CRAY RESEARCH, INC.

IOS Model C Hardware Reference Card

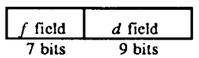
 CMQ11090A0

NOTATION

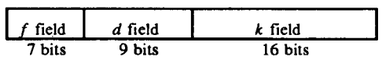
A	Accumulator
B	Operand register, index register (B register)
(B)	Contents of the operand register addressed by B
C	Carry Flag. 2^{16} of accumulator
E	Exit Stack pointer, 4 bits
(E)	Exit Stack storage location pointed to by E
I	Interrupt Enable flag
P	Program address register
R	Return jump program address
R!*sym*	Operand register with index *sym*, *sym* is any positive symbol <512
dd	Operand register with value of *dd*, *dd* is <512
[*dd*]	Value of symbol *dd*
(*dd*)	Memory parcel addressed by contents of operand register *dd*
k	Unsigned 16 bit value <65536
d	Unsigned numeric value <512
(*k*)	Memory parcel addressed by the value of *k*
(*dd* + *k*)	Memory parcel addressed by sum of operand register *dd* contents and contents of *k*
iod	3-character channel mnemonic, Ex. IOR, MOS...
&	Logical Product (perform AND function)
>	Shift right end off. Carry will shift with accum.
<	Shift left end off. Carry will shift with accum.
>>	Shift right circular through carry
<<	Shift left circular through carry
#	APML symbol for not equal, (≠)
†	Requires Maintenance Switch on
BZ	Busy
DN	Done
M.C.	Master Clear (MC)
L.M.	Local Memory

INSTRUCTION FORMAT

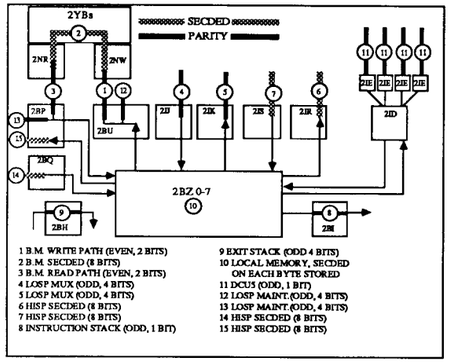
1-parcel instruction



2-parcel instruction



ERROR REPORTING PATHS



ASCII OCTAL CHARACTER SET

CHR	ASC	CHR	ASC	CHR	ASC	CHR	ASC	CHR	ASC
NUL	000	ESC	033	5	065	O	117	i	151
SOH	001	FS	034	6	066	P	120	j	152
STX	002	GS	035	7	067	Q	121	k	153
ETX	003	RS	036	8	070	R	122	l	154
EOT	004	VS	037	9	071	S	123	m	155
ACK	006	Sp.	040	:	072	T	124	n	156
BEL	007	!	041	;	073	U	125	o	157
BS	010	"	042	<	074	V	126	p	160
HT	011	#	043	=	075	W	127	q	161
LP	012	$	044	>	076	X	130	r	162
VT	013	%	045	?	077	Y	131	s	163
FF	014	&	046	@	100	Z	132	t	164
CR	015	'	047	A	101	[	133	u	165
S0	016	(	050	B	102	\	134	v	166
S1	017	)	051	C	103	]	135	w	167
DLE	020	*	052	D	104	^	136	x	170
DC1	021	+	053	E	105	_	137	y	171
DC2	022	'	054	F	106	`	140	z	172
DC3	023	-	055	G	107	a	141	{	173
DC4	024	,	056	H	110	b	142	|	174
NAK	025	/	057	I	111	c	143	}	175
SYN	026	0	060	J	112	d	144	~	176
ETB	027	1	061	K	113	e	145	DEL	177
CAN	030	2	062	L	114	f	146	ASCII>177	
EM	031	3	063	M	115	g	147	undefined	
SUB	032	4	064	N	116	h	150		

CHANNEL NOTES

Program Exit Stack & History Log (PXS)

†PXS : 16 Enter diag. mode. If 2^0 is set, force parity bits to 0's on a write. If bit 2^1 is set, force parity bits to 1's on a write. This mode is cleared by PXS : 0, or another PXS : 16 with A=0.

PXS : 13 Read History Log. Read back contents of jump history log. Once this instruction is used, do not use the stack with return jumps or allow interrupts until a PXS : 0 is issued. See diagnostic modes.

Machine Hardware Error (MHE)

This channel, if enabled, interrupts on Local Memory double bit errors, and Instruction Stack parity errors. To read status of errors see diagnostic modes.

Local Memory single bit syndrome codes:

L.M. Bit	Syndrome	Checkbit	Syndrome
0	23	8	1
1	31	9	2
2	32	10	4
3	13	11	10
4	34	12	20
5	15		
6	26		
7	7		

Buffer Memory Channel

MOS : 10

With accumulator bit 2^0 set, resulting accumulator value
- 2^0=Double Bit Error on B.M. Read
- 2^1=Write Path Parity Error
- 2^2=Read Path Parity Error
- 2^3=HISP Error on Bypass

With accumulator bit 2^1 set, resulting accumulator value
- 2^0=Bypass Mode Active to B.M.
- 2^1=Bypass Mode Active from B.M.
- 2^2=Previous transfer was Bypass to B.M. (Residue bit)
- 2^3=Previous transfer was Bypass from B.M. (Residue bit)

†MOS : 14 Set Control flags

With accumulator bit 2^1 set, Check Byte written=00000000_2

With accumulator bit 2^2 set, Disable Refresh

†MOS : 15 Set Control flags

With accumulator bit 2^1 set, parity is forced on R/W Path

With accumulator bit 2^2 set, Maintenance Mode is enabled. (parcel zero of a B.M. read will be check byte and syndrome)

With accumulator bit 2^3 set, disable SECDED

MOS : 16 SET BYPASS MODE

With accumulator set to 14, link B.M. to HISP input

With accumulator set to 15, link B.M. to HISP output

IPC

AOA : 1 Set Control flags

With accumulator bit 2^0 set, Master Clear

With accumulator bit 2^1 set, Deadstart

With accumulator bit 2^2 set, Dead Dump

With accumulator bit 2^3 set, Short Deadstart

With accumulator bit 2^2 set, 2^0 cleared, Buffer Memory and Error Logger interface is reset

CHANNEL NOTES

CONSOLES

TIA : 3 Set baud rate.

With accumulator set to 1, baud rate = 300
With accumulator set to 3, baud rate = 1200
With accumulator set to 4, baud rate = 2400
With accumulator set to 5, baud rate = 4800
With accumulator set to 6, baud rate = 9600 (default)
With accumulator set to 7, baud rate = 19200

LOSP Channel (20-27-DMA 1) (30-37-DMA 2)

CIA : 11
- 2^2- Parity Error flag
- 2^{14}-Sequence Error
- 2^{15}-Ready Data waiting

COA : 4
- 2^8- Write Disconnect
- 2^9- Hold Disconnect
- 2^{11}-Dead Dump (ch.21 only)
- 2^{12}-RTC Interrupt (ch.21 only)
- 2^{14}-I/O Master Clear
- 2^{15}-CPU Master Clear (ch.21 only)

COA : 11
- 2^{15}=sequence error

Maint. LOSP Channel (50/51-DMA 3)

LIA : 11
- 2^0- Parity Error flag for bits 2^0 - 2^3
- 2^1- Parity Error flag for bits 2^4 - 2^7
- 2^2- Parity Error flag for bits 2^8 - 2^{11}
- 2^3- Parity Error flag for bits 2^{12} - 2^{15}
- 2^{15}- Ready waiting flag

LOA : 4
- 2^8- Write Disconnect
- 2^9- Hold Disconnect
- 2^{14}- Master Clear

LOA : 11
- 2^1 - 2^0 IOP number
- 2^{15}- Sequence error

HISP Channel

HIA : 10 , HOA : 10 can be used on HISP Channel 14/15 - DMA 4 only. Remaining functions apply to both 14/15 and HISP Channel 16/17. See HR-0081 for function and status codes.

DIAGNOSTIC MODES

Channel	M.C.	Description
140403		Clear History Log address
140404		Log all jumps/incrementing read
140405	On	Log return jumps/decrementing read
140406		Turn History Log-off
140407	On	Turn History Log-on
140410 †		Instruction Stack parity mode
140411 †		Clear Error flags
140412 †		Disable check bits and error correct
140413 †		Sample into B reg. SECDED errors
140416 †		Read SECDED upper and parity flag
140417 †	On	Disable modes 410 through 416

† = Maintenance switch on the deadstart panel must be on.

M.C. column shows state after Master Clear.

GATE ARRAY PINOUTS

(A) 5/6, 8/7 = 13 14 15 16 + 9 10 11 + 1 2 3
(B) 10/9, 7/8, 6/5 = 11 13 + 14 15 + 16 1 + 2 3
(C) 7/8 = 5 6+9 10 + 11 13 + 14 15 + 16 1 + 2 3
(D) 7/8 = 2 6 5 1 16 9 13 14 11 15 10 3
P6 = C0 E1 E2 E3 B4 E5 + C1 E2 E3 B4 E5 +
C2 E3 B4 E5 + C3 B4 E5 + C4 E5 + C5 G
Macro: P6 = C5 C4 C3 C2 C1 C0 E5 B4 E3 E2 E1 G
(E) 5/6, 8/7,9/10, 14/13, 15/16, 1/2 = 3 11
(F) 10 9 8 7 1 16 15 14 = DCD(2 3 5)/13 11
6 = (13 11)'
Q7-0 = DCD(2^2 2^1 2^0)/E1 E0
(G) 10/9 7/8 6/5 = SUM(11 13 14, -)/3 2 1
16/15 = 11 13 14
(H) 8/7 = 6 5 3 2 1 16
9/10 = 11 13 14 15
(I) 10/9 = 11 13 14
6/5 = 3 2 1
8/7 = 15 16
(J) 8/7 = 6 5
9/10 = 11 5
16/15 = 14 13
1/2 = 3 13
(K) 9/10 = 13 14 11 + 15 16 3
7/8 = 1 2 11 + 5 6 3
(L) 8/7 = 6 5;13
9/10 = 11 5;13
16/15 = 14 5;13
1/2 = 3 5;13
(M) 6/5, 7/8, 10/9 = 15 16 1 + 11 14 + 2 3;13
(N) 10/9 = 3 15 16 5 8 11 1 6 7 14 2;13
P5 = C0 E1 E2 E3 E4 + C1 E2 E3 E4 + C2 E3 E4 +
C3 E4 F + C4 G;T
Macro: P5 = C4 C3 C2 C1 C0 E4 E3 E2 E1 F G;T
(O) 15/16 = MUX (14 1):DCD(5);13
10/9 = MUX (11 8):DCD(5);13
7/6 = MUX (2 3):DCD(5);13
Q = MUX (2^1 2^0): DCD (2^0);T
(P) 2/3 = 15 10 + 16 11 + 1 14;13
5/6 = 7 10 + 8 11 + 9 14; 13
(Q) 7/8 = SUM (14, 15, 16): (3, 2, 1)
5/6 = CARY (14, 15, 16): (3, 2 1)
10/9 = 11 13
(R) 14 = 12:DCD (6 7 9 10)/3 + 13
15 = 11:DCD (6 7 9 10)/3 + 13
1 = 4:DCD (6 7 9 10)/3 + 13
2 = 5:DCD (6 7 9 10)/3 + 13
(S) 1 = 17:DCD (14 13 12 11 10 8 7 6 5 4 3 2)/16 + 15
(T) 2/1 = SUM (8, 7, 3): (5, 14, 6); 13
16/15 = CARY (8, 7, 3): (5, 14, 6); 13
9/10 = 11; 13
(U) 7/8 = MUX (1 16 13 5): DCD (3 2)/6
10/9 = MUX (1 16 13 5): DCD (14 15)/11
Q = MUX (2^0 2^1 2^2 2^3): DCD (2^1 2^0)/E1
(W) 5/6, 8/7, 9/10, 14/13, 15/16, 1/2 = 3 11
(Y) 7/8, 6/5 = 10 9 + 14 15 + 16 1 + 2 3 + 11;13
(8) 3 = 1
2 = 16
11 = 14
10 = 13
7 = 9
6 = 8
(15) 6 = 7*8*9*10*11*13*15*16*14*1*2*3*

INSTRUCTIONS

IOP	APML	Description
000	PASS	No operation
001	EXIT	Exit from subroutine
002	I = 0	Disable system interrupts
003	I = 1	Enable system interrupts
004	A=A>*d*	Right shift C and A by *d* places, end off
005	A=A<*d*	Left shift C and A by *d* places, end off
006	A=A>>*d*	Right shift C and A by *d* places, circular
007	A=A<<*d*	Left shift C and A by *d* places, circular
010	A=*d*	Transmit *d* to A
011	A=A&*d*	Logical product of A and *d* to A
012	A=A+*d*	Add *d* to A
013	A=A-*d*	Subtract *d* from A
014	A=*k*	Transmit *k* to A
015	A=A&*k*	Logical product of A and *k* to A
016	A=A+*k*	Add *k* to A
017	A=A-*k*	Subtract *k* from A
020	A=*dd*	Transmit operand register *d* to A
021	A=A&*dd*	Logical product of A and operand register *d* to A
022	A=A+*dd*	Add operand register *d* to A
023	A=A-*dd*	Subtract operand register *d* from A
024	*dd*=A	Transmit A to operand register *d*
025	*dd*=A+*dd*	Add operand register *d* to A, result to operand register *d*
026	*dd*=*dd*+1	Transmit operand register *d* to A, add 1, result to operand register *d*
027	*dd*=*dd*-1	Transmit operand register *d* to A, subtract 1, result to operand register *d*
030	A=(*dd*)	Transmit contents of memory addressed by operand register *d* to A
031	A=A&(*dd*)	Logical product of A and contents of memory addressed by operand register *d*, result to A
032	A=A+ (*dd*)	Add contents of memory addressed by operand register *d* to A
033	A=A-(*dd*)	Subtract contents of memory addressed by operand register *d* from A
034	(*dd*)=A	Transmit A to memory addressed by operand register *d*
035	(*dd*)=A+(*dd*)	Add memory addressed by operand register *d* to A, result to same memory location
036	(*dd*)=(*dd*)+1	Transmit memory addressed by operand register *d* to A, add 1, result to same memory location
037	(*dd*)=(*dd*)-1	Transmit memory addressed by operand register *d* to A, subtract 1, result to same memory location
040	C=1 *iod*=DN	Set carry equal to 1 if channel *d* done
041	C=1, *iod*=BZ	Set carry equal to 1 if channel *d* busy
042	C=1, IOB-DN	Set carry equal to 1 if channel B done
043	C=1, IOB-BZ	Set carry equal to 1 if channel B busy
044	A=A>B	Right shift C and A by B places, end off
045	A=A<B	Left shift C and A by B places, end off
046	A=A>>B	Right shift C and A by B places, circular
047	A=A<<B	Left shift C and A by B places, circular
050	A=B	Transmit B to A
051	A=A&B	Logical product of A and B to A
052	A=A+B	Add B to A
053	A=A-B	Subtract B from A
054	B=A	Transmit A to B
055	B=A+B	Add B to A, result to B
056	B=B+1	Transmit B to A, add 1, result to B
057	B=B-1	Transmit B to A, subtract 1, result to B
060	A=(B)	Transmit operand register B to A
061	A=A&(B)	Logical product of A and operand register B to A
062	A=A+(B)	Add operand register B to A
063	A=A-(B)	Subtract operand register B from A

IOP	APML	Description
064	(B)=A	Transmit A to operand register B
065	(B)=A+(B)	Add operand register B to A, result to operand register B
066	(B)=(B)+1	Transmit operand register B to A, add 1, result to operand register B
067	(B)=(B)-1	Transmit operand register B to A, subtract 1, result to operand register B
070	P=p+*d*	Jump to P+*d*
071	P=p-*d*	Jump to P-*d*
072	R=p+*d*	Return jump to P+*d*
073	R=p-*d*	Return jump to P-*d*
074	P=*dd*	Jump to address in operand register *d*
075	P=*dd*+*k*	Jump to sum of *k* and operand register *d*
076	R=*dd*	Return jump to address in operand register *d*
077	R=*dd*+*k*	Return jump to address sum of *k* and operand register *d*
100	P=p+*d*, C=0	Jump to p+*d* if carry=0
101	P=p+*d*, C#0	Jump to P+*d* if carry ≠0
102	P=p+*d*, A=0	Jump to P+*d* if A=0
103	P=p+*d*, A#0	Jump to P+*d* if A≠0
104	P=p-*d*, C=0	Jump to P-*d* if carry =0
105	P=p-*d*, C#0	Jump to P-*d* if carry ≠0
106	P=p-*d*, A=0	Jump to P-*d* if A=0
107	P=p-*d*, A#0	Jump to P-*d* if A≠0
110	R=p+*d*, C=0	Return jump to P+*d* if carry=0
111	R=p+*d*, C#0	Return jump to P+*d* if carry≠0
112	R=p+*d*, A=0	Return jump to P+*d* if A=0
113	R=p+*d*, A#0	Return jump to P+*d* if A≠0
114	R=p-*d*, C=0	Return jump to p-*d* if carry=0
115	R=p-*d*, C#0	Return jump to p-*d* if carry≠0
116	R=p-*d*, A=0	Return jump to p-*d* if A=0
117	R=p-*d*, A#0	Return jump to p-*d* if A≠0
120	P=*dd*, C=0	Jump to address in operand register *d* if carry=0
121	P=*dd*, C#0	Jump to address in operand register *d* if carry≠0
122	P=*dd*, A=0	Jump to address in operand register *d* if A=0
123	P=*dd*, A#0	Jump to address in operand register *d* if A ≠ 0
124	P=*dd*+*k*, C=0	Jump to address in operand register *d*+*k* if carry=0
125	P=*dd*+*k*, C#0	Jump to address in operand register *d*+*k* if carry≠0
126	P=*dd*+*k*, A=0	Jump to address in operand register *d*+*k* if A=0
127	P=*dd*+*k*, A#0	Jump to address in operand register *d*+*k* if A≠0
130	R=*dd*, C=0	Return jump to address in operand register *d* if carry=0
131	R=*dd*, C#0	Return jump to address in operand register *d* if carry≠0
132	R=*dd*, A=0	Return jump to address in operand register *d* if A=0
133	R=*dd*, A#0	Return jump to address in operand register *d* if A≠0
134	R=*dd*+*k*, C=0	Return jump to address in operand register *d*+*k* if carry=0
135	R=*dd*+*k*, C#0	Return jump to address in operand register *d*+*k* if carry≠0
136	R=*dd*+*k*, A=0	Return jump to address in operand register *d*+*k* if A=0
137	R=*dd*+*k*, A#0	Return jump to address in operand register *d*+*k* if A≠0
140- 157	*iod* : 0-17	Channel *d* function 0-17
160- 177	IOB: 0-17	Channel B function 0-17

CHANNEL FUNCTIONS

Device	APML	Function
I/O REQUEST CHANNEL 0	IOR : 10	Read interrupting channel number.
PROGRAM FETCH CHANNEL 1	PFR : 0	Clear P.P.R flag
	PFR : 6	Disable channel interrupts
	PFR : 7	Enable channel interrupts
	PFR : 10	Read operand register No.
PROGRAM EXIT STACK CHANNEL 2 • See channel notes	PXS : 0	Clear Exit Stack boundary flag
	PXS : 6	Disable channel interrupts
	PXS : 7	Enable channel interrupts
	PXS : 10	Read Exit Stack pointer, E.
	PXS : 11	Read Exit Stack address, (E).
	PXS : 13	Read History Log
	PXS : 14	Enter Exit Stack pointer, E.
	PXS : 15	Enter Exit Stack address, (E).
†	PXS : 16	Enter diagnostic mode
MACHINE HARDWARE ERROR CHANNEL 3 • See channel notes	MHE : 0	Clear L.M./Inst.Stk. error flag
	MHE : 6	Disable channel interrupts
	MHE : 7	Enable channel interrupts
REAL-TIME CLOCK CHANNEL 4	RTC : 0	Clear Channel Done flag
	RTC : 6	Disable channel interrupts
	RTC : 7	Enable channel interrupts
	RTC : 10	Read real time clock
BUFFER MEMORY CHANNEL 5 • See channel notes DMA 3	MOS : 0	Clear Channel Busy/Done
	MOS : 1	Enter L.M. address for next transfer
	MOS : 2	Enter upper bits of Buff. Mem. address
	MOS : 3	Enter lower bits of Buff. Mem. address
	MOS : 4	Read Buff. Mem. to L.M., enter block length
	MOS : 5	Write Buff. Mem. from L.M., enter block length
	MOS : 6	Disable channel interrupts
	MOS : 7	Enable channel interrupts
	MOS : 10	Read errors
†	MOS : 14	Set Control flags
†	MOS : 15	Set Control flags
	MOS : 16	Enter bypass mode
INTERPROCESSOR COM. CHANNELS 6,10,12	AIA : 0	Clear Channel Busy/Done
	AIA : 6	Disable channel interrupts
	AIA : 7	Enable channel interrupts
	AIA : 10	Read input to accumulator and resume channel
INTERPROCESSOR COM. CHANNELS 7,11,13 (AOA,AOB,AOC) • See channel notes	AOA : 0	Clear Channel Busy/Done
	AOA : 1	Enter control bits from accum.
	AOA : 6	Disable channel interrupts
	AOA : 7	Enable channel interrupts
	AOA : 14	Set Channel Busy /output accumulator data

Device	APML	Function
INPUT FROM LOSP (CIA,CIB,LIA) 20 30 50 22 32 24 34 26 36 • See channel notes	CIA : 0	Clear Channel Busy/Done
	CIA : 1	Enter L.M. address, start input
	CIA : 2	Enter parcel count
	CIA : 3	Clear Channel error flags
	CIA : 4	Clear Ready Waiting flag
	CIA : 6	Disable channel interrupts
	CIA : 7	Enable channel interrupts
	CIA : 10	Read L.M. address
	CIA : 11	Read errors/status
OUTPUT TO LOSP (COA,COB,LOA) 21 31 51 23 33 25 35 27 37 • See channel notes	COA : 0	Clear Channel Busy/Done
	COA : 1	Enter L.M. address,start output
	COA : 2	Enter parcel count
	COA : 3	Clear Sequence Error flag
	COA : 4	Set/clear external controls
	COA : 6	Disable channel interrupts
	COA : 7	Enable channel interrupts
	COA : 10	Read L.M. address
	COA : 11	Read error flag
INPUT FROM MEMORY HISP CHANNEL (HIA). • See channel notes	HIA : 0	Clear Channel Busy/Done
	HIA : 1	Enter L.M. address
	HIA : 2	Enter upper Cray Memory address (2^9 - 2^{23})
	HIA : 3	Enter lower Cray Memory address (2^0 - 2^8)
	HIA : 4	Read Cray Memory, enter block length
	HIA : 6	Disable channel interrupts
	HIA : 7	Enable channel interrupts
†	HIA : 10	Return Error information
†	HIA : 14	Enter diagnostic mode
OUTPUT TO MEMORY HISP CHANNEL (HOA). • See channel notes	HOA : 0	Clear Channel Busy/Done
	HOA : 1	Enter L.M. address
	HOA : 2	Enter upper Cray Memory address (2^9 - 2^{23})
	HOA : 3	Enter lower Cray Memory address (2^0 - 2^8)
	HOA : 5	Write Cray Memory, enter block length
	HOA : 6	Disable channel interrupts
	HOA : 7	Enable channel interrupts
†	HOA : 10	Return Error information
†	HOA : 14	Enter diagnostic mode
CONSOLE KEYBOARD (TIA,TIB,TIC...) CHANNELS 40, 42, 44, 46 •See channel notes	TIA : 0	Clear Channel Done flag
	TIA : 3	Set baud rate
	TIA : 6	Disable channel interrupts
	TIA : 7	Enable channel interrupts
	TIA : 10	Read data and clear Done flag
CONSOLE DISPLAY (TOA,TOB,TOC...) CHANNELS 41, 43, 45, 47	TOA : 0	Clear Channel Busy/Done
	TOA : 6	Disable channel interrupts
	TOA : 7	Enable channel interrupts
	TOA : 14	Send accumulator data to display

Device	APML	Function
DISK STORAGE UNIT (DCU-5) CHANNELS 20-37 DMA 0,1,2,5	DIA : 0	Clear channel controller
	DIA : 1	Drive control functions
	DIA : 2	Request Read
	DIA : 3	Request Write
	DIA : 4	Diagnostic echo
	DIA : 5	Select cylinder
	DIA : 6	Disable channel interrupts
	DIA : 7	Enable channel interrupts
	DIA : 10	Read I/O Memory addr. reg. 0
	DIA : 11	Read I/O Memory addr. reg. 1
	DIA : 12	Read status register 0
	DIA : 13	Read status register 1
	DIA : 14	Enter I/O Memory Addr reg. 0
	DIA : 15	Enter I/O Memory Addr reg. 1
	DIA : 16	Enter next read/write parameter
	DIA : 17	Select special cntr mode/status
PERIPHERAL EXPANDER CHANNEL 17 DMA 0	EXB: 0	Idle channel.
	EXB: 1	Data input from A register (DIA).
	EXB: 2	Data input from B register (DIB).
	EXB: 3	Data input from C register (DIC).
	EXB: 4	Read interrupt number
	EXB: 5	Load device address
	EXB: 6	Send interface mask (MSKO)
	EXB: 7	Set interrupt mode
	EXB: 10	Read data bus status
	EXB: 11	Read status 1
	EXB: 13	Read status 2
	EXB: 14	Data output to A register (DOA)
	EXB: 15	Data output to B register (DOB)
	EXB: 16	Data output to C register (DOC)
	EXB: 17	Send control
BLOCK MULTIPLEXER CHANNEL (BMA,BMB...BML) 20-33 DMA 0,1,2,5	BMA : 0	Clear channel control
	BMA : 1	Send reset functions
	BMA : 2	Channel command
	BMA : 3	Read request in addr (wait request in)
	BMA : 4	Single byte I/O (wait service in/status in)
	BMA : 5	Maintenance delay Accumulator = 0 10 Micro second; 1 5 Micro second; 2 2.5 Micro second; 3 1.0 Micro second
	BMA : 6	Disable channel interrupts
	BMA : 7	Enable channel interrupts
	BMA : 10	Read L.M. address
	BMA : 11	Read byte count
	BMA : 12	Read status
	BMA : 13	Read input tags
	BMA : 14	Enter L.M. address
	BMA : 15	Enter byte count
	BMA : 16	Enter address
	BMA : 17	Enter output tags