CRAY RESEARCH, INC.

CRAY Y-MP8
HARDWARE REFERENCE CARD

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		Jump 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
†030i0k	Ai Ak	A Int Add	Transmit (Ak) to Ai
†030ij0	Ai Aj + 1	A Int Add	Transmit integer sum of (Aj) plus 1 to Ai
031ijk	Ai Aj-Ak	A Int Add	Integer difference of (Aj) less (Ak) to Ai
†031i00	Ai -1	A Int Add	Transmit -1 to Ai (Ai = 37777777777) in Y-mode
†031i0k	Ai -Ak	A Int Add	Transmit the negative of (Ak) to Ai
†031ij0	Ai Aj-1	A Int Add	Integer difference of (Aj) less 1 to Ai
032ijk	Ai Aj*Ak	A Int Mult	Integer product of (Aj) and (Ak) to Ai
033i00	Ai CI		Transmit lowest interrupting channel number to Ai (j = 0)
033ij0	Ai CA,Aj		Transmit address of channel (Aj) to Ai (j ≠ 0)
033ij1	Ai CE,Aj		Transmit Error flag of channel (Aj) to Ai (j ≠ 0)
034ijk	Bjk,Ai ,A0	Memory	Read (Ai) words to B registers starting at Bjk from memory address ((A0) + (DBA))
†034ijk	Bjk,Ai, 0,A0	Memory	Read (Ai) words to B registers starting at Bjk from memory address ((A0) + (DBA))
035ijk	,A0 Bjk,Ai	Memory	Write (Ai) words from B registers Bjk to memory address ((A0) + (DBA))
†035ijk	0,A0 Bjk,Ai	Memory	Write (Ai) words from B registers Bjk to memory address ((A0) + (DBA))
036ijk	Tjk,Ai ,A0	Memory	Read (Ai) words to T registers starting at Tjk from memory address ((A0) + (DBA))
†036ijk	Tjk,Ai 0,A0	Memory	Read (Ai) words to T registers starting at Tjk from memory address ((A0) + (DBA))
037ijk	,A0 Tjk,Ai	Memory	Write (Ai) words from T registers Tjk to memory address ((A0) + (DBA))
†037ijk	0,A0 Tjk,Ai	Memory	Write (Ai) words from T registers Tjk to memory address ((A0) + (DBA))
X 040ijkm	Si *exp*		Transmit *exp* = jkm to Si
Y 040i00mn	Si *exp*		Transmit *exp* = nm to Si
X 041ijkm	Si *exp*		Transmit ones complement of *exp* = jkm to Si
Y 041i00mn	Si *exp*		Transmit ones complement of *exp* = nm to Si
042ijk	Si < *exp*	S Logical	Form ones mask *exp* = 100_8 - jk bits in Si from the right
†042ijk	Si # > *exp*	S Logical	Form zeroes mask *exp* = jk bits in Si from the left
†042i77	Si 1	S Logical	Enter 1 into Si
†042i00	Si -1	S Logical	Enter -1 into Si (Si = 177777 177777 177777 177777)
043ijk	Si > *exp*	S Logical	Form ones mask *exp* = jk bits in Si from the left
†043ijk	Si # < *exp*	S Logical	Form zeroes mask *exp* = 100_8 - jk bits in Si from the right
†043i00	Si 0	S Logical	Clear Si
044ijk	Si Sj&Sk	S Logical	Logical product of (Sj) and (Sk) to Si

INSTRUCTION	CAL		UNIT	DESCRIPTION
†044ij0	Si	Sj&SB	S Logical	Sign bit of (Sj) to Si
†044ij0	Si	SB&Sj	S Logical	Sign bit of (Sj) to Si (j ≠ 0)
045ijk	Si	#Sk&Sj	S Logical	Logical product of (Sj) and ones complement of (Sk) to Si
†045ij0	Si	#SB&Sj	S Logical	Transmit (Sj) with sign bit cleared to Si
046ijk	Si	Sj\Sk	S Logical	Logical difference of (Sj) and (Sk) to Si
†046ij0	Si	Sj\SB	S Logical	Toggle sign bit of (Sj), enter into Si
†046ij0	Si	SB\Sj	S Logical	Toggle sign bit of (Sj), enter into Si (j ≠ 0)
047ijk	Si	#Sj\Sk	S Logical	Logical equivalence of (Sk) and (Sj) to Si
†047i0k	Si	#Sk	S Logical	Transmit ones complement of (Sk) to Si
†047ij0	Si	#Sj\SB	S Logical	Logical equivalence of (Sj) and sign bit to Si
†047ij0	Si	#SB\Sj	S Logical	Logical equivalence of (Sj) and sign bit to Si (j ≠ 0)
†047i00	Si	#SB	S Logical	Transmit ones complement of sign bit into Si
050ijk	Si	Sj\Si&Sk	S Logical	Logical product of ((Si) and (Sk) complement) ORed with logical product of ((Sj) and (Sk)) to Si •*scalar merge*
†050ij0	Si	Sj\Si&SB	S Logical	Scalar merge of (Si) and sign bit of (Sj) to Si
051ijk	Si	Sj\Sk	S Logical	Logical sum of (Sj) and (Sk) to Si
†051i0k	Si	Sk	S Logical	Transmit (Sk) to Si
†051ij0	Si	Sj\SB	S Logical	Logical sum of (Sj) and sign bit to Si
†051ij0	Si	SB\Sj	S Logical	Logical sum of (Sj) and sign bit to Si (j ≠ 0)
†051i00	Si	SB	S Logical	Transmit sign bit into Si
052ijk	S0	Si < *exp*	S Shift	Shift (Si) left *exp* = jk places to S0
053ijk	S0	Si > *exp*	S Shift	Shift (Si) right *exp* = 100_8 - jk places to S0
054ljk	Si	Si < *exp*	S Shift	Shift (Si) left *exp* = jk places to Si
055ijk	Si	Si > *exp*	S Shift	Shift (Si) right *exp* = 100_8 - jk places to Si
056ijk	Si	Si,Sj < Ak	S Shift	Shift (Si and Sj) left (Ak) places to Si
†056ij0	Si	Si,Sj < 1	S Shift	Shift (Si and Sj) left one place to Si
†056i0k	Si	Si < Ak	S Shift	Shift (Si) left (Ak) places to Si
057ijk	Si	Sj,Si > Ak	S Shift	Shift (Sj and Si) right (Ak) places to Si
†057ij0	Si	Sj,Si > 1	S Shift	Shift (Sj and Si) right one place to Si
†057i0k	Si	Si > Ak	S Shift	Shift (Si) right (Ak) places to Si
060ijk	Si	Sj + Sk	S Int Add	Integer sum of (Sj) and (Sk) to Si
†060i0k	Si	Sk	S Int Add	Transmit (Sk) to Si
†060ij0	Si	Sj + S0	S Int Add	Integer sum 2^{63} and (Sj) to Si
061ijk	Si	Sj-Sk	S Int Add	Integer difference of (Sj) less (Sk) to Si
†061i0k	Si	-Sk	S Int Add	Transmit negative of (Sk) to Si

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
077ijk	Vi,Ak Sj		Transmit (Sj) to Vi element (Ak)
†077i0k	Vi,Ak 0		Clear Vi element (Ak)
X 10hijkm	Ai *exp*,Ah	Memory	Read from memory address ((Ah) + jkm + (DBA)) to Ai (h ≠ 0)
Y 10hi00mn	Ai *exp*,Ah	Memory	Read from memory address ((Ah) + nm + (DBA)) to Ai (h ≠ 0)
†, X 100ijkm	Ai *exp*,0	Memory	Read from memory address (jkm + (DBA)) to Ai
Y 100i00mn	Ai *exp*,0	Memory	Read from memory address (nm + (DBA)) to Ai
†, X 100ijkm	Ai *exp*,	Memory	Read from memory address (jkm + (DBA)) to Ai
Y 100i00mn	Ai *exp*,	Memory	Read from memory address (nm + (DBA)) to Ai
†, X 10hi000	Ai ,Ah	Memory	Read from memory address ((Ah) + (DBA)) to Ai (h ≠ 0)
Y 10hi0000	Ai ,Ah	Memory	Read from memory address ((Ah) + (DBA)) to Ai (h ≠ 0)
X 11hijkm	*exp*,Ah Ai	Memory	Write (Ai) to memory address ((Ah) + jkm + (DBA)) (h ≠ 0)
Y 11hi00mn	*exp*,Ah Ai	Memory	Write (Ai) to memory address ((Ah) + nm + (DBA)) (h ≠ 0)
†, X 110ijkm	*exp*,0 Ai	Memory	Write (Ai) to memory address (jkm + (DBA))
Y 110i00mn	*exp*,0 Ai	Memory	Write (Ai) to memory address (nm + (DBA))
†, X 110ijkm	exp, Ai	Memory	Write (Ai) to memory address (jkm + (DBA))
Y 110i00mn	*exp*, Ai	Memory	Write (Ai) to memory address (nm + (DBA))
†, X 11hi000	,Ah Ai	Memory	Write (Ai) to memory address ((Ah) + (DBA)) (h ≠ 0)
Y 11hi0000	,Ah Ai	Memory	Write (Ai) to memory address ((Ah) + (DBA)) (h ≠ 0)
X 12hijkm	Si *exp*,Ah	Memory	Read from memory address ((Ah) + jkm + (DBA)) to Si (h ≠ 0)
Y 12hi00mn	Si *exp*,Ah	Memory	Read from memory address ((Ah) + nm + (DBA)) to Si (h ≠ 0)
X 120ijkm	Si *exp*,0	Memory	Read from memory address (jkm + (DBA)) to Si
Y 120i00mn	Si *exp*,0	Memory	Read from memory address (nm + (DBA)) to Si
†, X 120ijkm	Si *exp*,	Memory	Read from memory address (jkm + (DBA)) to Si
Y 120i00mn	Si *exp*,	Memory	Read from memory address (nm + (DBA)) to Si
†, X 12hi000	Si ,Ah	Memory	Read from memory address ((Ah) + (DBA)) to Si (h ≠ 0)
Y 12hi0000	Si ,Ah	Memory	Read from memory address ((Ah) + (DBA)) to Si (h ≠ 0)
X 13hijkm	*exp*,Ah Si	Memory	Write (Si) to memory address ((Ah) + jkm + (DBA)) (h ≠ 0)
Y 13hi00mn	*exp*,Ah Si	Memory	Write (Si) to memory address ((Ah) + nm + (DBA)) (h ≠ 0)
†, X 130ijkm	*exp*,0 Si	Memory	Write (Si) to memory address (jkm + (DBA))
Y 130i00mn	*exp*,0 Si	Memory	Write (Si) to memory address (nm + (DBA))
X 130ijkm	*exp*, Si	Memory	Write (Si) to memory address (jkm + (DBA))

INSTRUCTION	CAL	UNIT	DESCRIPTION
Y 130i00mn	exp, Si	Memory	Write (Si) to memory address (nm + (DBA))
X 13hi000	,Ah Si	Memory	Write (Si) to memory address ((Ah) + (DBA)) (h ≠ 0)
Y 13hi0000	,Ah Si	Memory	Write (Si) to memory address ((Ah) + (DBA)) (h ≠ 0)
140ijk	Vi Sj&Vk	V Logical	Logical products of (Sj) and (Vk) to Vi
141ijk	Vi Vj&Vk	V Logical	Logical products of (Vj) and (Vk) to Vi
142ijk	Vi Sj|Vk	V Logical	Logical sums of (Sj) and (Vk) to Vi
† 142i0k	Vi Vk	V Logical	Transmit (Vk) to Vi
143ijk	Vi Vj|Vk	V Logical	Logical sums of (Vj) and (Vk) to Vi
144ijk	Vi Sj\Vk	V Logical	Logical differences of (Sj) and (Vk) to Vi
145ijk	Vi Vj\Vk	V Logical	Logical differences of (Vj) and (Vk) to Vi
† 145iii	Vi 0	V Logical	Clear Vi
146ijk	Vi Sj|Vk&VM	V Logical	Transmit (Sj) if VM bit = 1; (Vk) if VM bit = 0 to Vi *•scalar•vector merge*
† 146i0k	Vi #VM&Vk	V Logical	Vector merge of (Vk) and 0 to Vi
147ijk	Vi Vj|Vk&VM	V Logical	Transmit (Vj) if VM bit = 1; (Vk) if VM bit = 0 to Vi *•vector•vector merge*
150ijk	Vi Vj < Ak	V Shift	Shift (Vj) left (Ak) places to Vi
150ij0	Vi Vj < 1	V Shift	Shift (Vj) left one place to Vi
151ijk	Vi Vj > Ak	V Shift	Shift (Vj) right (Ak) places to Vi
† 151ij0	Vi Vj > 1	V Shift	Shift (Vj) right one place to Vi
152ijk	Vi Vj,Vj < Ak	V Shift	Double shift (Vj) left (Ak) places to Vi
† 152ij0	Vi Vj,Vj < 1	V Shift	Double shift (Vj) left one place to Vi
153ijk	Vi Vj,Vj > Ak	V Shift	Double shift (Vj) right (Ak) places to Vi
† 153ij0	Vi Vj,Vj > 1	V Shift	Double shift (Vj) right one place to Vi
† 154ijk	Vi Sj + Vk	V Int Add	Integer sums of (Sj) and (Vk) to Vi
155ijk	Vi Vj + Vk	V Int Add	Integer sums of (Vj) and (Vk) to Vi
156ijk	Vi Sj-Vk	V Int Add	Integer differences of (Sj) and (Vk) to Vi
† 156i0k	Vi -Vk	V Int Add	Transmit negative of (Vk) to Vi
157ijk	Vi Vj-Vk	V Int Add	Integer differences of (Vj) and (Vk) to Vi
160ijk	Vi Sj*FVk	Fp Mult	Floating-point products of (Sj) and (Vk) to Vi
161ijk	Vi Vj*FVk	Fp Mult	Floating-point products of (Vj) and (Vk) to Vi
162ijk	Vi Sj*HVk	Fp Mult	Half-precision rounded floating-point products of (Sj) and (Vk) to Vi
163ijk	Vi Vj*HVk	Fp Mult	Half-precision rounded floating-point products of (Vj) and (Vk) to Vi
164ijk	Vi Sj*RVk	Fp Mult	Rounded floating-point products of (Sj) and (Vk) to Vi
165ijk	Vi Vj*RVk	Fp Mult	Rounded floating-point products of (Vj) and (Vk) to Vi

RUCTION	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 ....*

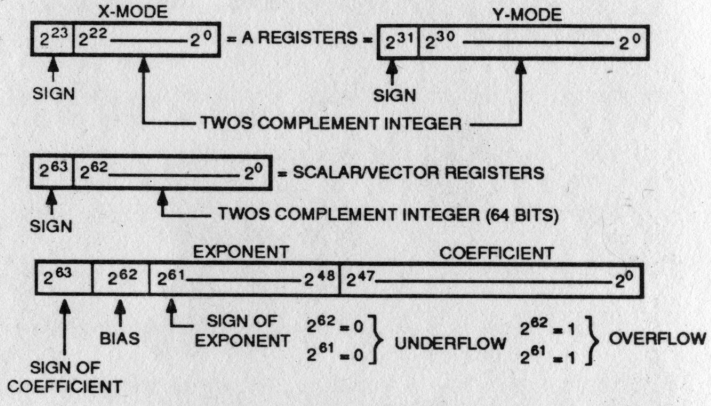
Register	Value
Ah, h = 0	0
Ai, i = 0	(A0)
Aj, j = 0	0
Ak, k = 0	1
Si, i = 0	(S0)
Sj, j = 0	0
Sk, k = 0	2^{63}

FUNCTIONAL UNITS

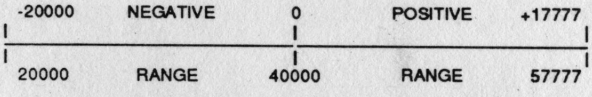
FUNCTIONAL UNIT	UNIT TIME (CLOCK PERIODS)	INSTRUCTIONS
Address Integer Add	2	030, 031
Address Integer Multiply	4	032
Scalar Integer Add	4	060, 061
Scalar Logical	1	042 - 051
Scalar Shift Single	3	052 - 055
Scalar Shift Double	4	056, 057
Scalar Pop./Parity	4	026
Leading Zero	3	027
Vector Integer Add	4	154 - 157
Full Vector Logical	1	140 - 147, 175
Second Vector Logical	1	140 - 145
Vector Shift	2	150, 151, 153
Vector Shift Double Left	3	152
Vector Pop./Parity	3	174ij1, 174ij2
Floating-point Add	6	062, 063, 170 - 173
Floating-point Multiply	7	064 - 067, 160 - 167
Floating-point Reciprocal	14	070, 174ij0
Memory (Scalar)	17	120 - 127
Memory (Scalar)	18	10h

READY TO USE
Vj/k = VL + 3
FU = VL + 4
Vi = VL + FU + 6

DATA FORMATS

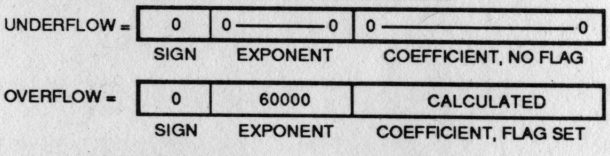


EXPONENT RANGE

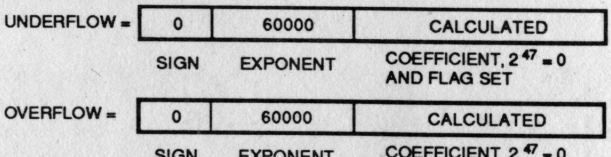


FLOATING-POINT RANGE ERRORS

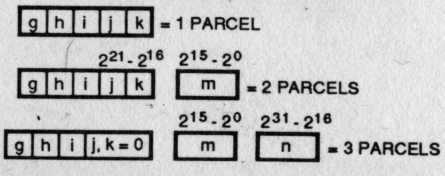
FLOATING-POINT ADD OR FLOATING-POINT MULTIPLY



FLOATING-POINT RECIPROCAL



INSTRUCTION FORMATS



A-7440B

CRAY PROPRIETARY

SECTION 0	1	BITS 0 - 8	CPU 0 CPU 2	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						BK BUSY (2^0) MEMORY
	2	BITS 9 - 17	CPU 1 CPU 3	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						BK BUSY (2^1) MEMORY
	3	BITS 18 - 26	CPU 2 CPU 0	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				I/O M. C. SECTION 0
	4	BITS 27 - 35	CPU 3 CPU 1	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				3 CP W. E.
	5	BITS 36 - 44	CPU 7 CPU 5	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				
	6	BITS 45 - 53	CPU 6 CPU 4	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				
	7	BITS 54 - 62	CPU 5 CPU 7	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						MCU INTER. ALL CPUs
	8	BITS 63 - 71	CPU 4 CPU 6	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						I/O M. C. SECTION 0, 1
SECTION 1	9	BITS 0 - 8	CPU 1 CPU 3	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						BK BUSY (2^0) MEMORY
	10	BITS 9 - 17	CPU 2 CPU 0	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2) GO WRITE						BK BUSY (2^1) MEMORY
	11	BITS 18 - 26	CPU 3 CPU 1	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				I/O M. C. SECTION 1
	12	BITS 27 - 35	CPU 0 CPU 2	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				3 CP W.E.
	13	BITS 36 - 44	CPU 4 CPU 6	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				CPU M. C. CPU 0 - 7
	14	BITS 45 - 53	CPU 7 CPU 5	A0 - 16	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				I/O M. C. CPU 0 - 7
	15	BITS 54 - 62	CPU 6 CPU 4	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						FORCED 1s TO CPU
	16	BITS 63 - 71	CPU 5 CPU 7	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						FORCED 1s TO CPU
CPUs	17	CPU 0	CONFLICT RESOLUTION (YL) SECTION 0 SS: 0 - 3 0, 4, 10, 14			SHARED (JR) 0 8 16 24 32 40 48 56		LOSP 20/21 (DC)	VHISP 1 (DE)				PRIORITY (HH) LOSP 24 - 27 ERROR LOGGER BIT (2^0)
	18	CPU 1	SECTION 1 SS: 0 - 3 1, 5, 11, 15			1 9 17 25 33 41 49 57		LOSP 22/23	VHISP 1				PRIORITY (HH) LOSP 20 - 23 ERROR LOGGER BIT (2^1)
	19	CPU 2	SECTION 2 SS: 0 - 3 2, 6, 12, 16			2 10 18 26 34 42 50 58		LOSP 24/25	VHISP 5				CPU 0 - 3 (HH) D.L., W.S.
	20	CPU 3	SECTION 3 SS: 0 - 3 3, 7, 13, 17			3 11 19 27 35 43 51 59		LOSP 26/27	VHISP 5				CPU 4 - 7 (HH) D.L., W.S.
	21	CPU 4	SECTION 0 SS: 4 - 7 20, 24, 30, 34			4 12 20 28 36 44 52 60		LOSP 30/31	VHISP 11				LOWEST INTERRUPT CHANNEL NUMBER (HH)
	22	CPU 5	SECTION 1 SS: 4 - 7 21, 25, 31, 35			5 13 21 29 37 45 53 61		LOSP 32/33	VHISP 11				ERROR LOGGER CONTROL (HH)
	23	CPU 6	SECTION 2 SS: 4 - 7 22, 26, 32, 36			6 14 22 30 38 46 54 62		LOSP 34/35	VHISP 15				PRIORITY (HH) LOSP 30 - 33 ERROR LOGGER BIT (2^2)
	24	CPU 7	SECTION 3 SS: 4 - 7 23, 27, 33, 37			7 15 23 31 39 47 55 63		LOSP 36/37	VHISP 15				PRIORITY (HH) LOSP 34 - 37 ERROR LOGGER BIT (2^3)
SECTION 2	25	BITS 0 - 8	CPU 2 CPU 0	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						BK BUSY (2^0) MEMORY
	26	BITS 9 - 17	CPU 3 CPU 1	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						BK BUSY (2^1) MEMORY
	27	BITS 18 - 26	CPU 0 CPU 2	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				I/O M. C. SECTION 2
	28	BITS 27 - 35	CPU 1 CPU 3	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				3 CP W.E.
	29	BITS 36 - 44	CPU 5 CPU 7	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				SELECT DELAY TO CPUs
	30	BITS 45 - 53	CPU 4 CPU 6	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				SELECT DELAY TO CPUs
	31	BITS 54 - 62	CPU 7 CPU 5	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						I/O M.C. SECTION 2, 3
	32	BITS 63 - 71	CPU 6 CPU 4	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						SS BUSY SEL. TO CPUs
SECTION 3	33	BITS 0 - 8	CPU 3 CPU 1	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						BK BUSY (2^0) MEMORY
	34	BITS 9 - 17	CPU 0 CPU 2	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						BK BUSY (2^1) MEMORY
	35	BITS 18 - 26	CPU 1 CPU 3	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				I/O M. C. SECTION 3
	36	BITS 27 - 35	CPU 2 CPU 0	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				3 CP W.E.
	37	BITS 36 - 44	CPU 6 CPU 4	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				SS BUSY TO CPUs
	38	BITS 45 - 53	CPU 5 CPU 7	A7 - 12	BK BITS 0 - 2 BK BITS 0 - 1	GOSS 0 - 7 ABORT	SS RDO. (0 - 2) GO WRITE	ABORT	GO WRITE				BK BUSY (2^0) TO CPUs
	39	BITS 54 - 62	CPU 4 CPU 6	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						BK BUSY (2^1) TO CPUs
	40	BITS 63 - 71	CPU 7 CPU 5	A0 - 16 A0 - 8	BK BIT 2 A13 - 16	GOSS 0 - 7	SS RDO. (0 - 2)						4 CP BK BUSY SEL. CPUs
	41	CLOCK	(ALL CPUs LISTED ARE PHYSICAL)										

A-7437A

CRAY Y-MP8 CHASSIS LOCATIONS

CPU TO MEMORY PRIORITY

SEC.		0		1		2		3	
SS		0 1 4 5	2 3 6 7	0 1 4 5	2 3 6 7	0 1 4 5	2 3 6 7	0 1 4 5	2 3 6 7
CPU PATH	0	A	H	D	E	C	F	B	G
	1	B	G	A	H	D	E	C	F
	2	C	F	B	G	A	H	D	E
	3	D	E	C	F	B	G	A	H
	4	H	A	E	D	F	C	G	B
	5	G	B	H	A	E	D	F	C
	6	F	C	G	B	H	A	E	D
	7	E	D	F	C	G	B	H	A

CN 0 CPU A
CN 1 CPU B
CN 2 CPU C
CN 3 CPU D
CN 4 CPU E
CN 5 CPU F
CN 6 CPU G
CN 7 CPU H

OUTBOARD FANOUT CN NAME - WIRENAME

MAINTENANCE PANEL SELECTION

SWITCH SETTINGS
CPU 0 MASTER

	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	0	3	2	5	4	7	6
2	2	3	0	1	6	7	4	5
3	3	2	1	0	7	6	5	4
4	4	5	6	7	0	1	2	3
5	5	4	7	6	1	0	3	2
6	6	7	4	5	2	3	0	1
7	7	6	5	4	3	2	1	0

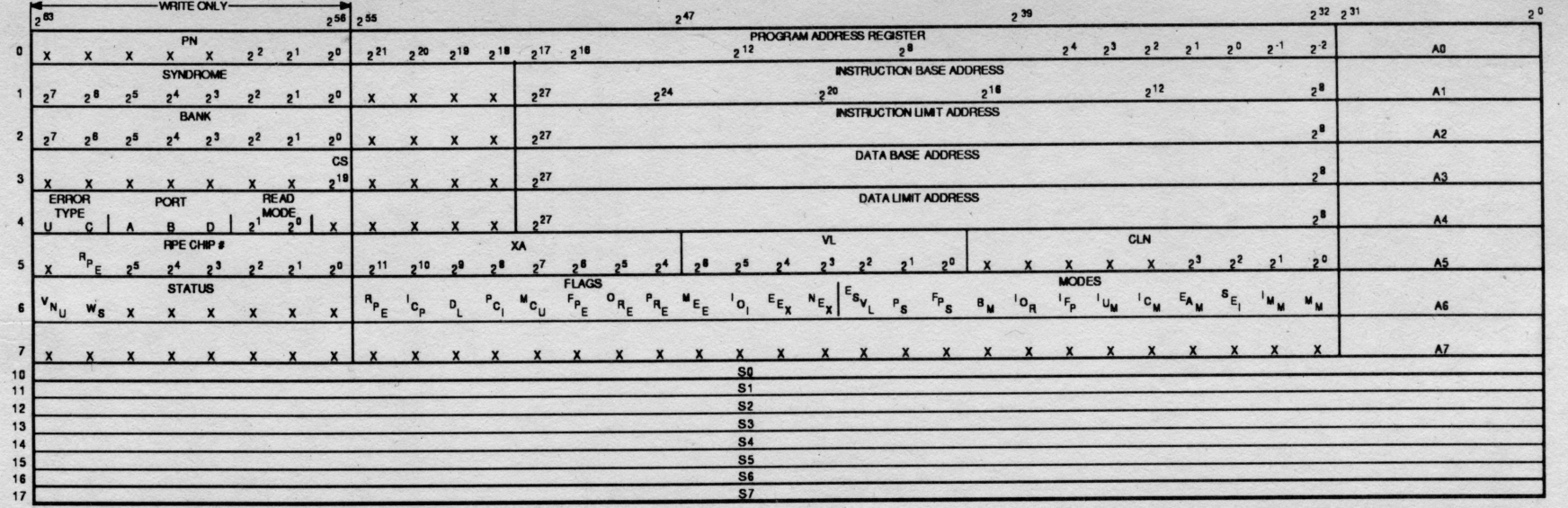
PHYSICAL CPU

LOGICAL OPERATORS	
&	0101
AND	1100
	0100
I	0101
OR	1100
	1101
\	0101
XOR	1100
	1001

MEMORY MODULE ASSOCIATION CRAY Y-MP/8

	SEC. 0	SEC. 1	SEC. 2	SEC. 3	DATA POSITIONS								
					N+0	N+1	N+2	N+3	N+4	N+5	N+6	N+7	N+8
MEM. 0	1,	9,	25,	33	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8
MEM. 1	2,	10,	26,	34	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}	2^{16}	2^{17}
MEM. 2	3,	11,	27,	35	2^{18}	2^{19}	2^{20}	2^{21}	2^{22}	2^{23}	2^{24}	2^{25}	2^{26}
MEM. 3	4,	12,	28,	36	2^{27}	2^{28}	2^{29}	2^{30}	2^{31}	2^{32}	2^{33}	2^{34}	2^{35}
MEM. 4	5,	13,	29,	37	2^{36}	2^{37}	2^{38}	2^{39}	2^{40}	2^{41}	2^{42}	2^{43}	2^{44}
MEM. 5	6,	14,	30,	38	2^{45}	2^{46}	2^{47}	2^{48}	2^{49}	2^{50}	2^{51}	2^{52}	2^{53}
MEM. 6	7,	15,	31,	39	2^{54}	2^{55}	2^{56}	2^{57}	2^{58}	2^{59}	2^{60}	2^{61}	2^{62}
MEM. 7	8,	16,	32,	40	2^{63}	2^{64}	2^{65}	2^{66}	2^{67}	2^{68}	2^{69}	2^{70}	2^{71}
						CB 0	CB 1	CB 2	CB 3	CB 4	CB 5	CB 6	CB 7
MEMORY MODULE LOCATION													

A-7439A



CRAY PROPRIETARY

RPE CHIP #

CHIP FUNCTION	$2^5 2^4 2^3$	$2^2 2^1 2^0$ CHIP NUMBER						
NO PARITY ERROR	0 0 0	N/A						
V REG. A BOARD	0 0 1	0 = (VS0)	1 = (VS10)	2 = (VS20)	3 = (VS30)	4 = (VS40)	5 = (VS50)	6 = (VS60)	7 = (VS70)
V REG. B BOARD	0 1 0	0 = (VS1)	1 = (VS11)	2 = (VS21)	3 = (VS31)	4 = (VS41)	5 = (VS51)	6 = (VS61)	7 = (VS71)
V REG. C BOARD	0 1 1	0 = (VS2)	1 = (VS12)	2 = (VS22)	3 = (VS32)	4 = (VS42)	5 = (VS52)	6 = (VS62)	7 = (VS72)
V REG. D BOARD	1 0 0	0 = (VS3)	1 = (VS13)	2 = (VS23)	3 = (VS33)	4 = (VS43)	5 = (VS53)	6 = (VS63)	7 = (VS73)
IB REG.	1 0 1	0 = (HS0)	1 = (HS1)	2 = (HS2)	3 = (HS3)	4 = (HS4)	5 = (HS5)	6 = (HS6)	7 = (HS7)
T/B SR REG.	1 1 0	0 = (HS8)	1 = (HS9)	2 = (HS10)	3 = (HS11)	4 = (HS12)	5 = (HS13)	6 = (HS14)	7 = (HS15)

READ MODE

$2^1 2^0$	PORT A	PORT B	PORT D
0 0	EX		FETCH
0 1	B	T	LOSP
1 0	VECTOR	VECTOR	HSP
1 1	A/S		SSD

MEMORY ADDRESS - CRAY Y-MP 8/32

$2^{24} 2^{23} 2^{22} 2^{21} 2^{20}$	2^{19}	$2^{18} 2^{17} 2^{16} 2^{15} 2^{14} 2^{13} 2^{12} 2^{11} 2^{10} 2^9 2^8$	$2^7 2^6 2^5$	$2^4 2^3 2^2$	$2^1 2^0$
X X X X X	X	X X X X X X X X X X X	X X X	X X X	X X
$2^{15} 2^{14} 2^{13} 2^{12} 2^{11}$	2^0 C.S.	$2^{10} 2^9 2^8 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0$	$2^2 2^1 2^0$ BANK SELECT	0-7 GOSS	0-3 SEC.

A-7438

CRAY Y-MP EXCHANGE PACKAGE