

**NCR**

---

**NCR Century  
Handbook for  
Systems Analysts**

---

**NCR**

WORLDWIDE HEADQUARTERS  
DAYTON, OHIO 45479

PRINTED  
IN  
U.S.A. SP-1425 0677

Newer version of handbook (0677) minor changes from 1074

T. Peavey - ok!

**NCR**

# NCR Century Handbook for Systems Analysts



**NCR**

NCR Limited

206 Marylebone Road  
London NW1 6LY  
01-723 7070

© 1976

DC/507

UK version (same as US 0677) - Note coffee cup stain!!

I. SIMPSON

TABLE OF CONTENTS

LEGEND: (For Control Characters in USASI Code Set)

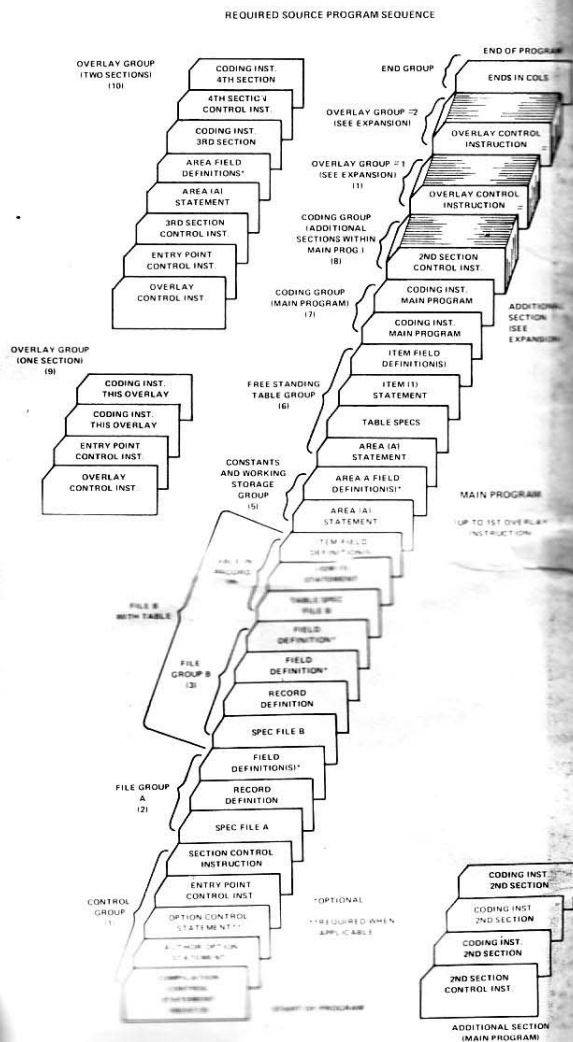
NUL Null  
 SOH Start of Heading  
 STX Start of Text  
 ETX End of Text  
 EOT End of Transmission  
 ENQ Enquiry  
 ACK Acknowledge  
 BEL Bell (Audible or Attention Signal)  
 BS Backspace  
 HT Horizontal Tabulation (Punched Card Back)  
 LF Line Feed  
 VT Vertical Tabulation  
 FF Form Feed  
 CR Carriage Return  
 SO Shift Out  
 SHI Shift In  
 DLE Data Link Escape  
 DC1 Device Control 1  
 DC2 Device Control 2  
 DC3 Device Control 3  
 DC4 Device Control 4  
 NAK Negative Acknowledge  
 SYN Synchronous Idle (Form Code)  
 ETB End of Transmission Block  
 CAN Cancel (Form Code)  
 EM End of Message  
 SUB Substitute  
 ESC Escape  
 FS File Separator  
 GS Group Separator  
 RS Record Separator  
 US Unit Separator

**NCR CENTURY CODE CHART**

B <sub>4</sub> -B <sub>1</sub>	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1100	1101	1110	1111
B <sub>8</sub> -B <sub>5</sub>	0	1	2	3	4	5	6	7	8	9	A	B	C	D
	NUL	SOH	STX	ETX	DC3	DC4	ENQ	ACK	BEL	BS	LF	VT	FF	CR
	0	1	2	3	4	5	6	7	8	9	A	B	C	D
	DLE	DC1	DC2	#	\$	%	&	'	(	)	*	+	,	-
	0	1	2	3	4	5	6	7	8	9	A	B	C	D
	@	A	B	C	D	E	F	G	H	I	J	K	L	M
	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]
	^	a	b	c	d	e	f	g	h	i	j	k	l	m
	n	o	p	q	r	s	t	u	v	w	x	y	z	{

NCR Century Code Chart ..... Inside Front Cover  
 Organization of NEAT/3 Source Program ..... 2  
 NEAT/3 Instructions - Level 1 ..... 3  
 NEAT/3 Instructions - Level 2 ..... 13  
 NEAT/3 Systems Tags ..... 16  
 Flowrite Instructions ..... 18  
 Monitor Control Instructions ..... 18  
 Program Overlay Calls ..... 21  
 Loading COT Boots ..... 22  
 Error Start Procedures ..... 23  
 Memory Map - NCR Century 50/100 ..... 24  
 Memory Map - NCR Century 101/151/200/201 ..... 26  
 Memory Map - NCR Century 251/300 ..... 28  
 Hexadecimal and Decimal Conversion ..... 30  
 Hardware Command Format ..... 31  
 File Buffer Chain ..... 31  
 Common Section, File, Buffer, & Extremity Tables ..... 32  
 Patch Card Formats - OPURCARE ..... 33  
 Relocation Constants ..... 33  
 Pertinent Memory Locations ..... 34  
 Dynamic Dump Patches ..... 34  
 Supervisor Transfer Table ..... 35  
 Table Control Entry ..... 36  
 IOSET Macro ..... 36  
 Simulated Option Switch ..... 36  
 Maximum Length of Operands ..... 37  
 Editing Mask ..... 38  
 Printer Character Sets ..... 38  
 Printer Control Block ..... 39  
 Common Status Characters ..... 39  
 ANSI Paper Tape Code ..... 40  
 Symbolic Debug Formats ..... 41  
 Indexed Sequential Macros ..... 41  
 Random Filing System Macros ..... 42  
 Symbolic Unit Designators ..... 42  
 Monitor Flag Settings ..... 43  
 Peripheral Type Codes ..... 44  
 Data Format Codes ..... 46  
 Hexadecimal Op Codes ..... 47  
 Holerith Extended A Set ..... 50  
 Holerith Extended H Set ..... 51  
 File Specifications Worksheets ..... 52  
 Compiler Specifications Worksheets ..... 58  
 Miscellaneous Specifications Worksheets ..... 59  
 Sort Worksheets ..... 60  
 Random Filing System Worksheets ..... 62  
 Index ..... 63  
 Memory Dump Line Guide ..... Inside Back Cover

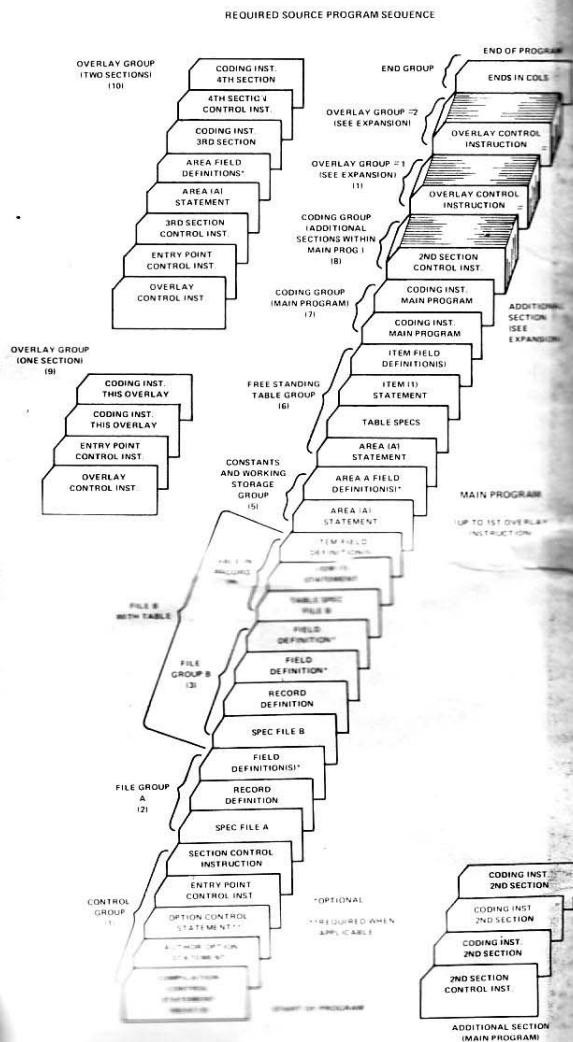
## ORGANIZATION OF NEAT/3 SOURCE PROGRAM



## NEAT/3 INSTRUCTIONS – LEVEL 1

Instr.	Operands	Description
ADD	A,B	(A) + (B) → B
ADD	A,B,C	(A) + (B) → C
ADDC	A,B,Z	If overflow, branches to Z.
ADDC	A,B,C,Z	
ADDL	A,B,Z	If overflow, links to Z.
ADDL	A,B,C,Z	
ADDR	A,B	Rounds off decimal places.
ADDR	A,B,C	
ADDRC	A,B,Z	Rounds off decimal places; if overflow, branches to Z.
ADDRC	A,B,C,Z	
ADDRL	A,B,Z	Rounds off decimal places; if overflow, links to Z.
ADDRL	A,B,C,Z	
BEGDBG		Establishes the point at which debugging begins.
BLKCHK	FR,A,Z	If block length ≥ (A), branches to Z.
BLKOUT	FR	Outputs a short block.
BR	Z	Branches unconditionally; stores no link.
BRE	Z	If E flag is on, branches to Z.
BRG	Z	If G flag is on, branches to Z.
BRL	Z	If L flag is on, branches to Z.
BRGE	Z	If G or E flag is on, branches to Z.
BRLE	Z	If L or E flag is on, branches to Z.
BRU	Z	If G or L flag is on, branches to Z.
BRDEP	A,B,C,Z	Branches to address in branch table B depending upon value of (A); if (A) points past end-of-table, branches to Z.
CALL	Z,AR1, AR2,..	Links to Z and sets up arguments AR1-AR9.
CALL	Z,A,AR1, AR2,..	Links to Z, saves index registers of calling module in A, sets up arguments AR1-AR9.

## ORGANIZATION OF NEAT/3 SOURCE PROGRAM



## NEAT/3 INSTRUCTIONS – LEVEL 1

Instr.	Operands	Description
ADD	A,B	(A) + (B) → B
ADD	A,B,C	(A) + (B) → C
ADDC	A,B,Z	If overflow, branches to Z.
ADDC	A,B,C,Z	
ADDL	A,B,Z	If overflow, links to Z.
ADDL	A,B,C,Z	
ADDR	A,B	Rounds off decimal places.
ADDR	A,B,C	
ADDRC	A,B,Z	Rounds off decimal places; if overflow, branches to Z.
ADDRC	A,B,C,Z	
ADDRL	A,B,Z	Rounds off decimal places; if overflow, links to Z.
ADDRL	A,B,C,Z	
BEGDBG		Establishes the point at which debugging begins.
BLKCHK	FR,A,Z	If block length ≥ (A), branches to Z.
BLKOUT	FR	Outputs a short block.
BR	Z	Branches unconditionally; stores no link.
BRE	Z	If E flag is on, branches to Z.
BRG	Z	If G flag is on, branches to Z.
BRL	Z	If L flag is on, branches to Z.
BRGE	Z	If G or E flag is on, branches to Z.
BRLE	Z	If L or E flag is on, branches to Z.
BRU	Z	If G or L flag is on, branches to Z.
BRDEP	A,B,C,Z	Branches to address in branch table B depending upon value of (A); if (A) points past end-of-table, branches to Z.
CALL	Z,AR1, AR2,..	Links to Z and sets up arguments AR1-AR9.
CALL	Z,A,AR1, AR2,..	Links to Z, saves index registers of calling module in A, sets up arguments AR1-AR9.

Instr.	Operands	Description
CLOSE	FR	Closes any file.
CLOSEO	FR	Closes and obsoletes a magnetic media file.
CLOSEN	FR	Closes a print file, does not eject page.
CLOSES	FR	Closes currently open section of file.
CLOSES	FR,A	Closes the magnetic tape file section named by (A).
CLOSET	FR,A	Closes all sections of a random file except those listed in table A.
CNSIN	A,B	Displays (A) to the operator; requires a hex input to B.
CNSINA	A,B	Displays (A) to the operator; requires an alpha input to B.
CNSOUT	A	Displays (A) to the operator.
COMP	A,B	Compares (A) to (B).
COMPE	A,B,Z	If (A) = (B), branches to Z.
COMPG	A,B,Z	If (A) > (B), branches to Z.
COMPGE	A,B,Z	If (A) ≥ (B), branches to Z.
COMPL	A,B,Z	If (A) < (B), branches to Z.
COMPLE	A,B,Z	If (A) ≤ (B), branches to Z.
COMPU	A,B,Z	If (A) ≠ (B), branches to Z.
CONC	A,B,C,Z	Moves record (A) to B, removing extraneous zeros and spaces beginning at C; links to Z on error. If C is not used, zeros and spaces are removed for entire record.
CONC	A,B,Z	
COND	A,B	Packs (A) → B; length of B must be one-half length of A.
COPYA	A	Copies entire source program A.
COPYA	A,B	Copies entire source program A from SUD B.
COPYP	A,B,C	Copies partial source program A; B = starting page/line, C = page/line following last to be copied or END\$.
COPYP	A,B,C,D	Copies partial source program from SUD D.
COPYR	A,B,C	Copies partial source program A; B = starting reference name, C = reference of statement following last to be copied or END\$.
COPYR	A,B,C,D	Copies partial source program from SUD D.

Instr.	Operands	Description
DCONC	A,B,C,Z	Moves record (A) to B, replacing extraneous zeros and spaces beginning at C; links to Z on error. If C is not used, zeros and spaces are removed for entire record.
DEFAULT	FR1,FR2,..	Closes current section of files FR1-FR4 and opens new section.
DELETE	FR	Removes current record from chained S/D file and closes up block.
DELETE	FR,WA	Removes current record from chained S/D file and closes up block; stores record in named workarea.
DIV	A,B,C	(B) ÷ (A) → C.
DIVC	A,B,C,Z	If overflow, branches to Z.
DIVL	A,B,C,Z	If overflow, links to Z.
DIVR	A,B,C	Rounds off decimal places.
DIVRC	A,B,C,Z	Rounds off decimal places; if overflow, branches to Z.
DIVRL	A,B,C,Z	Rounds off decimal places; if overflow, links to Z.
DISCOFF	A	Informs operator to remove disc pack referenced by A which may be either literal or a reference.
DUMP	A,B,C	Dumps memory from loc. A to loc. B for length of C. All operands are optional.
DYDUMP	A,B,C	
END\$	(col 1)	Indicates end of input data.
ENDDBG		Establishes the point at which debugging ends.
ENTER	A,AR1,AR2,..	Receives control and arguments from the calling module; restores contents of index registers stored in A.
ENTRY		Establishes entry point into program overlay.
FILECT	FR,A,B,Z	Accesses file table or directory entry; performs function A, using data (B), branches to Z on error.
FINISH		Returns control to monitor.

Instr.	Operands	Description
GET	FR	Presents records sequentially.
GET	FR,WA	Presents records sequentially; moves them to a workarea.
GETPAR	A,Z	Reads SPEC\$ card from WACS into A; transfers control to Z if no parameters are found, if STOPRD is read, or if disc read error occurs.
GLCOMP	A,B,C	Compares (A) to (B) for length (C).
GLMOVE	A,B,C,Z	Moves (A) to B for length specified by (C); branches to Z if the length of B is < (C).
IFAL	A,Z	If (A) is alphabetic, branches to Z.
IFNAL	A,Z	If (A) is not alphabetic, branches to Z.
IFNU	A,Z	If (A) is numeric, branches to Z.
IFNNU	A,Z	If (A) is not numeric, branches to Z.
INSERT	FR,WA	Places a record from workarea into current record position and turns WRITSP flag on.
LGET	FR	Reads a user-label from mag. tape.
LINK	Z	Transfers control to Z; stores address of next command in link list.
LINKE	Z	If E flag is on, links to Z.
LINKG	Z	If G flag is on, links to Z.
LINKGE	Z	If G or E flag is on, links to Z.
LINKL	Z	If L flag is on, links to Z.
LINKLE	Z	If L or E flag is on, links to Z.
LINKU	Z	If G or L flag is on, links to Z.
LOG	A	Places (A) into the log.
LPUT	FR	Places a user-label on magnetic tape.
MARK	FR,A	Stores in A the file location of the current record in memory.
Wedit (Not for NCR Century 50/100/101/151)	A,B,C	Moves (A) to B, editing the data using the literal edit mask C and the hardware Edit Command.

Instr.	Operands	Description
MOVE	A,B	Moves (A) to B.
MOVEVF	A,B,C,Z	Moves (A) to B for length specified in (C); branches to Z if the length of B is < (C).
MULT	A,B,C	(A) x (B) → C or (B) x (A) → C (smaller x larger).
MULTC	A,B,C,Z	If overflow, branches to Z.
MULTL	A,B,C,Z	If overflow, links to Z.
MULTR	A,B,C	Rounds off decimal places.
MULTRC	A,B,C,Z	Rounds off decimal places; if overflow, branches to Z.
MULTRL	A,B,C,Z	Rounds off decimal places; if overflow, links to Z.
OMIT		Omits source line indicated in positions 1-6 during compilation.
OMIT	A	Omits source lines during recompilation; positions 1-6 = beginning page/line, A = ending page/line.
OPEN	FR	Opens any file.
OPENN	FR	Opens printer file; does not eject page.
OPENPG	FR	Opens magnetic tape piggyback file; does not force rescue dump.
OPENS	FR,A	Opens file section specified by (A).
OPENT	FR,A	Opens all sections of a random file, except those specified in table A.
OVRLAY		Indicates the beginning of a new program overlay.
OVRLAYG		Indicates the beginning of a new program overlay and new group.
PUT	FR	Places a record in a file.
PUT	FR,WA	Places a record from a workarea into a file.
RDUMP		Creates a rescue dump.
RELINK		Removes last link stored in link list; returns control to last address.
RELINK	Z	Removes last link stored in link list; returns control to routine referenced by Z.
*(RENAME)		Assigns reference without establishing new program region.

Instr.	Operands	Description
RESET	FR,A	Restores to memory the record whose address was stored in A by MARK.
RETADD	A	Returns to sort to add the record in A after own code intervention.
RETDEL		Returns to sort to delete the current record after own code intervention.
RETNOR		Returns to sort to process the current record normally after own code intervention.
RETURN	A	Stores contents of index registers in A, and relinks to the calling module.
RGET	FR,A	Presents record addressed by (A).
RGET	FR,A,Z	Presents record addressed by (A); branches to Z if null block.
RFILE	FR,WA	Places a record in the next available space in a file.
RINITR	FR,WA	Initiates a read operation at location specified by (WA), then releases control to next instruction.
RINITW	FR	Initiates a write operation for the current block, then releases control to the next instruction.
RLNKE		If E flag is on, relinks.
RLNKE	Z	If E flag is on, relinks to Z.
RLNKG		If G flag is on, relinks.
RLNKG	Z	If G flag is on, relinks to Z.
RLNKGE		If G or E flag is on, relinks.
RLNKGE	Z	If G or E flag is on, relinks to Z.
RLNKL		If L flag is on, relinks.
RLNKL	Z	If L flag is on, relinks to Z.
RLNKLE		If L or E flag is on, relinks.
RLNKLE	Z	If L or E flag is on, relinks to Z.
RLNKU		If G or L flag is on, relinks.
RLNKU	Z	If G or L flag is on, relinks to Z.
ROPEND	FR	Closes file; reopens as destination.
ROPENR	FR	Closes file; reopens as source-destination.
ROPENP	FR	Closes file; reopens as piggyback.
ROPENS	FR	Closes file; reopens as source.
ROPENS	FR,A	Closes magnetic tape file; reopens section named by (A) as source file.

Instr.	Operands	Description
SAVLST		Allows a relink at random error exit.
SECT		Starts a program section; positions 8-17 contain the program name.
SETPL		Causes renumbering at next source statement; A = desired page/line.
SGET	FR,Z	Presents next record; branches if null block, E-O-B, E-O-S.
SGETC	FR,Z1,Z2	Presents next record; branches to Z1 as SGET; branches to Z2 if last record in block has been presented and record overflow flag is off.
SGETL	FR,Z1,Z2	Presents next record; branches to Z1 as SGET; branches to Z2 if last record in the block has been presented.
SPREAD	A,B	Spreads (A) throughout B; A may be either a reference or a literal.
SUB	A,B	$(B) - (A) \rightarrow B$
SUB	A,B,C	$(B) - (A) \rightarrow C$
SUBC	A,B,Z	If overflow, branches to Z.
SUBC	A,B,C,Z	
SUBL	A,B,Z	If overflow, links to Z.
SUBL	A,B,C,Z	
SUBR	A,B,Z	Rounds off decimal places.
SUBR	A,B,C,Z	
SUBRC	A,B,Z	Rounds off decimal places; if overflow, branches to Z.
SUBRC	A,B,C,Z	
SUBRL	A,B,Z	Rounds off decimal places; if overflow, links to Z.
SUBRL	A,B,C,Z	
SUDOP	A,B,'C','D'	Places compiler file A on SUD B, allocating 'C' sectors on zone 'D'.
TBEGB	TR	Initializes the table to be built.
TBEGF	TR	Initializes the table to perform all other functions.



Instr.	Operands	Descriptions
TBILDD	TR,A,B,Z	Builds or inserts (A) into item location specified by (B); branches to Z when (B) points beyond the limits of the table or when the specified item is active.
TBILDN	TR,A,Z	Builds (A) into the next location; branches to Z when next location is beyond limits of the table.
TDEL	TR, ,Z	Deletes the current item; branches to Z when no items remain in table or when the current item is inactive.
TDEL	TR,A,Z	Stores current item in A, then deletes current item from table; branches to Z as TDEL above.
TFINDB	TR,A, ,Z	Performs a binary search for item whose key is (A); branches to Z when either the desired item is beyond the range of the table (turns G flag on) or is within the range, but physically missing from the table (turns E flag on).
TFINDB	TR,A,B,Z	Performs a binary search for the item whose keys are (A) and (B); branches to Z as TFINDB above.
TFINDD	TR,A,Z	Accesses the item whose position is specified by (A); branches to Z either when the specified location is beyond the range of the table or when the specified item is nonactive.
TFINDN	TR,Z	Accesses the next item; branches to Z when the next item is beyond the limits of the table.
TFINDO	TR,A, ,Z	Performs sequential search for item whose key is (A), beginning where previous search ended; branches to Z when desired item is either beyond the range of the table (turns G flag on) or is within range but physically missing from the table (turns E flag on).
TFINDO	TR,A,B,Z	Performs sequential search for item whose keys are (A) and (B); branches to Z as TFINDO above.

Instr.	Operands	Description
TFINDP	TR,Z	Accesses the previous item; branches to Z when the previous item is beyond the limits of the table.
TFINDR	TR,A, ,Z	Performs a serial search for the item whose key is (A); branches to Z if key does not exist.
TFINDR	TR,A,B,Z	Performs a serial search for the item whose keys are (A) and (B); branches to Z if keys do not exist.
TFINDS	TR,A, ,Z	Performs sequential search for the item whose key is (A); branches when the desired item either is beyond the range of the table (turns G flag on) or is within the range but physically missing from the table (turns E flag on).
TFINDS	TR,A,B,Z	Performs sequential search for the item whose keys are (A) and (B); branches to Z as TFINDS above.
TJUMP	TR,A,Z	Calculates relative location of an item in a table and transfers control to the corresponding transfer-of-control instruction in a list whose base is referenced by A; branches to Z either if instruction in list is other than a LINK or BR or if an off table condition exists.
TMARK	TR,A	Stores in A the address of the current item.
TPACK	TR,Z	Moves active items to beginning of table, inactive items to end; branches to Z if table contains no active items.
TRESET	TR,A,Z	Makes accessible the item whose address is stored in A; branches to Z if the address is no longer within the current limit of the table.
TSERT	TR,A,Z	Inserts (A) into the table at the current location; branches to Z if current table length is maximum.

Instr.	Operands	Description
TSHIFT	TR,A	Destroys last item; moves other items toward end of table; inserts (A) into first position.
TSORTA	TR	Sorts items in table into ascending sequence.
TSORTD	TR	Sorts items in table into descending sequence.
VCOMP	A,B,C,D,E	Compares (A), offset by (D), to (B), offset by (E), for a length specified in C.
VMOVE	A,B,C,D, E,Z	Moves (A), offset by (D), to (B), offset by (E), for a length specified by C; transfers control to Z if the destination field is smaller than the source field.
WRITEBI	FR	Causes software to write immediately the current block of a source-destination file.
WRITSP	FR	Causes the software to write the current block.
XPAND	A,B	Unpacks the decimal (A) → B; the length of B must be twice the length of A.
ZSUP	A,B	Replaces leading zeros or spaces in (A) with the character in B; B may be either a reference or a literal.

**Legend:**

A,B,C	operands	FR	file reference
WA	workarea	TR	table reference
( )	contents	Z	branch address

**NEAT/3 INSTRUCTIONS – LEVEL 2**

Instr.	Oper.	Description	Code
BADD	A,B,T	Binary (A) + (B) → (B)	A
BCOMP	A,B,T	Binary compare (A) to (B)	A
BROV	Z	Branch if overflow occurred	A
BSUB	A,B,T	Binary (B) – (A) → (B)	A
CALLA	A,B	Link to software overlay A, entry point B. Restore PSOA upon return	A
CALLB	A,B	Branch to software overlay A, entry point B. Restore PSOA upon return	A
CALLC	A,B	Branch to software overlay A, entry point B	A
COUNT	A	Subtract 1 from counter if not 0, go to A	C,E
CVB	A,B	Convert packed (A) to binary → (B)	E
CVD	A,B	Convert binary (A) to packed → (B)	E
DCODA	A,B,T	Translate (B) using table (A)	B,E
DCODD	A,B,T	Translate (B) using table (A)	B,E
EDIT	A,B,T	Edit (A) → (B)	C,E
FADD	A,B	Floating (A) + (B) → (B)	D,E
FADD	A,B	Double FI. (A) + (B) → (B)	D,E
FCOMP	A,B	Floating compare (A) to (B)	D,E
FCOMPD	A,B	Double FI. compare (A) to (B)	D,E
FDIV	A,B	Floating (B) ÷ (A)	D,E
FDIVD	A,B	Double FI. (B) ÷ (A)	D,E
FMULT	A,B	Floating (A) X (B)	D,E
FMULTD	A,B	Double FI. (A) X (B)	D,E
IOSET	FR,WA	Set up I/O registers	A
IPOFF	A	Set IP off, go to A	B,E
IPON	A	Set IP on, go to A	B,E
JUMP	A	Go to A, set return in XR8 (X615J)	B,E
LDMONR	A	Moves (A) → Mon register	D,E
LDTR	A	Turn trace permit on	D,E
LINK	Z,P <sub>1</sub> , P <sub>2</sub> ,P <sub>n</sub>	Transfer control to the instruction at Z and set IR8 pointing to parameter table.	A

Instr.	Oper.	Description	Code
LINKJ	Z,P	Same as above, only one parameter permitted	A
LIST		Resume printed listing	A
LMARK	A	Moves (A) → Link List	A
LOAD	Prog. Addr.	Load program into addr.	A
LOCK	A	Lock file or data	A
LOGIC	A,B,T	Boolean functions	D,E,F
LUMARK	A	Reset link list to mark	A
MVAL	A,B,T	Move (A) → (B) left to right	B,E
MVAR	A,B,T	Move (A) → (B) right to left	A
MVBR	A,B,T	Move (B) → (A) right to left	B,E
MVEB	A,B	Move eff B addr → (A)	A
OPENC	FR	Open file at new cycle	A
ORIGIN	A	Set location counter to A	A
PACK	A,B,T	Pack (A) into (B)	A
PADD	A,B,T	Signed add (A) + (B) → (B)	B,E
PCOMP	A,B,T	Signed compare (A) to (B)	B,E
PDIV	A,B,T	Packed (B) ÷ (A)	E,F
PMULT	A,B,T	Packed (A) X (B)	D,E,F
PSUB	A,B,T	Packed (B) - (A) → (B)	B,E
READ	FR	Read physical block	A
REPEAT	A	Repeat next instr. (A) times	A
RESERV	A	Reserve A bytes in data list	A
RESTOR	A	Restore processor state	B,E
SAVE	N	Save SSOA flag as 'N'	A
SAVEQ		Save QSOA	A
SAVES		Save SSOA	A
SAVET		Save TSOA	A
SCANE	A,B,T	Scan (B) for (A) equal	C,E
SCANG	A,B,T	Scan (B) for (A) greater	C,E
SCANL	A,B,T	Scan (B) for (A) less	C,E
SLL	A,B,T	Shift left (A) → (B)	E
SLLD	A,B,T	Shift left double	E
SRL	A,B,T	Shift right (A) → (B)	E
SRLD	A,B,T	Shift right double	E

Instr.	Oper.	Description	Code
STTR	A	Turn trace permit off	D,E
SWIN	A	Read option switches (A)	C,E
TCOMP	A,B,T	Table compare (A) to (B)	D,E
TESTB	A,B,T	Test (B) vs. (T), if true go to A	B,E
TESTCE	A,B,T	Test (B) vs. (T), if = go to A	B,E
TESTCU	A,B,T	Test (B) vs. (T), if ≠ go to A	B,E
UADD	A,B,T	Unsigned (A) + (B) → (B)	A
UNLOCK	A	Unlock file or data	A
UNPACK	A,B,T	Unpack (A) → (B)	A
UNSAV	N	Unsave SSOA flagged 'N'	A
UNSAVQ		Unsave QSOA	A
UNSAVS		Unsave SSOA	A
UNSAVT		Unsave TSOA	A
USE	A	Set compiler option A	A
USUB	A,B,T	Unsigned (B) - (A) → (B)	A
WADD	A,B	Word binary (A) + (B) → (B)	E
WAITI	A	Wait, Display A	A
WDIV	A,B	Word binary (B) ÷ (A)	E
WMULT	A,B	Word binary (A) X (B)	E
WRITE	FR	Write physical block	A
WRITTM	FR	Write tape mark	A
WRTMOF	FR	Write EBCDIC tape mark	A
WSUB	A,B	Word binary (B) - (A) → (B)	E

#### Legend

- A - All Processors
- B - C-101, C-151, C-200, C-201
- C - C-200
- D - C-200, C-201 - Optional
- E - C-251, C-300
- F - C-101, C-151 - Optional

## NEAT/3 SYSTEMS TAGS

General	Data Type/ Length	File-Oriented — P.P.T.	
>EXEC.ERRORSTART	Error Start Entrance	B,8	
>EXEC.ACDATE	Actual Date	X,6	
>EXEC.ACDATEDA	Actual Day	X,2	
>EXEC.ACDATEMO	Actual Month	X,2	
>EXEC.ACDATEYR	Actual Year	X,2	
>EXEC.ACDATEU	Actual Date	U,6	
>EXEC.ACDATEDAU	Actual Day	U,2	
>EXEC.ACDATEMOU	Actual Month	U,2	
>EXEC.ACDATEYRU	Actual Year	U,2	
>EXEC.VRDATE	Virtual Date	X,6	
>EXEC.VRDATEDA	Virtual Day	X,2	
>EXEC.VRDATEMO	Virtual Month	X,2	
>EXEC.VRDATEYR	Virtual Year	X,2	
>EXEC.VRDATEU	Virtual Date	U,6	
>EXEC.VRDATEDAU	Virtual Day	U,2	
>EXEC.VRDATEMOU	Virtual Month	U,2	
>EXEC.VRDATEYRU	Virtual Year	U,2	
>EXEC.VRJDULATE	Virtual Jul. Date	X,5	
>EXEC.VRJDULATEU	Virtual Jul. Date	U,5	
>EXEC.VRSEQDAY	Virtual Seq. Day	X,3	
>EXEC.MF01/MF30	Monitor Flags	X,1	
>EXEC.REMAINDER	Remainder	U,19	
>EXEC.SIMOPTSW	Simulated Opt. Sw.	B,1	
>EXEC.RMPRGFINAD	End of Program	B,3	
>EXEC.SORTSUD	Sort Final Output	X,2	
>EXEC.ZERO	Relocatable Absolute Zero		
>EXEC.RELINK	Relink Routine	B,4	
>EXEC.RMRESFLAG	Rescue/Restart Flag	B,1	
>EXEC.INVALIDIO		B,3	
<b>File-Oriented — Disc and CRAM</b>			
FR.\$NULLFLAG	Current Block Info.	B,1	
FR.\$NONRESTOR	Do not save current rec. loc. after insert	B,1	
FR.\$BINARYTBL	Random Table	B,4	
FR.\$SECTION	Section No.	B,1	
FR.\$RFLAG	Random Flag	B,1	
FR.\$SECTOR	Relative Sector +1	B,2	
FR.\$CARDFLAG	Card Release Flag	B,1	
FR.\$CARD	Relative Card	B,2	
FR.\$TRACK	Relative Track	B,1	
<b>File-Oriented — Magnetic Tape</b>			
FR.\$FILCONFLG	File Control Flag	B,1	
<b>File-Oriented — P.P.T.</b>			
FR.\$FILSTATUS	File Status	B,1	
FR.\$BADCHAR	Illegal char. count — current record	B,1	
FR.\$ERRTYPE	Type of Format Error	B,1	
FR.\$RLENGTH	Max. Record Length Read or Written	B,1	
FR.\$ORIGINATE	Char. which initiated record	B,1	
FR.\$TERMINATE	Char. which terminated record	B,1	
FR.\$REPLACHAR	Char. which replaces illegal char.	B,1	
FR.\$EXCEPCHAR	Location of Invalid Char.	B,1	
FR.\$CHARCOUNT	Number of Chars. in Current Record of Source File.		
	Number of Chars. Encoded for Dest. File.	B,1	
FR.\$PTFLAG1	Decode/Encode Work Flag	B,1	
<b>File-Oriented — Punched Cards</b>			
FR.\$BADCHAR	Illegal Char. Count	B,1	
FR.\$REPLACHAR	Char. Which Will Replace Illegal Char.	B,1	
<b>File-Oriented — Printer</b>			
FR.\$CURLNUM	Current Line No.	B,1	
FR.\$LSTLINE	Last Line No.	B,1	
FR.\$LSTLINEZZ	Last Line No. Interim Files	B,1	
FR.\$CURLNUMZZ	Current Line No. and Seq. No. of Report	B,2	
<b>Table-Oriented</b>			
TR.\$STEMLEN	Item Length	B,3	
TR.\$TOFFSET	Off Set of Table Base	B,3	
<b>Indexed Sequential</b>			
FR.\$ISCONTFILAG	Indexed Seq. Control Flag	B,1	

## FLOWRITE INSTRUCTIONS

Instr.	Operands	Description
(NAME)	TEXT	Page Heading
(NOTE)	TEXT	Notes
(---)	TEXT	Continuation
(XREF)		X Ref. listing
(NLST)		No. instr. listing
(SECT)		Begin section
(ENTR)	TEXT	Entry
(EXIT)	TEXT	Exit
(PROC)	TEXT	Process
(PERF)	TOREF*TEXT	Subroutine link
(PERF)	TOREF,CON1/TOREF1,CONN/TOREFN*TEXT	
(PREX)	TOREF*TEXT	External Routine
(PREX)	TOREF0,CON1/TOREF1,CONN/TOREFN*TEXT	
(INOT)	TEXT	Input/Output
(IOBR)	CON1/TOREF1,CONN/TOREFN*TEXT	
(MANL)	TEXT	Manual operation
(GOTO)	TOREF*TEXT	Branch point
(TEST)	CON1/TOREF1,CON2*TEXT	Decision
(****)	TEXT	Reference insert

## MONITOR CONTROL INSTRUCTIONS

18            24  
**CHAR**      **OO,O,ADDR,LL,CCC**  
 ADDR — Hex address (4 or 6 characters); if address > 64K, use an index register and offset.  
 LL — Number of bytes in decimal to patch.  
 CCC — ASCII patch characters.

18            24    Virtual            Actual  
**DATE**      **DD/MM/YY/DAY,DD/MM/YY/DAY (,P)**  
**DATE**      **DD/MM/YY/DAY,DD/MM/YY/DAY (,P),\***  
**DATE**      **DD/MM/YY/DAY**  
 (,P) — PAL print parameter (optional)  
 \* — Single-system-disc environment

18            24                    18            24  
**DIAL**      **SUD/SUD(,P)** | **DIAL**      **TPUU/SUD(,P)**  
 (,P) — PAL print parameter (optional)

18            24                    18            24  
**DSPLY**    **LL,XXX..XX** | **DSPLYS**    **LL,XXX..XX**  
 XXX..XX — Message to be displayed; DSPLYS requires a response from the operator.

## MONITOR CONTROL INSTRUCTIONS (CONT'D)

18            24  
**EXFILE**    **FN,OF,LL,HHHH**  
**EXFILE**    **FN,TAG,CCCCC**  
 FN — File number assigned to Extremity file table.  
 OF — Relative offset to Extremity file table.  
 LL — Number of bytes in decimal to patch.  
 HHHH — Hex patch data.  
 TAG — May be NAME (for filename), DATE1 (for acceptable period), or DATE2 (for retention period).  
 CCCCC — ASCII patch char.; up to 10 for name, 6 for dates.

18            24  
**FILEP**     **FN,SN,SSSS,EEEE,T**  
 FN — File number.  
 SN — Section number.  
 SSSS — Starting Sector number (5 digits).  
 EEEEE — Ending Sector number (5 digits); if EEEEE is entered, Extremity calculates ending sector number.  
 T — Type of disc: 5=655; 6=656; 7=657. If blank, any type will be used.

18            24  
**FINISH**    **(X)**  
 (X) — Monitor flag parameter (optional); causes the current MF settings to be carried over to the original disc control string when used in an interrupt control string.

8            18            24            35    36  
**CSNAME**    **HEADCS**    **N**            **S**    **R**  
**CSNAMEA**   **HEADCS**    **CSNAME**      **S**    **R**  
**CSNAME**    **HEADCS**    **OVERRIDE**    **S**    **R**  
 CSNAME — Name assigned to new or override control string.  
 CSNAMEA — Name of old control string to be reconstructed.  
 N — New control string.  
 S — Sort request; Y to sort by P/L or merge new instructions with old string; N to renumber but not sort. Enter N for override CS; renumbering must also be inhibited.  
 R — Renumbering request: N to inhibit renumbering; 1-0 for renumbering by increments of 10-100. Must be set to N for override control string.  
 OVERRIDE — Override control string to be constructed.

18            24  
**HEX**        **OO,O,ADDR,LL,HHHH**  
 ADDR — Hex address (4 or 6 characters); if address > 64K, use an index register and offset.  
 LL — Number of bytes in decimal to patch.  
 HHHH — Hex patch data.

## MONITOR CONTROL INSTRUCTIONS (CONT'D)

18	24	18	24
IF	DDD	IF	-DDD
IF	MFXX/C	IF	-MFXX/C

DDD – Dating information; may contain: WD, BWW, EWW, BWM, EWM, MON through SUN.

MFXX – Monitor flag 01-30.

C – Flag value to test.

18	24	18	24
INTYPE	I	INTYPE	C
INTYPE	P		

I – I/O Writer input.

P – Punched Paper Tape input.

C – Punched Card input.

18	24
LIBDSK	SUD

SUD – Symbolic unit designator used to specify library unit.

18  
NOLIB

18	24
NEXTBR	CSNAME
NEXTBR	CSNAMEVN
NEXTBR	CSNAME.XXXXXX
NEXTBR	CSNAMEVN.XXXXXX

CSNAME – Control string or program name.

VN – Version number of control string or program.

XXXXXX - Page and line (CS) or entry point (program).

18	24
NEXTDO	PROGNAME
NEXTDO	PROGNAMEVN
NEXTDO	PROGNAME.AAAAA
NEXTDO	PROGNAME.XXXXXX
NEXTDO	PROGNAMEVN.XXXXXX

PROGNAME – Name of program or control string.

VN – Version number of program or control string.

XXXXXX – Page and line (CS) or entry point (program).

AAAAA – Decimal address in program to receive control.

18	24
PALENT	PT,TPUUU,CWLOC,(D)
PALENT	PT,TPUUU,CWLOC,(A)
PALENT	PT,TPUU,CWLOC,TN,(S)

PT – Peripheral type code.

(,D) – Duplicate entry parameter (optional).

(,A) – Alternate display device parameter (optional).

TN – Train number (646/647 printers).

(,S) – Suppress train no. display at boot (optional).

## MONITOR CONTROL INSTRUCTIONS (CONT'D)

18	24	18	24
PURGE	O	PURGE	A
PURGE	L		

O – Override to be purged from disc work storage.

L – Monitor link list to be purged.

A – Overrides and Monitor Link list are to be purged.

18	24
REMDSC	SUD,(P)

(,P) – PAL print parameter (optional).

18  
STOPRD

18	24	18	24
SYSPER	SUD,(P)	SYSPER	SUD/SUD,(P)
SYSPER	SUD/SUD,(A)		

(,P) – PAL Print Parameter (optional).

(,A) – Alternate display device parameter (optional).

18	24
TRAIN	SUD,NN

NN – Train array to be stored in image memory of 646/647 printer specified by SUD.

## PROGRAM OVERLAY CALLS

Overlay Group Table Entry (4 bytes)

OVN	ADDR
-----	------

OVN – Current overlay in memory for this group.

ADDR – Base address (entry point statement)

OVERLAY CALLS

9C	G <sub>1</sub> C	OV	G <sub>2</sub> EP
----	------------------	----	-------------------

9C – Overlay call opcode

G<sub>1</sub> – Overlay group of this command

C – Opcode –

4 – Link W/Parameters

8 – Relink specific

0 – All other command

OV – Overlay No. to be called

G<sub>2</sub> – Group No. of overlay to be called

EP – Entry point

TRANSFER TO MAIN PROGRAM FROM OVLY

9C	G <sub>1</sub> C	00	00	00	RETURN ADDR
----	------------------	----	----	----	-------------

## LOADING COT BOOTS

### 615-50/100

1. Place Previous Systems Disc on D01 or D02.
2. Place HALT switch ON.
3. Place DATE card behind the COT BOOT in reader.
4. Press STOP, FEED, START for card reader.
5. Toggle RESET switch.
6. Turn FUNCTION SELECT to LOAD ADDRESS.
7. Set ADDRESS ENTER to 00A0 (Start Address).
8. Press ACT.
9. Press LOAD.
10. Turn FUNCTION SELECT to NEW CRS.
11. Press ACT.
12. Place HALT switch OFF.
13. Press COMPUTE.

### 615-101/151/200/201

1. Place Previous Systems Disc on D01 or D02.
2. Place HALT switch ON.
3. Place DATE card behind COT BOOT in reader.
4. Press STOP, FEED, START for card reader.
5. Toggle RESET switch.
6. Turn FUNCTION SELECT to NEW LA.
7. Set ADDRESS ENTER to 000A0 (Start Address).
8. Press ACT.
9. Press LOAD.
10. Turn FUNCTION SELECT to NEW CR.
11. Press ACT.
12. Place HALT switch OFF.
13. Press COMPUTE.

### 615-251/300

1. Place Previous Systems Disc on D01 or D02.
2. Place HALT switch ON.
3. Set LOAD SELECT dials to the TRUNK, POSITION, UNIT, and FUNCTION of the peripheral to provide COT BOOT input (freestanding card reader).
4. Load COT BOOT cards in the freestanding card reader. Press STOP, FEED, and START for card reader.
5. Toggle RESET switch.
6. Press ENTER and CONTROL REGISTER.
7. Type in address 0000A0.
8. Press LOAD.
9. Place HALT switch OFF.
10. Press COMPUTE. If calling monitor from a high density 657 Disc, a wait will occur after reading the COT BOOT cards. Press COMPUTE to continue.

## ERROR START PROCEDURES

### 615-50/100

1. Place HALT switch ON.
2. Toggle RESET switch.
3. Turn FUNCTION SELECT to DATA ADDRESS.
4. Set ADDRESS ENTER to 0228 (Simulated Option Switch).
5. Press ACT.
6. Turn FUNCTION SELECT to DATA on ENTER side.
7. Set DATA ENTER to desired hexadecimal simulated option switch setting.
8. Press ACT.
9. Turn FUNCTION SELECT to NEW CRU.
10. Set ADDRESS ENTER to 0388.
11. Press ACT.
12. Turn FUNCTION SELECT to NEW CRS.
13. Set ADDRESS ENTER to 039C.
14. Press ACT.
15. Place HALT switch OFF.
16. Press COMPUTE.

### 615-101/151/200/201

1. Place HALT switch ON.
2. Toggle RESET switch.
3. Turn FUNCTION SELECT to NEW LA.
4. Set ADDRESS ENTER to 00228.
5. Press ACT.
6. Set FUNCTION SELECT to DATA (M) on ENTER side.
7. Set DATA ENTER to desired hexadecimal simulated option switch setting.
8. Press ACT.
9. Turn FUNCTION SELECT to NEW CR.
10. Set ADDRESS ENTER to 00388.
11. Press ACT.
12. Place HALT switch OFF.
13. Press COMPUTE.

### 615-251/300

1. Place HALT switch ON.
2. Toggle RESET switch.
3. Press NEW MEMORY ADDRESS.
4. Type 000A28.
5. Press ENTER.
6. Type desired hexadecimal simulated option switch setting.
7. Press CONTROL REGISTER.
8. Type 000B88.
9. Place HALT switch OFF.
10. Press COMPUTE.

NCR CENTURY 50/100 MEMORY MAP

DEC	HEX	CONTENTS			DEC	HEX		CONTENTS	DEC	HEX	CONTENTS	
0000	0000	To	T	"0" CC	0112	0070		IR28L	0224	00E0		IR56
0004	0004	FLAGS		EFF. A	0116	0074		IR29M	0228	00E4		IR57
0008	0008	R		EFF. B	0120	0078		IR30A	0232	00E8		IR58
0012	000C			CRS	0124	007C		IR31B	0236	00EC		IR59
0016	0010	To	T	"0" CC	0128	0080	Print Cntl	IR320	0240	00F0		IR60
0020	0014	FLAGS		EFF. A	0132	0084		IR33K	0244	00F4		IR61
0024	0018	R		EFF. B	0136	0088		IR34N	0248	00F8		IR62T
0028	001C			CRU	0140	008C		IR35Z	0252	00FC		IR63R
0032	0020	Repeat Cntr		IR8J	0144	0090		IR36X	0256	0100	ME BR COMMAND	
0036	0024			IR9S	0148	0094		IR37Y	0260	0104	PE BR COMMAND	
0040	0028			IR10	0152	0098		IR38W	0264	0108	TI BR COMMAND	
0044	002C			IR11	0156	009C		IR39V				
0048	0030			IR12	0160	00A0		IR40	1024	0400	CWO	
0052	0034			IR13	0164	00A4		IR41			COT	
0056	0038			IR14	0168	00A8		IR42	1032	0408	CW1 I/OW & IN Swtchs	
0060	003C			IR15	0172	00AC		IR43				
0064	0040			IR16	0176	00B0		IR44	1040	0410	CW2	
0068	0044			IR17	0180	00B4		IR45			PRINTER	
0072	0048			IR18	0184	00B8		IR46	1048	0418	CW3	
0076	004C			IR19	0188	00BC		IR47			DISC	
0080	0050			IR20	0192	00C0		IR48	1056	0420	CW4	
0084	0054			IR21G	0196	00C4		IR49				
0088	0058			IR22E	0200	00C8		IR50	1064	0428	CW5	
0092	005C			IR23U	0204	00CC		IR51				
0096	0060			IR24P	0208	00D0		IR52				
0100	0064			IR25F	0212	00D4		IR53				
0104	0068			IR26D	0216	00D8		IR54	1272	04F8	CW31	
0108	006C			IR27C	0220	00DC		IR55				



# NCR CENTURY 101/151/200/201 MEMORY MAP

DEC	HEX	CONTENTS		DEC	HEX	CONTENTS	DEC	HEX	CONTENTS
0000	0000			0192	00C0				
0004	0004		X6151 IR1	0196	00C4				
0008	0008		X6152 IR2	0200	00C8				
0012	000C		X6153 IR3	0204	00CC				
0016	0010		X6154 IR4	0208	00D0				
0020	0014	Error Status Word		0212	00D4				
0024	0018	for ME, PE, or CC Trap		0216	00D8				
0028	001C			0220	00DC				
0032	0020	Repeat Counter	X615J LINK	0224	00E0				
0036	0024	PE TCC	X615S NAIR	0228	00E4				
0040	0028	Program Status Word		0232	00E8				
0044	002C	For Program Interrupt		0236	00EC				
0048	0030			0240	00F0				
0052	0034	Program Status Word *		0244	00F4				
0056	0038	For Trace Trap		0248	00F8				
0060	003C			0252	00FC				
0064	0040	Count Counter		0256	0100				
0068	0044			0260	0104				
0072	0048			0264	0108				
0076	004C			0268	010C				
0080	0050			0272	0110				
0084	0054		X615G IR21	0276	0114				
0088	0058		X615E IR22	0280	0118				
0092	005C		X615U IR23	0284	011C				
0096	0060		X615P IR24	0288	0120				
0100	0064		X615F IR25	0292	0124				
0104	0068		X615D IR26	0296	0128				
0108	006C		X615C IR27	0300	012C				
0112	0070		X615L IR28	0304	0130				
0116	0074		X615M IR29	0308	0134				
0120	0078		X615A IR30	0312	0138				
0124	007C		X615B IR31	0316	013C				
0128	0080		X615O IR32	0320	0140				
0132	0084		X615K IR33	0324	0144				
0136	0088		X615N IR34	0328	0148				
0140	008C		X615Z IR35	0332	014C				
0144	0090		X615X IR36	0336	0150				
0148	0094		X615Y IR37	0340	0154				
0152	0098		X615W IR38	0344	0158				
0156	009C		X615V IR39	0348	015C				
0160	00A0			1024	0400				
0164	00A4								
0168	00A8			1032	0408				
0172	00AC								
0176	00B0			1040	0410				
0180	00B4								
0184	00B8								
0188	00BC								

CONTENTS	DEC	HEX	CONTENTS
IR48	3064	0BF8	CW255
IR49			
IR50	6144	1800	Used by * 315 Simulator Option
IR51			
IR52			
IR53			
IR54			
IR55			
IR56			
IR57			
IR58	6352	18D0	
IR59			
IR60			
IR61			
X615T			
X615R			
ME Trap Address			
PE CC Trap Address			
Prog. Inter. Address			
Trace Cont. Address *			
Table Comp. Address *			
X1	Temporary Storage for		C-101 Divide Work Area
X2	Multiply and		
X4	Floating Point		
	Commands		
X8	Tables		
Memory Accumulator Used by Multiply And Floating Point Commands			
Special Control word for Interval Timer			
Base 1401 Convert. Comm. *			
Initial B1401 Add Comm. *			
CW0	C-101	Card Reader	* Not Applicable To C-101
	C-200	CR&IN Swtchs	
CW1	C-101	Console	
	C-200	Printer	
CW2	C-101	Printer	
	C-200	I/O Writer	
CW3	C-101	I/O Writer	



## HEXADECIMAL AND DECIMAL CONVERSION

To find the decimal number, locate the Hex number and its decimal equivalent for each position. Add these to obtain the decimal number. To find the Hex number, locate the next lower decimal number and its Hex equivalent. Each difference is used to obtain the next Hex number until the entire number is developed.

Byte		Byte		Byte		Byte		Byte			
0123		4567		0123		4567		0123		4567	
Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
0	0	0	0	0	0	0	0	0	0	0	0
1	1,048,576	1	65,536	1	4,096	1	256	1	16	1	1
2	2,097,152	2	131,072	2	8,192	2	512	2	32	2	2
3	3,145,728	3	196,608	3	12,288	3	768	3	48	3	3
4	4,194,304	4	262,144	4	16,384	4	1,024	4	64	4	4
5	5,242,880	5	327,680	5	20,480	5	1,280	5	80	5	5
6	6,291,456	6	393,216	6	24,576	6	1,536	6	96	6	6
7	7,340,032	7	458,752	7	28,672	7	1,792	7	112	7	7
8	8,388,608	8	524,288	8	32,768	8	2,048	8	128	8	8
9	9,437,184	9	589,824	9	36,864	9	2,304	9	144	9	9
A	10,485,760	A	655,360	A	40,960	A	2,560	A	160	A	10
B	11,534,336	B	720,896	B	45,056	B	2,816	B	176	B	11
C	12,582,912	C	786,432	C	49,152	C	3,072	C	192	C	12
D	13,631,488	D	851,968	D	53,248	D	3,328	D	208	D	13
E	14,680,064	E	917,504	E	57,344	E	3,584	E	224	E	14
F	15,728,640	F	983,040	F	61,440	F	3,840	F	240	F	15
	6		5		4		3		2		1

Example: Decimal 10,484 to Hex

$$10,484 = 10484 \div 8192 = 2292 \div 2048 = 244 \div 240 = 4 \div 4 = 0$$

2                      8                      F                      4 = Hex 28F4

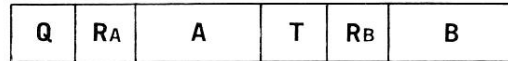
Example: Hex 28F4 to Decimal

$$28F4 = 2 + 8 + F + 4$$

$$8192 + 2048 + 240 + 4 = \text{Decimal } 10484$$

Powers of 16		Powers of 2	
$16^n$	n	$2^n$	n
1	0	512	9
16	1	1024	10
256	2	2048	11
4096	3	4096	12
65536	4	8192	13
1048576	5	16384	14
16777216	6	32768	15
268435456	7	65536	16
4294967296	8	131072	17
68719476736	9	262144	18
1099511627776	10	524288	19
1759218604416	11	1048576	20
281474976710656	12	2097152	21
4503599627370496	13	4194304	22
72057594037927936	14	8388608	23
1152921504606846976	15	16777216	24

## HARDWARE COMMAND FORMAT

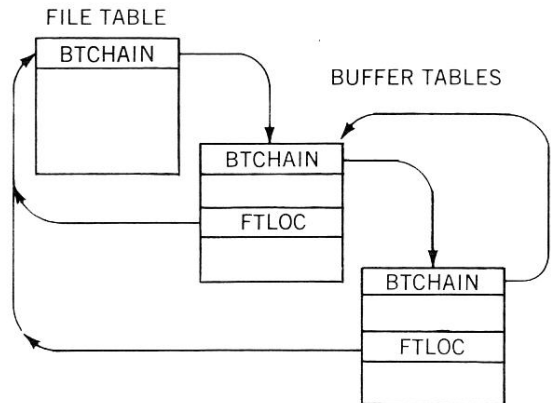


- Q — OPERATION CODE
- R<sub>A</sub> — A1 ADDRESS REGISTER
- A — A ADDRESS  
(R<sub>A</sub>) + A = EFFECTIVE A ADDRESS
- T — LENGTH
- R<sub>B</sub> — B ADDRESS REGISTER
- B — B ADDRESS  
(R<sub>B</sub>) + B = EFFECTIVE B ADDRESS

### SINGLE ADDRESS FORMAT



## FILE BUFFER CHAIN



FILE TABLE POINTS TO CURRENT BUFFER TABLE  
 CURRENT B.T. POINTS TO OLDEST BUFFER TABLE  
 LAST BUFFER TABLE POINTS TO FIRST B.T.

### COMMON SECTION OF THE FILE TABLE

Tag	Loc.	Len.	Tag	Loc.	Len.
BTCHAIN	0	3	TRANLOC	21	3
VFY2OVNO	3	2	HEADER	24	2
VERIFYLOC	5	3	TRAILER	26	2
INITIOFLAG	8	1	RECORDSIZE	28	2
WRITELOC	9	3	EXTOVNO	30	2
COMPIOFLAG	12	1	EXTPARTFT	32	2
READLOC	13	3	FILENO	34	1
RECORDFLAG	16	1	ERRORANS	35	1
AXRLOC	17	3	ERRORCODE	36	1
WORKAGUAR	20	1	ERROREXIT	37	3

### COMMON SECTION OF THE BUFFER TABLE

Tag	Loc.	Len.	Tag	Loc.	Len.
BTCHAIN	0	3	VSTATUS	16	1
CWLOC	3	2	VRETRYCNT	17	1
ACTLIST	5	3	NOVFY	18	1
BUFSTATUS	8	1	MODULE	19	1
FTLOC	9	3	CBUFNA	20	2
FBUFNA	12	2	CBUFTA	22	2
FBUFTA	14	2			

### COMMON SECTION OF THE EXTREMITY TABLE

Tag	Loc.	Len.	Tag	Loc.	Len.
LENGTH	0	2	READLOC	17	3
CLASS	2	1	CSECTION	20	1
VERSION	3	1	WRITELOC	21	3
AOPENFLAG	4	1	GENERATION	24	1
FTLOC	5	3	STARTEXIT	25	3
OPENFLAG	8	1	USAGEFLAG	28	1
FTTYPE	9	1	ENDEXIT	29	3
TPU	10	2	FILENAME	32	10
CWLOC	12	2	DATE1	42	6
EXTFLAG1	14	1	DATE2	48	6
FILENO	15	1	DATE3	54	6
OVFLAG	16	1			

*SUD = 72 1K*  
*90 1K*  
*32 File Size 99 28*

### PATCH CARD FORMATS - OPURCARE

7	18	24
CSPEC\$ PLACE	PROGNAMEVR,SUD <sup>1</sup> ,SUD <sup>2</sup>	
CSPEC\$ PLACEI	PROGNAMEVR,SUD <sup>3</sup>	
CSPEC\$ HEX	RC,GROVN,ADDR,LL,HHHH	
CSPEC\$ CHAR	RC,GROVN,ADDR,LL,CCC	
CSPEC\$ EXFILE	FN,OF,LL,HHHH	
CSPEC\$ EXFILE	FN,TAG,CCCCC	
CSPEC\$ RELOC	ADDR	
CSPEC\$ ADBUFF	P,FN,SIZE,N	

PROGNAMEVR - Program name and version number.  
 SUD<sup>1</sup> - SUD of disc containing object program.  
 SUD<sup>2</sup> - SUD of disc to contain new object program.  
 SUD<sup>3</sup> - SUD of disc containing object prog.; patch in place.  
 RC - Relocation Constant (see list below).  
 GROVN - Two char. group and three char. overlay number.  
 ADDR - Hex address (4 or 6 characters); if > 64K use an index register and offset.  
 LL - Number of bytes in decimal to patch; can be any length that will fit on a single card. Maximum for EXFILE is 22.  
 HHHH - Hex patch data.  
 CCC - ASCII patch characters; if EXFILE, up to 10 for name, 6 for dates.  
 FN - File number assigned to Extremity file table.  
 OF - Relative offset to Extremity file table.  
 TAG - May be NAME (for filename), DATE1 (for acceptable period), or DATE2 (for retention period).  
 P - Processor type: 1=100; 2=101,151,200,201; 3=251,300.  
 SIZE - Size of buffer in decimal; must equal size of current buffer for the file.  
 N - Number of buffers to add.

### RELOCATION CONSTANTS

OO - The relocation flag associated with the patch need not be altered.  
 CR - The patch is coding that contains an operand address that must be adjusted if the program base address is changed.  
 CN - The patch is coding that contains an operand address that should not be adjusted if the program base address is changed.  
 A2 - The patch is a 2-byte binary address constant that must be adjusted if the program base address is changed.  
 A3 - The patch is a 3-byte binary address constant that must be adjusted if the program base address is changed.  
 A4 - The patch is a 4-byte reference address constant that must be adjusted if the program base address is changed.  
 KN - The patch is data that does not contain addresses.

## PERTINENT MEMORY LOCATIONS

Tag	Loc.	Len.	Contents
X615M	0075	3	Data List Pointer
X615J	0021	3	LINKP Parameter Pointer
X615G	0055	3	EXEC.BASE Register
X615U	005D	3	>EXEC.REMAINDER Ptr.
X615P	0061	3	Address of PSOA
X615L	0071	3	Link List Pointer
X615Z	008D	3	Control Word Pointer
*RMTYPEPROC	0160	1	Processor Type
*RMACMEMSIZE	0161	3	Memory Size
*RMCVOLSER	0182	3	Current Vol. Ser. No.
*RMTYPEEXEC	0187	1	Executive Type Used
*RMLOCISOAFLG	018A	3	Location of SOA flags
*RMMAXSOAFL	018E	1	No. of SOA Flags in Memory
*RMCSTSCSOA	0194	2	Start Sector for Saving SOAs
*RMCPAFTP	019E	1	Current Disc Trunk & Position
*RMCPAFUNIT	019F	1	Current Disc Unit
*RMCBSFOSEC	01A1	2	Software Ovly Start Sector
*RMAVOLSER	01AC	3	Alternate Disc Vol. Ser. No.
*RMAPAFTP	01B8	1	Alt. Disc Trunk & Position
*RMAPAFUNIT	01B9	1	Alt. Disc Unit
*RMADDOGTBL	020F	3	Ovly Group Table Address
*RMSIMOPCDS	021A	4	User CCT Table Address
*RMPROGRAM	022A	10	Program Name
*RMPROGTPU	0234	2	Program Disc TPU
*MF01-MF30	0236	30	Monitor Flags
*RMSOFTACCU	0254	50	Software Accumulator
*RMMEPESAVE	0289	16	Processor State at ME/PE

\* Relative to X615G

## DYNAMIC DUMP PATCHES

This coding may be patched into a program at run time.

RESERV	6	A1000008
MVEB	0(X615M),BEGIN,3	2474000003XXXXXX
MVAR	-10(X615U),2(X615M),1	645CFF601740002
MVEB	3(X615M),END,3	2474000303YYYYYY
CALLA	208,1	225403E4040000D0
RELINK	1	A254036C

This coding will dump all of memory.

CALLA	208,0	225403E4000000D0
-------	-------	------------------

## SUPERVISOR TRANSFER TABLE

(These common entry points are relative to X615G, and receive control via link, branch or Sup. table transfer H22.)

Tag	Loc.	Function
CCTRAPENT	0360	Entrance to Simulate CCT
UNLOCK	0364	Unlock Routine
LOCK	0368	Lock Routine
LINK RELINK	036C	Special Relink (Relink 1)
MARK	0374	Link List Mark Routine
TESTIO	0378	I/O Die Down Routine
RSMFLAG	037C	Exec. Supports RSM if FF
RSMLOCAT	037D	RSM or Special I/O Entrance
PACKCHANGE	0380	Operator Initiated Pack Change
ERRORSTART	0388	Errorstart Entrance
INVALIDIO	0390	Invalid I/O Routine
SPECIALIO	0398	Special I/O Routine
CCTRAPRTN	039C	Return to User
CCTRAPRTN2	03A0	Special Return to User
COMMUNICAT	03A4	Exec. Data Pass Along Area
UNSAVE	03A8	SOA Unsave Routine
SAVE	03AC	SOA Save Routine
NOTOK	03B0	} Data Traffic Controller Entrances
CLEARLIST	03B4	
CALLVfy2	03B8	
STACK	03BC	
REACTIVATE	03C0	
WRITEOK	03C4	
READOK	03C8	
COMPCCRIO	03CC	
COMPOLDIO	03D0	
RINITIATE	03D4	
INITIATE	03D8	
CALLC	03DC	Type C Software OVLY Call
CALLB	03E0	Type B Software OVLY Call
CALLA	03E4	Type A Software OVLY Call
RELINK	03E8	Relink Routine
CCTABLE	03EC	Addr. of Exec CCT Table

## TABLE CONTROL ENTRY

Loc.	Len.	Contents
0	3	Location of Table Base Address
3	1	Table Structure
4	4	Address & Length — Key 1
8	4	Address & Length — Key 2
12	1	Ordering of Keys
13	3	Absolute Address of Base
16	1	Rel. Addr. Last Entry
17	3	Addr. of Assoc. I.R. +1
20	3	Maximum Length of Table
23	3	Table Offset
26	3	Rel. Item Pointer

## IOSET MACRO

IOSET FR  
IOSET FR,WORKAREA

I.R.	Addr	Tag	Contents
30	78	X615A	File Table Address
31	7C	X615B	Current Buffer Table Address
27	6C	X615C	Address of Workarea
26	68	X615D	Address of Assoc. I.R.
22	58	X615E	Contents of Assoc. I.R.

## SIMULATED OPTION SWITCH (LOC 228)

Bit	Hex Entry	Function
1	01	Interrupt Desired
2	02	Halt After Present Program
3	04	Memory Print
4	08	Print Program Status Report
5	10	Print PAL Status Report
6	20	Abort Disc Control String
7	40	Close All Open Files; If Not Set, Obsolete
		All Open First Time Files
8	80	Unused

## MAXIMUM LENGTH OF OPERANDS

Instruction	Perform On Data		Maximum Length Of Operand	
	Type	Type	Source	Destination
Compare	B	B	8	8
	D	D	20	20
	U&Z	U&Z	19	19
	D	U&Z	20	19
	U&Z	D	19	20
	X&S	X&S	64K	64K
	K	K	10	10
	P\$	P\$	10	10
Add/Sub	B	B	8	8
	D	D		
	D	U&Z	19 unsigned	19 unsigned
	U&Z	U&Z	20 signed	20 signed
Mult/Div	D	D		
	D	U&Z	19 unsigned	19 unsigned
	U&Z	U&Z	20 signed	20 signed
	U&Z	D		
Standard Move	B	B	8	8
	D	D	20	20
	U&Z	U&Z	19	19
	D	U&Z	20	19
	U&Z	D	19	20
	P	P		
	K	K	10	10
	K	P		
	P	K		
	X&S	X&S	64K	64K
Conversion Move	B	D	8	20
	B	U&Z	8	19
	K	B	20	8
	U&Z	B	19	8
	D	P	20	10
	U&Z	P	19	10
	D	K	20	10
	U&Z	K	19	10
	P	D	10	20
	P	U&Z	10	19
	K	D	10	20
	K	U&Z	10	19
	K	X&S	64K	64K
	Editing Move	U&Z	E	19
D		E	20	43
X&S		E*	43	43
<b>Legend</b>				
B — Binary				
D — Signed Decimal				
U&Z — Unsigned Decimal				
X&S — Alphanumeric Character Set				
K — Unsigned Packed Decimal				
P — Signed Packed Decimal				
E — Edited (Numeric)				
E* — Edited (Alphanumeric)				
P — On C-100, P Type Data Cannot be Compared				
K — On C-100, Both Operands Should Have the Same Decimal Length				

## EDITING MASK

Alphanumeric Mask		Editing Character - X Insertion Character - Other than X using USASI code		
Mask	Explanation	Examples		
		Mask	Data	Output
X	Should include insertion chars.			
B	Blank	XBXXXXX XBXXXXXXX	TEATON RHAMILTON	TTEATON R/RHAMILTON
Others	USASI characters besides B should be inserted as is	XX/XX/XX	052473	05/24/73
ATTENTION: Left justify, leave DP blank				
Numeric Mask		Editing Character - X, Z, *, +, -, DB, CR Insertion Character - Others using USASI code		
Mask	Explanation	Examples		
		Mask	Data	Output
X	Software aligns DP's. Includes insertion chars. (overflow to the left/right will be truncated)	XX.XX	55555(1)	55.50
Z	Leading zero replaced with space. Must not have any editing char. to the left of the Z except + or -	ZZ.ZZ ZZZX +ZZZX	0550(2) 0000 00760+	05.50 00 +00760
*	Leading zero replaced with *. Must not have any editing char. to the left of the * except " or -	***XX **.*	000172 0000	***172 **.*
+	Place in either first or last char. position of the editing mask	+	+	+
-	Place in either first or last char. position of the editing mask	-	-	-
CR	Place in either first or last char. position of the editing mask	CR CR	+	CR
DB	Place in either first or last char. position of the editing mask	DB DB	+	DB
B	Blank insertion	XXXBXX	01234	01234
Others	Should be inserted as is	TOTALB\$XX.XX	4960(2)	TOTAL\$49.60
\$	Floating currency symbol	\$\$\$XX.XX	402700(2)	\$4027.00
£		-££.XX	143-(2)	-£1.43
.	Ordinarily used as DP. If all chars. are not zero the DP is not suppressed.	ZZZ.ZZ ZZZ.ZZ	00001(2) 00000	.01 SPACE

ATTENTION: Right Justify, Align on DP Fill in DP Positions

## PRINTER CHARACTER SETS

N	B	A	E	U
0	A	N	\$	/
1	B	O	£	+
2	C	P	*	:
3	D	Q		=
4	E	R		,
5	F	S		(
6	G	T		)
7	H	U		<
8	I	V		>
9	J	W		←
.	K	X		↑
,	L	Y		#
-	M	Z		!

## PRINTER CONTROL BLOCK

F	G	P	S
---	---	---	---

### F Code:

- P Print after slewing 'S' lines
- L Print after slewing to line 'S'
- N Do not print; slew 'S' lines
- E Eject form to top of page

### G Code:

- N Numeric character set
- B Basic alphanumeric set
- A Alphanumeric set
- E Extended character set
- U Upper/lower character set

### P Code:

- Selective print character

### S Code:

- Number of lines to slew or line number (F code L)

## COMMON STATUS CHARACTERS

### S2 STATUS

Hex	Condition
02	Inoperative
40	Command initiated
80	Busy
82	Standby

### S3 STATUS

Binary	
00XXXXXX	Operation complete
11XXXXXX	Segment complete
Hex	
01	Special
02	Inoperative
08	Media failure
10	System overload
81	Transmission error
82	Standby

### S4 STATUS

81	Transmission error
84	Latent ME
88	Latent PE

## ANSI PAPER TAPE CODE

CHARACTER I/D	8	7	6	5	4	3	2	1	
NUL									*1
SOH	•								
STX	•								
ETX									
EOT	•								
ENO									
ACK									
BEL	•								
BS	•								
HT									
LF									
VT	•								
FF									
CR	•								
SO									*2
SI									*3
DLE	•								
DC1									
DC2									
DC3	•								
DC4									
NAK	•								
SYN	•								*4
ETB									
CAN									
EM	•								*5
SS									*6
ESC									
FS	•								*7
GS									
RS									*8
US	•								*10

CHARACTER I/D	8	7	6	5	4	3	2	1	
@	•	•							
A									
B									
C									
D									
E									
F									
G									
H									
I									
J									
K									
L									
M									
N									
O									
P									
Q									
R									
S									
T									
U									
V									
W									
X									
Y									
Z									
[									
\									
]									
^ or									
_ or									

SP (2)	8	7	6	5	4	3	2	1	
!									
"									
# (£)									
\$									
%									
&									
'									
(									
)									
*									
+									
,									
-									
.									
/									
0									
1									
2									
3									
4									
5									
6									
7									
8									
9									
:									
;									
<									
=									
>									
?									

	8	7	6	5	4	3	2	1	
a									
b									
c									
d									
e									
f									
g									
h									
i									
j									
k									
l									
m									
n									
o									
p									
q									
r									
s									
t									
u									
v									
w									
x									
y									
z									
{									
}									
DEL									*11

\* One of the following control characters:

1 Null	5 Cancel	9 End of Record
2 Upper Shift	6 End of Media	10 End of Field
3 Lower Shift	7 Escape Character	11 Delete
4 Sync Code	8 End of File	

## SYMBOLIC DEBUG

7 18 24  
**CSPECSDEBUG DUMP** DDDDD,PPLLLL,NNNNNN,FLD1,  
 CC,FLD2,AREA1,AREA2,ETC.  
**CSPECSDEBUG BRPATH** DDDDD,PPLLLL,PPLLLL,  
 NNNNNN

\*DDDDD – Delay counter

PPLLLL – Page line no. in program to trigger DUMP or  
 BRPATH

\*NNNNNN – No. of times to DUMP or BRPATH

\*FLD1,CC,FLD2 – Conditionals: If FLD1 and FLD2 meet  
 condition, perform DUMP.

EQ – Equal

LE – Less or equal

NE – Not equal

GE – Greater or equal

LT – Less than

FC – Field change

GT – Greater than

AREA1,AREA2,Etc – Fields/areas to be dumped

\*Optional Parameters – May be omitted for unconditional  
 DUMP.

## INDEXED SEQUENTIAL MACROS

Macro	Operands	Description
ISLODR	IS,KEY,Z,WA	Loads record in file from workarea. Routine Z receives control on error.
ISXIND	IS,Z	Updates all levels of index file.
ISRETR	IS,KEY,Z	Random retrieval of record
ISADDR	IS,KEY,Z,WA	Inserts record in file
ISDELR	IS,KEY,Z,WA	Deletes record from file
ISSUPR	IS,KEY,Z,N,O	Replaces old record with new record
ISNXTR	IS,Z	Sequentially accesses next record
ISWRTR	IS	Updates current record
IS	– Indexed sequential name	
KEY	– Record key	
Z	– Error exit	
WA	– Workarea	
N	– Area containing new record	
O	– Area to receive old record	



## RANDOM FILING SYSTEM MACROS

Macro	Operands	Description
RFGET	RFS,KEY,Z	Get random record
RFPUT	RFS,Z	Put random record
RFDEL	RFS,KEY,Z	Delete random record
RFUPD	RFS	Update current record
RFREL	RFS	Permits simultaneity
RFADDR	RFS,KEY,ADDR	Key randomizing routine

RFS — Random filing system name  
 KEY — Record key  
 Z — Error exit  
 ADDR — Area to receive random address

## SYMBOLIC UNIT DESIGNATORS

Disc	D01,D02,Dnn
Magnetic Tape	M01,M02,Mnn
CRAM	C01,C02,Cnn
Printer	P01,P02,P0n
Card Reader	P11,P12,P1n
Card Punch	P21,P22,P2n
P.T. Reader	P31,P32,P3n
P.T. Punch	P41,P42,P4n
OCR	P51,P52,P5n
MICR	P61,P62,P6n
621, 622	R01,R02,Rnn
I/O Writer	I01,I02,Inn
CalComp	P71,P72,P7n
Com. Prog. Lib.	D90,D91,D9n
315 RMC EM	Enn
B3/B4 Common Sys. Disc	D00
Online Adapter	A00,A01,Ann

## MONITOR FLAGS

Flag	Set	Meaning
MF01 (cc51)	C	Clear All Monitor Flags to Alpha Zero.
	0	No overrides are present; do not clear MF.
	1	Overrides are present; do not clear MF.
	2	Same as C.
MF02	3	Overrides are Present; Clear all MF.
	0	Do Not Use Printer for Log Entries.
	1	Use the Printer for Log Entries.
	2	Use the Printer for Prog. Status Rept.
	3	Use the Printer for Both.
	4	Do Not Use Printer for Log Entries. Suppress all But Hardware Messages.
	5	Use the Printer for Log Entries. Suppress All But Hardware Messages.
6	Use the Printer for Prog. Status Report. Suppress All But Hardware Messages.	
7	Use the Printer for Both. Suppress All But Hardware Messages.	
MF03	0	No Halt is Desired.
	1	Halt Prior to Execution.
	2	Halt After Execution.
MF04	3	Halt Prior to and After Execution.
	0	No Memory Dump is Desired.
	1	Memory Dump Prior to Execution.
MF05	2	Memory Dump After Execution.
	3	Memory Dump Prior to & After Execution.
	0	A Memory Clear is Not Desired.
MF06	1	Clear Memory Before Loading Program.
	2	Do not clear Memory; Call RSM.
	3	Clear Memory; Call RSM.
MF07	0	Run in Production Mode.
	P	Production Debug Mode Without Trace.
	T	Production Debug Mode With Trace.
	S	Symbolic Debug Mode.
MF08	0	Display the Program Name.
	1	Do Not Display the Program Name.
MF09	0	Do Not Change Pack on Library Unit.
	1	Change Before and After Execution.
	0	Make Memory Check.
MF10	1	Do Not Make Memory Check.
	2	B1-2A & Check Memory.
	3	B1-2A & Do Not Check Memory.
MF10	0	B1 Executive.
	1	Dedicated On-Line B2.
	2	Dual Programming B2.

## PERIPHERAL TYPE CODES

Peripheral Description	Code
<b>Cards</b>	
682-100 Card Reader (300 cpm)	00†
682-300 Card Reader (300 cpm)	00
686-102 Card Reader/Punch (800/83-294 cpm)	01
686-111 Card Reader/Punch (560/60-180 cpm)	01
684-101 Card Reader/Punch (500/100 cpm)	01*
686-201 Card Reader (750 cpm)	02
686-302 Card Punch (82-240 cpm)	03
686-311 Card Punch (60-180 cpm)	03
684-301 Card Punch (100 cpm)	03*
687-301 Card Punch (100 cpm)	05
680-201 Card Reader (1200/1600 cpm)	06
684-101 Card Reader/Punch (500/100 cpm)	07**
684-301 Card Punch (100 cpm)	09**
<b>Paper Tape</b>	
662-100 Paper Tape Reader (1000 cps)	10†
660-101 Paper Tape Reader (1500 cps)	11
665-101 Paper Tape Punch (200 cps)	12
<b>Printers</b>	
640-102 132 column, single numeric	20†
640-102 132 column, double numeric	21†
640-200 132 column, single numeric	22†
640-200 132 column, double numeric	23†
640-210 160 column, single numeric	24†
640-210 160 column, double numeric	25†
640-300 132 column, expanded alpha/num	26†
640-102 132 column, single numeric	60
640-102 132 column, double numeric	61
640-200 132 column, single numeric	62
649-300 132 column, single numeric	62
640-200 132 column, double numeric	63
640-210 160 column, single numeric	64
640-210 160 column, double numeric	65
640-300 132 column, expanded alpha/num	66
646 132 column, train	68
647 132 column, train	69
<b>Discs</b>	
655-101 Dual Disc Unit (108 Kb)	30†
655-102 Dual Disc Unit (108 Kb)	31†
655-201 Dual Disc Unit (108 Kb)	32
656-102 Disc Unit (312.5 Kb)	36
657-101 Disc Unit (315 Kb)	37
657-102 Disc Unit (500 Kb)	37
658-101 Disc Unit (806 Kb)	37*
658-101 Disc Unit (806 Kb)	38**
<b>CRAM</b>	
653-101 145 million character capacity	50

## Mag Tape

633-111 9-ch, 1600 bpi, Phase, single unit (50 ips)	40
634-All 9-ch, 800/1600 bpi, NRZI/Phase (25/50 ips)	40
635-109 9-ch, 800/1600 bpi, NRZI/Phase (100 ips)	40
635-209 9-ch, 800/1600 bpi, NRZI/Phase (200 ips)	40
633-211 9-ch, 1600 bpi, Phase, single unit (90 ips)	41
633-311 9-ch, 1600 bpi, Phase, single unit (150 ips)	42
633-121 9-ch, 1600 bpi, Phase, dual unit (50 ips)	43
633-221 9-ch, 1600 bpi, Phase, dual unit (90 ips)	44
633-119 9-ch, 800 bpi, NRZI, (50 ips)	45
633-117 7-ch, 200/556/800 bpi, NRZI (50 ips)	48
634-All 7-ch, 200/556/800 bpi, NRZI (25 ips)	48

## Console Peripherals

Integrated CRT Screen	F1†
Integrated I/O Writer	F2†
I/O Writer through Multiplexor	F3
CRT Screen — C-251,C-300	F4†
CRT Keyboard — C-251,C-300	F5†
CRT Touchplate — C-251,C-300	F6†
Thermal Printer	F7†

## MICR

670-101 MICR Sorter (600 dpm, 11 pockets)	80
671-101 MICR Sorter (1200 dpm, 18 pockets)	81

## OCR

420-1 Optical Character Reader	70
420-2 Optical Character Reader	71

## Encoders

736 Magnetic Tape Encoder	90
---------------------------	----

## Plotter

Calcomp Plotter	A0
-----------------	----

## Communications Multiplexor

621-101 Communications Multiplexor	B0
621-102 Communications Multiplexor	B1
621-103 Communications Multiplexor	B2

## Remote Terminals

Online Adapter	C0
----------------	----

## Emulators

627-300 RMC Emulator	D0
----------------------	----

## Cassette

636 NCR Century Cassette	45
--------------------------	----

\* File Specifications Sheet Only

\*\* PALENT Cards Only

† Integrated Unit

## DATA FORMAT CODES

External Code Set	Data Format Code
<b>Punched Card Stacking Functions Not Used</b>	
No Translation –	
1 Character Per Column	00
Standard Century H Set	01
Standard Century A Set	02
*315 Hollerith	03
*Binary – No Translation	04
Non-Standard Code Set	0D
<b>Punched Card Stacking Functions Used</b>	
No Translation –	
1 Character Per Column	10
Standard Century H Set	11
Standard Century A Set	12
*315 Hollerith	13
*Binary – No Translation	14
Non-Standard Code Set	1D
<b>Punched Paper Tape</b>	
Century Standard Code Set – USASI	20
User Defined	2D
No Translation	00
<b>Printer</b>	
No Format Control	00
Standard Vertical Format Control	30
Reporter	31
<b>Magnetic Tape</b>	
Century Internal Code – USASI	00
IBM BCD Code	40
IBM EBCDIC Code	41
315 Internal Code	42
315 Internal Code and Label Format	43
315 Internal Code and Label – No Translation	44
<b>Disc and CRAM</b>	
Century Internal Code – USASI	00

\*NOTE: Binary Reader required for these codes.

## HEXADECIMAL OP CODES

	Single Stage (4 bytes)	Double Stage (8 bytes)
00 80		
01 81	Algebraic Compare	
02 82	Add, Round, Check, Store in C	
03 83	Sub, Round, Check, Store in C	
04 84	Interpretive – Unsigned Multiply	
05 85	Interpretive – Unsigned Divide	
06 86	Round, Check, Store in C, Save Remainder	
07 87	Interpretive – Compare	
08 88	Interpretive – Unsigned Decarith Setup	
09 89	Interpretive – Signed Decarith Setup	
0A 8A	Interpretive – Signed Multiply	
0B 8B	Interpretive – Signed Divide	
0C 8C	Interpretive – MOVEINT	
0D 8D	Interpretive – DECEDIT	
0E 8E		
0F 8F		
10 90		
11 91		
12 92		
13 93		
14 94		
15 95		
16 96		
17 97		
18 98		
19 99		
1A 9A		
1B 9B		
1C 9C	Overlay Caller	
1D 9D	Symbolic Debug	
1E 9E	Symbolic Debug	
1F 9F	ZLINKI	

## HEXADECIMAL OP CODES (Cont'd)

	Single Stage	Double Stage
20 A0	IOSET	
21 A1	RESERV	
22 A2	Supervisor Table Transfer	
23 A3		
24 A4	MVEB	
25 A5	LOCK	
26 A6	UNLOCK	
27 A7	RELINK	
28 A8	LINK	
29 A9	LINKP	
2A AA		
2B AB		
2C AC		
2D AD		
2E AE		
2F AF		
30 B0		
31 B1	LOADT	
32 B2	TESTSL	
33 B3	WCOMP	
34 B4		
35 B5	SLL, SLLD, SRL, SRLD	
36 B6	WMULT	
37 B7	WDIV	
38 B8	XADD	
39 B9	XSUB	
3A BA	XSCAN	
3B BB	XMOVE	
3C BC	CADR	
3D BD	XECUTE	
3E BE	XPACK	
3F BF	XUNPACK	

## HEXADECIMAL OP CODES (Cont'd)

	Single Stage	Double Stage
40 C0	PADD	
41 C1	PSUB	
42 C2	WADD	
43 C3	WSUB	
44 C4	MVBR	
45 C5	PCOMP	
46 C6	IPON	
47 C7	IPOFF	
48 C8	RESTOR	
49 C9	EDIT	
4A CA	COUNT	
4B CB	JUMP	
4C CC	PACK	
4D CD	UNPACK	
4E CE	DCODD	
4F CF	DCODA	
50 D0	SWIN	
51 D1	TESTCE	
52 D2	TESTCU	
53 D3	TESTB	
54 D4	MVAL	
55 D5	SCANL	
56 D6	SCANE	
57 D7	SCANG	
58 D8	LDBAR	
59 D9	LDMONR	
5A DA	LDTR	
5B DB	STTR	
5C DC	TCOMP	
5D DD	PMULT	
5E DE	LOGIC	
5F DF	SUMCK	
60 E0	BADD	
61 E1	BSUB	
62 E2	UADD	
63 E3	USUB	
64 E4	MVAR	
65 E5	BCOMP	
66 E6	REPEAT	
67 E7	WAITI	
68 E8	BROV	
69 E9	BRL	
6A EA	BRE	
6B EB	BRLE	
6C EC	BRG	
6D ED	BRLG (BRU)	
6E EE	BRGE	
6F EF	BR	
70 F0	INOUT	
71 F1	PDIV	
72 F2	CVD	
73 F3	CVB	
74 F4	FADD	
75 F5	FADDD	
76 F6	FSUB	
77 F7	FSUBD	
78 F8	FMADD	
79 F9	FMADDD	
7A FA	FCOMP	
7B FB	FCOMPD	
7C FC	FMULT	
7D FD	FMULTD	
7E FE	FDIV	
7F FF	FDIVD	





FILE SPECIFICATIONS (CONT'D)

Sheet No. 3 CRAM																																																																								
PAGELINE	SUD	SECTION NUMBER	NAME OF ASSOCIATED DOMINANT FILE	NUMBER OF CARDS	SECTION LEVEL	OVERFLOW	BUCKET SIZE	ALLOC. CONST.	NO. TRACKS	OVERFLOW	ALT. SUD																																																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73

Sheet No. 4 Paper Tape																																																																								
PAGELINE	F	FILE REFERENCE	SUD	DATA FORMAT CODE	DATA FORMAT ERROR EXIT	END OF FILE EXIT	RESCUE	PROG. MODE	BUCKET SIZE																																																															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73

Sheet No. 5 Paper Tape (Optional)																																																																								
PAGELINE	F	FILE REFERENCE	USER ROUTINE AFTER SECTION OPEN	NO. TAPE FEEDS TO INDICATE	ALTERNATION																																																																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73

FILE SPECIFICATIONS (CONT'D)

Sheet No. 6 Printer																																																																								
PAGELINE	F	FILE REFERENCE	SUD	RECORD LENGTH	DATA FORMAT CODES	END OF PAGE ROUTINE	LAST DATA LINE NUMBER	TEST PATTERN	FORM NUMBER	MAG. OUTPUT																																																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73

Sheet No. 7 Disc-Magnetic Tape-CRAM																																																																								
PAGELINE	F	FILE REFERENCE	PERIPHERAL TYPE CODE	FILE USAGE	RECORD LENGTH	REC. TYPE	MAX. PACK	BLOCK LENGTH	RES. POINT	STD. RESCUE	DATING	DATE	RENTION	DATE	FILE NAME	END OF FILE EXIT	M	REWIND	BLI																																																					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73

Sheet No. 8 Not Used











## INDEX (CONT'D)

Memory Locations, Pertinent . . . . .	.34
Memory Map, NCR Century 50/100 . . . . .	.24
Memory Map, NCR Century 101/151/200/201 . . . . .	.26
Memory Map, NCR Century 251/300 . . . . .	.28
Monitor Control Instructions . . . . .	.18
Monitor Flag Settings . . . . .	.43
Monitor Patch Card Formats . . . . .	.18
Monitor Simulated Option Switch . . . . .	.36
NEAT/3 Instructions — Level 1 . . . . .	.3
NEAT/3 Instructions — Level 2 . . . . .	.13
NEAT/3 Source Program Organization . . . . .	.2
NEAT/3 System Tags . . . . .	.16
Op Codes, Hexadecimal . . . . .	.47
Operand Lengths . . . . .	.37
Option Switch, Simulated . . . . .	.36
OPURCARE Patch Card Formats . . . . .	.33
Organization of NEAT/3 Source Program . . . . .	.2
Overlay Calls, Program . . . . .	.21
Patch Card Formats — Monitor . . . . .	.18
Patch Card Formats — OPUrcARE . . . . .	.33
Peripheral Type Codes . . . . .	.44
Pertinent Memory Locations . . . . .	.34
Printer Character Sets . . . . .	.38
Printer Control Block . . . . .	.39
Program Overlay Calls . . . . .	.21
Punched Card Code Sets . . . . .	.50
Punched Paper Tape Code Set, ANSI . . . . .	.40
Relocation Constants . . . . .	.33
Random Filing System Macros . . . . .	.42
Random Filing System Worksheets . . . . .	.62
Reserved Memory Locations . . . . .	.34
Simulated Option Switch . . . . .	.36
Sort Worksheets . . . . .	.60
Source Program Organization, NEAT/3 . . . . .	.2
Status Characters, Common . . . . .	.39
Supervisor Transfer Table . . . . .	.35
Symbolic Debug Formats . . . . .	.41
Symbolic Unit Designator . . . . .	.42
System Tag, NEAT/3 . . . . .	.16
Table Control Entry . . . . .	.36
Table Specifications Sheet . . . . .	.59
Transfer Table, Supervisor . . . . .	.35
Worksheets, Compiler . . . . .	.58
Worksheets, File Specifications . . . . .	.52
Worksheets, Indexed Sequential Control . . . . .	.59
Worksheets, Random Filing System . . . . .	.62
Worksheets, Sort . . . . .	.60
Worksheets, Table Specifications . . . . .	.59

very minor changes – additional blank pages for notes.