ECLIPSE S/20 I/O Device Codes [5]

Octal Device Code	Mnemonic	Priority Mask Bit	Device Name
00	—	—	Returned by power monitor in response to INTA
01	APL	—	Auto program load register
02	PAR	—	Parity checking
03	MAP	—	Memory allocation and protection
04			
05			
06			
07			
10	TTI	14	TTY input
11	TTO	15	TTY output
12	PTR	11	Paper tape reader
13			
14	RTC	13	Real-time clock
15			
16			
17	LPT	12	Line printer
20			
21	ADCV	8	A-D converter
22	MTA	10	Magnetic tape
23	DACV	8	D-A converter
24	NVM	10	I/O memory
25	NVM1	10	I/O memory
26	DPH [or NVM2]	7 [10]	12.5- or 25-Mbyte disk and 1.25-Mbyte diskettes [or I/O memory]
27	DPD [or NVM3]	7 [10]	10-Mbyte cartridge disk [or I/O memory]
30			
31			
32			
33	DPX	10	315-Kbyte diskette subsystem
34	MUX [ASLM]	8 [9]	Sync/async controller [sync/async controller]
35	CRC	8	Cyclic redundancy checker
36			
37			
40			
41		5	Digital I/O interface
42	DIO	11	Programmable interval timer
43	PIT		
44	MUX1	8	Second sync/async controller
45	CRC	8	Second cyclic redundancy checker controller
46			
47	VID	6	Video interface
50	TTI1	14	Async controller/remote restart receiver
51	TTO1	15	Async controller/remote restart transmitter
52	PTR1	11	Second paper tape reader
53			
54	RTC1	13	Second real-time clock
55			
56			
57	LPT1	12	Second line printer
60			
61	ADCV1	8	Second A-D interface
62	MTA1	10	Second magnetic tape
63	DACV1	8	Second D-A interface
64	SNVM	10	I/O memory
65	SNVM1	10	I/O memory
66	DPH1 [SNVM2]	7 [10]	Second 12.5- or 25-Mbyte disk and 1.25-Mbyte diskettes [I/O memory]
67	DPD1 [SNVM3]	7 [10]	Second 10-Mbyte cartridge disk [I/O memory]
70			
71			
72			
73	DPX	10	Second 315-Kbyte diskette subsystem
74	ASLM1	9	Second sync/async controller
75			
76			
77	CPU	—	Central processor and console functions

ECLIPSE I/O Device Codes
(except for the ECLIPSE S/20)

Octal Device Code	Mnemonic	Priority Mask Bit	Device Name
00	—	—	Unused
01	WCS [APL]	—	Writable control store option [or APL register]
02	ERCC	—	Error checking and correction
03	MAP	—	Memory allocation and protection
04	HOST		IOP MAP
05	BMC	—	Burst multiplexor channel
06	MCAT	12	Multiprocessor adapter transmitter
07	MCAR	12	Multiprocessor adapter receiver
10	TTI	14	TTY input
11	TTO	15	TTY output
12	PTR	11	Paper tape reader
13	PTP	13	Paper tape punch
14	RTC	13	Real-time clock
15	PLT	12	Incremental plotter
16	CDR	10	Card reader
17	LPT	12	Line printer
20	DSK	9	Fixed-head disc
21	ADCV	8	A-D converter

ECLIPSE I/O Device Codes (continued) [6]

Octal Device Code	Mnemonic	Priority Mask Bit	Device Name
22	MTA	10	Magnetic tape
23	DACV	None	D-A converter
24	DCM	0	Data communications multiplexor
25			
26	DKB	9	Fixed-head DG/Disc
27	DPF	7	DG/Disk storage subsystem
30	QTY	14	Asynchronous hardware multiplexor
30	SLA	14	Synchronous line adapter
31[1]	IBM1	13	IBM 360/370 interface
32	IBM2	13	IBM 360/370 interface
33	DKP	7	Moving head disk
34	CAS[1]	10	Cassette tape
	DCU[4]	4	Data control unit
34	MX1	11	Multiline asynchronous controller
35	MX2	11	Multiline asynchronous controller
36	IPB	6	Interprocessor bus—half-duplex
37	IVT	6	IPB watchdog timer
40[2]	DPI	8	IPB full-duplex input
40	SCR	8	Synchronous communication receiver
41[3]	DPO	8	IPB full-duplex output
41	SCT	8	Synchronous communication transmitter
42	DIO	7	Digital I/O
43	DIOT	6	Digital I/O timer
43	PIT	11	Programmable interval timer
44	MXM	12	Modem control for MX1/MX2
45			
46	MCAT1	12	Second multiprocessor transmitter
47	MCAR1	12	Second multiprocessor receiver
50	TTI1	14	Second TTY input
51	TTO1	15	Second TTY output
52	PTR1	11	Second paper tape reader
53	PTP1	13	Second paper tape punch
54	RTC1	13	Second real-time clock
55	PLT1	12	Second incremental plotter
56	CDR1	10	Second card reader
57	LPT1	12	Second line printer
60	DSK1	9	Second fixed-head disk
61	ADCV1	8	Second A-D converter
62	MTA1	10	Second magnetic tape
63	DACV1	None	Second D-A converter
64			
65	IOP1	5[5]	Host to IOP interface
66	DKB1	9	Second fixed-head DG/Disk
67	DPF1	7	Second DG/Disk storage subsystem
70	QTY1	14	Second asynchronous hardware multiplexor
70	SLA1	14	Second synchronous line adapter
71[1]		13	Second IBM 360/370 interface
72		13	Second IBM 360/370 interface
73	DKP1	7	Second moving head disk
74	CAS1	10	Second cassette tape
74[1]		11	Second multiline asynchronous controller
75		11	Second multiline asynchronous controller
76	DPU	4	DCU to host interface
77	CPU	—	Central processor and console functions

[1]Code returned by INTA and used by VCT.
[2]Can be set up with any unused even device code equal to 40 or above.
[3]Can be set up with any unused odd device code equal to 41 or above.
[4]Can be set to any unused device code between 1 and 76.
[5]Micro interrupts are not maskable.

ASCII Character Codes

Decimal	7-Bit Octal	Character	Even Parity 8-Bit Code	Decimal	7-Bit Octal	Character	Even Parity 8-Bit Code
0	000	NUL	000	20	024	DC4	024
1	001	SOH	201	21	025	NAK	225
2	002	STX	202	22	026	SYN	226
3	003	ETX	003	23	027	ETB	027
4	004	EOT	204	24	030	CAN	030
5	005	ENQ	005	25	031	EM	231
6	006	ACK	006	26	032	SUB	232
7	007	BEL	207	27	033	ESC	033
8	010	BS	210	28	034	FS	234
9	011	HT	011	29	035	GS	035
10	012	NL	012	30	036	RS	036
11	013	VT	213	31	037	US	237
12	014	FF	014	32	040	SP	240
13	015	CR	215	33	041	!	041
14	016	SO	216	34	042	"	042
15	017	SI	017	35	043	#	243
16	020	DLE	220	36	044	$	044
17	021	DC1	021	37	045	%	245
18	022	DC2	022	38	046	&	246
19	023	DC3	023	39	047	'	047

ASCII Character Codes (continued) [7]

Decimal	7-Bit Octal	Character	Even Parity 8-Bit Code	Decimal	7-Bit Octal	Character	Even Parity 8-Bit Code
40	050	(	050	84	124	T	324
41	051	)	251	85	125	U	125
42	052	*	252	86	126	V	126
43	053	+	053	87	127	W	327
44	054	,		88	130	X	330
45	055	—	055	89	131	Y	131
46	056	.	056	90	132	Z	132
47	057	/	257	91	133	[	333
48	060	0	060	92	134	\	134
49	061	1	261	93	135	]	335
50	062	2	262	94	136	^	336
51	063	3	063	95	137	_	137
52	064	4	264	96	140	`	140
53	065	5	065	97	141	a	341
54	066	6	066	98	142	b	342
55	067	7	267	99	143	c	143
56	070	8	270	100	144	d	344
57	071	9	071	101	145	e	145
58	072	:	072	102	146	f	146
59	073	;	273	103	147	g	347
60	074	<	074	104	150	h	350
61	075	=	275	105	151	i	151
62	076	>	276	106	152	j	152
63	077	?	077	107	153	k	353
64	100	@	300	108	154	l	154
65	101	A	101	109	155	m	355
66	102	B	102	110	156	n	356
67	103	C	303	111	157	o	157
68	104	D	104	112	160	p	360
69	105	E	305	113	161	q	161
70	106	F	306	114	162	r	162
71	107	G	107	115	163	s	363
72	110	H	110	116	164	t	164
73	111	I	311	117	165	u	365
74	112	J	312	118	166	v	366
75	113	K	113	119	167	w	167
76	114	L	314	120	170	x	170
77	115	M	115	121	171	y	371
78	116	N	116	122	172	z	372
79	117	O	317	123	173	{	173
80	120	P	120	124	174	|	374
81	121	Q	321	125	175	}	175
82	122	R	322	126	176	~	176
83	123	S	123	127	177	DEL	377

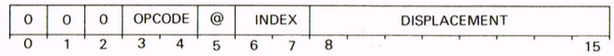
Power of 2 Table

2^n	n
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
131 072	17
262 144	18
524 288	19
1 048 576	20
2 097 152	21
4 194 304	22
8 388 608	23
16 777 216	24

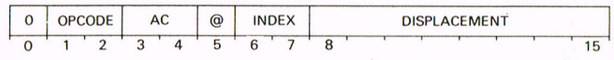
ECLIPSE® LINE COMPUTERS Instruction Reference Card

Accumulator Formats
Single-Word Instructions

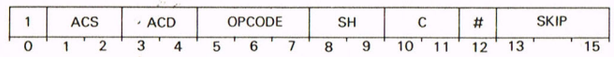
No accumulator — effective address



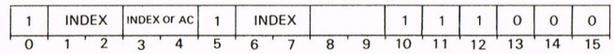
Single accumulator — effective address

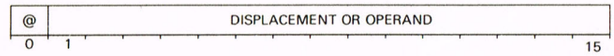


Two accumulators — multiple operations

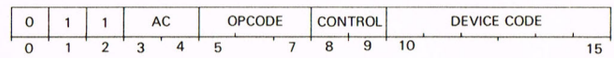


Multiple-Word Instructions



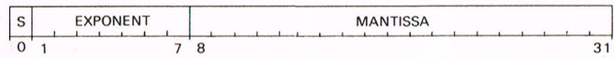


I/O Format

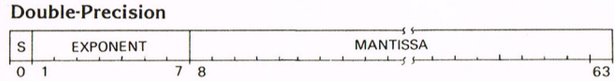


Floating-Point Number Format

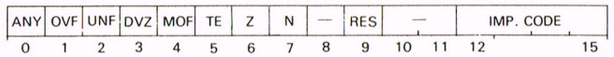
Single-Precision

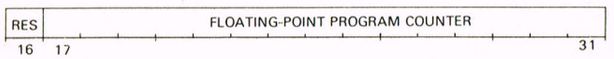


Double-Precision



Floating-Point Status Register





DataGeneral
Technical Products Publications Department
Westboro, Massachusetts 01581

014-000627-01

Coding Conventions

Symbols listed in upper case are to be coded exactly as shown.

The symbols <> are not coded. They act only to indicate how an assembly language instruction may be written. They indicate optional operands. The operand enclosed in the brackets (for example, <#>) may be coded or not, depending on whether or not the associated option is desired.

The use of italic typeface in the formats indicates that a specific substitution is required. Replace the italicized abbreviation with the indicated accumulator, address, name, or number.

Abbreviations

AC	= Accumulator
ACS	= Source Accumulator
ACD	= Destination Accumulator
N	= Digit in the Range of 0 to 3
FPAC	= Floating-Point Accumulator
FACS	= Floating-Point Source AC
FACD	= Floating-Point Destination AC
n	= Digit in the Range of 1 to 4
i	= 16-bit Signed or Unsigned Integer

Instructions that use an effective address follow these coding conventions. The indirect bit is set by coding the symbol @ somewhere in the operand string. The index bits are set by coding a comma followed by one of the digits 0 to 3 as the last operand of the operand string. If no index is coded, the bits are set to 00. The displacement is coded as a signed number in the current assembler radix. If no sign is coded, the number is assumed to be positive.

The standard instructions that use the two accumulator/multiple operation formats have several options that can be obtained by coding the optional operands. The characters to be coded are listed below with their results.

Class Abbreviation	Coded Character	Result Bits	Operation
c	(omitted)	00	Do not initialize the carry bit.
	Z	01	Initialize the carry bit to zero.
	O	10	Initialize the carry bit to one.
	C	11	Initialize the carry bit to the complement of its present value.
sh	(omitted)	00	Leave the result of the operation unaffected.
	L	01	Combine the carry and the 16-bit result into a 17-bit number and rotate it one bit left.
	R	10	Combine the carry and the 16-bit result into a 17-bit number and rotate it one bit right.
	S	11	Exchange the two 8-bit halves of the 16-bit result without affecting the carry.
#	(omitted)	0	Load the result of the shift operation into ACD.
	#	1	Do not load the result of the shift operation into ACD.

The following operands initiate operations that test the result of the shift operation. If the condition is true, the next sequential instruction is skipped.

Skip		
(omitted)	000	Never skip.
SKP	001	Always skip.
SZC	010	Skip if carry equals zero.
SNC	011	Skip if carry does not equal zero.
SZR	100	Skip if result equals zero.
SNR	101	Skip if result does not equal zero.
SEZ	110	Skip if either carry or result equals zero.
SBN	111	Skip if both carry and result do not equal zero.

NOTE: *Instructions that specify both "no load" and "no skip" or "no load" and "always skip" will not operate in the expected manner. These bit combinations are used for other instructions in the instruction set.*

This card provides the "base value" for an instruction, along with its name and how to code it. This base value is the 6-digit octal number that represents how the instruction would be assembled if all its options were omitted and all its operands were zero. For example, the base value for ADD is 103000. This represents an ADD 0,0 instruction.

Unsigned Integer Comparisons

SUB#	*acs,acd* ,SZR	Skip if contents of ACS = contents of ACD.
SUB#	*acs,acd* ,SNR	Skip if contents of ACS ≠ contents of ACD.
ADCZ#	*acs,acd* ,SNC	Skip if contents of ACS < contents of ACD.
SUBZ#	*acs,acd* ,SNC	Skip if contents of ACS ≤ contents of ACD.
SUBZ#	*acs,acd* ,SZC	Skip if contents of ACS > contents of ACD.
ADCZ#	*acs,acd* ,SZC	Skip if contents of ACS ≥ contents of ACD.

Standard Instruction Set

Instruction Name	Base Value	Mnemonic and Operands
Add	103000	ADD<*c* ><*sh* ><#>*acs,acd* <,*skip* >
Add Complement	102000	ADC<*c* ><*sh* ><#>*acs,acd* <,*skip* >
Add Immediate	100010	ADI *n,ac*
AND	103400	AND<*c* ><*sh* ><#>*acs,acd* <,*skip* >
AND Immediate	143770	ANDI *i,ac*
AND with Complemented Source	100610	ANC *acs,acd*
Block Add and Move	113710	BAM
Block Move	133710	BLM
Character Compare	157650	CMP
Character Move	153650	CMV
Character Move Until True	167650	CMT
Character Translate	163650	CTR
Compare to Limits	102370	CLM *acs,acd*
Complement	100000	COM<*c* ><*sh* ><#>*acs,acd* <,*skip* >
Count Bits	102610	COB *acs,acd*
Decimal Add	100210	DAD *acs,acd*
Decimal Subtract	100310	DSB *acs,acd*
Decrement and Skip if Zero	014000	DSZ <@>*displacement* <,*index* >
Dispatch	142710	DSPA *ac,*<@>*displacement* <,*index* >
Double Hex Shift Left	101610	DHXL *n,ac*
Double Hex Shift Right	101710	DHXR *n,ac*
Double Logical Shift	101310	DLSH *acs,acd*
Exchange Accumulators	100710	XCH *acs,acd*
Exclusive OR	100510	XOR *acs,acd*
Exclusive OR Immediate	123770	XORI *i,ac*
Execute	123370	XCT *ac*
Extended Add Immediate	163770	ADDI *i,ac*
Extended Decrement and Skip if Zero	116070	EDSZ <@>*displacement* <,*index* >
Extended Increment and Skip if Zero	112070	EISZ <@>*displacement* <,*index* >
Extended Jump	102070	EJMP <@>*displacement* <,*index* >
Extended Jump to Subroutine	106070	EJSR <@>*displacement* <,*index* >
Extended Load Accumulator	122070	ELDA *ac,*<@>*displacement* <,*index* >
Extended Load Byte	102170	ELDB *ac,displacement* <,*index* >
Extended Operation	100030	XOP *acs,acd,operation number*
Extended Operation 1	100070	XOP1 *acs,acd,entry number*
Extended Store Accumulator	142070	ESTA *ac,*<@>*displacement* <,*index* >
Extended Store Byte	122170	ESTB *ac,displacement* <,*index* >
Halve	143370	HLV *ac*
Hex Shift Left	101410	HXL *n,ac*
Hex Shift Right	101510	HXR *n,ac*
Inclusive OR	100410	IOR *acs,acd*
Inclusive OR Immediate	103770	IORI *i,ac*
Increment	101400	INC<*c* ><*sh* ><#>*acs,acd* <,*skip* >
Increment and Skip if Zero	010000	ISZ <@>*displacement* <,*index* >
Jump	000000	JMP <@>*displacement* <,*index* >
Jump to Subroutine	004000	JSR <@>*displacement* <,*index* >
Load Accumulator	020000	LDA *ac,*<@>*displacement* <,*index* >
Load Byte	102710	LDB *acs,acd*
Load Effective Address (All Modes)	162070	ELEF *ac,*<@>*displacement* <,*index* >
Load Effective Address (User Mode)	060000	LEF *ac,*<@>*displacement* <,*index* >
Load Map	113410	LMP
Locate and Reset Lead Bit	102510	LRB *acs,acd*
Locate Lead Bit	102410	LOB *acs,acd*
Logical Shift	101210	LSH *acs,acd*
Modify Stack Pointer	103370	MSP *ac*
Move	101000	MOV<*c* ><*sh* ><#>*acs,acd* <,*skip* >
Negate	100400	NEG<*c* ><*sh* ><#>*acs,acd* <,*skip* >
Pop Block	107710	POPB
Pop Multiple Accumulators	103210	POP *acs,acd*
Pop PC and Jump	117710	POPJ
Push Jump	102270	PSHJ
Push Multiple Accumulators	103110	PSH *acs,acd*
Push Return Address	103710	PSHR
Restore	167710	RSTR
Return	127710	RTN
Save	163710	SAVE *i*
Set Bit to One	102010	BTO *acs,acd*
Set Bit to Zero	102110	BTZ *acs,acd*
Sign Extend and Divide	137710	DIVX
Signed Divide	157710	DIVS
Signed Multiply	147710	MULS
Skip if ACS > ACD	101010	SGT *acs,acd*
Skip if ACS ≥ ACD	101110	SGE *acs,acd*
Skip on Nonzero Bit	102770	SNB *acs,acd*
Skip on Zero Bit	102210	SZB *acs,acd*
Skip on Zero Bit and Set to One	102310	SZBO *acs,acd*
Store Accumulator	040000	STA *ac,*<@>*displacement* <,*index* >
Store Byte	103010	STB *acs,acd*
Subtract	102400	SUB<*c* ><*sh* ><#> *acs,acd* <,*skip* >
Subtract Immediate	100110	SBI *n,ac*
System Call	103510	SYC *acs,acd*
Unsigned Divide	153710	DIV
Unsigned Multiply	143710	MUL

*Character instructions (CMP through CTR) are optional or omitted on some ECLIPSE computers.

Floating-Point Instruction Set

Instruction Name	Base Value	Mnemonic and Operands
Absolute Value	143050	FAB *fpac*
Add Double (FPAC)	100150	FAD *facs,facd*
Add Double (Memory)	101150	FAMD *facd,*<@>*displacement* <,*index* >
Add Single (FPAC)	100050	FAS *facs,facd*
Add Single (Memory)	101050	FAMS *facd,*<@>*displacement* <,*index* >
Clear Errors	153350	FCLE
Compare Floating-Point	103450	FCMP *facs,facd*
Divide Double (FPAC)	100750	FDD *facs,facd*
Divide Double (Memory)	101750	FDMD *facd,*<@>*displacement* <,*index* >
Divide Single (FPAC)	100650	FDS *facs,facd*
Divide Single (Memory)	101650	FDMS *facd,*<@>*displacement* <,*index* >
Fix to AC	102650	FFAS *ac,fpac*
Fix to Memory	102750	FFMD *fpac,*<@>*displacement* <,*index* >
Float from AC	102450	FLAS *ac,fpac*
Float from Memory	102550	FLMD *fpac,*<@>*displacement* <,*index* >
Halve	163150	FHLV *fpac*
Integerize	143150	FINT *fpac*
Load Double	102150	FLDD *fpac,*<@>*displacement* <,*index* >
Load Exponent	123150	FEXP *fpac*
Load Single	102050	FLDS *fpac,*<@>*displacement* <,*index* >
Load Status	123350	FLST <@>*displacement* <,*index* >
Move Floating-Point	103550	FMOV *facs,facd*
Multiply Double (FPAC)	100550	FMD *facs,facd*
Multiply Double (Memory)	101550	FMMD *facd,*<@>*displacement* <,*index* >
Multiply Single {FPAC)	100450	FMS *facs,facd*
Multiply Single (Memory)	101450	FMMS *facd,*<@>*displacement* <,*index* >
Negate	163050	FNEG *fpac*
No Skip	103250	FNS
Normalize	103050	FNOM *fpac*
Pop Floating-Point State	167350	FPOP
Push Floating-Point State	163350	FPSH
Read High Word	123050	FRH *fpac*
Scale	103150	FSCAL *fpac*
Skip Always	107250	FSA
Skip on EQ Zero	113250	FSEQ
Skip on GE Zero	127250	FSGE
Skip on GT Zero	137250	FSGT
Skip on LE Zero	133250	FSLE
Skip on LT Zero	123250	FSLT
Skip on NE Zero	117250	FSNE
Skip on No DVZ	147250	FSND
Skip on No Error	177250	FSNER
Skip on No MOF	143250	FSNM
Skip on No OVF	163250	FSNO
Skip on No OVF and No DVZ	167250	FSNOD
Skip on No UNF	153250	FSNU
Skip on No UNF and No DVZ	157250	FSNUD
Skip on No UNF and No OVF	173250	FSNUO
Store Double	102350	FSTD *fpac,*<@>*displacement* <,*index* >
Store Single	102250	FSTS *fpac,*<@>*displacement* <,*index* >
Store Status	103350	FSST <@>*displacement* <,*index* >
Subtract Double (FPAC)	100350	FSD *facs,facd*
Subtract Double (Memory)	101350	FSMD *facd,*<@>*displacement* <,*index* >
Subtract Single (FPAC)	100250	FSS *facs,facd*
Subtract Single (Memory)	101250	FSMS *facd,*<@>*displacement* <,*index* >
Trap Disable	147350	FTD
Trap Enable	143350	FTE

Octal to Decimal Conversion

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

	8^5	8^4	8^3	8^2	8^1	8^0
0	0	0	0	0	0	0
1	32,768	4,096	512	64	8	1
2	65,536	8,192	1,024	128	16	2
3	98,304	12,288	1,536	192	24	3
4	131,072	16,384	2,048	256	32	4
5	163,840	20,480	2,560	320	40	5
6	196,608	24,576	3,072	384	48	6
7	229,376	28,672	3,584	448	56	7

I/O Instruction Set

The instructions that use the I/O format can set bits 8 and 9 by coding optional operands. The table lists the characters to be coded and their results.

Class Abbreviation	Coded Character	Result Bits	Operation
f	(omitted)	00	Do not affect the Busy and Done flags.
	S	01	Set the Busy flag to one and the Done flag to zero—starts the device.
	C	10	Set the Busy flag to zero and the Done flag to zero—idles the device.
	P	11	Pulse the I/O bus control line—the effect, if any, depends on the device.
t	BN	00	If the Busy flag does not equal zero, the next sequential instruction is skipped.
	BZ	01	If the Busy flag equals zero, the next sequential instruction is skipped.
	DN	10	If the Done flag does not equal zero, the next sequential instruction is skipped.
	DZ	11	If the Done flag equals zero, the next sequential instruction is skipped.

Instruction Name	Op Code	Mnemonic and Operands
Data In A	1	DIA<*f* > *ac,device*
Data In B	3	DIB<*f* > *ac,device*
Data In C	5	DIC<*f* > *ac,device*
Data In Status	7	DIS<*f* > *ac,device* [1]
Data Out A	2	DOA<*f* > *ac,device*
Data Out B	4	DOB<*f* > *ac,device*
Data Out C	6	DOC<*f* > *ac,device*
I/O Skip	7	SKP<*t* > *device*[2]
No I/O Transfer	0	NIO<*f* > *device*[2]

[1]Applies only to ECLIPSE S/20 and S/120. DIS<*f* >0,*device* has the same effect as SKP<*f* > *device*.

[2]For this instruction, the AC field (bits 3 and 4), must not be set to zero.

Reserved Storage Locations

Location (octal)	Name	Contents
0	I/O RETURN	Return address from I/O interrupt.
1	I/O HANDLER	Address of I/O interrupt handler.
2	SC HANDLER	Address of system call handler.
3	PF HANDLER	Address of protection fault handler.
4	VCT STACK PTR	Address of beginning of VCT stack.
5	CURRENT MASK	Current interrupt priority mask.
6	VCT STACK LMT	Point in VCT stack where overflow occurs.
7	VCT STACK FLT	Address of VCT stack fault handler.
10	CONTEXT BLK PTR	Pointer to context block saved at the time of a page fault or hardware breakpoint.*
11	TRAP HANDLER	Address of emulator trap handler.
40	STACK POINTER	Address of the last word placed on the stack.
41	FRAME POINTER	Address of start of current frame minus one.
42	STACK LIMIT	Point in stack where stack overflow occurs.
43	STACK FAULT	Address of stack fault handler.
44	XOP ORIGIN	Address of the beginning of the XOP table.
45	FP FAULT	Address of floating-point fault handler.
46	—	Reserved for future use.
47	—	Reserved for future use.

*Applies only to ECLIPSE M/600.

CPU Instructions

Instruction Name	Mnemonic	I/O Instruction	Base Value
CPU Skip		SKP <*t* >CPU	063477
Halt	HALTA *ac*	DOC *ac* ,CPU	063077
Interrupt Acknowledge	INTA *ac*	DIB *ac* ,CPU	061477
Interrupt Disable	INTDS	NIOC CPU	060277
Interrupt Enable	INTEN	NIOS CPU	060177
I/O Reset	IORST	DICC 0,CPU	062677
Mask Out	MSKO *ac*	DOB *ac* ,CPU	062077
Read Processor Status[1]		DIS *ac* ,CPU	067477
Read Virtual Console Register[1]	READS *ac*	DIA *ac* ,CPU	060477
Vector	VCT [@]*displacement* [,*index*]	DIBP 0,CPU	061777

[1]Applies only to the ECLIPSE S/20 and S/120.