MODCOMP

MAX IV REFERENCE DATA

ABBREVIATIONS

A	Memory address
BIT	Bit number (0 to 15)
B	Branch address
BR	Branch
C	Shift count (0 to 15)
CH	Channel number of DMP
CT	Context
D	Displacement (0 to 15)
EA	Effective address
IM	Instruction map
L	Interrupt level (0 to 15)
M	Memory
N	Signed displacement (-63 to +64)
NP	Number of pages
NR	Number of registers (1 to 16)
NW	Number of words (0 to 255); NW ≥ NR ≥ 1
OM	Operand map
PR	Protection register number
PSD	Program Status Doubleword
R	Destination register for R to R and M to R instructions, Source register for R to M instructions
RP	Double R register
RT	Triple R register
RQ	Quadruple R register
S	Source register
SP	Double S register
ST	Triple S register
SQ	Quadruple S register
SN	Executive service number
U	Input/output unit number
V	Value
X	Index register (1 to 7)
*	Indirect addressing (optional)
$	Current location counter

CONDITION CODES

Z	Zero result		C	Carry out
N	Negative result		O	Overflow result
Ch	Character classification (as below)			
NZOC	NZOC		NZOC	
1000	(	1100	,	0000 alphabetic
1001	)	1101	-	0001 unclassified
1010	*	1110	.	0011 space
1011	+	1111	/	0101 numeric

ADDRESS MODES

DIRECT	EA = ($ + 1)
INDEXED	EA = ($ + 1) + (X)
INDIRECT	EA = (($ + 1))
INDEXED INDIRECT	EA = (($ + 1)X + (X))
IMMEDIATE	EA = $ + 1
SHORT DISPLACED	EA = (R1) + D
SHORT INDEXED	EA = (X)
RELATIVE DISPLACED	EA = $ + N
BIT	EA_{word} computed as above EA_{bit} = BIT
BYTE	EA_{word} = (X) + (X v 1)/2 EA_{byte} = left if (X v 1) even, right if (X v 1) odd
STACK	EA = top of stack

Push (PSM) and pull (PLM) are complex instructions.
Their EA references a stack descriptor with addresses for:
- Low Stack Pointer
- Current Stack Pointer
- High Stack Pointer + 1
- Overflow/Underflow Return
- Save of R1 on Overflow/Underflow

The stack grows from high memory to low.

MACHINE INSTRUCTIONS (2)

Mnemonic and Operands	Op code	Operation	Cond. Code	Time (µs)
MOVE INSTRUCTIONS				
Load Bit			ZN	
LBR,R,BIT	65	in R		.56
LBRB,R,BIT B	75	in R, BR unconditionally		1.44
Load Byte			Ch	
LBX,R,X	AE	from M		1.44
Load Register			ZN	
LDM*,R,X A	E5	from M		1.60
LDMD*,RP,X A	CD-1	from M double		2.08
LDI,R V	ED	from M Immediate		1.04
LDS,R,D	F5	from M Short Displaced		1.44
LDX,R,X,	FD	from M Short Indexed		1.44
LDXD,RP,X	8D-1	from M Short Indexed double		1.04
Load File			ZN	
LFM*,R,X A	A4	from M		1.36+.56(NxR)
LFS,R,D	B4	from M Short Displaced		1.04+.56(NxR)
LFX,R,X	BC	from M Short Indexed		1.00+.56(NxR)
Store Byte				
SBX,R,X	AF	in M		2.32
Store Register				
STM*,R,X A	E6	in M		2.16
STMD*,RP,X A	CE-1	in M double		1.76
STI,R V	EE	in M Immediate		1.04
STS,R,D	F6	in M Short Displaced		2.00
STX,R,X	FE	in M Short Indexed		2.00
STXD,RP,X	8E-1	in M Short Indexed double		1.36
Store File				
SFM*,R,X A	A5	in M		1.76+.56(NxR)
SFS,R,D	B5	in M Short Displaced		1.20+.56(NxR)
SFX,R,X	BD	in M Short Indexed		1.20+.56(NxR)
Stack Manipulation			ZNCO	
PLM*,R,X A,NR,NW	BA	pull M into R(s)	4.76+.56(N)	N≤16
PSM*,R,X A,NR,NW	BB-1	push R(s) into M	4.76+.56(N)	N≤16
Move Lower Byte			Ch	
MLR,R,S	0C	R to R		.56
MBR,R,S	08	right R to R		.56
Move Upper Byte			ZNCO	
MUR,R,S	0B	R to R		.56
MBL,R,S	09	left R to R		.56
Interchange Bytes			Ch	
IBR,R,S	0A	R to R		.56
Interchange Registers			ZN	
IRM*,R,X A	B6	R and M		1.84
IRR,R,S	B7	R and R		.56
Transfer Register			ZN	
TRR,R,S	6D	to R		.56
TRRB,R,S B	7D	to R, BR if nonzero		1.68
TRRD,RP,SP	CD-0	to R double		.64
TRRQ,RQ,SQ	8D-0	to R quadruple		1.04
Shift Arithmetic			ZNCO	
RAS,R,C	2B	right single		.88+.16(C)
RAD,RP,C	2A-0	right double		.68+.16(C)
RAQ,RQ,C	2A-1	right quadruple		1.36+.16(C)
LAS,R,C,	2F	left single		.88+.16(C)
LAD,RP,C	2E-0	left double		.88+.16(C)
LAQ,RQ,C	2E-1	left quadruple		1.36+.16(C)
Shift Logical			ZNCO	
RLS,R,C	29	right single		.88+.16(C)
RLD,RP,C	28-0	right double		.88+.16(C)
RLQ,RP,C	28-1	right quadruple		1.36+.16(C)
LLS,R,C	2D	left single		.88+.16(C)
LLD,RP,C	2C-0	left double		.88+.16(C)
LLQ,RQ,C	2C-1	left quadruple		1.36+.16(C)
Rotate Register			ZNCO	
LRS,R,S	0F	left single		.64
Generate Mask			ZNO	
GMR,R,BIT	67	in R		.56
GMBR,R,BIT	77	in R, BR unconditionally		1.44

LOGICAL INSTRUCTIONS (3)

Zero Bit			ZNCO	
ZBR,R,BIT	62	in R		.56
ZBRB,R,BIT B	72	in R, BR if nonzero		1.68
ZBMM*,BIT,X A	81	in M		2.72
ZMMB*,BIT,X A,B	85	in M, BR if nonzero		2.80
ZBSM,BIT,D	91	in M Short Displaced		2.64
ZBSB,BIT,D B	95	in M Short Displaced, BR if nonzero		2.40
ZBXM,BIT,X	99	in M Short Indexed		2.64
ZBXB,BIT,X B	9D	in M Short Indexed, BR if nonzero		2.40
Extract (AND) Register			ZN	
ETR,R,S	6A	from R		.56
ETRB,R,S B	7A	from R, BR if nonzero		1.68
ETMM*,R,X A	C1	from M		2.80
ETMB*,R,X A,B	C5	from M, BR if nonzero		2.80
ETSM,R,D	D1	from M Short Displaced		2.64
ETSB,R,D B	D5	from M Short Displaced, BR if nonzero		2.40
ETXM,R,X	D9	from M Short Indexed		2.64
ETXB,R,X B	DD	from M Short Indexed, BR if nonzero		2.64
Extract (AND) Memory			ZN	
ETM*,R,X A	E2	from R		1.60
ETI,R V	EA	from R Immediate		1.04
ETS,R,D	F2	from R Short Displaced		1.44
ETX,R,X	FA	from R Short Indexed		1.44
OR Bit			ZNCO	
OBR,R,BIT	63	in R		.56
OBRB,R,BIT B	73	in R, BR unconditionally		1.44
OBMM*,BIT,X A	82	in M		2.40
OBSM,BIT,D	92	in M Short Displaced		2.64
OBXM,BIT,X	9A	in M Short Indexed		2.64
OR Register			ZN	
ORR,R,S	6B	and R		.56
ORRB,R,S B	7B	and R, BR if nonzero		1.68
ORMM*,R,X A	C2	and M		2.80
ORSM,R,D	D2	and M Short Displaced		2.64
ORXM,R,X	DA	and M Short Indexed		2.64
OR Memory			ZN	
ORM*,R,X A	E3	and R		1.60
ORI,R V	EB	and R Immediate		1.04
ORS,R,D,	F3	and R Short Displaced		1.44
ORX,R,X	FB	and R SHort Indexed		1.44
Exclusive OR Bit			ZNCO	
XBR,R,BIT	64	in R		.56
XBRB,R,BIT B	74	in R, BR in nonzero		1.68
Exclusive OR Register			ZN	
XOR,R,S,	6C	and R		.56
XORB,R,S B	7C	and R, BR if nonzero		1.68
Exclusive OR Memory			ZN	
XOM*,R,X A	E4	and R		1.60
XOI,R V	EC	and R Immediate		1.04
XOS,R,D,	F4	and R Short Displaced		1.44
XOX,R,X	FC	and R Short Indexed		1.44
Transfer One's Complement			ZN	
TOR,R,S	0D	R to R		.56

COMPARE AND TEST INSTRUCTIONS

Compare Bit ...,BR if less than or equal (≤ condition)			ZNCO	
CBMB*,BIT,X A,B,B	87	with M,		3.16
CBSB,BIT,D B,B	97	with M Short Displaced,		2.88
CBXB,BIT,X B,B	9E	with M Short Indexed,		2.88
Compare Register			ZNCO	
CRR,R,S	6E	with R		.56
CRRD,RP,SP	CF-0	with R double		.64
CRM*,R,X A	C3	with M		1.60
CRMB*,R,X A,B,B	C7	with M and BR on ≤ condition		3.20
CRMD*,RP,X A	CF-1	with M double		2.08
CRI,R	E9-1	with M Immediate		1.04
CRS,R,D	D3	with M Short Displaced.		1.44
CRSB,R,D B,B	D7	with M Short Displaced, BR on ≤ condition		2.88
CRX,R,X	DB	with M Short Indexed		1.44
CRXB,R,X B,B	DF	with M Short Indexed, BR on ≤ condition		2.88

Test Bit			ZNOC
TBR,R,BIT	66	in R	.56
TBRB,R,BIT B	76	in R, BR if one	1.68
TBMM*,BIT,X A	83	in M	1.60
TBMB*,BIT,X A,B	86	in M, BR if one	2.96
TBSM,BIT,D	93	in M Short Displaced	1.44
TBSB,BIT,D B	96	in M Short Displaced, BR if one	2.80
TBXM,BIT,X	9B	in M Short Indexed	1.44
TBXB,BIT,X B	9E	in M Short Indexed, BR if one	2.80
Test Register ..., BR if any one's compare			ZN
TERB,R,S B	7E	and R,	1.69
TRMB*,R,X A,B	C6	and M,	2.96
TRSB,R,D B	D6	and M Short Displaced,	2.40
TRXB,R,X B	DE	and M Short Indexed,	2.40

BRANCH INSTRUCTIONS

Hop (short branch)			.64 NB, 1.44 B
HOP,N	F7-0	unconditionally	"
HZR,N	A9-8	on condition code Z reset	"
HZS,N	A9-0	on condition code Z set	"
HNR,N	A8-8	on condition code N reset	"
HNS,N	A8-0	on condition code N set	"
HCR,N	AB-8	on condition code C reset	"
HCS,N	AB-0	on condition code C set	"
HOR,N	AA-8	on condition code O reset	"
HOS,N	AA-0	on condition code O set	"
HLS,N	AC-0	on less than condition	"
HLE,N	AD-0	on less than or equal condition	"
HGE,N	AC-8	on greater than or equal condition	"
HGT,N	AD-8	on greater than condition	"
HNH,N	A6-8	on magnitude not higher condition	"
HHI,N	A6-0	on magnitude higher condition	"

Branch Short Indexed			:64 NB, 1.44 B
BXZR,X	8F-9	on condition code Z reset	"
BXZS,X	8F-1	on condition code Z set	"
BXNR,X	8F-8	on condition code N reset	"
BXNS,X	8F-0	on condition code N set	"
BXCR,X	8F-B	on condition code C reset	"
BXCS,X	8F-3	on condition code C set	"
BXOR,X	8F-A	on condition code O reset	"
BXOS,X	8F-2	on condition code O set	"
BXLS,X	8F-4	on less than condition	"
BXLE,X	8F-5	on less than or equal condition	"
BXGE,X	8F-C	on greater than or equal condition	"
BXGT,X	8F-D	on greater than condition	"
BXNH,X	8F-E	on magnitude not higher condition	"
BXHI,X	8F-6	on magnitude higher condition	"

Branch and Link (parenthesized instructions do not save link)			
BLM*,R,X (BRU*,X) B	E7	unconditionally	1.44
BLI,R	EF	Immediate	.56
BLX,R,X (BRX,X) B	FF	Short Indexed unconditionally	1.44
BLT,N	F7-1	Indexed through Table	2.32
BLZR,R (BZR) B	A7-9	on condition code Z reset	1.12 NB,1.44B
BLZS,R (BZS) B	A7-1	on condition code Z set	"
BLNR,R (BNR) B	A7-8	on condition code N reset	"
BLNS,R (BNS) B	A7-0	on condition code N set	"
BLCR,R (BCR) B	A7-B	on condition code C reset	"
BLCS,R (BCS) B	A7-3	on condition code C set	"
BLOR,R (BOR) B	A7-A	on condition code O reset	"
BLOS,R (BOS) B	A7-2	on condition ocde O set	"
BLLS,R (BLS) B	A7-4	on less than condition	"
BLLE,R (BLE) B	A7-5	on less than or equal condition	"
BLGE,R (BGE) B	A7-C	on greater than or equal condition	"
BLGT,R (BGT) B	A7-D	on greater than condition	"
BLNH,R (BNH) B	A7-E	on magnitude not higher condition	"
BLHI,R (BHI) B	A7-6	on magnitude higher condition	"

Add Bit			ZNCO
ABR,R,BIT	60	in R	.56
ABRB,R,BIT B	70	in R, BR if nonzero	1.68
ABMM*,BIT,X A	80	in M	2.52
ABMB*,BIT,X A,B	84	in M, BR if nonzero	3.04
ABSM,BIT,D	90	in M Short Displaced	2.64
ABSB,BIT,D B	94	in M Short Displaced, BR if nonzero	2.40
ABXM,BIT,X	98	in M Short Indexed	2.64
ABXB,BIT,X B	9C	in M Short Indexed, BR if nonzero	2.40
Add Register			ZNCO
ADR,R,S	68	to R	.56
ADRB,R,S B	78	to R, BR if nonzero	1.68
ADRD,RP,SP (DAR)	22-0	to R double	.64
ADMM*,R,X A	C0	to M	2.80
ADMB*,R,X A,B	C4	to M, BR if nonzero	2.80
ADSM,R,D	D0	to M Short Displaced	2.64
ADSB,R,D, B	D4	to M Short Displaced, BR if nonzero	2.40
ADXM,R,X,	D8	to M Short Indexed	2.64
ADXB,R,X, B	DC	to M Short Indexed, BR if nonzero	2.64
Add Memory			ZNCO
ADM*,R,X A	E0	to R	1.60
ADMD*,RP,X A	CB-1	to R double	2.08
ADI,R V	E8	to R Immediate	1.04
ADS,R,D,	F0	to R Short Displaced	1.44
ADX,R,X	F8	to R Short Indexed	1.44

Subtract Bit			ZNCO
SBR,R,BIT	61	in R	.56
SBRB,R,BIT B	71	in R, BR if nonzero	1.68
Subtract Register			ZNCO
SUR,R,S	69	from R	.56
SURB,R,S, B	79	from R, BR in nonzero	1.68
SURD,RP,SP	C9-0	from R double	.64
Subtract Memory			ZNCO
SUM*,R,X A	E1	from R	1.60
SUMD*,RP,X A	C9-1	from R double	2.08
SUI,R V	E9	from R Immediate	1.04
SUS,R,D	F1	from R Short Displaced	1.80
SUX,R,D	F9	from R Short Indexed	1.80

Multiply Register			ZNCO
MPR,R,S	20	by R	2.24
MPRD,RP,SP	A2-0	by R double	3.52
Multiply Memory			ZNCO
MPM*,R,X A	A0	by R	3.48
MPMD*,RP,X A	A2-1	by R double	5.32
MPS,R,D	B0	by R Short Displaced	3.28
MPX,R,X	B8	by R Short Indexed	3.28

Divide...by Register			ZNCO
DVR,R,S	21	R	6.00
DVRD,RQ,SP	A3-0	R quadruple (by R double)	10.48
Divide...by Memory			ZNCO
DVM*,R,X A	A1	R	7.00
DVMD*,RQ,X A	A3-1	R quadruple (by M double)	11.80
DVS,R,D	B1	R Short Displaced	6.80
DVX,R,X	B9	R Short Indexed	6.80

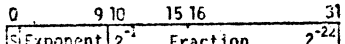
Transfer Two's Complement			ZNCO
TTR,R,S	6F	R to R	.56
TTRB,R,S B	7F	R to R, BR is nonzero	1.69
TTRD,RP,SP	8A-0	R to R double	.64
TTRQ,RQ,SQ	8B-0	R to R quadruple	1.04

Extend Sign			ZNCO
ESS,RP,S	8C-1	single	.96
ESD,RQ,SP	8C-0	double	1.44

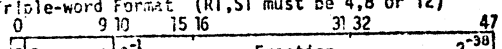
(Operand format shown below)

			ZNCO	
Add Register				
FAR,RP,SP	30-00	to R double		6.24
FARD,RT,ST	34-00	to R triple		8.48
FARQ,RQ,SQ	34-10	to R quadruple		10.16
Add Memory			ZNCO	
FAM,RP,X A	38-0	to R double		8.04
FAMD,RT,X A	3C-0	to R triple		11.12
FAMQ,RQ,X A	3C-1	to R quadruple		13.24
Subtract Register			ZNCO	
FSR,RP,SP	31-00	from R double		6.24
FSRD,RT,ST	35-00	from R triple		8.80
FSRQ,RQ,SQ	35-10	from R quadruple		10.24
Subtract Memory			ZNCO	
FSM,RP,X A	39-0	from R double		8.08
FSMD,RT,X A	3D-0	from R triple		11.20
FSMQ,RQ,X A	3D-1	from R quadruple		13.36
Multiply Register			ZNCO	
FMR,RP,SP	32-00	by R double		5.22
FMRD,RT,ST	36-00	by R triple		6.96
FMRQ,RQ,SQ	36-10	by R quadruple		
Multiply Memory			ZNCO	
FMM,RP,X A	3A-0	by R double		6.20
FMMD,RT,X A	3E-0	by R triple		9.13
FMMQ,RQ,X A	3E-1	by R quadruple		11.36
Divide...by Register			ZNCO	
FDR,RP,SP	33-00	R double		6.66
FDRD,RT,ST	37-00	R triple		10.64
FDRQ,RQ,SQ	37-10	R quadruple		11.16
Divide...by Memory			ZNCO	
FDM,RP,X A	3B-0	R double		8.74
FDMD,RT,X A	3F-0	R triple		11.20
FDMQ,RQ,X A	3F-1	R quadruple		14.44
Convert Double Precision Integer			ZNCO	
CDIF,RP,SP	30-01	to floating point		5.92
Convert Floating Point			ZNCO	
CFDI,RP,SP	34-01	to double precision integer		6.96
CQFF,RP,SQ	35-01	quadruple to floating point		3.12

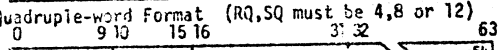
Double-word Format (RP,SP must be even and >0)



Triple-word Format (RT,ST must be 4,8 or 12)



Quadruple-word Format (RQ,SQ must be 4,8 or 12)



REQUEST AND NONPRIVILEGED CONTROL INSTRUCTIONS

Request			
REX,SN	23	Executive service	.56
RMI	01-0	multiprocessor interrupt(s)	.56
Null Operation			
NOP	F7-01	no operation	.64
Execute Instruction Word			?
EXR,R,S	B3	in R(s)	.88 + ?
(? - Condition code(s) and time of resultant executed instruction)			
Load Register with Current Interrupt			
LCIE,R	0E- 5	enable latches	.56
LCIA,R	0E- 4	active latches	.56
LCIP,R	0E- 6	request latches	.56
Load Register with...of PSD			
LCCC,R	0E- 3	current Condition Codes	.64
LCPS,R	0E- 2	current Program Status Register	.56
LCPR,R	0E- 1	current Program Register	.56
Select		(Reset to bits 12-15 of R)	
SCCC,R	B2-0	current Condition Codes in PSD	.56
Transfer and Reset			
TRO,R	0E- 0	overflow status history in PSD	.56

PRIVILEGED CONTROL INSTRUCTIONS

Halt			
HLT	00	program execution	.56
Enter Virtual Mode			
EVMO	01-10	of CPU execution	.56
Move			
MRBM,R,X	07	R-block section of CT file to M-file	9.20
MMRB,R,X	06	M-file to R-block section of CT file	9.44
Select			
SCRB	01-5	current R-block in PSD	2.72
Memory Allocation Initialize			
MABI	01-80	memory allocation block initialize	3.80
Memory Allocation			
AMEM,RP	01-C	allocate pages to map	4.96+.56 (NP)
DMEM,RP	01-A	deallocate pages from map image	5.20+1.36(NP)
Load			
LIMP,RP	01-D	IM image into hardware map	2.80+1.0(NP)
LOMP,RP	01-E	OM image into hardware map	2.80+1.0(NP)
Zero Section			
ZIMP,RP	01-9	of instruction map	2.88+.32(NP)
ZOMP,RP	01-B	of operand map	2.88+.32(NP)
Select			
SIOM,R	01-2	another program's IM as current OM	.56
SOOM,R	01-3	another program's OM as current OM	.56
SZOM	01-4	map zero as current OM	.56
Load Register			ZNCO
LDVM,R,RP	BE- 1	from M (via map image)	ZNCO 2.24
LDAM,R,RP	BE- 0	from actual M	ZN 1.60
Store Register			
STVM,R,RP	BF- 1	into M (via map image)	CO 1.68
STAM,R,RP	BF- 0	into actual M	.64
Set			
SPR	02-00	protect registers	.64
SGP,PR	02-8	global protect register	.64
SLP,PR	03-0	lower protect register	.64
SUP,PR	04-0	upper protect register	.64

INTERRUPT INSTRUCTIONS (privileged)

Set Interrupt Latch			
SIE,L	26-4	enable	1.04
SIA,L	26-0	active	1.04
SIR,L	26-8	request	1.04
Reset Interrupt Latch			
RIE,L	27-4	enable	1.04
RIA,L	27-0	active	1.04
RIR,L	27-8	request	1.04
Clear Interrupt			
CAR,RP	24-0	active latch and return	1.76
CIR,RP	25-0	active and request latch and return	2.72

INPUT/OUTPUT INSTRUCTIONS (privileged)

(i=A,B,C, or D, for which j = 0,1,2, or 3 respectively)

Input Status			ZNCO	
ISi,R,U	48+j	from I/O group i		1.90
Input Data			Ch	
IDi,R,U	4C+j	from I/O group i		1.90
Output Command				
OCi,R,U	40+j	to I/O group i		1.10
Output Data				
ODi,R,U	44+j	to I/O group i		1.10
Initialize				
DMPI,RP,CH	05	Direct Memory Processor		1.10

ASSEMBLER INSTRUCTIONS

Function Directive	Meaning
Program Sections and Addressing	
ABS	Absolute assembly mode
REL	Relocatable assembly mode
CTR	Select counter for assembly code
ATR	Assign attributes to counter
ORG	Set the location counter
RES	Reserve block of memory
BND	Set location counter to integral boundary
Symbol Definition	
EQU	Equate label to expression
SET	Equate name to expression
INT	Define internal name
EXT	Define external name
PGM	Define program name
COM	Define common block
CEQ	Equate label to common location
INC	Initialize common
Data Definition	
DFC	Define constant
DFF	Define fields within a memory word
DFP	Define fixed or floating point constant
Assembly Control	
INS	Insert source from alternate file
END	End of source program
EJT	Page eject
SPC	Space source listing
TTL	Define title
LST	Define listing type
NOL	Turn off listing
SCW	Suppress object control words
GCW	Generate object control words
Macro Definition	
MAC	Macro definition
MAD	Macro definition with default arguments
EMP	End of macro prototype
EMS*	End of resident prototypes
Conditional Assembly	
IF	Bypass code if both terms equal
IFN	Bypass code if terms not equal
IFA	Bypass code if argument is absolute
IFR	Bypass code if argument is relocatable
IFP	Bypass code if macro argument present
IFM	Bypass code if macro argument missing
IFI	Bypass code if argument is internal
IFE	Bypass code if argument is external
IFC	Bypass code if argument is relative to selected counter
IFF	Bypass code if file is assigned
INF	Bypass code if file is not assigned
IFS	Bypass code if in specified pass
IFO	Bypass code if option is selected
INO	Bypass code if option is not selected
GTC	Computed go to
GTO	Bypass source code unconditionally
EXM	Terminate macro evaluation
AOP	Assembler no operation

* Overlay macro assembler only

ASSEMBLER ERROR CODES

Error	Explanation
A	Address field error
B	Macro element expanded to more than 80 columns
C	Error in constant specification
D	Error in default argument
E	Expression error
F	Field specification error
G	IF or GTO search failure
I	Indirect addressing syntactically incorrect
L	Illegal label
M	Multiply defined symbol
N	Numeric field error
O	Operation code error
P	Phasing error
Q	Illegal usage of address mode
R	Invalid register designation
S	Syntax error
T	Truncation error; value specified is too large
U	Undefined symbol
V	Symbol table overflow
W	Illegal directive usage under SCW directive

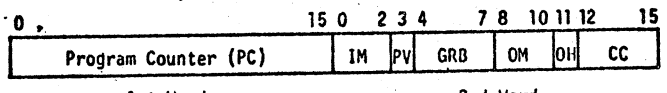
UFT FORMAT

Word	Operation	Set by	Purpose
0	DFC 0	I/O Sys	Status after operation
1	DFC @XYZ	User	File name
2	DFC #AOCC	User	Formats and options
3	DFC 0	User or I/O Sys	Device position index
4	DFC 0	I/O Sys	Byte count
5	DFC 0	I/O Sys	Pointer to assign list
6	DFC 0	User	Extended options
7	DFC 0	User	Mapping information
[8]	[DFC BUF] or R14	User	Buffer address (chain)
[9]	[DFC CNT] or R15	User	Buffer size

REX CALLS

Hex Code	Service	Hex Code	Service	Hex Code	Service
0	READ	16	ACTIVATE	39	CTA
1	WRITE	17	KILL	3A	BTD
2	REWIND	18	CONNECT	3B	HEX
3	BKFILE	19	UNCONNECT	3C	DTD
4	BKRECORD	1A	THAW	3F	DUMP
5	AVRECORD	1B	FREEZE	40	GETSYS
6	AVFILE	1C	CHANGE	43	GETASK
7	WEOF	1D	RELINQUISH	50	MODOPTION
8	HOME	23	TAKE	51	GETOPT
9	TERMINATE	24	GIVE	52	MODPOP
A	ASSIGN	27	ESTABLISH	53	GETPOP
B	TASSIGN	28	DEESTABLISH	54	SETVAR
C	I/O,WAIT	29	ALLOCATE	55	GETVAR
10	MESSAGE	2A	DEALLOCATE	56	DELVAR
11	WAIT	2D	LOVER	57	MARS
12	EXIT	34	GETPAR	58	CREPRIVATE
13	ABORT	35	COLLECT	59	INSGLO
14	DELAY	37	ATCAN	5A	INSPRI
15	RESUME	38	ATN	5C	EXTSHA

PROGRAM STATUS DOUBLEWORD



IM Instruction Map
OM Operand Map
PV Privilege State
0 = Unprivileged
1 = Privileged
OH Integer Overflow History
GRB General Purpose Register Block
CC Condition Codes
Bit 12 - Negative Result (N)
Bit 13 - Zero Result (Z)
Bit 14 - Overflow (O)
Bit 15 - Adder Carry Out (C)

ASCII Dec	Char.	Hex Code	ASCII Dec	Char.	Hex Code	ASCII Dec	Char.	Hex Code	ASCII Dec	Char.	Hex Code
0	NUL	00	32	SPACE	20	64	@	40	96	`	60
1	SOH	01	33	!	21	65	A	41	97	a	61
2	STX	02	34	"	22	66	B	42	98	b	62
3	ETX	03	35	#	23	67	C	43	99	c	63
4	EOT	04	36	$	24	68	D	44	100	d	64
5	ENQ	05	37	%	25	69	E	45	101	e	65
6	ACK	06	38	&	26	70	F	46	102	f	66
7	BEL	07	39	'	27	71	G	47	103	g	67
8	BS	08	40	(	28	72	H	48	104	h	68
9	HT	09	41	)	29	73	I	49	105	i	69
10	LF	0A	42	*	2A	74	J	4A	106	j	6A
11	VT	0B	43	+	2B	75	K	4B	107	k	6B
12	FF	0C	44	,	2C	76	L	4C	108	l	6C
13	CR	0D	45	-	2D	77	M	4D	109	m	6D
14	SO	0E	46	.	2E	78	N	4E	110	n	6E
15	SI	0F	47	/	2F	79	O	4F	111	o	6F
16	DLE	10	48	0	30	80	P	50	112	p	70
17	DC1	11	49	1	31	81	Q	51	113	q	71
18	DC2	12	50	2	32	82	R	52	114	r	72
19	DC3	13	51	3	33	83	S	53	115	s	73
20	DC4	14	52	4	34	84	T	54	116	t	74
21	NAK	15	53	5	35	85	U	55	117	u	75
22	SYN	16	54	6	36	86	V	56	118	v	76
23	ETB	17	55	7	37	87	W	57	119	w	77
24	CAN	18	56	8	38	88	X	58	120	x	78
25	EM	19	57	9	39	89	Y	59	121	y	79
26	SUB	1A	58	:	3A	90	Z	5A	122	z	7A
27	ESC	1B	59	;	3B	91	[	5B	123	{	7B
28	FS	1C	60	<	3C	92	\	5C	124	|	7C
29	GS	1D	61	=	3D	93	]	5D	125	}	7D
30	RS	1E	62	>	3E	94	^	5E	126	~	7E
31	US	1F	63	?	3F	95	_	5F	127	DEL	7F

CONVERSION TABLES

Hexadecimal-Decimal Conversion

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

Powers of 2

2^n	n
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1 024	10
2 048	11
4 096	12
8 192	13
16 384	14
32 768	15
65 536	16
2 147 483 648	31

$2^0 = 16^0$
$2^4 = 16^1$
$2^8 = 16^2$
$2^{12} = 16^3$
$2^{16} = 16^4$
$2^{20} = 16^5$
$2^{24} = 16^6$
$2^{28} = 16^7$
$2^{32} = 16^8$
$2^{36} = 16^9$
$2^{40} = 16^{10}$
$2^{44} = 16^{11}$
$2^{48} = 16^{12}$
$2^{52} = 16^{13}$
$2^{56} = 16^{14}$
$2^{60} = 16^{15}$

Powers of 16

16^n	n
1	0
16	1
256	2
4 096	3
65 536	4
1 048 576	5
16 777 216	6
268 435 456	7
4 294 967 296	8
68 719 476 736	9
1 099 511 627 776	10
17 592 186 044 416	11
281 474 976 710 656	12
4 503 599 627 370 496	13
72 057 594 037 927 936	14
1 152 921 504 606 846 976	15