NIC-1080

Computer Assembler Instructions

500	A+M	ac + memory
520	AMP	ac + memory + 1
460	A-M	ac - memory
320	M-A	memory - ac
440	ACM	ac + complement of memory
300	CAM	complement of ac + memory
000	AND	and ac and memory
100	MEM	memory
120	MPO	memory + 1
700	MMO	memory - 1
040	MCP	complement of memory
060	MNG	- memory
400	ACC	ac
420	APO	ac + 1
540	AMO	ac - 1
200	ACP	complement of ac
220	ANG	- ac
160	ZER	0
020	ONE	1
140	MON	- 1
740	MTO	- 2
20000	JMS	jump to subroutine
00000	JMP	jump

suffixes

A = 10
M = 4
Z = 2

addressing

(immediate = 0000
memory ref = 2000
@ indirect = 3000

NICOLET INSTRUMENT CORPORATION
5225 Verona Road, Madison, Wisconsin 53711

shifts

5000	LASH n	left arithmetic
5020	RASH n	right arithmetic
5040	LLSH n	left logical
5060	RLSH n	right logical
405020	RISH n	right integer
105000	VDSH	vertical display scale

$0 \leqslant n \leqslant 17$

test instructions

SKIP			EXCT
405120	ZAC	on ac = 0	405160
425120	MOAC	on ac = - 1	425160
545120	POAC	on ac = + l	545160
5110	ACO	on ac0 = 1	5150
5104	AC19	on ac negative	5144
5101	L	on Link = 1	5141

input/output

6454	TTYRF	skip on tty keyboard-reader ready
44453	RDTTY	read tty keyboard-reader into ac0-7
6444	TTYPF	skip on tty printer ready
4443	PRTTY	print (punch) ac0-7
6464	HSRF	skip on high speed reader ready
44463	RHSR	read high speed reader into acO-7
6474	HSPF	skip on high speed punch ready
4473	PHSP	punch ac0-7

miscellaneous

5220	STOP	stop processor
5210	CLL	clear link
5204	STL	set link
5202	TACL	link > acl9
5201	TLAC	acl9 > link

Automatic Arithmetic

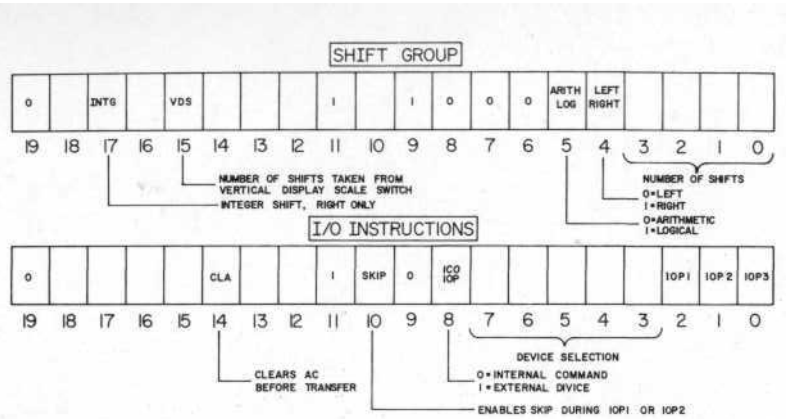
505320	MULT	multiply mq by next location
465300	DIVD	divide ac-mq (LASH 1) by next location
4354	TACMQ	ac -> mq
4343	TMQAC	mq -> ac
44354	ZRAM	zero ac and mq
4347	BITINV	bit invert ac

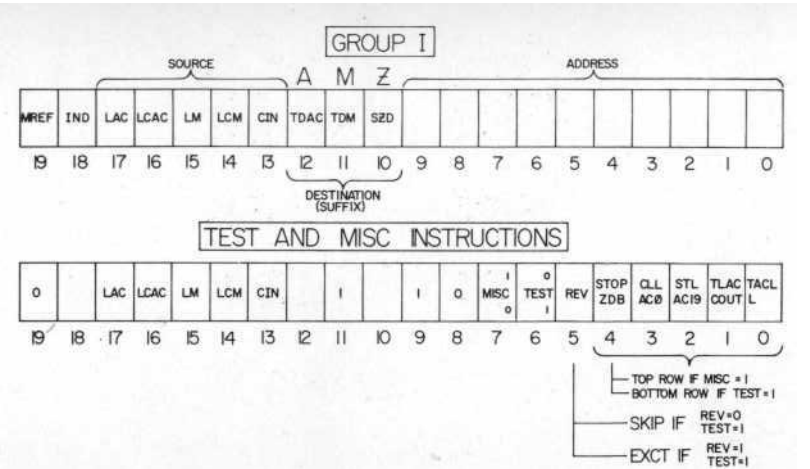
Floating Point (7118)

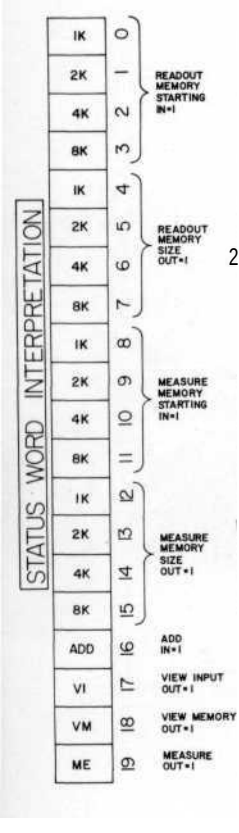
FADD	fac + far -> fac	7245		
FSUB	fac - far -> fac	7314		
FNEG	-fac -> fac	7320		
FMULT	fac x far -> fac	7416		
FDIV	fac / far -> fac	7461	ERRF	= 7555
FLOP	floating output	6510	CARCNT	= 7021
FLIP	floating input	6736	VFLAG	= 7004
GETAC	x->fac	7062	FACE	= 7572
GETAR	x -> far	7050	FACM	= 7573
FACFAR	fac -> far	7026	FARE	= 7575
PUTAC	fac -> x	7074	FARM	= 7576
FLOAT	floats facm-facml	7534		
FIX	fixes fac	7541		
FSIN	sin(fac) -> fac	6001		
FCOS	cos(fac) -> fac	6113		
FARCTN	arctan(fac) fac	6121		
FRIP	1/fac -> fac	6170		
FSQRT	$fac^{1/2}$ -> fac	6176		
FLOG	log(fac) -> fac	6322		
FLN	ln(fac) -> fac	6330		
FSQAR	fac^2 -> fac	6352		
FEXP	10^{fac} -> fac	6370		
FEXPN	e^{fac} -> fac	6376		

NICO-LOADEON

	7736	7744	7745	2705751
	7737	5007	7746	2001736
SA =	7740	4453	7747	2001736
	7741	6454	7750	2001736
	7742	1741	7751	2407777
	7743	1001736	7752	1744
	7744	0171736	7753	1750


SHIFT GROUP
0
INTG
VDS
1
1
0
0
0
ARITH LOG
LEFT RIGHT
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
NUMBER OF SHIFTS TAKEN FROM VERTICAL DISPLAY SCALE SWITCH
INTEGER SHIFT, RIGHT ONLY
NUMBER OF SHIFTS
0=LEFT
1=RIGHT
0=ARITHMETIC
1=LOGICAL
I/O INSTRUCTIONS
0
CLA
1
SKIP
0
ICO IOP
IOP1
IOP2
IOP3
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
DEVICE SELECTION
CLEARS AC BEFORE TRANSFER
0=INTERNAL COMMAND
1=EXTERNAL DIVICE
ENABLES SKIP DURING IOP1 OR IOP2


GROUP I
SOURCE
A M Z
ADDRESS
MREF IND LAC LCAC LM LCM CIN TDAC TDM SZD
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
DESTINATION (SUFFIX)
TEST AND MISC INSTRUCTIONS
0 LAC LCAC LM LCM CIN 1 1 1 0 MISC 1 0 TEST 0 1 REV STOP ZDB CLL AC0 STL AC19 TLAC COUT TACL L
19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
TOP ROW IF MISC = 1
BOTTOM ROW IF TEST = 1
SKIP IF REV=0 TEST=1
EXCT IF REV=1 TEST=1



display

214001	TACXD	ac0-13 to x (ac compl)
4012	TACYD	ac6-19 to y
4014	INCXD	increment x
4011	INTENS	intensify

hardware access

44034	STATUS	status word to ac
4371	REDS	reset digitizer
4372	STDG	start digitization
4374	RDG	read digitizer
4361	ASRMP	advance sweep ramp
6362	DWSK	skip on dwell
4364	RSWP	reset sweep ramp

ASCII CODE

Character	Code	Character	Code
A	301	"	242
B	302	#	243
C	303	$	244
D	304	%	245
E	305	&	246
F	306	'	247
G	307	(	250
H	310	)	251
I	311	*	252
J	312	+	253
K	313	,	254
L	314	-	255
M	315	.	256
N	316	/	257
O	317	:	272
P	320	;	273
Q	321	<	274
R	322	=	275
S	323	>	276
T	324	?	277
U	325	@	300
V	326	[	333
W	327	\	334
X	330	]	335
Y	331	^	336
Z	332	<-	337
0	260	EOT	204
1	261	WRU	205
2	262	RU	206
3	263	BELL	207
4	264	TAB	211
5	265	Line Feed	212
6	266	FORM	214
7	267	Return	215
8	270	Space	240
9	271	ALT MODE	375
!	241	Rub Out	377

ASSEMBLER-EDITOR COMMANDS

Assembler

R — read in source tape
LSR if ADD, HSR if SUB
B — punch binary tape
E — error analysis
L — list assembled code
H — all output on HSP
CTRL/E — Enter Editor

Editor

W — write out source tape
Pm nnnn — Print line **nnnn** in mode **m**
Im nnnn — Insert before line **nnnn** (CTRL/D to exit)
Dm nnnn — Delete line **nnnn**
N — Print next line
A — Append more tape
LSR if ADD, HSR if SUB
CTRL/FORM — Exit Editor

FFT Commands

BC baseline correction
EM exponential multiplication
TC enter time constant
FT Fourier transform
PA enter zero order phase const
PB first order phase const
PC do phase correction
MC magnitude calculation
DT data transfer
AC add constant
PV print value
IM integrate memory
IL integrate between limits
AP apodization
AN enter apodization %
LI link commands
AU perform linked commands

Binary Punch SA = 5000