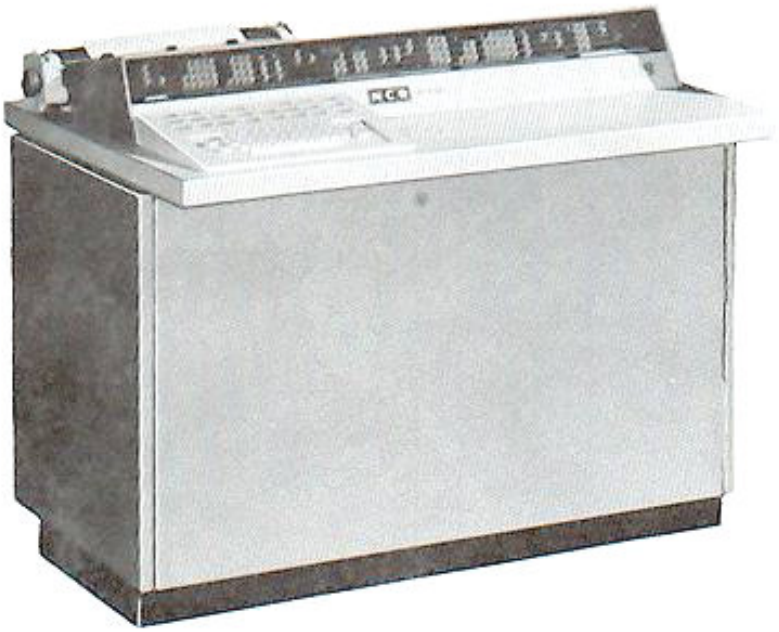
*R = 15 OR ≠ 15

INSTRUCTION GROUP	OP/V MNEMONIC	F C Q G	N-REG SETTING
BUFFER	STRT; S	3 - 0	3 3
BUFFER	RCK;	3 - 1	3 3
BUFFER	PKT;	3 - 2	3 3
BUFFER	STOP; S	3 - 3	3 3
BUFFER	PRNT;	3 - 4	3 3
BUFFER	PNCH;	3 - 4	3 3
MAG. TAPE	RMT;	4 - 0 $\bar{G}$	3 4
MAG. TAPE	WMT;	4 - 1 $\bar{G}$	3 4
MAG. TAPE	BACK;	4 - 2 $\bar{G}$	3 4
MAG. TAPE	WIND;	4 - 3 $\bar{G}$	3 4
MAG. TAPE	WIND; L	4 - 4 $\bar{G}$	3 4
MAG. CARD	RCC	5 - 0 $\bar{G}$	3 5
MAG. CARD	WCC	5 - 1 $\bar{G}$	3 5
INQUIRY	RQ	6 - 0	3 6
INQUIRY	WQ	6 - 1	3 6
INQUIRY			

$\bar{G}$ = "THOUSANDS" DIGIT FOR I

Operator Control Console

The Control Console is part of the Processor and gives the operator a means of controlling the 315 system. Illustrated below is the console which consists of a typewriter, control buttons, and indicators.



315 Control Console

Console Typewriter

The electric typewriter provides direct communication with the Processor memory. Data may be entered into the memory through the keyboard, or the Processor can transmit data to the typewriter for output through the typewriter printer. The console typewriter allows the operator to interrogate the memory, modify a program, and to note the various conditions of the program during a production run.

Control Keys

As shown below, the typewriter keyboard contains TAB, RETURN, and SHIFT keys. These keys affect the typewriter only, therefore no input is made to the Processor.

- The TAB key moves the carriage to a tabulator stop which had previously been set by the engineer or operator.

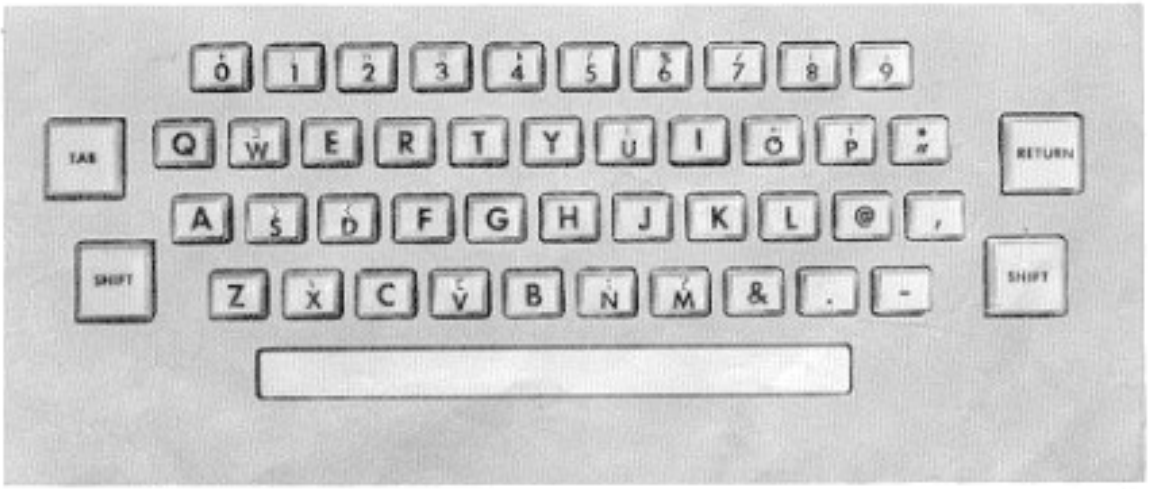
- The RETURN key causes the carriage to return to the left margin and advance the paper from one to three lines. The typewriter is often used by the programmer to type a message which indicates information to the operator. If the programmer has not included a carriage return in the coding, the operator must manually press RETURN. There is no automatic return at the end of a typed line.

- The two SHIFT keys determine whether the upper or lower position of the key is to be typed. For example, in the lower shift the character "6" will be printed; however, in the upper shift the character "¢" will be printed. When a message is typed from the Processor, the shift position is automatically determined.

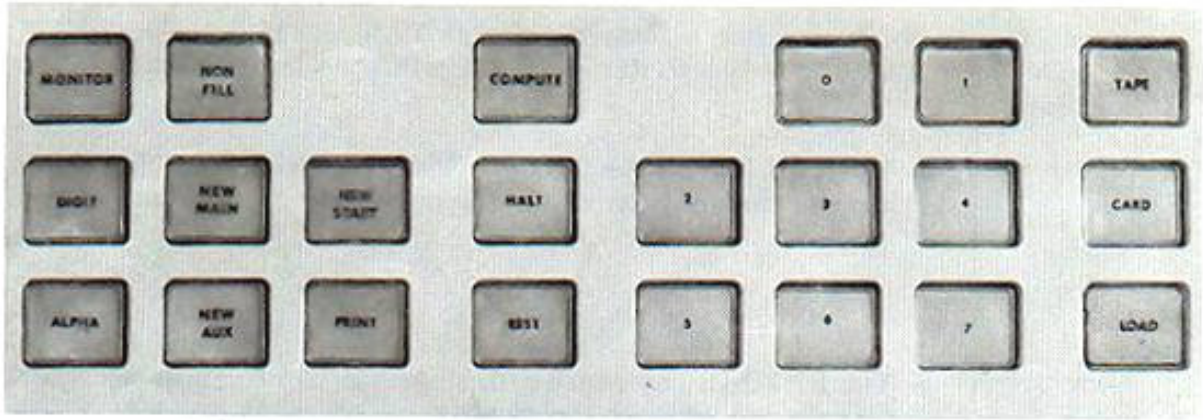
NOTE: When the typewriter is in the upper shift position only the keys where two terms (a symbol and an alpha, or a symbol and a digit) appear on one key, are operable.

Character Keys

There are 63 character keys and a space bar. When the space bar is pressed, the binary configuration 001100 is transmitted to the Processor. The paper in the typewriter will show only a blank space. The character keys permit addressing or accessing any location in memory, and the entry into the Processor may be any of the 63 six-bit characters.



Typewriter Keyboard



Console Control Buttons

Control Buttons

The Control Buttons are located above the typewriter keys. Press the button to set it ON. The button may be set OFF by pressing it again. The ON condition is indicated by the button being lighted.

- ON — Permits monitoring.
- OFF — Does not permit monitoring.

Monitoring is a technique which causes the program to print each instruction address and other pertinent data, such as the contents of the Accumulator and the contents of certain registers. In order to accomplish this, the program must have a sub-routine which indicates what is to be printed, a SETF:T instruction, and the address of the sub-routine in Jump Register 31.

- ON — Information that is typed on the console typewriter is entered in digit format. This means that the data is entered three characters per slab.

- ON — Information that is typed on the console typewriter is entered in alpha format. This means that the data is entered two characters per slab.

- ON — The console typewriter functions only as a standard typewriter. The information typed is not transmitted to memory. This may be used when it is necessary to add a comment to the log for later reference.
- OFF — This is the normal position, and all information which is typed is transmitted to memory.

* 315-100 - A system which utilizes ARØ will have an ARØ button in place of the NON-FILL button on the console.
315-RMC - This is the ARØ button on this processor... (standard equipment)

- ON — Main memory may be accessed by entering an address.

- ON — The auxiliary memory may be accessed. This memory consists of the Accumulator, the Index Registers and the Jump Registers.

- ON — The Processor is ready to receive a new starting address of the present program. This address will be placed in Index Register 31.

- ON — The typewriter will print data which was received from the Processor.

- ON — When the button is pressed, the program will begin computing at the address which is presently in Index Register 31.

- ON — The Processor halts upon completion of the current instruction.

- ON — The Processor is in the REST state. This means that the Processor is still operational, but awaiting action by the operator. After the Processor is halted, it must be placed in REST before any of the other buttons may be utilized.

- ON — These option switches exercise control when the conditions of the TEST:SW instruction in the object program are met.

- ON Input to the Processor is to be punched paper tape.

- ON Input to the Processor is to be punched cards.

- ON Conditions the Processor to be ready to read punched cards or punched paper tape when the COMPUTE button is pressed. To read a reel of punched paper tape, press TAPE, LOAD, COMPUTE. To read a deck of punched cards, press CARD, LOAD, COMPUTE.

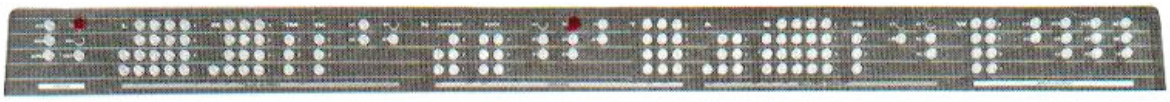
Processor ON and OFF buttons

- These buttons are located to the right of the console, and control the power supply to the Processor and Console.

Console Indicators

The Console Indicators display the status of the Processor. They are of visual aid during the REST or HALT stage. The console indicators are divided into the following elements:

- Status
- Main Memory
- Control
- Auxiliary Memory
- Input-Output



Console Indicators

Status Indicators

REST
- ON — The Processor is in the REST state. The indicator is illuminated when the REST button is pressed.

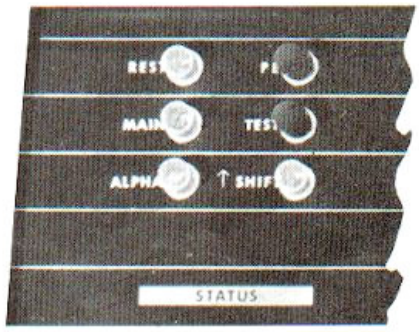
MAIN
- ON — Main memory may be accessed. The indicator is illuminated when the NEW MAIN button is pressed.

ALPHA
- ON — Information entered is in the alpha format. The indicator is illuminated when the ALPHA button is pressed.

PE
- ON — There is a Program Error.

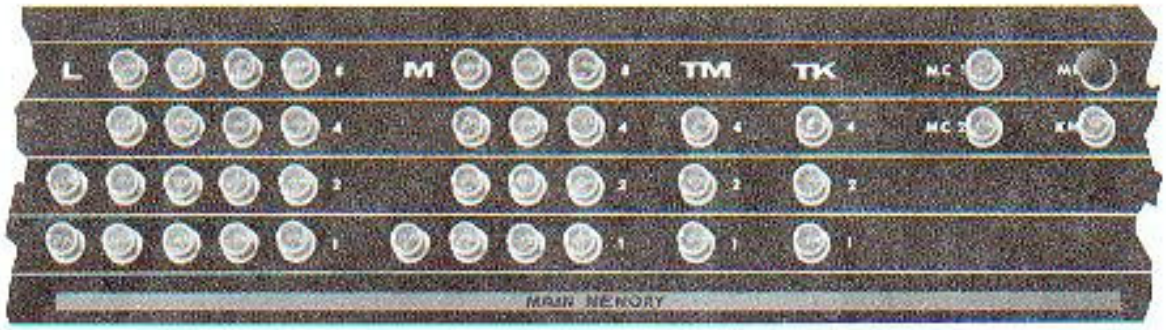
TEST
- ON — The Processor is in an engineering test mode and internally stored programs can not be executed.

SHIFT
- ON — The console typewriter is in the upper shift position. The indicator is illuminated when the SHIFT button is pressed.
- OFF — The console typewriter is in the lower shift position.



Status Indicators

Main Memory Indicators



Main Memory Indicators

L Indicators

The L Indicators display the next Main Memory address. (If stepping through the program with the halt button depressed the address of the next command in sequence is displayed). In case of a Halt instruction, the "L" register contains the contents of XA address (Fill address).
If Main Memory is being accessed, the L Indicators display the next address into which data is to be entered or the next address from which data is to be printed.

The indicators are read from left to right, one column at a time. Each column represents a decimal digit. The right column is the least significant digit.

- The address 29873 is displayed:

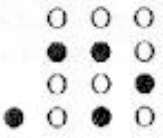
- The address 00806 is displayed:

M Indicators

The M Indicators display the contents of the current main memory address while the program is running. These indicators are not significant since they are set to zero when the program halts.

However, the M indicators are read from left to right, one column at a time. Each light represents a bit; the M-register lights store 12 bits which can be two alpha-numeric characters or three digits. The 13th left-most bit represents the parity bit which is stored automatically. In example C, two alpha characters (AB) are represented in binary form as AB = 010001 010010. Reduced further into digit format as in the M indicators, AB = 0100 0101 0010. Therefore, characters AB appear in the M-register as shown in example C.

EXAMPLE C PARITY --



TM Indicators

These indicators display the number of the peripheral unit which was selected in the program. If the program halts because of an incorrect peripheral setting, the operator may compare the unit number setting with the unit number displayed in the TM Indicators.

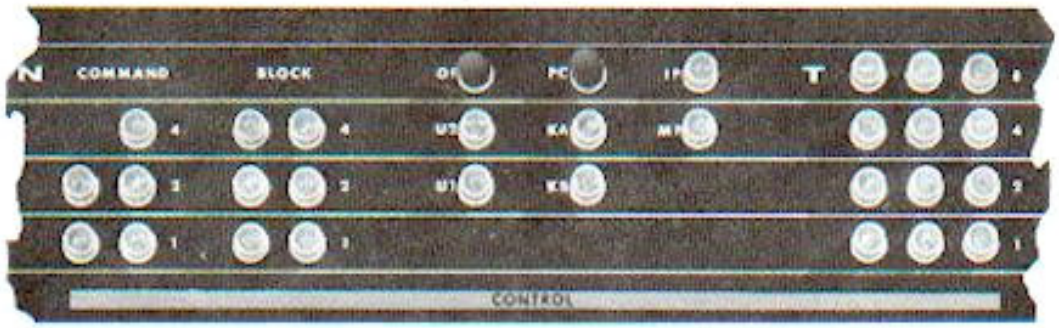
TK MC1 MC2 Indicators

These indicators are used for maintenance purposes only.

- ON — A slab having incorrect parity is read into or from main memory. Pressing REST causes the Processor to again become operational, but this procedure does not correct the parity error.

- ON — The result of the last arithmetic operation was negative.

Control Indicators



Control Indicators

The N COMMAND and N BLOCK Indicators are read from left to right, one column at a time. Each column represents an octal digit. The right column is the least significant digit. The number 7 is the maximum digit which can be displayed in octal format. For example:

Represents the number 3

Represents the number 7

Represents the number 10

N COMMAND

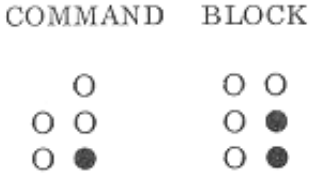
These indicators display the instruction in octal digit format. The operation Codes and their representations are on the following pages.

N BLOCK

These indicators display the next micro block of the instruction which is indicated by the N COMMAND indicators.

Single Stage Instruction

The F portion of a single stage instruction is not displayed in the indicators. For example; if the instruction is LOAD (an operation code of 1) and the micro block is 3, the indicators will appear as:

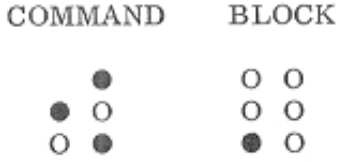


INSTRUCTIONS	OP CODES F	OP CODES C	CONTENTS OF N COMMAND INDICATORS
LOAD		1	01
STORE		2	02
ADD		3	03
SUBTRACT		4	04
MULTIPLY		5	05
COMPARE		6	06
TEST INSTRUCTIONS	0→7	7	07
SHIFT INSTRUCTIONS	0→7	8	10
ADD TO MEMORY		9	11
BINARY ADD		@	12
DIVIDE		,	13
JUMP INSTRUCTIONS	0→7	☐	14
EDIT		&	15
SUPPRESS		.	16

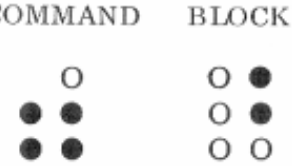
Single Stage Instructions

Double Stage Instruction

The Operation Codes zero (0) and hyphen (-) and the F portion of a double stage instruction are displayed in the N COMMAND Indicators. If the Operation Code is 0 it will appear as a 2 in the left column of the indicators. The value of the F portion appears in the right column. For example; if the instruction is LOAD ALPHA TO DIGIT (an operation code of 0 and an F of 5) and the micro block is 10, the indicators will appear as:



If the Operation Code is hyphen (-) it will appear as a 3 in the left column of the indicators. The value of the F portion appears in the right column. For example; if the instruction is PRNT (an operation code of - and an F of 3) and the micro block is 6, the indicators will appear as:



Listed here are the double stage instructions and their respective representations.

INSTRUCTIONS	OP CODES F	C	CONTENTS OF N COMMAND INDICATORS
COUNT	1	0	21
INDEX AND JUMP REGISTER MODIFICATION	2	0	22
MOVE-SPREAD	3	0	23
SCAN	4	0	24
LOAD ALPHA-TO-DIGIT	5	0	25
STORE DIGIT-TO-ALPHA PARTIAL ALPHA STORE	6	0	26
SELECT - TEST SWITCH SET AND CLEAR	7	0	27
HALT - TYPE - PUNCH PAPER TAPE	1	-	31
READ PAPER TAPE READ CARDS	2	-	32
START - READ - POCKET - STOP PRINT - PUNCH	3	-	33
MAGNETIC TAPE INSTRUCTIONS	4	-	34
CRAM INSTRUCTIONS	5	-	35
INQUIRY INSTRUCTIONS	6	-	36

Double Stage Instructions

- ON Overflow has occurred. This may be the result of the program instruction SETF:6 which sets the overflow flag ON, or it may be the result of an operation. The Processor will not halt unless the program specifies a halt.

- Various programming instructions or conditions may cause the Greater, Less, and Equal flags to be set ON. These indicators reflect the condition of these flags.

U2	U1	Flag Condition
OFF	OFF	The G flag is ON
OFF	ON	The L flag is ON
ON	OFF	The E flag is ON
ON	ON	No flag is ON

- ON A Processor Control Error has occurred. It may be an error in selecting the next micro block in sequence during the execution of an instruction, or a malfunction which was introduced by programming errors.

KA

- ON — A carry has resulted from an arithmetic operation of the internal adder.

IP

- ON — The Demand Permit flag has been set ON in the program through the use of the instruction SETF:D.

MP

- ON — The Monitor Permit flag has been set ON in the program through the use of the instruction SETF:T.

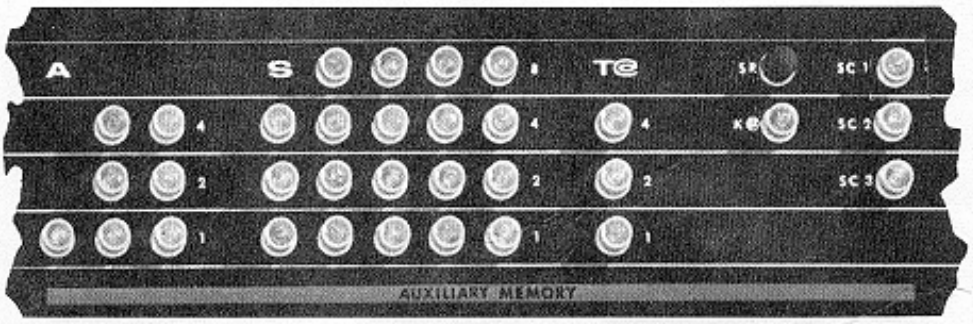
KB

- This indicator is used for maintenance purposes only.

T Indicators

These indicators are used for maintenance purposes only.

Auxiliary Memory Indicators



Auxiliary Memory Indicators

Auxiliary memory consists of the Index Registers, the Jump Registers, and the Accumulator. Any part of Auxiliary memory may be interrogated and/or changed from the console.

The thirty-two index registers are numbered 0 through 31. They are internally divided into two groups of sixteen registers, each group being designated as Group 0 or Group 1. The thirty-two jump registers are numbered in the same manner.

RELATIVE ADDRESS (INDEX) R—REGISTERS		JUMP J—REGISTERS	
0 40000	16 10000	0 STEP uses Registers 00 through 05	16 MICR Sorter-Reader uses Registers 16 through 19
1 01000	17 11000	1	17
2 02000	18 12000	2	18
3 03000	19 13000	3	19
4 04000	20 14000	4	20 Inquiry System uses Registers 20 through 22
5 05000	21 15000	5	21
6 06000	22 16000	6 CRAM EXECUTIVE uses Registers 06 through 11	22
7 07000	23 17000	7	23 ARO Link
8 08000	24 38000	8	24 (ARO 1)
9 09000	25 39000	9	25 (ARO 2)
10	26	10	26 (ARO 3)
11	27	11	27 Special Interrupt Control
12	28 Registers 28 and 29 are used by STEP, CRAM EXECUTIVE,	12	28
13	29 macros and sub-routines. Their contents are not preserved.	13	29
14	30 Processor stores an address (Next address)	14 Jump-Table Link (JY-Link)	30 Demand-Program Jump Control
15 Main Link (Program Decision)	31 Sequence-Control Register	15 Demand-Program Link (Auto-Link)	31 Tracer-Program Jump Control

*Class 407 uses 16 through 20

The table illustrated on the opposite page shows how index and jump registers may be accessed. For example.

The accessing addresses of Index Register 10 is @00.

- The @ represents the index register number in hexadecimal.
- The 0 represents Group 0 (i.e. the first fifteen index registers).
- The 0 means that this address relates to an index register.

The accessing address of Index Register 20 is 480.

- The 4 represents the fifth index register.
- The 8 signifies that this is Group 1.
- The 0 means that this address relates to an index register.

The accessing address of Jump Register 14 is .01.

- The . represents the jump register number in hexadecimal.
- The 0 represents Group 0 (i.e. the first fifteen jump registers).
- The 1 means that this address relates to a jump register.

A Indicators

The A Indicators display the Auxiliary Memory address octally. This may be the address where the program has halted; or if auxiliary memory is being accessed, the address into which data is to be entered or the address from which data is to be printed.

The indicators are read from left to right, one column at a time. Each column represents a decimal digit. The right column is the least significant digit.

The A Indicators show which index or jump register is being accessed In the previous example of accessing Index Register 20, the A indicators would display:

Effective Length	Address Typed	A Indicators
1 slab @	702	117 (least significant end)
2 slab @	602	116
3 slab @	502	115
4 slab @	402	114
5 slab @	302	113
6 slab @	202	112
7 slab @	102	111
8 slab @	002	110

Accumulator Addressing Table

	INDEX REGISTERS				JUMP REGISTERS				
	Group 0 (IR 0 thru 15)		Group 1 (IR 16 thru 31)		Group 0 (JR0 thru 15)		Group 1 (JR16 thru 31)		
	Access Address	A indi-cators	Access Address	A indi-cators	Access Address	A indi-cators	Access Address	A indi-cators	
0	000	000	080	020	001	040	081	060	16
1	100	001	180	021	101	041	181	061	17
2	200	002	280	022	201	042	281	062	18
3	300	003	380	023	301	043	381	063	19
4	400	004	480	024	401	044	481	064	20
5	500	005	480	025	501	045	581	065	21
6	600	006	680	026	601	046	681	066	22
7	700	007	780	027	701	047	781	067	23
8	800	010	880	030	801	050	881	070	24
9	900	011	980	031	901	051	981	071	25
10	@00	012	@80	032	@01	052	@81	072	26
11	,00	013	,80	033	,01	053	,81	073	27
12	☑00	014	☑80	034	☑01	054	☑81	074	28
13	&00	015	&80	035	&01	055	&81	075	29
14	.00	016	.80	036	.01	056	.81	076	30
15	-00	017	-80	037	-01	057	-81	077	31

Index and Jump Register, Addressing Tables

In the example of accessing Jump Register 14, the A Indicators would display:

The Accumulator is an eight slab working area which is used to hold the results of all arithmetic operations, and to store alpha or digit characters as a result of certain instructions. The effective length of the Accumulator is the left most non-zero slab of the Accumulator. For example, in the following instance, the effective length is five.

000	000	000	831	000	006	321	794

The illustration on the opposite page shows the relative positions of the Accumulator, the accessing addresses, and the digit representations in the A Indicators. If the third relative position of the previous example is to be typed, the accessing address is 502. The A Indicators would display:

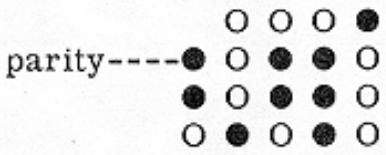
The console typewriter would print:

```
0 0 6,3 2 1,7 9 4 ← Typeout (digit)
115¦116¦117 ← Accumulator Address
```

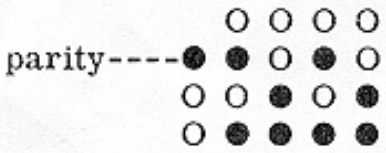
Note that the third position of the Accumulator contains 006. When entering information into the Accumulator, one slab is filled at a time until slab 117 is filled. When interrogating the Accumulator, the contents of the address accessed and the contents of the remaining slabs to the right are printed.

S Indicators

The S Indicators display the contents of the auxiliary memory address. The indicators are read from left to right, one column at a time. The right column is the least significant digit. The left column contains a parity bit. For example, if the contents of an index register were 21678, the S Indicators would display:



If the contents of a jump register were 05353, the S Indicators would display:



T@ [0–7]

- ON — The least significant digit of the assigned address of the first slab of the effective length of the Accumulator. Illustrated below is an Accumulator which has the effective length of three slabs. The assigned address of the third position of 115. The 5 (least significant digit) would be displayed in the T@ Indicator. To determine the effective length of the Accumulator, subtract the digit which is displayed from the number 8. The result is the effective length.

Relative position	8	7	6	5	4	3	2	1
Address	110	111	112	113	114	115	116	117

effective length

Applying the formula:

8 - T@= Effective Length
8 - 5 = 3

SR

- ON — Data having incorrect parity has been read from auxiliary memory.

K@

- ON — The sign of the Accumulator is negative.

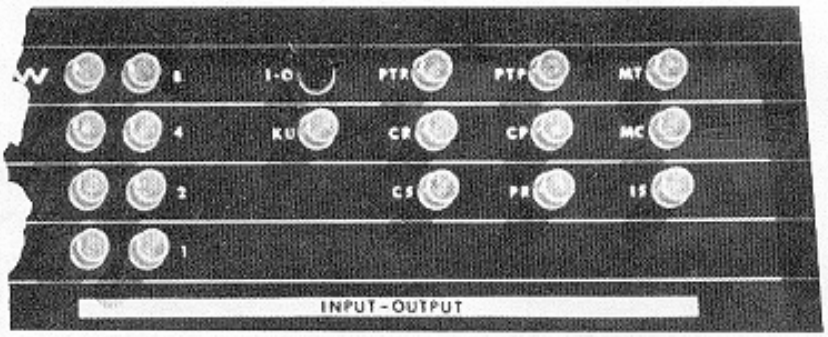
SC 1

- These indicators are used for maintenance purposes only.

SC2

SC 3

Input-Output Indicators



Input-Output Indicators

W Indicators (used on 315 only)

- ON — The indicators display data which has been transferred to or from peripheral units.

I-O

- ON — A buffered peripheral or CRAM unit has malfunctioned.

KU

- ON — A peripheral unit has its Unit Demand Flag ON, and is in the READY state.

PTR

- ON — Input is being received from the Paper Tape Reader.

CR

- ON — Input is being received from a Card Reader.

CS

- ON — Input is being received from a Document Sorter-Reader.

- ON — Data is being output to the Paper Tape Punch.

- ON — Data is being output to a Card Punch Buffer.

- ON — Data is being output to a Printer Buffer.

- ON — The Processor is sending or receiving data to or from a Magnetic Tape Handler.

- ON — The Processor is sending or receiving data to or from a CRAM Unit.

- ON — The Processor is sending or receiving data to or from an Inquiry Station.

Processor Hang-up and Halt Situations

27-05:*
Error Halt for literals or unspecified modes.

27-11:
Hang-up if more than one CRAM Unit assigned same unit number.

27-12:
Hang-up if more than one print/punch Buffer assigned same unit number.

27-13:
Hang-up if more than one Check Sorter Buffer assigned same unit number.

27-14:
Hang-up if more than one Inquiry Station Buffer assigned same unit number.

27-16:
Hang-up if Unit Interrupt Indication will not clear in selected buffer.

27-17:
Hang-up if Unit Interrupt Indicator will not set in selected buffer.

27-24:
Hang-up if Select Indicator will not clear.

27-25:
Hang-up if Select Indicator will not set.

27-31:
Hang-up if CRAM Select Indicator will not clear.

27-32:
Hang-up if CRAM Select Indicator will not set.

27-35:
Hang-up if CRAM partially dropped card (hanger).

27-36:
Hang-up if CRAM partially dropped card (hanger).

27-45:
Hang-up if CRAM card not released from drum.

*Command (N-Register) and Block number associated with Processor Hang-up and Halt situations.

31-04:
Error Halt if literal or unspecified modes.

31-12:
Hang-up waiting for go-ahead from either the output-writer or paper tape punch (allow for mechanical sync).

31-13:
Hang-up waiting for go-ahead from output-writer (allow for mechanical sync).

31-15:
Same as Block 13.

31-16:
Hang-up waiting for go-ahead from output-writer.

31-24:
Hang-up waiting for go-ahead from output-writer.

32:04:
Error Halt if literal or mode not specified.

32-07:
Hang-up waiting for clock (generated by sprocket hole) from paper tape reader.

32-10:
Hang-up, can't read data - tape reader malfunction or photo cell out (PTR).

32-11:
Hang-up, can't read data (PTR).

32-12:
Hang-up if tape malfunction or photo cell out (PTR).

32-32:
Hang-up waiting for data clock from card reader.

32-33:
Hang-up, can't read data - card reader malfunction or photo cell out (CR).

33-04:
Error Halt for literal and undefined modes.

33-05:
Error Halt if no unit selected.

33-07:
Hang-up if Check Sorter ready line not true.

33-11:
Hang if Check Sorter buffer busy.

33-12:
Error Halt if short fill (I/O).

33-13:
Same as Block 12.

33-14:
Error Halt if long fill (I/O).

33-15:
Hang-up waiting for Check Sorter branch digit echo.

33-25:
Hang-up waiting for Check Sorter sort digit received echo.

33-26:
Hang-up waiting for Check Sorter feed (start or stop) echo.

33-27:
Hang-up if printing/punch buffer not ready.

33-47:
Hang-up when out of paper.

33-51:
Hang-up waiting for Top-of-Page indicator to be cleared or there is a card punch error.

34-04:
Error Halt if literal or unspecified mode.

34-05:
Hang-up if more than one handler on same unit number.

34-07:
Hang-up, wait for clear select in unit.

34-10:
Hang-up, wait for selection in desired unit.

34-11:
Hang-up if tape busy rewinding.

34-17:
Error Halt if,
1) Write status set and read command issued.
2) Tape positioned on trailer.

34-24:
Error Halt if tape positioned on trailer.

34-25:
Hang-up, wait for Write Status to be set.

34-30:
Hang-up, wait for beginning of information marker to be seen.

34-34:
Hang-up, wait to write check sum.

34-35:
Hang-up, waiting for inter-block gap to be erased.

34-36:
Hang-up, wait for beginning of information marker to be seen.

34-52:
Hang-up, until complete block is read.

34-53:
Hang-up, wait for sum check to be read and checked.

34-64:
Hang-up, waiting for tape to get up to speed (forward).

34-66:
Error Halt if on Load Point or Leader, and command is Index Backwards.

34-70:
Hang-up, waiting for tape to get up to speed (reverse).

34-71:
Hang-up, wait for first character when reversing.

34-72:
Hang-up, wait until you have backed up to the beginning of the block.

34-73:
Hang-up, wait until unit starts to rewind.

34-74:
Hang-up, wait until Use Lockout Set.

35-04:
Error Halt for unspecified modes.

35-05:
Error Halt if unit not selected.

35-10:
Hang-up, wait for channel No. echo.

35-12:
Error Halt if 'N' is greater than 1999.

35-13:
Error Halt if three least significant digits of 'N' are > 550.

35-25:
Hang-up, waiting to go ahead for read or write operation.

35-27:
Hang-up, waiting for lead edge of card to be erased before write operation can begin.

35-33:
Hang-up, waiting to write check sum.

35-35:
Hang-up, wait for first character written to reach read head.

35-36:
Hang-up, wait for lead edge of card to pass read head.

35-52:
Hang-up, wait until all characters are read including the check sum.