

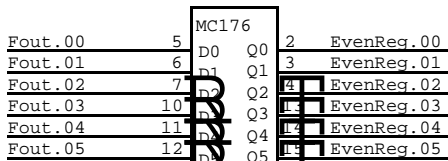
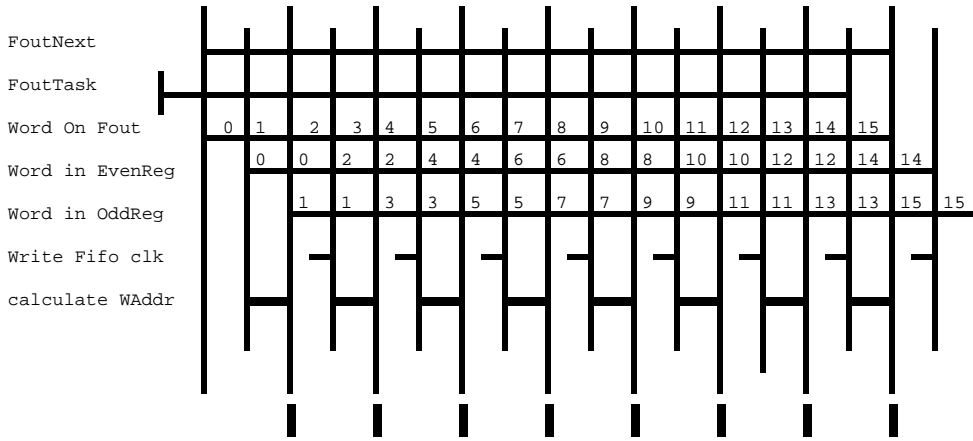
D O R A D O S C H E M A T I C S

D i s p l a y Y

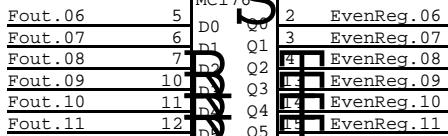
Table of contents

<u>TITLE</u>	<u>Page</u>
FOUT interface	01
Pointers	02
Fifo Address Logic	03
Fifo	04
Item Permutation Logic	05
Intermediate Buffer Control	06
Channel A Data Path	07
Channel A Items and Shift/Load Control	08
Channel A Current Line Control Block	09
Channel B Data Path	10
Channel B Items and Shift/Load Control	11
Channel B Current Line Control Block	12
Next Line Control Block	13
Horizontal RAM	14
Flags	15
DWT Task WakeUp Logic	16
Cursor	17
Slow IO Interface	18
Alto Display Drivers	19
MiniMixer	20
OIS Terminal Interface	21
Statics Register and Muffler Control	22
Mufflers	23
Clock Drivers	24
Pre Clock Drivers	25
Layout	26
Slow IO System Formats	27
Next Line Control Block Formats	28
DDC to DDM Interface Table	29
Cabling Summary	30
Configuration	31

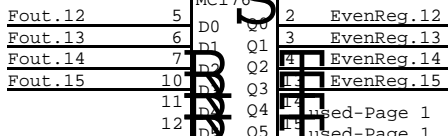
CLK: 0 1 0 1 . . .



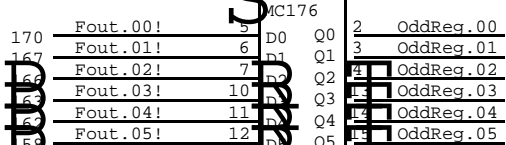
clk1'Da



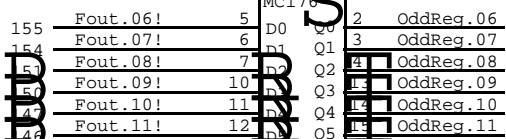
clk1'Da



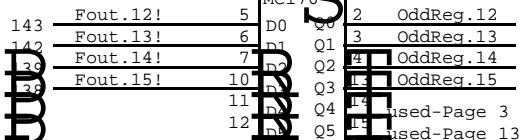
clk1'Db



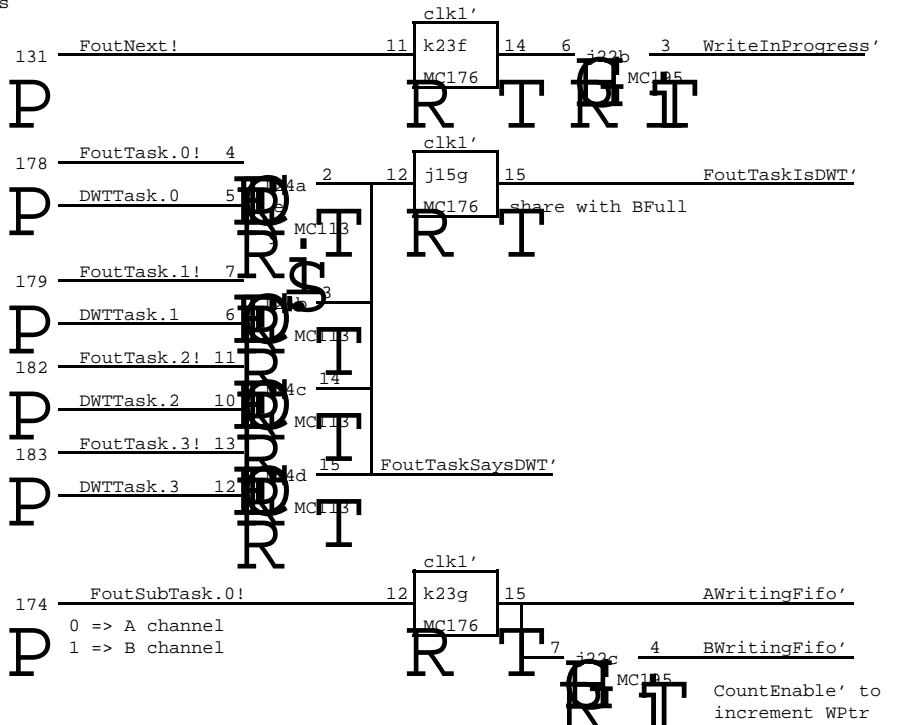
clk0'Da



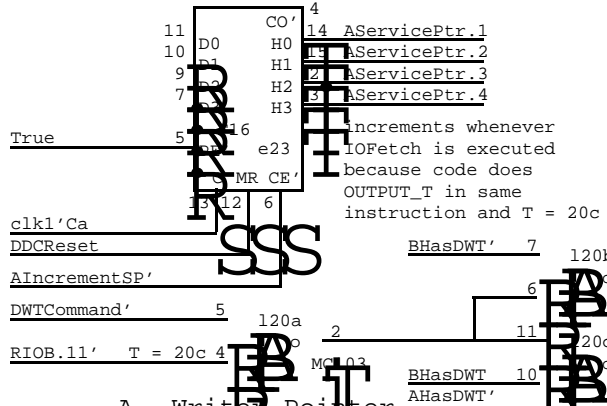
clk0'Db



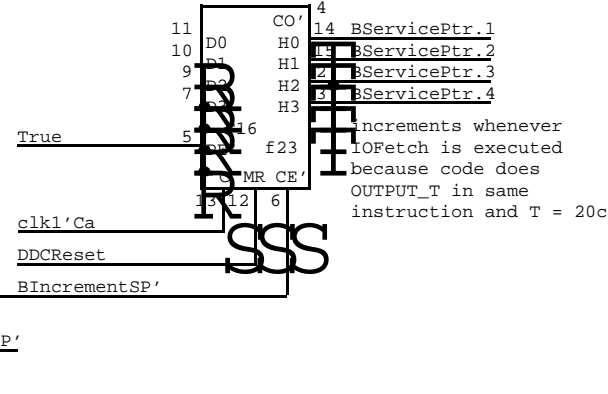
clk0'Dc



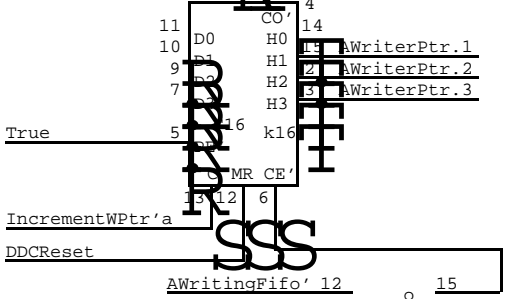
A Service Pointer



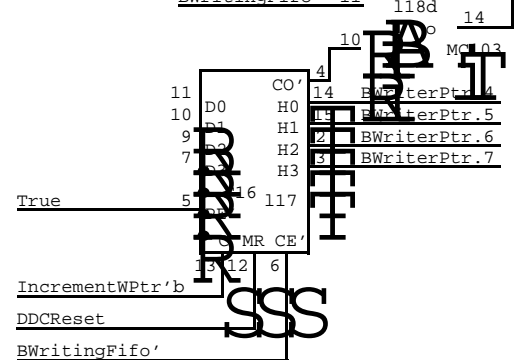
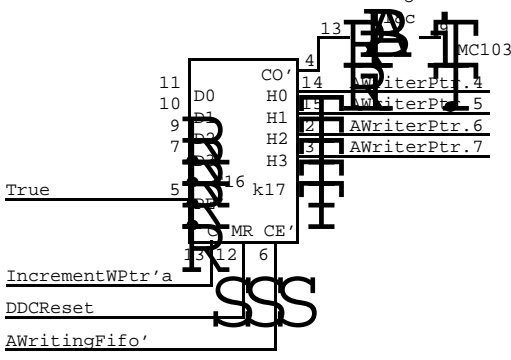
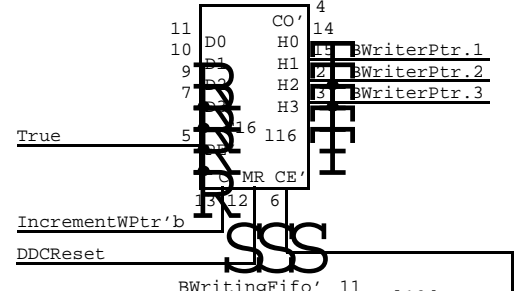
B Service Pointer



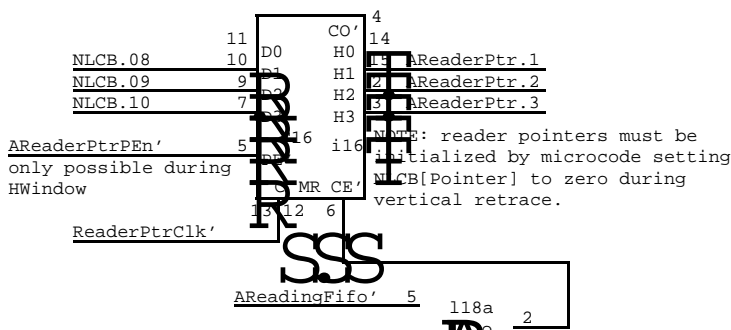
A Writer Pointer



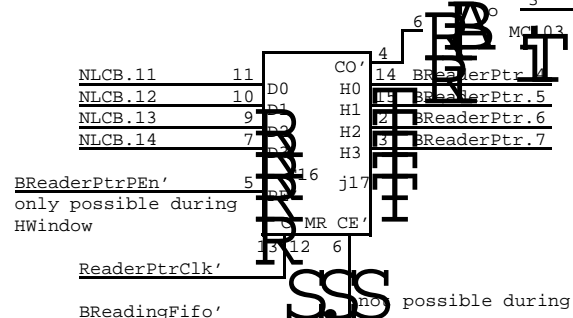
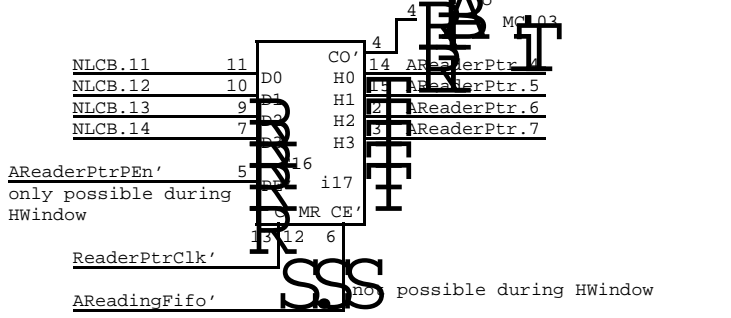
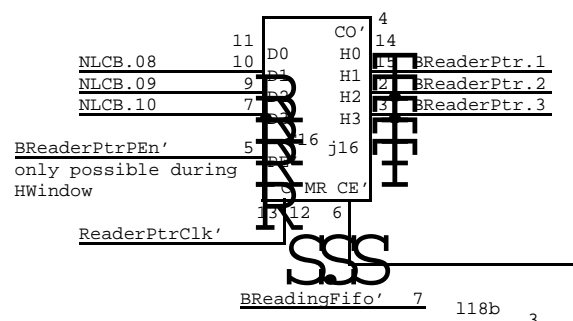
B Writer Pointer

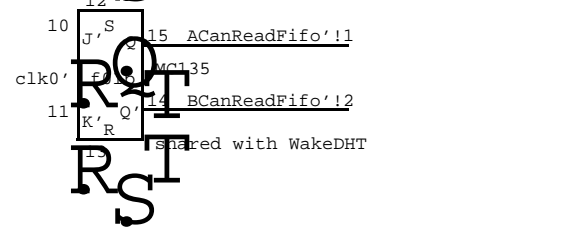
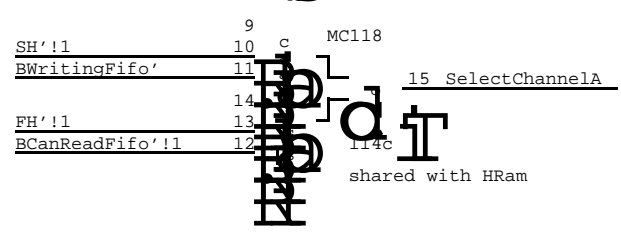
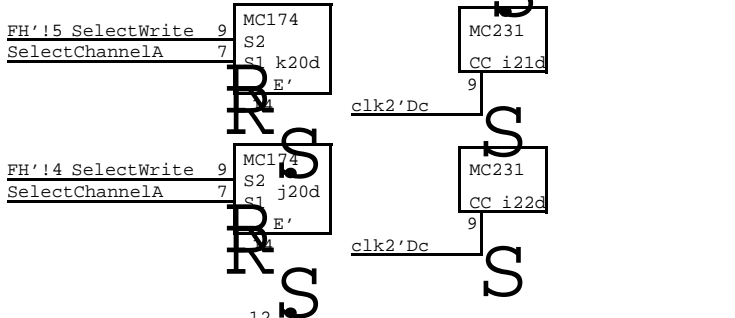
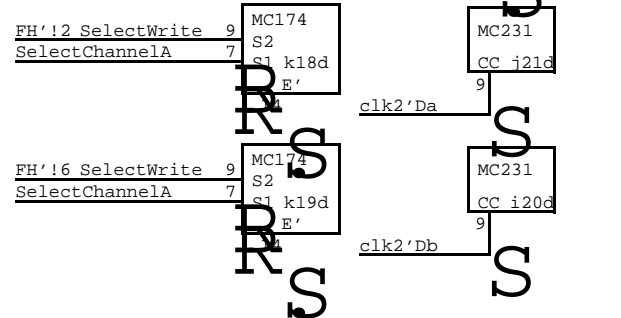
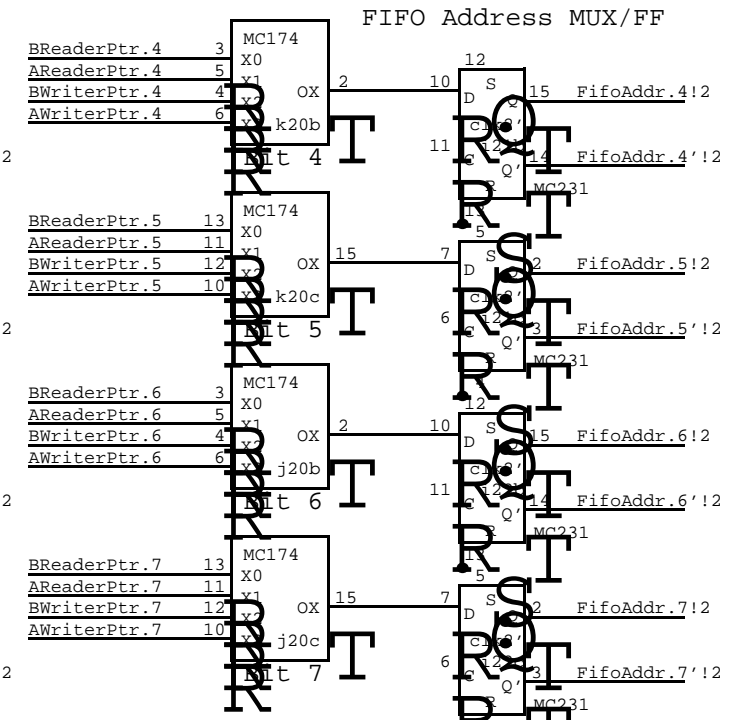
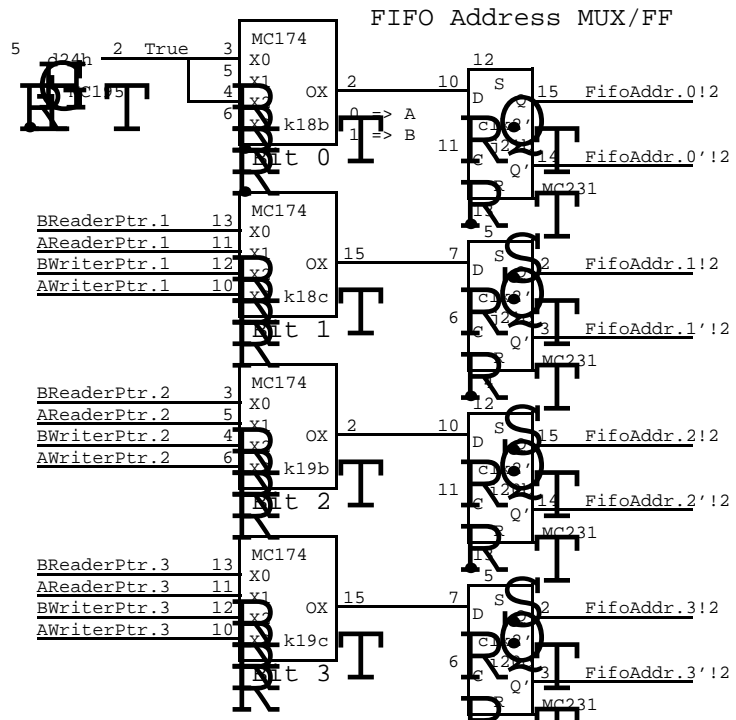


A Reader Pointer



B Reader Pointer





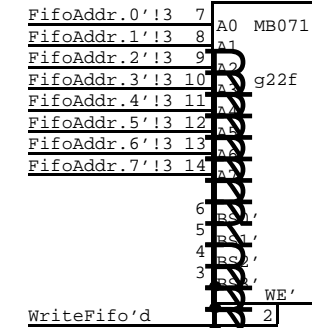
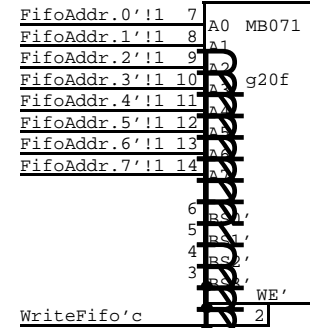
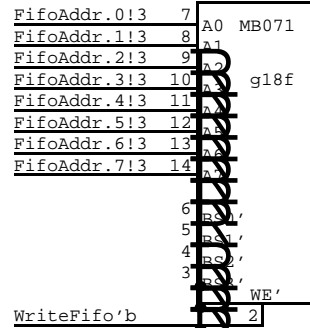
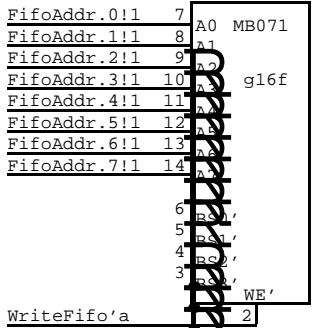
Write In FH
Read In SH

EvenReq.00	19	g16b	23	Fifo.00
		MB071		
EvenReq.01	18	g16c	23	Fifo.01
		MB071		
EvenReq.02	17	g16d	23	Fifo.02
		MB071		
EvenReq.03	16	g16e	20	Fifo.03
		MB071		
EvenReq.04	19	g18b	23	Fifo.04
		MB071		
EvenReq.05	18	g18c	23	Fifo.05
		MB071		
EvenReq.06	17	g18d	23	Fifo.06
		MB071		
EvenReq.07	16	g18e	23	Fifo.07
		MB071		
EvenReq.08	19	g20b	23	Fifo.08
		MB071		
EvenReq.09	18	g20c	23	Fifo.09
		MB071		
EvenReq.10	17	g20d	23	Fifo.10
		MB071		
EvenReq.11	16	g20e	23	Fifo.11
		MB071		
EvenReq.12	19	g22b	23	Fifo.12
		MB071		
EvenReq.13	18	g22c	23	Fifo.13
		MB071		
EvenReq.14	17	g22d	23	Fifo.14
		MB071		
EvenReq.15	16	g22e	23	Fifo.15
		MB071		

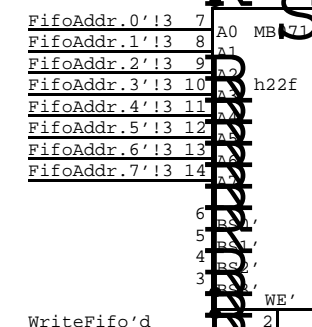
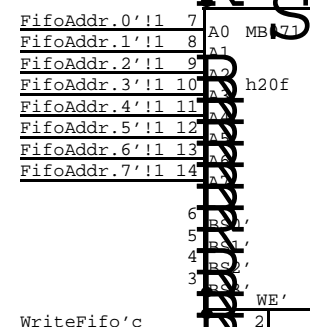
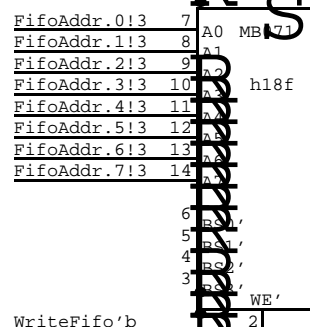
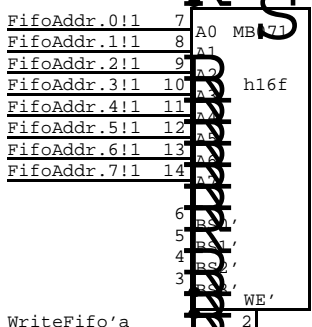
EVEN WORDS

OddReq.00	19	h16b	23	Fifo.16
		MB071		
OddReq.01	18	h16c	23	Fifo.17
		MB071		
OddReq.02	17	h16d	23	Fifo.18
		MB071		
OddReq.03	16	h16e	20	Fifo.19
		MB071		
OddReq.04	19	h18b	23	Fifo.20
		MB071		
OddReq.05	18	h18c	23	Fifo.21
		MB071		
OddReq.06	17	h18d	23	Fifo.22
		MB071		
OddReq.07	16	h18e	23	Fifo.23
		MB071		
OddReq.08	19	h20b	23	Fifo.24
		MB071		
OddReq.09	18	h20c	23	Fifo.25
		MB071		
OddReq.10	17	h20d	23	Fifo.26
		MB071		
OddReq.11	16	h20e	23	Fifo.27
		MB071		
OddReq.12	19	h22b	23	Fifo.28
		MB071		
OddReq.13	18	h22c	23	Fifo.29
		MB071		
OddReq.14	17	h22d	23	Fifo.30
		MB071		
OddReq.15	16	h22e	23	Fifo.31
		MB071		

ODD WORDS

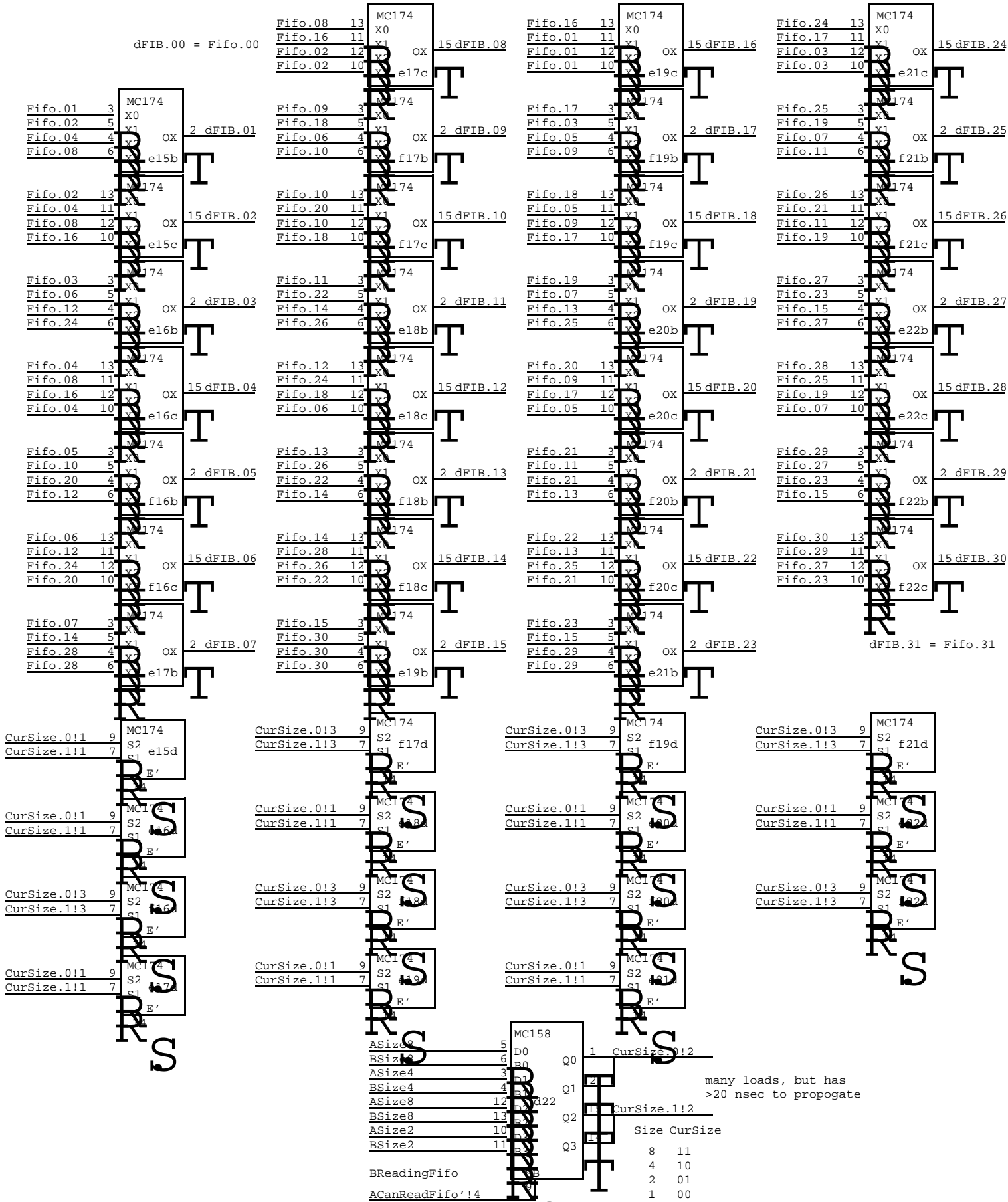


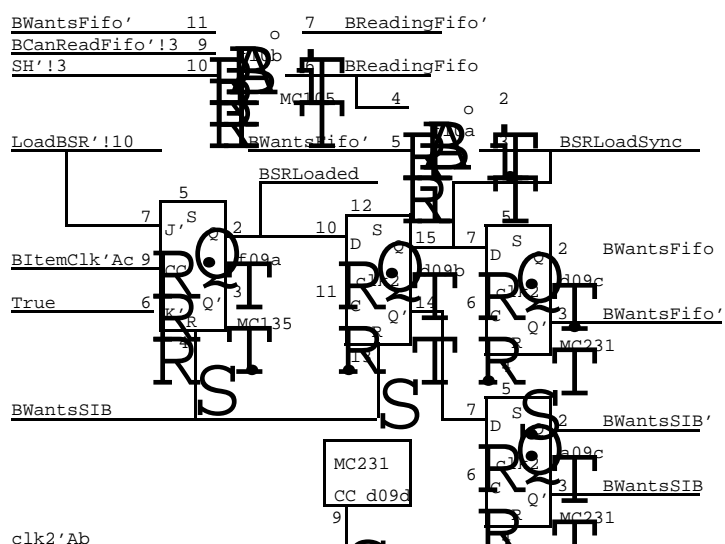
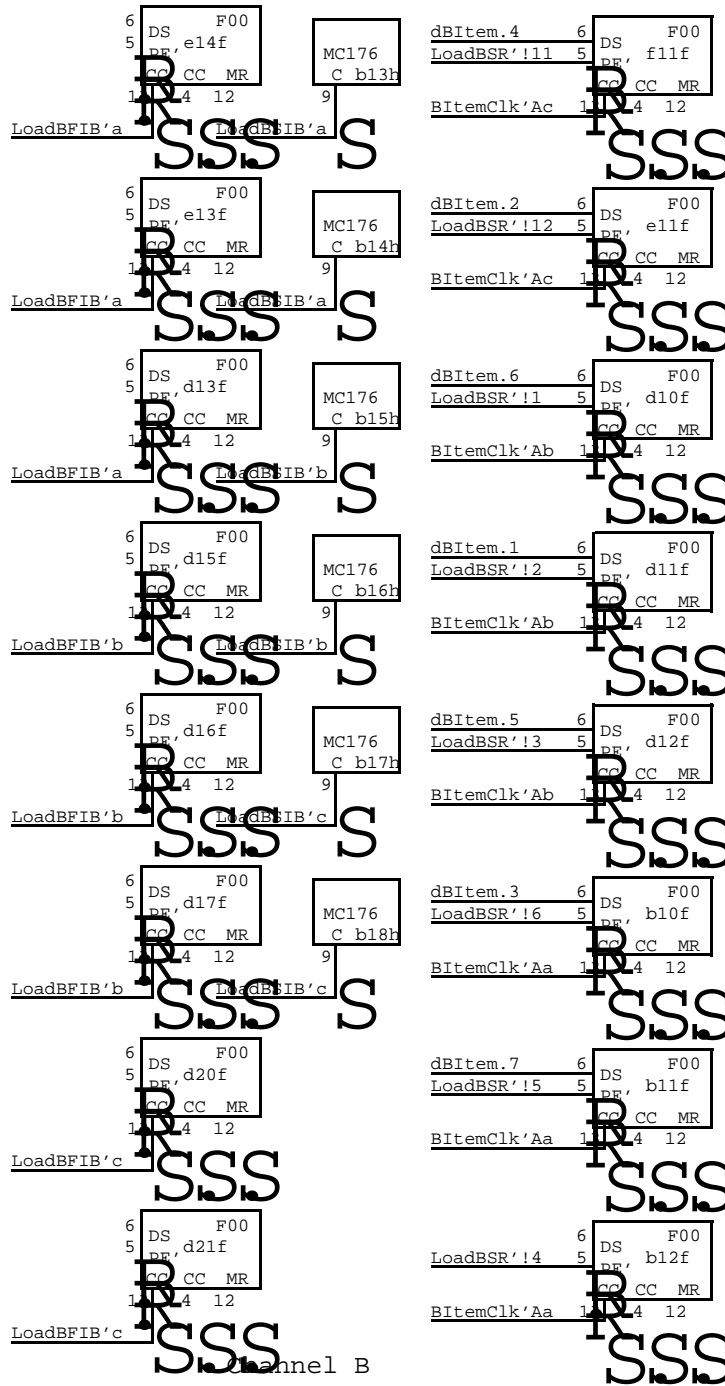
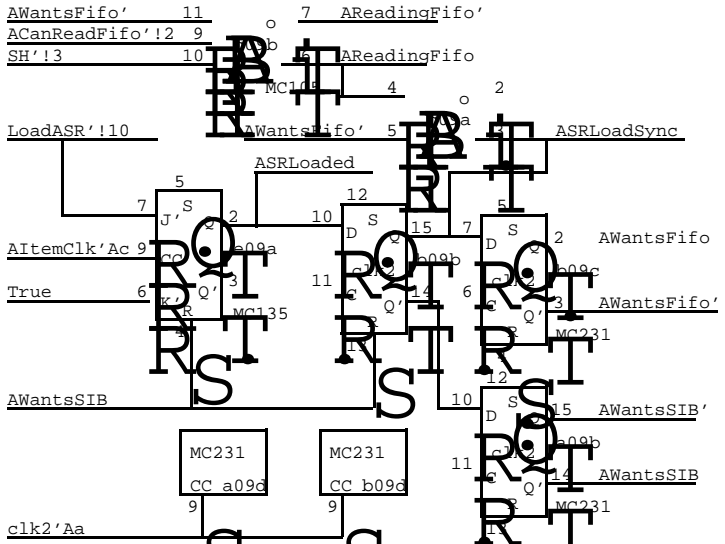
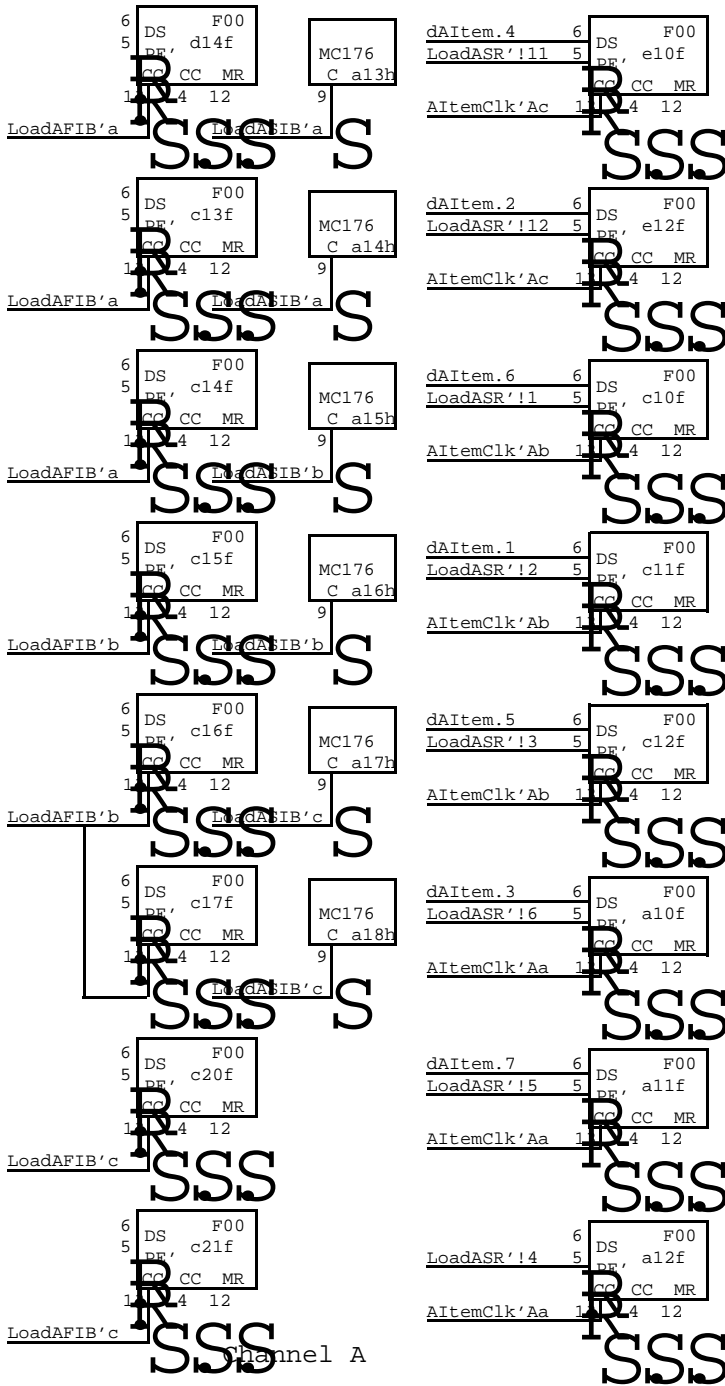
EVEN WORDS

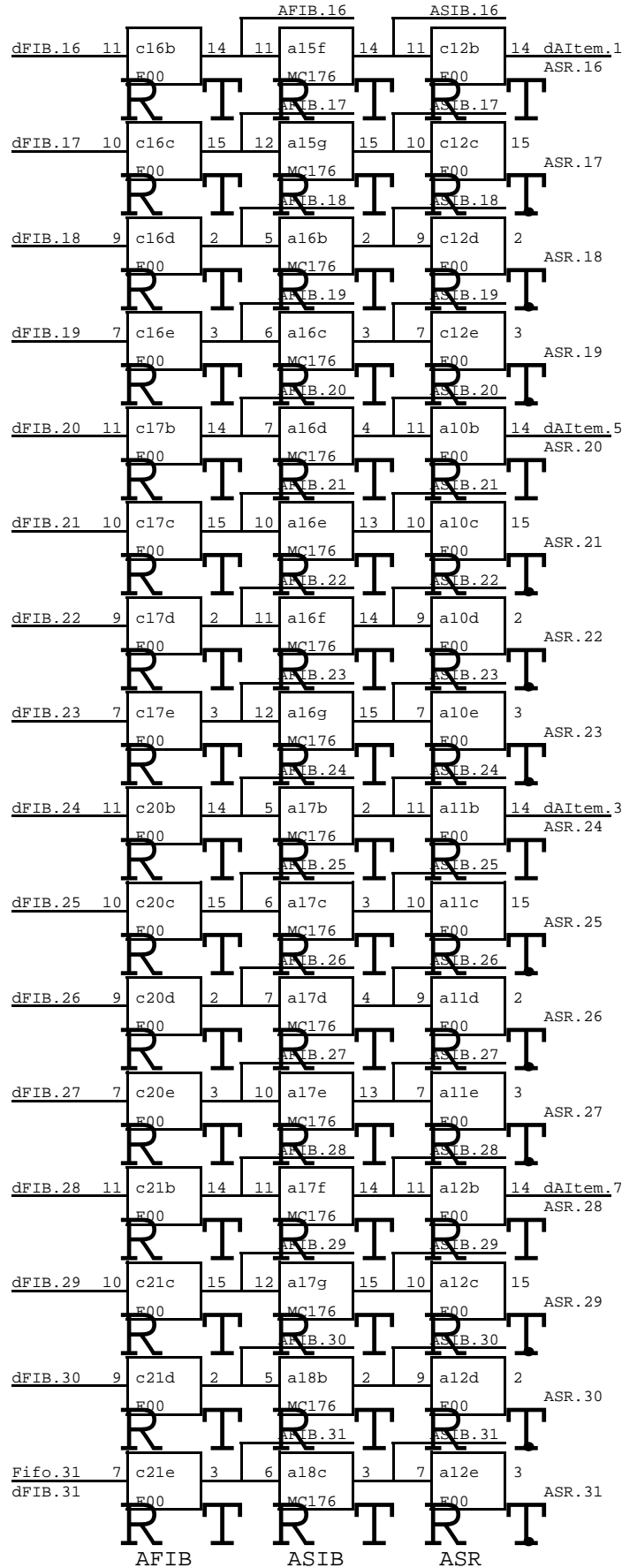
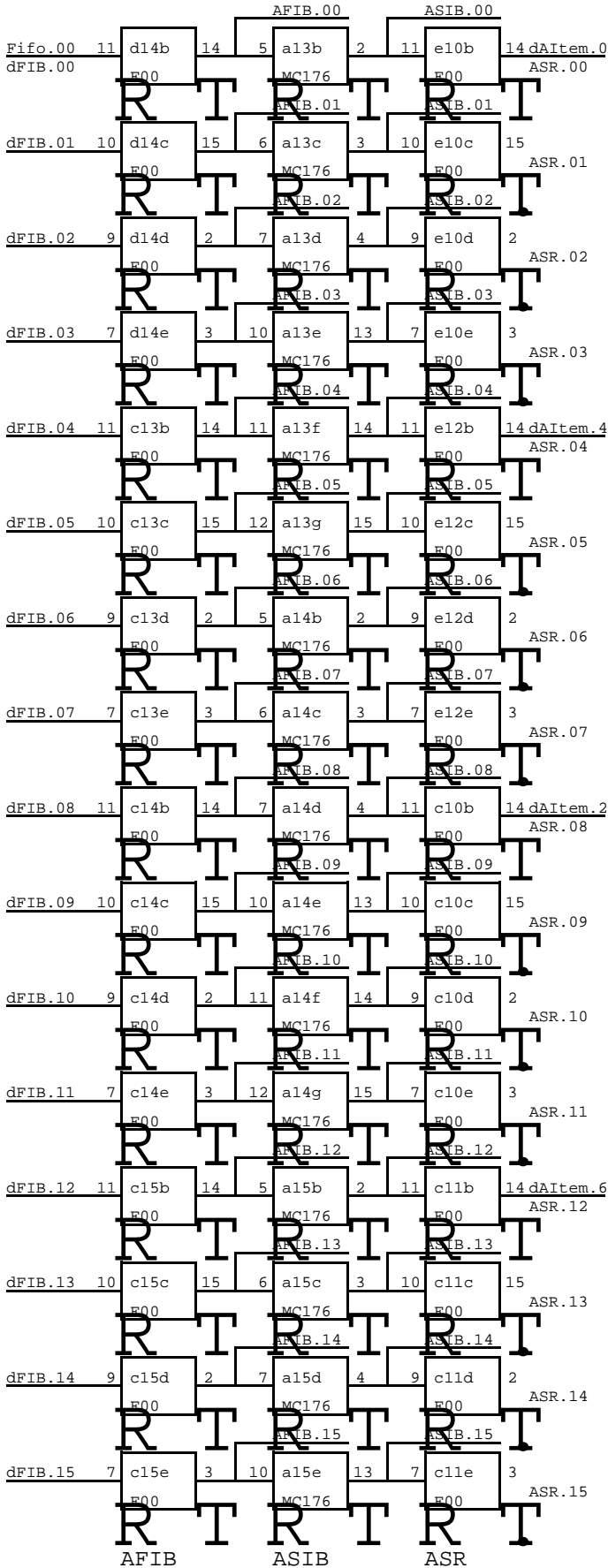


ODD WORDS

Item Generator Permuter
See ItemGenerator table for input specs

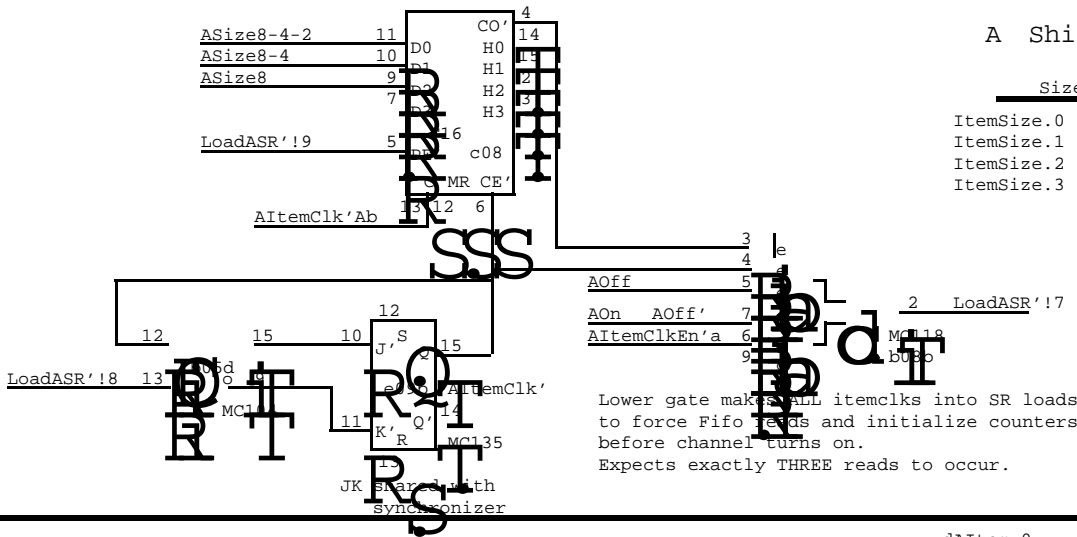






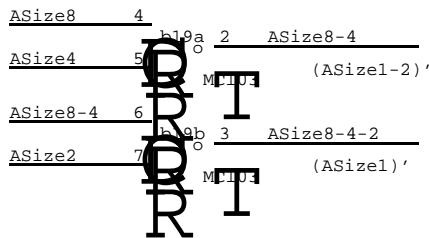
A Shift Register Control

Size	8	4	2	1
ItemSize.0	1	1	1	0
ItemSize.1	1	1	0	0
ItemSize.2	1	0	0	0
ItemSize.3	0	0	0	0

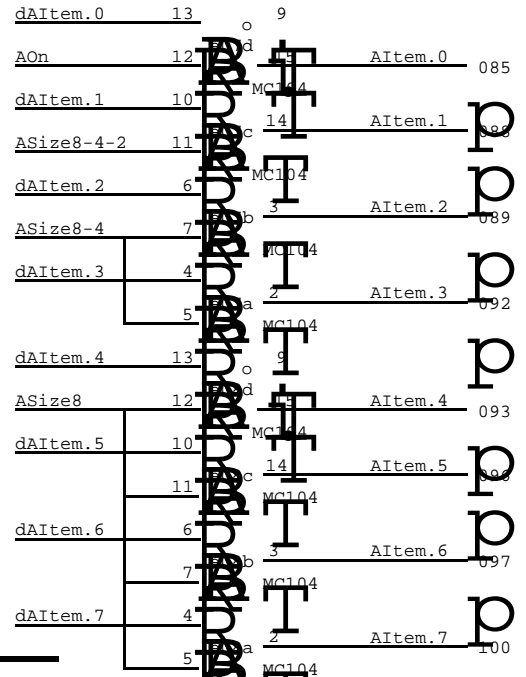


Lower gate makes ALL itemclks into SR loads to force Fifo reads and initialize counters before channel turns on.
 Expects exactly THREE reads to occur.

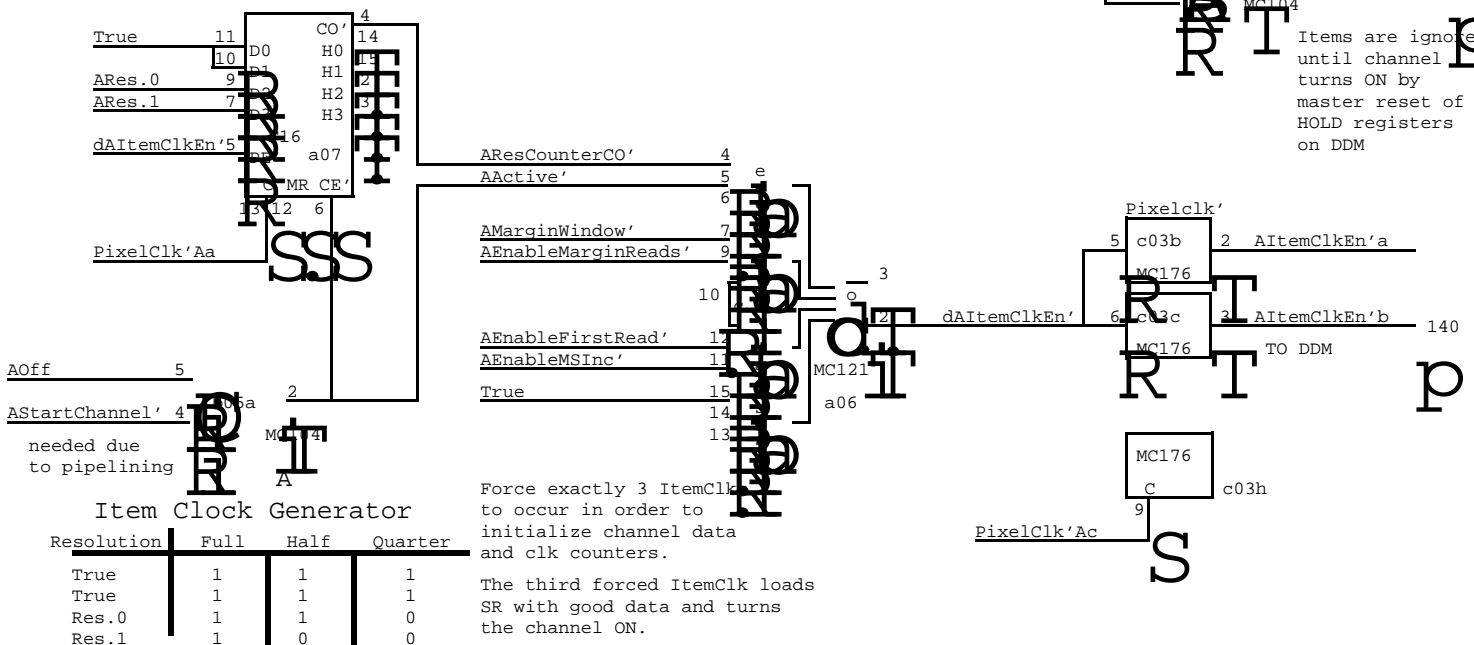
A Item buffer logic



dItems are garbage until channel becomes active



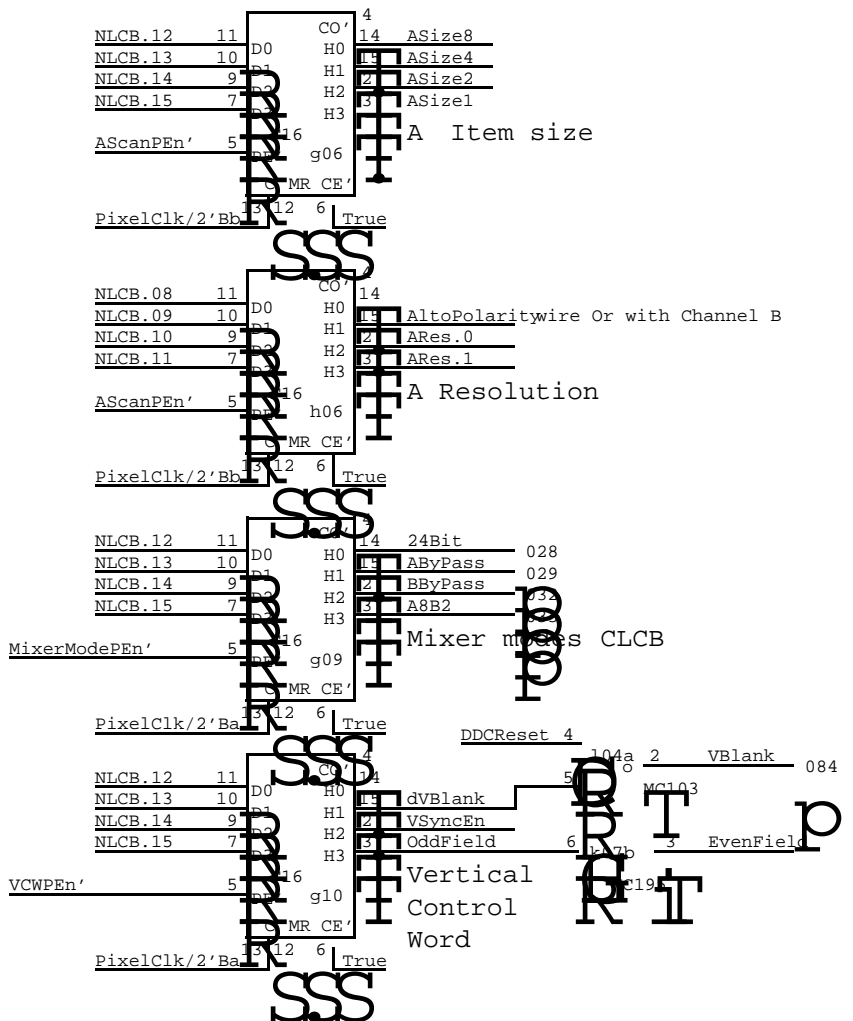
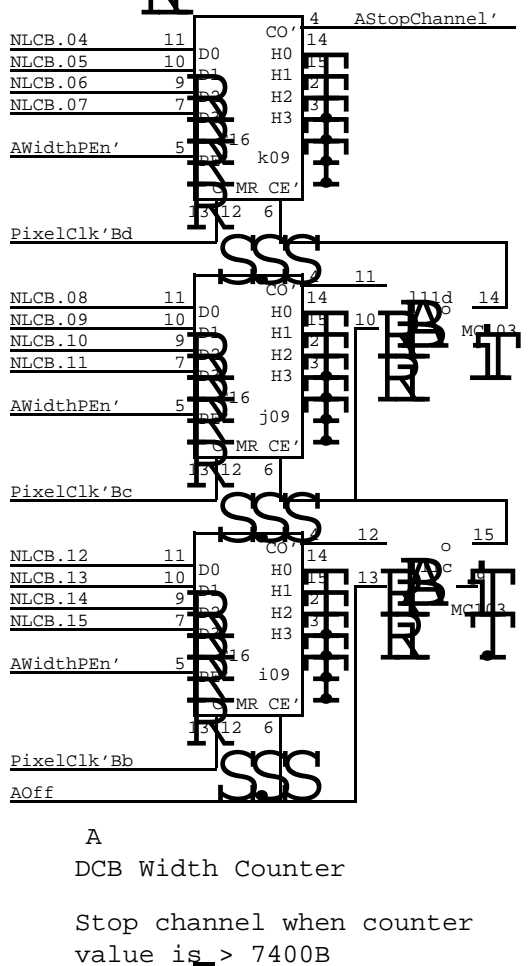
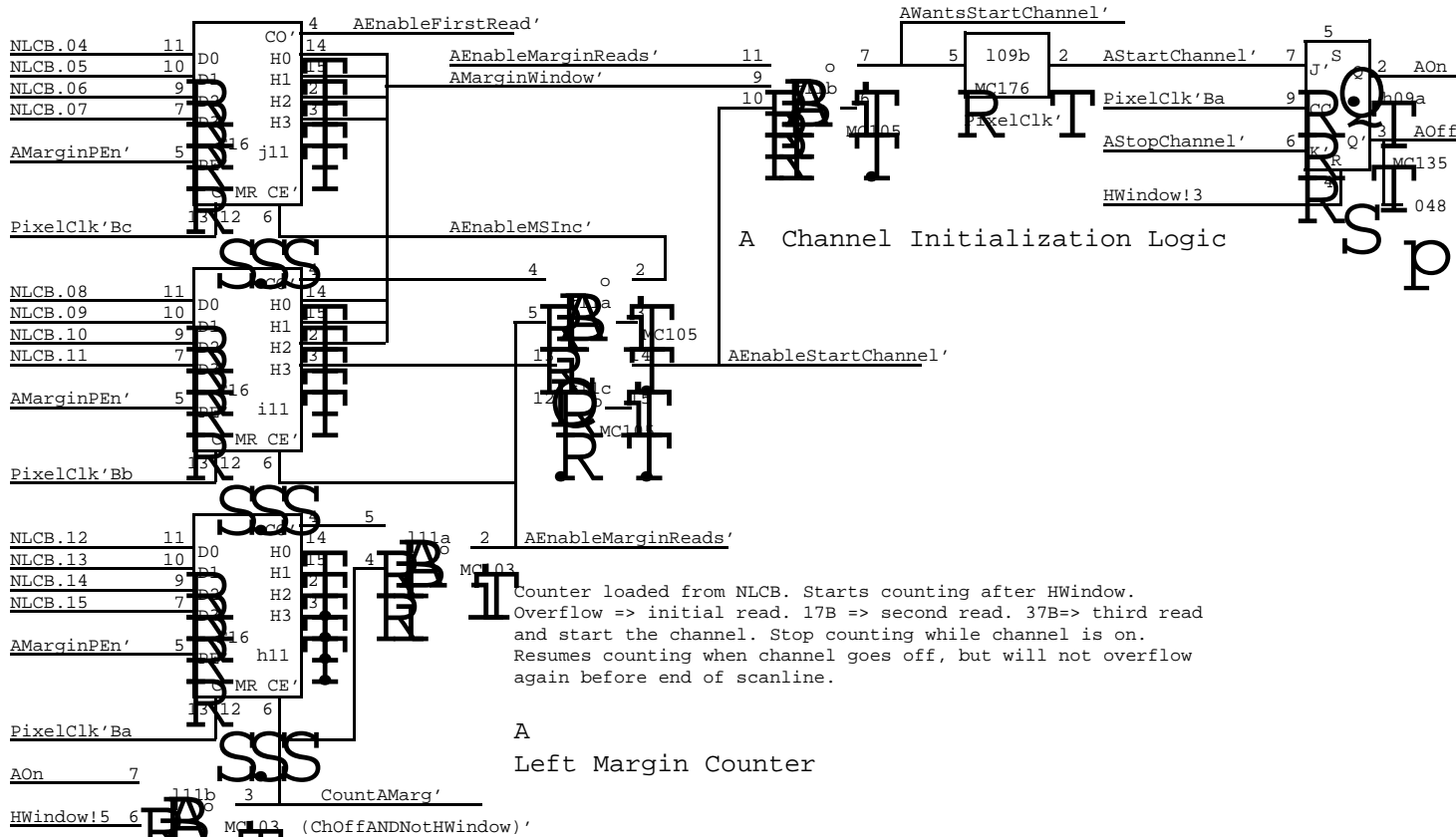
Items are ignored until channel turns ON by master reset of HOLD registers on DDM

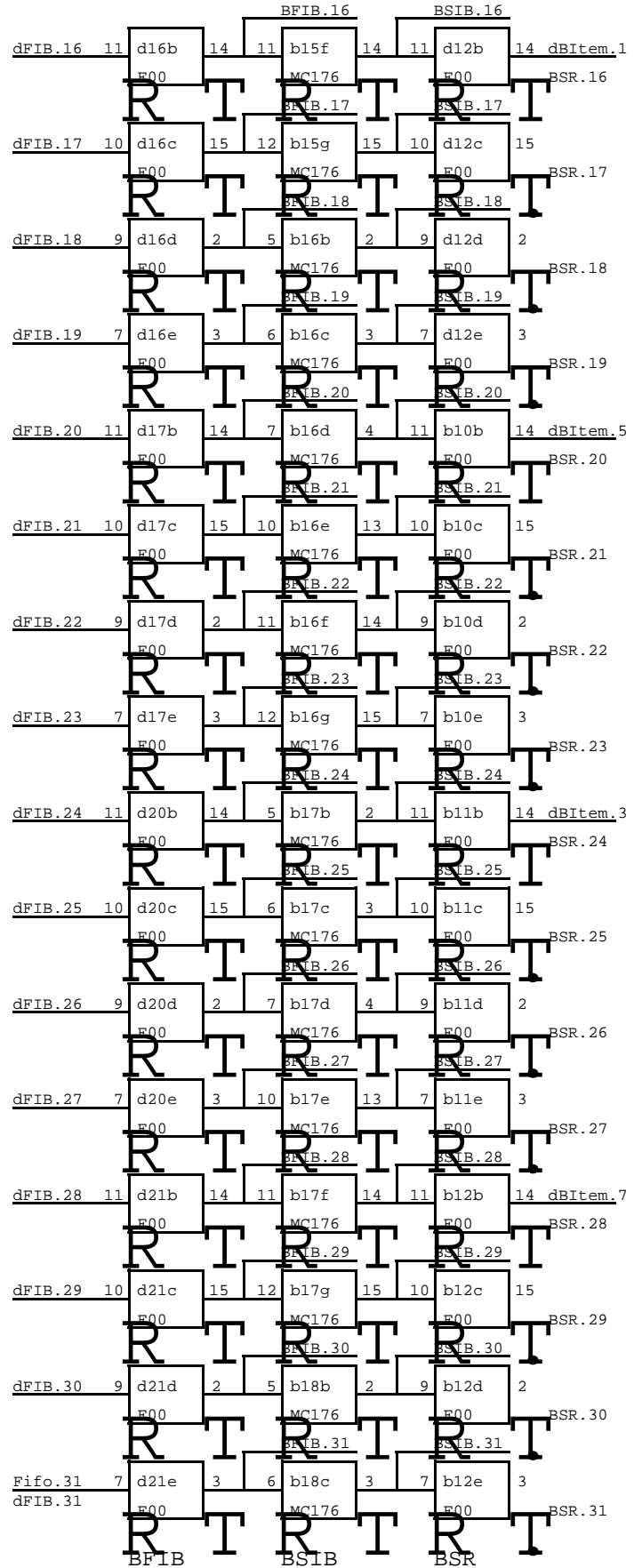
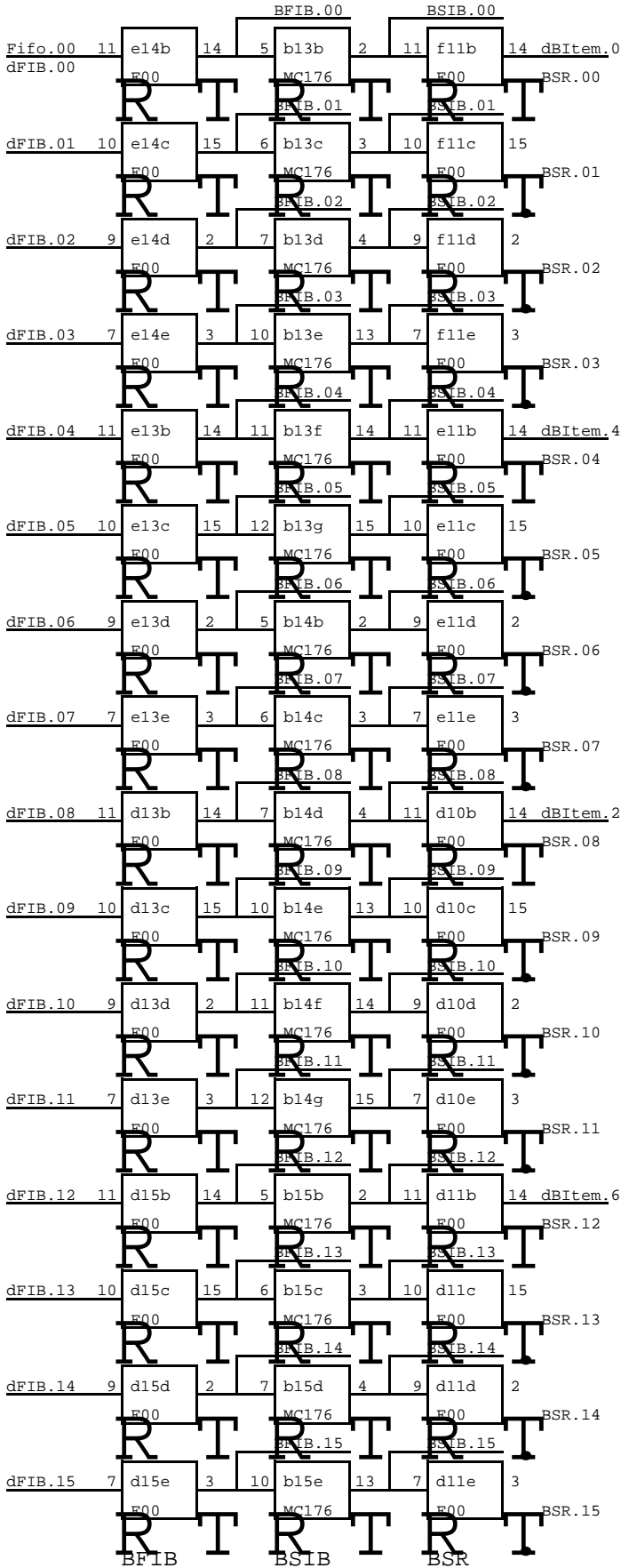


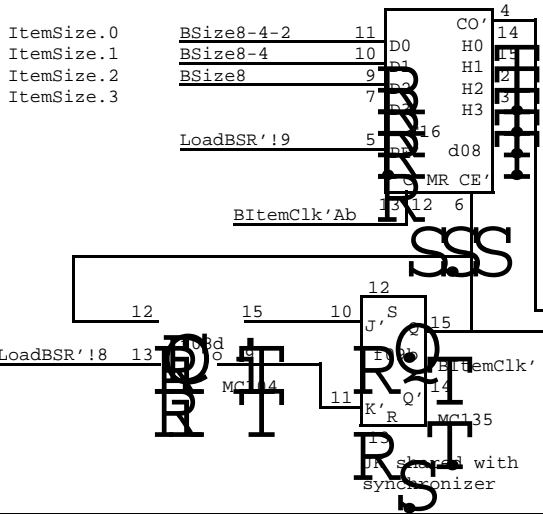
Item Clock Generator

Resolution	Full	Half	Quarter
True	1	1	1
True	1	1	1
Res.0	1	1	0
Res.1	1	0	0

Force exactly 3 ItemClks to occur in order to initialize channel data and clk counters.
 The third forced ItemClk loads SR with good data and turns the channel ON.





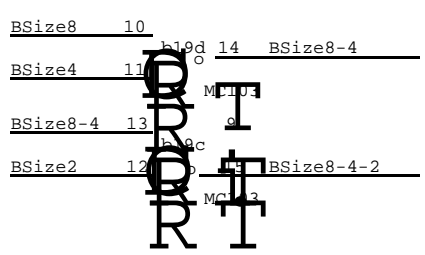


B Shift Register Control

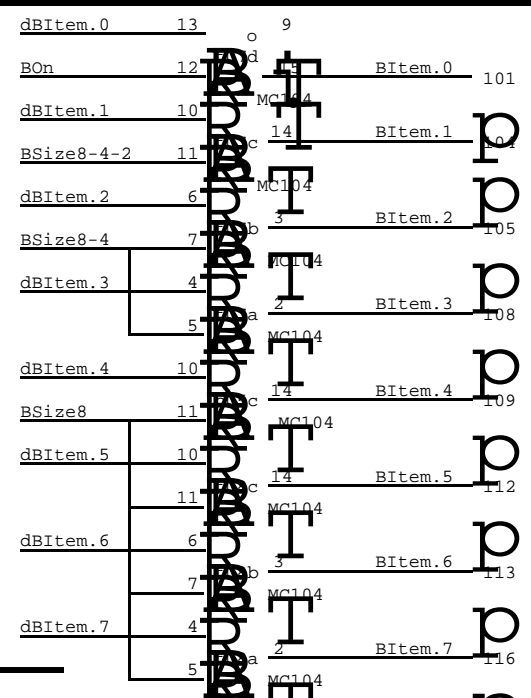
Size	8	4	2	1
ItemSize.0	1	1	1	0
ItemSize.1	1	1	0	0
ItemSize.2	1	0	0	0
ItemSize.3	0	0	0	0

Upper term makes ALL ItemClks into BSR loads to force Fifo reads and initialize counters before channel turns on. Expects exactly THREE reads to occur.

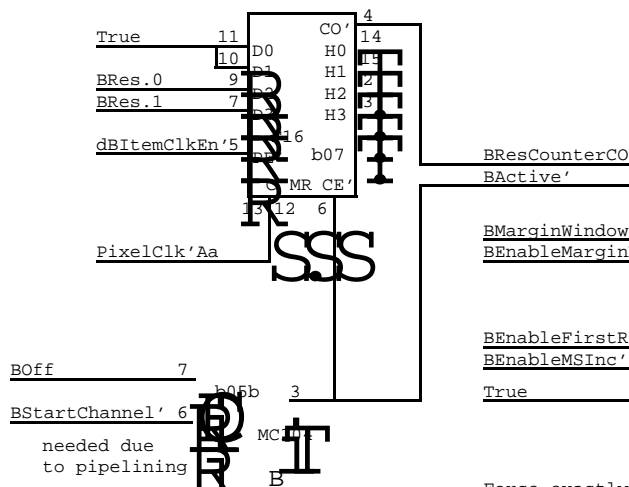
B Item buffer logic



dItems are garbage until channel becomes active



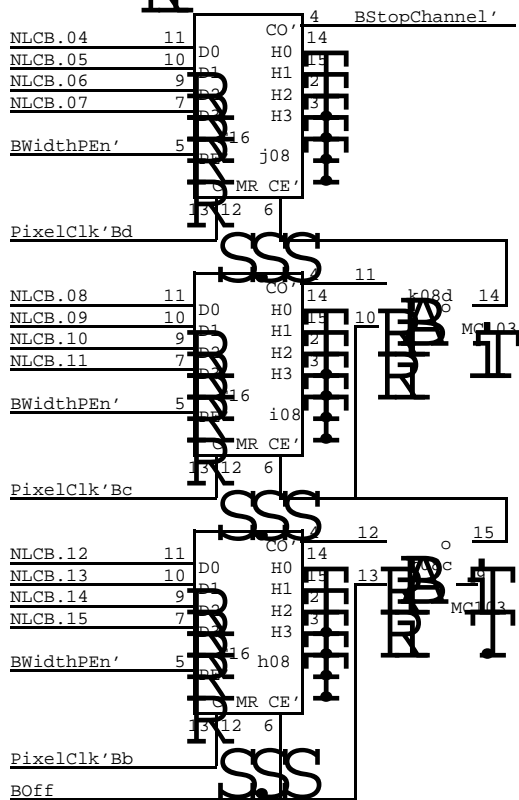
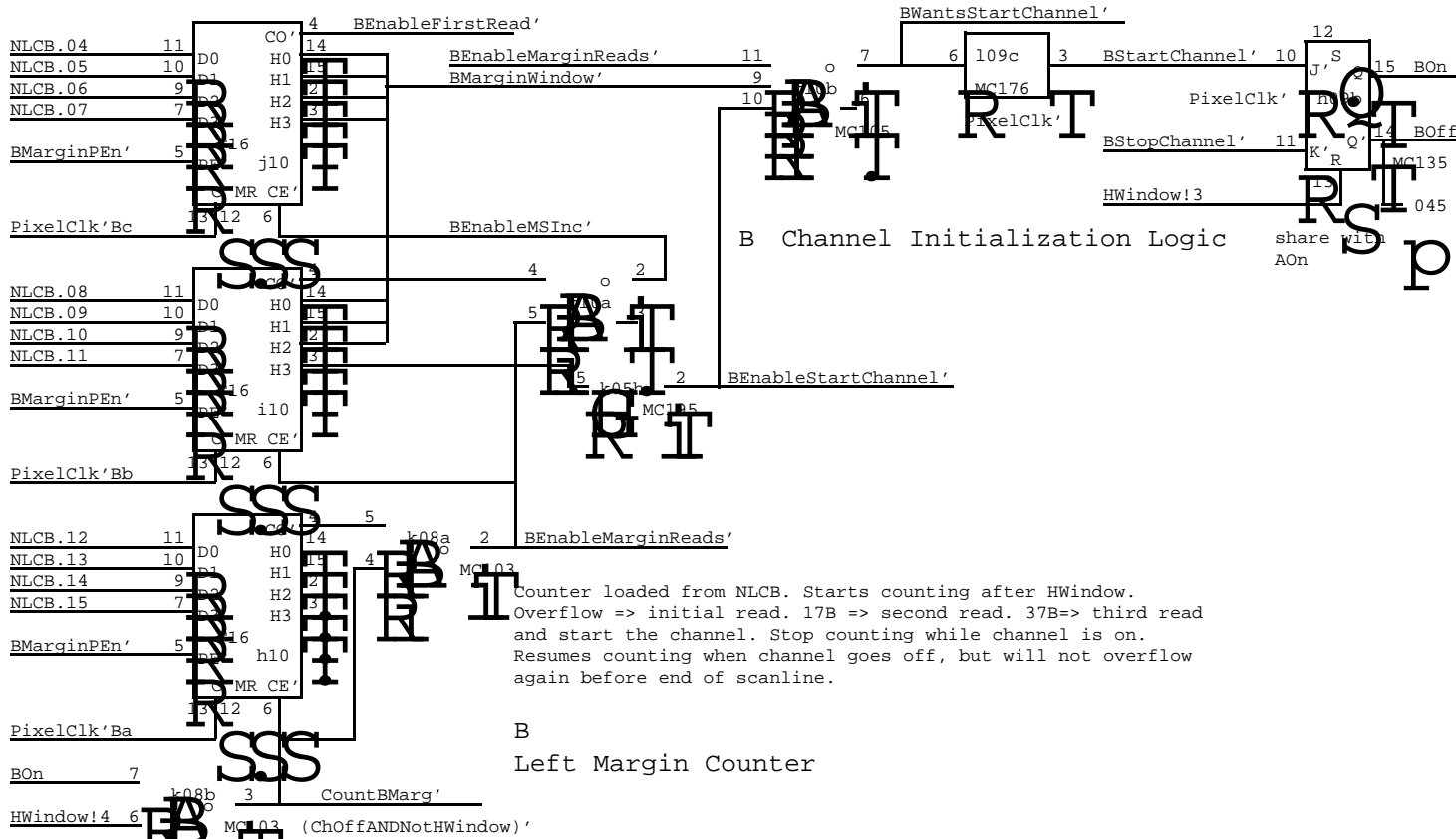
Items are ignored until channel turns ON by master reset of HOLD registers on DDM



Force exactly 3 ItemClks to occur in order to initialize channel data and clk counters. The third forced ItemClk loads SR with good data and turns the channel ON.

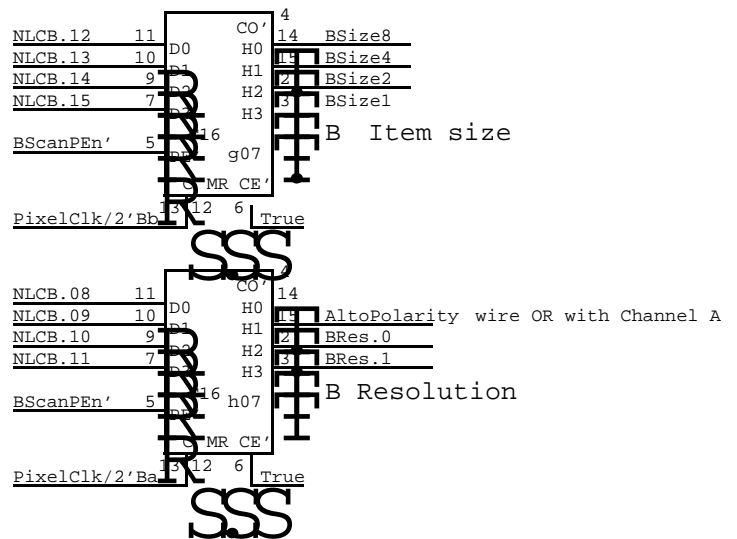
Item Clock Generator

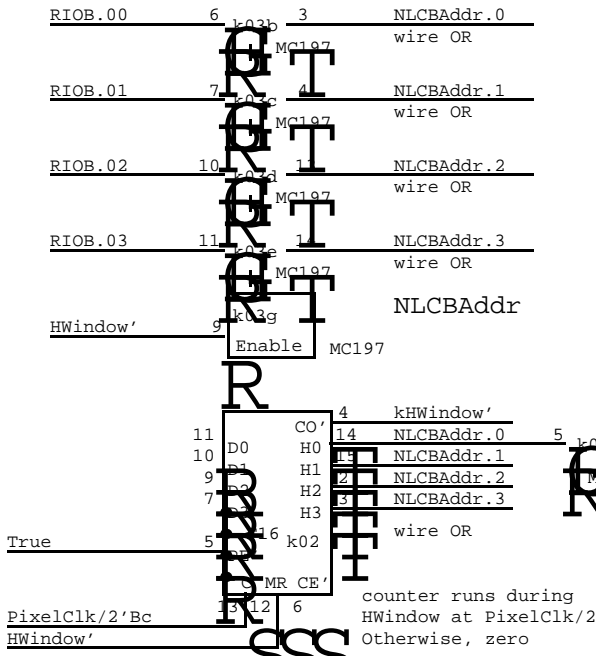
Resolution	Full	Half	Quarter
True	1	1	1
True	1	1	1
Res.0	1	1	0
Res.1	1	0	0



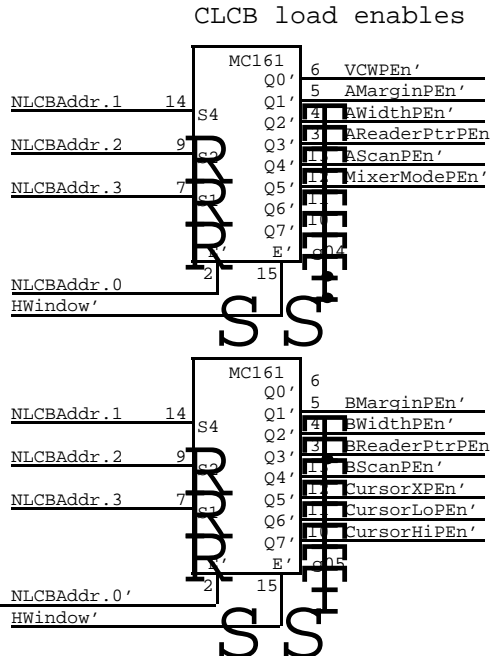
B
DCB Width Counter

Stop channel when
counter > 7400b

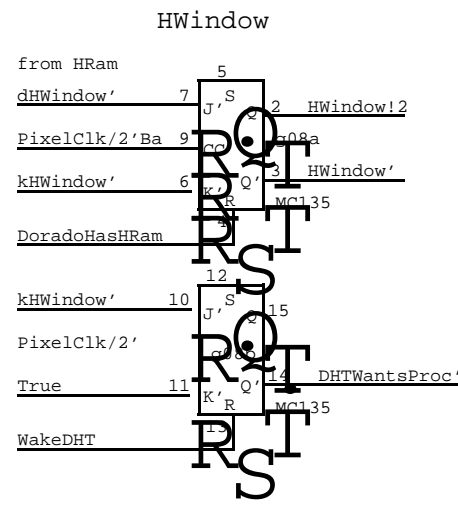
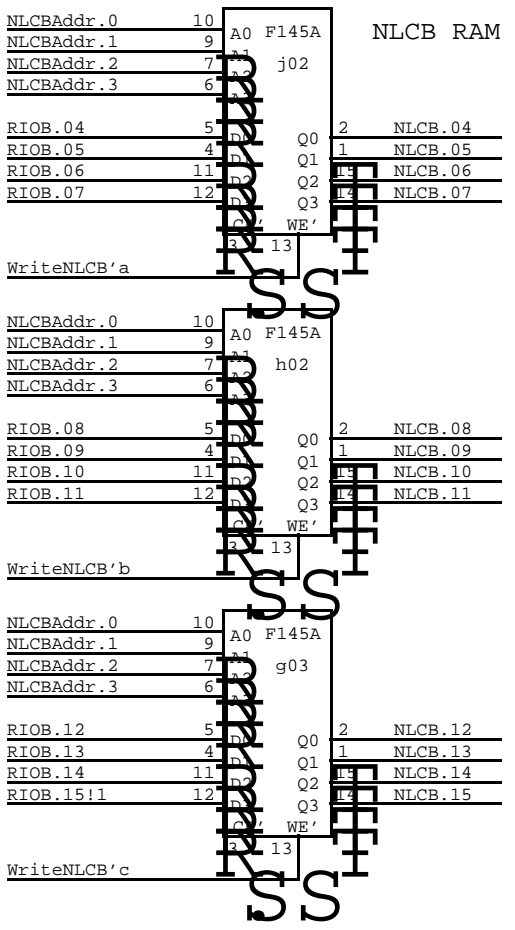




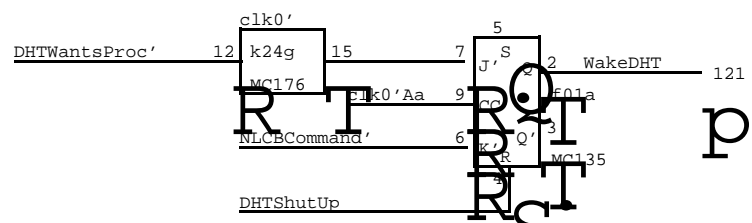
NOTE: PC/2 only valid while HRam address counter is incrementing



CursorXPEen' must occur before CursorLo/HiPEen'



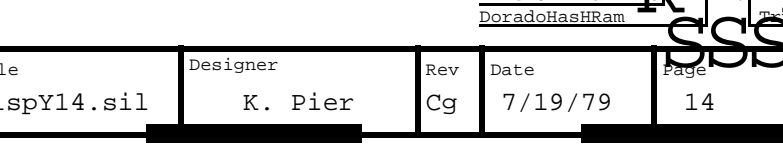
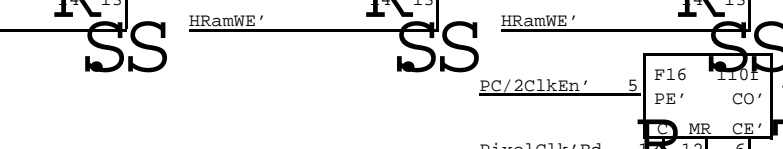
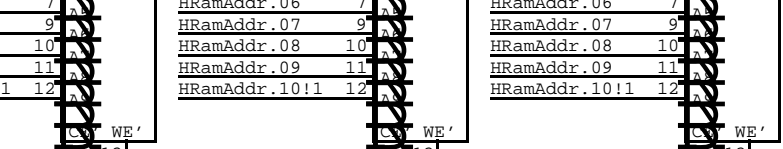
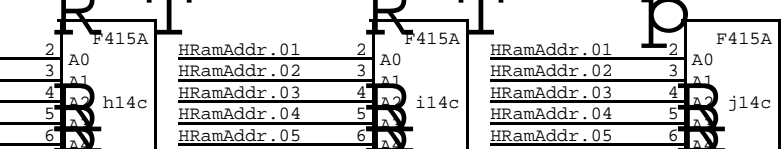
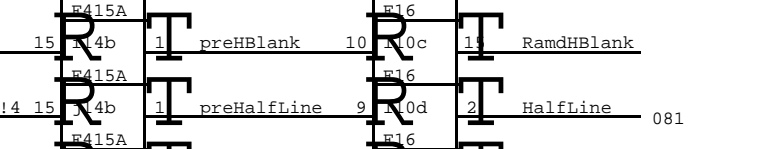
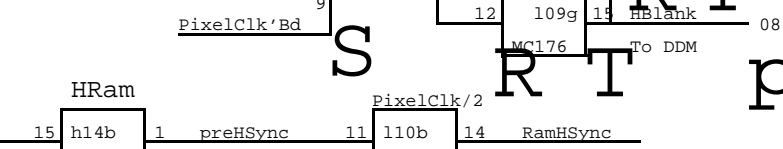
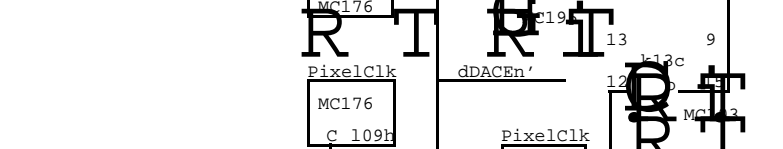
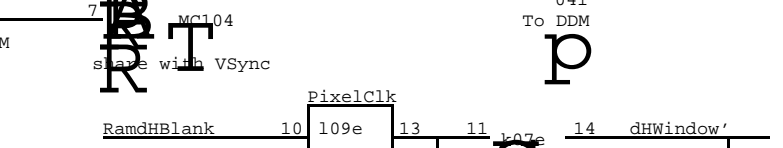
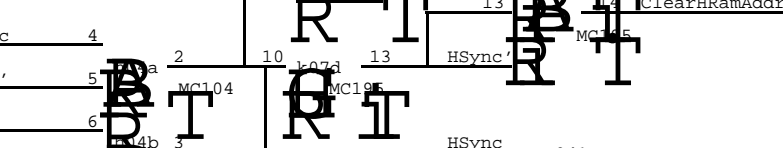
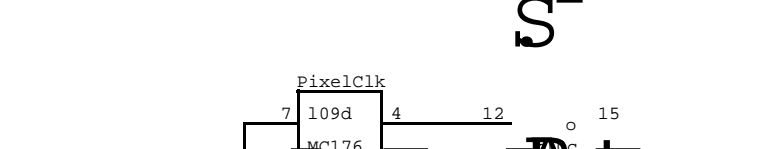
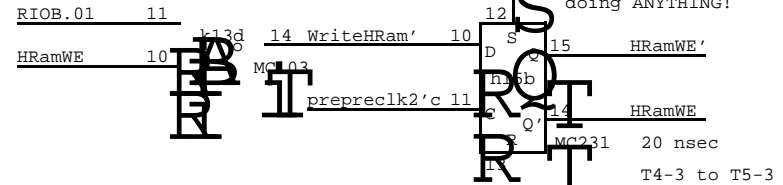
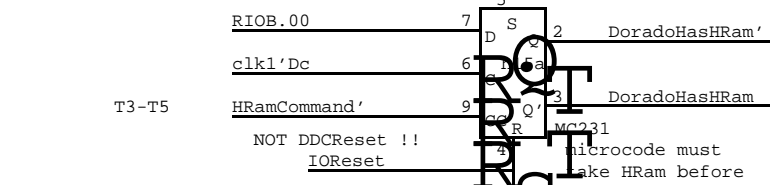
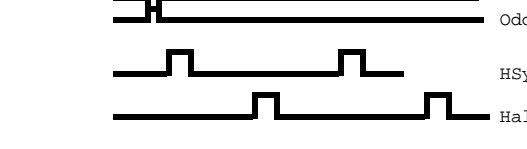
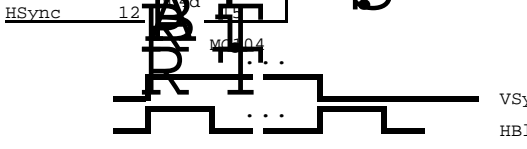
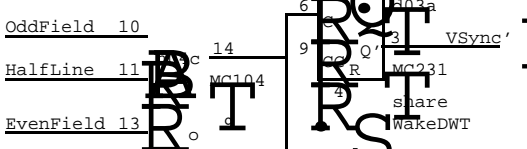
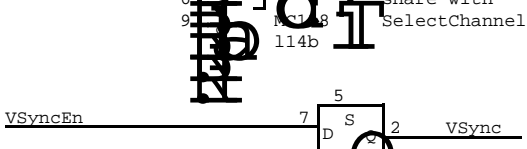
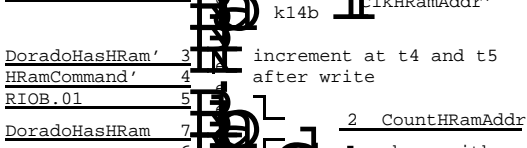
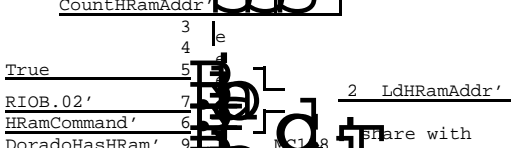
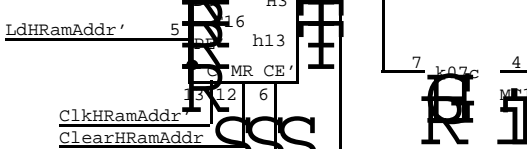
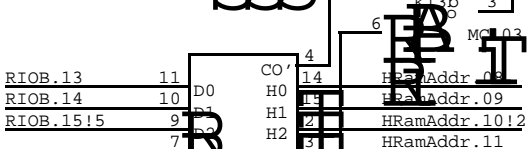
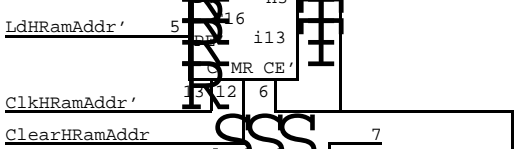
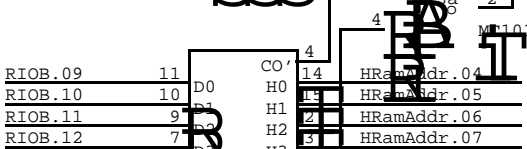
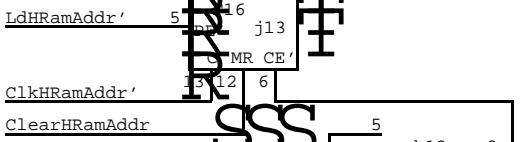
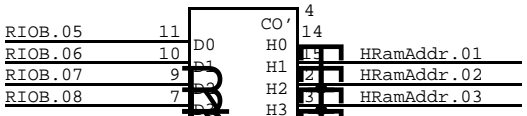
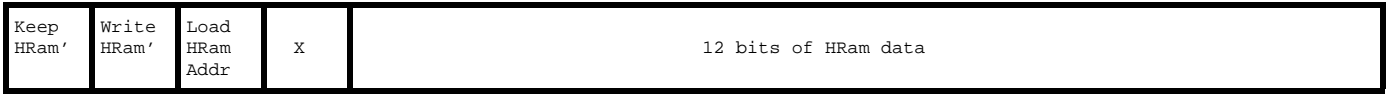
DHT WakeUp Synchronizer



For convenience, use NLCBCommand to kill wakeup. Assumes DHT will always do some NLCB command whenever it is awakened. Default would be to load an unused, don't care address in NLCB.

requires minimum 4 instruction loop

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15



THE DDC MAINTAINS, FOR EACH CHANNEL, TWO FLAGS:

NEXT Word Control Block Flag (NextWCBFlag)
 Current Word Control Block Flag (CurrentWCBFlag)

A Word Control Block is a pair of values called Address and MunchCount, for either the CURRENT or the NEXT scanline.

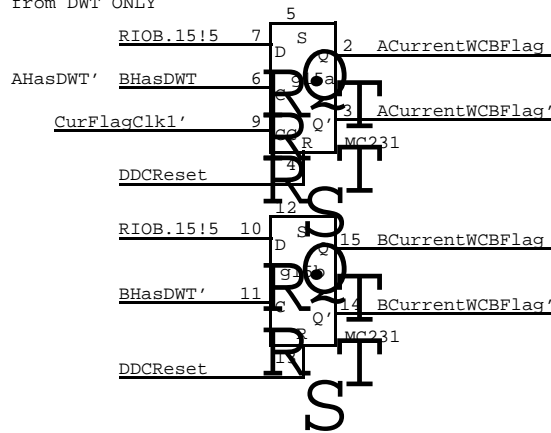
Flag management	SET	CLEARED
NextWCBFlag	by DHT when it has filled the NextWCB	by DWT when it has copied NextWCB into CurrentWCB
CurrentWCBFlag	by DWT when it has copied NextWCB into CurrentWCB	by DWT after it has sent out all the data from the CurrentWCB

WakeUp conditions:

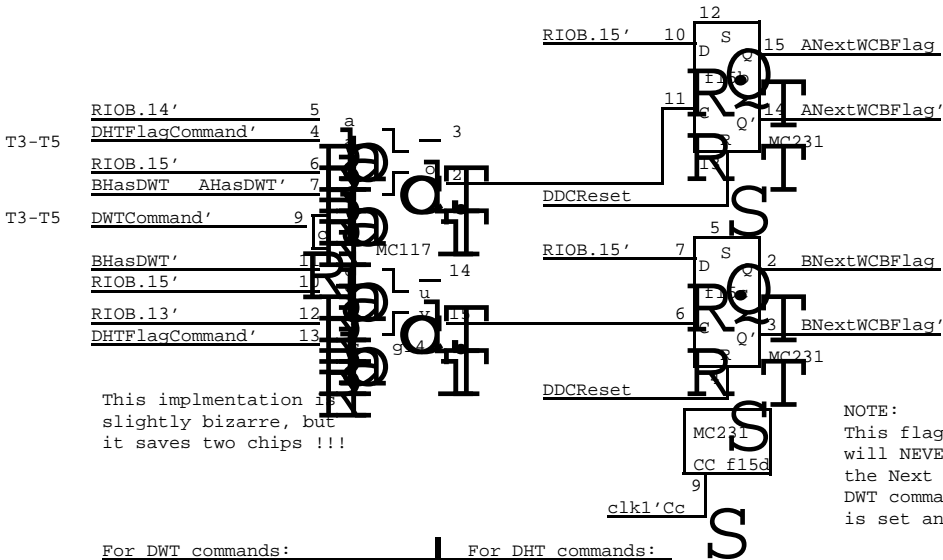
WakeDHT: whenever CLCB _ NLCB i.e.: end of every HWindow

WakeDWT: (CurrentWCBFlag AND BufferAvailable)OR (NextWCBFlag AND NOT(CurrentWCBFlag))

1 => Set CWCBFlag
 0 => Clear CWCBFlag
 from DWT ONLY



This circuit ASSUMES:
 1. a DWT command with RIOB.11 set is to be ignored. See clock page.
 2. Any DWT command with RIOB.11 low is a CWCBFlag command, either set or clear. See clock page.



This implementation is slightly bizarre, but it saves two chips !!!

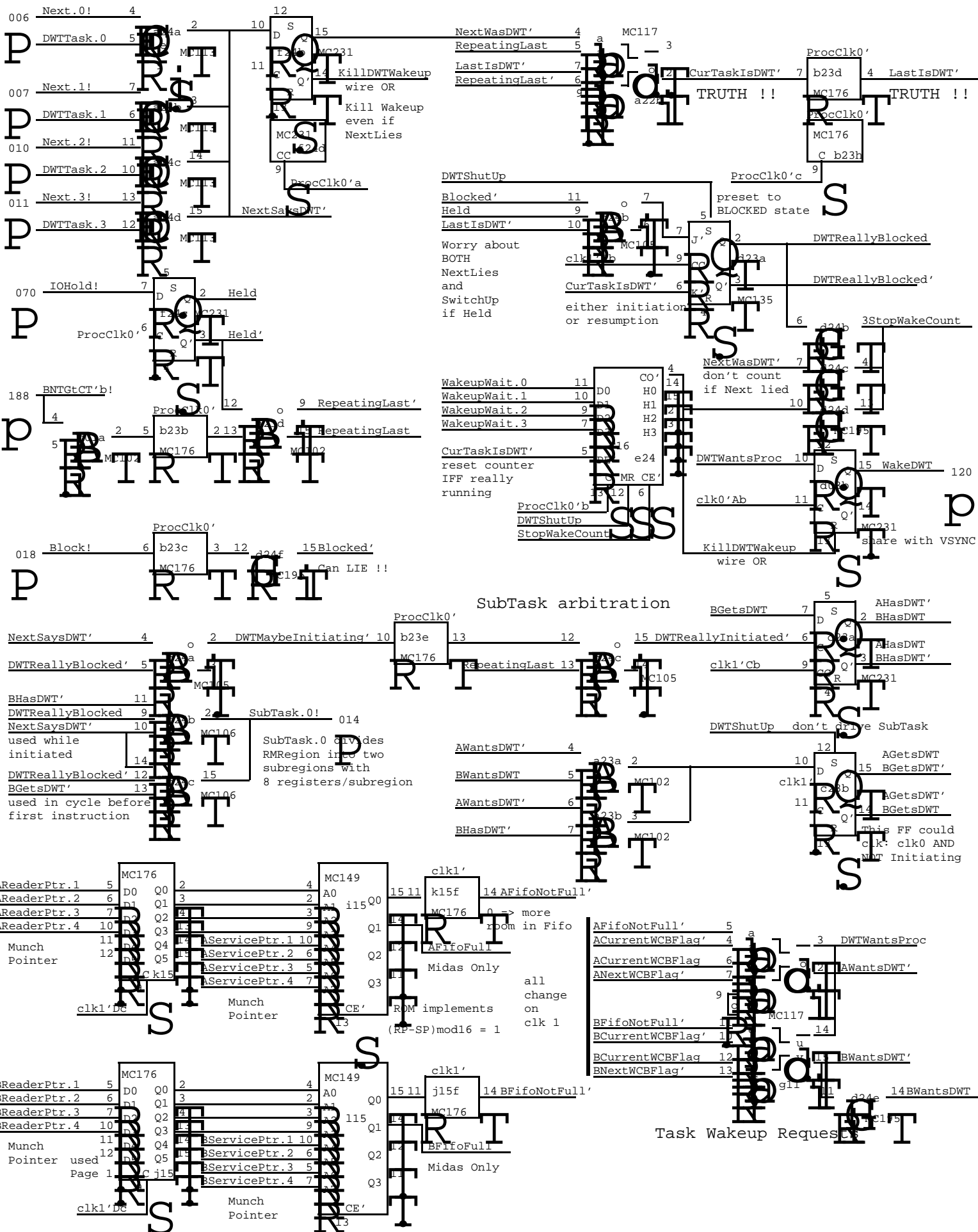
NOTE:
 This flag implementation assumes that DHT flag commands will NEVER set bit 15 and will ALWAYS set either 13 or 14, the Next flag to set. Also, it assumes the ONLY kind of DWT command it should respond to is one in which bit 15 is set and therefore NextWCBFlag is to be cleared.

For DWT commands:

RIOB	Means
=20c	IOFetch signal don't touch Flags
=1c	Set CWCBFlag and Clear NWCBCFlag
=0c	Clear CWCBFlag

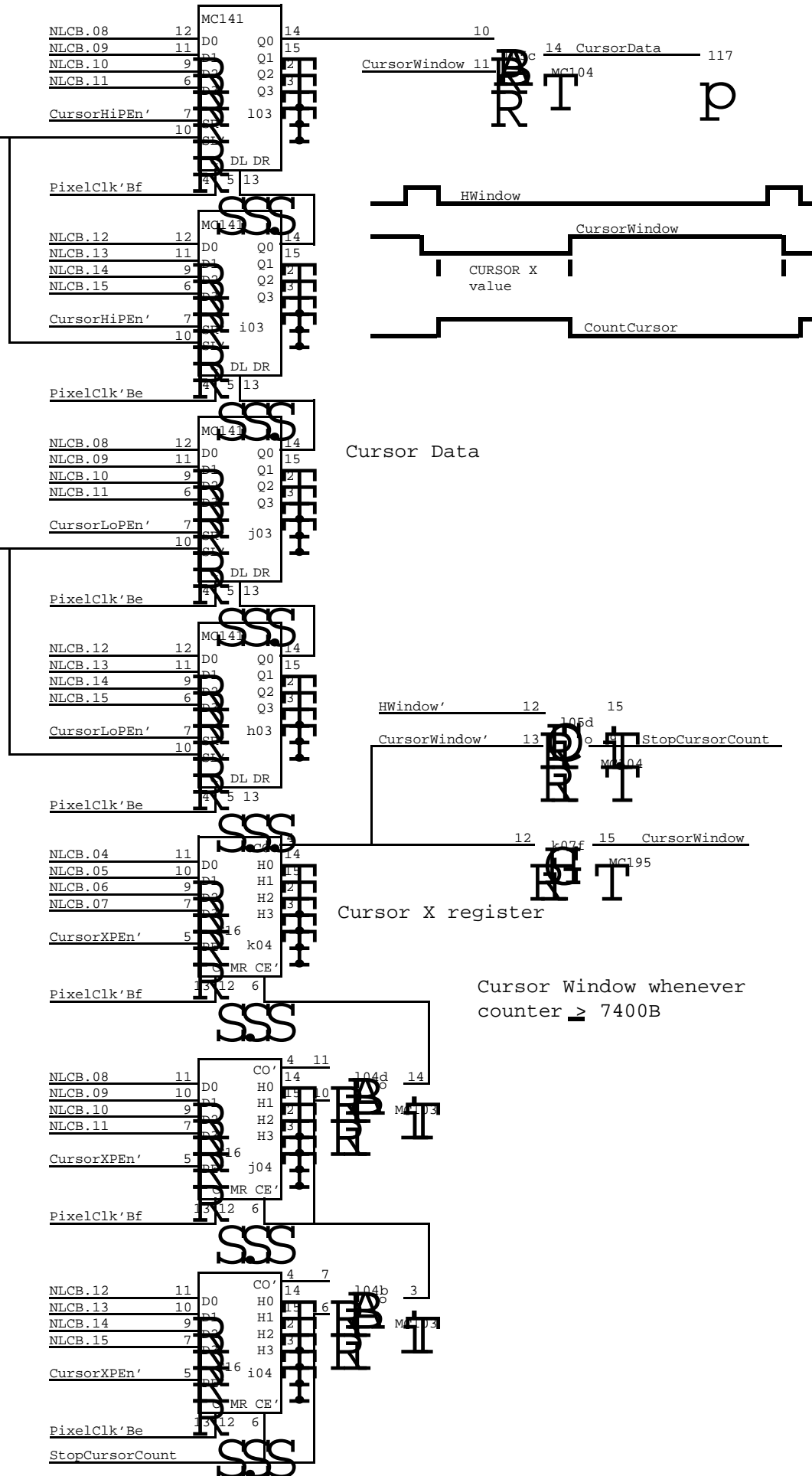
For DHT commands:

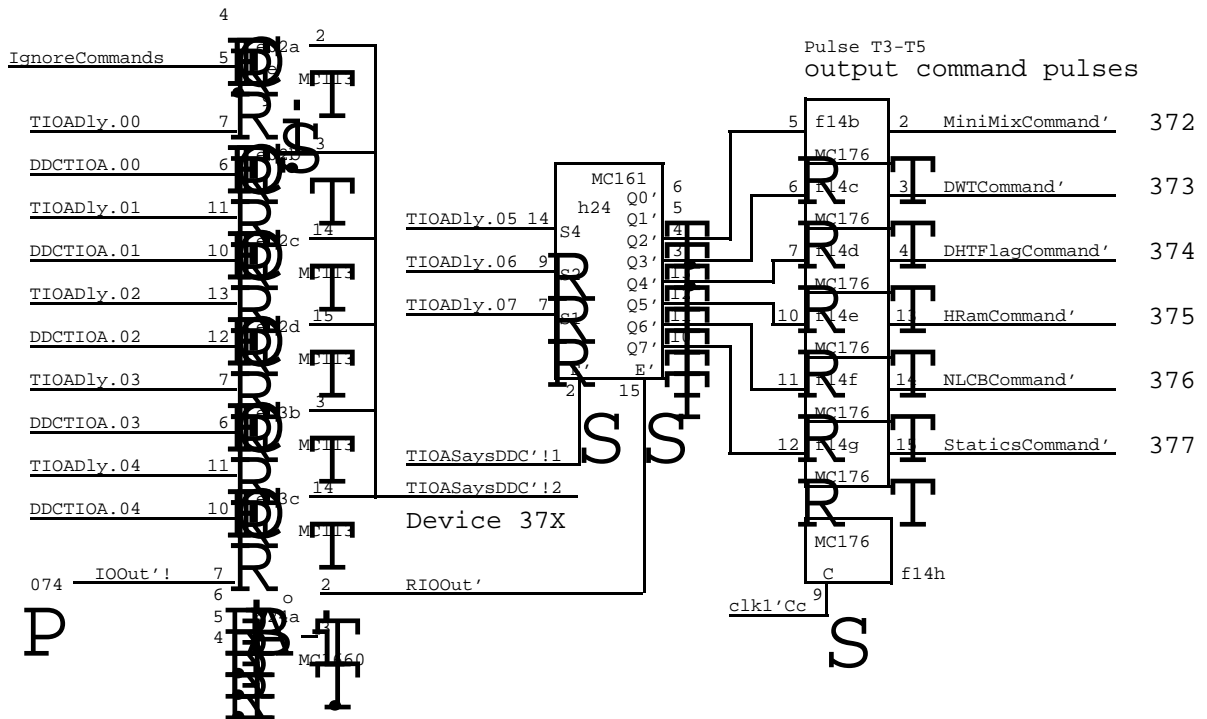
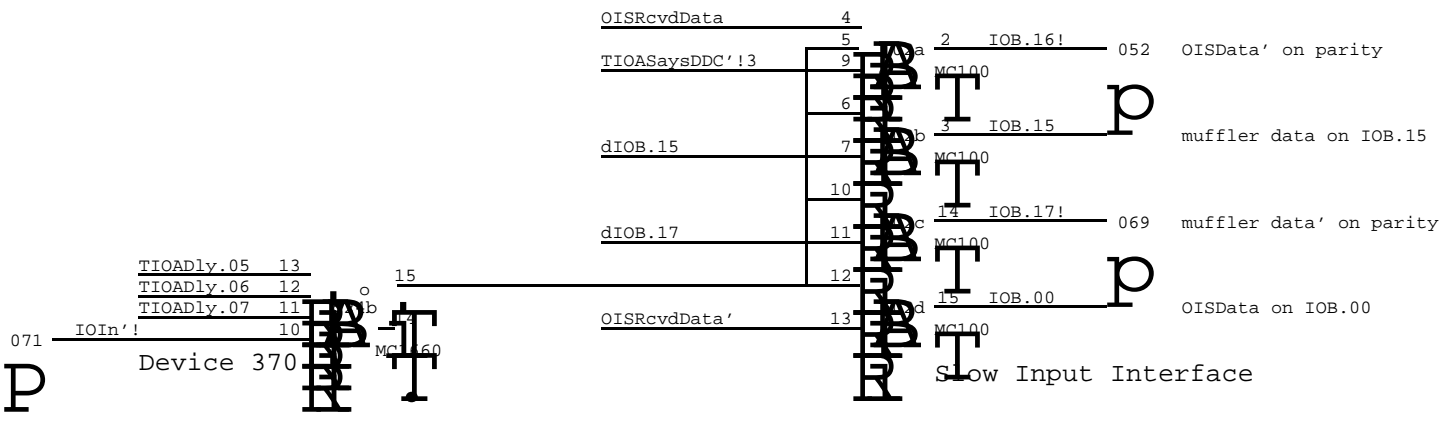
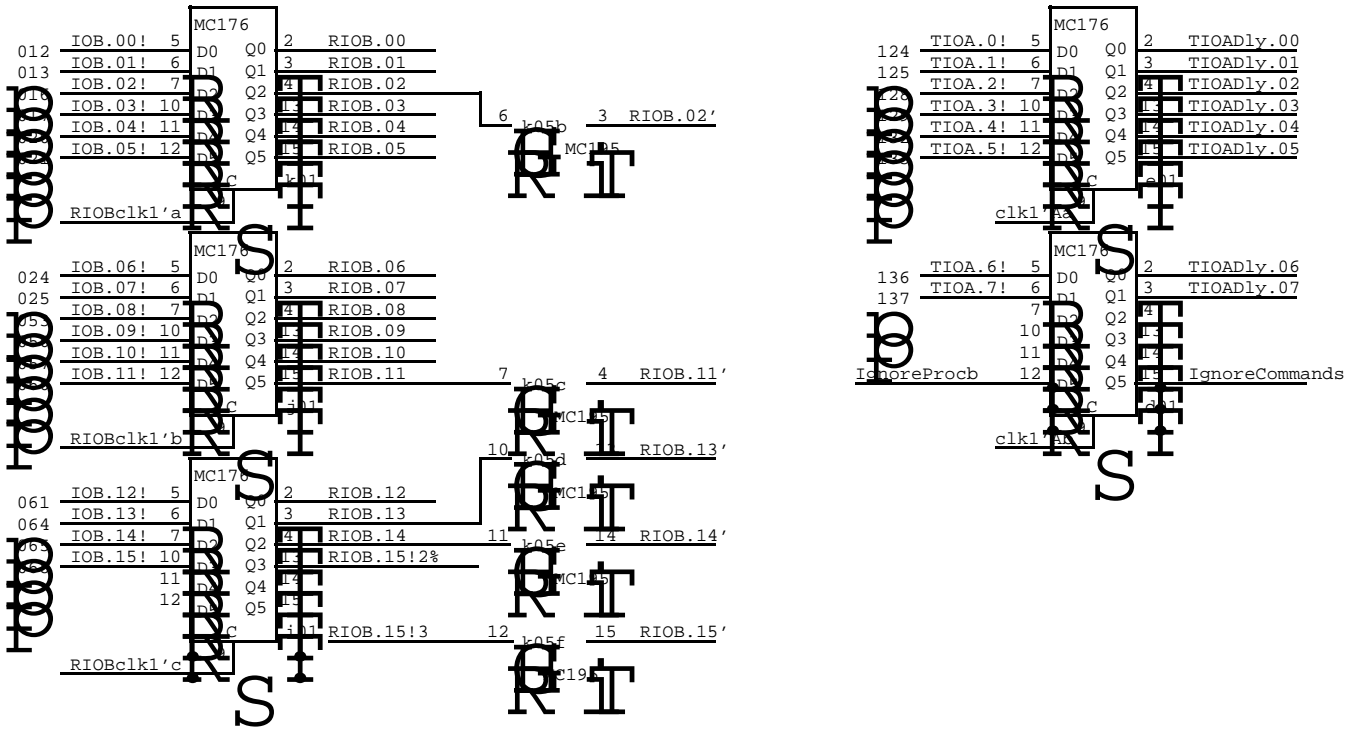
RIOB	Means
=2c	Set ANextWCBFlag
=4c	Set BNextWCBFlag

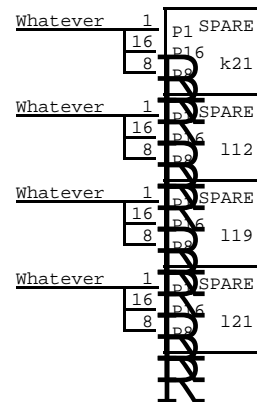
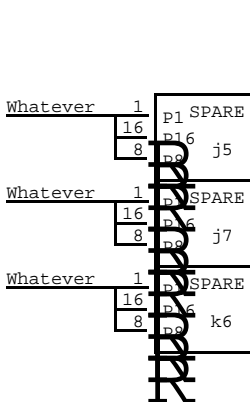
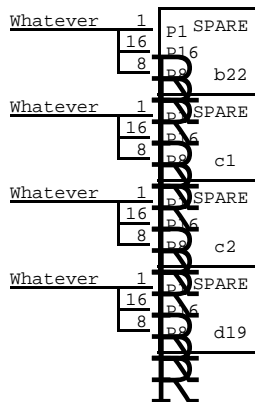
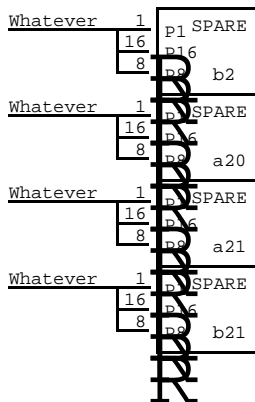
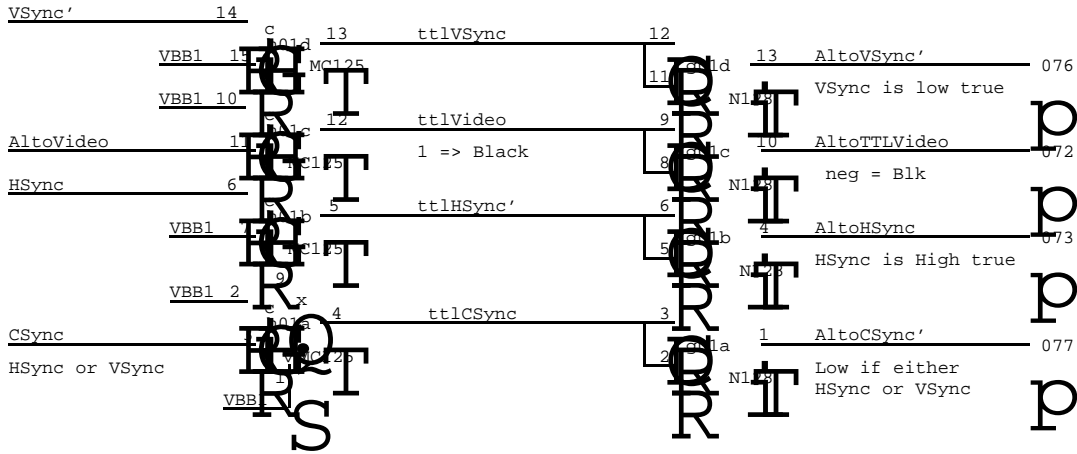
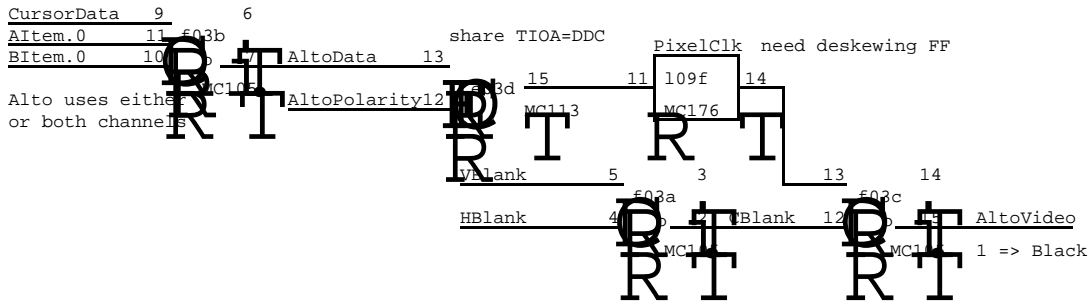


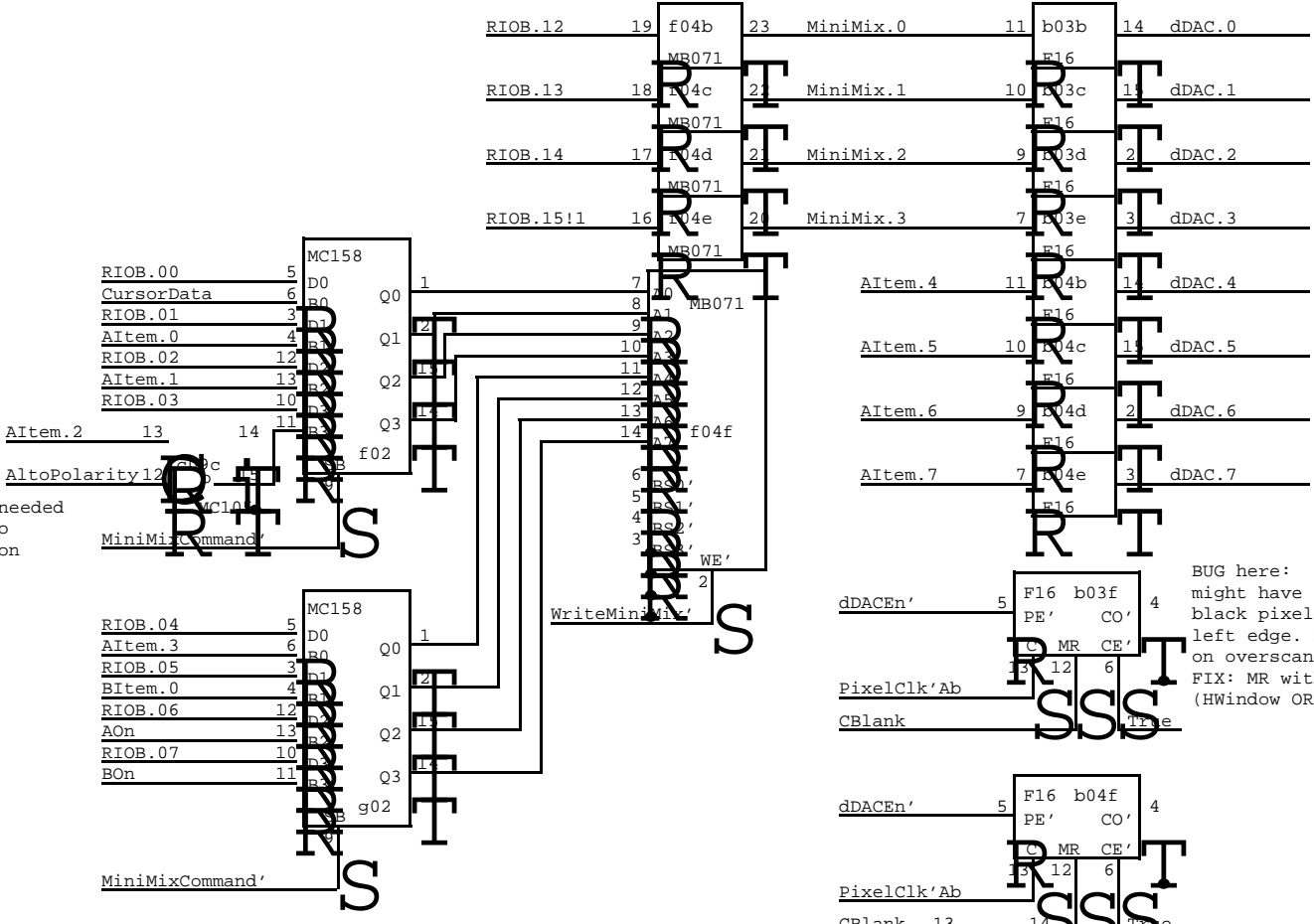
CursorHiPEn' 5
 CursorWindow' 4
 during HWindow
 CursorWindow must
 go inactive
 before loading
 CursorHi/lo with
 new data, so
 CursorX is loaded
 before CursorHi/Lo

CursorLoPEn' 7
 CursorWindow' 6



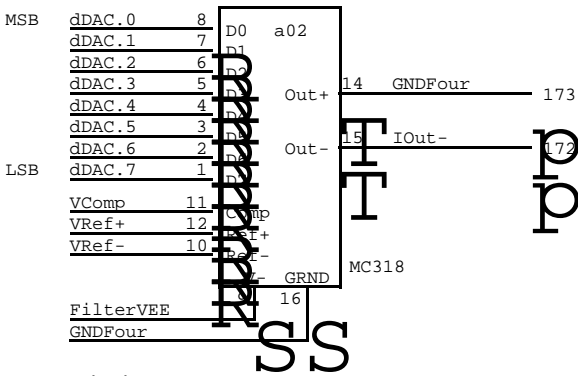




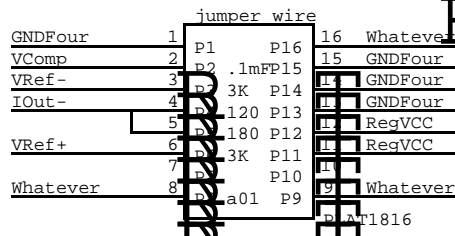


NOTES:

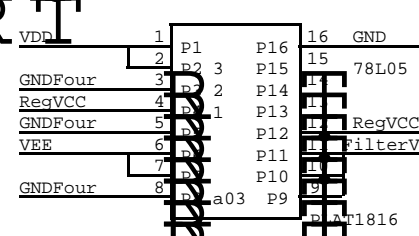
GNDFour is used as single point GND for DAC system



Digital/Analog converter



Platform DAC components



Voltage filters

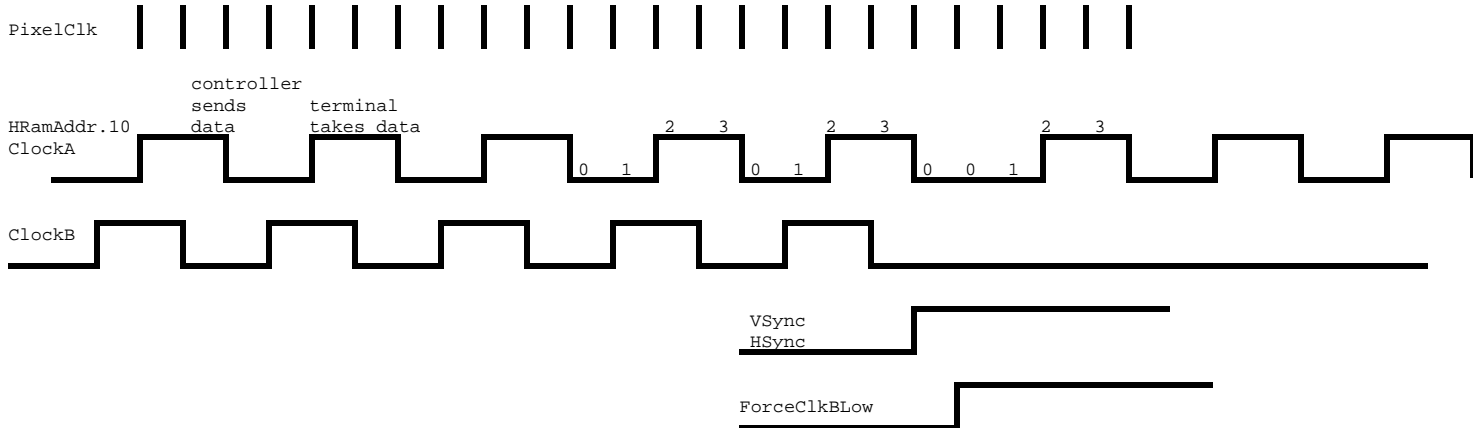
1	jumper wire	16	Whatever
2	0.1mFarad	15	GNDFour
3	3 KOhm	14	GNDFour
4	120 Ohm	13	GNDFour
5	180 Ohm	12	ReqVCC
6	3 KOhm	11	ReqVCC
7		10	Whatever
8		09	Whatever

location a01

1	0.1mFarad	16	GND	
2	in3	78L05	15	78L05
3	gnd	2	14	
4	out1	13		
5	0.1mFarad	12		
6	12 mHnry	11		
7	2 ohm	10		
8	+ 22 mF	09		

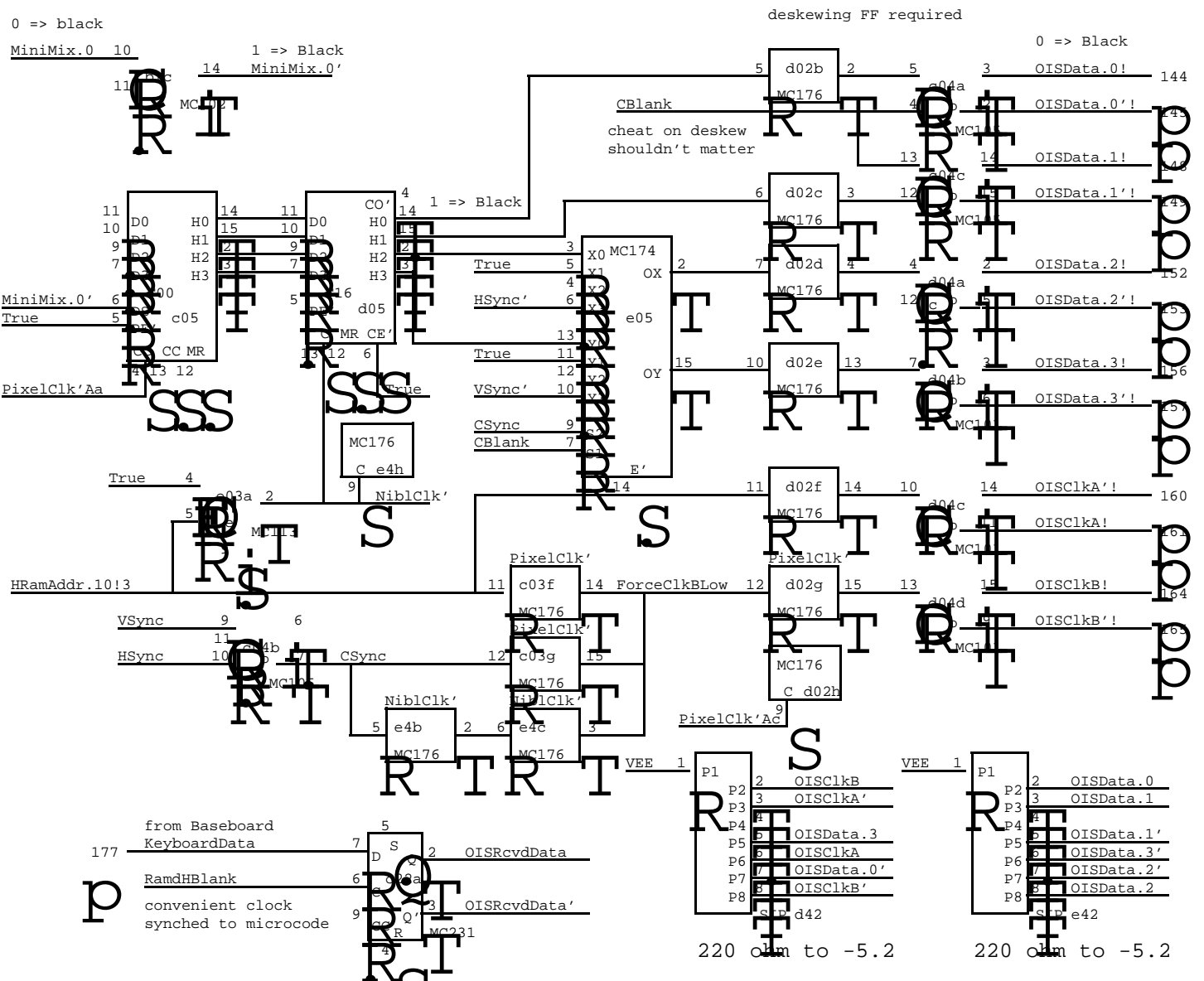
location a03

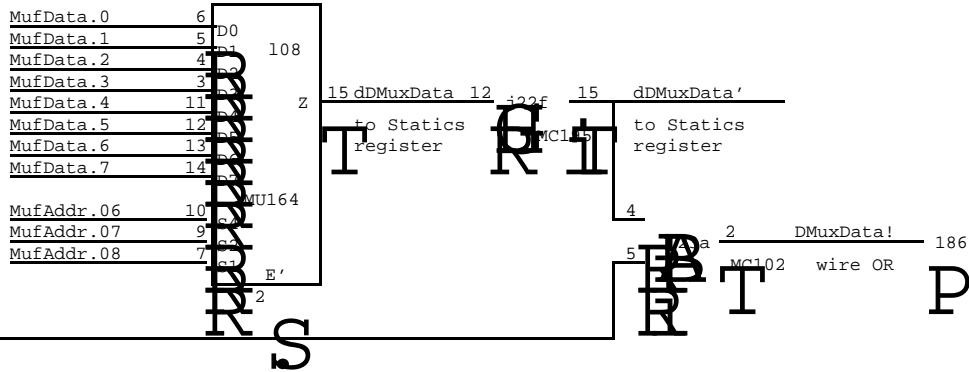
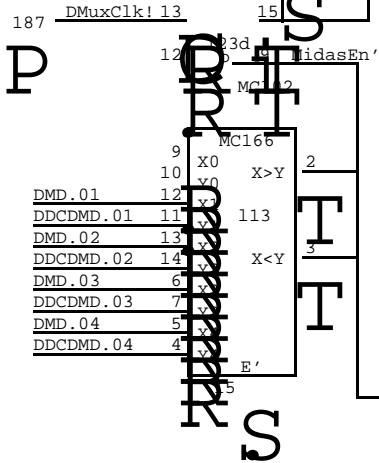
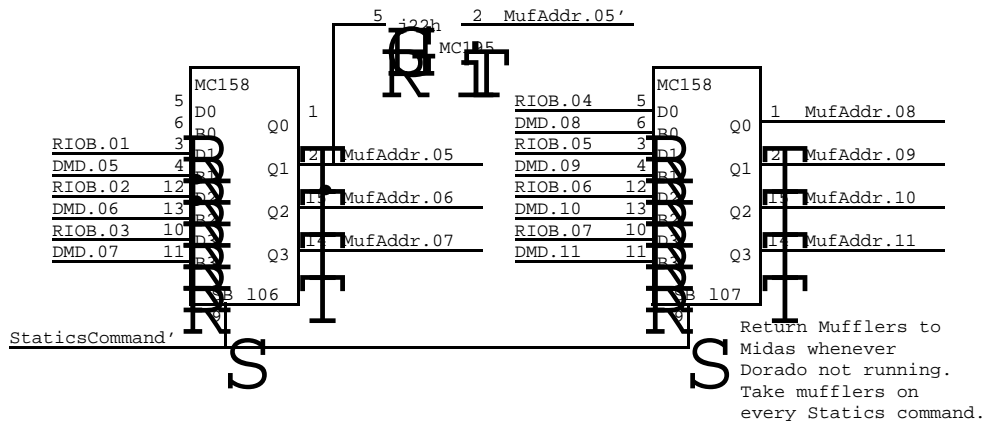
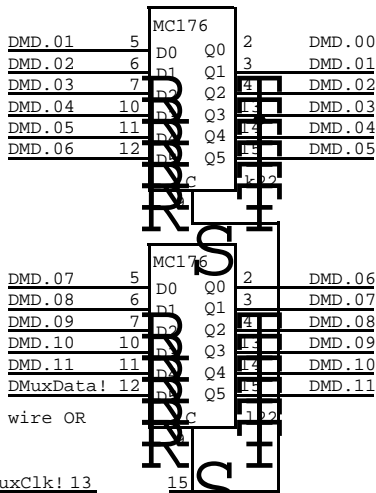
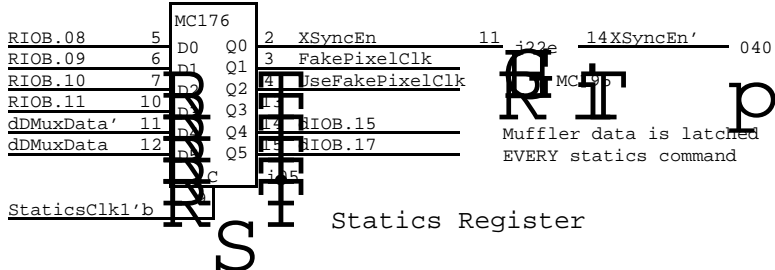
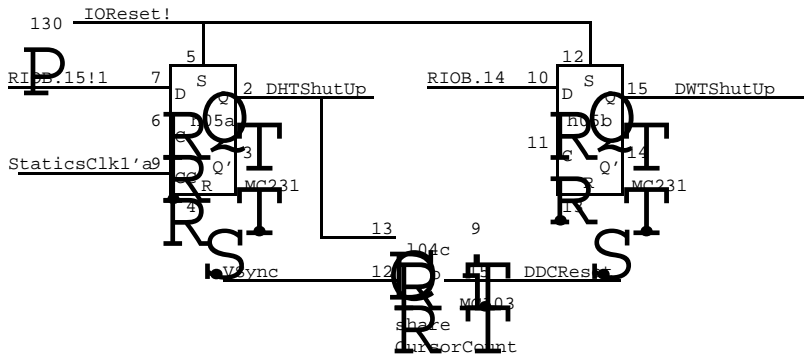
RevCcd: swapped positions of 12mH and 2mF

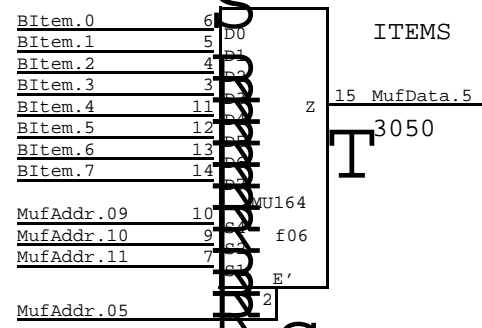
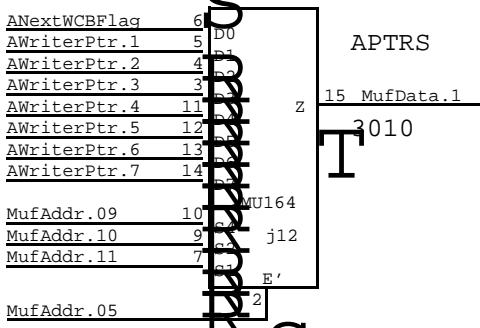
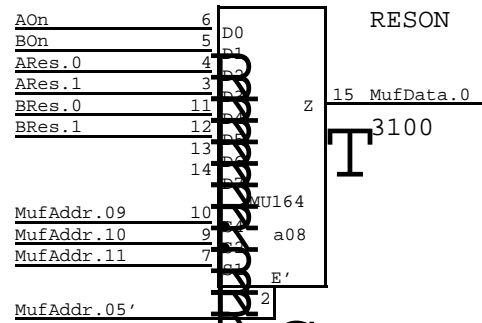
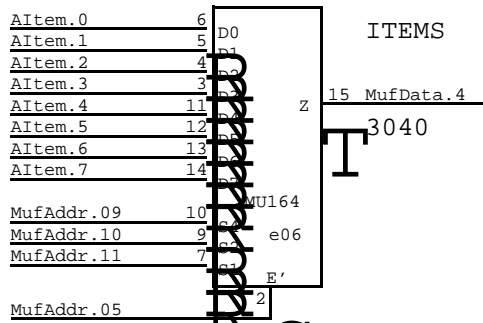
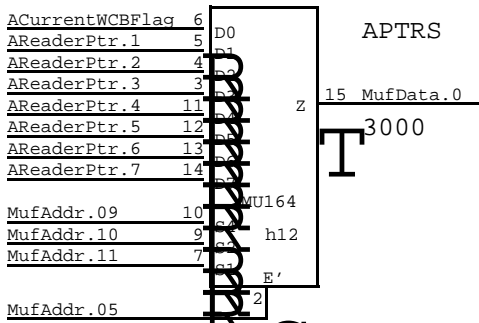


CLAIM: HRamAddr.10 is the waveform for ClockA. Delay it by one pixelclock, and invert, then you get the waveform for ClockB. ClockB is also jammed low by (HSync OR VSync) delayed by one PixelClk. To avoid hiccups in the phasing of ClockA with respect to ClockB, HSync should only be generated just after HRamAddr.10-11=3, that is, a hiccup would occur if HRamAddr.01-11 = 1 just before HSync occurred.

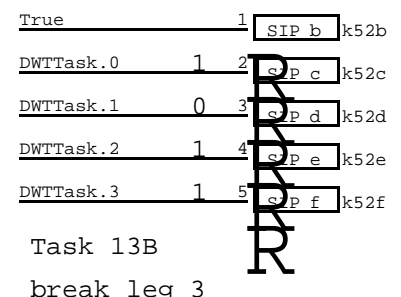
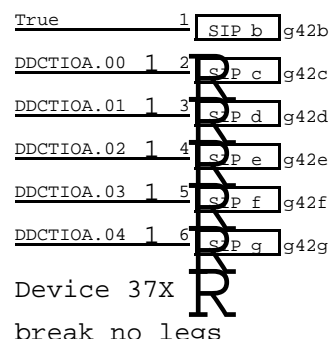
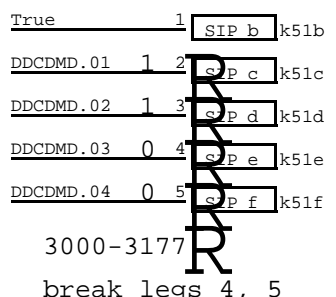
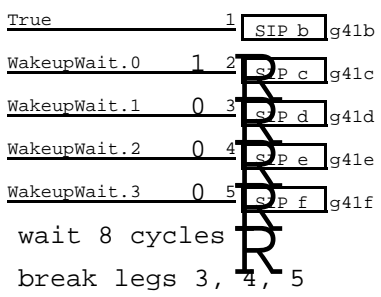
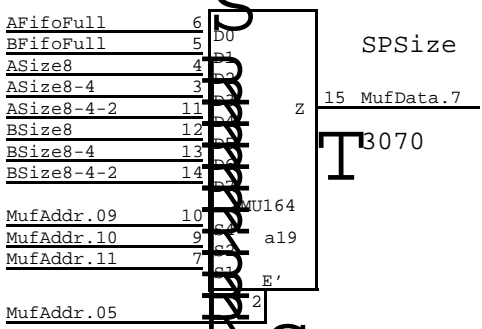
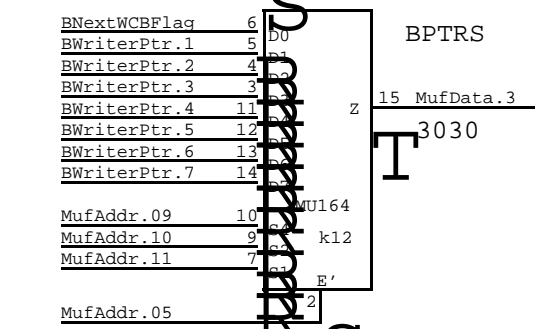
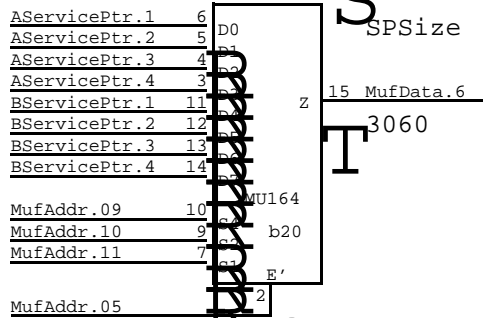
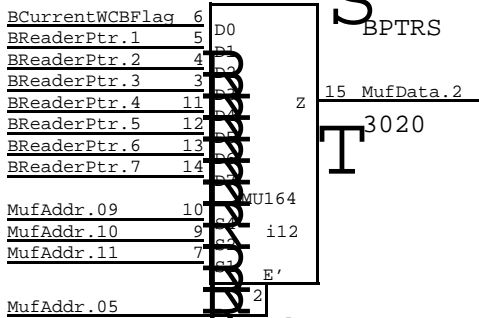
CBlank must be phased so that it changes when HRamAddr.10 goes from 1 to 0. However, CBlank is delayed by two pixel clocks from RamdHBlank, so the HRam must be programmed to make RamdHBlank change on the 0 to 1 transition of HRamAddr.10. This is slightly confusing, so be careful.



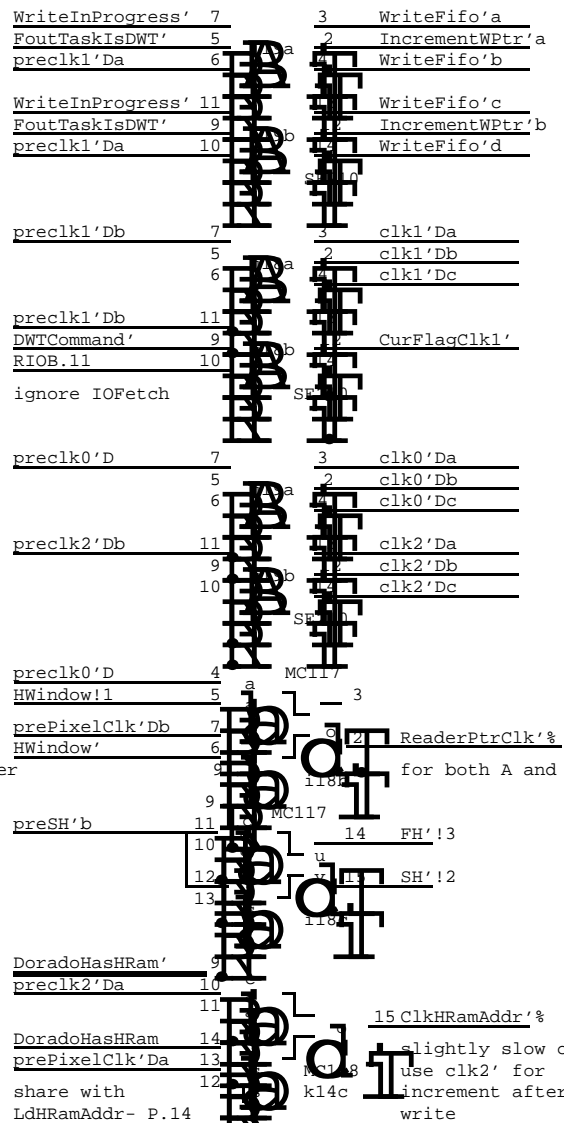
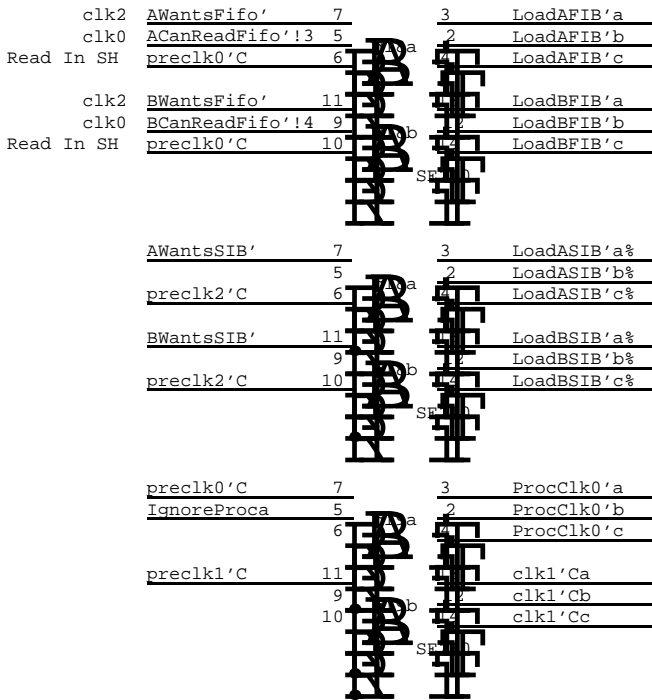
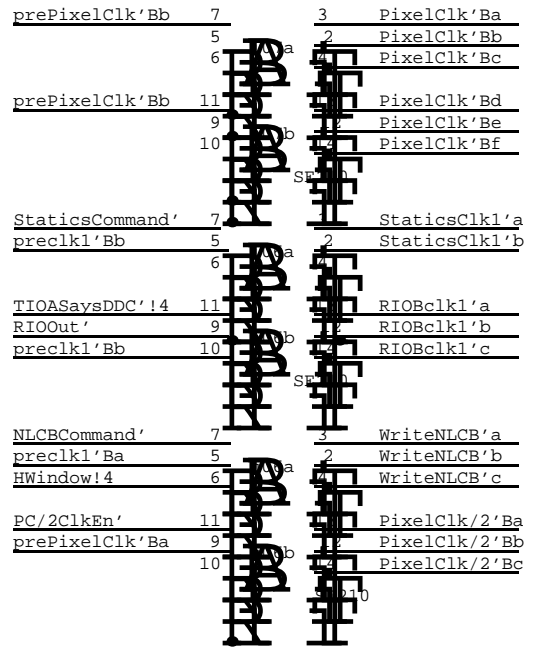
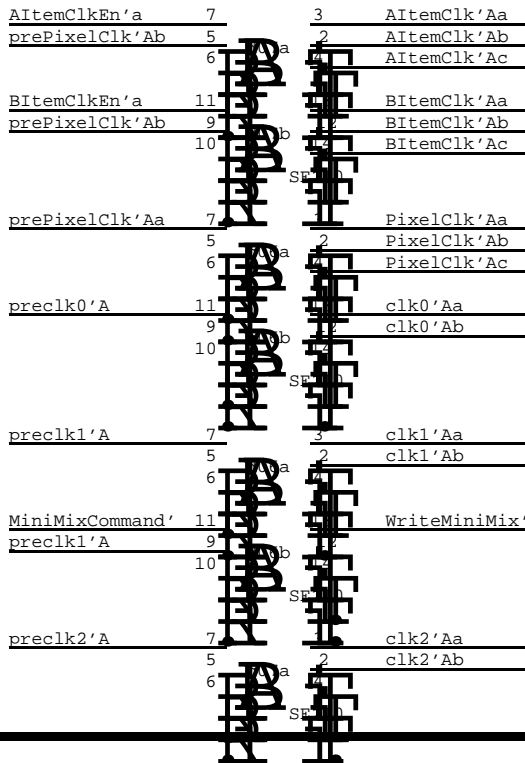




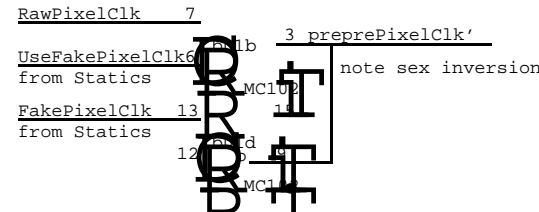
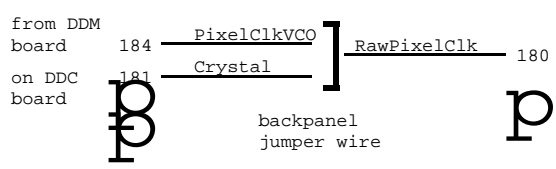
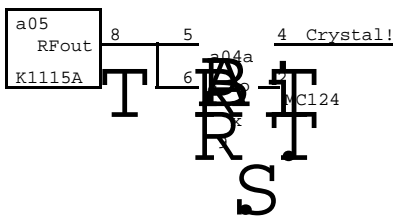
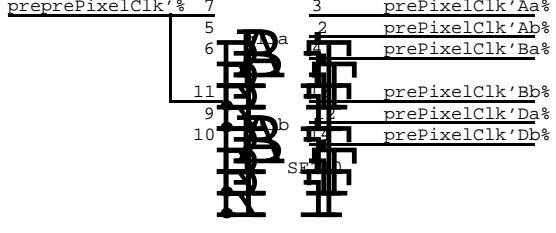
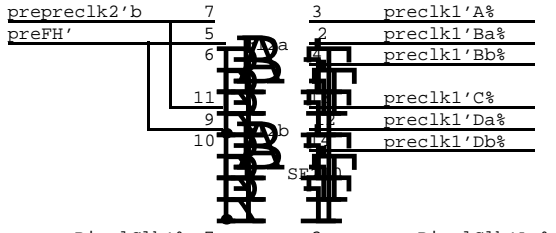
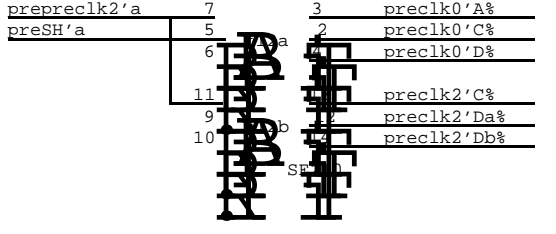
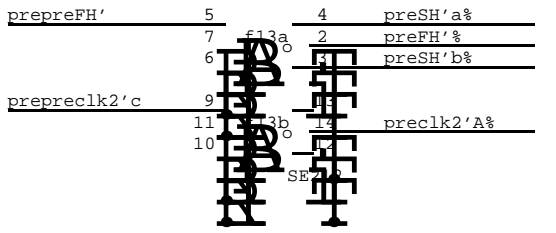
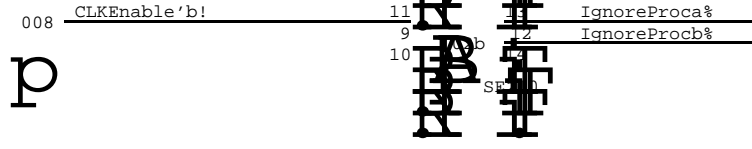
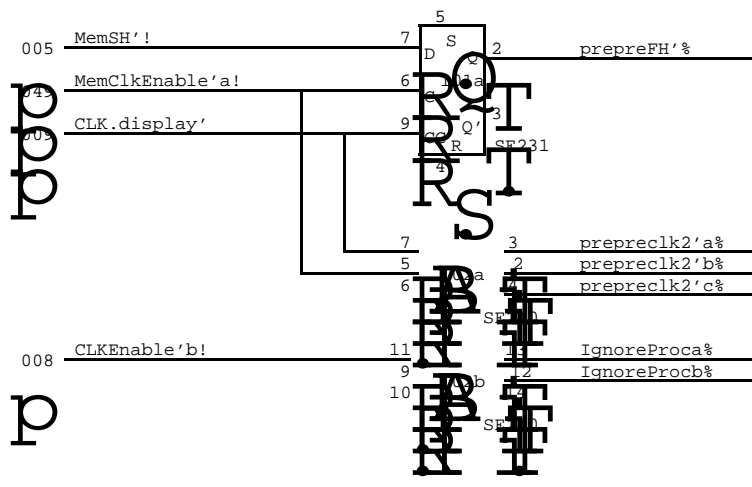
3110-3177 unused



All SIPs are standard 100 ohm terminators with legs broken as shown



CAUTION:
flakey
synchronizer



C	CONNECTOR		CONNECTOR		TIOA	TW	CONNECTOR		IOBLo	CONNECTOR		IOBHi	CLK	C
A	B>Ct	a 181	b 168	c 153	d 137	e 124	f 109	93 g	80 h	64 i	48 j	33 k	20 l	B
1	DAC Plat	PCLK B>Ct 102			TIOADEF 176 1	TIOA 176 1	DHTTw RdFifo 135	ALTO 74128	ALTO 125	RIOB 176 rc1	RIOB 176 rc1	RIOB 176 rc1	SE231	1
2	DAC Plat				OIS 176	TIOA=D 113	MMMux 158	MMMux 158	NLCB 145 W	DIOB 100	NLCB 145 W	NlcbAd F16 P2	SE210	2
3	DAC Plat	MMix F16 PC	OIS ICEN 176 PC		DWTTw 231 0	TIOA=D 113	ALTO 105	NLCB 145 W	CURSR 141 P	CURSR 141 P	CURSR 141 P	NlcbAd 197	CURSR 141 P	3
4	PCLK Xtal	MMix F16 PC	OIS 105		OIS 101	OIS 176	MMix	ClcbEn 161	VSyC 104	CURX F16 P	CURX F16 P	CURX F16 P	CURX 103	4
5	PCLK Xtal	LSR 104	OIS F00		OIS F16	OIS 174	MMix	ClcbEn 161	STAT 176 sc1	STAT 231 sc1		195 RIOB'	CURSR 104	5
4	AOICEN 121	BdICEN 121	PC MMix		0,1	MU AITEM	MU BITEM	ASIZE F16 P2	ARES F16 P2	sc1 rc1	NLCB 1 PC/2		MIDAS 158	6
7	ARESC F16 P	BRESC F16 P	2 210 B		ABITEMCLK	AITEM 104	BITEM 104	BSIZE F16 P2	BRES F16 P2	PC PC		PULSE 195	MIDAS 158	7
4	MU ASOFF OIS RES	LSR 118	ASRC F16 I		BSRC F16 I	AITEM 104	BITEM 104	HWDO 135 P2	BWDTH F16 P	BWDTH F16 P	BWDTH F16 P	BWDTH F16 P	MIDAS 164	8
9	ASYNC 231 2	ASYNC 231 2	ASYNC 105		BSYNC 231 2	ASYNC 135 I	BSYNC 135 I	MODE F16 P2	ABOFF 135 P	AWDTH F16 P	AWDTH F16 P	AWDTH F16 P	176 PC P	9
4	ASR I F00	BSR I F00	ASR F00		BSR F00	ASR F00	BSYNC 105	VCW F16 P2	BLMarg F16 P	BLMarg F16 P	BLMarg F16 P	BLMarg F16 P	HRmOut F16 P	10
11	ASR I F00	BSR I F00	ASR F00		BSR F00	ASR F00	BSR F00	WANT 117	ALMarg F16 P	ALMarg F16 P	ALMarg F16 P	ALMarg F16 P	AWDTH 103	11
4	ASR I F00	BSR I F00	ASR F00		BSR F00	ASR F00	pc0' pc2' SE210	pc1' pc1' SE210	MU ARP AF	MU BRP BF	MU AWP AF	MU BWP BF		12
13	ASIB 176	BSIB 176	AFIB F00		BFIB F00	BFIB F00	pc2 FHSH SE212	ppc' ppc' SE210	HRAddr F16 H	HRAddr F16 H	HRAddr F16 H	HRAddr F16 H	MIDAS 166	13
4	ASIB 176	BSIB 176	AFIB F00		BFIB F00	BFIB F00	COMM' 176 1	flags 117	HRam 415 HW	HRam 415 HW	HRam 415 HW	HRamC 118	HRamC 118	14
15	ASIB 176	BSIB 176	AFIB F00		BFIB F00	IP	NFlags 231 1	CFlags 231 cf	DHashR 231 1,P2	AFULL 139	BFULL 176 1	AFULL 176 1	BFULL 139	15
4	ASIB 176	BSIB 176	AFIB F00		BFIB F00	IP	IP			ARP rpc F16	BRP rpc F16	AWP wpc F16	BWP wpc F16	16
17	ASIB 176	BSIB 176	AFIB F00		BFIB F00	IP	IP	FIFO	FIFO	ARP rpc F16	BRP rpc F16	AWP wpc F16	BWP wpc F16	17
4	ASIB 176	BSIB 176	ABSIB 2 ABSIB 2		ABFIB 0	IP	IP			FHSH rpc 0 117	1, cflg 1	FAM 174	PTRS 103	18
19	MU BA-SZE	ABSize 103	PROCO' 1'			IP	IP	FIFO	FIFO	fifo 1 wpc 1	2 0	FAM 174		19
20		MU SP	AFIB F00		BFIB F00	IP	IP			FA 2 231	FAM FH 174 SH	FAM 174	IncSP 103 c	20
21			AFIB F00		BFIB F00	IP	IP	FIFO	FIFO	FA 2 231	FA 2 231			21
22	CTDWT 117 c		OIS 231		IP	IP	IP			FA 2 231	d 195	MIDAS 176	MIDAS 176	22
23	102 c	clk0 176 FG	GETS 231 1		BLKD B 135 1	ASP 1 F16	BSP 1 F16	FIFO	FIFO	FOUT 1 176	FOUT 1 176	FOUT 1 176	MIDAS 102 BC	23
24	NEXT 172	SUBT 106 A	105		195	WSpace F16 0	HELD 231 0	IOInOut 1660	TIOA=D 161	FOUT 0 176	FOUT 0 176	FOUT 0 176 f	FTsk 113	24

C	a 11	b 26	c 39	d 55	e 70	Hold f 86	99 g	114 h	129 i	143 j	159 k	174 l	D
E	NEXT	SubT	BLK	FIN	IOIn/out	CONNECTOR	IOF	FNXT	FOUT	FTsk	DMux	E	

PEROX	Project	Reference	File	Designer	Rev	Date	Page
PARC	Dorado	DDC Board Layout	DispY26.sil	K. Pier	Cg	11/02/79	26

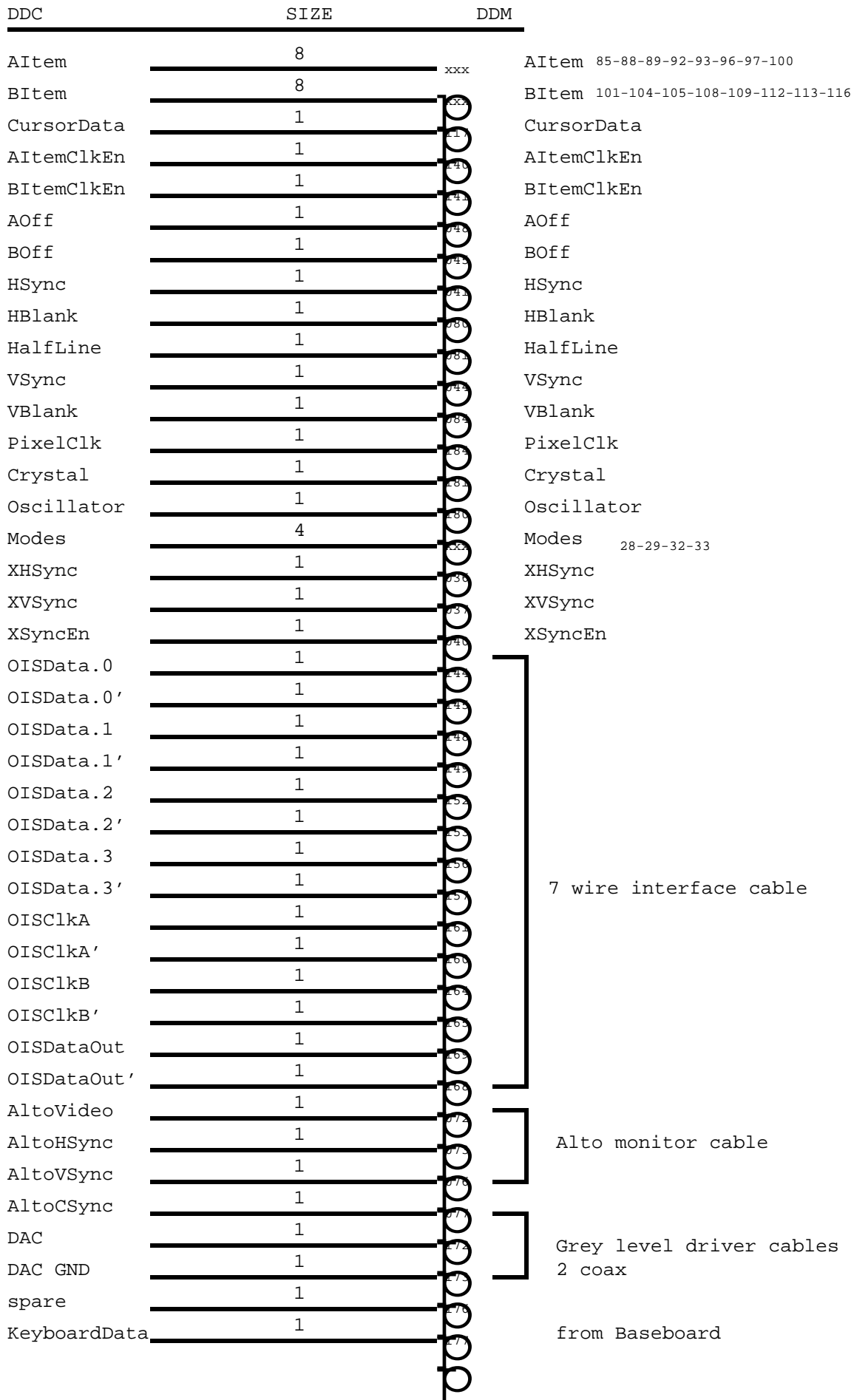
DDC Slow IO System

DEVICE	TIOA	I/O	TASK	FORMAT and COMMENTS
STATICS	377	0	DHT, EMU	See Below
NLCB	376	0	DHT, EMU	NLCBAddr.0-3,,Data.4-15
HRAM	375	0	DHT, EMU	Keep,Write,LoadAddr,0,Data.4-15
DHTFLAG	374	0	DHT	...,ANextWCBFlagSet,BNextWCBFlagSet,0 bits 13,14,15
DWT	373	0	DWT	IOFetch signal,....,Set/Clr CWCBFlagANDClrNWCBFlag bit 11, , bit 15
MiniMixer	372	0	DHT, EMU	address,,data always writes MiniMixer RAM
STATUS	370	I	DHT, EMU	Selected muffler input returned in bit 15
PIXELCLK	367	0	DHT, EMU	Pixel clock rate located on mixer board
MIXER	366	0	DHT, EMU	Keep,Write,LoadAddr,0,Data.4-15 located on mixer board

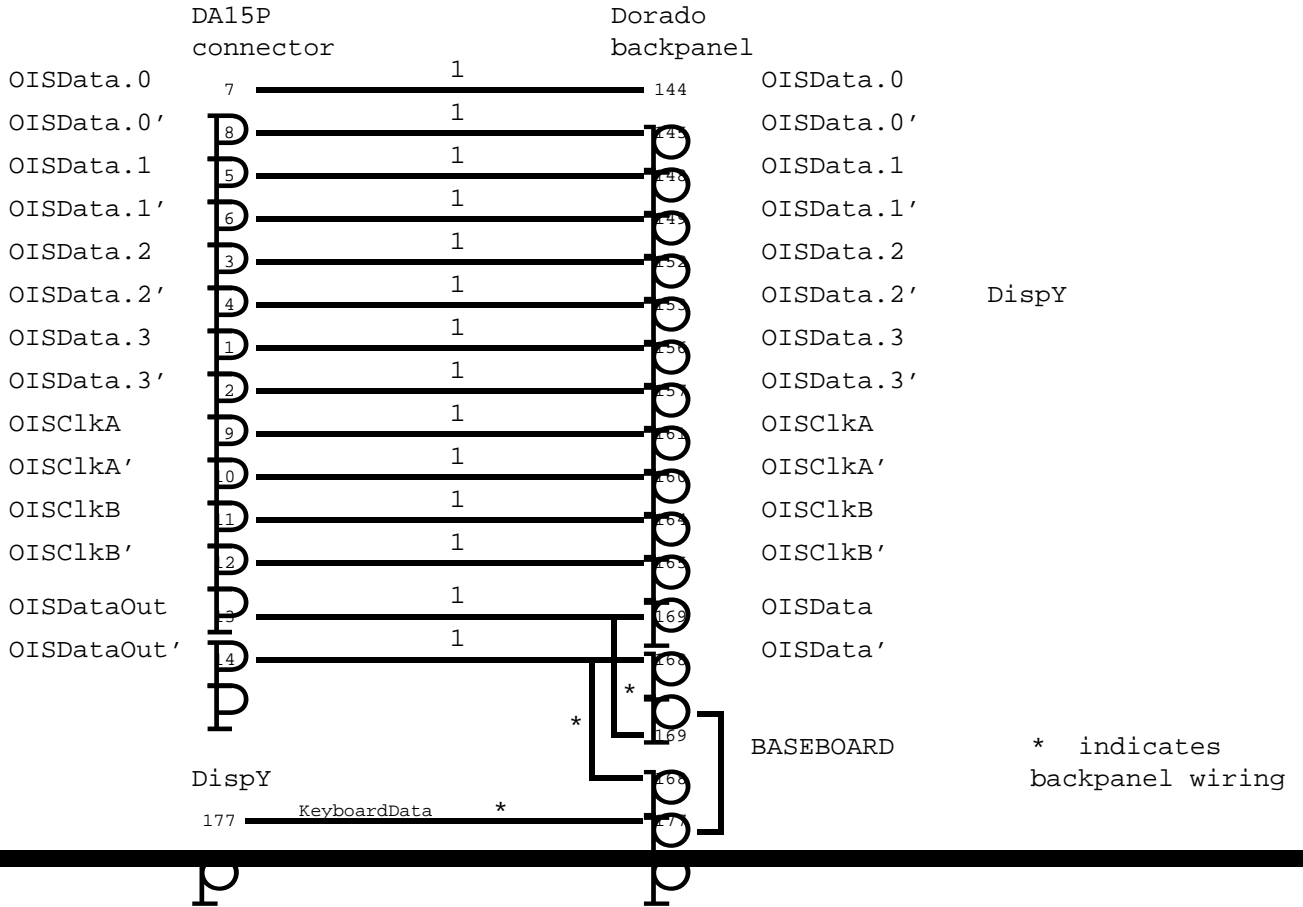
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
X	Muff Addr .05	Muff Addr .06	Muff Addr .07	Muff Addr .08	Muff Addr .09	Muff Addr .10	Muff Addr .11	XSync Enable	Fake Pixel Clock	Use Fake Pixel Clock	Unused	Unused	Unused	DWT Shut Up	DHT Shut Up- Reset DDC	Statics
NLCB Addr .00	NLCB Addr .01	NLCB Addr .02	NLCB Addr .03	NLCB Data 12 bits												NLCB
Keep HRam'	Write HRam'	Load Hram Addr				Addr.0	Addr.1	Addr.2	Addr.3	Addr.4	Addr.5	Addr.6	Addr.7	Addr.8	Addr.9	HRAM
													HSync	HBlank	Half Line	
													Set BNext WCB Flag	Set ANext WCB Flag	Must Be 0	DHTFLAG
										IOFetch signal	Must Be 0	Must Be 0	Must Be 0	Set/ Clr Cur WCB Flag	DWT	
Address.0-7								Data.0-7								MiniMixer
								ClkRate.0-7								PIXELCLK
Keep Mixer'	Write Mixer'	Load Mixer Addr	X		Addr.0	Addr.1	Addr.2	Addr.3	Addr.4	Addr.5	Addr.6	Addr.7	Addr.8	Addr.9	Hi/Lo select	MIXER
												Mixer Data 12 bits				

Next Line Control Block Format

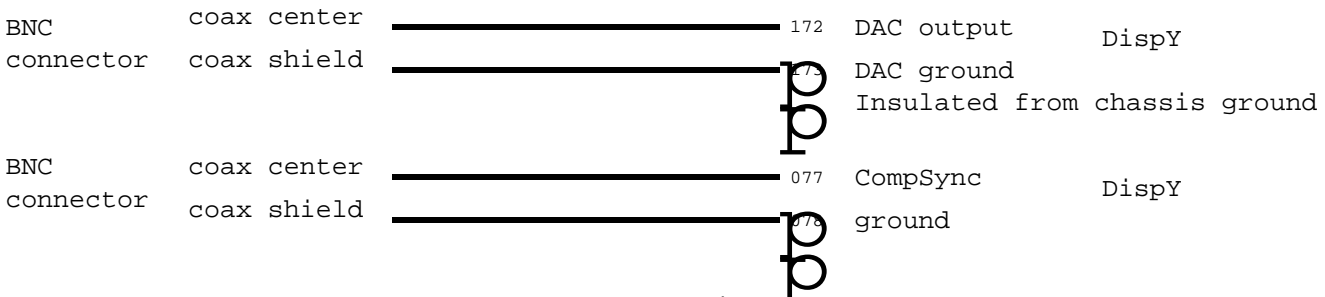
Address	Name	Format
0	VCW vertical control word	0...0, VBlank, VSync, EvenField
1	AMargin	LMarg[00..11]
2	AWidth	Width[00..11]
3	AFifoAddr	FifoAddr[0..7] *must be even
4	AScan	0...0,Polarity,Resolution[0..1],Size8,Size4,Size2,S
5	ModeControl	0..0,24Bit,ABypass,BByPass,A8B2
6	reserved	
7	reserved	
10	unused	
11	BMargin	LMarg[00..11]
12	BWidth	Width[00..11]
13	BFifoAddr	FifoAddr[0..7] *must be even
14	BScan	0...0,Polarity,Resolution[0..1],Size8,Size4,Size2,S
15	CursorX	CursorXCount[0..11]
16	CursorLo	CursorLoByte[4..11]
17	CursorHi	CursorHiByte[4..11]



Seven Wire Interface Cable



Grey level cabling



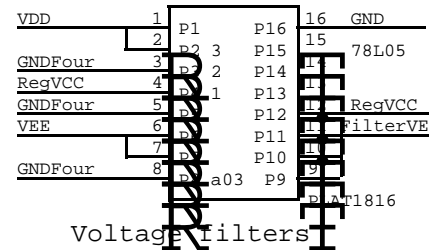
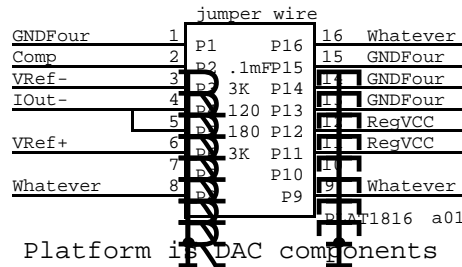
Note: CompSync currently appears in the Alto cable. The Alto display doesn't make use of CompSync. We could simply replace the CompSync pair in the Alto cable with the coax. We then have the capability to attach a local Alto type display and a remote display.

Six BNC connectors mounted on the "square" side. Cabling not yet defined.

1. Plug an MC10318 D/A converter into location a02 .
2. Plat1816 in locations a01 and b02 are discrete components, shown below.
3. SIPs are 100 ohm terminator package with legs broken:

location	break legs	See page 23.
g41	3,4,5	
g42	none	
k51	4,5	
k52	3	

4. SIPs at locations d42 and e42 are 220 ohm value instead of 100 ohm.
5. Crystal oscillator K1115A, location a05, value 20 MHz.



1	jumper wire	16
2	0.1mFarad	15
3	3 KOhm	14
4	120 Ohm	13
5	180 Ohm	12
6	3 KOhm	11
7		10
8		09

location a01

1	0.1mFarad	16
2	in ³	78L05 15
3	gnd	2 14
4	out ¹	13
5	0.1mFarad	12
6	12 mHnry	11
7	2 ohm	10
8	+ 22 mF	09

location a03

See Page 20.

