CRAY RESEARCH, INC. VILLA

CRAY Y-MP8
HARDWARE REFERENCE CARD

HTA-0867 Cray Research, Inc. Unpublished proprietary information - all rights reserved under the copyright laws of the United States and the U.C.C.

Hardware Training Department, 890 Industrial Blvd., Chippewa Falls, WI 54729

INSTRUCTIONS

INSTRUCTION	CAL	UNIT	DESCRIPTION
000000	ERR		Error Exit
††0010jk	CA,Aj Ak		Set the channel (Aj) CA register to (Ak) and begin I/O sequence
001000	PASS		Pass
††0011jk	CL,Aj Ak		Set the channel (Aj) CL register to (Ak)
††0012j0	CI,Aj		Clear channel (Aj) Interrupt and Error flags; clear device Master Clear (output channel)
††0012j1	MC,Aj		Clear channel (Aj) Interrupt and Error flags; set device Master Clear (output channel); clear device ready-held (input channel)
††0013j0	XA Aj		Transmit (Aj) to XA register
††0014j0	RT Sj		Transmit (Sj) to RTC register
††0014j1	SIPI Aj		Set Interprocessor Interrupt request to CPU (Aj)
001401	SIPI		Set interprocessor interrupt of CPU 0
††001402	CIPI		Clear interprocessor interrupt
††0014j3	CLN Aj		Transmit (Aj) to CLN register
††0014j4	PCI Sj		Enter Interrupt Interval (II) register with (Sj)
††001405	CCI		Clear Programmable Clock Interrupt (PCI) request
††001406	ECI		Enable Programmable Clock Interrupt (PCI) request
††001407	DCI		Disable Programmable Clock Interrupt (PCI) request
††0015j0	¶		Select performance monitor
††0015j1	¶¶		Set maintenance mode j
00200k	VL Ak		Transmit (Ak) to VL register
†002000	VL 1		Transmit 1 to VL register
002100	EFI		Enable interrupt on Floating-point error
002200	DFI		Disable interrupt on Floating-point error
002300	ERI		Enable Operand Range error interrupts

INSTRUCTION	CAL	UNIT	DESCRIPTION
002400	DRI		Disable Operand Range error interrupts
002500	DBM		Disable bi-directional memory transfers
002600	EBM		Enable bi-directional memory transfers
002700	CMR		Complete memory reference
0030j0	VM Sj		Transmit (Sj) to VM register
†003000	VM 0		Clear VM register
0034jk	SMjk 1,TS		Test and set Semaphore jk; $0 \le jk \le 37_8$
0036jk	SMjk 0		Clear Semaphore jk; $0 \le jk \le 37_8$
0037jk	SMjk 1		Set Semaphore jk; $0 \le jk \le 37_8$
004000	EX		Normal exit
0050jk	J Bjk		Ju mp to (Bjk)
006ijkm	j *exp*		Jump to *exp* = ijkm
007ijkm	R *exp*		Return jump to *exp* = ijkm; set B00 to (P) + 2
010ijkm	JAZ *exp*		Jump to *exp* = ijkm if(A0) = 0 (2^2 of i = 0)
011ijkm	JAN *exp*		Jump to *exp* = ijkm if (A0) ≠ 0 (2^2 of i = 0)
012ijkm	JAP *exp*		Jump to *exp* = ijkm if (A0) positive (2^2 of i = 0)
013ijkm	JAM *exp*		Jump to *exp* = ijkm if (A0) negative (2^2 of i = 0)
014ijkm	JSZ *exp*		Jump to *exp* = ijkm if (S0) = 0 (2^2 of i = 0)
015ijkm	JSN *exp*		Jump to *exp* = ijkm if (S0) ≠ 0 (2^2 of i = 0)
016ijkm	JSP *exp*		Jump to *exp* = ijkm if (S0) positive (2^2 of i = 0)
017ijkm	JSM *exp*		Jump to *exp* = ijkm if (S0) negative (2^2 of i = 0)
01hijkm	Ah *exp*		Transmit *exp* = ijkm to Ah (2^2 of i = 1)
†††, X 020ijkm	Ai *exp*		Transmit *exp* = jkm to Ai
†††, Y 020i00mn	Ai *exp*		Transmit *exp* = nm to Ai
†††021ijkm	Ai *exp*		Transmit ones complement of *exp* = jkm to Ai
†††, Y 021i00mn	Ai *exp*		Transmit ones complement of *exp* = nm to Ai
†††022ijk	Ai *exp*		Transmit *exp* = jk to Ai
023ij0	Ai Sj		Transmit (Sj) to Ai
023i01	Ai VL		Transmit (VL) to Ai
024ijk	Ai Bjk		Transmit (Bjk) to Ai
025ijk	Bjk Ai		Transmit (Ai) to Bjk
026ij0	Ai PSj	S Pop	Transmit population count of (Sj) to Ai
026ij1	Ai QSj	S Pop	Transmit population count parity of (Sj) to Ai
026ij7	Ai SBj		Transmit (SBj) to Ai
027ij0	Ai ZSj	S/LZ	Transmit Leading Zero Count of (Sj) to Ai
027ij7	SBj Ai		Transmit (Ai) to SBj
030ijk	Ai Aj + Ak	A Int Add	Integer sum of (Aj) and (Ak) to Ai

INSTRUCTION	CAL	UNIT	DESCRIPTION
†061ij0	Si Sj-S0	S Int Add	Integer difference of (Sj) less 2^{63} to Si
062ijk	Si Sj+FSk	Fp Add	Floating-point sum of (Sj) and (Sk) to Si
†062i0k	Si +FSk	Fp Add	Normalize (Sk) to Si
063ijk	Si Sj-FSk	Fp Add	Floating-point difference of (Sj) and (Sk) to Si
†063i0k	Si -FSk	Fp Add	Transmit normalized negative of (Sk) to Si
064ijk	Si Sj*FSk	Fp Mult	Floating-point product of (Sj) and (Sk) to Si
065ijk	Si Sj*HSk	Fp Mult	Half-precision rounded floating-point product of (Sj) and (Sk) to Si
066ijk	Si Sj*RSk	Fp Mult	Full-precision rounded floating-point product of (Sj) and (Sk) to Si
067ijk	Si Sj*ISk	Fp Mult	Two minus the floating-point product of (Sj) and (Sk) to Si
070ij0	Si /HSj	Fp Recp	Floating-point reciprocal approximation of (Sj) to Si
071i0k	Si Ak		Transmit (Ak) to Si with no sign extension
071i1k	Si +Ak		Transmit (Ak) to Si with sign extension
071i2k	Si +Fak		Transmit (Ak) to Si as unnormalized floating-point number (exponent = 40060)
071i30	Si 0.6		Transmit constant 0.75 x 2^{48} to Si (Si = 040060 140000 000000 000000)
071i40	Si 0.4		Transmit constant 0.5 to Si (Si = 040000 100000 000000 000000)
071i50	Si 1.		Transmit constant 1.0 to Si (Si = 040001 100000 000000 000000)
071i60	Si 2.		Transmit constant 2.0 to Si (Si = 040002 100000 000000 000000)
071i70	Si 4.		Transmit constant 4.0 to Si (Si = 040003 100000 000000 000000)
072i00	Si RT		Transmit (RTC) to Si
072i02	Si SM		Transmit (SM) to Si
072ij3	Si STj		Transmit (STj) to Si
073i00	Si VM		Transmit (VM) to Si
073i01	Si SRj		Transmit status register (SRj) bits to Si (j = 0)
073i11	¶		Read performance counter to Si
073021	¶		Increment performance counter (upper)
073031	¶		Clear all maintenance modes
073061	¶		Increment performance counter (lower)
073i02	SM Si		Transmit (Si) to SM
0731j3	STj Si		Transmit (Si) to STj
074ijk	Si Tjk		Transmit (Tjk) to Si
075ijk	Tjk Si		Transmit (Si) to Tjk
076ijk	Si Vj,Ak		Transmit (Vj, element (Ak)) to Si

INSTRUCTION	CAL	UNIT	DESCRIPTION
X 166ijk	Vi Sj*IVk	Fp Mult	Two minus the floating-point products of (Sj) and (Vk) to Vi
Y 166ijk	Vi Sj*Vk	Fp Mult	32-bit integer products of (Sj) and (Vk) to Vi
167ijk	Vi Vj*IVk	Fp Mult	Two minus the floating-point products of (Vj) and (Vk) to Vi
170ijk	Vi Sj + FVk	Fp Add	Floating-point sums of (Sj) and (Vk) to Vi
† 170i0k	Vi +FVk	Fp Add	Normalize (Vk) to Vi
171ijk	Vi Vj + FVk	Fp Add	Floating-point sums of (Vj) and (Vk) to Vi
172ijk	Vi Sj-FVk	Fp Add	Floating-point differences of (Sj) and (Vk) to Vi
† 172i0k	Vi -FVk	Fp Add	Transmit normalized negatives of (Vk) to Vi
173ijk	Vi Vj-FVk	Fp Add	Floating-point differences of (Vj) and (Vk) to Vi
174ij0	Vi /HVj	Fp Recip	Floating-point reciprocal approximations of (Vj) to Vi
174ij1	Vi PVj	V Pop	Population counts of (Vj) to Vi
174ij2	Vi QVj	V Pop	Population count parities of (Vj) to Vi
1750j0	VM Vj,Z	V Logical	VM = 1 if (Vj) = 0
1750j1	VM Vj,N	V Logical	VM = 1 if (Vj) ≠ 0
1750j2	VM Vj,P	V Logical	VM = 1 if (Vj) positive; 0 is positive
1750j3	VM Vj,M	V Logical	VM = 1 if (Vj) negative; 1 is negative
175ij4	Vi, VM Vj,Z	V Logical	VM bit = 1 if (Vj element) = 0 and element index is loaded into (compressed Vi)
175ij5	Vi, VM Vj, N	V Logical	VM bit = 1 if (Vj element) ≠ 0 and element index is loaded into (compressed Vi)
175ij6	Vi, VM Vj, P	V Logical	VM bit = 1 if (Vj element) ≥ 0 and element index is loaded into (compressed Vi)
175ij7	Vi, VM Vj, M	V Logical	VM bit = 1 if (Vj element) < 0 and element index is loaded into (compressed Vi)
176i0k	Vi, A0, Ak	Memory	Read (VL) words to Vi from memory address ((A0) + (DBA)) incremented by (Ak)
176i00	Vi, A0,1	Memory	Read (VL) words to Vi from memory address ((A0) + (DBA)) incremented by 1
176i1k	Vi, A0, Vk	Memory	Read (VL) words to Vi from memory address ((A0) + (Vk) + (DBA)) •*gather*
1770jk	,A0,Ak Vj	Memory	Write (VL) words from Vj to memory address ((A0) + (DBA)) incremented by (Ak)
1770j0	,A0, 1 Vj	Memory	Write (VL) words from Vj to memory address ((A0) + (DBA)) incremented by 1
1771jk	,A0, Vk Vj	Memory	Write (VL) words from Vj to memory address ((A0) + (Vk) + (DBA)) •*scatter*

†	-	Special Syntax mode
††	-	Privileged to monitor mode
†††	-	Generated depending on exp.
¶¶	-	Not supported by CAL version 2
X	-	X-mode instruction
Y	-	Y-mode instruction
()	-	Read as *the contents of ....*

Register	Value
Ah, h = 0	0
Ai, i = 0	(A0)
Aj, j = 0	0
Ak, k = 0	1
Si, i = 0	(S0)
Sj, j = 0	0
Sk, k = 0	2^{63}