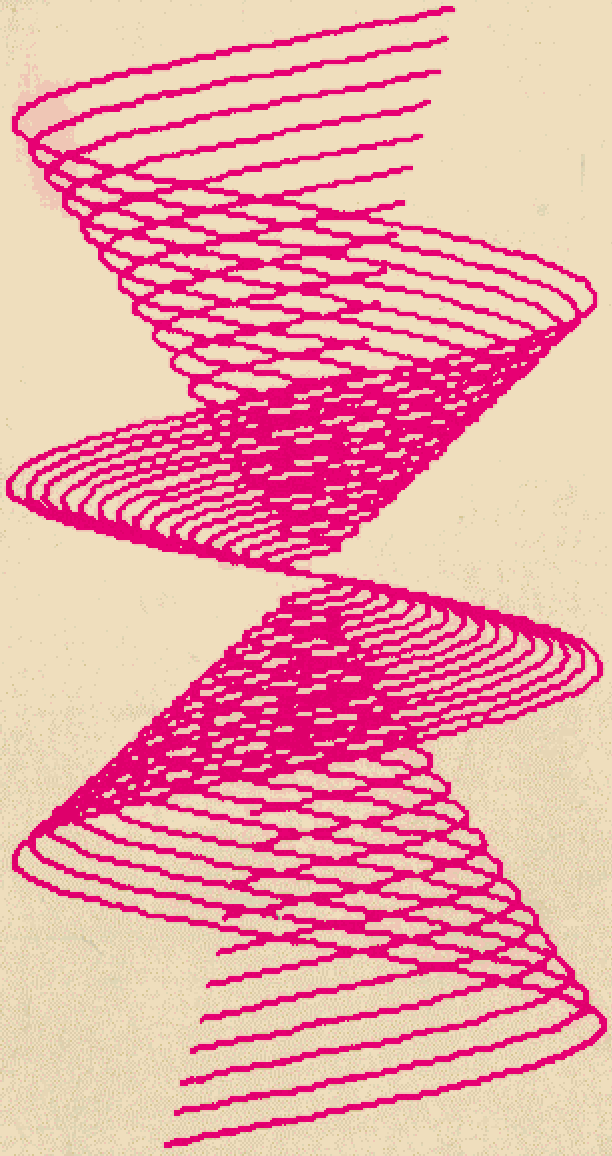
8510/a

GRAPHICS COMPUTER SYSTEM



terak
CORPORATION

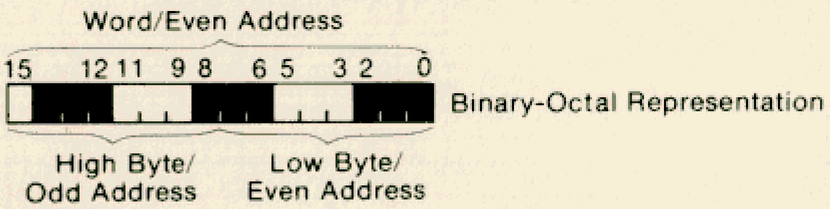
14405 NORTH SCOTTSDALE ROAD
SCOTTSDALE, ARIZONA 85254
(602) 991-1580

Publication No. 60-0022-001

8510/a

Reference Card

WORD FORMAT



GENERAL REGISTER ADDRESSING

MODE | R

Mode	Name	Symbolic	Description
0	register	R	(R) is operand [ex. R2=%2]
1	register deferred	(R)	(R) is address
2	auto-increment	(R)+	(R) is adrs; (R)+(1 or 2)
3	auto-incr deferred	@(R)+	(R) is adrs of adrs; (R)+2
4	auto-decrement	—(R)	(R)—(1 or 2); is adrs
5	auto-decr deferred	@—(R)	(R)—2; (R) is adrs of adrs
6	index	X(R)	(R)+X is adrs
7	index deferred	@X(R)	(R)+X is adrs of adrs

PROGRAM COUNTER ADDRESSING

MODE | 7

2	immediate	#n	operand n follows instr
3	absolute	@#A	address A follows instr
6	relative	A	instr adrs+4+X is adrs
7	relative deferred	@A	instr adrs+4+X is adrs of adrs

LEGEND

Op Codes

■	= 0 for word/1 for byte
SS	= source field (6 bits)
DD	= destination field (6 bits)
R	= general register (3 bits), 0 to 7
XXX	= offset (8 bits), +127 to -128
N	= number (3 bits)
NN	= number (6 bits)

Operations

( )	= contents of
s	= contents of source
d	= contents of destination
r	= contents of register
l,h	= double word register l=low order h=high order
←	= becomes
X	= relative address
%	= register definition

Boolean

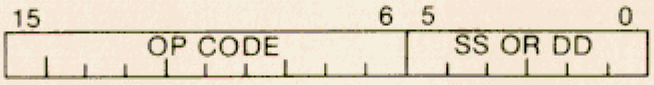
∧	= AND
∨	= inclusive OR
⊻	= exclusive OR
¬	= NOT

PC	= Program Counter (R7)
SP	= Stack Pointer (R6)
PS	= Processor Status Word

Condition Codes

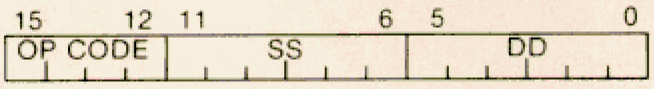
*	= conditionally set/cleared
—	= not affected
0	= cleared
1	= set

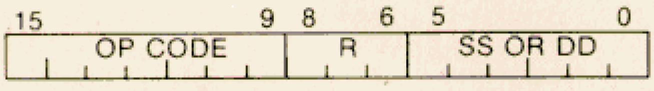
SINGLE OPERAND **OPR dst**



Mne- monic	Op Code	Instruction	dst Result	N	Z	V	C
General							
CLR(B)	■ 050DD	clear	0	0	1	0	0
COM(B)	■ 051DD	complement (1's)	∼ d	•	•	0	1
INC(B)	■ 052DD	increment	d+1	•	•	•	—
DEC(B)	■ 053DD	decrement	d-1	•	•	•	—
NEG(B)	■ 054DD	negate (2's compl)	-d	•	•	•	•
TST(B)	■ 057DD	test	d	•	•	0	0
Rotate & Shift							
ROR(B)	■ 060DD	rotate right	→ C, d	•	•	•	•
ROL(B)	■ 061DD	rotate left	C, d ←	•	•	•	•
ASR(B)	■ 062DD	arith shift right	d/2	•	•	•	•
ASL(B)	■ 063DD	arith shift left	2d	•	•	•	•
SWAB	0003DD	swap bytes		•	•	0	0
Multiple Precision							
ADC(B)	■ 055DD	add carry	d+C	•	•	•	•
SBC(B)	■ 056DD	subtract carry	d-C	•	•	•	•
SXT	0067DD	sign extend	0 or -1	—	•	0	—
Processor Status (PS) Operators							
MFPS	1067DD	move byte from PS	d ← PS	•	•	0	—
MTPS	1064SS	move byte to PS	PS ← s	•	•	•	•

DOUBLE OPERAND **OPR src, dst** **OPR src, R or OPR R, dst**





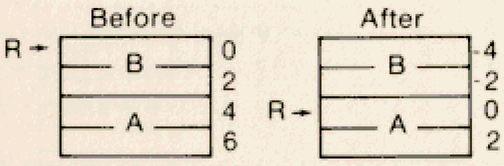
Mne- monic	Op Code	Instruction	Operation	N	Z	V	C
General							
MOV(B)	■ 1SSDD	move	d ← s	•	•	0	—
CMP(B)	■ 2SSDD	compare	s − d	•	•	•	•
ADD	06SSDD	add	d ← s +d	•	•	•	•
SUB	16SSDD	subtract	d ← d − s	•	•	•	•
Logical							
BIT(B)	■ 3SSDD	bit test (AND)	s ∧ d	•	•	0	—
BIC(B)	■ 4SSDD	bit clear	d ← (∼ s) ∧ d	•	•	0	—
BIS(B)	■ 5SSDD	bit set (OR)	d ← s ∨ d	•	•	0	—
XOR	074RDD	exclusive OR	d ← r ⊻ d	•	•	0	—

EXTENDED INSTRUCTION SET

Mnemonic	Op Code	Instruction	Operation	N	Z	V	C
MUL	070RSS	multiply	If r even: (rv1, r) ← r×s if r odd: r ← r×s	•	•	0	•
DIV	071RSS	divide	r is even: r ← (rv1, r) div s rv1 ← (rv1, r) mod s	•	•	•	•
ASH	072RSS	arithmetic shift	r ← r×$2^{(ss)}$	•	•	•	•
ASHC	073RSS	arith shift combined	(rv1, r) ← (rv1, r)×$2^{(ss)}$	•	•	•	•

FLOATING POINT INSTRUCTION SET

				N	Z	V	C
FADD	07500R	floating add	A ← A op B (R) ← (R) + 4	•	•	0	0
FSUB	07501R	floating subtract					
FMUL	07502R	floating multiply					
FDIV	07503R	floating divide					

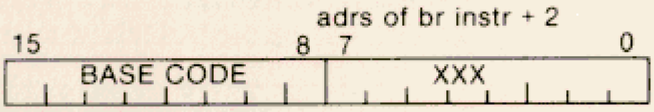


If error detected:
(R) not affected,
FIStrap (PC at 244, PS at 246)
and codes set as:

	N	Z	V	C
overflow	0	0	1	0
underflow	1	0	1	0
divide by 0	1	0	1	1

BRANCH B -- location

If condition is satisfied: Branch to location,
New PC ← $\underbrace{\text{Updated PC}}_{\text{adrs of br instr + 2}}$ + (2 × offset)



Op Code = Base Code + XXX

Mnemonic	Base Code	Instruction	Branch Condition	
Branches				
BR	000400	branch (unconditional)	(always)	
BNE	001000	br if not equal (to 0)	≠ 0	Z = 0
BEQ	001400	br if equal (to 0)	= 0	Z = 1
BPL	100000	branch if plus	+	N = 0
BMI	100400	branch if minus	—	N = 1
BVC	102000	br if overflow is clear		V = 0
BVS	102400	br if overflow is set		V = 1
BCC	103000	br if carry is clear		C = 0
BCS	103400	br if carry is set		C = 1
Signed Conditional Branches				
BGE	002000	br if greater or equal (to 0)	≥ 0	$N \veebar V = 0$
BLT	002400	br if less than (0)	< 0	$N \veebar V = 1$
BGT	003000	br if greater than (0)	> 0	$Z \vee (N \veebar V) = 0$
BLE	003400	br if less or equal (to 0)	≤ 0	$Z \vee (N \veebar V) = 1$
Unsigned Conditional Branches				
BHI	101000	branch if higher	>	$C \vee Z = 0$
BLOS	101400	branch if lower or same	≤	$C \vee Z = 1$
BHIS	103000	branch if higher or same	≥	C = 0
BLO	103400	branch if lower	<	C = 1

JUMP & SUBROUTINE

Mnemonic	Op Code	Instruction	Notes
JMP	0001DD	jump	PC ← dst
JSR	004RDD	jump to subroutine	use same R
RTS	00020R	return from subroutine	
MARK	0064NN	mark	aid in subr return
SOB	077RNN	subtract 1 & br (if ≠ 0)	(R) — 1, then if (R) ≠ 0: PC ← Updated PC — (2 × NN)

TRAP & INTERRUPT

Mnemonic	Op Code	Instruction	Notes
EMT	104000 to 104377	emulator trap (not for general use)	PC at 30, PS at 32
TRAP	104400 to 104777	trap	PC at 34, PS at 36
BPT	000003	breakpoint trap	PC at 14, PS at 16
IOT	000004	input/output trap	PC at 20, PS at 22
RTI	000002	return from interrupt	
RTT	000006	return from interrupt	inhibit T bit trap

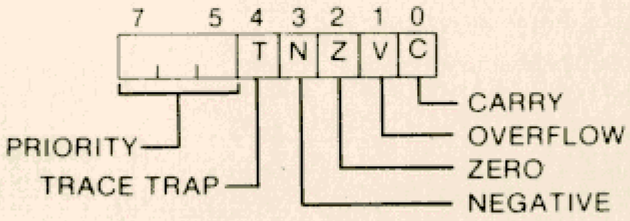
MISCELLANEOUS

Mnemonic	Op Code	Instruction
HALT	000000	halt
WAIT	000001	wait for interrupt
RESET	000005	reset external bus
NOP	000240	(no operation)

CONDITION CODE OPERATORS

Mnemonic	Op Code	Instruction	N	Z	V	C
CLC	000241	clear C	—	—	—	0
CLV	000242	clear V	—	—	0	—
CLZ	000244	clear Z	—	0	—	—
CLN	000250	clear N	0	—	—	—
CCC	000257	clear all cc bits	0	0	0	0
SEC	000261	set C	—	—	—	1
SEV	000262	set V	—	—	1	—
SEZ	000264	set Z	—	1	—	—
SEN	000270	set N	1	—	—	—
SCC	000277	set all cc bits	1	1	1	1

PROCESSOR STATUS WORD



NUMERICAL OP CODE LIST

OP Code	Mnemonic
00 00 00	HALT
00 00 01	WAIT
00 00 02	RTI
00 00 03	BPT
00 00 04	IOT
00 00 05	RESET
00 00 06	RTT
00 00 07 00 00 77	(unused)
00 01 DD	JMP
00 02 0R	RTS
00 02 10 ↕ 00 02 27	(reserved)
00 02 40	NOP
00 02 41 ↕ 00 02 77	cond codes
00 03 DD	SWAB
00 04 XXX	BR
00 10 XXX	BNE
00 14 XXX	BEQ
00 20 XXX	BGE
00 24 XXX	BLT
00 30 XXX	BGT
00 34 XXX	BLE
00 4R DD	JSR
00 50 DD	CLR
00 51 DD	COM
00 52 DD	INC
00 53 DD	DEC
00 54 DD	NEG
00 55 DD	ADC
00 56 DD	SBC
00 57 DD	TST

OP Code	Mnemonic
00 60 DD	ROR
00 61 DD	ROL
00 62 DD	ASR
00 63 DD	ASL
00 64 NN	MARK
00 67 DD	SXT
00 70 00 ↕ 00 77 77	(unused)
01 SS DD	MOV
02 SS DD	CMP
03 SS DD	BIT
04 SS DD	BIC
05 SS DD	BIS
06 SS DD	ADD
07 0R SS	MUL
07 1R SS	DIV
07 2R SS	ASH
07 3R SS	ASHC
07 4R DD	XOR
07 50 0R	FADD
07 50 1R	FSUB
07 50 2R	FMUL
07 50 3R	FDIV
07 50 40 ↕ 07 67 77	(unused)
07 7R NN	SOB
10 00 XXX	BPL
10 04 XXX	BMI
10 10 XXX	BHI
10 14 XXX	BLOS
10 20 XXX	BVC
10 24 XXX	BVS
10 30 XXX	BCC BHIS
10 34 XXX	BCS, BLO

OP Code	Mnemonic
10 40 00 ↕ 10 43 77	EMT
10 44 00 ↕ 10 47 77	TRAP
10 50 DD	CLRB
10 51 DD	COMB
10 52 DD	INCB
10 53 DD	DECB
10 54 DD	NEGB
10 55 DD	ADCB
10 56 DD	SBCB
10 57 DD	TSTB
10 60 DD	RORB
10 61 DD	ROLB
10 62 DD	ASRB
10 63 DD	ASLB
10 64 SS	MTPS
10 67 DD	MFPS
11 SS DD	MOVB
12 SS DD	CMPB
13 SS DD	BITB
14 SS DD	BICB
15 SS DD	BISB
16 SS DD	SUB
17 00 00 ↕ 17 77 77	RESERVED

RESERVED LOW MEMORY, TRAP AND INTERRUPT VECTORS

000	(Reserved)
004	Bus Timeout and Illegal Instructions (eg. JMP R0) (Odd Address and Stack Overflow Traps Not Implemented)
010	Illegal and Reserved Instruction
014	BPT Instruction and T Bit
020	IOT Instruction
024	Power Fail
030	EMT Instruction
034	TRAP Instruction
40 thru 56	reserved for system communication
060	Console Input Device
064	Console Output Device
074	Console Emulator, Alt
100	Vertical Retrace Start
120, 124	Serial Interface Unit 1
130, 134	Serial Interface Unit 2
150, 154	Serial Interface Unit 3
164	Console Emulator

7-BIT ASCII CODE

Octal Code	Char	Octal Code	Char	Octal Code	Char	Octal Code	Char
000	NUL	040	SP	100	@	140	`
001	SOH	041	!	101	A	141	a
002	STX	042	"	102	B	142	b
003	ETX	043	#	103	C	143	c
004	EOT	044	$	104	D	144	d
005	ENQ	045	%	105	E	145	e
006	ACK	046	&	106	F	146	f
007	BEL	047	'	107	G	147	g
010	BS	050	(	110	H	150	h
011	HT	051	)	111	I	151	i
012	LF	052	*	112	J	152	j
013	VT	053	+	113	K	153	k
014	FF	054	,	114	L	154	l
015	CR	055	-	115	M	155	m
016	SO	056	.	116	N	156	n
017	SI	057	/	117	O	157	o
020	DLE	060	0	120	P	160	p
021	DC1	061	1	121	Q	161	q
022	DC2	062	2	122	R	162	r
023	DC3	063	3	123	S	163	s
024	DC4	064	4	124	T	164	t
025	NAK	065	5	125	U	165	u
026	SYN	066	6	126	V	166	v
027	ETB	067	7	127	W	167	w
030	CAN	070	8	130	X	170	x
031	EM	071	9	131	Y	171	y
032	SUB	072	:	132	Z	172	z
033	ESC	073	;	133	[	173	{
034	FS	074	<	134	\	174	|
035	GS	075	=	135	]	175	}
036	RS	076	>	136	^	176	~
037	US	077	?	137	—	177	DEL

EMULATOR CONTROL CODES

Octal Code	Function	Octal Code	Function	Octal Code	Function
4	Dec Pan Rate	16	Upper Charset	33, 106, Y+40, X+40	
5	Inc Pan Rate	17	Lower Charset		Cursor Address
6	Click	25	Cursor Off	33, 110	Home Cursor
7	Bell	26	Cursor On	33, 112	EOS Clear
10	Backspace	30	Cancel Seq	33, 113	EOL Clear
11	Tab	33, 101	Cursor Up	33, 127	Format On
12	Line Feed	33, 102	Cursor Down	33, 130	Format Off
13	Reverse LF	33, 103	Cursor Right	33, 133	Lower Charset
14	Reset & Clear	33, 104	Cursor Left	33, 135	Upper Charset
15	Return	33, 105	Clear	177, 377	Reserved

VECTORS (CONT'D)

174	Console Emulator, Alt
200	Line Printer #0
240	Line Printer #1
244	FIS Error Trap
250	QX Floppy Disk
264	RX Floppy Disk
300, 304	Parallel Interface Unit 0
310, 314	Parallel Interface Unit 1
320, 324	Serial Interface Unit 4
330, 334	Serial Interface Unit 5
340, 344	Serial Interface Unit 6
350, 354	Serial Interface Unit 7

STACK SEQUENCES

TRAP AND INTERRUPT

(SP) ← (SP) -2
((SP)) ← (PS)
(SP) ← (SP) -2
((SP)) ← (PC)
(PS) ← (Vector +2)
(PC) ← (Vector)

JSR R,DD

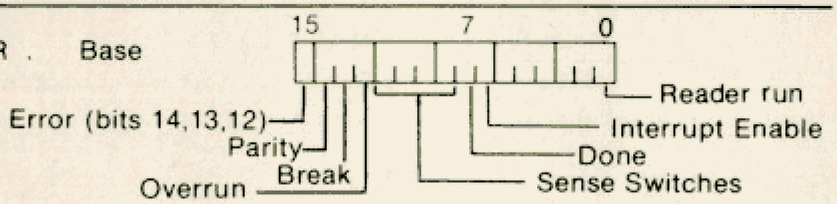
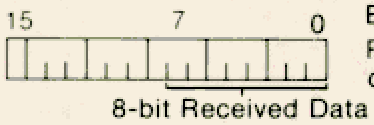
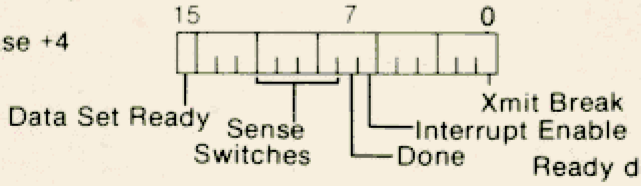
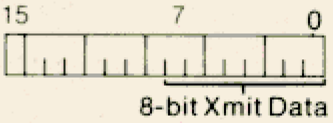
(SP) ← (SP) -2
((SP)) ← (R)
(R) ← (PC)
(PC) ← (DD)

MICRO-ODT COMMANDS

Note: To use ODT, system console must be a serial interface set to unit 0. Emulated console cannot be used (set to ALT).

Format	Octal Code	Description
RETURN	015	Close opened location and accept next command.
LINE FEED	012	Close current location; open next sequential location.
>or]	135	Open previous location.
← or —	137	Take contents of opened location, index by opened location plus 2, and open that location.
@	100	Take contents of opened location as an absolute address and open that location.
r/	057	Open location r.
/	057	Reopen last location.
$n or Rn	044 or 122	Open general register n (0-7) or S (PS register).
r;G or rG	073 107 or 107	Go to location r, initialize the bus, and start program.
nL		Execute bootstrap loader using n as device CSR address.
;P or P	073 120 or 120	Proceed with program execution.
RUBOUT or DELete	177	Erase previous character. Response is a backslash (134) each time RUBOUT is entered.
M	115	Maintenance. Display of an internal CPU register follows the M command. Only the last digit displayed is significant, indicating how the CPU entered the Halt (ODT) mode, as follows:
	Last Digit	Halt Source
	0 or 4	HALT instruction or BHALT L bus signal.
	1 or 5	Bus error occurred while getting device interrupt vector.
	2 or 6	Bus error occurred while doing memory refresh.
	3	Double bus error occurred (stack was non-existent)
	4	Reserved instruction trap occurred (non-existent Micro-PC address occurred on internal CPU bus).
	7	A combination of 1, 2, and 4 occurred.
CTRL-SHIFT-s	023	For manufacturing tests only. Escape this command function by typing NULL and @ (000 and 100).

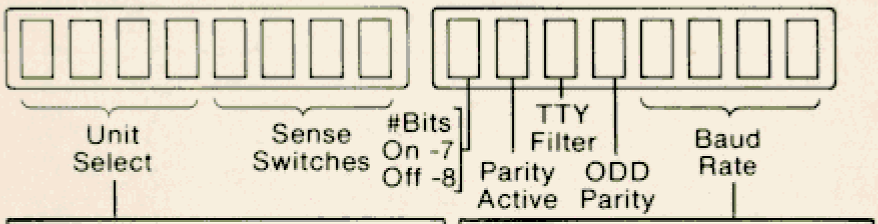
SERIAL INTERFACE REGISTERS

MNE—MONIC	ADDRESS	MAP	NOTES
RSR .	Base		
RBD	Base +2		Error sets done Reading RDB clears done
XSR	Base +4		Ready does not affect done
XDB	Base +6		Writing XDB clears done

SERIAL INTERFACE, RS-232C PINNING AS DCE.

Pin	Signal	Notes
J1-1	Frame GND	
J1-7	Signal GND	
J1-8	Carrier detect	Output, always ON
J1-4	Request to Send	Input, wired to J1-20
J1-3	Receive Data	Output
J1-20	Data Terminal Ready	Input, drives bit 15 of XSR
J1-5	Clear to Send	Output, always ON
J1-6	Data Set Ready	Output, always ON
J1-2	Transmit Data	Input

SERIAL INTERFACE EIB SWITCHES

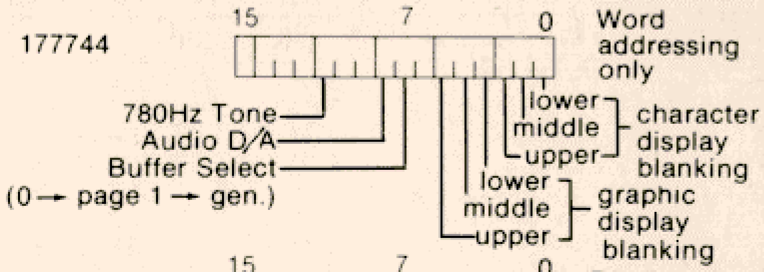


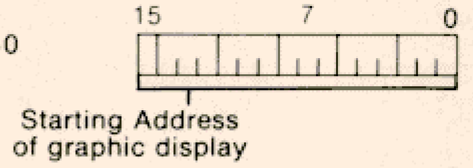
Setting				Unit	Base Address
OFF	OFF	ON	ON	0	177560
ON	OFF	ON	OFF	1	177520
ON	OFF	OFF	OFF	2	177530
OFF	OFF	OFF	OFF	3	177570
ON	ON	ON	OFF	4	176520
ON	ON	OFF	OFF	5	176530
OFF	ON	ON	OFF	6	176560
OFF	ON	OFF	OFF	7	176570

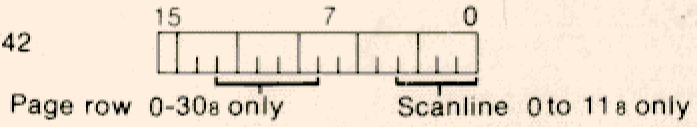
Notes: Never set two serial interfaces to the same unit number. Unit 0 (if selected) requires Video EIB switch to be set to "ALT".

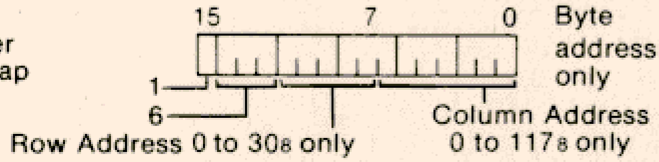
Setting				Rate
ON	ON	ON	OFF	50
ON	ON	OFF	OFF	75
OFF	OFF	OFF	OFF	110
OFF	ON	ON	ON	134
OFF	OFF	ON	OFF	150
OFF	ON	OFF	ON	200
OFF	OFF	OFF	ON	300
OFF	ON	ON	OFF	600
ON	OFF	OFF	OFF	1200
ON	OFF	ON	OFF	1800
OFF	ON	OFF	OFF	2400
ON	OFF	OFF	ON	4800
ON	OFF	ON	ON	9600
ON	ON	ON	ON	19200

GRAPHIC DISPLAY CONTROLLER REGISTERS

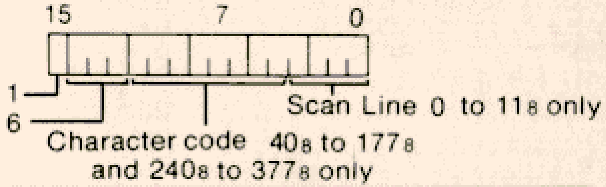
MNE— MONIC	ADDRESS	MAP	NOTES
VCR	177744		Word addressing only

MNE— MONIC	ADDRESS	MAP	NOTES
GAR	177740		Displayed memory must be > 20000 and < 160000

MNE— MONIC	ADDRESS	MAP	NOTES
VIR	177742		

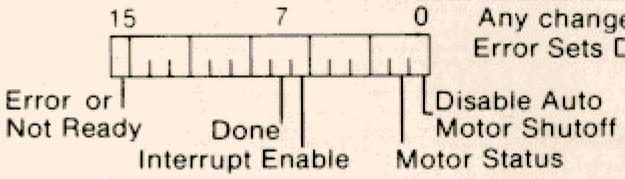
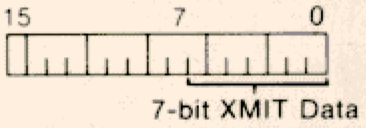
MNE— MONIC	ADDRESS	MAP	NOTES
Page Buffer Address Map			Byte addressing only

15
7
0
1
6
Row Address 0 to 30_8 only
Column Address 0 to 117_8 only

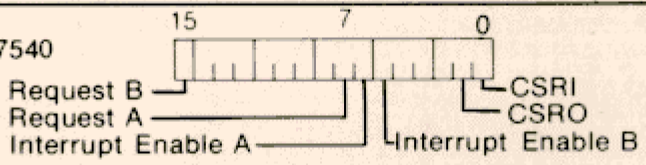
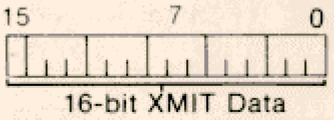
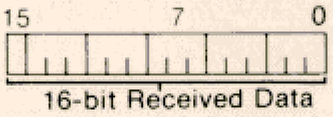
MNE— MONIC	ADDRESS	MAP	NOTES
Generator Buffer Address Map			

15
7
0
1
6
Scan Line 0 to 11_8 only
Character code 40_8 to 177_8 and 240_8 to 377_8 only

LINE PRINTER REGISTERS

MNE- MONIC	ADDRESS	MAP	NOTES
LPCS	177514		Any change to Error Sets Done
LPDB	177516		Unit #1 Base Address = 176514

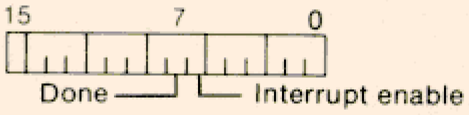
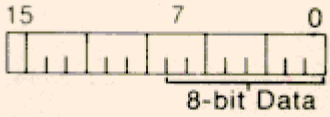
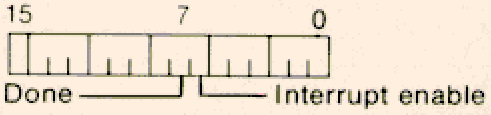
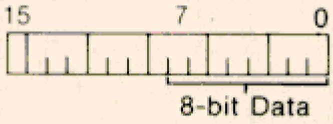
PARALLEL INTERFACE REGISTERS

MNE- MONIC	ADDRESS	MAP	NOTES
DRCS	177540		
DROB	177542		Unit #1 Base Address = 177440
DRIB	177544		

DATA BUS PINNING

Row A (Same as Row C)		Row B (Same as Row D)	
	Module Side 1 (Component Side)		
AA1	BIRQ5 L	BA1	BDCOK H
AB1	BIRQ6 L	BB1	BPOK H
AC1	BDAL16 L	BC1	SSPARE4
AD1	BDAL17 L	BD1	SSPARE5
AE1	SSPARE1	BE1	SSPARE6
AF1	SSPARE2	BF1	SSPARE7
AH1	SSPARE3	BH1	SSPARE8
AJ1	GND	BJ1	GND
AK1	MSPARE A	BK1	MSPARE B
AL1	MSPARE A	BL1	MSPARE B
AM1	GND	BM1	GND
AN1	BDMR L	BN1	BSACK L
AP1	BHALT L	BP1	BIRQ7 L
AR1	BREF L	BR1	BEVNT L
AS1	+12B	BS1	PSPARE4
AT1	GND	BT1	GND
AU1	PSPARE1	BU1	PSPARE2
AV1	+5B	BV1	+5
	Module Side 2 (Solder Side)		
AA2	+5	BA2	+5
AB2	—12	BB2	—12
AC2	GND	BC2	GND
AD2	+12	BD2	+12
AE2	BDOUT L	BE2	BDAL2 L
AF2	BRPLY L	BF2	BDAL3 L
AH2	BDIN L	BH2	BDAL4 L
AJ2	BSYNC L	BJ2	BDAL5 L
AK2	BWTBT L	BK2	BDAL6 L
AL2	BIRQ4 L	BL2	BDAL7 L
AM2	BIAKI L	BM2	BDAL8 L
AN2	BIAKO L	BN2	BDAL9 L
AP2	BBS7 L	BP2	BDAL10 L
AR2	BDMGI L	BR2	BDAL11 L
AS2	BDMGO L	BS2	BDAL12 L
AT2	BINIT L	BT2	BDAL13 L
AU2	BDAL0 L	BU2	BDAL14 L
AV2	BDAL1 L	BV2	BDAL15 L

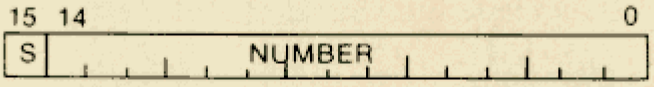
CONSOLE EMULATOR AND KEYBOARD REGISTERS

MNE— MONIC	ADDRESS	MAP	NOTES
KSR	177560		
KDB	177562		Alternate Base Address = 177760
ESR	177564		
EDB	177566		Read 64/74 vector Write 164/174 vector

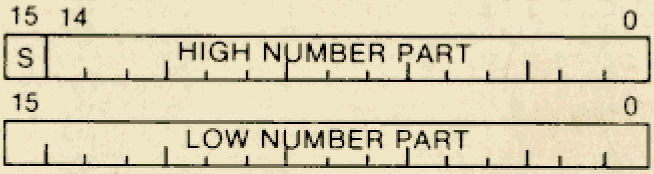
OPERAND FORMATS

FIXED POINT DATA

16-bit single word:

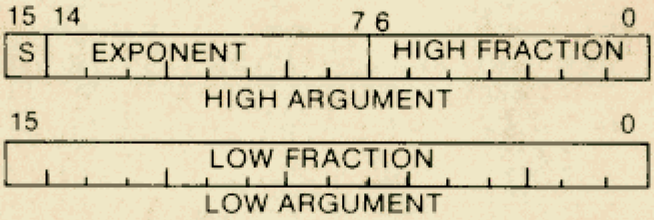


32 -bit double word:



S is the sign bit. S = 0 for positive quantities
S = 1 for negative quantities; number is in 2's complement notation

FLOATING POINT DATA



S = sign of fraction; 0 for positive, 1 for negative
Exponent = 8 bits for the exponent, in excess (200) notation
Fraction = 23 bits plus 1 hidden bit (all numbers are assumed to be normalized)

VARIOUS NUMBERS

320 × 240 = 76800	177520_8 = -176	π = 040511.007732_8
24 × 80 = 1920	160000_8 = -8192	e = 040455.174125_8
25 × 80 = 2000	177740_8 = -32	1 = 040200.000000_8

POWERS OF 2

n	2^n	n	2^n
0	1	10	1,024
1	2	11	2,048
2	4	12	4,096
3	8	13	8,192
4	16	14	16,384
5	32	15	32,768
6	64	16	65,536
7	128	17	131,072
8	256	18	262,144
9	512	19	524,288

EMULATOR CONTROL CODES (for Pascal O/S)

Decimal Code	Function	Decimal Code	Function
7	Bell	13	Return
8	Backspace/Cursor Left	25	Home Cursor
9	Tab	28	Forespace / Cursor Right
10	Line Feed/Cursor Down	29	EOL Clear
11	EOS Clear	30, X+32, Y+32	Cursor Address
12	Reset & Clear	31	Reverse LF/Cursor Up

ARROW KEY CODES IN OCTAL (DECIMAL)

△ 32 (26)   ▽ 14 (12)   ◁ 27 (23)   ▷ 13 (11)

5M/DP/379/TS&F   LITHO IN U.S.A.