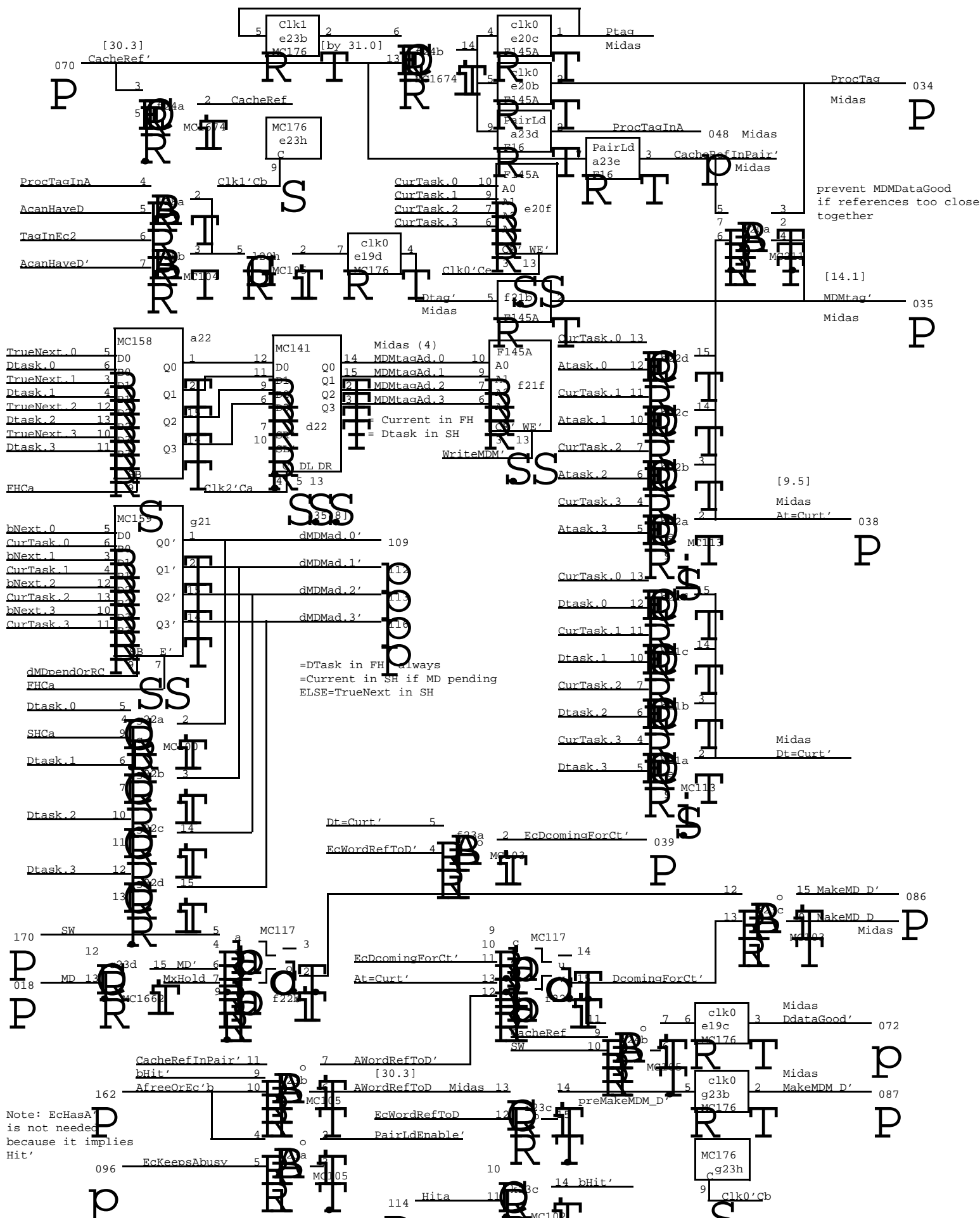


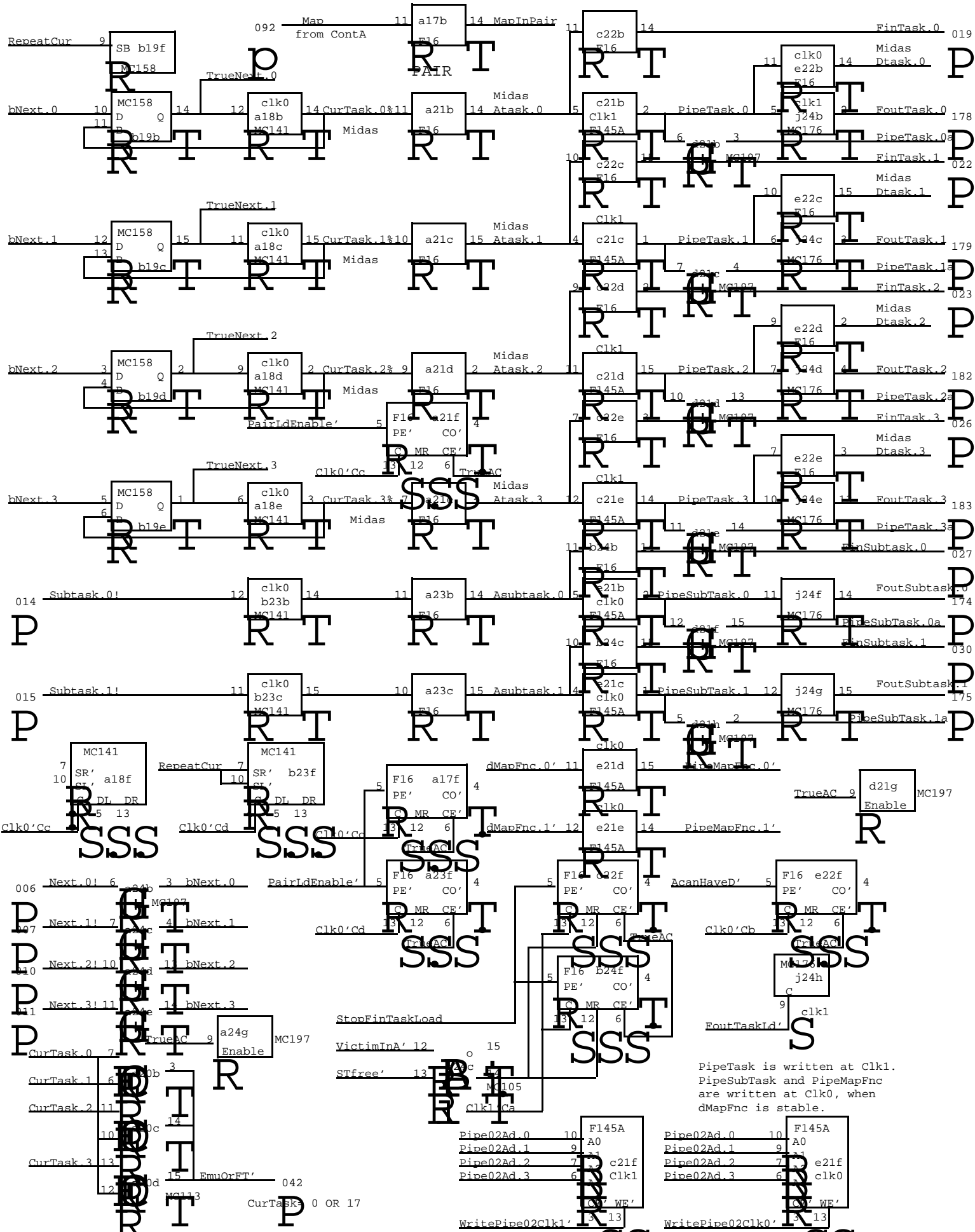
D O R A D O S C H E M A T I C S

M e m X

Table of contents

<u>TITLE</u>	<u>Page</u>
Tags, Task, and MD register controls	01
Next, ATask, FinTask, FouTask, PipeTask, CurTask, DTask, PipeTask	02
SRNs and Fault Information	03
Refresh Control, Hold and RepeatCurrent, Parity errors, _FaultInfo logic	04
Store Transport Automaton	05
Map Automaton	06
Memory Automaton	07
CAS and RAM size control logic	08
Ec1 Automaton	09
Ec2 Automaton	10
BMux drivers and receivers	11
Map Real Page 00-11, Mem Address	12
Map Real Page 12-15, Map Trouble	13
Map Chip Controls	14
Map Address generator	15
Map Chips	16
Map Chips	17
Midas Interface 1	18
Midas Interface 2	19
Clocks	20
Layout	21
PROM programming	22
Configuration instructions	23
Multiwire Rev Changes	24

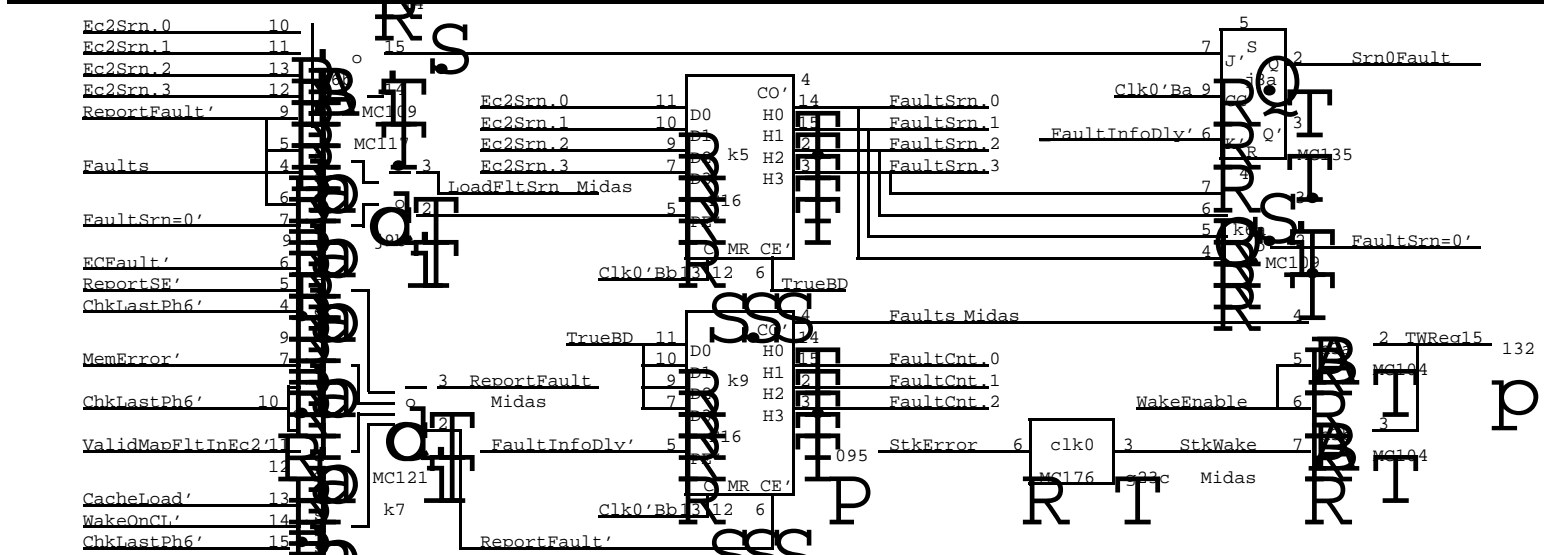
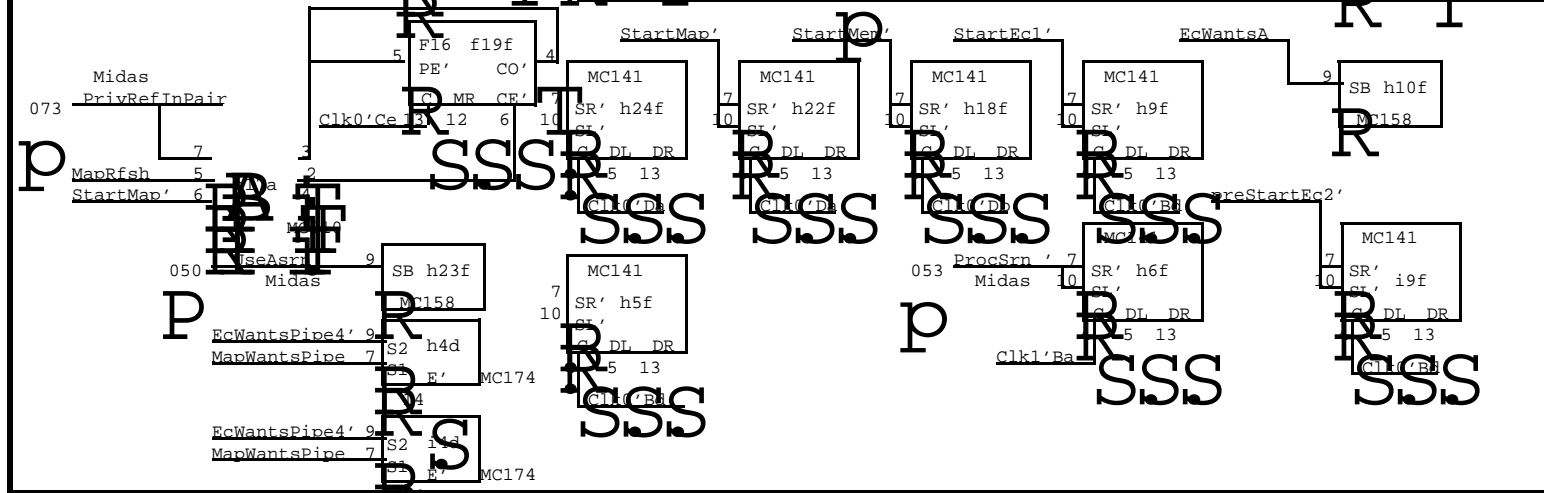
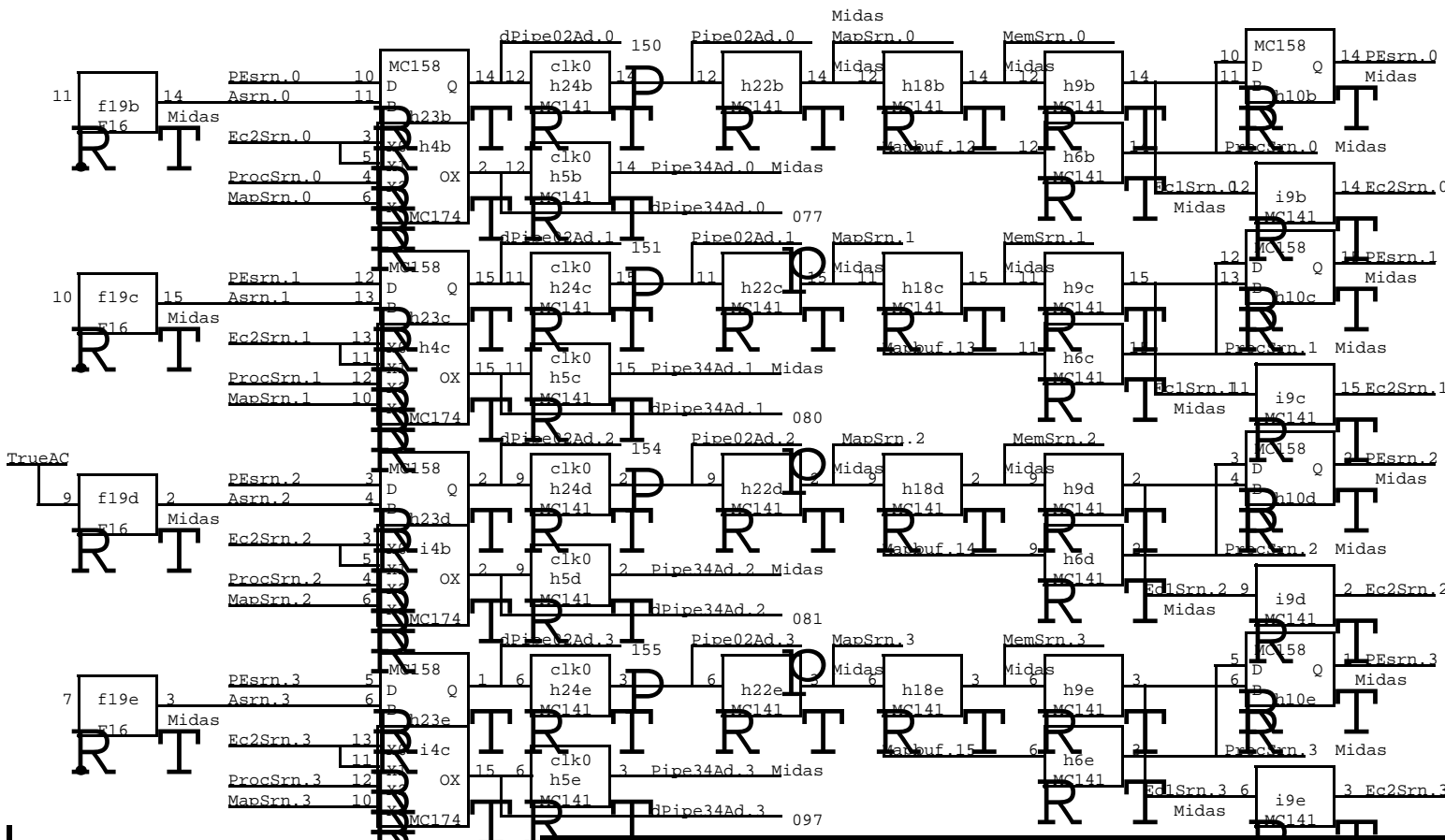




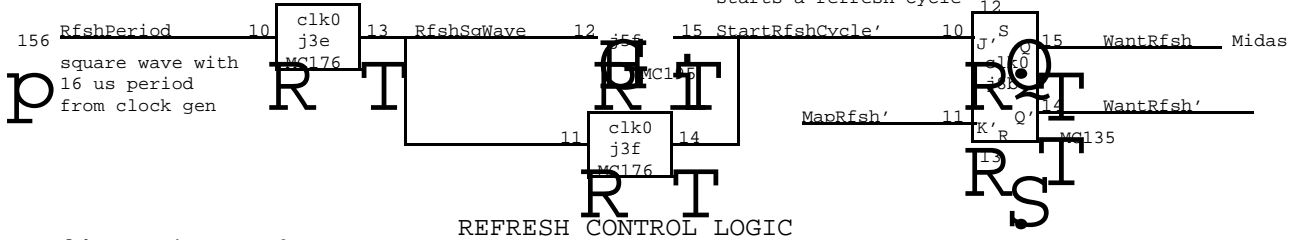
PipeTask is written at Clk1.
 PipeSubTask and PipeMapFnc
 are written at Clk0, when
 dMapFnc is stable.

Pipe02Ad_0	10	F145A
Pipe02Ad_1	9	A0
Pipe02Ad_2	7	c21f
Pipe02Ad_3	6	Clk1

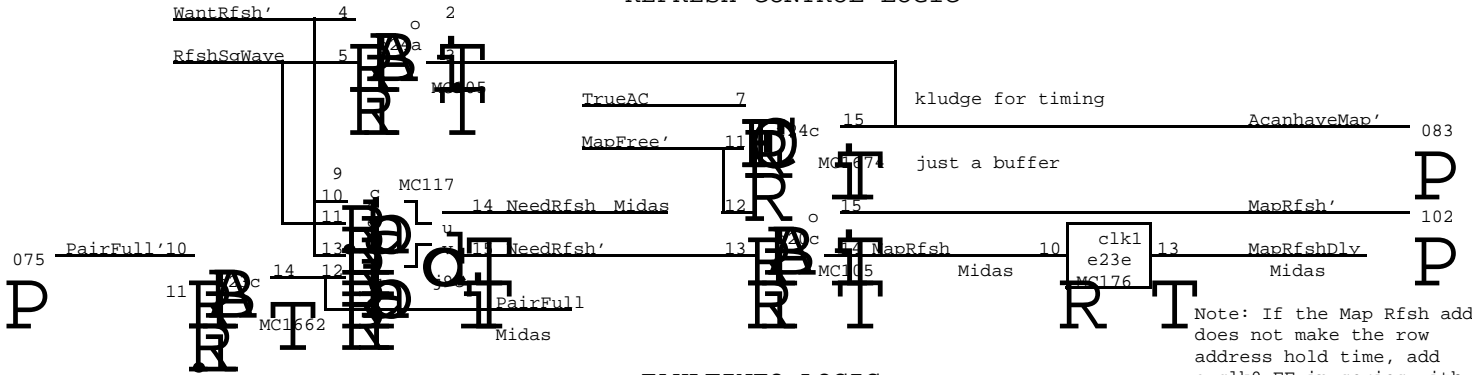
Pipe02Ad_0	10	F145A
Pipe02Ad_1	9	A0
Pipe02Ad_2	7	c21f
Pipe02Ad_3	6	clk0



rising edge of RefreshPeriod starts a refresh cycle

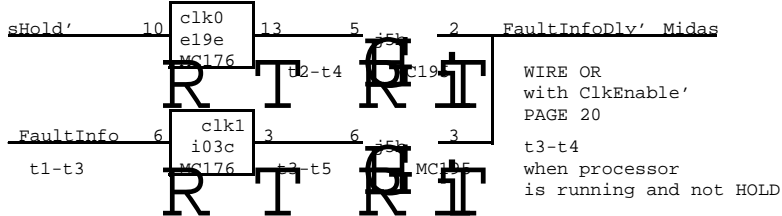


REFRESH CONTROL LOGIC

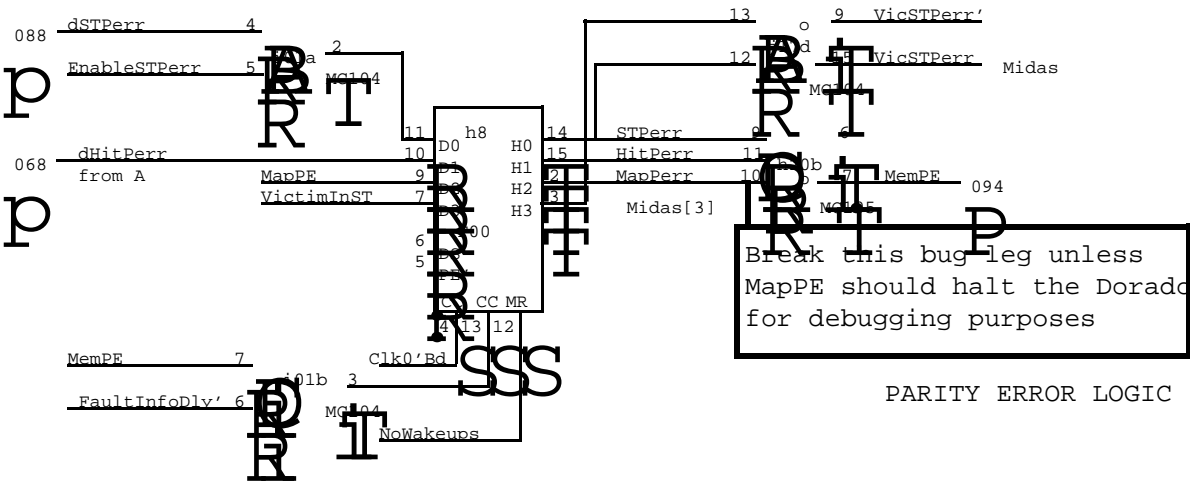


FAULTINFO LOGIC

Note: If the Map Rfsh addr does not make the row address hold time, add a clk0 FF in series with this clk1 FF and OR the outputs with an OR gate.



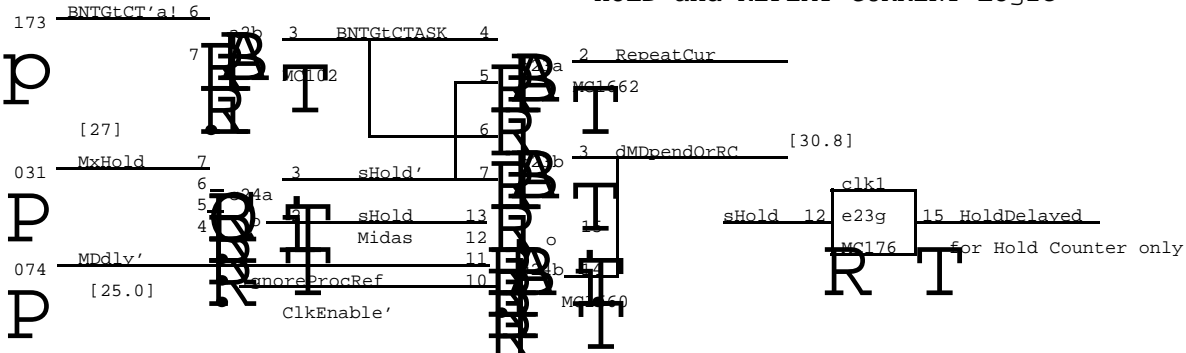
WIRE OR with ClkEnable' PAGE 20
t3-t4 when processor is running and not HOLD



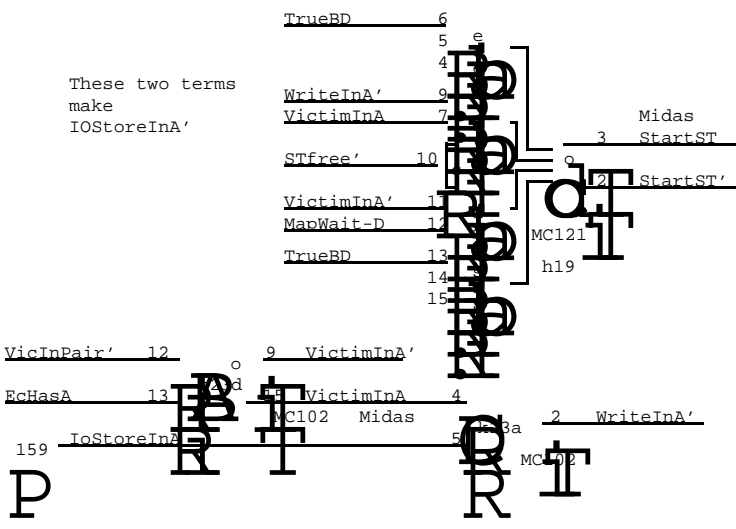
Break this bug leg unless MapPE should halt the Doradd for debugging purposes

PARITY ERROR LOGIC

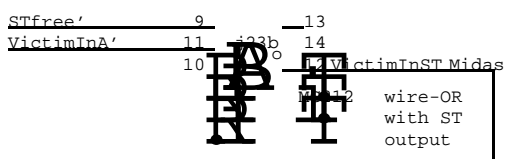
HOLD and REPEAT CURRENT Logic



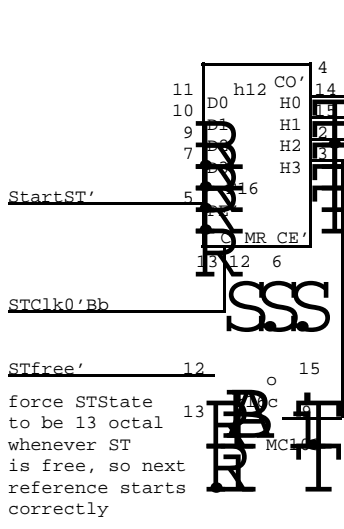
These two terms
make
IOStoreInA'



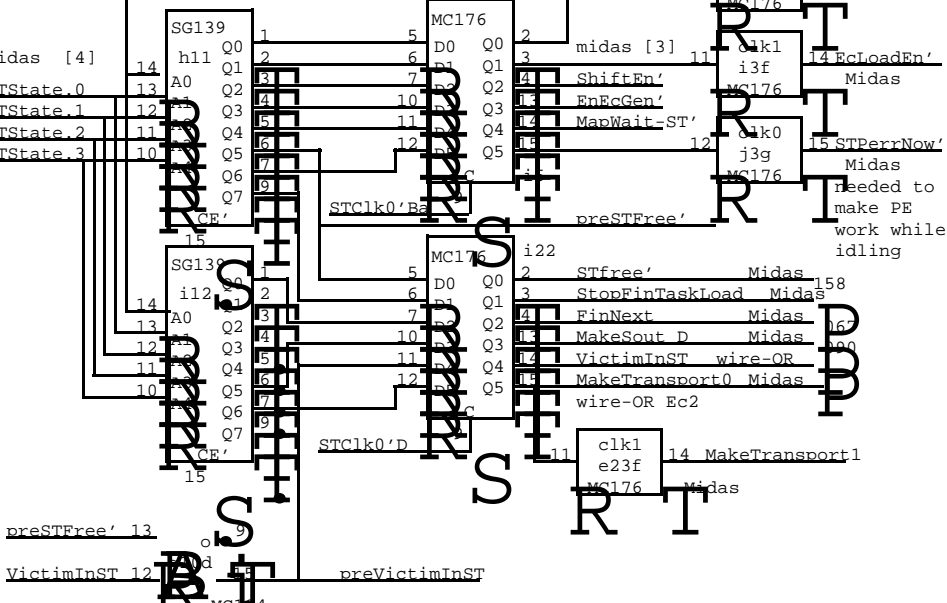
P



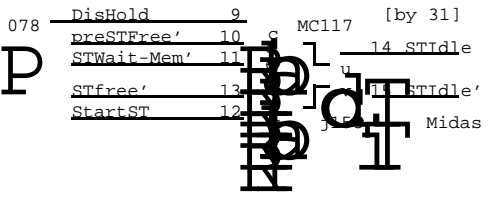
wire-OR
with ST
output



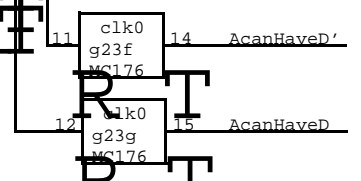
force STState
to be 13 octal
whenever ST
is free, so next
reference starts
correctly

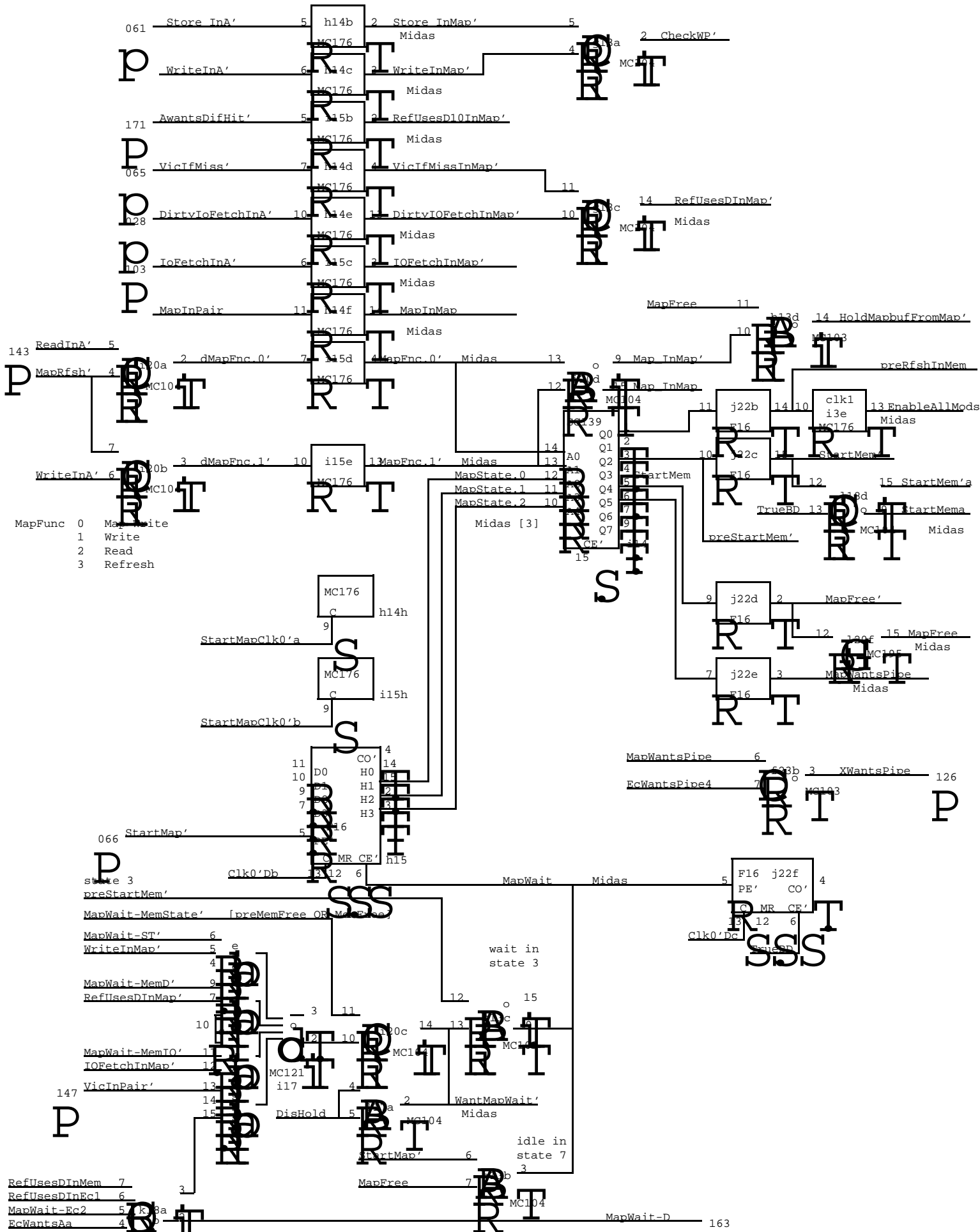


preSTFree' 13
VictimInST 12
preVictimInST



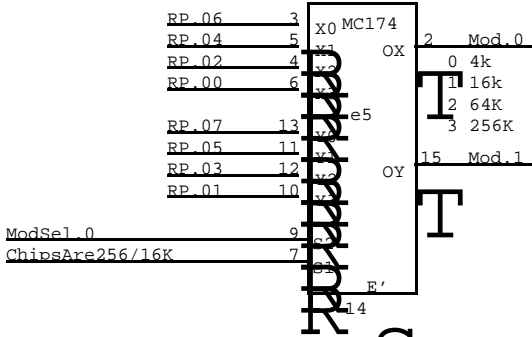
P



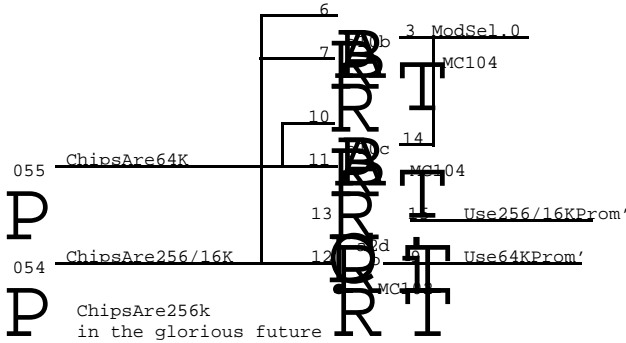


Note: Chip size options are available:
 16K only, 16K or 64K, 64K only, 64K
 or 256K, 256K only.

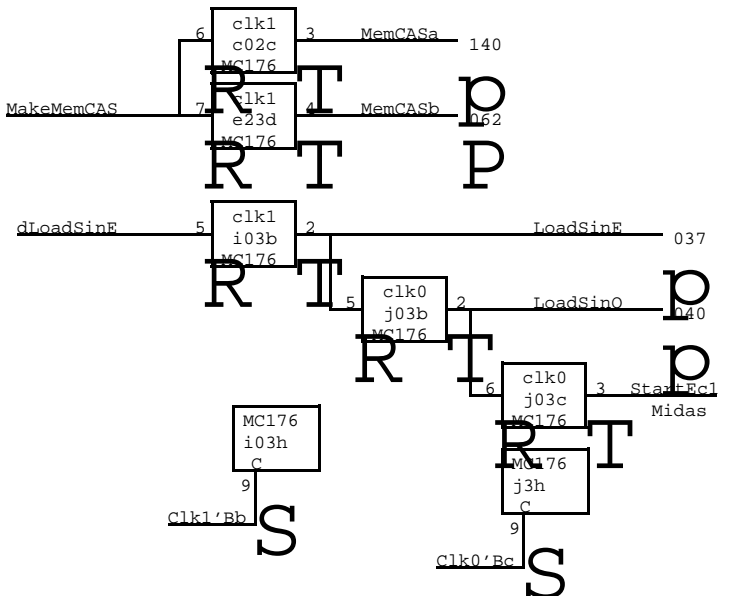
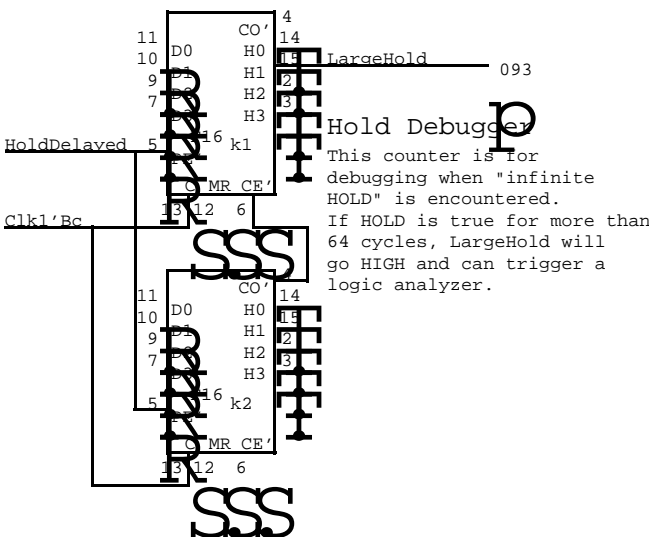
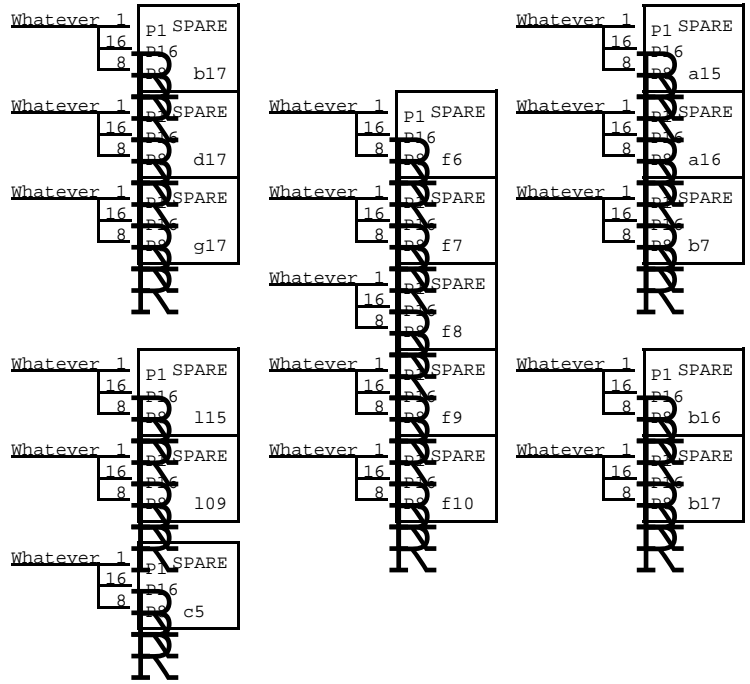
Only one of the ChipsAreXXX signals should
 be true at any time.

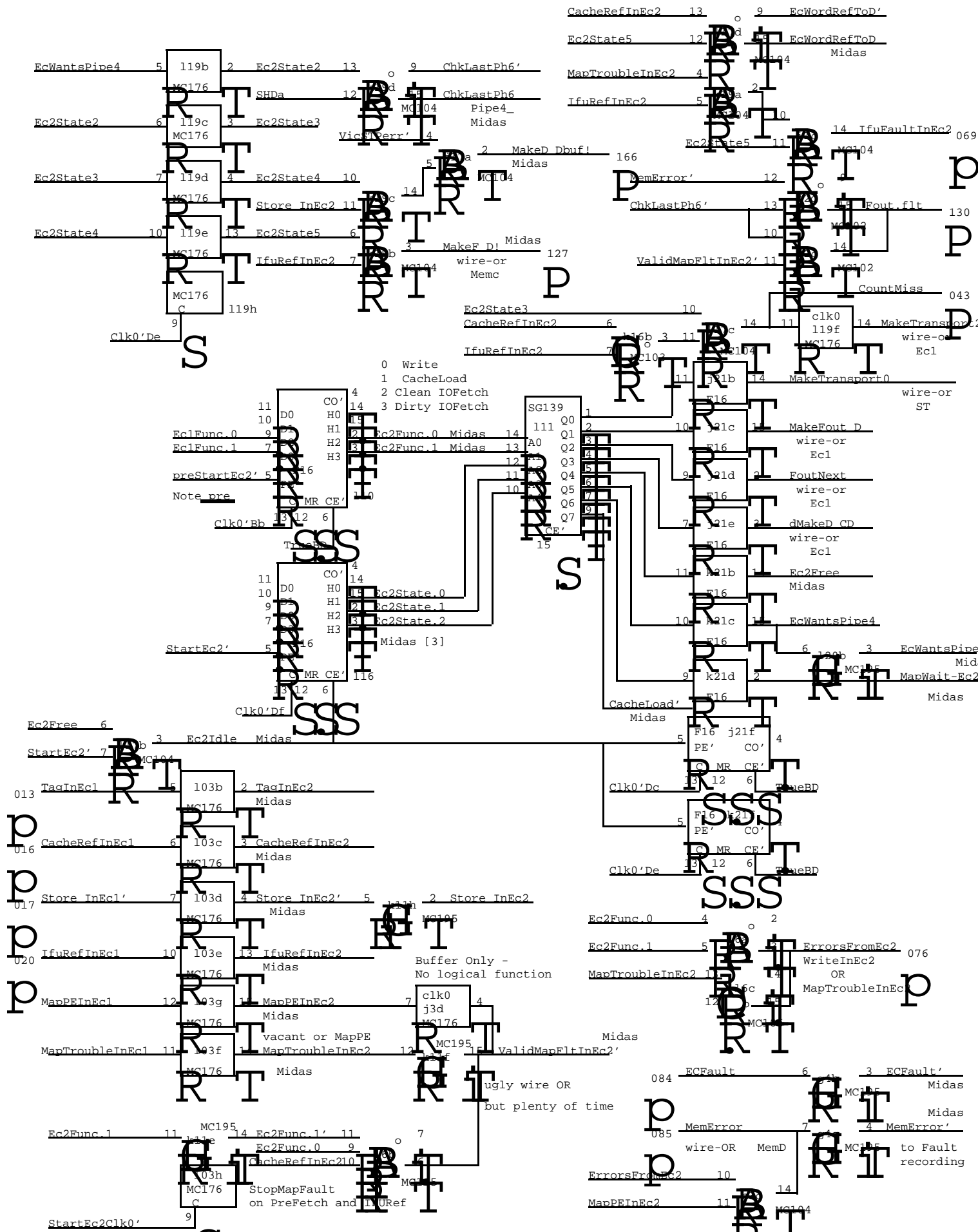


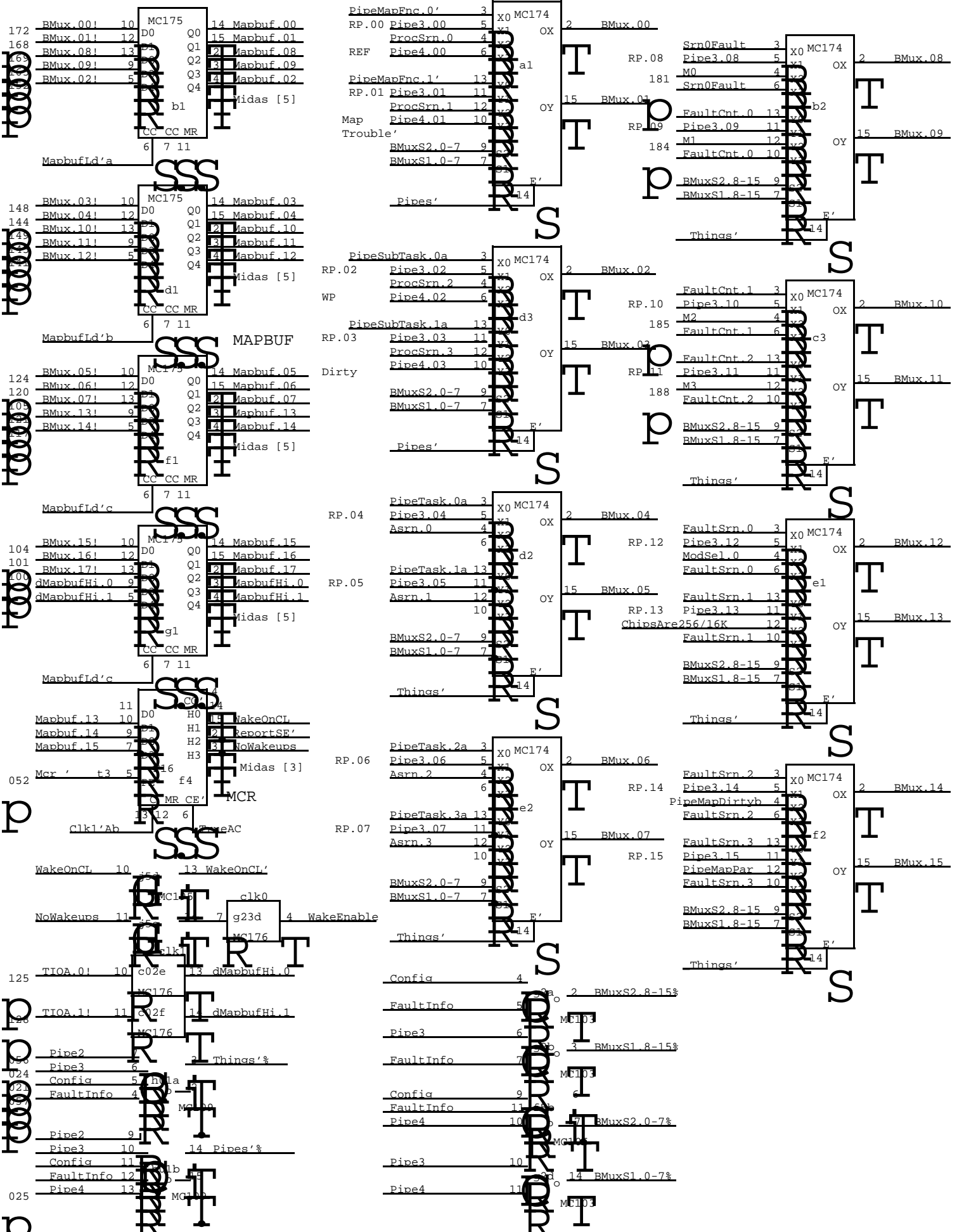
BREAK THIS BUG
 UNTIL 256K CHIPS
 ARE USED

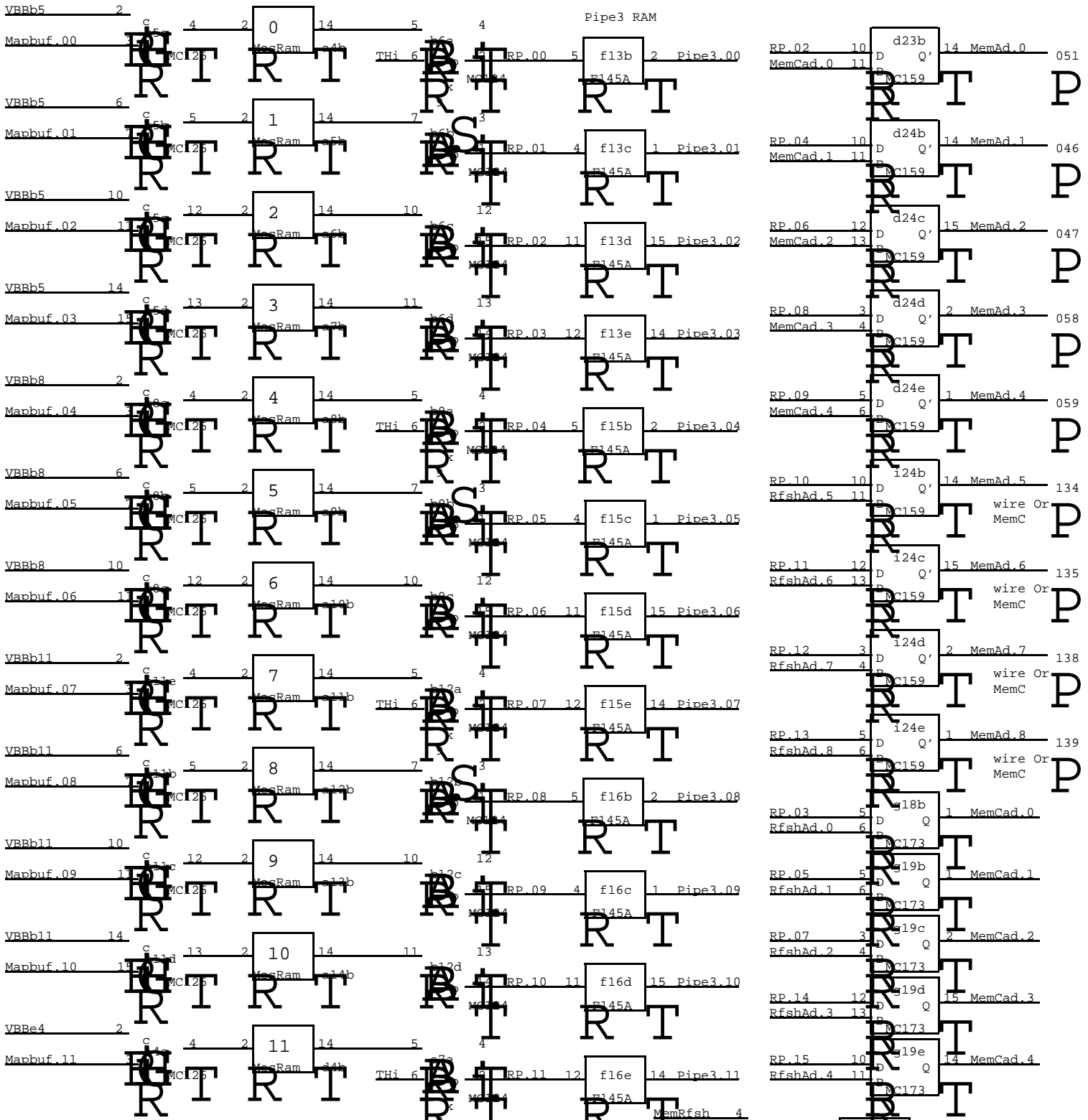


When the only two chip
 options for Dorado are
 64K or 256K, then interpret
 this signal as ChipsAre256K
 and enable the upper gate of ModSel.0



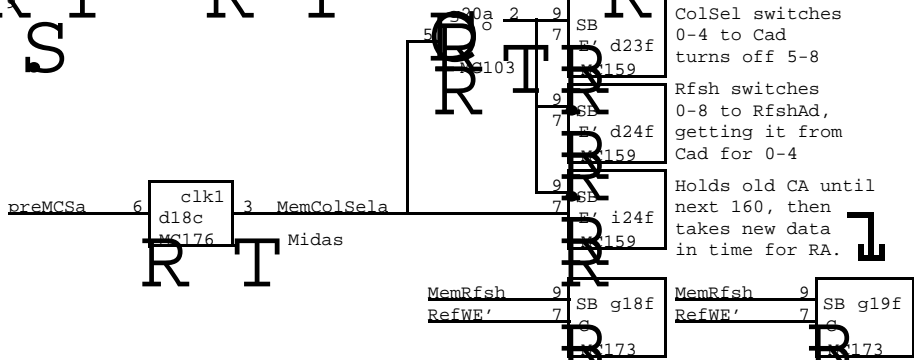


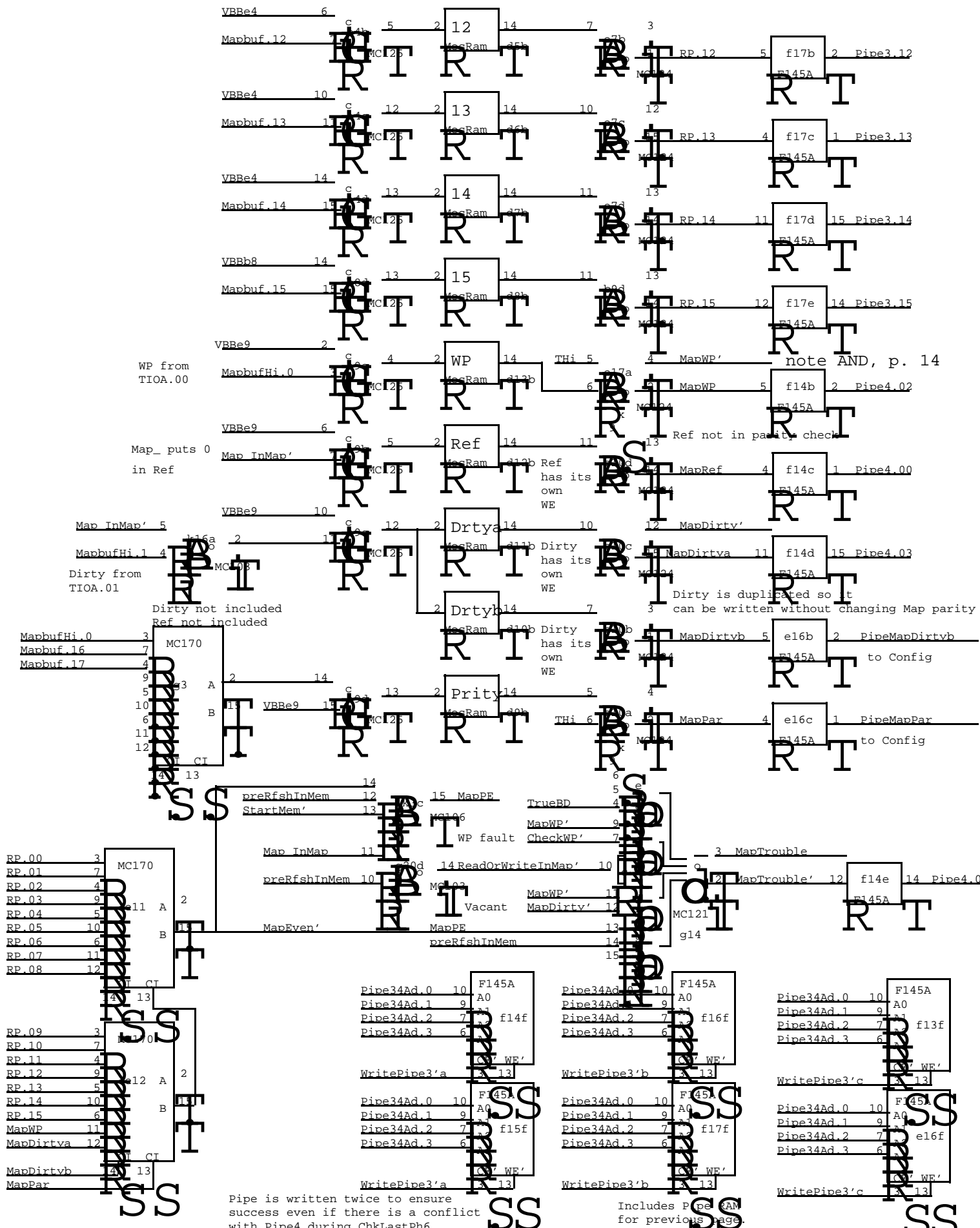




Chir	Mod	Row	Col
0 1	0 1 2 3 4 5 6 7 8	0 1 2 3 4 5 6 7 8	0 1 2 3 4 5 6 7 8
4k	6 7 - - - 8 9 10 11 12 13	- - - 14 15	from
16k	4 5 - - 6 8 9 10 11 12 13	- - 7 14 15	C
65k	2 3 - 4 6 8 9 10 11 12 13	- 5 7 14 15	board
256	0 1 2 4 6 8 9 10 11 12 13	3 5 7 14 15	

Real memory is always contiguous, and RP bits which select the module shift left as storage chips grow, thus:





WP from TIOA.00

Map_puts 0 in Ref

Map InMap' 5
MapbufHi.1 4
Dirty from TIOA.01

Dirty not included
Ref not included

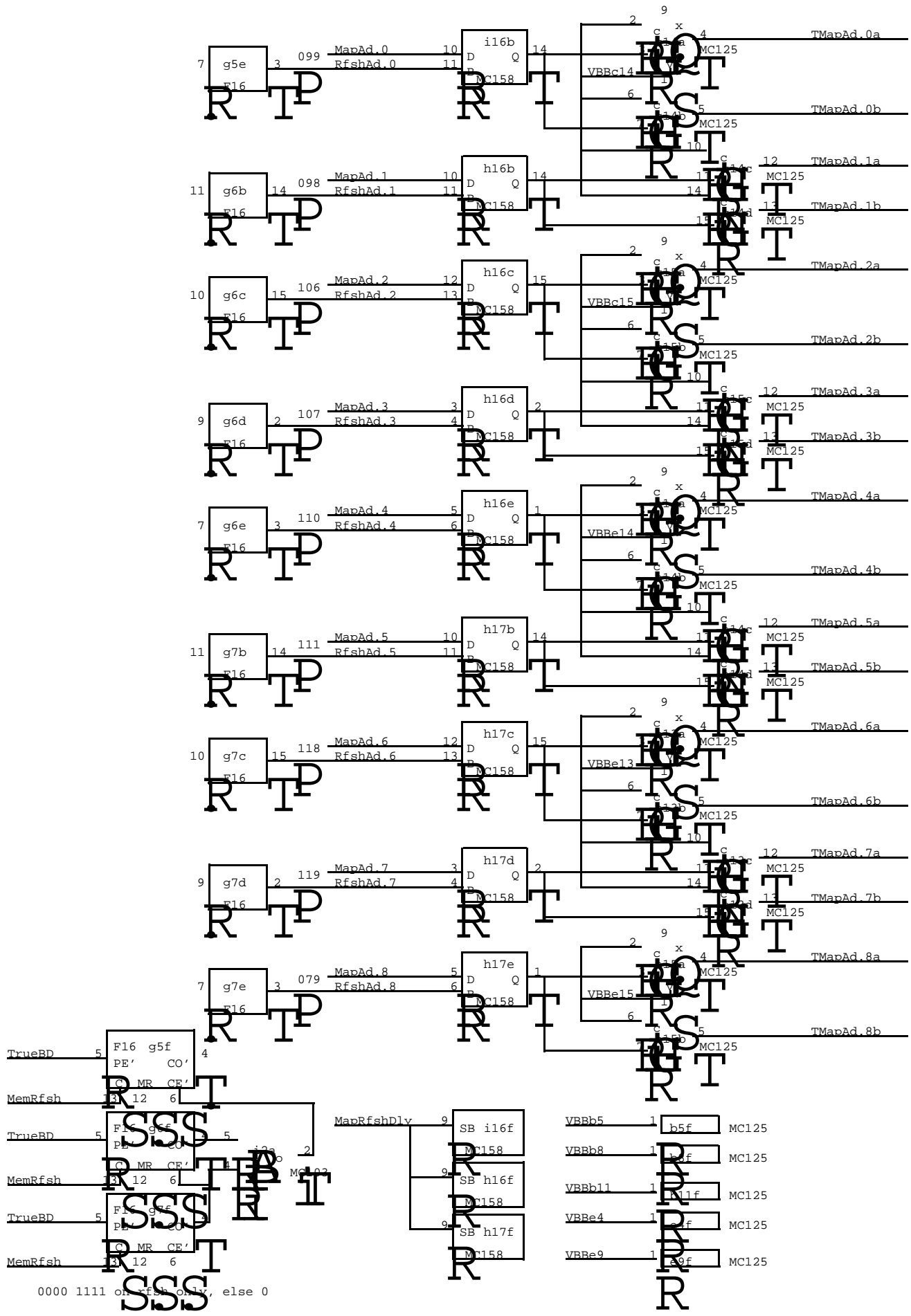
MapbufHi.0 3
Mapbuf.16 7
Mapbuf.17 4

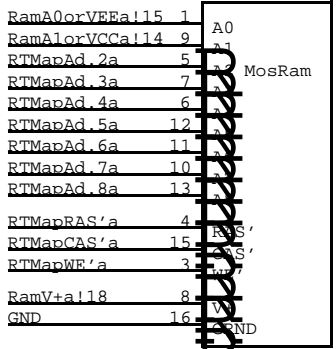
RP.00 3
RP.01 7
RP.02 4
RP.03 9
RP.04 5
RP.05 10
RP.06 6
RP.07 11
RP.08 12

RP.09 3
RP.10 7
RP.11 4
RP.12 9
RP.13 5
RP.14 10
RP.15 6
MapWP 11
MapDirtya 12
MapDirtyb
MapPar

Pipe is written twice to ensure success even if there is a conflict with Pipe4 during ChkLastPh6

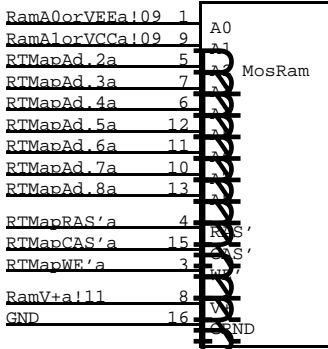
Includes Pipe4M for previous page





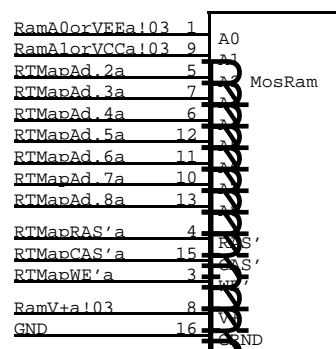
RP.00

a4



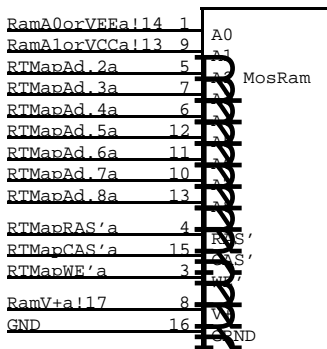
RP.05

a9



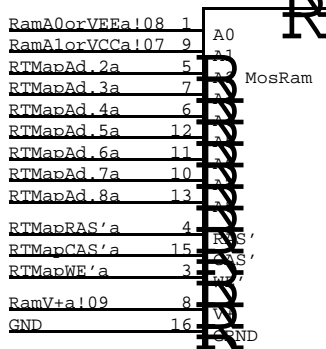
RP.10

a14



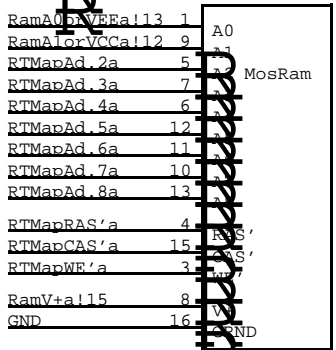
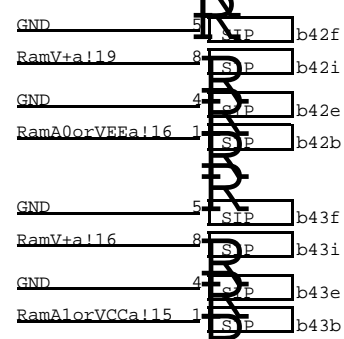
RP.01

a5



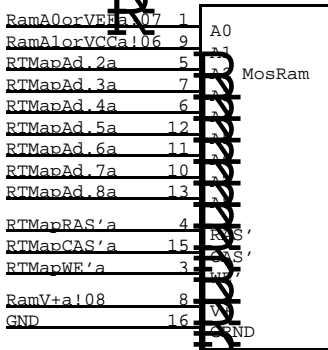
RP.06

a10



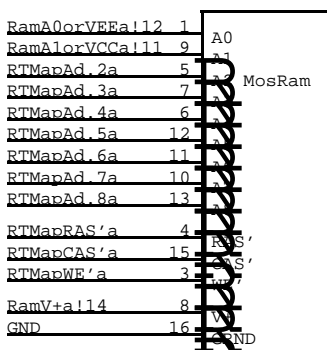
RP.02

a6



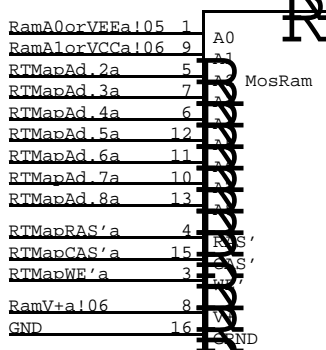
RP.07

a11



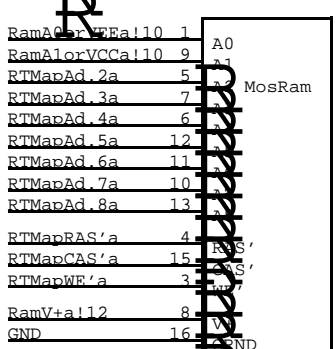
RP.03

a7



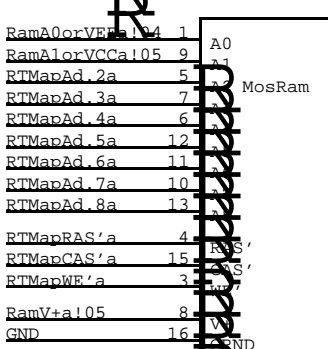
RP.08

a12



RP.04

a8



RP.09

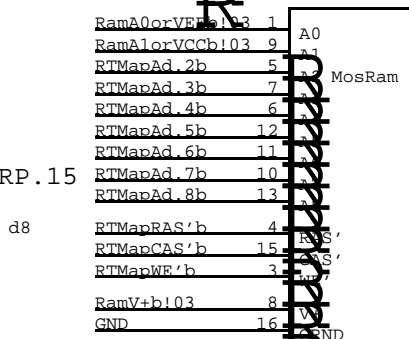
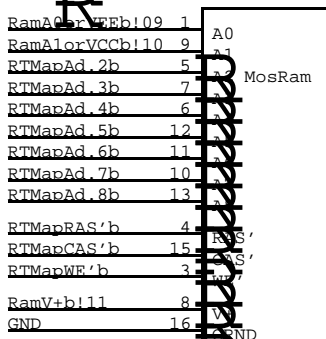
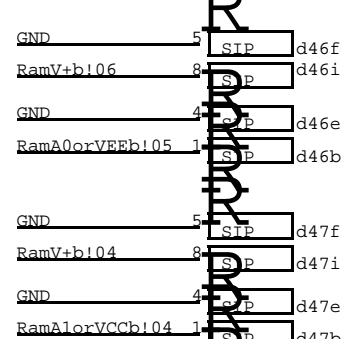
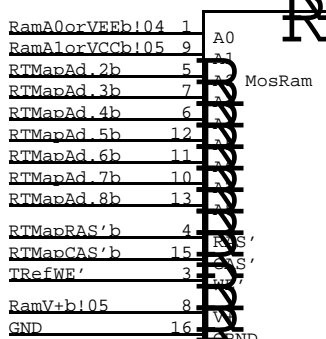
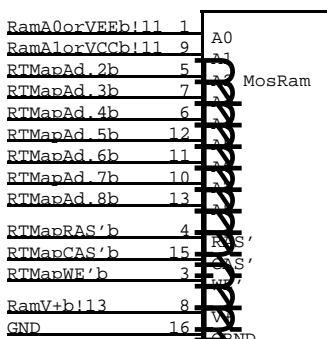
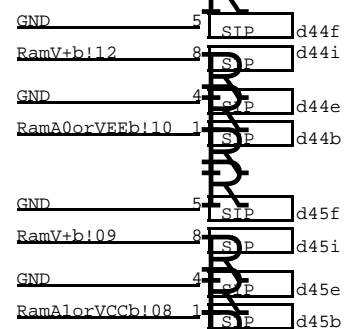
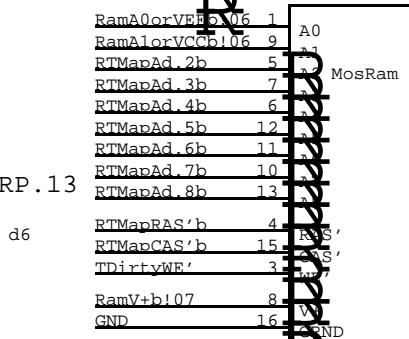
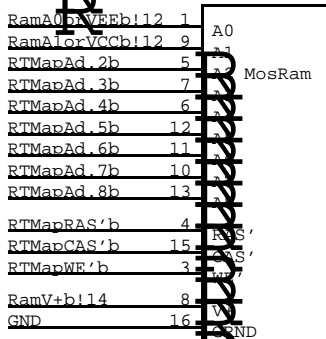
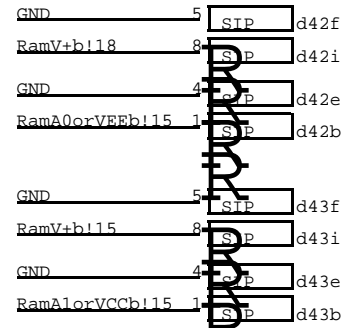
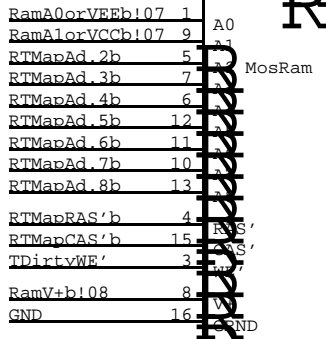
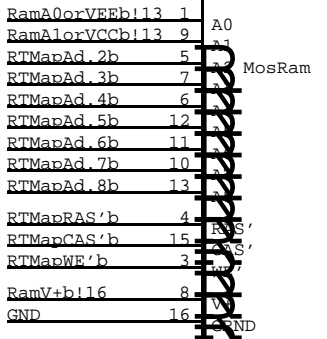
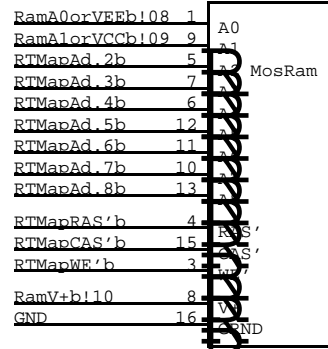
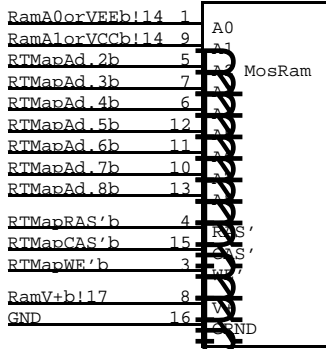
a13

Capacitor Options:

16K RAMs: install CAPS between pins 5 and 8 AND between pins 1 and 4, b42,b43,b44,b45,b46,b47

64K RAMs: install CAPS between pins 5 and 8 ONLY, b42,b43,b44,b45,b46,b47

256K RAMs: same as 64K.

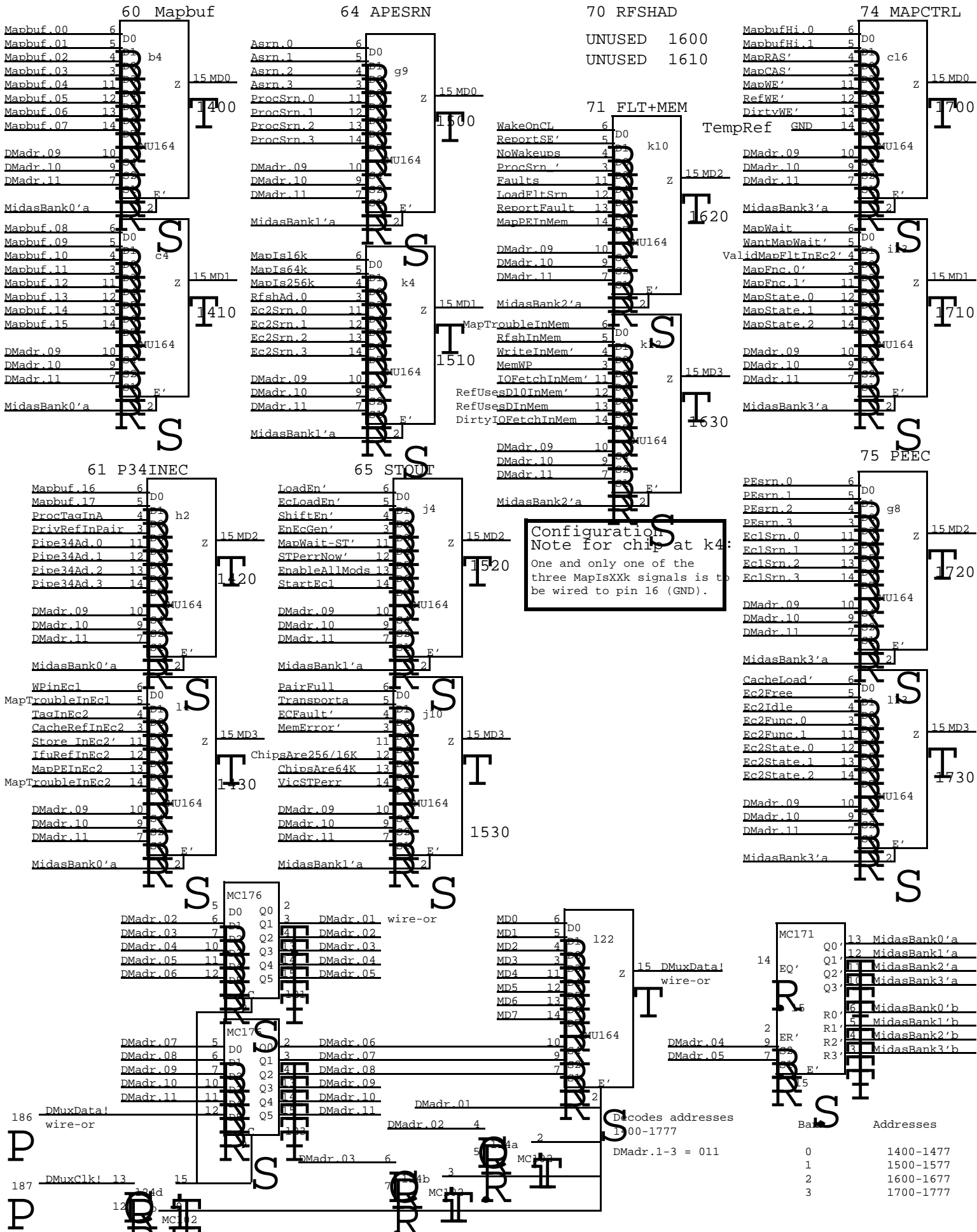


Capacitor Options:

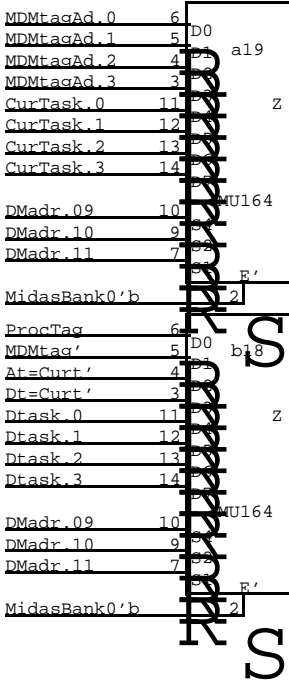
16K RAMs: install CAPS between pins 5 and 8 AND between pins 1 and 4, d42,d43,d44,d45,d46,d47

64K RAMs: install CAPS between pins 5 and 8 ONLY, d42,d43,d44,d45,d46,d47

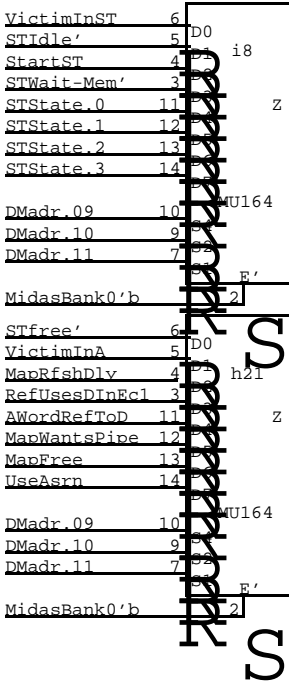
256K RAMs: same as 64K.



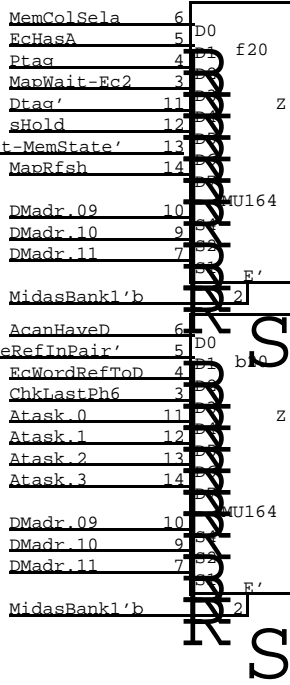
62 MCDTSK



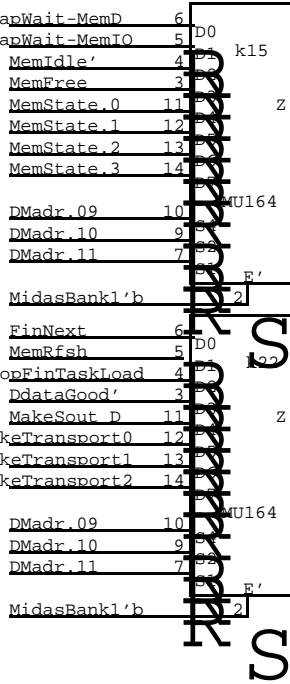
63 ST+A



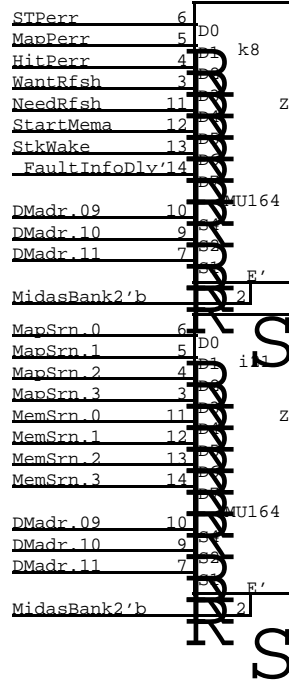
66 TAG+AT



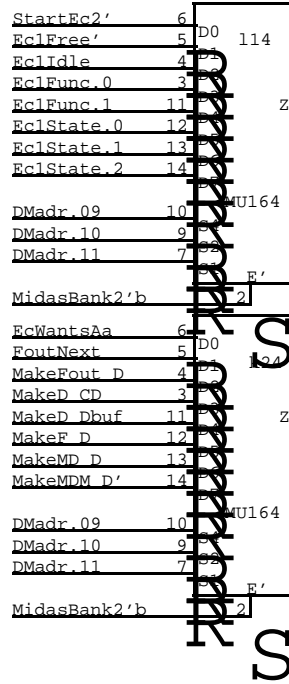
67 MEMST



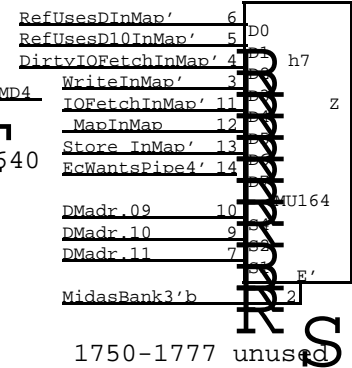
72 RFS+SRN



73 EC1+MAKE



76 INMAP



Bank	Addresses
0	1400-1477
1	1500-1577
2	1600-1677
3	1700-1777

C	a 181	b 168	c 153	d 137	e 124	f 109	93 g	80 h	64 j	48 i	33 k	20 l	B
1	BmuxDrivr 0-1 174 11	Mapbuf 0-1-8-9 175 11	RAS-CAS A,7 231	Mapbuf 2-3-4-10 11-12 175 11	BmuxDrivr 12-13 174 11	Mapbuf 5-6-7-13 14-15 175 11	Mapbuf 16,17 175 11	BMuxEn 11,11 109	ErrorRcvr 4,4,10,D 104	ModSinEn F16 9	HOLDcntr F16 8	ppclk/1PR 210 20	1
2	BmuxDrivr a,4,c,8 102	BmuxDrivr 8-9 174 11	CAS-WE clk1 7,8,20 176 11,11	BmuxDrivr 4-5 174 11	BmuxDrivr 6-7 174 11	BmuxDrivr 14-15 174 11	BmXsel MU 103 11	MU P34INEC 103	ShifInEver 15,9,5,5 102	STOut 5,5,10,10 102	HOLDcntr F16 8	Clock 231 20	2
3	ModEn F16 7	SHIFTS LOADS 103 5,5,5,5	BmuxDrivr 10-11 174 11	BmuxDrivr 2-3 174 11	ModDcode 171 7	ModEna F16 7	ParityIn 170 13	3,3,10,20 104	clk1 8,4,5,6,5, 176 4,4,5	clk0 8,8,10, 176 4,4,5	A,5,13 106	Ec2AuIn 176 10	3
4	2	MU Mapbuf	MU Mapbuf	11	IN 125 13	MCR F16 11	10,10,9, 20,20,4 195	dPipe34Ad 174 3	dPipe34Ad 174 3	MU STOUT	MU MAPEC2	MU P34INEC	4
5	1	IN 125 12	+	12	Mod 174 8	A,11,C 105	RfshCnt F16 15	Pipe34Ad 141 3	STOut 176 5	4,7,11 11,4,4 195	FaultSrn F16 3	MidasBank 171 16	5
6	3	OUT 124 12	CLK 212 20	13	+	+	RfshCnt F16 15	ProcSrn 141 3	CLK 210 20	CLK 210 20	EcSrn=0 FSrn=0 109 3,3	10,10,10 105	6
7	3	+	CLK 210 20	14	OUT 124 13	+	RfshCnt F16 15	MU INMAP	CLK 212 20	CLK 20,B 210	ReportFlt 121 3	Ec1AuIn 176 9	7
8	4	IN 125 13	+	15	+	+	MU PEEC	MemPE F00 4	MU ST+A	srn0,Wrfsh 135 3,4	MU RFS+SRN	Ec1Func 117 9	8
9	5	OUT 125 12	+	PAR 125 12	IN 125 12	+	MU APESRN	Ec1Srn 141 3	Ec2Srn 141 3	dFltCnt NeedRfsh 117 3,4	FaultCnt F16 3	+	9
10	5	+	+	DTYb 124 12	OUT 124 12	+	A,8,8,5 104	PEsrn 158 3	MemAuOut F16 7	MU STOUT	MU FLT+MEM	Ec2Func F16 10	10
11	7	IN 125 12	PLAT BYPASS	DTYa 170 13	PARITY 210 20	CLK 210 20	CLK 210 20	STProm 139 4	7,7,7 105	MemAuIn 176 7	7,7,9 10,10,10 195	Ec2Prom 139 10	11
12	6	OUT 124 12	PLAT SWITCH	REF 170 13	PARITY 212 20	CLK 210 20	CLK 210 20	STState F16 5	STProm 139 5	MemAuOut F16 7	MU FLT+MEM	Ec1Prom 139 9	12
13	9	PLAT BYPASS	WE 125 14	WP 145 13	ADDR 125 15	Pipe34 145 13	CheckWP 6,6,6,7 104	MapAu 7,B,6,6 103	MU MAPCTRL	MemProm 139 7	MemAuIn 176 7	MU PEEC	13
14	7	PLAT SWITCH	ADDR 125 15	RESISTOR 14	ADDR 125 15	Pipe34 145 12	MapTrbl 121 13	MapAuIn 176 6	MapProm 139 6	MemProm 139 7	clk0 7,7,,,,, 176	MU EC1+MAKE	14
15	+	RasCas 125 14	ADDR 125 15	RESISTOR 14	ADDR 125 14	Pipe34 145 12	MapProm 139 14	MapState F16 6	MapAuIn 176 6	STIdle MapWait 117 7,5	MU MEMST	+	15
16	+	+	MU MAPCTRL	RESISTOR 14	Pipe34 145 13	Pipe34 145 12	_Map 105 14	MapAdMux 158 15	MapAdMux 158 15	MemState F16 7	13,10,5,9 103	Ec2State F16 10	16
17	Pair 2,,, F16	+	ASRNInc 3,20 210	+	OUT 124 13	Pipe34 145 13	+	MapAdMux 158 15	MapWait 121 6	VicSTPerr 10,9,C,4 104	Ec1State F16 9	6,10,10 104 10	17
18	4	MU MCPTSK	CLK 212 20	clk1 14,12,, 176	MapRCW F16 14	MapRCW F16 14	MemCad 173 12	MemSrn 141 3	CLK 210 20	CLK 210 20	VicInSt MapWaitD 109 6,9	1,1,9,6 104	18
19	MU MCPTSK	TrueNext 158 2	CLK 210 20	CLK 210 20	clk0 14,1,1,4 176 f,g	ASRN F16 3	MemCad 173 12	StartST 121 5	CLK 210 20	CLK 210 20	clk0 5*10,9 176	10,9,9 9,6,1 195	19
20	5	MU TAG+AT	CLK 210 20	Tag211 211 1,B	ProcTag F145	MU TAG+AT	12,20,C, 103 13	A,4,4 105	MapAu 104 6	Ec1AuOut 176 9	Ec1AuOut 176 9	10,9,9 9,6,1 195	20
21	Atask F16 2	Dt=Ct 113 1	PipeTask Pipe2 F145 2	PipeTaska 197 2	PipeSubT Pipe2 F145 2	MDMtag F145 1	dMDMad 159 1	MU ST+A	MU RFS+SRN	Ec2AuOut F16 10	Ec2AuOut F16 10	Midas 176 18	21
22	5	MDMtagAd 158 1	At=Ct 113 1	FinTask F16 2	MDMtagAd 141 1	DTask F16 2	MakeMD_D 117 1	dMDMad 100 1	MapSRN 141 3	STOut 176 5	MapAUOut F16 6	MU MEMST	Midas 164 18
23	ASubTask 2,2,1,1 F16	CurSubT 2,2,d,e 141	RptCur 4,4,4,1 1662	MemAd 0 159 12	CAS-WE 1,7,8,4,5,4 176 clk1	1,6,1,D 103	clk0 1,3,11,e 176 5,5	dPipe02Ad 158 3	Hit 1,1,1 105	Transport 212 5,5	5,B,1,5 102	Midas 176 18	23
24	5	FinSubT 2,2,d,e F16	HOLD 1660 4	MemAd 1-4 159 12	RAS-CAS A,7 231	CacheRef 1,1,4 1674	Page1 4,1,2 105	Pipe02Ad 141 3	MemAd 5-8 159 12	FoutTask 176 2	MU EC1+MAKE	Midas 18,18, 102 C,18	24

Location h11					Location l12					Clean	Dirty
ST		11 cycles		12 cycles		Write		CacheLoad	or Miss	Hit	
ADDR		IO Store		VictimStore		0 - 7		10 - 17	IOFetch	IOFetch	
STATE		0-13	14-17	20- 33	34-37	0 - 7		10 - 17	20 - 27	30 - 37	
Q		012345678910	11	01234567891011		Q	EC1	01234567	01234567	01234567	01234567
0	prepreLoadEn'	00000001110	0 1	00000000111 1	1	0	preEcEn'	01111111	01111111	01111111	01111111
1	preEcLoadEn'	10101010111	1 1	11010101011 1	1	1	preMakeTrspt2	00000000	00111100	00000000	00111100
2	preShiftEn'	00000001111	1 1	10000000111 1	1	2	preMakeFout_D	00000000	00000000	00000000	00011100
3	preEnEcGen'	11111111110	0 1	01111111111 1	1	3	preFoutNext	00000000	00000000	00011100	00001100
4	preMapWait-ST'	00001111110	0 1	0000111111 0	1	4	preMakeD_CD	00000000	00111100	00000000	00000000
5	preSTfree'	11111111101	1 1	11111111110 1	1	5	preEcWantsA	01000000	01000000	01000000	01000000
6	prepreSTPerrNow'	00000000111	1 1	10000000011 1	1	6	preEc1Free'	11111101	11111101	11111101	11111101
7	preStopFinTaskLd	11111100001	1 0	11111110000 0	0	7	preStartEc2'	11111011	11111011	11111011	11111011

Location il4					
MAP		Refresh	Read	Write	Map write
ADDR		0 - 7	10 - 17	20 - 27	30 - 37
STATE		01234567	01234567	01234567	01234567
Q		01234567	01234567	01234567	01234567
0	prepreRfshInMem	00010000	00000000	00000000	00000000
1	UNUSED				
2	preStartMem'	11101111	11101111	11101111	11111111
3	preStartMem	00010000	00010000	00010000	00000000
4	preMapFree'	11111101	11111101	11111101	11111101
5	preMapWantsPipe	00000000	00110000	00110000	00110000

Location l11					
EC2		Write	CacheLoad	CLEAN	Dirty
ADDR		0 - 7	10 - 17	or Miss	Hit
STATE		0 - 7	10 - 17	IOFetch	IOFetch
Q		01234567	01234567	01234567	01234567
0	prepreMakeTnspt1	00000000	11100001	00000000	11100001
1	preMakeFout_D	00000000	00000000	00000000	11110001
2	preFoutNext	00000000	00000000	11110001	11111001
3	preMakeD_CD	00000000	11100001	00000000	00000000
4	preEc2Free	00000010	00000010	00000010	00000010
5	preEcWantsPipe4	10000000	10000000	10000000	10000000
6	preMapWait-Ec2	00000000	11110001	00000000	11110001
7	CacheLoad'	11111111	00000010	11111111	11111111

IDENTICAL

Location g15					
MAP		Refresh	Read	Write	Map Write
ADDR		0 - 7	10 - 17	20 - 27	30 - 37
STATE		01234567	01234567	01234567	01234567
Q		01234567	01234567	01234567	01234567
0	preMemRfsh	00011110	00000000	00000000	00000000
1	preRAS1'	00000010	00000010	00000010	00000010
2	preRAS2'	00001000	00001000	00001000	00001000
3	prepreCAS1'	11111110	00000010	00000010	00000010
4	prepreCAS2'	11111100	00000000	00000000	00000000
5	preMapWE'	11111111	11111111	11111111	11100111
6	preRefWE'	11100111	11100111	11100111	11100111
7	preDirtyWE'	11111111	11111111	11100111	11100111

Location j13					
MEM		READ	WRITE	REFRESH	
ADDR		0 - 7	0 - 7	10-17	20 - 27
STATE		01234567	01234567	10-17	30-37
Q	16K chips	01234567	01234567	10-17	30-37
0	prepreMemWE'	10011111	10011111	1	11111111 1
1	preSTWait-Mem'	10111111	10111111	1	11111111 1
2	prepreMemCAS	11111100	11111100	0	00000000 0
3	prepreShiftLoadEven	00010000	00010000	0	00000000 0
4	prepreMCS	11111000	11111000	0	00000000 0
5	preMemFree	00000010	00000010	0	00000010 0
6	preMemState6'	11111011	11111011	1	11111011 1
7	preMemRAS	11110001	11110001	1	11110001 1

IDENTICAL

Location j14																
MEM		READ				WRITE				REFRESH						
ADDR		0 - 14				15-17				20 - 34		35-37				
STATE		0123456789	10	11	12	13-15	0123456789	10	11	12	13-15	0123456789	10	11	12	13-15
Q	4K chips	0123456789	10	11	12	13-15	0123456789	10	11	12	13-15	0123456789	10	11	12	13-15
0	prepreMemWE'	1100000111	1	1	1	1	1100000111	1	1	1	1	1111111111	1	1	1	1
1	preSTWait-Mem'	1111101111	1	1	1	1	1111101111	1	1	1	1	1111111111	1	1	1	1
2	prepreMemCAS	0001111100	0	0	0	0	0001111100	0	0	0	0	0001111100	0	0	0	0
3	prepreShiftLoadEven	0000000100	0	0	0	0	0000000100	0	0	0	0	0000000000	0	0	0	0
4	prepreMCS	0111111000	0	0	0	0	0111111000	0	0	0	0	0000000000	0	0	0	0
5	preMemFree	0000000000	0	1	0	1	0000000000	0	1	0	1	0000000000	0	1	0	1
6	prepreState7'	1111111111	1	1	1	1	1111111111	1	1	1	1	1111111111	1	1	1	1
7	preMemRAS	1111111100	0	0	1	0	1111111100	0	0	1	0	1111111100	0	0	1	0

IDENTICAL

Dorado Memory Extension Board
 Stuffing and Configuration Instructions

1. If 256 chips are NOT installed in the Memory Storage Arrays (MSA), then
 - a. Break g10.6 (MC104) before stuffing. Label this chip as MemX-g10.

2. Break h20.10 before stuffing. Label this chip as MemX-h20.
 All SG10139 chips are PROMS which must be blown and labeled before stuffing.
 Stuff all PLAT1816 with Beckman type 898-3-R27 resistor packs or equivalent.

3. Do all the actions in one of the following three columns:

Action	16K Map Chips	64K Map Chips	256K Map Chips
Blue Wire	k4.6 to k4.16	k4.5 to k4.16	k4.4 to k4.16
Stuff into MosRam sockets	MK4116-2 or equivalent	TMS4156 or equivalent	256K Ram or equivalent
Jumper wires in socket b14	3 to 14 5 to 12 7 to 10	6 to 11 4 to 13	6 to 11 4 to 13 2 to 15
Jumper wires in socket c12	3 to 14 5 to 12 7 to 10	6 to 11 4 to 13	6 to 11 4 to 13 2 to 15
capacitors in SIP sockets b42,b43,b44,b45,b46,b47 d42,d43,d44,d45,d46,d47	0.1 mmf pins 5 to 8 each socket AND 0.1 mmf pins 1 to 4 each socket	0.1 mmf pins 5 to 8 each socket	0.1 mmf pins 5 to 8 each socket
1 KOhm resistor in socket b14 and 1 KOhm resistor in socket c12	NO	1 to 15 IF Ram has self refresh feature- Motorola or equivalent	NO
22 uf Cap in PLAT c11 & b13	+ 2 to 15 +4 to 14 +11 to 6	+ 3 to 14 + 5 to 12	+ 3 to 14 + 5 to 12

XEROX PARC	Project Dorado	Reference MultiWire Changes	File MemX24.sil	Designer K. Pier	Rev Ch	Date 6/23/79	Page 24
---------------	-------------------	--------------------------------	--------------------	---------------------	-----------	-----------------	------------