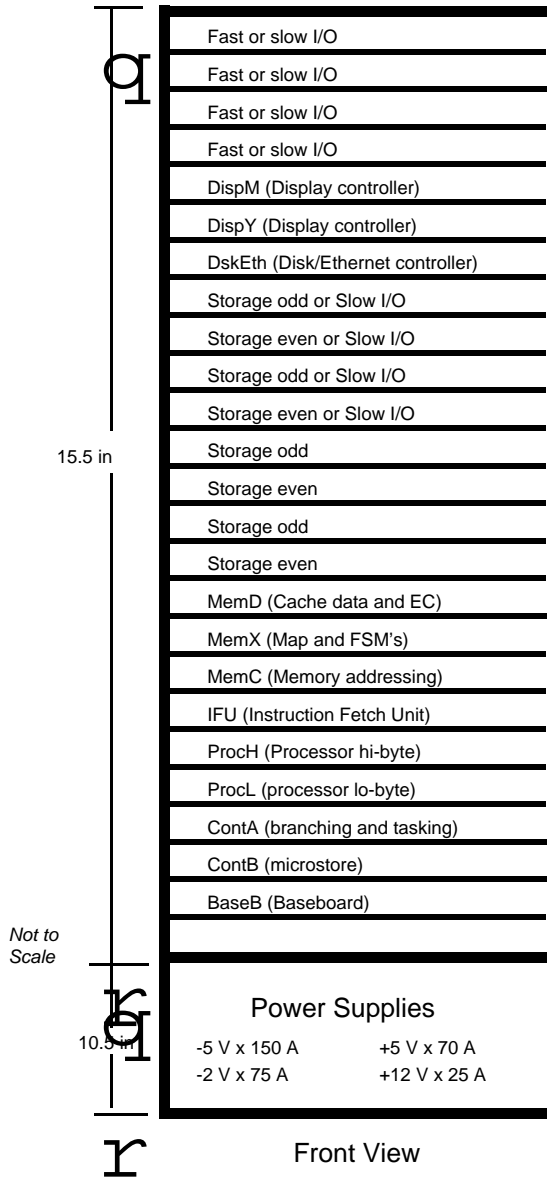


\* Task-Specific  
 {xxx} Source of Control

**Figure 1**  
**Dorado**  
**Programmer's View**



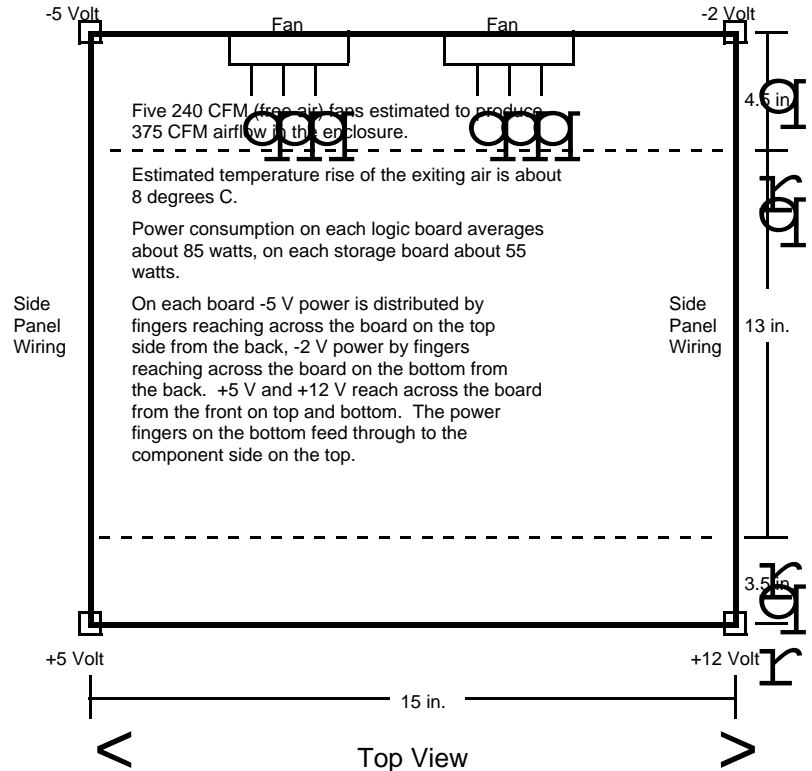
The +5 V supply and one fan are controlled by a switch; -5 V, +12 V, and -2 V supplies, the other four fans, and the disk logic and spindle power are controlled by the baseboard microcomputer (or the controlling Alto).

BaseB and ContB boards are equipped with temperature sensors that are repetitively monitored by the baseboard microcomputer; most other boards have temperature sensors that can be monitored when the microprocessor is halted. In the event some temperature exceeds 60 degrees C, the microcomputer will shut down the three power supplies that it controls.

The microcomputer also monitors power supplies; when any voltage or current deviates from its allowed range, the microcomputer shuts off power to the three supplies that it controls.

The card cage shown here is beneath the Trident T80 disk, and both are inside an enclosure designed to reduce the noise level for an office environment. The total enclosure size is about 4 feet high x 4 feet deep x 2 feet wide (ugh).

The machine weighs between 500 and 600 lbs.



The 11 logic boards in production models will be multiwire; 2 to 8 storage boards and the two backpanels are printed circuits.

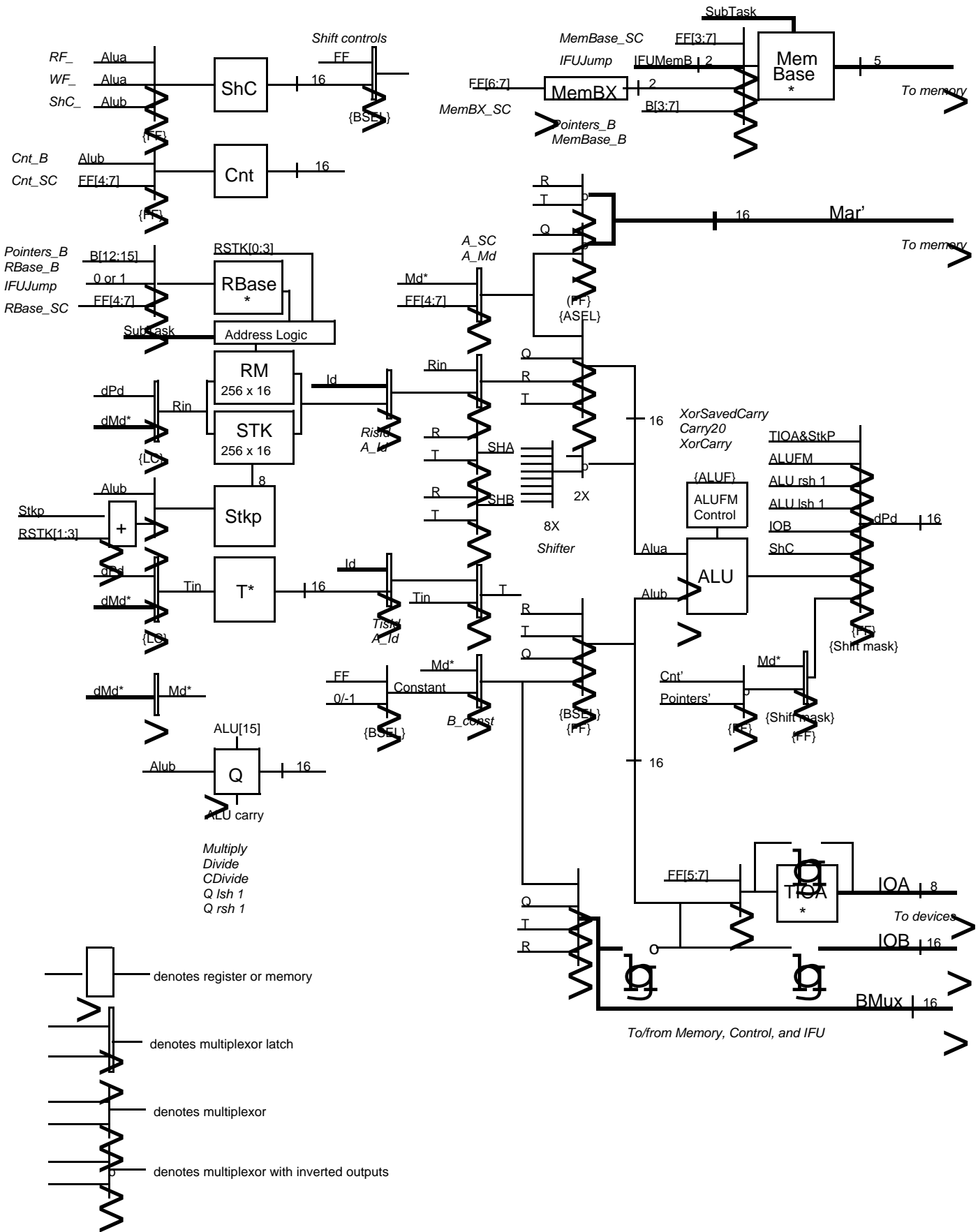
The following shows approximate component count:

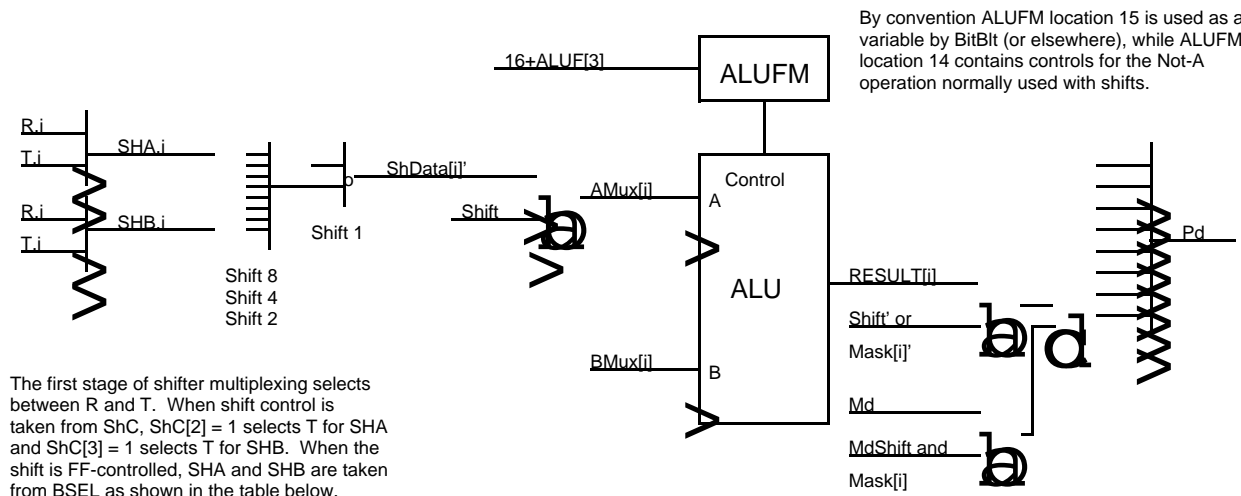
11 Logic boards:		8 Storage boards:	
2315	ic's of random logic	1056	ic's of random logic
246	1kx1 ECL RAM's	1152	16Kx1 MOS RAM's
71	16x4 ECL RAM's		
24	256x4 ECL RAM's		
21	16Kx1 MOS RAM's		
1600	SIP's		

Each board can mount 24 x 12 or 288 16-pin DIPs. Normal MECL-10000 DIP's are connected to the ground plane and -5V supply. Logic nets are terminated through 100-ohm resistors at one or both ends to the -2V supply. The resistors are in low-profile SIPs that mount between the DIPs (144 8-pin SIPs per board).

The +5 V supply is used for TTL/ECL conversions and for TTL components. The MOS IC's on the memory storage boards and in the Map use the +12 V supply.

**Figure 2  
Card Cage**





By convention ALUFM location 15 is used as a variable by BitBit (or elsewhere), while ALUFM location 14 contains controls for the Not-A operation normally used with shifts.

The first stage of shifter multiplexing selects between R and T. When shift control is taken from ShC, ShC[2] = 1 selects T for SHA and ShC[3] = 1 selects T for SHB. When the shift is FF-controlled, SHA and SHB are taken from BSEL as shown in the table below.

The 32-bit quantity SHA..SHB is then left-shifted through an 8-in multiplexor controlled by the shift 8, shift 4, and shift 2 controls.

The final stage is an inverting 2-in multiplexor which is disabled when no shift is taking place.

Mask[i] = LMask[i] or RMask[i]  
ALUF[0:2] controls masking

The hardware actually uses two inputs of the Pd multiplexor when shifting. One of these is the normal ALU path, and the other is either Md (on a replace-with-Md shift) or 0. The multiplexor select is changed to the Md/0 path when the bit is being masked out.

### Shift Data Paths

Field	SHA	SHB	Shift Count	RMask	LMask
ShC bits:	2	3	4:7	8:11	12:15
RF_A	A[2]	A[3]	P+S+1	undefined	15-S
WF_A	A[2]	A[3]	16-P-S-1	16-P-S-1	P
ShC_B	B[2]	B[3]	B[4:7]	B[8:11]	B[12:15]
BSEL.0=1	BSEL.1	BSEL.2	FF[4:7]	FF[4:7]	FF[0:3]

Functions that load ShC

Shift controls come from ShC except when BSEL.0 is 1 in the microinstruction that shifts

Shift controls come from FF when BSEL.0 is 1, and the source for B is changed to Q.

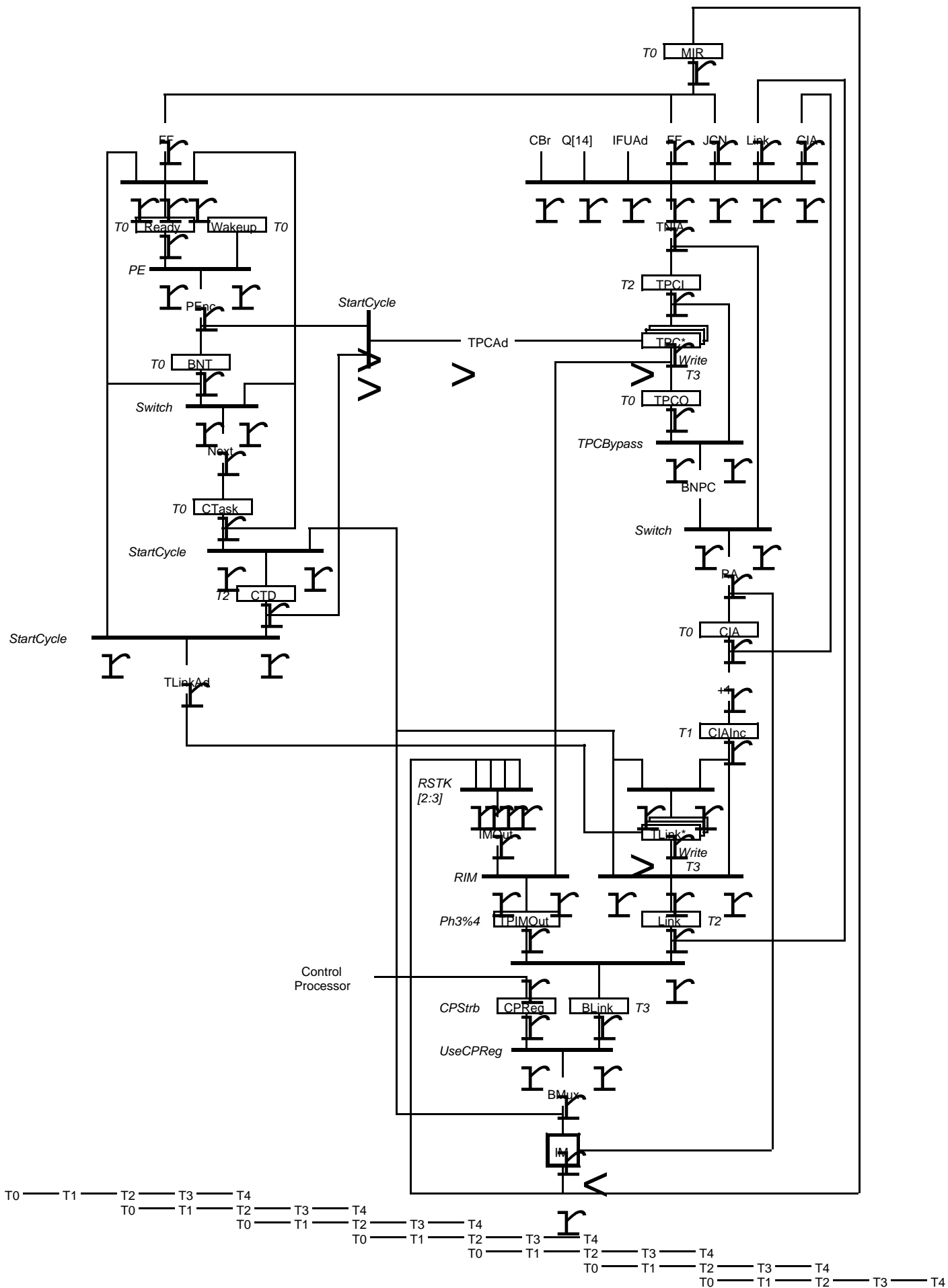
P = A[8:11] = number of bits to the left of the field  
S = A[12:15] = number of bits in the field - 1

The values for RMask, LMask, and Shift Count are and'ed by 17-octal. The 32-bit quantity SHA[0:15]..SHB[0:15] are left-cycled by the shift count and the right-most 16 bits are the shift data.

RF\_ and WF\_ are intended for use with "reasonable" values of P and S.

### Derivation of Shift Controls

Figure 4  
Shifter

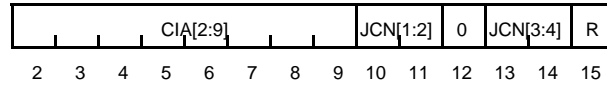
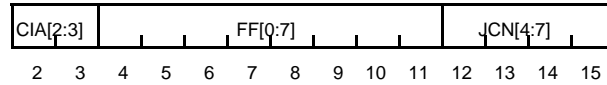
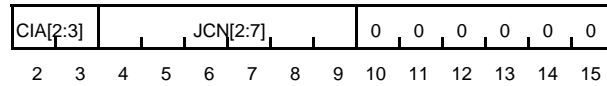
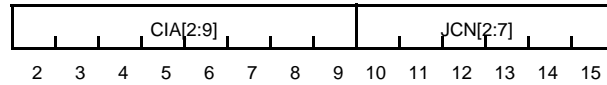


**Figure 5**  
**Control Section**

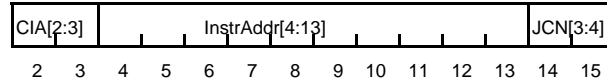
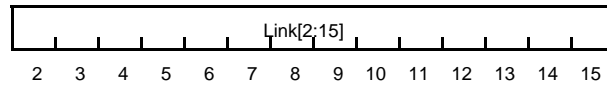
JCN  
 0 1 2 3 4 5 6 7

1 0	ADDRESS BITS		Local Jump/Call
1 1	ADDRESS BITS		Global Call
0 0 0 0	ADDRESS BITS		Long Jump/Call
0	ADDRESS BITS #000x	BRANCH CONDITION #111	Conditional Jump/Call
0 1	RETURN FUNCTION	1 1 1	Return
0 0 1	NEXT NUMBER	1 1 1	IFU Jump
0 0 0 1	x	1 1 1	undefined

TNIA:



R is result



A long, local, or conditional branch is a call iff, before any modification of TNIA by branch conditions or dispatches, TNIA[12:15] is 0; otherwise, it is a jump.

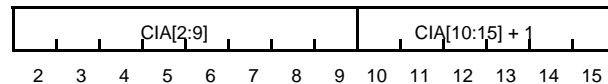
Conditional Branch

JCN[5:7]	-or-	FF	BRANCH CONDITION
0		60	ALU = 0
1		61	ALU < 0
2		62	Carry'
3		63	Cnt=0&-1 (decrement Cnt after testing)
4		64	R < 0
5		65	R odd
6		66	IOAtten' (non-emulator) -or- Reschedule (emulator)
--		67	Overflow

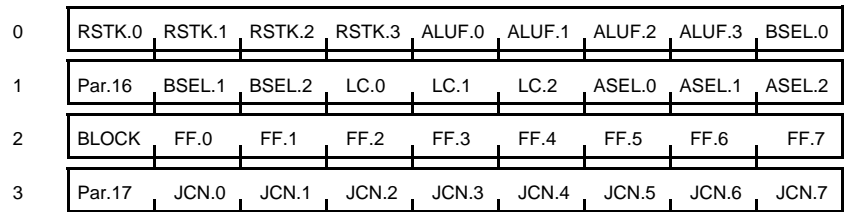
Return

JCN[2:4]	FUNCTION
0	Subroutine Return
1	unused
2	unused
3	unused
4	Read TPC
5	WriteTPC
6	Read Instruction Memory
	Address is in Link.
	Data appears on B[7:15] when B_Link executed in following microinstruction.
7	Write Instruction Memory
	Address is in Link. RSTK.3 is 1 to write the left half of IM, 0 to write the right half.

Loaded into Link by Call, Return, or IFUJump



RSTK[2:3]



Good (odd) parity is written if RSTK.1 is 0, else bad (even) parity is written.

The most significant bit of data is RSTK.2 and the least significant 16 bits are B[0:15].

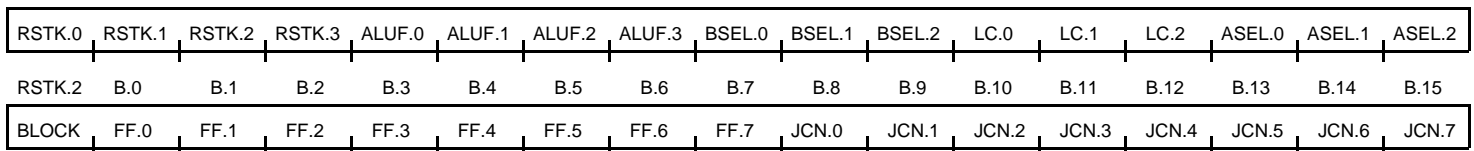
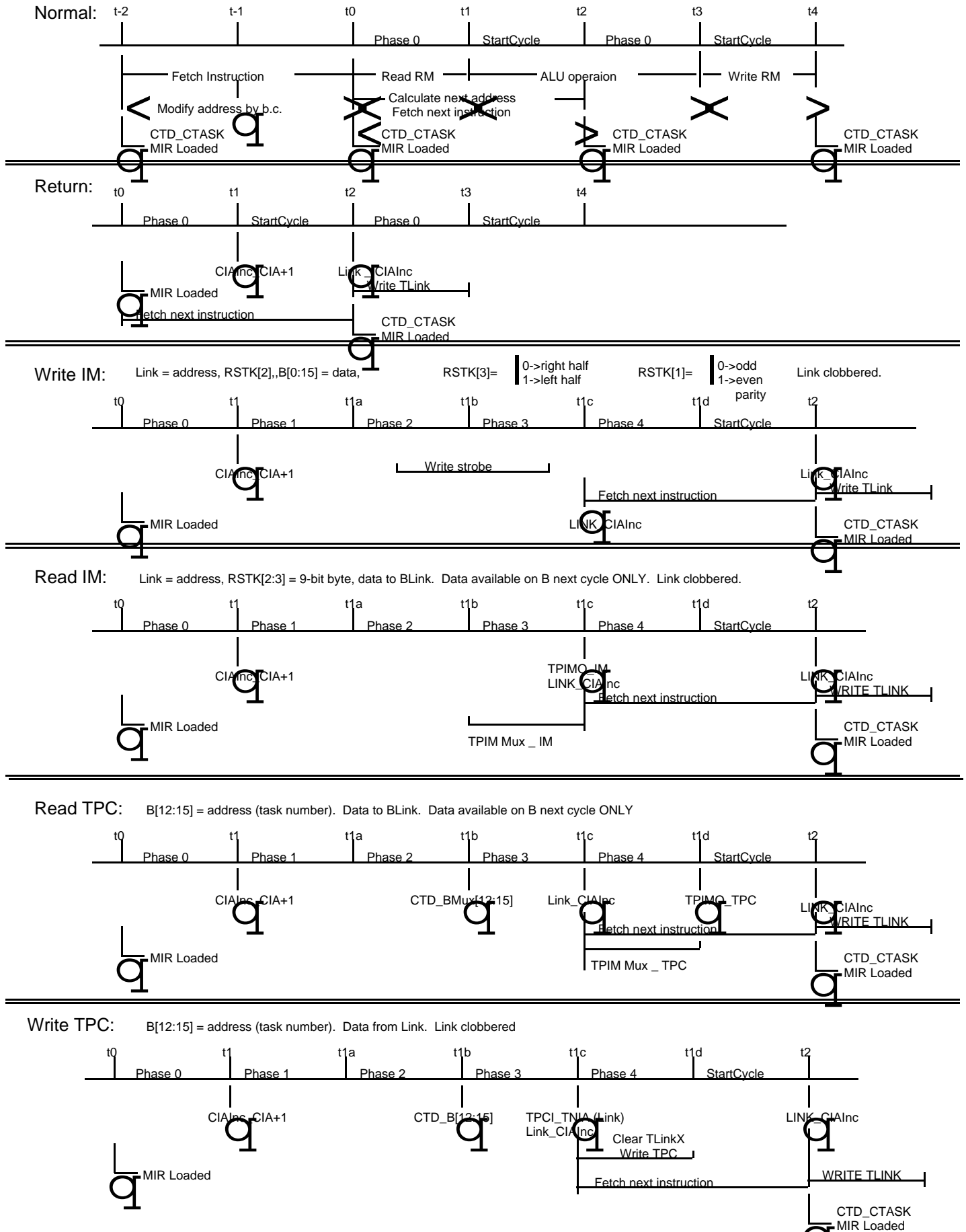
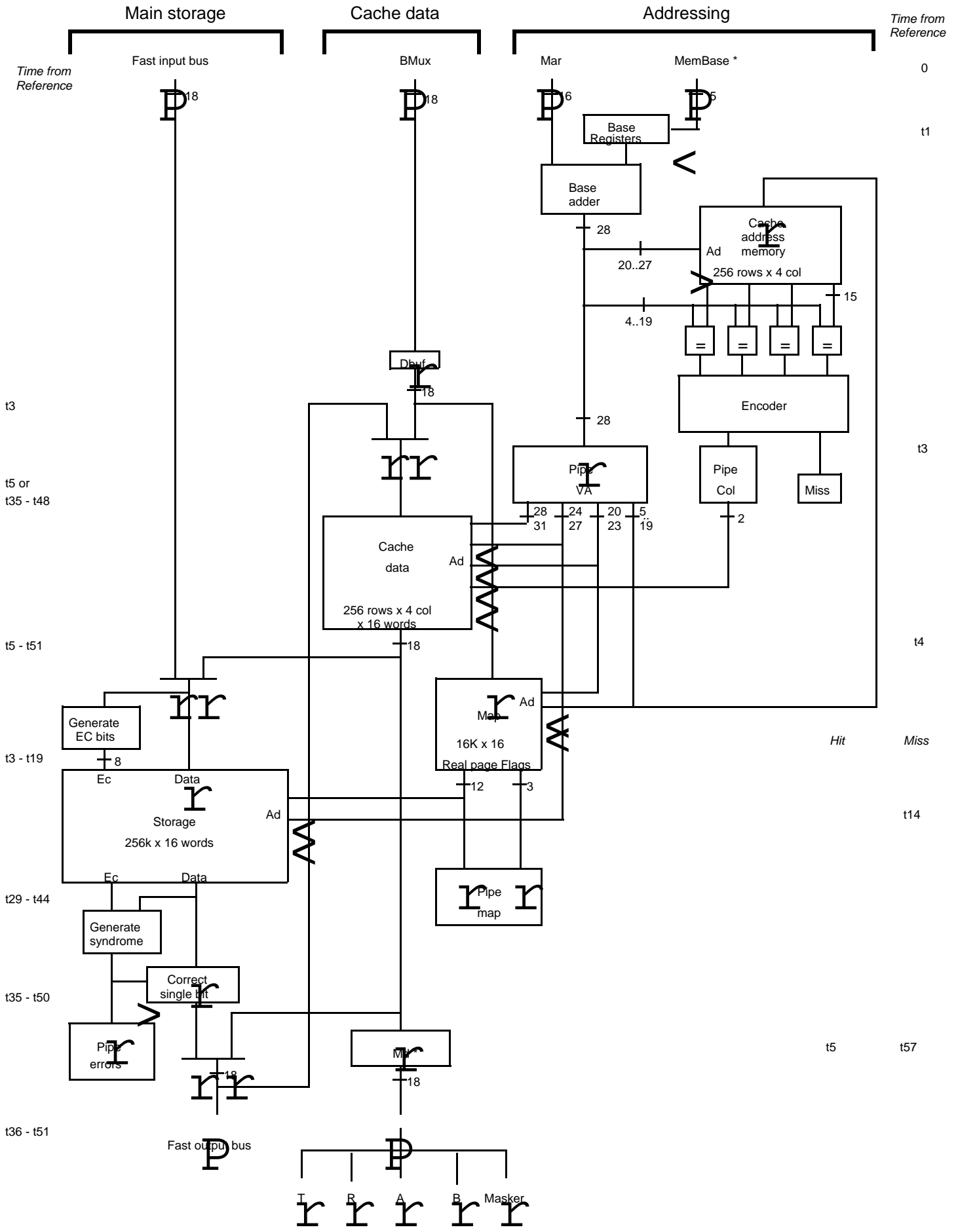


Figure 6

Next Address Formation

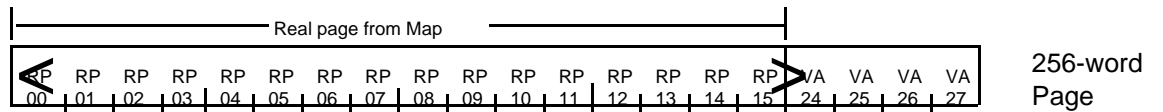
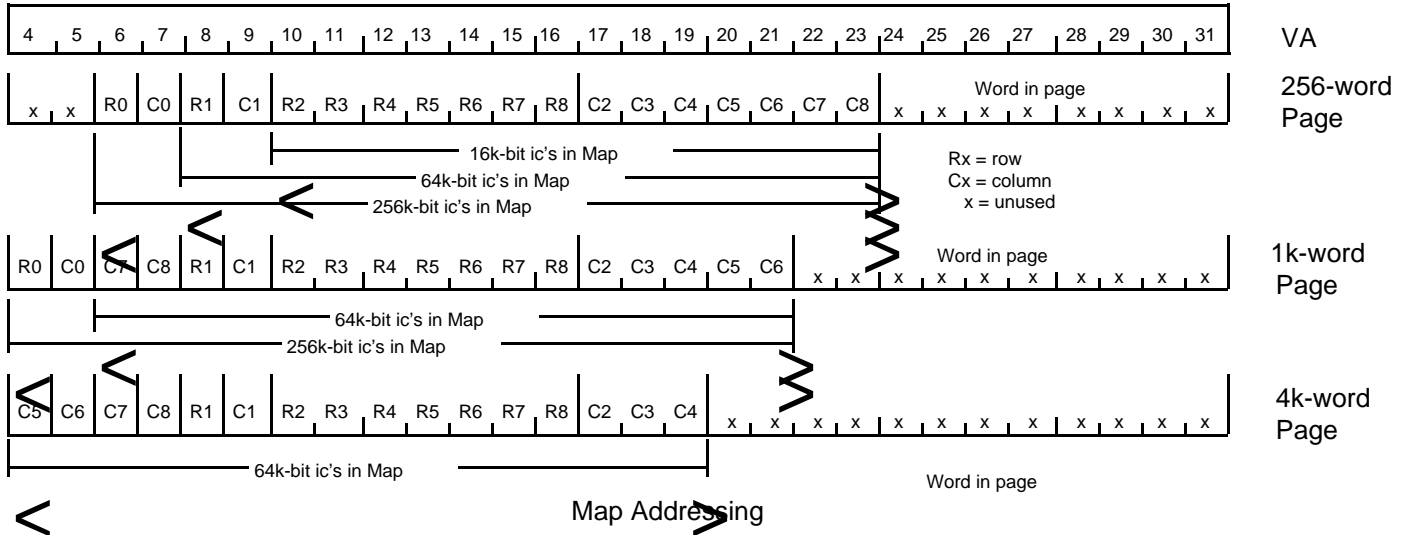
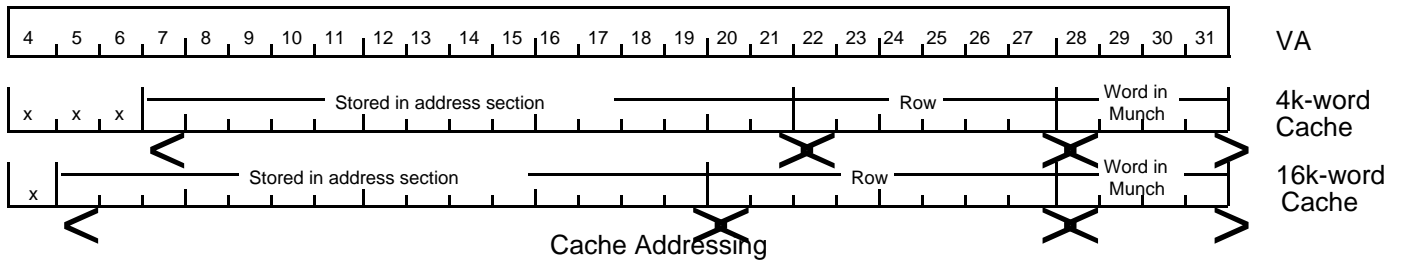


**Figure 7**  
**Instruction Timing**



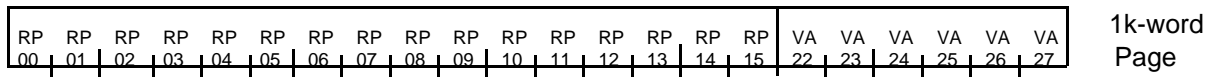
**Figure 8**  
Overall Structure of the Memory System





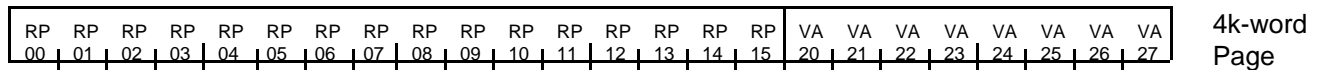
Storage has:

16k ic's	x	x	x	x	M0	M1	R2	C2	R3	R4	R5	R6	R7	R8	C3	C4	C5	C6	C7	C8
64k ic's	x	x	M0	M1	R1	C1	R2	C2	R3	R4	R5	R6	R7	R8	C3	C4	C5	C6	C7	C8
256k ic's	M0	M1	R0	C0	R1	C1	R2	C2	R3	R4	R5	R6	R7	R8	C3	C4	C5	C6	C7	C8



Storage has:

16k ic's	x	x	x	x	x	x	M0	M1	R2	C2	R3	R4	R5	R6	C3	C4	R7	R8	C5	C6	C7	C8
64k ic's	x	x	x	x	M0	M1	R1	C1	R2	C2	R3	R4	R5	R6	C3	C4	R7	R8	C5	C6	C7	C8
256k ic's	x	x	M0	M1	R0	C0	R1	C1	R2	C2	R3	R4	R5	R6	C3	C4	R7	R8	C5	C6	C7	C8

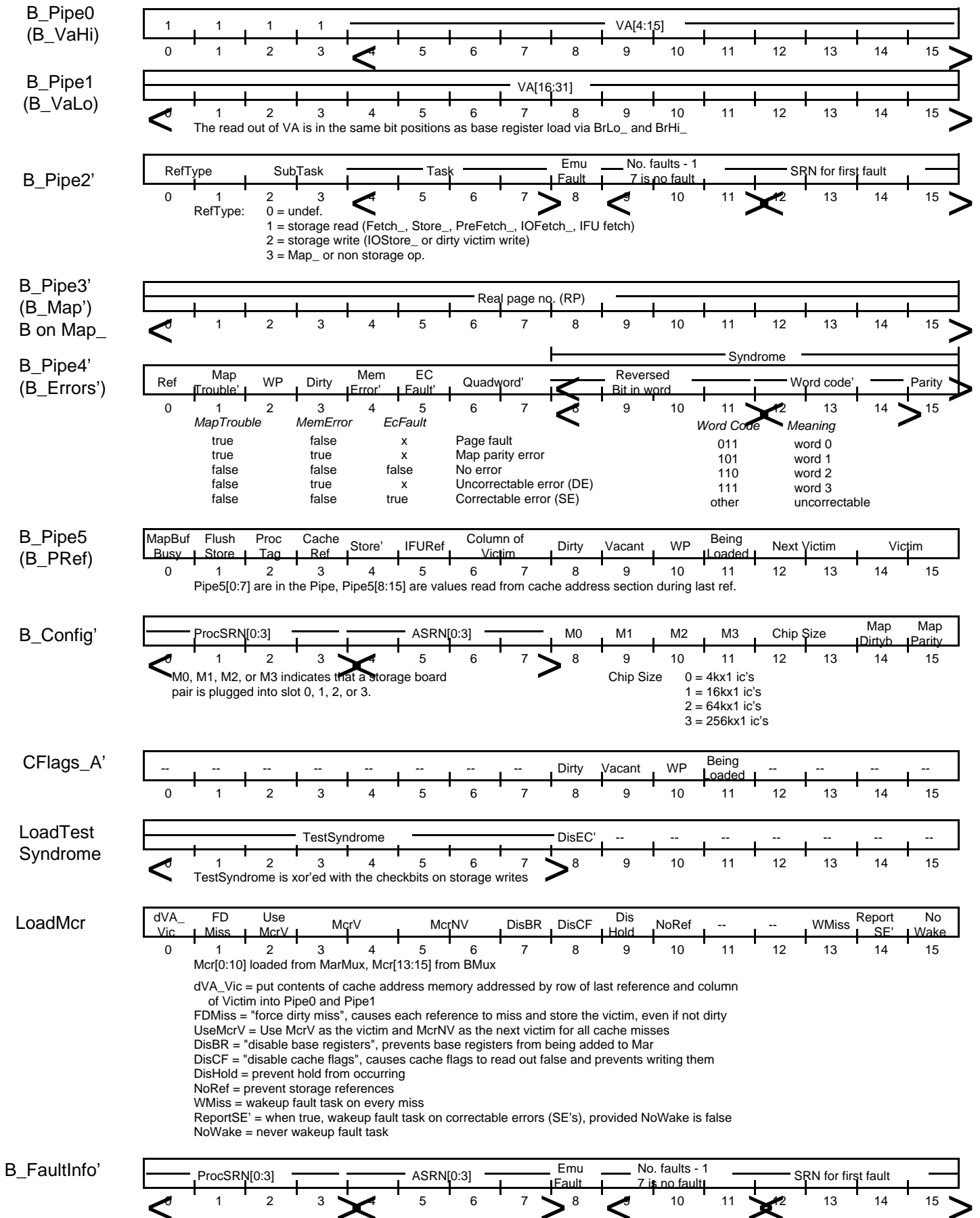


Storage has:

16k ic's	x	x	x	x	x	x	x	x	M0	M1	R2	C2	R3	R4	C3	C4	R5	R6	R7	R8	C5	C6	C7	C8
64k ic's	x	x	x	x	x	x	M0	M1	R1	C1	R2	C2	R3	R4	C3	C4	R5	R6	R7	R8	C5	C6	C7	C8
256k ic's	x	x	x	x	M0	M1	R0	C0	R1	C1	R2	C2	R3	R4	C3	C4	R5	R6	R7	R8	C5	C6	C7	C8

Storage Addressing

**Figure 9**  
**Cache, Map, and Storage Addressing**



**Figure 10**  
**The Pipe and Other Memory Registers**

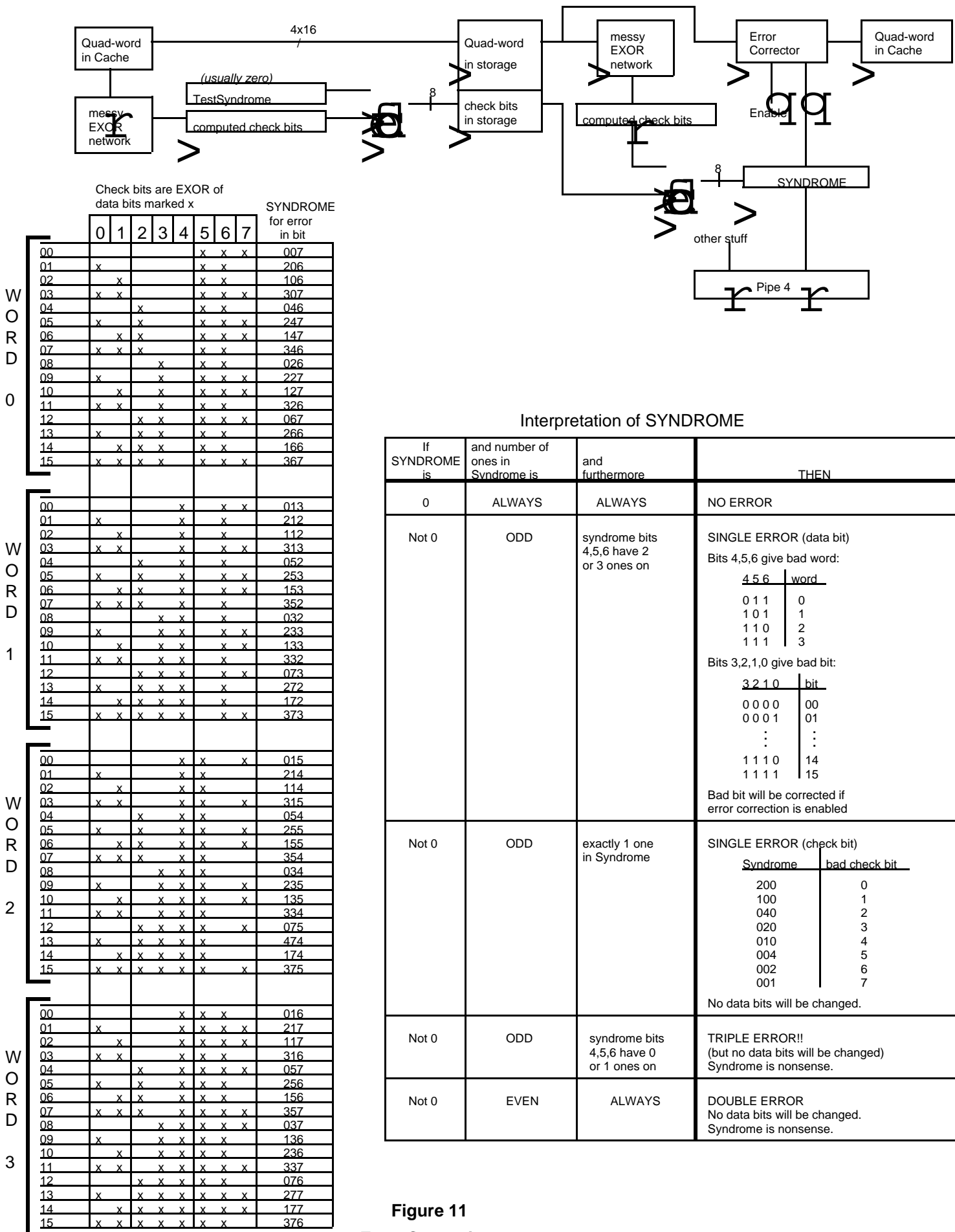
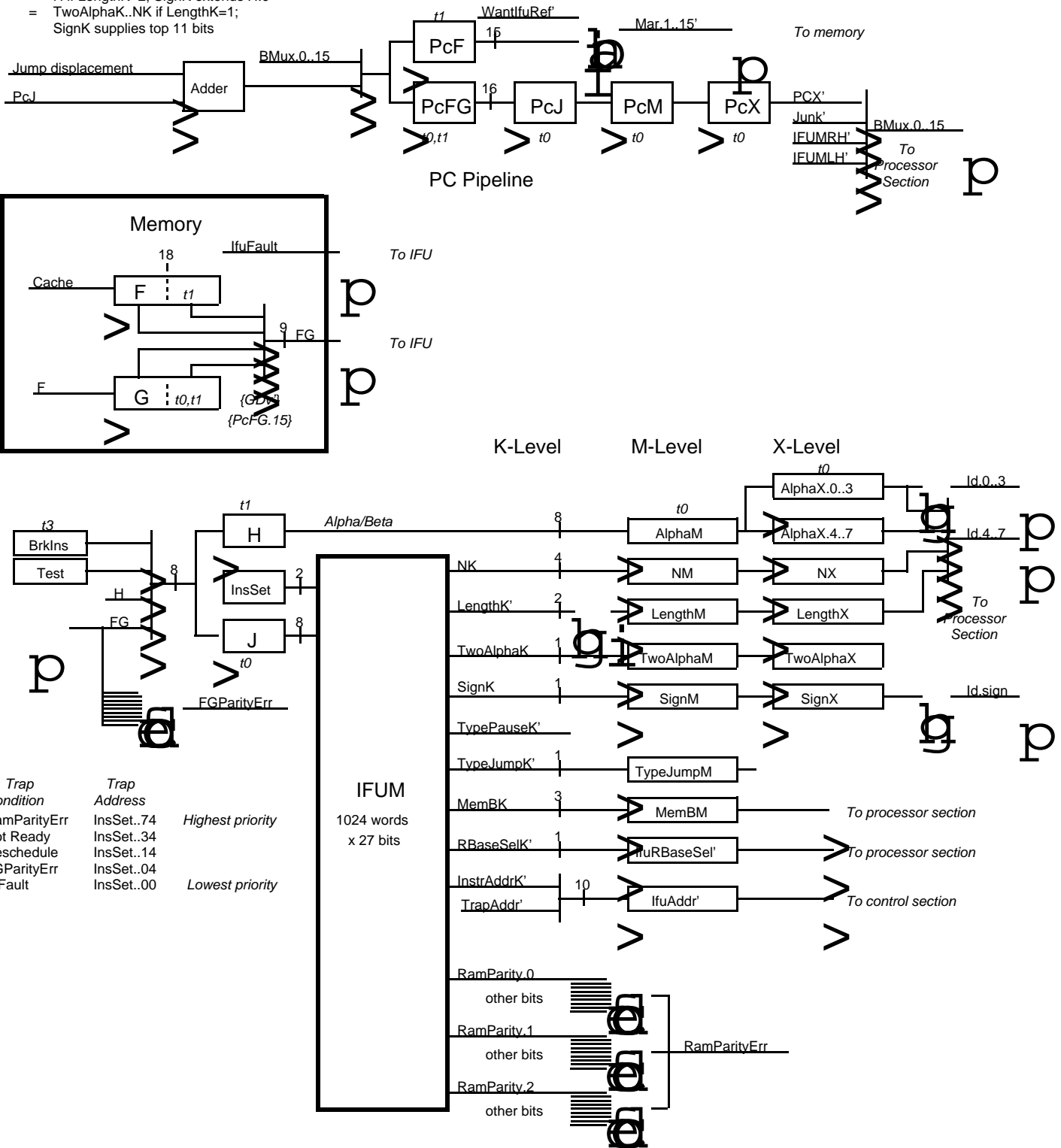


Figure 11  
Error Correction

Jump displacement  
 = H if LengthK=2; SignK extends H.0  
 = TwoAlphaK..NK if LengthK=1;  
 SignK supplies top 11 bits



**Figure 12**  
**Instruction Fetch Unit Organization**

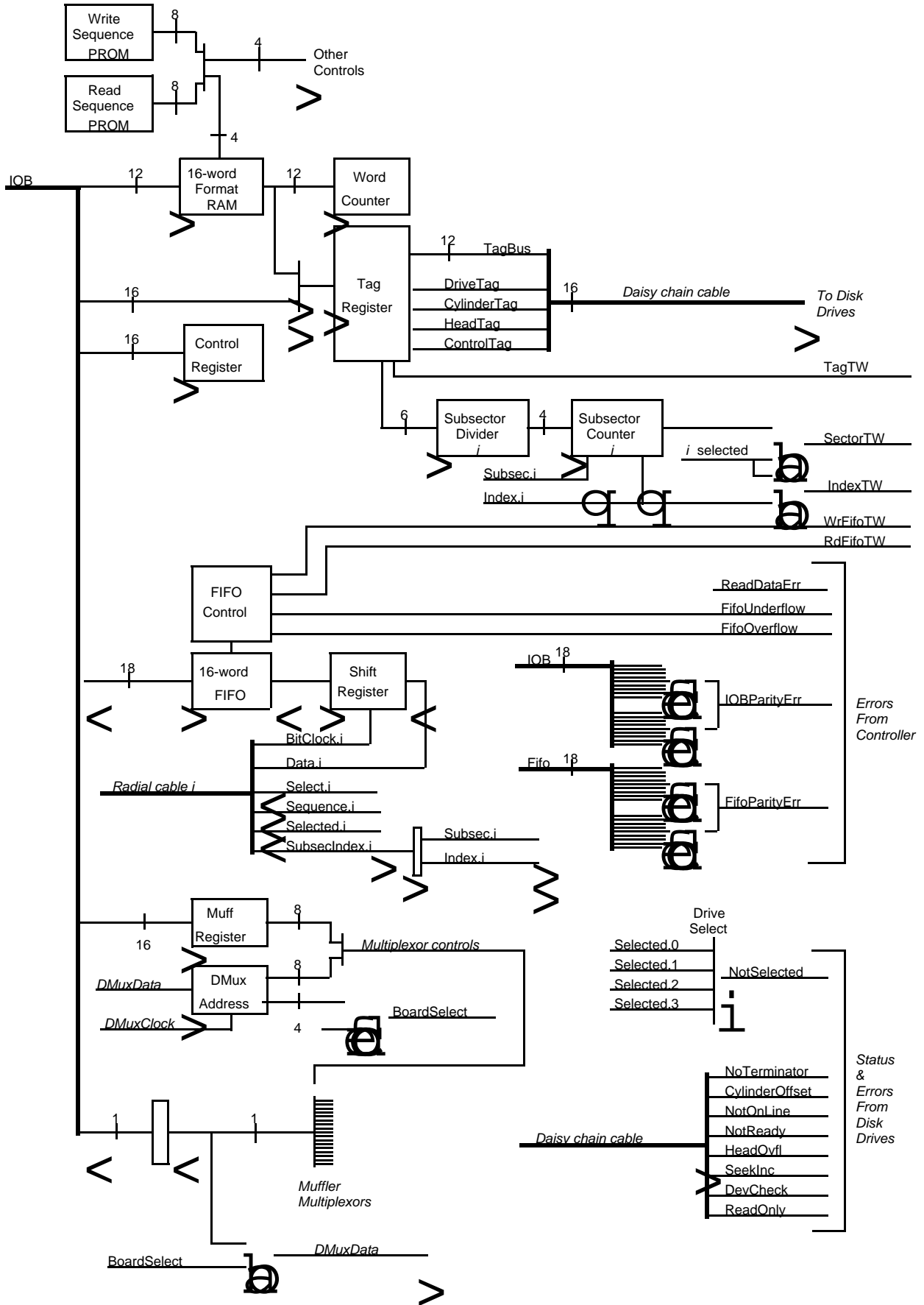
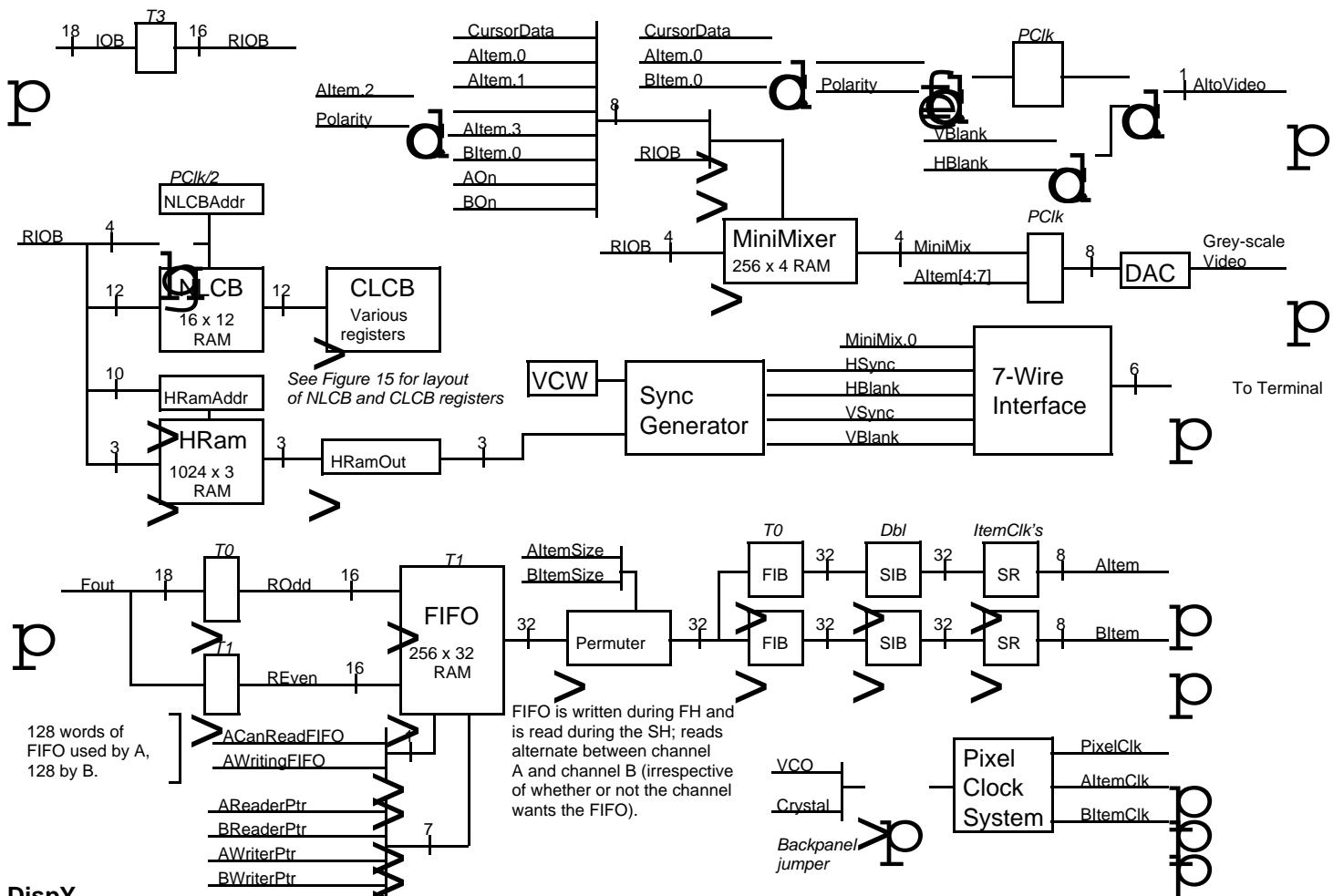
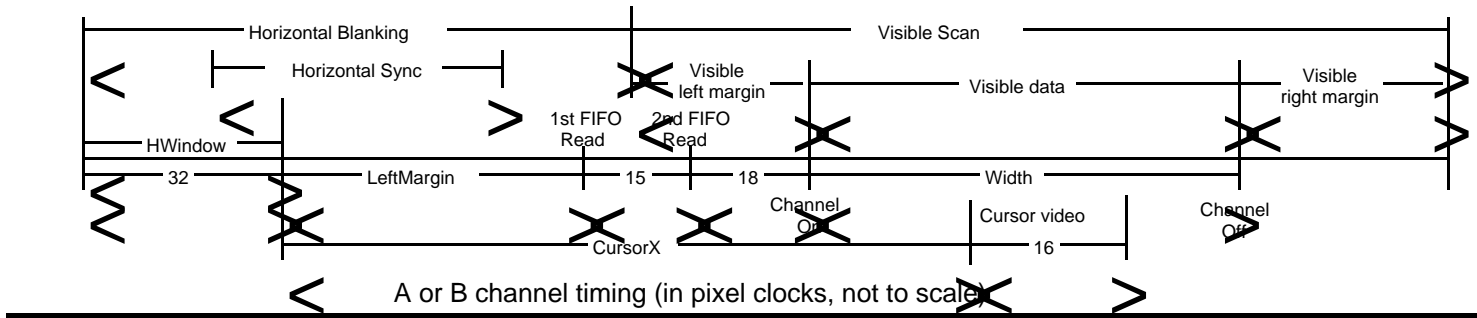
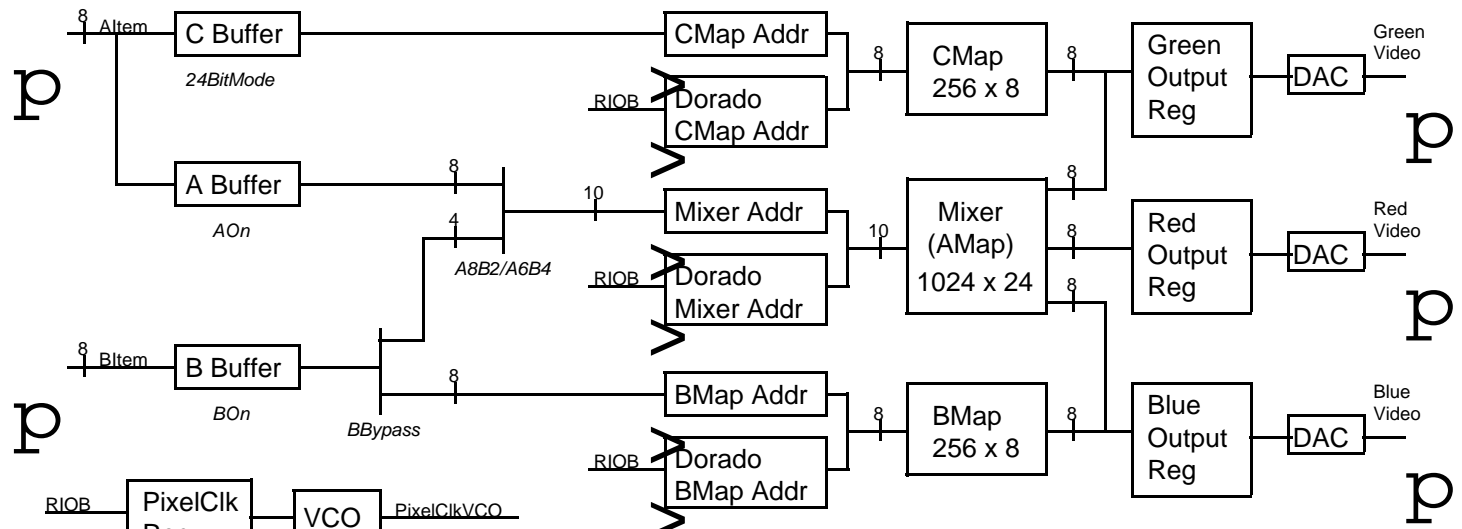


Figure 13  
Disk Controller



**DispY**

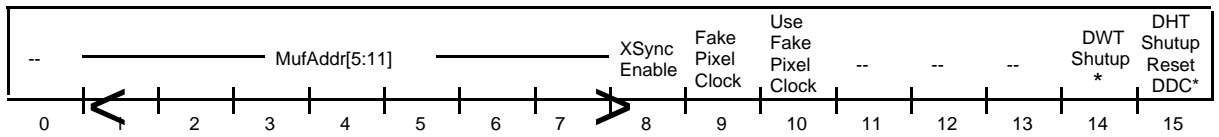
**DispM** (not including independent terminal interface)



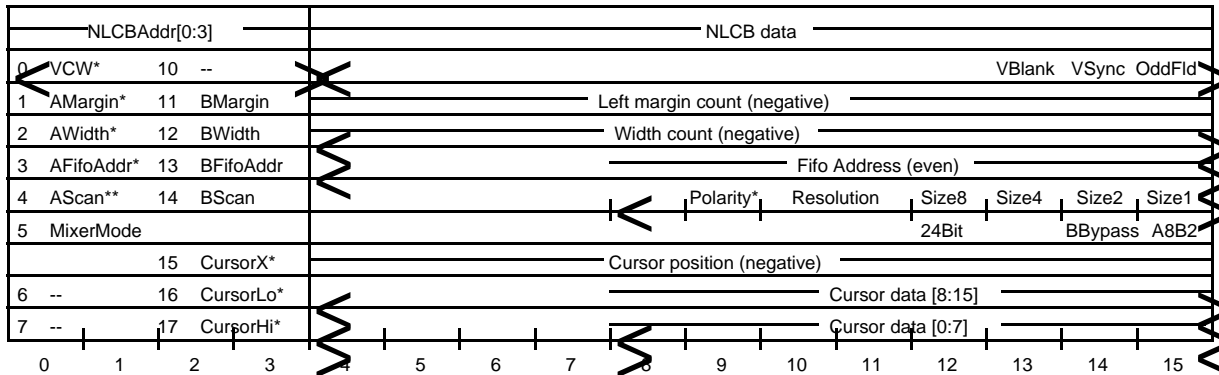
**Figure 14**  
Display Controller

TIOA Output

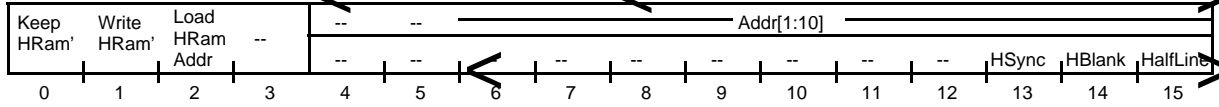
377 (Y)  
367 (M) Statics\*\*



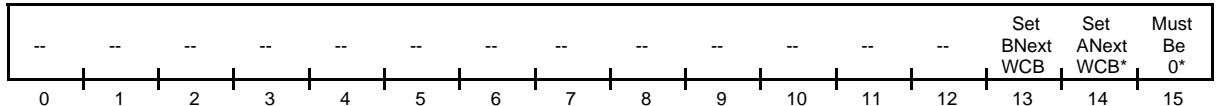
376 (Y)  
366 (M) NLCB\*\*



375 (Y) HRam



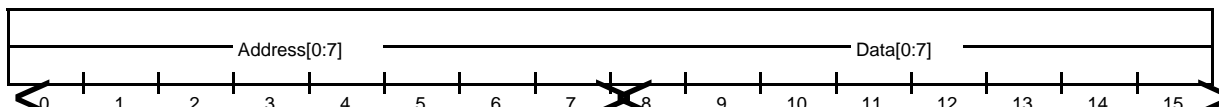
374 (Y)  
364 (M) DHTFlag\*\*



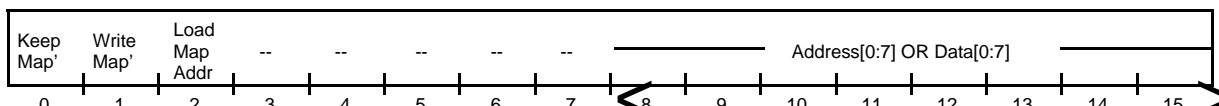
373 (Y)  
363 (M) DWTFlag\*



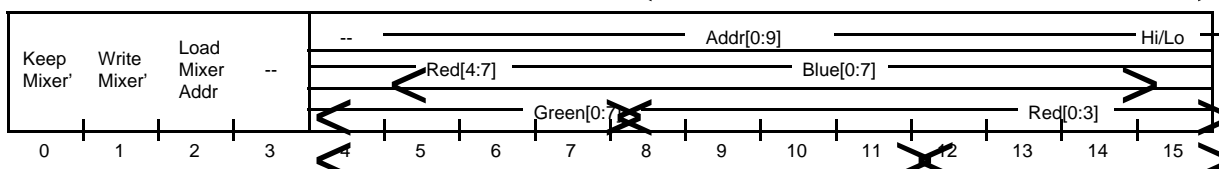
372 (Y) MiniMixer



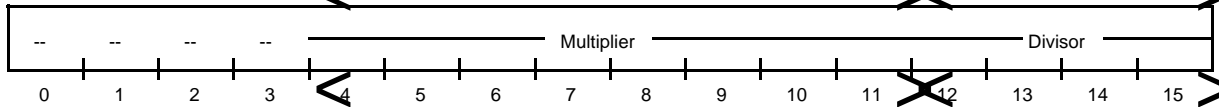
365 (M) BMap  
362 (M) CMap



361 (M) Mixer

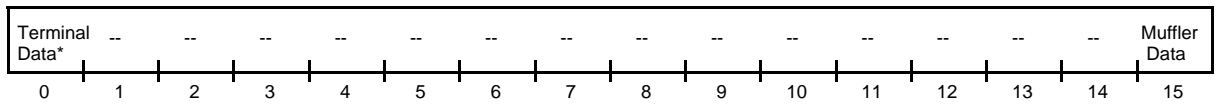


360 (M) PixelClk



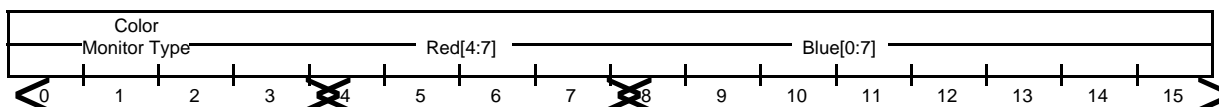
TIOA Input

370 (Y) Status\*\*

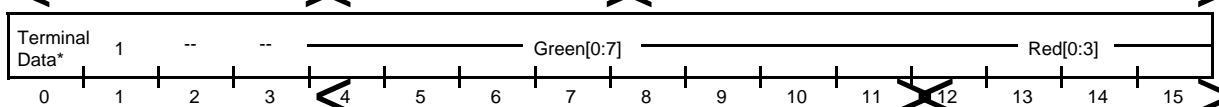


TIOA InputNoPE

361 (M) MapInLo



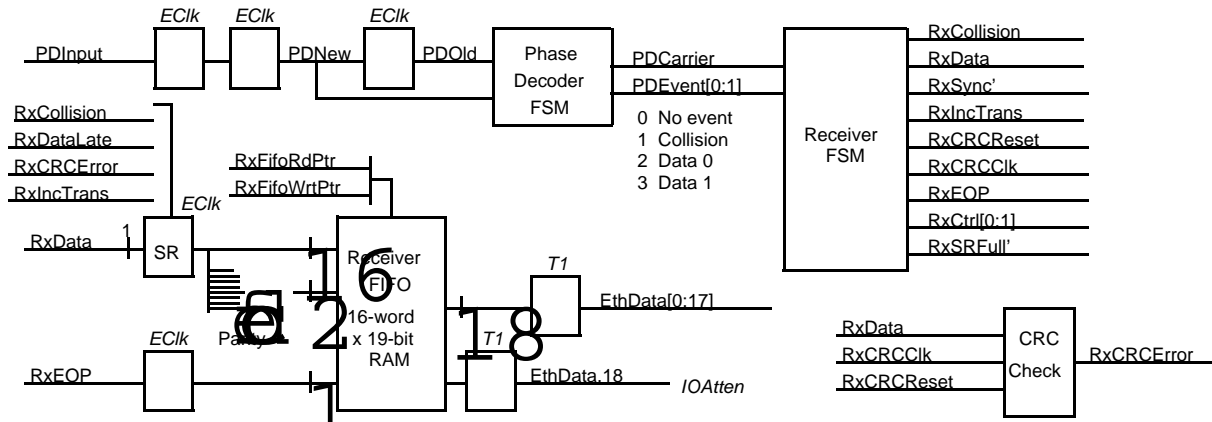
360 (M) MapInHi\*\*



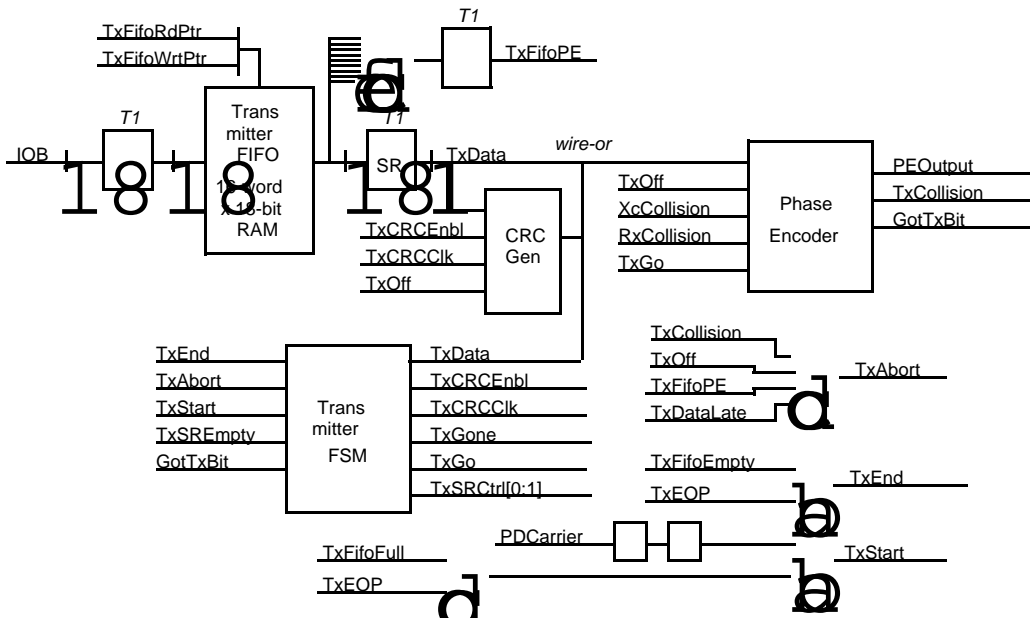
\* Parallel registers DispY/DispM

\*\* Only starred bits or fields are used on DispM; all others are ignored

Figure 15  
Display Controller IO Registers



Receiver



Transmitter

TIOA = 016  
EthC  
Output\_B

Tx Cmd Enbl'	TxOn	TxEOP	Tx Cnt Dwn	Rx Cmd Enbl'	RxOn	RxBOP'	--	Test Cmd Enbl'	Loop Back	Single Step	No Wake ups	Test Clock	Test Coll'	Test Data	Report Colls
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

TestCmdEnbl' enables setting of LoopBack, SingleStep, NoWakeups, TestClock, TestColl', TestData, and ReportColls  
 RxCmdEnbl' enables setting of RxOn and RxBOP'  
 TxCmdEnbl' enables setting of TxOn and TxEOP

EthC  
Pd\_Input

Host Address	RxOn	TxOn	Loop Back	TxColl	No Wake ups	Tx Data Late	Single Step	Tx Fifo PE						
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Host Address is set by backpanel jumpers

TIOA = 015  
EthD  
Pd\_Input

--	--	--	--	--	--	--	Rx Collision	--	Rx Data Late	--	Rx CRC Error	--	--	Rx Inc Trans	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(Receiver status word following end-of-packet)

Figure 16  
Ethernet Controller



### BSEL Decodes

BSEL	Primary	External
0	Md	--
1	RM/STK	--
2	T	--
3	Q	Q_B
4	0,,FF	
5	377,,FF	
6	FF,,0	
7	FF,,377	

### LC Decodes

LC	Meaning
0	No action
1	T_Pd
2	T_Md, RM/STK_Pd
3	T_Md
4	RM/STK_Md
5	T_Pd, RM/STK_Md
6	RM/STK_Pd
7	T_Pd, RM/STK_Pd

### FF Decodes

000-17	A[12:15]_FF[4:7]
020	A_RM/STK
021	A_T
022	A_Md
023	A_Q
024	XorCarry
025	XorSavedCarry
026	Carry20
027	ModStkPBeforeW
030	--
031	ReadMap
032	Pd_Input
033	Pd_InputNoPE
034	RisId
035	TisId
036	Output_B
037	FlipMemBase
040-57	Replace RSTK by FF[4:7] for write
060-67	Branch conditions
070	BigBDispatch_B
071	BDispatch_B
072	Multiply
073	Q_B
074	--
075	TgetsMd
076	FreezeBC
077	Noop
100	PCF_B
101	IFUTest_B
102	IFUTick
103	RescheduleNow
104	--
105	MemBase_B[3:7]
106	RBase_B[12:15]
107	Pointers_B
110-17	--
120-21	--
122	CFlags_A'
123	BrLo_A
124	BrHi_A
125	LoadTestSyndrome
126	LoadMcr[A,B]
127	ProcSRN_B[12:15]
130	InsSetorEvent_B
131	EventCntB_B
132	Reschedule
133	NoReschedule
134	IFUMRH_B
135	IFUMLH_B
136	IFUReset
137	BrkIns_B
140	UseDMD
141	MidasStrobe_B
142	TaskingOff

### ASEL Decodes (FF is ok)

ASEL	FF[0:1]	Meaning
0	0	PreFetch_RM/STK
1	1	Map_RM/STK (emu/fit)
		IOFetch_RM (io)
	2	LongFetch_RM/STK
	3	Store_RM/STK
1	0	DummyRef_RM/STK
	1	Flush_RM/STK (emu/fit)
		IOStore_RM (io)
	2	IFetch_RM/STK
	3	Fetch_RM/STK
2	0	Store_Md
	1	Store_Id
	2	Store_Q
	3	Store_T
3	0	Fetch_Md
	1	Fetch_Id
	2	Fetch_Q
	3	Fetch_T
4	--	A_RM/STK
5	--	A_Id
6	--	A_T
7	--	Shift operation

### ASEL Decodes (FF not ok)

ASEL	Meaning
0	Store_RM/STK
1	Fetch_RM/STK
2	Store_T
3	Fetch_T
4	A_RM/STK
5	A_Id
6	A_T
7	Shift operation

### RSTK Decodes for STK Operations

RSTK[0]	0 = No ovfl/undfl check 1 = Ovfl/undfl check
RSTK[1:3]	Meaning
0	No StkP change
1	StkP_StkP+1
2	StkP_StkP+2
3	StkP_StkP+3
4	StkP_StkP - 4
5	StkP_StkP - 3
6	StkP_StkP - 2
7	StkP_StkP - 1

### ALUFM Control Values

Logical			Arithmetic (no carry)		
Value	Addr	Meaning	Value	Addr	Meaning
1	16	NOT A	0	1	A
3		NOT A OR NOT B	6		2*A
5		NOT A OR B	14	2	A+B
7		A1 (all ones)	22	5	A - B - 1
11		NOT A AND NOT B	36	13	A - 1
13	14	NOT B			
15		A XNOR B, A EQV B, A=B			
17		A OR NOT B			
21		NOT A AND B	200	12	A+1
23	10	A XOR B, A#B	206		2*A+1
25	0	B	214	3	A+B+1
27	7	A OR B	222	4	A - B
31	11	A0 (all zeroes)	236		A
33	15	A AND NOT B			
35	6	A AND B			
37		A			

ALUFM addresses for operations in standard system microcode

### ALUF Shift Decodes

ALUF[0:2]	Meaning
0	ShiftNoMask
1	ShiftLMask
2	ShiftRMask
3	ShiftBothMasks
4	ShMdNoMask
5	ShMdLMask
6	ShMdRMask
7	ShMdBothMasks

### Derivation of Shift Controls

Field:	SHA	SHB	Count	RMask	LMask
ShC bits:	2	3	4:7	8:11	12:15
RF_A	A[2]	A[3]	P+S+1	undefined	15-S
WF_A	A[2]	A[3]	16-P-S-1	16-P-S-1	P
ShC_B	B[2]	B[3]	B[4:7]	B[8:11]	B[12:15]

Shift controls come from Shc when BSEL[0]=0 in the microinstruction that shifts

Shift controls come from FF when BSEL[0]=1, and the source for B is changed to Q

P=A[8:11]=number of bits to the left of the field  
S=A[12:15]=number of bits in the field - 1

143	TaskingOn	165	B_Pipe4' (Errors')	260-61	--
144	StkP_B[8:15]	166	B_Config'	262	Pd_ALUFMRW
145	RestoreStkP	167	B_Pipe5	263	Pd_ALUFMEM
146	Cnt_B	170	B_PCX'	264	Pd_Cnt
147	Link_B	171	B_EventCntA'	265	Pd_Pointers
150	Q lsh 1	172	B_IFUMRH'	266	Pd_TIOA&StkP
151	Q rsh 1	173	B_IFUMLH'	267	Pd_ShC
152	TIOA[0:7]_B[0:7]	174	B_EventCntB'	270	Pd_ALU rsh 1
153	--	175	B_DBuf	271	Pd_ALU rcy 1
154	Hold&TaskSim_B	176	B_RWCPRreg	272	Pd_ALU brsh 1
155	WF_A	177	B_Link	273	Pd_ALU arsh 1
156	RF_A			274	Pd_ALU lsh 1
157	ShC_A	200-17	RBase_FF[4:7]	275	Pd_ALU lcy 1
160	B_FaultInfo'	220-37	Replace RBase by FF[4:7] for write	276	Divide
161	B_Pipe0 (VaHi)		TIOA[5:7]_FF[5:7]	277	CDivide
162	B_Pipe1 (VaLo)	240-47	MemBaseX_FF[6:7]	300-37	MemBase_FF[3:7]
163	B_Pipe2'	250-53	MemBX_FF[6:7]	340-57	Cnt_FF[4:7]
164	B_Pipe3' (Map')	254-57		360-67	WakeUp[FF[4:7]]

Figure 17