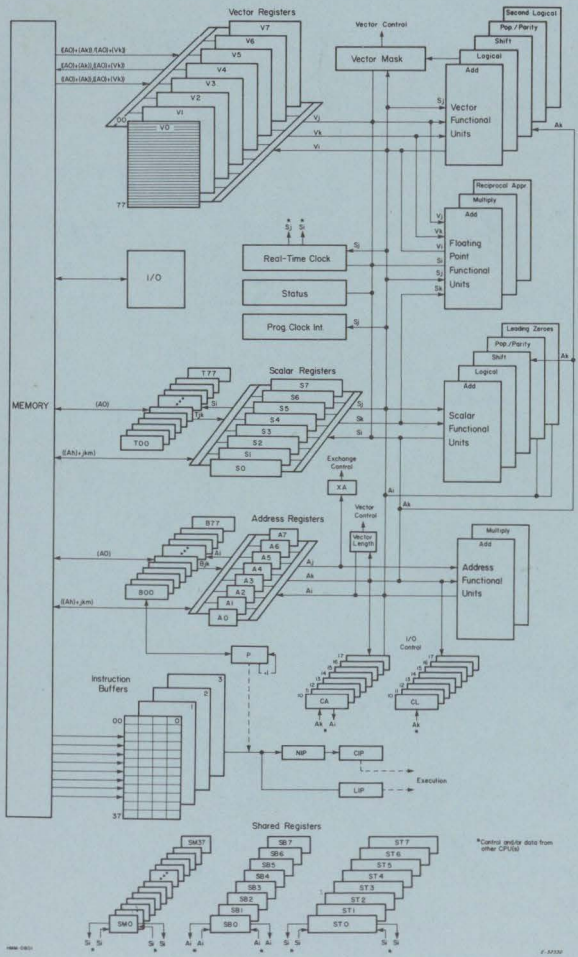
CRAY RESEARCH, INC.

CRAY X-MP CAL VERSION 1 HARDWARE REFERENCE CARD

CRAY X-MP BLOCK DIAGRAM



INSTRUCTIONS

CRAY X-MP	CAL	Unit	Description
000000	ERR	N/A	Error exit
††0010jk	CA,Aj Ak	N/A	Set channel (Aj) CA register to (Ak) and begin I/O sequence
††0011jk	CL,Aj Ak	N/A	Set channel (Aj) CL register to (Ak)
††0012j0	CI,Aj	N/A	Clear channel (Aj) Interrupt and Error flags; clear device Master Clear (output channel)
††0012j1	MC,Aj	N/A	Clear channel (Aj) Interrupt and Error flags, set device Master Clear (output channel); clear device ready-held (input channel)
††0013j0	XA Aj	N/A	Enter XA register with (Aj)
††0014j0	RT Sj	N/A	Enter RTC register with (Sj)
††0014j1	¶SIPI *exp*	N/A	Set interprocessor interrupt request of CPU *exp*; $0 \leq exp \leq 5$ (0 = no-op)
††001402	¶CIPI	N/A	Clear interprocessor interrupt
††0014j3	CLN *exp*	N/A	Select CLN register *exp*; $0 \leq exp \leq 5$ (0 = no-op)
††0014j4	PCI Sj	N/A	Enter II register with (Sj)
††001405	CCI	N/A	Clear programmable clock interrupt (PCI) request
††001406	ECI	N/A	Enable PCI request
††001407	DCI	N/A	Disable PCI request
††0015j0	¶¶	N/A	Select performance monitor
††001501	¶¶	SECDED	Set maintenance read mode; must be in maintenance mode (MM)
††001511	¶¶	SECDED	Load diagnostic check byte with S1; must be in MM
††001521	¶¶	SECDED	Set maintenance write mode 1; must be in MM
††001531	¶¶	SECDED	Set maintenance write mode 2; must be in MM
00200k	VL Ak	N/A	Transmit (Ak) to VL register
†002000	VL 1	N/A	Transmit 1 to VL register
002100	EFI	N/A	Enable interrupt on floating-point error
002200	DFI	N/A	Disable interrupt on floating-point error
002300	ERI	N/A	Enable operand range error interrupts
002400	DRI	N/A	Disable operand range error interrupts
002500	DBM	N/A	Disable bidirectional memory transfers
002600	EBM	N/A	Enable bidirectional memory transfers
002700	CMR	N/A	Complete memory references
0030j0	VM Sj	N/A	Transmit (Sj) to VM register
†003000	VM 0	N/A	Clear VM register
0034jk	SMjk 1,TS	N/A	Test & set semaphore jk $0 \leq jk \leq 37_8$
0036jk	SMjk 0	N/A	Clear semaphore jk $0 \leq jk \leq 37_8$
0037jk	SMjk 1	N/A	Set semaphore jk $0 \leq jk \leq 37_8$
004000	EX	N/A	Normal exit
0050jk	J Bjk	N/A	Jump to (Bjk)
006ijkm	J *exp*	N/A	Jump to *exp*
007ijkm	R *exp*	N/A	Return jump to *exp*; set B00 to (P) + 2
010ijkm	JAZ *exp*	N/A	Branch to *exp* if (A0) = 0 (bit 2^2 of i = 0)
011ijkm	JAN *exp*	N/A	Branch to *exp* if (A0) ≠ 0 (bit 2^2 of i = 0)
012ijkm	JAP *exp*	N/A	Branch to *exp* if (A0) positive; 0 is positive (bit 2^2 of i = 0)
013ijkm	JAM *exp*	N/A	Branch to *exp* if (A0) negative (bit 2^2 of i = 0)
014ijkm	JSZ *exp*	N/A	Branch to *exp* if (S0) = 0 (bit 2^2 of i = 0)
015ijkm	JSN *exp*	N/A	Branch to *exp* if (S0) ≠ 0 (bit 2^2 of i = 0)
016ijkm	JSP *exp*	N/A	Branch to *exp* if (S0) positive; 0 is positive (bit 2^2 of i = 0)
017ijkm	JSM *exp*	N/A	Branch to *exp* if (S0) negative (bit 2^2 of i = 0)
010ijkm	¶Ah *exp*	N/A	Transmit *exp* = ijkm to A0 (bit 2^2 of i = 1)
011ijkm	¶Ah *exp*	N/A	Transmit *exp* = ijkm to A1 (bit 2^2 of i = 1)
012ijkm	¶Ah *exp*	N/A	Transmit *exp* = ijkm to A2 (bit 2^2 of i = 1)

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
Marco P6 = C5 C4 C3 C2 C1 C0 E5 E4 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)'

(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
Marco P5 = C4 C3 C2 C1 C0 E4 E3 E2 E1 F G

(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

(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

(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*

CRAY X-MP LOSP CHANNELS

LOSP	Cables	
10	LOSP 0 Input	} MCU
11	LOSP 0 Output control	} MCU
12	LOSP 1 Input	
13	LOSP 1 Output	
14	LOSP 2 Input	
15	LOSP 2 Output	
16	LOSP 3 Input	
17	LOSP 3 Output	

LOSP	1 CPU	2 CPUs	4 CPUs
Channels available	10 - 17 or 10 - 13	10 - 17	10 - 17

HISP 1 and 2 CPUs	Cables	
HISP A pair	HISP 0 In A1 Data	HISP 0 Out B1 Data
	In A2 Data	Out B2 Data
	In A3 Data	Out B3 Data
	In A4 Control	Out B4 Control
	In A5 Control	
HISP B pair	HISP 2 In E1 Data	HISP 2 Out F1 Data
	In E2 Data	Out F2 Data
	In E3 Data	Out F3 Data
	In E4 Control	Out F4 Control
	In E5 Control	

HISP 4 CPUs	Cables	
HISP C pair	HISP 4 In I1 Data	HISP 4 Out J1 Data
	In I2 Data	Out J2 Data
	In I3 Data	Out J3 Data
	In I4 Control	Out J4 Control
	In I5 Control	
HISP D pair	HISP 6 In M1 Data	HISP 6 Out N1 Data
	In M2 Data	Out N2 Data
	In M3 Data	Out N3 Data
	In M4 Control	Out N4 Control
	In M5 Control	

VARIOUS MCU BASIC COMMANDS

DS - Dead Start
MC - Master Clear
DR - Display Right
DL - Display Left
DX - Display Exchange Package
DT - Display Text
DF - Display Forward
DB - Display Backward
DE - Display Memory Error
S - Set Memory/Parcel
S+ - Set Memory/Parcel + 1

:FI - Display All Files
:LO - Load File Or (/)
:SA - Save File
:DE - Delete File
:RE - Stop Refresh
:SN - Snap Display
:KI - Kill Mode
:TD - Test Dead Mode
:DD - Dead Dump Mode
:TB - Test Basic Mode
:RU - Run Mode