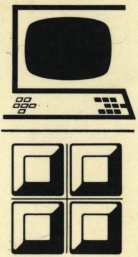
Series IV
Reference Summary



FOUR-PHASE SYSTEMS, INC.

ISSUED 15 NOVEMBER 1981

DOCUMENT SIV/70-04-1F

STOCK NO. 87600084F

SERIES IV REFERENCE SUMMARY

This booklet contains information at the assembly language and machine code levels for all Series IV processors. It is intended as a quick reference for the programmer or engineer familiar with the Four-Phase Systems product line.

In this guide, unless otherwise indicated, the data for the IV/70 refer equally to the following Series IV processors: Systems IV/10, IV/40, IV/50, and IV/70. Similarly, data for the IV/90 refer equally to the Series IV High-Speed processors: Systems IV/60, IV/65, IV/80, IV/90, and IV/95.

This booklet supersedes the System IV/70 Machine Instruction Card (SIV/70-04-1E). It adds the following IV/90 instructions: bit and byte manipulation, binary/decimal conversion, the instruction (MVEL) to move long blocks, the I/O cross window feature, and map instructions. Macro assembler (MACROL) directives and calls have also been added.

This booklet also contains operation codes, mnemonics, machine instructions, and instruction formats as well as the code executed for the various condition tests. It gives special word formats, the bootstrap words to use on the console keys, the character set, assembler directives, and macro assembler directives and calls. In addition, this booklet provides the status words and control words for the peripheral devices.

CONTENTS

OPERATION CODES

00x	HLT	10x	ORM	20x	ANM	30x	XOM
01x	LDA1	11x	INR	21x	DEC	31x	STA1
02x	LD23	12x	ADM	22x	SKN	32x	ST23
03x	LDA	13x	ADA,ADD	23x	SBA,SUB	33x	CPA
04x	LDB	14x	ORA,OR	24x	ANA,AND	34x	XOA,XOR
05x	LD1	15x	AD1	25x	SB1	35x	CP1
06x	LD2	16x	AD2	26x	SB2	36x	CP2
07x	LD3	17x	AD3	27x	SB3	37x	CP3
40x	STZ	50x	SLR	60x	SKZ	70x	XEC
41x	SAM	51x	SLRD	61x	MCC	71x	BRM
42x	STP	52x	SLA	62x	BOF	72x	BRA
43x	STA	53x	SLAD	63x	BZO	73x	BNZ
44x	STB	54x	SRL	64x	BMI	74x	BPL
45x	ST1	55x	SRLD	65x	BCR	75x	BC1
46x	ST2	56x	SRA	66x	BAL	76x	BC2
47x	ST3	57x	SRAD	67x	BGT	77x	BC3
007	LCL	107	ROR	207	RAND	307	RXOR
017	LPL	117	RADD,RCC	217	RSUB	317	RCM2
027	RCL	127	MPY	227	DIV	327	POP
037	RLC	137	MVL	237	MVE	337	UP
047	LCR	147	UFA	247	CDA2	347	IN
057	LPR	157	FAD	257	FSB	357	TRT
067	RCR,RCPY,NOP	167	FMP	267	FDV	367	*
077	RRC	177	MVR	277	IOB	377	BOOT
407	SCL	507	BRD	607	CPN	707	BDEC
417	SPL	517	BRR	617	MVCR	717	DBIN
427	PUSH	527	EXCT	627	DADD	727	BYTE
437	DOWN	537	EXSN	637	CPL	737	MAP
447	SCR	547	PIA	647	DSUB	747	IOXW
457	SPR	557	PID	657	ODD	757	MVEL
467	TRAP	567	PIR	667	MVCL	767	BIT
477	ECS	577	IOID	677	IO777	*	

OPERATION CODES FOR DERIVATIVE INSTRUCTIONS

Byte		Bit		Map	
727xxxxx	BYTE	767xxxxx	BIT	737xxxxx	MAP
727xxx01	LBP	767xx200	BFPA	7370x010	RMAPP
727xxx02	SBP	767xx400	BFAP	737xx000	RWIN
727xxx03	SWBP	767x06wx	LBF	737xx001	RMAP
727xxx04	INRBP	767x06vx	TBFZ	737xx002	SWWIN
727xxx05	LBPI	767x07vx	SBF	737xx0002	WWIN
727xxx06	SBPI	767x07wx	SWBF	737x2002	BRAWIN
727xxx07	SWBPI			737xx003	SWMAP
727xxx10	DECBP			737x0003	WMAP
727xxx11	LBPD			737xx004	RMEMP
727xxx12	SBPD				
727xxx13	SWBPD				
727xxx20	BYPA				
727xxx40	BYAP				
727xxx61	LB				
727xxx62	SB				
727xxx63	SWB				
727xxx65	LBI				
727xxx66	SBI				
727xxx67	SWBI				
727xxx71	LBD				
727xxx72	SBD				
727xxx73	SWBD				
727x0x00	BPINX				
727x0x60	BAINX				

* Unpredictable results are obtained if this op code is executed.
v Means bit 18 = 0
w Means bit 18 = 1
x Varies according to programmer options selected.

MNEMONICS

① Mnemonics	Op Code	Instr Type
ADA	13x	E
ADD	13x	E
ADM	12x	E
AD1	15x	E
AD2	16x	E
AD3	17x	E
ANA	24x	K
AND	24x	K
ANM	20x	K
BAINX	727x0x60	R
BAL	66x	I
BCR	65x	I
BC1	75x	I
BC2	76x	I
BC3	77x	I
BDEC	707	P
BFAP	767xx400	Q
BFPA	767xx200	Q
BGT	67x	I
BIT	767	Q
BMI	64x	I
BNZ	73x	I
BOF	62x	I
BOOT	377	N
BPINX	727x0x00	R
BPL	74x	I
BRA	72x	I
BRAWIN	737x2002	V
BRD	507	I
BRM	71x	I
BRR	517	I
BYAP	727xxx40	R
BYPA	727xxx20	R
BYTE	727xxxxx	R
BZO	63x	I
CDA2	247	J
CPA	33x	G
CPL	637	A
CPN	607	A
CP1	35x	G
CP2	36x	G
CP3	37x	G
DADD	627	A
DBIN	717	P
DEC	21x	I
DECBP	727xxx10	R
DIV	227	E
DOWN	437	C
DSUB	647	A
ECS	477	N
EXCT	527	N
EXSN	537	N
FAD	157	F
FDV	267	F
FMP	167	F
FSB	257	F
HLT	00x	L
IN	347	C
INR	11x	I
INRBP	727xxx04	R
IO	677	N
IOB	277	N
IOID	577	M
IOXW	747	N
LB	727xxx61	S
LBD	727xxx71	S
LBF	767x06wx	Q
LBI	727xxx65	S
LBP	727xxx01	S
LBPD	727xxx11	S
LBPI	727xxx05	S
LCL	007	B
LCR	047	B
LDA	03x	D
LDA1	01x	D
LDB	04x	D
LD1	05x	D
LD2	06x	D
LD3	07x	D
LD23	02x	D
LPL	017	B
LPR	057	B
MAP	737xxxxx	V
MCC	61x	I
MPY	127	E
MVCL	667	A
MVCR	617	A
MVE	237	B
MVEL	757	B
MVL	137	B
MVR	177	B
NOP	067	L
ODD	657	L
OR	14x	K
ORA	14x	K
ORM	10x	K
PIA	547	M
PID	557	M
PIR	567	M
POP	327	C
PUSH	427	C
RADD	117	J
RAND	207	J
RCC	117	I
RCL	027	J
RCM2	317	J
RCPY	067	J
RCR	067	J
RLC	037	J
RMAP	737xx001	V
RMAPP	7370x010	V
RMEMP	737xx004	V
ROR	107	J
RRC	077	J
RSUB	217	J
RWIN	7370x000	V
RXOR	307	J
SAM	41x	D
SB	727xxx61	T
SBA	23x	E
SB1	25x	E
SB2	26x	E
SB3	27x	E
SBD	727xxx71	T
SBF	767x07vx	Q
SBI	727xxx66	T
SBP	727xxx02	T
SBPD	727xxx12	T
SBPI	727xxx06	T
SCL	407	B
SCR	447	B
SKN	22x	I
SKZ	60x	I
SLA	52x	H
SLAD	53x	H
SLR	50x	H
SLRD	51x	H
SPL	417	B
SPR	457	B
SRA	56x	H
SRAD	57x	H
SRL	54x	H
SRLD	55x	H
STA	43x	D
STA1	31x	D
STB	44x	D
STP	44x	D
STZ	40x	D
ST1	45x	D
ST2	46x	D
ST3	47x	D
ST23	32x	D
SUB	23x	E
SWB	727xxx63	U
SWBD	727xxx73	U
SWBF	767x07wx	Q
SWBI	727xxx67	U
SWBP	727xxx03	U
SWBPD	727xxx13	U
SWBPI	727xxx07	U
SWMAP	737xx003	V
SWWIN	737xx002	V
TRAP	467	L
TBFZ	767x06vx	Q
TRT	357	B
UFA	147	F
UP	337	C
WMAP	737x0003	V
WWIN	737x0002	V
XEC	70x	L
XOA	34x	K
XOM	30x	K
XOR	34x	K

Assembler Format Mnemonics and Operands (1)	Octal Code (1)	Cond Code (2)	Name
Instruction Type A: DECIMAL OPTION (Optional on IV/70)			
CPL sbs,sbd,l	637	ZMC	Compare Block Logic
CPN sbs,sbd,l1,l2	607	OZMC	Compare Block Numeric
DADD sbs,sbd,l1,l2	627	OZMC	Decimal Addition
DSUB sbs,sbd,l1,l2	647	OZMC	Decimal Subtraction
MVCR sbs,sbd,l	617	– –	Move Character Right
MVCL sbs,sbd,l	667	– –	Move Character Left
Instruction Type B: WORD AND CHARACTER MANIPULATION			
LCL e	007	Z	Load Character Left
LCR e	047	Z	Load Character Right
LPL e	017	Z	Load Parallel Left
LPR e	057	Z	Load Parallel Right
MVE c	237	– –	Move Block
MVEL	757	– –	Move Long Block of Words
MVL b,c	137	– –	Move Block Left
MVR b,c	177	– –	Move Block Right
SCL e	407	Z	Store Character Left
SCR e	447	Z	Store Character Right
SPL e	417	Z	Store Parallel Left
SPR e	457	Z	Store Parallel Right
TRT e	357	– –	Translate Bytes
Instruction Type C: LIST PROCESSING			
DOWN e	437	– –	Down List
IN e	347	– –	Insert Into List
POP e	327	– –	Pop Up List
PUSH e	427	– –	Push Down List
UP e	337	– –	Up List
Instruction Type D: LOAD/STORE			
LDA* e,x	03x	– –	Load RA
LDA1* e,x	01x	– –	Load RA & X1
LDB* e,x	04x	– –	Load RB
LD1* e,x	05x	– –	Load X1
LD2* e,x	06x	– –	Load X2
LD3* e,x	07x	– –	Load X3
LD23* e,x	02x	– –	Load X2 and X3
SAM* e,x	41x	– –	Store RA Address
STA* e,x	43x	– –	Store RA
STA1* e,x	31x	– –	Store RA & X1
STB* e,x	44x	– –	Store RB
STP* e,x	42x	– –	Store RP
STZ* e,x	40x	– –	Store Zero
ST1* e,x	45x	– –	Store X1
ST2* e,x	46x	– –	Store X2
ST3* e,x	47x	– –	Store X3
ST23* e,x	32x	– –	Store X2 & X3
Instruction Type E: FIXED POINT			
ADA* e,x	13x	OZMC	Add to RA
ADM* e,x	12x	OZMC	Add to Memory
AD1* e,x	15x	OZMC	Add to X1
AD2* e,x	16x	OZMC	Add to X2
AD3* e,x	17x	OZMC	Add to X3
DIV c	227	ZMC	Divide
MPY c	127	ZMC	Multiply
SBA* e,x	23x	OZMC	Subtract from RA

SB1* e,x	25x	OZMC	Subtract from X1
SB2* e,x	26x	OZMC	Subtract from X2
SB3* e,x	27x	OZMC	Subtract from X3
Instruction Type F: FLOATING POINT			
FAD	157	ZMC	Floating Add
FDV	267	ZMC	Floating Divide
FMP	167	ZMC	Floating Multiply
FSB	257	ZMC	Floating Subtract
UFA	147	ZMC	Unnormalized Floating Add
Instruction Type G: COMPARISON			
CPA* e,x	33x	OZMC	Compare RA
CP1* e,x	35x	OZMC	Compare X1
CP2* e,x	36x	OZMC	Compare X2
CP3* e,x	37x	OZMC	Compare X3
Instruction Type H: SHIFT ACCUMULATOR			
SLA* e,x	52x	O	Left Arithmetic Single
SLAD* e,x	53x	O	Left Arithmetic Double
SLR* e,x	50x	– –	Left Rotate Single
SLRD* e,x	51x	– –	Left Rotate Double
SRA* e,x	56x	– –	Right Arithmetic Single
SRAD* e,x	57x	– –	Right Arithmetic Double
SRL* e,x	54x	– –	Right Logical Single
SRLD* e,x	55x	– –	Right Logical Double
Instruction Type I: BRANCH/SKIP/TEST			
BAL* e,x	66x	– –	Branch & Link
BCR* e,x	65x	– –	Branch If Carry
BC1* e,x	75x	– –	Branch & Count X1
BC2* e,x	76x	– –	Branch & Count X2
BC3* e,x	77x	– –	Branch & Count X3
BGT* e,x	67x	– –	Branch If Logically Greater
BMI* e,x	64x	– –	Branch on Minus
BNZ* e,x	73x	– –	Branch on Nonzero
BOF* e,x	62x	O–	Branch on Overflow
BPL* e,x	74x	– –	Branch on Not Minus
BRA* e,x	72x	– –	Branch Unconditional
BRD e	507	OZMC	Branch Return Debreak
BRM* e,x	71x	– –	Branch & Mark
BRR e	517	OZMC	Branch Return
BZO* e,x	63x	– –	Branch on Zero
DEC* e,x	21x	– –	Decrement Memory, skip if result zero
INR* e,x	11x	– –	Increment Memory, skip if result zero
MCC* e,x	61x	ZM	Memory Set CC
RCC s	117	ZM	Register Set CC
SKN* e,x	22x	– –	Test Memory, skip if negative
SKZ* e,x	60x	– –	Test Memory, skip if zero
Instruction Type J: REGISTER TO REGISTER			
CDA2	247	– –	Copy Double
RADD s,d,b	117	OZMC	Register Add
RAND s,d,b	207	ZM	AND Source to Dest
RCL s,d,b,c	027	– –	Copy Then Rotate Left
RCM2 s,d	317	OZMC	2's Complement
RCPY s,d,b	067	– –	Copy Source to Dest
RCR s,d,b,c	067	– –	Copy Then Rotate Right
RLC s,d,b,c	037	– –	Rotate Left Then Copy
ROR s,d,b	107	ZM	OR Source to Dest
RRC s,d,b,c	077	– –	Rotate Right Then Copy
RSUB s,d,b	217	OZMC	Register Subtract
RXOR s,d,b	307	ZM	XOR Source to Dest

MACHINE INSTRUCTIONS (Continued)

Instruction Type K: LOGICAL			
ANA* e,x	24x	ZM	AND to RA
ANM* e,x	20x	ZM	AND to Memory
ORA* e,x	14x	ZM	OR to RA
ORM* e,x	10x	ZM	OR to Memory
XOA* e,x	34x	ZM	XOR to RA
XOM* e,x	30x	ZM	XOR to Memory
Instruction Type L: MISCELLANEOUS			
HLT e	00x	– –	Halt
NOP	067	– –	No Operation
ODD s,d,b,c	657	– –	Compute Odd Parity
TRAP e	467	– –	Trap to 041
XEC* e,x	70x	– –	Execute [EA]
Instruction Type M: INTERRUPT			
IOID e	577	– –	Indexed Interrupt
PIA e	547	– –	Priority Interrupt Arm
PID e	557	– –	Priority Interrupt Disarm
PIR e	567	– –	Priority Interrupt Reset
Instruction Type N: INPUT/OUTPUT			
BOOT s,d	377	Z	Bootstrap Load
ECS d,b	477	– –	Enter Console Keys
EXCT e	527	– –	External Command
EXSN e	537	– –	External Sense
IO e	677	– –	Input/Output Words
IOB e	277	Z	Input/Output Bytes
IOXW e	747	– –	Cross Window I/O
Instruction Type P: DECIMAL/BINARY CONVERSION			
BDEC bz,dz,mz,c	707	OZMC	Binary to Decimal Conversion
DBIN hc,zd,dz,mz,c	717	OZMC	Decimal to Binary Conversion
Instruction Type Q: BIT MANIPULATION			
BFAP s,d	767xxx00	ZM	Bit Address to Bit Pointer Format Conversion
BFPA s,d,	767xx200	ZM	Bit Pointer to Bit Address Format Conversion
BIT s,d,sf,df,st,ld,c	767	ZM	Bit manipulation
LBF s,c	767x06wx	ZM	Load Bit String from Memory to RA
SBF s,c	767x07vx	ZM	Store Bit String from RA
SWBF s,c	767x07wx	ZM	Swap Bit String in Memory with RA
TBFZ s,c	767x06vx	ZM	Test Bit String for Zero
Instruction Type R: BYTE ADDRESS FORMAT CONVERSION			
BAINX s,f	727x0x60	ZMC	Byte Address Index Extraction
BPINX s,f	727x0x00	ZMC	Byte Pointer Index Extraction
BYAP s,d,f	727xxx40	ZMC	Byte Address to Byte Pointer Format Conversion
BYPA s,d,f	727xxx20	ZMC	Byte Pointer to Byte Address Format Conversion
BYTE s,d,f,sf,df,do, io,st,ld	727	ZMC	General Byte instruction
DECBP s,d,f	727xxx10	ZMC	Decrement Byte Pointer
INRBP s,d,f	727xxx04	ZMC	Increment Byte Pointer
Instruction Type S: LOAD BYTE			
LB s,d,f	727xxx61	ZMC	Load RA Using Byte Address
LBD s,d,f	727xxx71	ZMC	Load RA and Decrement Byte Address

LBI s,d,f	727xxx65	ZMC	Load RA Increment Byte Address
LBP s,d,f	727xxx01	ZMC	Load RA Using Byte Pointer
LBPD s,d,f	727xxx11	ZMC	Load RA and Decrement Byte Pointer
LBPI s,d,f	727xxx05	ZMC	Load RA and Increment Byte Pointer
Instruction Type T: STORE BYTE			
SB s,d,f	727xxx62	ZMC	Store from RA Using Byte Address
SBD s,d,f	727xxx72	ZMC	Store from RA and Decrement Byte Address
SBI s,d,f	727xxx66	ZMC	Store from RA and Increment Byte Address
SBP s,d,f	727xxx02	ZMC	Store from RA Using Byte Pointer
SBPD s,d,f	727xxx12	ZMC	Store from RA and Decrement Byte Pointer
SBPI s,d,f	727xxx06	ZMC	Store from RA and Increment Byte Pointer
Instruction Type U: SWAP BYTE			
SWB s,d,f	727xxx63	ZMC	Swap Using Byte Address
SWBD s,d,f	727xxx73	ZMC	Swap and Decrement Address
SWBI s,d,f	727xxx67	ZMC	Swap and Increment Address
SWBP s,d,f	727xxx03	ZMC	Swap Using Byte Pointer Address
SWBPD s,d,f	727xxx13	ZMC	Swap and Decrement Pointer Address
SWBPI s,d,f	727xxx07	ZMC	Swap and Increment Pointer Address
Instruction Type V: MAP			
BRAWIN s	737x2002		Window Read into RP
MAP s,d,rmmpr,rmpr alter,mode	737		Map Maintenance
RMAP s,d	737xx001		Read Memory Mapper
RMAPP d	7370x010		Read Mapper Parity Register
RMEMP s,d	737xx004		Read Extended Memory Parity Register
RWIN d	7370x000		Read Window Register
SWMAP s,d	737xx003		Swap Memory Mapper
SWWIN s,d	737xx002		Swap Window Register
WMAP s	737x0003		Write Memory Mapper
WWIN s	737x0002		Write Window Register

LEGEND OF CODES USED FOR MACHINE INSTRUCTIONS

Assembler coding has the following coding sequence: label (optional, not shown), instruction mnemonic (shown in capital letters), operand (shown in lower case letters), and comments (optional, not shown).

(1) Assembler Format Mnemonics and Octal Code Legend

*	Indirect addressing may be used.
alter	Alter window or mapper memory (0 = read-only, 1 = alter)
b	Byte control. If no value is given by the programmer where indicated, the assembler supplies 7 (all bytes).
c	Count, modulo 64. For Bit instructions, any value above 24 is set to 24.
d	Destination. Never optional.
df	Destination address format of DREG.
do	Decrement option (1 = decrement pointer by one byte).
e	Expression. Must be reducible to an address or count. Not optional.
e,x	Expression plus optional indexing. The expression must be reducible to an address or count.
f	Specifies the offset register.
io	Increment option (1 = increment pointer by one byte).
l	Length (in bytes-1) of memory block.
l1	Difference (in bytes) in lengths of memory blocks.
l2	Length (in bytes-1) of the source block.
ld	Load option (1 = load designated byte into RA).
mode	Select window or mapper memory (0 = window, 1 = mapper memory).
rmmpr	Read memory mapper parity register.
rmpr	Read extended mapper memory parity register.
s	Source registers. Not optional.
sbd	Destination starting bytes. Values can be 0, 1, or 2, for a Decimal Option instruction in IV/70. A value of 3 is allowed in IV/90.

sbs	Source starting bytes. Values can be 0, 1, or 2, for a Decimal Option instruction in IV/70. A value of 3 is allowed in IV/90.
sf	Source address format of SREG
st	Store option (1 = store RA in designated byte)
v	Means that bit 18 is zero.
w	Means that bit 18 is one.
x	(in assembler formats) Indexing allowed.
x	(in operation codes) Varies according to programmer options selected.

(2) Condition Code Legend

C = carry
M = minus
O = overflow
Z = zero

Note: The 700 series of instructions (instructions IOXW and MVEL and types P through V) are referred to as the extended instruction set usable only on the Series IV/90 processors.

Comparative Timing Table
Note: All times are in microseconds

Mnemonic	Op Code	IV/70	IV/90 Model 1	IV/90 Model 2
ADA	13x	16.24	8.12	1.9
ADD	13x	16.24	8.12	1.9
ADM	12x	20.3	16.24	2.66
AD1	15x	16.24	8.12	1.9
AD2	16x	16.24	8.12	1.9
AD3	17x	16.24	8.12	1.9
ANA	24x	16.24	8.12	1.9
AND	24x	16.24	8.12	1.9
ANM	20x	20.3	16.24	2.66
BAINX	727x0x60	— —	32.48av	22.8, 22.8, 23
BAL	66x	12.18	6.09av	1.9 to 7.98 (8)
BCR	65x	12.18	6.09av	1.14, 1.9
BC1	75x	20.3	4.06av	2.28, 1.9nb, 20.3 br and BCx $
BC2	76x	20.3	4.06av	See BC1
BC3	77x	20.3	4.06av	See BC1
BDEC	707	— —	— —	See note 1
BFAP	767xx400	— —	16.24av	28.12
BFPA	767xx200	— —	36.54av	12.16
BGT	67x	12.18	6.09av	1.14, 1.9
BIT	767	— —	— —	See note 7
BMI	64x	12.18	6.09av	1.14, 1.9
BNZ	73x	12.18	6.09av	1.14, 1.9
BOF	62x	12.18	6.09av	1.14, 1.9
BOOT	377	N.A.	N.A.	39.7 + J + R (9)
BPINX	727x0x00	— —	16.24av	9.5, 9.5, 9.69
BPL	74x	12.18	6.09av	1.14, 1.9
BRA	72x	12.18	4.06	1.71
BRAWIN	737x2002	— —	— —	2.66
BRD	507	16.24	16.24	2.85 to 8.93 (8)
BRM	71x	20.3 + I	16.24av	3.04 to 9.31 + IL (8)
BRR	517	16.24	8.12	1.9
BYAP	727xxx40	— —	32.48av	26, 26, 26.2
BYPA	727xxx20	— —	16.24	10, 10, 10.3
BYTE	727xxxxx	— —	— —	See note 3
BZO	63x	12.18	6.09av	1.14, 1.9
CDA2	247	16.24	4.06	1.33
CPA	33x	16.24	8.12	1.9
CPL	637	28.42 + 12.18D	12.18 + 8.12D	4.18 + 1.52CEQ
CPN	607	32.48 + 12.18D	12.18 + 12.18D	9.31 + 1.71CDS
CP1	35x	16.24	8.12	1.9
CP2	36x	16.24	8.12	1.9
CP3	37x	16.24	8.12	1.9
DADD	627	36.54 + 16.24D + C + RC	12.18 + 16.24D + RC	11.4 + 1.77CDS + RC
DBIN	717	— —	— —	See note 2

COMPARATIVE TIMING TABLE (Continued)
Note: All times are in microseconds

Mnemonic	Op Code	IV/70	IV/90 Model 1	IV/90 Model 2
DEC	21x	28.42	16.24	2.85
DECBP	727xxx10	– –	– –	13.9, 13.1, 13.1
DIV	227	36.54 + 16.24K	16.24 or 8.53 + 0.81Kav	4.56 + 0.76K + H + Q (8)
DOWN	437	24.36	24.36	3.61
DSUB	647	36.54 + 16.24D + C + RC	12.18 + 16.24D + RC	11.59 + 1.77CDS + RC
ECS	477	16.18	12.18	6.08 to 7.98
EXCT	527	12.18	12.18	4.56 to 8.74 (8)
EXSN	537	12.18	12.18	6.27 to 8.17 (8)
FAD	157	203av	20.3av	2.85 + F + G + H (8)
FDV	267	462.84av	24.36av	25.08 + H (8)
FMP	167	446.6av	24.36av	4.56 + M + G + H (8)
FSB	257	215.18av	20.3av	3.8 + F + G + H (8)
HLT	00x	12.18	NA	NA
IN	347	40.6, 32.48ns	36.54av	3.99, 5.89
INR	11x	20.3 + I	16.24	2.85 + IL
INRBP	727xxx04	– –	– –	13.3, 13.5, 13.7
IO	677	56.84 + I	64.96, 60.9	39.3 + J + IL (input) 37.81 + J + IL (output) (8)
IOB	277	174.58, 247.66	NA	See note 10
IOID	577	20.3	20.3	19.95 to 23.75 (8)
IOXW	747	– –	– –	See notes 5 and 9
LB	727xxx61	– –	36.54av	28.7, 28.7, 28.9
LBD	727xxx71	– –	– –	29.8, 29.1, 29.1
LBF	767x06wx	– –	40.6av	21.66 + U
LBI	727xxx65	– –	– –	29.3, 29.5, 29.7
LBP	727xxx01	– –	28.42av	18.1, 18.1, 18.2
LBPD	727xxx11	– –	– –	19.2, 18.4, 18.4
LBPI	727xxx05	– –	– –	18.6, 18.8, 19
LCL	007	58.26av	31.06av	6.65, 8.17, 7.98
LCR	047	56.84av	31.06av	5.89, 8.17, 8.74
LDA	03x	12.18	8.12	1.9
LDA1	01x	24.36	12.18	2.66
LDB	04x	12.18	8.12	1.9
LD1	05x	12.18	8.12	1.9
LD2	06x	12.18	8.12	1.9
LD3	07x	12.18	8.12	1.9
LD23	02x	24.36	12.18	2.66
LPL	017	73.08	32.48	6.08, 8.93, 8.74
LPR	057	69.02	32.48	6.08, 8.74, 8.74
MAP	737xxxxx	– –	– –	3.99
MCC	61x	16.24	8.12	1.9
MPY	127	48.72 + 2.03K	12.18 or 9.14 + 0.71K	5.13 + 0.57K + P (8)
MVCL	667	40.6 + 8.12D	20.3 + 12.18D	5.7 + 1.46CMV
MVCR	617	40.6 + 8.12D	20.3 + 12.18D	5.7 + 1.46CMV
MVE	237	20.6 + 8.12K	16.24 + 8.12K	3.8 + 2.09K
MVEL	757	– –	– –	See note 6
MVL	137	60.9 + 56.8K	24.36 + 16.24K	6.65 + 5.32K
MVR	177	60.9 + 56.84K	24.36 + 16.24K	6.65 + 5.32K
NOP	067	12.18	4.06, 8.12	1.14
ODD	657	20.3 + 6.09K	2.44 + 0.81K	2.85 + 0.57K, 2.66 + 0.57K, 3.23 + 0.57K (12)
OR	14x	16.24	8.12	1.9
ORA	14x	16.24	8.12	1.9
ORM	10x	20.3	16.24	2.66
PIA	547	12.18	12.18av	4.18 to 12.16 (8)
PID	557	12.18	12.18av	4.18 to 12.16 (8)
PIR	567	16.24 + 12.18T (11)	16.24 + 12.18T (11)	(14.44 to 22.42) + 16.34T (11)
POP	327	36.54	24.36av	3.61, 4.37
PUSH	427	44.66,36.54ns	28.42av	2.85,4.37

COMPARATIVE TIMING TABLE (Continued)
Note: All times are in microseconds

Mnemonic	Op Code	IV/70	IV/90 Model 1	IV/90 Model 2
RADD	117	16.24	4.06	1.52, 1.33, 1.9 (12)
RAND	207	16.24	4.06	1.9, 1.33, 2.28 (12)
RCC	117	16.24	4.06	1.33
RCL	027	16.24 + 4.06K	3.25 + 0.1K	3.23 + 0.19K, 1.14, 3.61 (12)
RCM2	317	20.3	4.06	1.9, 1.33, 2.28 (12)
RCPY	067	12.18	3.45 + 0.1K	1.71, 1.14, 2.28 (12)
RCR	067	12.18 + 4.06K	3.45 + 0.1K	3.42 + 0.19K,1.14,3.8 (12)
RLC	037	16.24 + 4.06K	8.12	2.66 + 0.19K, 1.14, 3 (12)
RMAP	737xx001	— —	— —	3.8
RMAPP	7370x010	— —	— —	3.99
RMEMP	737xx004	— —	— —	3.99 (MPRO or MAPR) 4.18 (MPR1)
ROR	107	16.24	4.06	1.9, 1.33, 2.28 (12)
RRC	077	16.24 + 4.06K	2.84 + 0.1K	2.66 + 0.19K, 1.14, 3 (12)
RSUB	217	16.24	4.06	1.52, 1.33, 1.9 (12)
RWIN	7370x000	— —	— —	2.09
RXOR	307	16.24	4.06	1.9, 1.33, 2.28 (12)
SAM	41x	24.36	16.24	3.04
SB	727xxx61	— —	40.6av	27.8, 27.6, 27.8
SBA	23x	16.24	8.12	1.9
SB1	25x	16.24	8.12	1.9
SB2	26x	16.24	8.12	1.9
SB3	27x	16.24	8.12	1.9
SBD	727xxx71	— —	— —	28.9, 27.9, 27.9
SBF	767x07vx	36.54av	24.32 + U + V	
SBI	727xxx66	— —	— —	28.3, 28.3, 28.5
SBP	727xxx02	— —	28.42av	17.1, 16.9, 17.1
SBPD	727xxx12	— —	— —	18.2, 17.3, 17.3
SBPI	727xxx06	— —	— —	17.7, 17.7, 17.9
SCL	407	67.6av	39.18av	7.79, 9.69, 9.50
SCR	447	67.6av	39.18av	7.03, 9.69, 10.26
SKN	22x	16.24	8.12	2.09
SKZ	60x	20.3	8.12	2.09
SLA	52x	18.27 + 6.09K	2.64 + 0.1K	2.66 + 0.19K
SLAD	53x	16.24 + 8.12K	2.84 + 0.1K	2.66 + 0.19(K + OV)
SLR	50x	12.18 + 4.06K	2.23 + 0.1K	2.09 + 0.19K
SLRD	51x	18.27 + 10.15K	2.64 + 0.1K	2.47 + 0.19K
SPL	417	28.42	24.36	3.42
SPR	457	28.42	24.36	3.42
SRA	56x	12.18 + 4.06K	3.05 + 0.1K	3.04 + 0.19K
SRAD	57x	16.24 + 8.12K	3.05 + 0.1K	3.04 + 0.19K
SRL	54x	12.18 + 4.06K	2.23 + 0.1K	2.09 + 0.19K
SRLD	55x	16.24 + 8.12K	2.64 + 0.1K	2.47 + 0.19K
STA	43x	16.24	12.18	1.9
STA1	31x	24.36	16.24	2.66
STB	44x	16.24	12.18	1.9
STP	44x	16.24	12.18	1.9
STZ	40x	16.24	12.18	1.9
ST1	45x	16.24	12.18	1.9
ST2	46x	16.24	12.18	1.9
ST3	47x	16.24	12.18	1.9
ST23	32x	24.36	16.24	2.66
SUB	23x	16.24	8.12	1.9
SWB	727xxx63	— —	40.6av	32.1, 31.9, 32.1
SWBD	727xxx73	— —	— —	33.3, 32.3, 32.3
SWBF	767x07wx	— —	40.6av	24.13 + U + V
SWBI	727xxx67	— —	— —	32.7, 32.7, 32.9
SWBP	727xxx03	— —	28.42	21.5, 21.3, 21.5
SWBPD	727xxx13	— —	— —	22.6, 21.7, 21.7
SWBPI	727xxx07	— —	— —	22, 22, 22.4
SWMAP	737xx003	— —	— —	11.4
SWWIN	737xx002	— —	— —	2.47 (RD = R0)
TRAP	467	12.18	8.12	1.33
TBFZ	767x06vx	— —	— —	21.85 + U

COMPARATIVE TIMING TABLE (Continued)
Note: All times are in microseconds

Mnemonic	Op Code	IV/70	IV/90 Model 1	IV/90 Model 2
TRT	357	292.32av	32.48av	25.65nb, 7.03br1, 14.82br2, 22.61br3 (8)
UFA	147	182.7av	20.3av	2.85 + F + H (8)
UP	337	24.36, 20.3ns	16.24av	2.85, 3.61
WMAP	737x0003	— —	— —	11.02 (RD = R0) 3.8 (RD = R1)
WWIN	737x0002	— —	— —	2.28
XEC	70x	4.06	4.06	1.14
XOA	34x	16.24	8.12	1.9
XOM	30x	20.3	16.24	2.66
XOR	34x	16.24	8.12	1.9

TIMING TABLE NOTES

All timing is given in microseconds. "av" following a timing value indicates that the timing given is an average value.

If a range is given (such as 2.82 to 8.93) it means that the execution time is variable because the microcode must synchronize to the interrupt logic of the processor. The mean execution time depends somewhat on the previous instruction. The longer the execution of the previous instruction is, the closer to the lower bound the execution time is on the current instruction.

Multiple values of timing separated by commas indicate the times for one of the following sequences of conditions, as applicable:

- No skip, skip
- Left byte, middle byte, right byte
- No shift, shift left, shift right
- Destination not RO, R1, RP; destination RO or R1; destination RP destination RP (15)

For IV/70, indexing adds 8.12 microseconds and indirect addressing adds 4.06 microseconds.

For IV/70 and IV/90 Model 1 Bit and Byte instructions, the timings are approximate (within 10%) depending on the bit count and the starting bit number. For these instructions, add 16.24 microseconds if the field spans two words.

IV/90 Model 2 timings assume that fast memory is being accessed. For each read from slow memory, add about 5 microseconds. The range is from 3.04 to 7.03 microseconds. For each write to slow memory, add about 6 microseconds. The range is from 3.99 to 7.98 microseconds.

When fast dynamic memory is used, it is refreshed every 12.16 microseconds. This can delay a memory access by 0 to 0.57 microseconds. The average delay is 0.228 microseconds. This is an average degradation over fast static memory of about 1.9%.

An indirect effective address adds 0.76 microseconds to a IV/90 Model 2 instruction execution time.

Timing Table Legend

$	Current address
av	The time varies around the given average value.
B	Number of blank bytes written.
br	Branch.
br1	Branch on 1st byte.
br2	Branch on 2nd byte.
br3	Branch on 3rd byte.
C	0 normal. If recomplement, add 12.18 + 12.18D.
C1	Number of right shifts required to right justify bit field.
CDS	Total number of bytes in the destination string.
CEQ	Total number of bytes that result in an equal compare.
CMV	Total number of bytes to be moved.
D	Number of destination words in the decimal string.
F	Time to perform the floating point add or subtract. This takes 1.52 microseconds if no alignment is required and 3.42 + 0.38a microseconds if it is required, where "a" is the alignment shift count.
G	Time taken to normalize the sum. This is 0.76 + 0.95n microseconds where "n" is the normalized shift count.
H	Plus 0 to 6.08 microseconds for synchronizing with the interrupt logic.
I	0, normal. 4 if fetched at interrupt.
IL	IV/90 Model 2 interrupt location timing: 10.64 minimum, 14.25 maximum.
J	0 to 3.8 microseconds for synchronizing with the IV/70 clocks. The time given is the time to transfer one word. Each additional word takes at least 8.1 microseconds.
K	The count in the six rightmost bits of the effective address.
M	Time to perform the floating point multiply. This takes 15.77 + 4.37n microseconds, where "n" is the normalized shift count.
N	Number of numeric bytes read or written.
NA	Not applicable.
nb	No branch.
ns	No skip.
OV	Value is 1 if overflow condition code is set by shift, otherwise 0.

P	0 to 1.52 + 0.19K depending on values in RA and X2.
Q	0 to 1.9 microseconds depending on values in RA, X2, and RB.
R	65.519 milliseconds if source register = R0.
RC	If recomplement is necessary, add 12.18 + 12.18D to IV/70 time, 4.06 + 12.18D to IV/90 Model 1 time, and 5.32 + 1.14CDS to IV/90 Model 2 time. This is when overflow is false and carry is true.
T	Number of active interrupts.
U	0.19(C + C1) + 0.95W
V	0.19 + 0.76W
W	Equals one if bit field spans two words.

MICROSECONDS

Numbered Notes		Description
(1) 707 — BDEC	32.11	Basic overhead.
	+ 22.04NZ	Where NZ is the number of nonzero bytes stored, not counting the byte in the units position.
	+ 6.08	For first leading zero or blank byte stored after the units position.
	+ 2.66SZ	Where SZ is the number of leading zero or blank bytes stored after the first one mentioned above.
	— 19	If RA = 0 at the start of this instruction.
	+ 1.14RW	Where RW is the number of words modified minus 1.
	+ .19	If overflow.
	+ LB	Where LB is the number of left-most bytes stored.
(2) 717 — DBIN	11.21	Basic overhead.
	+ 3.23SC	Where SC is the number of bytes scanned. This may be less than the specified count + 1 if the halt conversion option is specified and an illegal byte is encountered.
	+ .38	If halt conversion option is specified.
	+ .19	If zone decode option is specified.
	— 19	If the decimal string is negative.
	+ .38	If the decimal string is zero.
	+ OV1	Where OV1 is the number of times overflow occurs on the multiplication by 10.
	+ OV2	Where OV2 is the number of times overflow occurs when a new digit is added.
	+ .38RB	Where RB is the number of right bytes scanned.
	+ .57ZD	If zone decoding, then ZD is the number of digits scanned with a digit zone.
	+ .95	If zone decoding, and a minus zone is detected in the units position.
	— .76IZ	If zone decoding, then IZ is the number of illegal zones scanned.
	— 1.14ID	Where ID is the number of illegal digits scanned.
	+ .38R	Where R is the number of words read minus 1.
	+ 2.66	If the halt conversion option is specified and an illegal condition is detected.
(3) Refer to the byte derivative instructions for timing.		
(4) Refer to the MAP derivative instructions for timing.		
(5) 747 — IOXW	40.09	If input.
	38.57	If output.
	+ 0 to 3.8	For synchronizing with the IV/70 clocks.
	+ 10.64 to 14.25	If interrupt. See note 9.
(6) 757 — MVEL	2.66	Basic overhead.
	+ 2.28N	Where N is the number of words moved.
	— 0.76	If RA = 0 at the start of the instruction.
(7) Refer to the BIT derivative instructions for timing.		

(8) The execution time of this instruction is variable because its microcode must synchronize to the CPU's interrupt logic, which is operating asynchronously to the microcode. The mean execution time depends somewhat on the previous instruction. The longer the previous instruction takes, the closer to the lower bound the execution time is.

(9) The execution times of the I/O and IOXW instructions depend on how fast the particular I/O device can handshake and on how many words are involved per IO transfer. The timings shown above are the minimum timings for one word transfers. Add at least 8.12 usec. for each additional word transferred.

(10) Not supported by Four-Phase.

(11) Refer to the *Series IV Processor Reference Manual* for limitations.

(12) Values apply to destination not R0,R1, or RP; destination R0 or R1; destination RP.

CONDITION TESTS

Type Test	Arithmetic (Sign Bit); No Overflow Possible	Logical (No Sign)	CPL (A = dest, B = source)	Arithmetic (Sign Bit); Overflow Possible	Decimal Arithmetic (A = dest, B = source)
[A] < B] ?	LDA A CPA B BMI YES NO EQU $	LDA A CPA B BCR YES NO EQU $	LD23 @A CPL BCR YES NO EQU $	LDA A BOF $ + 1 CPA B BOF NO BMI YES NO EQU $ BMI NO BRA YES	LD23 @A CPN BMI YES NO EQU $
[A] ⩽ [B]?	LDA A CPA B BMI YES BZO YES NO EQU $	LDA A CPA B BGT NO BRA YES	LD23 @A CPL BGT NO BRA YES	LDA A BOF $ + 1 CPA B BZO YES BOF NO BMI YES NO EQU $ BMI NO BRA YES	LD23 @A CPN BMI YES BOF NO BZO YES NO EQU $
[A] = [B] ?	LDA A CPA B BZO YES NO EQU $	LDA A CPA B BZO YES NO EQU $	LD23 @A CPL BZO YES NO EQU $	LDA A CPA B BZO YES NO EQU $	LD23 @A CPN BOF NO BZO YES NO EQU $
[A] ≠ [B] ?	LDA A CPA B BNZ YES NO EQU $	LDA A CPA B BNZ YES NO EQU $	LD23 @A CPL BNZ YES NO EQU $	LDA A CPA B BNZ YES NO EQU $	LD23 @A CPN BOF NO BNZ YES NO EQU $
[A] ⩾ [B] ?	LDA A CPA B BPL YES NO EQU $	LDA A CPA B BCR NO BRA YES	LD23 @A CPL BCR NO BRA YES	LDA A BOF $ + 1 CPA B BOF NO BPL YES NO EQU $ BPL NO BRA YES	LD23 @A CPN BPL YES NO EQU $
[A] > [B]?	LDA A CPA B BMI NO BNZ YES NO EQU $	LDA A CPA B BGT YES NO EQU $	LD23 @A CPL BGT YES NO EQU $	LDA A BOF $ + 1 CPA B BZO NO BOF NO BPL YES NO EQU $ BPL NO BRA YES	LD23 @A CPN BMI NO BOF YES BNZ YES NO EQU $

BIT TESTS USING CC ZERO

TEST [LOC] Bit 'n' = 1?		TEST [LOC] Bit 'n' = 0?	
LDA MASK ANA LOC BNZ YES NO EQU $	LDA MASK ANA LOC BZO NO YES EQU $	LDA MASK ANA LOC BZO YES NO EQU $	LDA MASK ANA LOC BNZ NO YES EQU $
Where [MASK] is bit 'n' = 1, others = 0.			

CHARACTER SET

Video Attribute Characters

"300" System: bits 0 and 1 must be 1; bits 4 and 5 control the attributes.

Bits 4 and 5 = 00 or 01: normal; = 10: intensify; = 11: blank.

Also 05, 0205: cursor with blank; 010, 0210: cursor with normal; 031, 0231: cursor with intensify.

"5-10-31" System: 05, 0205: blank; 010, 0210: normal; 031, 0231: intensify

Octal Code	Keyboard* Character	Control Character Interpretation (ASCII)		Display Character
000	ctrl =	NUL	Null	●
001	ctrl A	SOH	Start of Heading (CC)	△ (1)
002	ctrl B	STX	Start of Text (CC)	♭ (1)
003	ctrl C	ETX	End of Text (CC)	¢
004	ctrl D	EOT	End of Transmission (CC)	◢ (NL)
005	ctrl E	ENQ	Enquiry (CC)	♯ (1) (2)
006	ctrl F	ACK	Acknowledge (CC)	♯ (1)
007	ctrl G	BEL	Bell	¤ (1)
010	ctrl H	BS	Backspace (FE)	← (1) (2)
011	ctrl I	HT	Horizontal Tabulation (FE)	←
012	ctrl J	LF	Line Feed (FE)	\ (45°)
013	ctrl K	VT	Vertical Tabulation (FE)	/ (45°)
014	ctrl L	FF	Form Feed (FE)	£
015	ctrl M	CR	Carriage Return (FE)	■ (Check)
016	ctrl N	SO	Shift Out	¬
017	ctrl O	SI	Shift In	| (Left)
020	ctrl P	DLE	Data Link Escape (CC)	| (Right)
021	ctrl Q	DC1	Device Control 1	[
022	ctrl R	DC2	Device Control 2	\ (Check)
023	ctrl S	DC3	Device Control 3	▬ (EOM)
024	ctrl T	DC4	Device Control 4	√ (1)
025	ctrl U	NAK	Negative Acknowledge (CC)	— (Over)
026	ctrl V	SYN	Synchornous Idle (CC)	^
027	ctrl W	ETB	End of Transmission Block (CC)	]
030	ctrl X	CAN	Cancel	□
031	ctrl Y	EM	End of Medium	↑ (1) (2)
032	ctrl Z	SUB	Substitute	■
033	ctrl +	ESC	Escape	◦
034	ctrl , (comma)	FS	File Separator (IS)	◄ (Stop)
035	ctrl – (minus)	GS	Group Separator (IS)	▌
036	ctrl .	RS	Record Separator (IS)	► (MI)
037	ctrl /	US	Unit Separator (IS)	\ (Slash)

ctrl CONTROL key pressed at same time
(CC) Communication Control
(FE) Format Effector
(IS) Information Separator
1 These symbols are currently displayed but not supported. Other symbols may be substituted on later models.
2 Blank for 7002 Processor
* 7200/7220/7240 keyboard — alphanumeric style

CHARACTER SET (Continued)

Octal Code	Keyboard Character
040	space bar
041	!
042	"
043	#
044	$
045	%
046	&
047	' (7 shift)
050	(
051	)
052	*
053	+
054	, (comma)
055	-
056	.
057	/ (slash)
060	0 (zero)
061	1
062	2
063	3
064	4
065	5
066	6
067	7
070	8
071	9
072	:
073	;
074	<
075	=
076	>
077	?

Octal Code	Keyboard Character
100	@ (= A shift)
101	A shift
102	B shift
103	C shift
104	D shift
105	E shift
106	F shift
107	G shift
110	H shift
111	I shift
112	J shift
113	K shift
114	L shift
115	M shift
116	N shift
117	O shift
120	P shift
121	Q shift
122	R shift
123	S shift
124	T shift
125	U shift
126	V shift
127	W shift
130	X shift
131	Y shift
132	Z shift
133	÷
134	× (mult)
135	(- shift) (centered)
136	EXP ↑ (shift ÷)
137	— (X shift) (under)

Octal Code	Keyboard Character
140	'(0 shift) (zero)
141	A
142	B
143	C
144	D
145	E
146	F
147	G
150	H
151	I
152	J
153	K
154	L
155	M
156	N
157	O
160	P
161	Q
162	R
163	S
164	T
165	U
166	V
167	W
170	X
171	Y
172	Z
173	ctrl ÷
174	ctrl X (mult)
175	ctrl *
176	ctrl ↑
177	ctrl ←

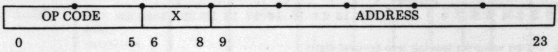
Octal Code	Keyboard Character	Interpretation
200	↑	Cursor Up
201	←	Cursor Left
202	→	Cursor Right
203	↓	Cursor Down
204	EOM	End of Message
205	ATTEN	Attention
206	ROLL (↓ shift)	Roll Down
207	ERASE (shift HOME)	Erase Screen
220	HOME	Cursor Home
211	TAB	Horizontal Tab
212	ROLL (↑ shift)	Roll Up
213	TAB shift	Vertical Tab
214	EOM shift	Shifted EOM
215	CURSOR RETURN	Cursor Return
216	CURSOR RETURN shift	Shifted Cursor Return
217	INSERT (→ shift)	Insert
220	DELETE (← shift)	Delete
221	F1	Function Key 1
222	F2	Function Key 2
223	F3	Function Key 3
224	F4	Function Key 4
225	F5	Function Key 5
226	F6	Function Key 6
227	F7	Function Key 7
230	F8	Function Key 8
231	F9	Function Key 9
232	F10	Function Key 10

Octal Code	Keyboard Character	Interpretation
233	F11	Function Key 11
234	ctrl →	Control →
235	TOTAL	Total
236	ctrl ↓	Control Roll Down
237	ctrl EOM	Control EOM
240		Not Used
.		.
.		.
257		Not Used
241	— — —	Lightpen
260	ctrl 0	Control 0
261	ctrl 1	Control 1
262	ctrl 2	Control 2
263	ctrl 3	Control 3
264	ctrl 4	Control 4
265	ctrl 5	Control 5
266	ctrl 6	Control 6
267	ctrl 7	Control 7
270	ctrl 8	Control 8
271	ctrl 9	Control 9
272		Not Used
.		.
.		.
374		Not used
375	ctrl HOME	Control Home
376	CURSOR ctrl RETURN	Control Cursor Return
377	ctrl TAB	Control TAB

s = SHIFT key pressed at same time c = CTRL key pressed at same time

INSTRUCTION FORMATS
(Machine Language Form)

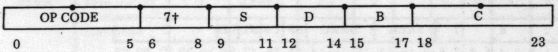
Memory Reference Instructions



X = Modification field

0,7 = No address modification
1 = Indirect address
2 = Index X1 and direct
3 = Index X1 and indirect
4 = Index X2 and direct
5 = Index X2 and indirect
6 = Index X3 and direct

Non-Memory Reference Instructions



Any or all of S, D, B, C may be optional, required, or unused. See the individual instruction for this information.

S = Source register D = Destination register

S or D	0	1	2	3	4	5	6	7
REGISTER	R0	R1	RP	RA	RB	X1	X2	X3

B = Byte Control

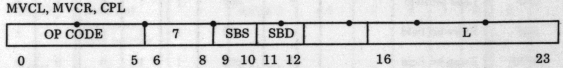
		Byte 0		Byte 1		Byte 2		
B	0	1	2	3	4	5	6	7
BYTES AFFECTED	None	2	1	1,2	0	0,2	0,1	0,1,2

C = Count; range = $0\text{-}63_8$

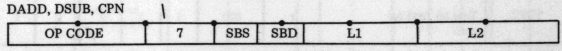
*For Shift Accumulator instructions, Mod Field 7_8

Decimal Option Instructions

MVCL, MVCR, CPL



DADD, DSUB, CPN



SBS = Starting byte source SBD = Starting byte destination

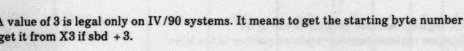
SBS or SBD	0	1	2	3
Starting Byte	leftmost byte	middle byte	rightmost byte	*

L1 = Difference in bytes between lengths of source and destination quantities
L2 = Length in bytes of source, −1

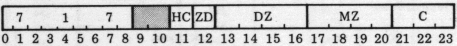
* A value of 3 is legal only on IV/90 systems. It means to get the starting byte number from X2 if sbs = 3; get it from X3 if sbd = 3.

IV/90 INSTRUCTION FORMATS

Binary-to-Decimal Conversion (BDEC)

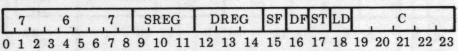


Decimal-to-Binary Conversion (DBIN)



BZ = 4-bit value for leading zero decimal digits
C = Decimal string length-1
DZ = Digit Zone
HC = Halt conversion on an illegal byte
MZ = Minus Zone
ZD = Zone Decoding, least significant digit

Bit Manipulation (BIT)

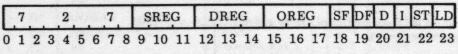


SREG = Source register number (address of bit field)
DREG = Destination register number (updated address of bit field)
SF = Source address format of SREG 0 = Bit pointer format
DF = Destination address format of DREG 1 = Bit address format
ST = Store option (1 = store RA into bit field)
LD = Load option (1 = load RA with bit field)
C = Number of bits in field

Mnemonics	Description	SREG	DREG	SF	DF	ST	LD	C
BFAP	Bit address to bit pointer format conversion	r	d,SREG	1	0	0	0	0
BFPA	Bit pointer to bit address format conversion	r	d,SREG	0	1	0	0	0
LBF	Load bit field	r	000	1	1	0	1	r
SBF	Store bit field	r	000	1	1	0	1	r
SWBF	Swap bit field	r	000	1	1	0	0	r
TBFZ	Test bit field for zero	r	000	1	1	0	0	r

Note: r = required operand; d = optional operand, default shown; all other operands supplied by the assembler.

Byte Manipulation (BYTE)



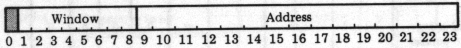
SREG = Source register number (pointer to the byte)
DREG = Destination register number (updated pointer to the byte)
OREG = Offset register number (updated byte offset)
SF = Source address format of SREG 0 = Byte pointer format
DF = Destination address format of DREG 1 = Byte address format
D = Decreement option (1 = decreement pointer by one byte)
I = Increment option (1 = increment pointer by one byte)
ST = Store option (1 = store RA in designated byte)
LD = Load option (1 = load designated byte into RA)

Mnemonic	Description	SREG	DREG	OREG	SF	DF	D	I	ST	LD
BAINX	Byte address index extraction	r	000	r	1	1	0	0	0	0
BPINX	Byte pointer index extraction	r	000	r	0	0	0	0	0	0
BYAP	Byte address to pointer conversion	r	d,SREG	d,0	1	0	0	0	0	0
BYPA	Byte pointer to address conversion	r	d,SREG	d,0	0	1	0	0	0	0
DECBP	Decrement byte pointer	r	d,SREG	d,0	0	0	0	0	0	0
INRBP	Increment byte pointer	r	d,SREG	d,0	0	0	0	1	0	0
LB	Load using byte address	r	d,0	d,0	1	1	0	0	0	1
LBD	Load and decrement byte address	r	d,SREG	d,0	1	1	1	0	0	1
LBI	Load and increment byte address	r	d,SREG	d,0	1	1	0	1	0	1
LBP	Load using byte pointer	r	d,0	d,0	0	0	0	0	0	1
LBPD	Load and decrement byte pointer	r	d,SREG	d,0	0	0	1	0	0	1
LBPI	Load and increment byte pointer	r	d,SREG	d,0	0	0	0	1	0	1
SB	Store using byte address	r	d,0	d,0	1	1	0	0	1	0
SBD	Store and decrement byte address	r	d,SREG	d,0	1	1	1	0	1	0
SBI	Store and increment byte address	r	d,SREG	d,0	1	1	0	1	1	0
SBP	Store using byte pointer	r	d,0	d,0	0	0	0	0	1	0
SBPD	Store and decrement byte pointer	r	d,SREG	d,0	0	0	1	0	1	0
SBPI	Store and increment byte pointer	r	d,SREG	d,0	0	0	0	1	1	0

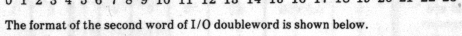
Mnemonic	Description	SREG	DREG	OREG	SF	D F	D	I	ST	LD
SWB	Swap using byte address	r	d,0	d,0	1	1	0	0	1	1
SWBD	Swap and decrement byte address	r	d,SREG	d,0	1	1	1	0	1	1
SWBI	Swap and increment byte address	r	d,SREG	d,0	1	1	0	1	1	1
SWBP	Swap using byte pointer	r	d,0	d,0	0	0	0	0	1	1
SWBPD	Swap and decrement byte pointer	r	d,SREG	d,0	0	0	1	0	1	1
SWBPI	Swap and increment byte pointer	r	d,SREG	d,0	0	0	0	1	1	1

Note: r = required operand; d = optional operand, default shown; all other operands are provided by the assembler.

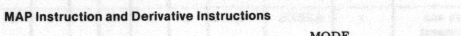
Move Long Block of Words (MVEL)



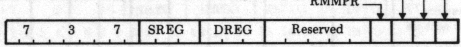
The format of X2 and X3 (the source and destination areas) is shown below. The word count is in RA.



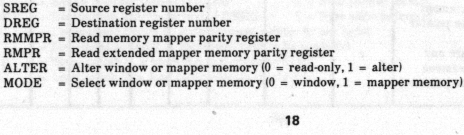
I/O Cross Window (IOXW)



The format of the second word of I/O doubleword is shown below.



MAP Instruction and Derivative Instructions



The MAP instruction reads or modifies the window register, or the mapper mamory, or reads the IV/90 Model 2 parity registers.

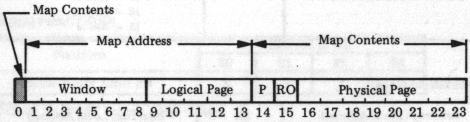
SREG = Source register number
DREG = Destination register number
RMMPR = Read memory mapper parity register
RMPR = Read extended mapper memory parity register
ALTER = Alter window or mapper memory (0 = read-only, 1 = alter)
MODE = Select window or mapper memory (0 = window, 1 = mapper memory)

Mnemonics	Description	SREG	DREG	RMMPR	RMPR	ALTER	MODE
BRAWIN	Change window and branch	r	RP	0	0	1	0
RMAP	Read mapper memory	r	d,SREG	0	0	0	1
RMAPP	Read mapper memory parity register	000	r	1	0	0	0
RMEMP	Read extended memory parity register	r	r	0	1	0	0
RWIN	Read window register	000	r	0	0	0	0
SWMAP	Swap mapper memory	r	d,SREG0	0	0	1	1
SWWIN	Swap window register	r	d,SREG	0	0	1	0
WMAP	Write mapper memory	r	000	0	0	1	1
WWIN	Write window register	r	000	0	0	1	0

Note: r = required operand; d = optional operand, default shown; all other operands supplied by the assembler

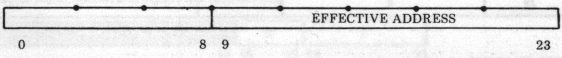
The general format of the source register for reading and altering the window register or the mapper memory is as follows:

(Set REPEAT switch down and DISPLAY SELECT switch to 6 to display physical page. Set switch to 7 to display Window number.)

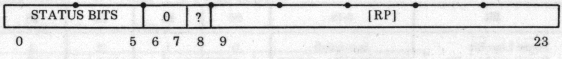


SPECIAL WORD FORMATS

Memory Reference Indirect (Reference of Indexed Address)



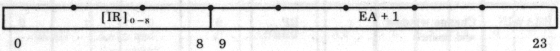
BRM, BRR, BRD (Effective Address)



Status Bits

Bit	0	1	2	3	4	5
Indication	Halt	Parity	Overflow	Zero	Minus	Carry

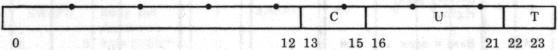
BRM (RP)



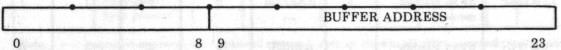
BAL (RP)

10,10B (Effective Address)

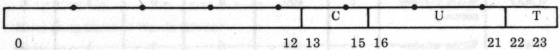
Word 0 (even address)



Word 1 (next odd address)



BOOT (D)



[Memory Location 1] = I

Word 0 and [D] Format

Bit	Meaning	Bit	Meaning
0-12 13-15 16-21	(not used) C = device channel number U = device unit number	22-23	T = type of I/O operation: 00 = data out 01 = data in 10 = control 11 = status

PIA, PID, PIR [Effective Address]

Bit	Interrupt Level Affected	Bit	Interrupt Level Affected
0-15	(not used)	20	3
16	7	21	2
17	6	22	1
18	5	23	0
19	4		

EXCT, EXSN [Effective Address]

Bit	0-19	20	21	22	23
Sense Line Bit	(not used)	0	1	2	3
Command Line Bit	Only 11-14 used	0	1	2	3

EXCT [EA] FUNCTIONS

Bit	Command
11	Disable error correction. When set (= 1), disables the error correction of single-bit errors. When reset (= 0), enables the error correction logic. Has no effect on static RAM memories.
12	Freeze memory error correction code (MECC). When set (= 1), latches the current MECC in the memory parity register. When reset (= 0), allows the MECC to be updated normally. Has no effect on static RAM memories.
13	Inhibit refresh on RAM Card 1 (for diagnostic use). When set (= 1), inhibits the refresh of the RAM devices on RAM Card 1. When reset (= 0), enables the refresh of RAM. Has no effect on static RAM memories.
14	Inhibit refresh on RAM Card 2 (for diagnostic use). When set (= 1), inhibits the refresh of the RAM devices on RAM Card 2. When reset (= 0), enables the refresh of RAM. Has no effect on static RAM memories.

Parity and Intensity Control

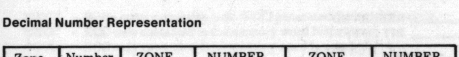
Bits				Meaning
20	21	22	23	
x	x	x	1	Interrupt on level 7
1	0	0	1	Enable 5-10-31 Intensity
1	0	1	0	Enable 300 Intensity
1	0	1	1	Disable variable intensity (also affected by reset)
1	1	0	0	Enable parity checking
1	1	0	1	Disable parity checking
1	1	1	0	Normal parity checking (odd parity)
1	1	1	1	Force a "one" bit into a parity cell (for diagnostic purposes only) (static RAM only)

Note that bits 11 through 14 apply to the extended memory portion of dynamic RAM memories only.

EXSN [EA] FUNCTIONS

Function	Bits Set in EA			
	20	21	22	23
Sense Parity Error Bit	X	X	X	1

UP, DOWN, IN (FREE Table): TRT (Table)



Character Manipulation and Input Pack Instructions

B = Byte Control C = Count S = Shift Direction

Decimal Number Representation

Either ASCII or EBCDIC may be used for the zone field.

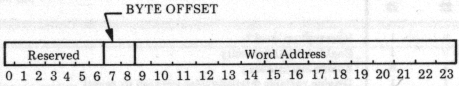
Legal Zone Fields

Type	Number System	
	ASCII	EBCDIC
Plus	0100	1100
Minus	0101	1101
Digit	0011	1111
Blank	0010	0100

The sign of a decimal number is indicated in the zone field of the least significant digit. Negative numbers have a minus zone in the least significant digit. All other legal zones denote a positive number.

All other digits in the decimal string should have a digit zone. Leading zeros may be suppressed (denoted as blanks) by using a blank zone instead of a digit zone. The number field of a leading blank must be zero.

Byte Pointer Format



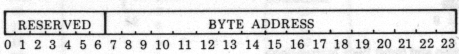
Byte Offset	Byte Indicated
00	0 (Most significant)
01	1
10	2 (Least significant)
11	Illegal

Byte Address Format

The byte address format allows bytes to be addressed directly. In this format the address itself is a byte address (rather than a word address and an offset). Thus, byte *n* is in word *n/3*. Its format is shown below. Byte addressability is useful in some COBOL routines.

When an address is in byte address format and greater than 0300000 or less than zero, it is reduced modulo 0300000 (the highest byte address + 1).

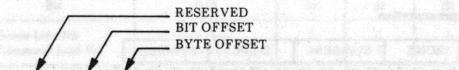
To convert byte address format to byte pointer format, divide the byte address by three. The quotient provides the word address and the remainder is the byte offset.



Bit Pointer Format

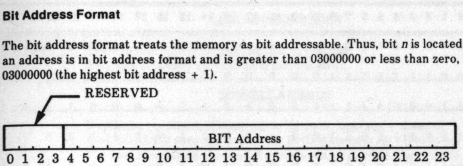
The bit pointer format is similar to byte pointer format, with the addition of a 3-bit offset. A bit offset of 0 designates the most significant bit while a bit offset of 7 designates the least significant bit.

The bit pointer format is illustrated as follows:



Bit Address Format

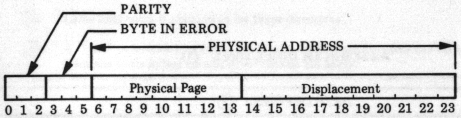
The bit address format treats the memory as bit addressable. Thus, bit *n* is located in word *n/24*. When an address is in bit address format and is greater than 03000000 or less than zero, it is reduced modulo 03000000 (the highest bit address + 1).



BOOTSTRAP WORDS (Console Keys)

Unit	Boot Word
Card Reader	37705101
Buffered Card Reader	37703115
4450/8250 Diskette	37707175
8230 Disc	37705121
8240 Disc	37705201
8261/8281/8291 Disc	37705165
8270 Disc	
Unit 0	37705121
Unit 1	37705125
Unit 2	37705131
Unit 3	37705135
8501/8502/8511/8512 Tape (800 bpi)	37705221
8503/8504/8513 Tape(1600 bpi)	37705241
8507 Tape (7-track)	37705261

Static RAM Extended Memory Parity Registers

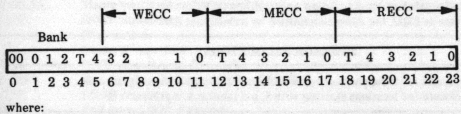


The "parity" field indicates the expected parity for each of the three bytes of the word. The error field indicates which byte, if any, was in error.

Dynamic RAM Extended Memory Parity Registers

There can be one or two DRAM boards in the system, each containing two memory parity registers (named MPR0 and MPR1 for the first board and MPR2 and MPR3 for the second board).

MPRO and MPR2



where:

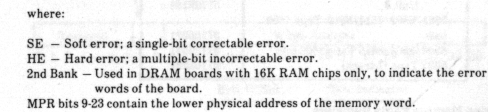
WECC — Write error correction code (ECC) generated during write cycle.
MECC — ECC code contained in the memory word being read.
RECC — ECC code generated from memory word being read.

In reading a memory word, the difference between the corresponding bits of MECC and RECC sets a "1" value in the corresponding bit of a syndrome code made up of S0-S4 and ST bits. The fact that S0-S4 is nonzero and that ST = 1 indicates a single-bit error. The syndrome code is then decoded to identify the bit in error, as follows:

Error Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
S4	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
S3	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
S2	1	1	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
S1	1	1	0	1	1	0	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
S0	0	1	1	0	1	0	1	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
ST	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

MPR bits 0-3 (used in DRAM board with 64K RAM chips only) identifies the physical bank involved. (The current single board DRAM with 64K chips offers up to six banks of memory. Future release may offer two boards with up to 15 banks.)

MPR1 and MPR3



where:

SE – Soft error; a single-bit correctable error.
HE – Hard error; a multiple-bit incorrectable error.
2nd Bank – Used in DRAM boards with 16K RAM chips only, to indicate the error is in the second 32K words of the board.
MPR bits 9-23 contain the lower physical address of the memory word.

ASSEMBLER DIRECTIVES ①

DATA DEFINERS ②	
DCA	Define Constant: delimited ASCII character string.
DCD	Define Constant: double-precision floating point expression.
DCN	Define Constant: numeric (decimal or octal).
DCS	Define Constant: single-precision floating point expression.
MZE	Prefix 5-digit operand with 777 (negative zero).
PZE	Prefix 5-digit operand with 000 (positive zero).
RPZE	Prefix S-D-B-C operand with 007.
SYMBOL DEFINERS ③	
EQU	Define label as equal to 5-digit operand.
SET	Same as EQU, but allow redefinition by subsequent SET.
STORAGE ALLOCATORS	
BES	Allocate /e/ locations starting with $; put label at $ + /e/. (oprnd = e)
BSS	Allocate /e/ locations starting with $; put label at $. (operand = e)
CONDITIONAL ASSEMBLY DIRECTIVES ④	
IFGT	Skip to statement labeled 'name' if /e/ 0. (operand = e,name)
IFLT	Skip to statement labeled 'name' if /e/ 0. (operand = e,name)
IFNZ	Skip to statement labeled 'name' if /e/ = 0. (operand = e,name)
IFZO	Skip to statement labeled 'name' if /e/ = 0. (operand = e,name)
SKIP	Skip next /e/ statements if /e/ 0. (operand = e)

ASSEMBLER DIRECTIVE (Continued)

LINKAGE CONTROLS	
ENTRY	Enter symbol stated in operand as a virtual link.
EOP	End; link to library via label field entry if given.
MISCELLANEOUS	
DUP	Duplicate, /e/ times, statements to 'name'-1. (operand = e,name)
END	End; LOADOV uses operand of last END statement for /E location.
FORCE	Put next instruction at even $ if oprnd even, or at odd $ if oprnd odd.
INCLD	Open named source file to RANGE statements. (operand = name,drive#)
INEND	Close file opened by last INCLD statement.
LTRL	Define literal key. (operand = key)
ORG	Assign label to $, then set $ equal to operand.
RANGE	INCLD range of SNEDIT-numbered records given. (oprnd = first,last)
IV/70	Disables the use of IV/90 type instructions. When this directive is in effect, the "sbs" or "sbd" fields may not contain a value of 3 for decimal instructions.
IV/90	Must precede the use of the extended instruction set.
LISTING CONTROLS	
EJECT	Eject page.
NPRNT	Suppress listing except for errors, virtuals, and XREF.
PRNT	Enable listing.
SPACE	Skip 'n' lines. (operand = n)
TITLE	Entitle each page with operand.

Notes:

① $ = current location count; /e/ = expression as evaluated by assembler.

② If label field is used, label is assigned to $.

③ Label field entry is mandatory for these directives.

④ A "label" referenced by the operand of a conditional directive can be any combination of from one to six bytes; normal label rules do not apply.

Statement (record) Format:

1 6	8 13	15 (first blank)	80
LABEL	DIRECTIVE	OPERAND	COMMENTS

MACRO ASSEMBLER (MACROL) DIRECTIVES and CALLS

MACRO	Prototype directive (keyword form)
MACROX	Prototype directive (positional form)
MLOCAL	Macro local declarative directive
NOTE	NOTE directive
IF	IF directive
SIF	String IF directive
ELSE	ELSE directive for IF and SIF
ENDIF	End directive for IF and String IF\SIF)
LOOP	LOOP directive
EXIT	Exit from LOOP directive
ENDL	End of LOOP directive
MEXIT	Exit from macro directive
MEND	Macro ending directive

MACRO Format:	mname	MACRO [key parameter list]
Example:		HAPPY MACRO WHO,WHEN,WHERE = "HERE"
MACRO Call:		mname [key = string],[key = string], ...
Example:		HAPPY WHO = "ME",WHEN = "NOW"

```
MACROX Format:    mname     MACROX   [key parameter list]
Example:                    PERSON MACROX FIRST,LAST
MACROX Call:                mname    [string, string, ...]
Example:                    PERSON AARON,SMITH

MLOCAL Format:              MLOCAL   [local symbol list]
Example:                     MLOCAL A,SEQ.3
                  A         EQU      $
                  SEQ.3     LDA      0,X1
```

key parameter list = [parameter-name [= string]][,parameter-name [= string]]...

Note: The value of an explicitly declared local variable is undefined at the beginning of each call to the containing macro. A SET or LOOP directive in the path taken during call evaluation is required to establish its value before any reference to it (including parameter substitution).

```
Example:          BIRDY     MACRO    P=0
                            MLOCAL   COUNT,I,J
                  COUNT     SET      <P>
                  I         SET      COUNT+1
                  J         LOOP     -28,I
                            LDA      J+X
                            ENDL
                            MEND
```

NOTE Format: NOTE severity, message

where: severity: C = Comment, W = Warning, E = Error, D = Disaster
message: a quoted string

IF Format: IF first-expression, relationship, second-expression
(text for relationship true)
ELSE Format: ELSE
(text for relationship false)
ENDIF Format: ENDIF

where: relationship = EQ, NE, LT, GT, LE, GE

SIF Format: SIF first-string, relationship, second-string
(text for relationship true)
ELSE
(text for relationship false)
ENDIF

LOOP Format: [loop-var] LOOP initial-value, limit-value, [increment]
(text)
Optional EXIT: EXIT
(optional text)
ENDL Format: ENDL

where: loop-var = the loop variable. The loop variable is set to "initial-value" and incremented by the value "increment" after each pass through the loop. When the loop variable becomes greater than "limit-value," the loop executes an EXIT from the loop. The loop variable retains its value upon exit from the loop until it is used in a SET directive or in another LOOP directive.

MEXIT: Format: MEXIT

Use: Immediate, unconditional exit from a macro during a call on a macro. It is equivalent to jumping immediately to the MEND directive which is the normal macro ending directive.

MEND Format: MEND

MACRO IMPLICIT VARIABLE REFERENCE

Local implicitly declared variables specify the environment of a macro call. (Global implicitly declared variables specify the modes of assembly for the macro assembler in JCL statements or assembler directives.)

?	A question mark following a macro name specifies that the directive is read only.
< >	Angle brackets are used to indicate either a parameter substitution inside a macro body or a variable reference anywhere in a program.
<ASTERISKS?>	A count of the number of asterisks following a macro name.
<CALLS?>	A count of the number of calls to a macro.
<POS?>	A count of the highest parameter position called. Useful in handling variable length calls and in detecting error conditions.
<LACK?>	A count of the number of skipped parameters.
<var: SIZE?>	A count of the number of characters in "var" not including the delimiting quotation marks.
<var: TYPE?>	Defines type of variable as follows. The type chosen is the lowest value that qualifies: 0 Null string 1 Absolute now expression 2 Relocatable now expression 3 Expression (not defined in first pass) 4 String

Channel 3 — Voltage Windowing and Lamp Controls

(For Diagnostic Use Only)

Unit Address	Control
060	Vcc High
061	Vcc Low
062	Vcc Nom
063	Vdd High
064	Vdd Low
065	Vdd Nom
066	Vpg High
067	Vpg Low
070	Vpg Nom
071	Lamp 1 on *
072	Lamp 1 off
073	Lamp 2 on
074	Lamp 2 off
075	Run lamp on
076	Fail lamp on
077	Run & Fail lamp off

*Unit addresses 071 to 077 apply only to IV/40, IV/50, IV/60, IV/65, and IV/80

DEVICE ADDRESSES AND INTERRUPT LEVELS

Relative Address (2)	Fixed Memory	Level 1	Level 2	Level 3	Level 5	Level 6
00	INR Level 0			BRM 8122 0 (3)	BRM Kbrd 0 (1)	
1				BRM 8122 1	BRM Kbrd 1 (1)	
2	IOID Level 1			BRM 8122 2	BRM Kbrd 2 (1)	
3				BRM 8122 3	BRM Kbrd 3 (1)	
4	IOID Level 2			BRM 8122 4	BRM Kbrd 4 (1)	
5				BRM 8122 5	BRM Kbrd 5 (1)	
6	IOID Level 3			BRM 8122 6	BRM Kbrd 6 (1)	
7				BRM 8122 7	BRM Kbrd 7 (1)	
10	BRM Level 4			BRM 8122 10	BRM Kbrd 10 (1)	
11				BRM 8122 11	BRM Kbrd 11 (1)	
12	IOID Level 5			BRM 8122 12	BRM Kbrd 12 (1)	
13				BRM 8122 13	BRM Kbrd 13 (1)	
14	IOID Level 6			BRM 8122 14	BRM Kbrd 14 (1)	
15				BRM 8122 15	BRM Kbrd 15 (1)	
16	BRM Level 7			BRM 8122 16	BRM Kbrd 16 (1)	
17				BRM 8122 17	BRM Kbrd 17 (1)	
20		IO Sync Data Set 8437 (9)	IO Card Reader (8001/8003) Char rdy	BRM 8122 20 (3)	BRM Kbrd 20 (1)	
21		IO Sync Data Set 8437 (9)		BRM 8122 21	BRM Kbrd 21 (1)	
22		BRM Sync Data Set 8437 (9)	IO Card Reader (8001/8003) end of cd	BRM 8122 22	BRM Kbrd 22 (1)	
23		BRM Sync Data Set 8437 (9)		BRM 8122 23	BRM Kbrd 23 (1)	IO 8001-5xx/8003-5xx
24		IO Sync Data Set 8437 (9)	BRM 8231 or 8271 Disc	BRM 8122 24	BRM Kbrd 24 (1)	
25		IO Sync Data Set 8437 (9)	8231 or 8271 Disc (4)	BRM 8122 25	BRM Kbrd 25 (1)	
26		BRM Sync Data Set 8437 (9)	8231 or 8271 Disc (4)	BRM 8122 26	BRM Kbrd 26 (1)	
27		BRM Sync Data Set 8437 (9)	8231 Address (4)	BRM 8122 27	BRM Kbrd 27 (1)	
30		IO Sync Data Set 8437 (9)		BRM 8122 30	BRM Kbrd 30 (1)	BRM 8135,8145,8149,8154
31		IO Sync Data Set 8437 (9)		BRM 8122 31	BRM Kbrd 31 (1)	BRM 8131
32		BRM Sync Data Set 8437 (9)		BRM 8122 32	BRM Kbrd 32 (1)	
33		BRM Data Sync Set (7) (8)		BRM 8122 33	BRM Kbrd 33 (1)	
34			BRM 8261 Disc (NP80) (5)	BRM 8122 34	BRM Kbrd 34 (1)	
35		BRM Async Data Set (8411)		BRM 8122 35	BRM Kbrd 35 (1)	
36				BRM 8122 36	BRM Kbrd 36 (1)	BRM KB/Printer/Auto-Diel
37				BRM 8122 37	BRM Kbrd 37 (1)	BRM 8251 Diskette

Relative Address (1)	Fixed Memory	Level 1	Level 2	Level 3	Level 5	Level 6
40		IO Sync Data Set 8437 (8)	BRM 8241 Disc	BRM Kbrd 40 (2)		
41	BRM TRAP	IO Sync Data Set 8437 (8)	8241 Address (4)	BRM Kbrd 41 (2)		
42		BRM Sync Data Set 8437 (8)	8241 Address (A)	BRM Kbrd 42 (2)		
43		BRM Sync Data Set 8437 (8)	8241 Address (C)	BRM Kbrd 43 (2)		
44		IO Sync Data Set 8437 (8)	IO Mag Tape (8511/8512) data int	BRM Kbrd 44 (2)		
45		IO Sync Data Set 8437 (8)	BRM Mag Tape (8511/8512) status int	BRM Kbrd 45 (2)		
46		BRM Sync Data Set 8437 (8)	8511/8512 Address (A)	BRM Kbrd 46 (2)		
47		BRM Sync Data Set 8437 (8)	8511/8512 Address (C)	BRM Kbrd 47 (2)		
50		BRM Ch Adap (7071/7072) init int	IO Mag Tape (8513/8504) data int	BRM Kbrd 50 (2)		
51		BRM Ch Adap (7071/7072) cont int	BRM Mag Tape (8513/8504) status int	BRM Kbrd 51 (2)		
52		BRM Ch Adap (7071/7072) end int	8513 Address (4)	BRM Kbrd 52 (2)		
53		IO Ch Adap (7071/7072) data in/out	8513 Address (6)	BRM Kbrd 53 (2)		
54		BRM Ch Adap (7073) init int	IO Mag Tape (8507) data int	BRM Kbrd 54 (2)		
55		BRM Ch Adap (7073) cont int	BRM Mag Tape (8507) status int	BRM Kbrd 55 (2)		
56		BRM Ch Adap (7073) end int		BRM Kbrd 56 (2)		
57		IO Ch Adap (7073) data in/out		BRM Kbrd 57 (2)		
60	Video Area A48			BRM Kbrd 60 (2)		
61	Video Area A48			BRM Kbrd 61 (2)		
62	Video Area A48			BRM Kbrd 62 (2)		
63	Video Area A48			BRM Kbrd 63 (2)		
64	Video Area A48			BRM Kbrd 64 (2)		
65	Video Area A48			BRM Kbrd 65 (2)		
66	Video Area A48			BRM Kbrd 66 (2)		
67	Video Area A48			BRM Kbrd 67 (2)		
70	Video Area A48			BRM Kbrd 70 (2)		
71	Video Area A48			BRM Kbrd 71 (2)		
72	Video Area A48			BRM Kbrd 72 (2)		
73	Video Area A48			BRM Kbrd 73 (2)		
74	Video Area A48			BRM Kbrd 74 (2)		
75	Video Area A48			BRM Kbrd 75 (2)		
76	Video Area A48			BRM Kbrd 76 (2)		
77	Video Area A48			BRM Kbrd 77 (2)		

(1) Data ready, no error.
(2) Data ready, character lost for System IV/70 keyboards 0-37 and System IV/40 keyboards 0-17.
(3) The controller for the 8122 Printers can support 020 printers; another controller must be added for each block of 020 or fewer printers.
(4) Discs and magnetic tape units may use certain unit addresses without corresponding IOID table addresses being assigned.
(5) The 8261 Disc is actually controlled by the NP/80.
(6) On 64 keyboard systems, add octal 100 to relative address for keyboard 040 through 077.
(7) May be 8435, 8436, or 8437. If both 8436 and 8437 are present, then first 8437 is at relative address 34-37.
(8) The 8460 Controller may be used anywhere an 8437 is specified.

C2674A

STATUS WORDS

7071/7072 Channel Adapters (5)	0	Data Transfer Required (2)
	1	Attn or DE accepted
	2	— —
	3	System Reset (2)
	4	— —
	5	Printer Status Request (1)
	6-7	Type:
		0 = New Command (2) 2 = End Operation
		1 = Continue (2) 3 = End Operation
	8	Keyboard Lock
	9-11	New Command Code:
		0 = No Command 4 = — —
		1 = Write Buffer 5 = Read Manual Input
		2 = Write Line 6 = Short Read
		3 = Erase Buffer 7 = Read Full Buffer
	12-15	Line Address (New Command)
	14-15	Byte Boundary:
		0 = Word Full 2 = Left Two Bytes
		1 = Left Byte
	16-19	2848 Address
	20-23	Address within 2848

7073 Channel Adapter		
Word 0	0	BOOT Address Equal
	1	Upper ≠ and Lower ≠
	2-4	— —
	5	Convert
	6	Bus Out Parity Odd
	7	Not Enable Bus In
	8	IBI-0
	9	IBI-1
	10	IBI-2
	11	IBI-3
	12	IBI-4
	13	IBI-5
	14	IBI-6
	15	IBI-7
	16	IBO-0
	17	IBO-1
	18	IBO-2
	19	IBO-3
	20	IBO-4
	21	IBO-5
	22	IBO-6
	23	IBO-7
Word 1	0	Interrupt
	1	Upper Less or Equal
	2	— —
	3	BOOT
	4	Read Buffer Full
	5	Status Request
	6	Service Request
	7	— —
	8	Our Address Okay
	9	Text I/O
	10	No Op
	11	Sense
	12	Gate Address to AR
	13	— —
	14	Help Required
	15	Select A
	16	Select B
	17	Bus Out Check
	18	Command Reject
	19	Control Unit End
	20	Command Chain

STATUS WORDS (Continued)

	21	Channel End
	22	Device End
	23	Unit Check
Word 2	0	Interrupt Required 1 & 2
	1	Printer Address Equal
	2	Control Unit Required
	3	Channel Request
	4	Device Busy
	5	Control Unit Busy
	6	Adapter Busy
	7	Interface Busy
	8	Proceed
	9	— —
	10	360 System Reset
	11	— —
	12	Attention
	13	Done
	14-17	— —
	18	Byte Position 1
	19	Byte Position 2
	20	Byte Position 3
	21-23	— —
Word 3	0	Data Interrupt
	1	Upper Less
	2	Clocked Read Buffer Full
	3	Write Buffer Full
	4	Write 470
	5	Request Next Word
	6	Continue
	7	End Op
	8	Hold/Select Out Gated
	9	Allow Bus
	10	Stop
	11	Allow Request In
	12	Not Used
	13	Underflow
	14	First Byte
	15	Read
	16	Write
	17	Stack
	18	Bus and Status
	19-23	— —
7166 Communications Controller		
Word 1	0	Receiver Data Available
	1	Read Request — examine register 2
	2	Loop Status
	3	Flag Detected
	4	Clear To Send
	5	Transmitter Underrun
	6	Transmission Data Register Available
	7	Interrupt Request
Word 2	0	Address Present in Rx Data Register
	1	Frame Valid or Complete
	2	Inactive Idle Received
	3	Abort Received
	4	Frame Check Sequence/Invalid Frame
	5	Data Carrier Detect
	6	Receiver Overrun
	7	Receiver Data Available
7200 Series Keyboards	0-15	— —
	16	Data Ready
	17	— —

STATUS WORDS (Continued)

	18	Data Lost (4)
	19-23	Unit Address
8001/8003	0-20	— —
Card Readers	21	Card In Reader (3)
(unbuffered) (5)	22	Busy (3)
	23	Not Ready (3)
8001/8003	0-18	— —
Card Readers	19	Motion Check (6)
(buffered)	20	Reader Error (6)
	21	Hopper Check (6)
	22	Buffer Not Full (3) (6)
	23	Not Ready (3)
812X	0-14	— —
Character Printer	15	Printer Restart
Unit (7)	16	System Not Ready (3)
	17	Paper Out
	18	Check Condition (2)
	19	Print Head Not Ready
	20	Paper In Motion
	21	Printing
	22	Busy
	23	Not Ready
8121/8122	0-17	— —
Character Printer	18	Receiving or Scanning
Controller	19	Sending Data
8131 Character Printer	0-20	— —
	21	Not Ready (3)
	22	Output Needed (2)
	23	— —
Line Printers	0-20	— —
	21	Paper Out (4)
	22	Busy (3)
	23	Not Ready
8100 KB/Printer	0-20	— —
	21	Baud Rate Error
	22	Output Needed (2)
	23	Input Ready (2)
8230/8231	0-16	— —
Cartridge Disc Unit	17	Seek Incomplete Error (2) (6) (8)
	18	Head Range Error (2) (8)
	19	Header Error
	20	Too Late
	21	CRC Error
	22	Busy
	23	Not Ready (3)
8240/8241	0	Interface Not Available (3) (4)
Disc Storage Unit	1-2	Drive That Completed Seek Last
	3-4	Reserved Bits; Should be 0 (4)
	5	Read Only
	6	Defect Detected
	7-15	Current Cylinder Address
	16	File Unsafe
	17	Seek Incomplete Error (2) (4) (8)
	18	Head Range Error
	19	Header Error
	20	Too Late
	21	CRC Error
	22	Busy
	23	Not Ready (3)

STATUS WORDS (Continued)

8250/4450 Diskette Unit	0-18	– –
	19	Busy
	20	Not Ready
	21	Transfer Pending
	22	Operation Check
	23	Track 0
8261/8281/8291 Disc Drives Status Words %DD0	1	Sector Marker
	2	Drive Identification Code
	3	Drive Identification Code
	4	Drive Identification Code
	5	Always 0
	6	Rate error (10)
	7	Index mark
	8	Fault (file unsafe) (11)
	9	Seek error (12) (13)
	A	Register parity error (10)
	B	Header error (10)
	C	Bus parity error
	D	CRC error (10)
	E	Transfer Pending (14)
	F	Not ready (14)
8270 Disc Storage Unit	0-2	Not used
	3	Controller Type
	4	Disc Type
	5	Multiple IOID
	6-13	Not used. Should be 0.
	14	Protect Violation
	15	File Protect
	16	Reserved
	17	Seek Incomplete
	18	Head Out of Range
	19	Header Error
	20	Too Late
	21	CRC Error
	22	Busy
	23	Not Ready
8411 Asynchronous Data Set Controller	0-11	– –
	12	Long Space (2)
	13-15	– –
	16	Ring Indicator (2)
	17	Data Set Ready (1)
	18	Clear to Send (2)
	19	Carrier Detect (1)
	20	Parity Error
	21	Baud Rate Error
	22	Output Needed (2)
	23	Input Ready (2)
8420 Auto-Dial Controller	0-14	Not used
	15-18	Digit
	19	Abandon Call and Retry
	20	Data Set status
	21	Present Next Digit
	22	Data Line Occupied
	23	Power Indication

STATUS WORDS (Continued)

8435 Synchronous Data Set Controller (7)	0-8	– –
	9	Output Needed (2)
	10	Data Set Ready (1)
	11	Clear to Send
	12	Sync Received
	13	Data Lost
	14	Carrier Detect (1)
	15	Input Ready in 16-23 (2)
	16-23	(Data Byte)
8436-1 Synchronous Data Set Controller (buffered)	0-7	– –
	8	Transmitted Data Lost
	9	Output Needed (2)
	10	Data Set Ready (1)
	11	Clear to Send (1)
	12	Sync Received
	13	Received Data Lost
	14	Carrier Detect (1)
	15	Input Ready (2)
8436-(2-5) Synchronous Data Set Controller (buffered, PROM)	0-6	– –
	7	Ring Indicator (1)
	8	Transmitted Data Lost
	9	Output Needed (2)
	10	Data Set Ready (1)
	11	Clear to Send
	12	Sync Received
	13	– –
	14	Carrier Detect (1)
	15	Input Ready (2)
8460 Communication Controller Primary Register %B001	0	Transmit Ready. 1 = Transmit buffer empty
	1	Receive Ready. 1 = Receive buffer full
	2	Status OK if a read caused an interrupt
	3	Parity Flag. 1 = Parity error, 0 = normal
	4	Overrun Flag. 1 = Overrun, 0 = normal
	5	Framing Error Detect 1 = Framing Error
	6	1 = A data carrier has been detected
	7	DSR 1 = Line is open for communication
Secondary Register %B002 (read only)	0	Ring Indicator Enabled
	1	Line Type 0 = Current Loop 1 = EIA RS-232C
	2	Secondary Receive Status 1 = Input is true
	3	Transmit Interrupt Disabled
	4-7	– –
8501/8502/8507/8511/8512 Magnetic Tape Units (5)	0-5	– –
	6	800 BPI Selected (8707 only)
	7	556 BPI Selected (8507 only)
	8	200 BPI Selected (8507 only)
	9	Even Parity Selected (8507 only)
	10	Short
	11-12	Byte Boundary: (8507) / (8511/8512) 0 = all four types / 0 = all three bytes 1 = right byte / 1 = right byte 2 = right two bytes / 2 = right two bytes 3 = right three bytes / 3 = all three bytes
	13	Rewinding
	14	Stopping Motion (3) (6)
	15	Data Transfer Required (2)
	16	File Mark (6)
	17	Too Late (2)
	18	EOT (6)
	19	BOT (6)
	20	Write Protect

STATUS WORDS (Continued)

	21	Parity Error
	22	Busy (4)
	23	Not Ready
8503/8504/8513 Magnetic Tape Units (5)	0-5	– –
	6	Operation Complete (2) (6)
	7	1600 BPI ID
	8	Corrected Parity Error (6)
	9	Reject (2) (6)
	10	Short (6)
	11-12	Byte Boundary (same as 8511)
	13	Rewinding (6)
	14	– –
	15	Data Transfer Required (2) (6)
	16	File Mark (6)
	17	Too Late (6)
	18	EOT (6)
	19	BOT (6)
	20	Write Protect
	21	Uncorrectable Error (6)
	22	Busy
	23	Not Ready

Status Words Notes:

(1) Interrupt on change of status
(2) Interrupt on true
(3) Interrupt on false
(4) Special considerations; see PUP Manual
(5) Controller's IOID depends on transaction
(6) May interact with other status bits
(7) Status obtained by Data-In IO
(8) Reset by Control-Restore
(9) Used during diagnostic testing only. Loop-back plug must be installed
(10) Cleared by taking status after BUSY is 0
(11) Requires fault clear sequence
(12) Requires restore
(13) Interrupt requested on 0-to-1 transition
(14) Interrupt requested on 1-to-0 transition

CONTROL WORDS

7071/7072/7073 Channel Adapters		
Initialization	0-1	Must be 01
	2-3	– –
	4-11	Printer Address
	12-19	Upper Control Unit Address
	20-23	Lower Control Unit Address
Load Byte	0-13	Must be 0 ... 0
	14-15	Starting Byte 0 = Left 1 = Middle 2 = Right
	16-23	Byte Count
Load Bit Register	0-1	Must be 00
	2	Printer Not Available
	3-7	Must be 0
	8	No Conversion
	9	Printer Intervention Required
	10	Must be 1
	11	Printer Busy
	12-23	– –
Load Address Register	0-6	Must be 0 ... 0
	7	Device End Only
	8-11	Must be 0 ... 0

	12	Attention
	13	Channel End/Device End
	14-15	– –
	16-23	Control Unit Address
7166 Communications Controller		
Word 0	0	Address Control
	1	Receiver Interrupt Enable
	2	Tx Interrupt Enable
	3	RDSR Mode (DMA)
	4	TDSR Mode (DMA)
	5	Rx Frame
	6	Rx Reset
	7	Tx Reset
Word 1	0	Prioritized Status
	1	2-Byte/1-Byte Transfer
	2	Flag-mark Idle
	3	Frame Complete/TDRA Select
	4	Transmit Last Data
	5	CLR Rx Status
	6	CLR Tx Status
	7	RTS Control
Word 2	0	Logical Control Field Select
	1	Extended Control Field Select
	2	Auto Address Extension Mode
	3	01/11 Idle
	4	Flag Detected ; Status Enable
	5	Loop/Non-Loop Mode
	6	Go Active on Poll/Test
	7	Loop On-Line Control DTR
Word 3	0	Double/Single Flag Interframe Ctrl
	1	Word Length Select Transmit = 1
	2	Word Length Select Transmit = 2
	3	Word Length Select Receive = 1
	4	Word Length Select Receive = 2
	5	Transmit Abort
	6	Abort Extended
	7	NRZI NRZ
7200 Series Keyboards (Alarms) (1)	0-12	– –
	13-15	Must be 011
	16-21	Unit Address
	22-23	Must be 10
8001/8003 Card Readers (unbuffered)	0-21	– –
	22	Convert Hollerith to ASCII
	23	Select Unpacked Mode
8001/8003 Card Readers (buffered)	0-20	– –
	21	Clear Controller
	22	Convert Hollerith to ASCII
	23	Select Unpacked Mode
8121/8122 Character Printer		
General Command (2)	0	Must be 1
	1	"Program 1" light (1 = on)
	2	"Program 2" light (1 = on)
	3	"Program 3" light (1 = on)
	4-11	– –
	12	Must be 1
	13-17	– –
	18	Set "Program" lights as per 1-3
	19-20	– –

CONTROL WORDS (Continued)

	21-22	11 = Set ribbon up (print) 01 - Set ribbon down (nonprint)
	23	Restore Print Head
Print Command (10 char/inch) (2)	0	Must be 0
	1-11	— —
	12	Must be 0
	17-23	ASCII Character Code
Print Command (char/inch and direction specified) (2)	0	Must be 0
	1	0 = move head right K/60 inches 1 = move head left K/60 inches
	2-11	Binary value of "K"
	12	Must be 1
	13-16	— —
	17-23	ASCII Character Code
Head Position and Paper Command (2)	0	Must be 1
	1	0 = head right K/60 inches 1 = head left K/60 inches
	2-11	Binary value of "K"
	12	Must be 0
	13	0 = paper up L/48 inches 1 = paper down L/48 inches
	14-23	Binary value of "L"
8121/8122 Printer Controller	0-21	— —
	22	Stop Scan
	23	Continue Scan
81LP Line Printers	0-22	— —
	23	Select Character Mode
8230/8231 Cartridge Disc Unit	0	0 = Read, 1 = Write
	1	Seek
	2	Brute Force
	3	Restore
	4	Header Only
	5-7	— —
	8-11	Sector Count (3)
	12-19	Cylinder Address
	20-23	Track and Sector Address
8240/8241 Disc Storage System		
Word 1	0	— —
	1-6	Sector Count (3)
	7-15	Cylinder Address
	16-20	Initial Head Address
	21-23	Initial Sector Address
Word 2	0	0 = Read, 1 = Write
	1	Seek
	2	Brute Force
	3	Restore
	4	Header Only
	5	Resume Scan
	6	Enable Defect Detect
	7	Read CRC
	8	Short Read
	9-16	Short Read Count (3)
	17-23	— —
8250/4450 Diskette Unit	0-16	— —
	17	Write to track 43
	18	Read
	19	Write
	20	Step +1 track

	21	Step −1 track
	22	Reset Op Check
	23	Restore Head
8261/8281/8291 Disc Drives		
Word 0 %DD0	0-3	0
	4-A	− −
	B	Unit Select Tag
	C	Unit Select Most Significant Bit
	D-E	Unit Select
	F	Unit Select Least Significant Bit
Word 1 %DD2	0-F	Most Sig. Word; Header Word 1
Word 2 %DD4	0-F	Least Sig. Word; Header Word 2
Word 3 %DD6	0-F	DMA Address
Word 4 %DD8	0-3	− −
	4	Bus Bit B
	5	Bus Bit A
	6	Bus Bit 9
	7	Bus Bit 8
	8	Bus Bit 7
	9	Bus Bit 6
	A	Bus Bit 5
	B	Bus Bit 4
	C	Bus Bit 3
	D	Bus Bit 2
	E	Bus Bit 1
	F	Bus Bit 0
Word 5 %DDA	0-9	− −
	A	Set Interrupt Pending
	B	Reset Interrupt
	C	Header Only
	D	Control Tag 3 (read, write, rest)
	E	Control Tag 2 (set head)
	F	Control Tag 1 (seek)
8261/8281/8291 Disc Drive Bus Bits		
Control Tag 1	0-B	Cylinder Address
Control Tag 2	0	LSB
	1	Head Address
	2	MSB
	3-B	− −
Control Tag 3	0	Write Gate
	1	Read Gate
	2	Servo Offset + error recovery
	3	Servo Offset − error recovery
	4	Fault Clear (5)
	5	Not Used
	6	Return to 0 (restore)
	7	Data Strobe Early (error recovery) (6)
	8	Data Strobe Late (error recovery) (6)
	9-B	− −
8270 Disc Storage Unit	0	Read or Write
	1	Seek
	2	Brute Force
	3	Restore
	4	Header Only
	5	IOID Mode

CONTROL WORDS (Continued)

	5-9	Sector Count
	10-18	Cylinder Address
	19-20	Head Select
	21-23	Sector Address
8411 Asynchronous Data Set Controller	0-11	— —
	12	Force Long Space
	13-20	— —
	21	Suppress Parity
	22	Set Data Set Ready
	23	Set Request to Send
8420 Auto-Dial Controller	0-16	— —
	17	Test 4 — not used presently
	18	Test 3 — diagnostics
	19	Test 2 — diagnostics
	20	Test 1 — diagnostics
	21	Digit Present
	22	Call Request
	23	Reset
8435 Synchronous Data Set Controller	0-20	— —
	21	Reset Transmitter
	22	Set Request to Send
	23	Reset Receiver
8436-(1...) Buffered Synchronous Data Set Controllers	0-18	— —
	19	Set Data Terminal Ready
	20	Reset Data Terminal Ready
	21	Reset Transmitter
	22	Set Request to Send
	23	Reset Receiver
8460 Communication Controller Primary Register %B003	0	Transmit Control. 1 = transmit enabled
	1	DTR Must be true to maintain communications
	2	Receive Control. 1 = receive enabled
	3	Break Control. 1 = forces break.
	4	Resets any error flags in UART register.
	5	RTS Forced true (1) to get Clear To Send
	6-7	Operating Mode 00 = Normal 01 = Local Loop Back 10 = Auto Echo 11 = Remote Loop Back
Secondary Register %B004	0	Ring Indicator Control. 1 = Enables Ring
	1	Secondary Transmit. 0 = Forces SBA true
	2	0 = Transmit Ready, IRQ Enabled Indicator
	3	Tx Interrupt Status. 1 = interrupt disabled
	4-7	— —
8501/8502/8507/8511/8512 Magnetic Tape Units	0	Write
	1	Read
	2	Backward
	3	Skip
	4	Set Lower Density (8507 only)
	5	Erase
	6	Rewind
	7	File Mark
	8	Set Word Mode
	9	Reset
	10	Set Even Parity (8507 only)
	11	— —
	12-23	Byte Count

CONTROL WORDS (Continued)

8503/8504/8513 Magnetic Tape Units	0	Write
	1	Read
	2	Backward
	3	Skip
	4	— —
	5	Erase
	6	Rewind
	7	File Mark
	8	Enable Read Timout
	9	Reset
	10-23	Byte Count

Control Words Notes:

(1) CUT word holds controls; CUT + 1 not used.
(2) Commands sent as Data-Out IO words.
(3) Count = (n-1), where n is the number of sectors to be transferred.
(4) Used during testing only. Loop-back plug must be installed.
(5) The 8291 does not support the Data Strobe Option
(6) The 8281's bus bit 4 is a volume selector 1 = fixed medium 0 = removable medium

DISPLAY CODES

8437 Communication Controller	Halt	The microprocessor has stopped running instructions. Run the diagnostics.
	Par.	The parity checking hardware has been disabled.
	Off	May have been left in diagnostics mode.
	PROM	The ROM is controlling the operation of the card.
	Fail	Parity error while running user code.
	Par.	(LED) The number displayed is the bank where
	Error	the parity failure occurred. The CCC must be powered down and up to turn the LED off.
7166 Display Codes		
Steady:	0	Usually means that everything is okay
	1	RCC trying to boot Remote Display Processor
	2	The modem is not ready at the host site.
	3	Host processor has not loaded RCC RAM code
	4	Interrupt diagnostic failed. (1)
	5	RAM diagostic failed. (1)
	6	Processor diagnostic failed. (1)
	7	RCC is under the control of ABUDDT
	8	Parity Error, see below
	A	Sync Error, see below
	C	PROM checksum error. (1)
	D	Offline loaded diagnostic in progress
	E	Error; part of the blinking code below
	F	Power-on reset; processor is not running.

DISPLAY CODES (Continued)

Blinking:

Blinking codes display an "E" error flag for two seconds, followed by a one-second code and four 1/4 second codes.

1 Second Code	1/4 Second Code	Meaning
0-2		Error in RCC application code.
3	xxxx	BRK opcode at xxxx
4	00xx	Interrupt error; Counter Timer in error is xx. (1)
5	xxyy	RAM error: RAM diagnostic type is xx, RAM IC in error is yy. (1)
6	00xx	Processor error; processor opcode that failed is xx.
8	xxxx	Parity error; xxxx is the program counter
A	xxxx	Sync failure; xxxx is the top of the stack
C	xxxx	PROM checksum error; incorrect checksum is xxxx. (1)

(1) See "RCC ROM Diagnostics Summary

NOTES

Four-Phase Systems
The Distributed Processing Company

10700 North DeAnza Blvd., Cupertino, California 95014 • (408) 255-0900