

Dandelion Central Processor
with
Extended Control Store and DES Encryption

Revision W

00	Table of Contents
01	IC Layout (Stitchweld)
02	IC Layout (Stitchweld) - Function Placement
03	PROM Index & Font 4 Macros
04	2901 ALU
05	Look-ahead, Shift Ends, Cin
06	SU Registers
07	RH Register, stackP
08	IB - Instruction Prefetch Buffer
09	XBus: LRotn, RH, ZeroHigh X
10	XBus: IB, Constants, ErrIntstackP
11	Microinstruction register
12	Microinstruction Decoding I
13	Microinstruction Decoding II
14	Dispatch/Branch
15	pNIA, pTC (Branching)
16	TPC, TC, Link
17	Schedule, Switch, Tasks
18	Error, Emulator, Kernel Proms
19	Clocks, Wait Logic
20	IOP Data In Buffers, Control Store Write Logic
21	Control Store Bank 0
22	Control Store Bank 1
23	Control Store Bank 2
24	Control Store Bank 3
25	Control Store Parity Checking Logic
26	IOP Data Out Multiplexor/Drivers
27	Bank Register, NIA Register
28	CS Address Terminators & Pullups
29	History Buffer Connector Pinout
30	Gate Allocation & Spare Parts
31	DES Encryption Logic
32	DES Data Sheet and Clock Generator
33	DES Finite State Machines
34	DES FSM State Diagrams
35	DES Timing Diagrams
36	DES Timing - Overview & Pipelining

Notes:

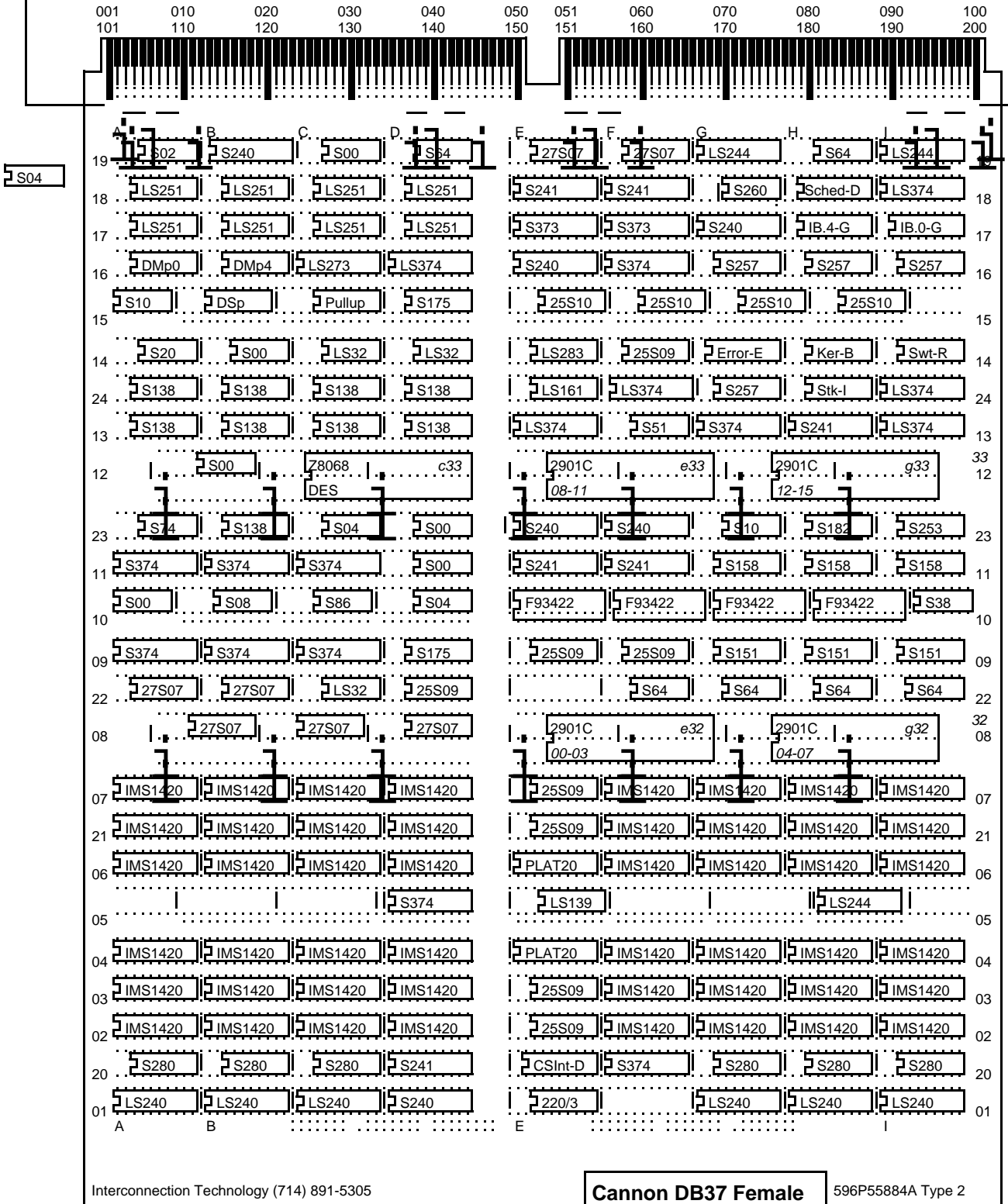
The drawings are filed in: [Indigo]<Dandelion>CPE>Drawings>sCPE*.sil
and in: [Indigo]<Dandelion>CPE>Archive>sCPE-W-DrawingsOnly.dmsil
Print press file is in [Indigo]<Dandelion>CPE>Archive>sCPE-W.press

Retrieve the files and tools using BringOver, as follows:
BringOver /a [Indigo]<Dandelion>CPE>DfFiles>sCPE-W.df

Refer any enquiries to David Boggs:
Boggs.PA Phone: 415-494-4421
Intelnet: 8-923-4421

Sil fonts in User.cm
0 = Helvetica10B Helvetica10N
1 = Helvetica7B Helvetica7B
3 = Gates32
5 = Sil.Ib5 from ED/SDD
6 = Sil.Ib6 from ED/SDD
7 = Sil.Ib7 from ED/SDD
8 = CAS.Ib8 from ED/SDD
9 = Sil.Ib9 from ED/SDD

NB: Disconnect VDD and VEE supplies. This board only uses VCC



Warning: This drawing contains font 4 macros! AND must be read via "SII/1 sCPE01.sil"

XEROX	Project	Reference	File	Designer	Rev	Date	Page
PARC-CSL	CPE	High-Density Stitchweld Layout	sCPE01.sil	TonyWest.PA	X	4/8/83	01

001	010	020	030	040	050
101	110	120	130	140	150

051	060	070	080	090	100
151	160	170	180	190	200

19	S02	S240	S00	S64
18	LS251	LS251	LS251	LS251
17	LS251	LS251	LS251	LS251
16	DesMpProm.0	DesMpProm.4	LS273	LS374
15	S10	DesSpProm	Pullups	S175
14	S20	S00	LS32	LS32
24	S138	S138	S138	S138
13	S138	S138	S138	S138
12	S00		Z8068	c33 Des Processor
23	S74	S138	S04	S00
11	S374	S374	S374	S00
10	S00	S08	S86	S04
09	S374	S374	S374	S175
22	27S07	27S07	LS32	25S09
08	27S07	27S07	27S07	
07	1420	1420	1420	1420
21	1420	1420	1420	1420
06	1420	1420	1420	1420
05	SPARE	SPARE	SPARE	S374
04	1420	1420	1420	1420
03	1420	1420	1420	1420
02	1420	1420	1420	1420
20	S280	S280	S280	S241
01	LS240	LS240	LS240	S240

27S07	27S07	LS244	S64	LS244
S241	S241	S260	ScheduleProm	LS374
S373	S373	S240	IBProm 4	IBProm 0
S240	S374	S257	S257	S257
25S10	25S10	25S10	25S10	SPARE
LS283	25S09	Error Prom	KernelPCProm	Switch Prom
LS161	LS374	S257	Stack Prom	LS374
LS374	S51	S374	S241	LS374
2901C	e33		2901C	g33
S240	S240	S10	S182	S253
S241	S241	S158	S158	S158
F93422	F93422	F93422	F93422	S38
25S09	25S09	S151	S151	S151
LS374	S64	S64	S64	S64
2901C	e32		2901C	g32
25S09	1420	1420	1420	1420
25S09	1420	1420	1420	1420
Platform 20	1420	1420	1420	1420
LS139	SPARE	SPARE	LS244	SPARE
Platform 20	1420	1420	1420	1420
25S09	1420	1420	1420	1420
25S09	1420	1420	1420	1420
CSInt Prom	S374	S280	S280	S280
220-330 res	SPARE	LS240	LS240	LS240

Cannon DB-37 Female

Cannon DB-37 Female

Rev X in bold

Prom Name	Rev	Part No.	Location	Page	Comments	
SwitchProm	R	F93427	i14	14	New Rev for 16K Control Store CP	256 x 4
KernPC16Prom	B	F93427	h14	15	Standard	256 x 4
CSIntProm	D	F93453	e20	22	Standard	1K x 4
StackVirtProm	I	F93427	h24	15	Standard	256 x 4
ScheduleProm	D	F93453	h18	14	Standard	1K x 4
ErrorProm	E	F93453	g14	15	Standard	1K x 4
IBProm-PC.0	G	F93453	i17	05	Standard	1K x 4
IBProm-PC.4	G	F93453	h17	05	Standard	1K x 4
DesMpProm.0	B	F93427	a16	33	Added to control DES logic	256 x 4
DesMpProm.4	B	F93427	b16	33	Added to control DES logic	256 x 4
DesSpProm	A	F93427	b15	33	Added to control DES logic	256 x 4

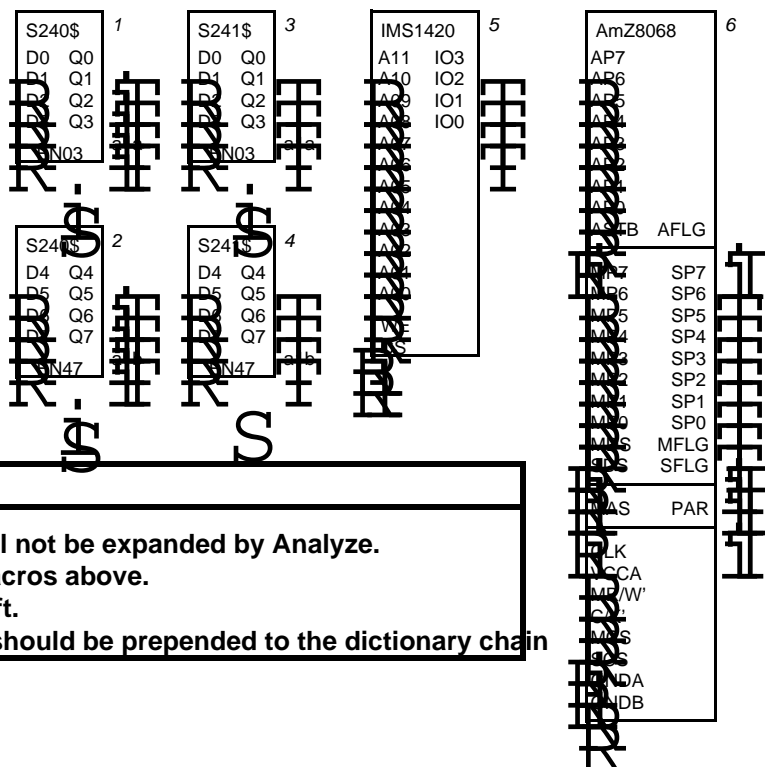
Prom files are stored on [Indigo]<Dandelion>CPE>Proms>*

Bringover /a [Indigo]<Dandelion>CPE>DfFiles>Proms.df to fetch all files, sources, tools, etc.

Labels:

Switch i14 Rev-R	KernPC16 h14 Rev-B	CSInt e20 Rev-D	StackVirt h24 Rev-I	Schedule h18 Rev-D	Error g14 Rev-E	IB-PC.0 i17 Rev-G	IB-PC.4 h17 Rev-G	DesMp.0 a16 Rev-B	DesMp.4 b16 Rev-B	DesSp b15 Rev-A
Switch i14 Rev-R	KernPC16 h14 Rev-B	CSInt e20 Rev-D	StackVirt h24 Rev-I	Schedule h18 Rev-D	Error g14 Rev-E	IB-PC.0 i17 Rev-G	IB-PC.4 h17 Rev-G	DesMp.0 a16 Rev-B	DesMp.4 b16 Rev-B	DesSp b15 Rev-A
Switch i14 Rev-R	KernPC16 h14 Rev-B	CSInt e20 Rev-D	StackVirt h24 Rev-I	Schedule h18 Rev-D	Error g14 Rev-E	IB-PC.0 i17 Rev-G	IB-PC.4 h17 Rev-G	DesMp.0 a16 Rev-B	DesMp.4 b16 Rev-B	DesSp b15 Rev-A
Switch i14 Rev-R	KernPC16 h14 Rev-B	CSInt e20 Rev-D	StackVirt h24 Rev-I	Schedule h18 Rev-D	Error g14 Rev-E	IB-PC.0 i17 Rev-G	IB-PC.4 h17 Rev-G	DesMp.0 a16 Rev-B	DesMp.4 b16 Rev-B	DesSp b15 Rev-A

Font 4 Macros



Important Notes:

Only macros 0-9 are valid component names and will not be expanded by Analyze.

Some of these drawings contain instances of the macros above.

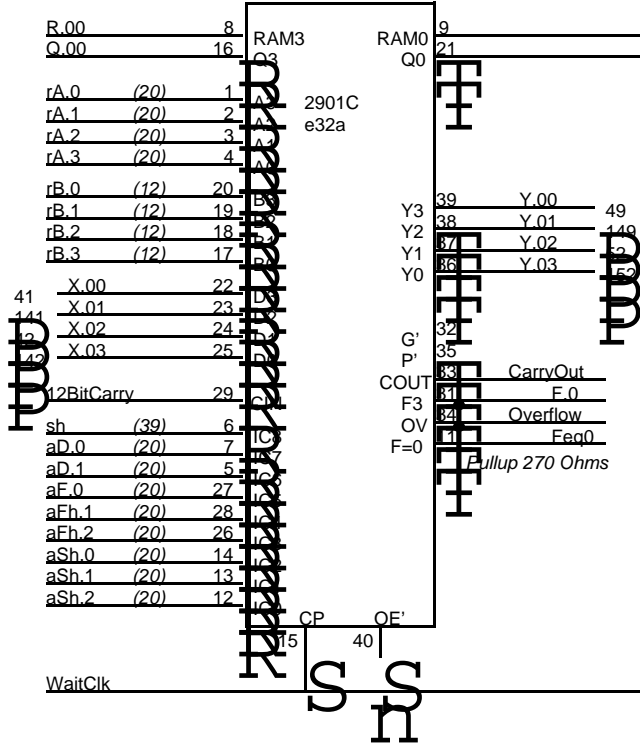
Those that do have a warning on them, see below left.

There is a corresponding sCPEDict.Analyze, which should be prepended to the dictionary chain

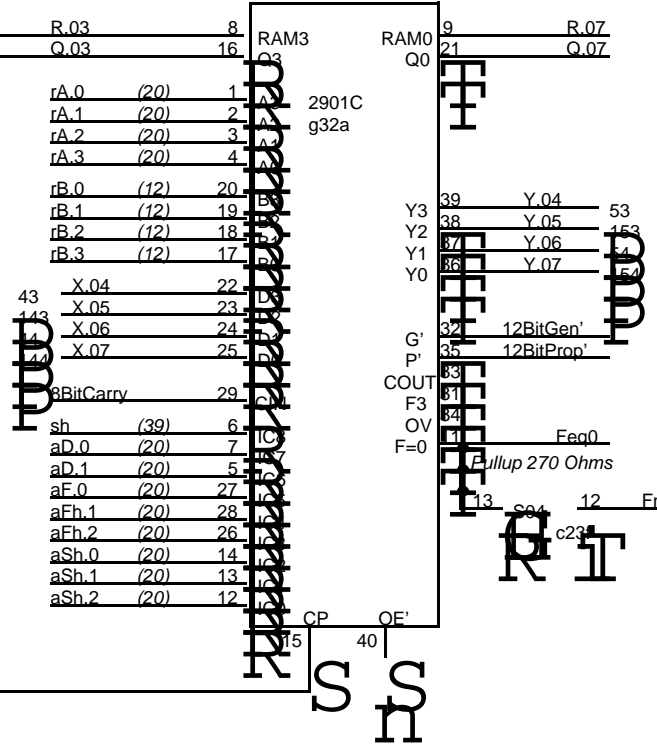
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project CPE	Reference Prom Index & Font 4 Macros	File sCPE03.sil	Designer TonyWest.PA	Rev X	Date 1/24/83	Page 03
-------------------	----------------	---	--------------------	-------------------------	----------	-----------------	------------

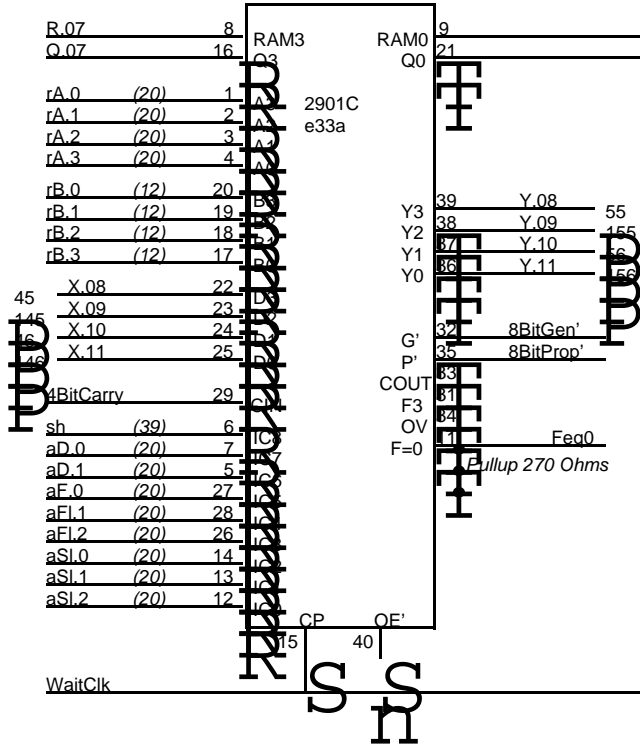
[00-03]



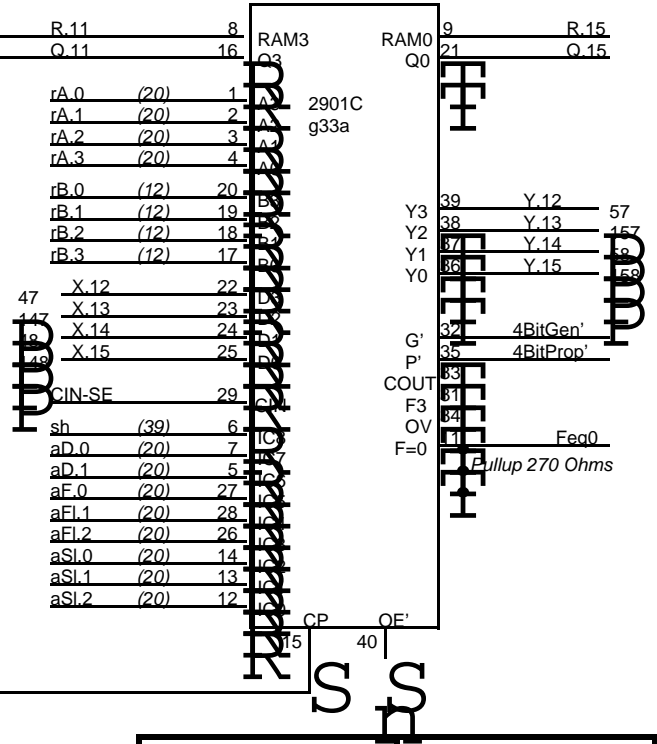
[04-07]



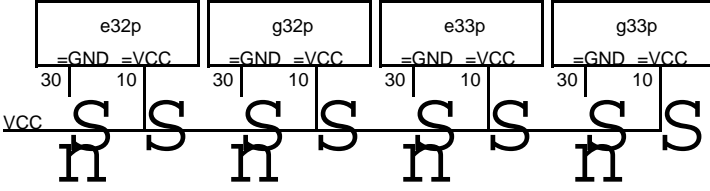
[08-11]



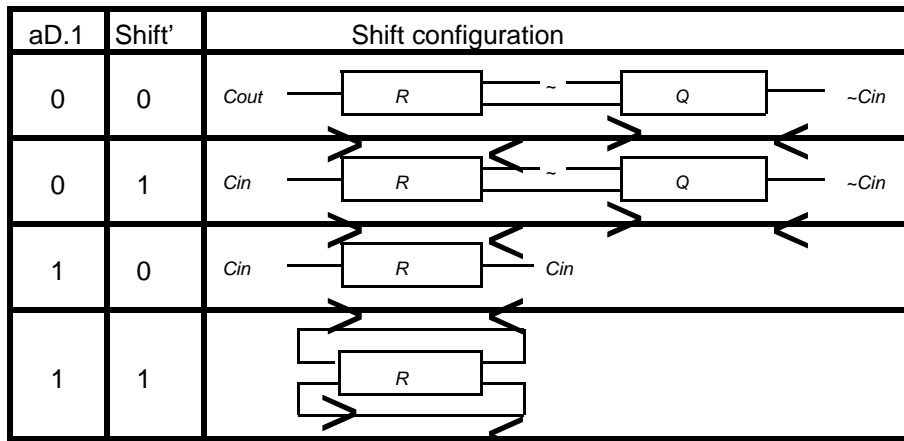
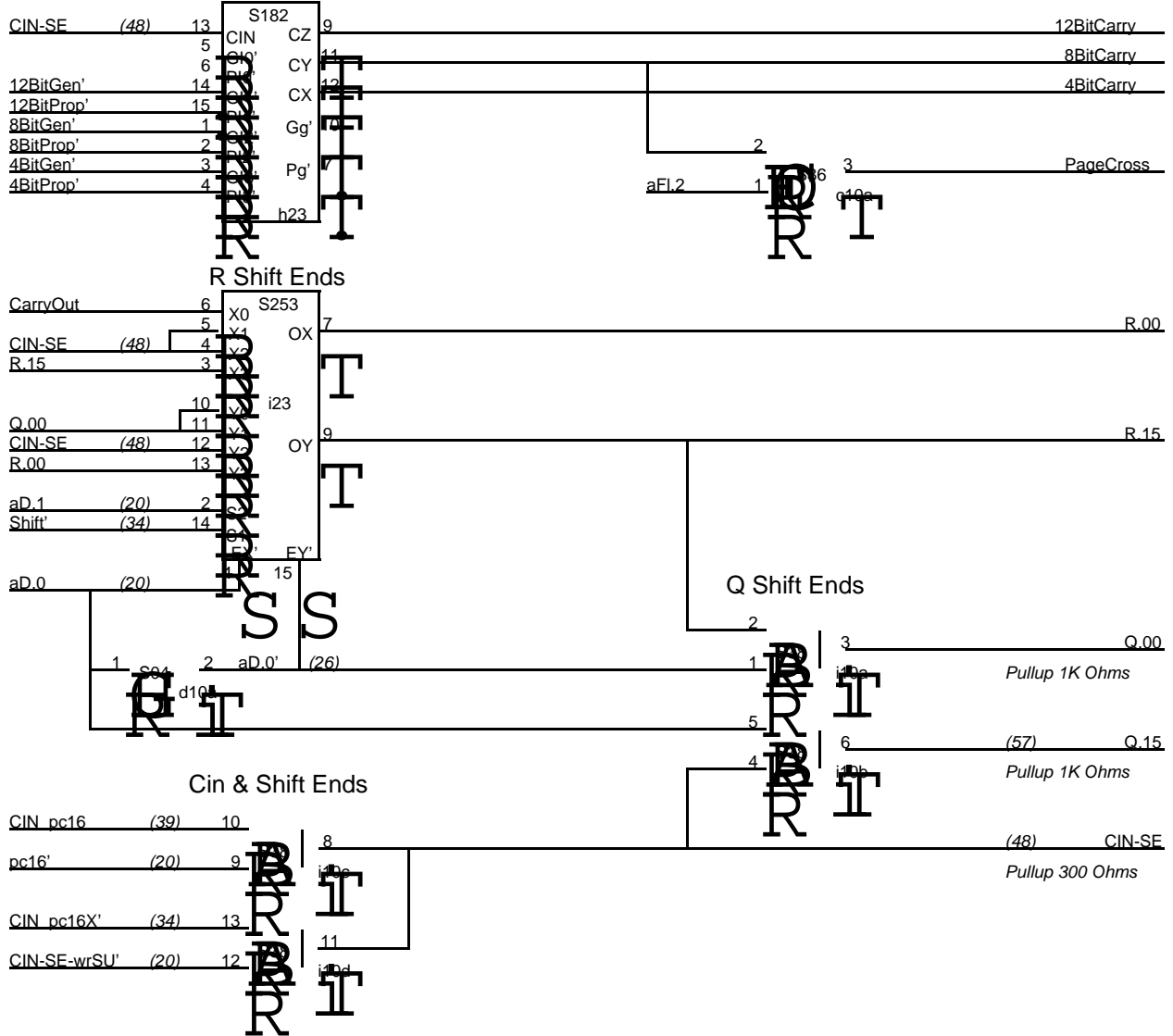
[12-15]



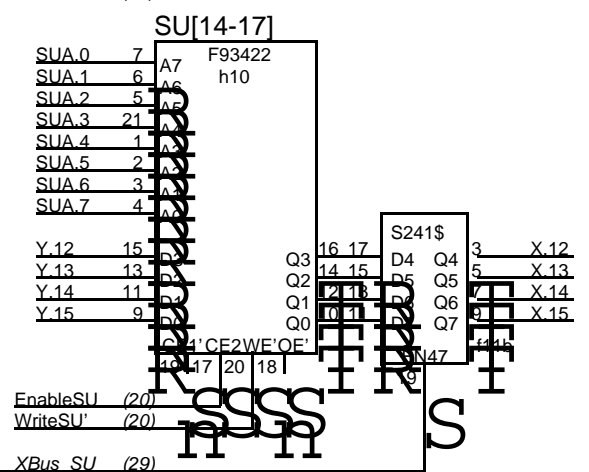
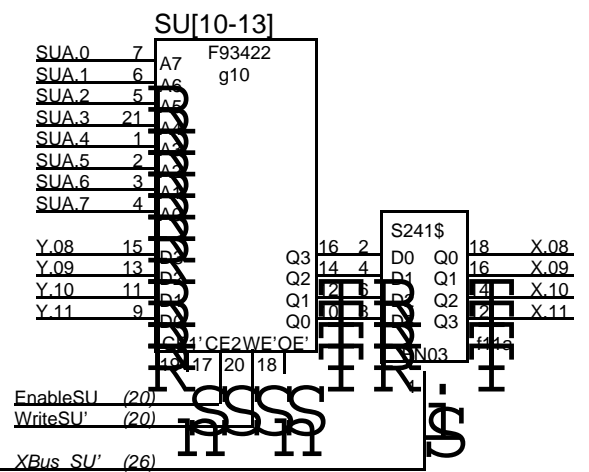
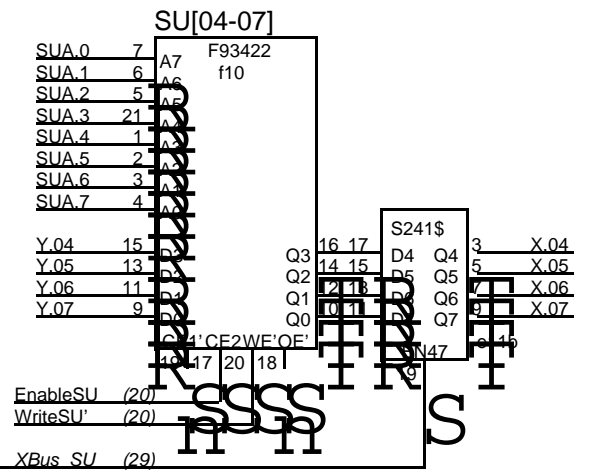
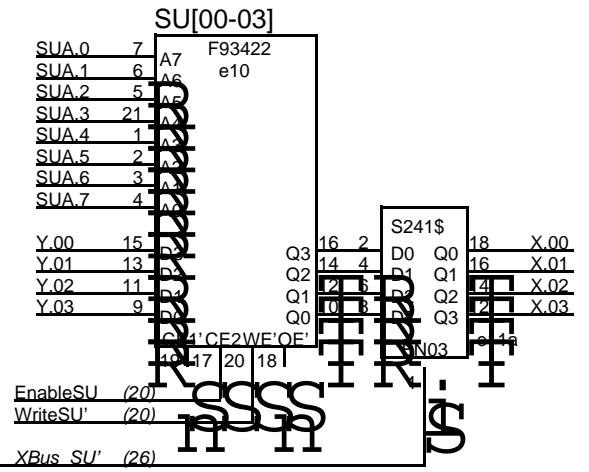
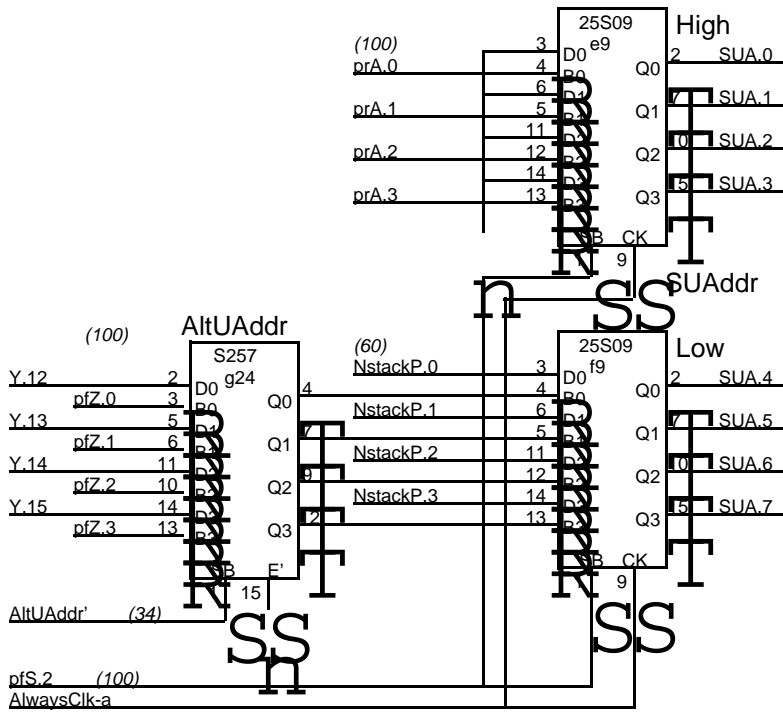
2901C Timing		As of 10/05/82	
Setup	sh.,aD	10	
Setup	aF	30	
Setup	aS	30	
rA	to G,P	37	rA to Y 40
D	to G,P	30	D to Y 30
			Cn to Y 22
rA	to Cout	40	aS to Y 35
D	to Cout	30	aF to Y 35
Cin	to Cout	20	aD to Y 25



2901 Carry Lookahead



aD.0 = 0 implies right shift



SU X-bus disable

15[3] ^ to CIN-SE-wrSU (tPLH)
 30 Output Disable
 10 X-bus

55[3] = 58 nS

XBus _ SU = max(75,60) nS

17[3] ^ to SUAddr
 45 tAA
 10 X-bus

72[3] = 75 nS

17[3] ^ to CIN-SE-wrSU/EnableSU
 30 F93422 OE'/CE2 to X-bus
 10 X-bus

57[3]=60 nS

SU write setup

5[1] Data setup
 39 WE

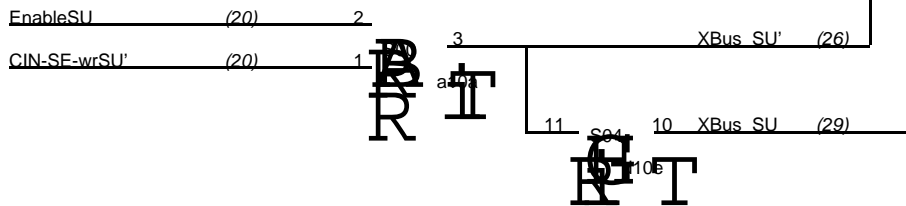
44[1] = 45 nS

F93422 data t-hold = 5 nS

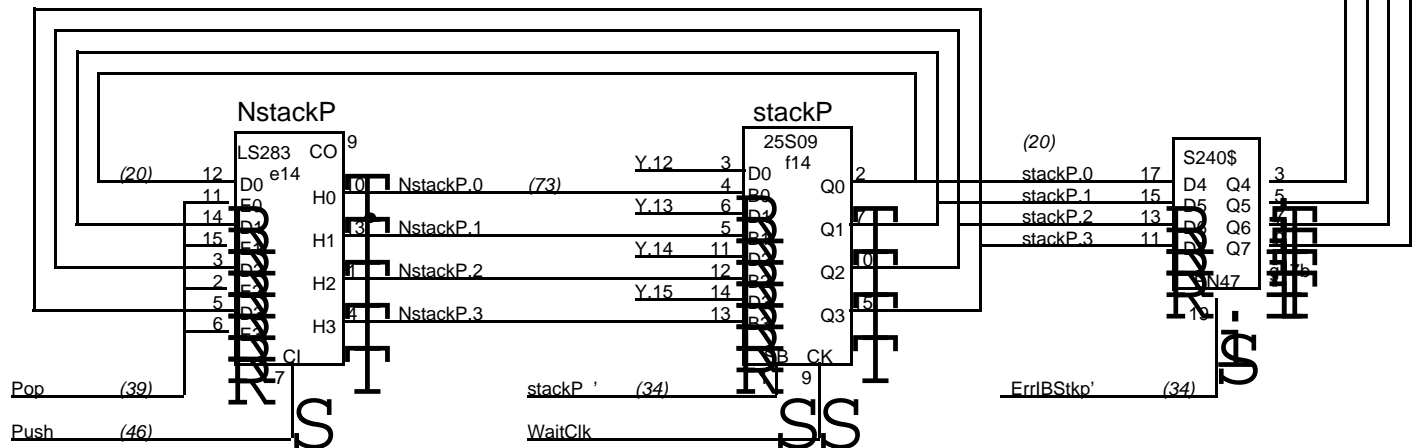
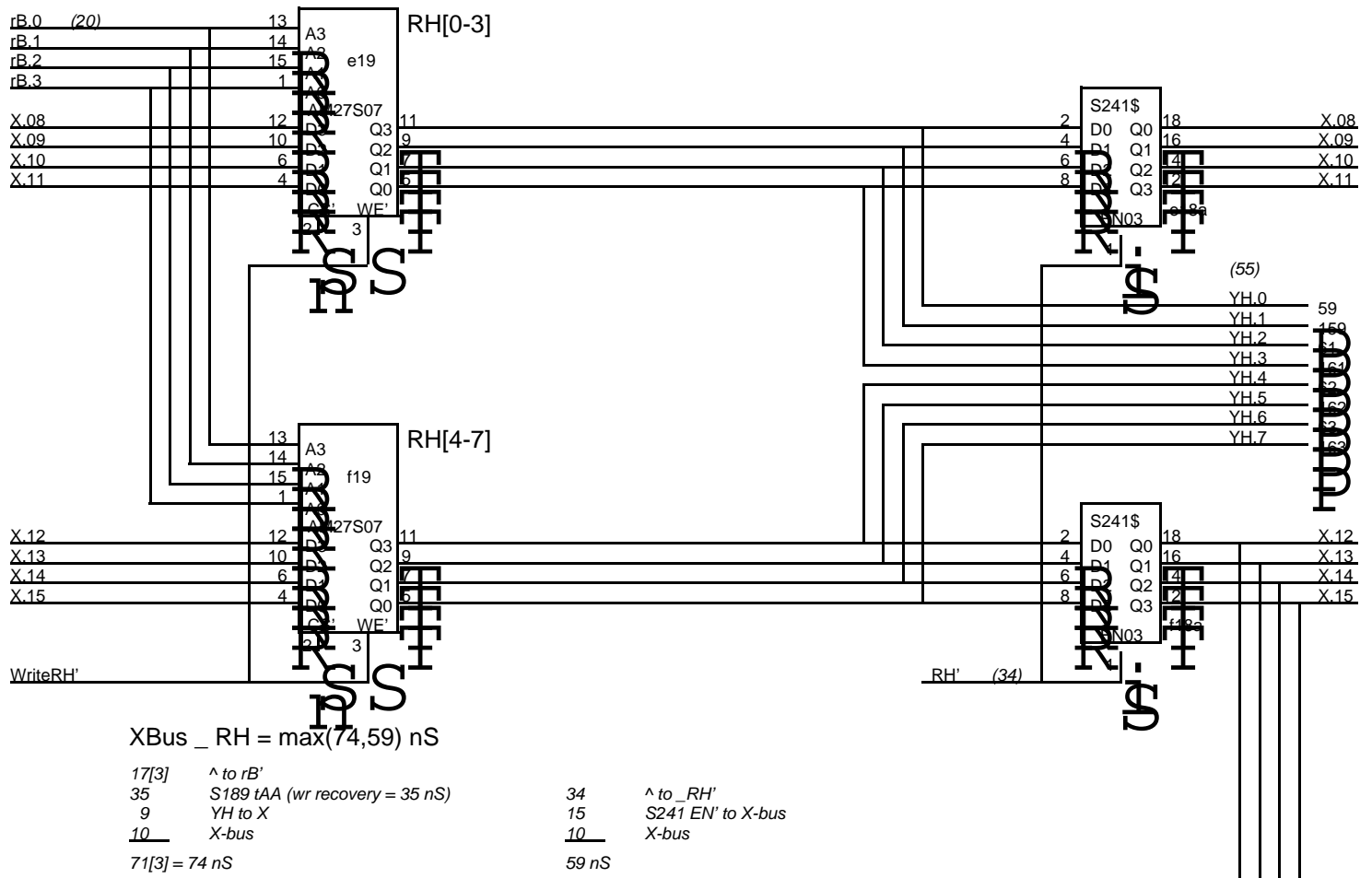
AltUAddr setup

5[1] 25S09 setup
 8[1] Y-> pU

13[2] = 15 nS (26 if LS257)



Warning: This drawing contains font 4 macros!



Push Timing

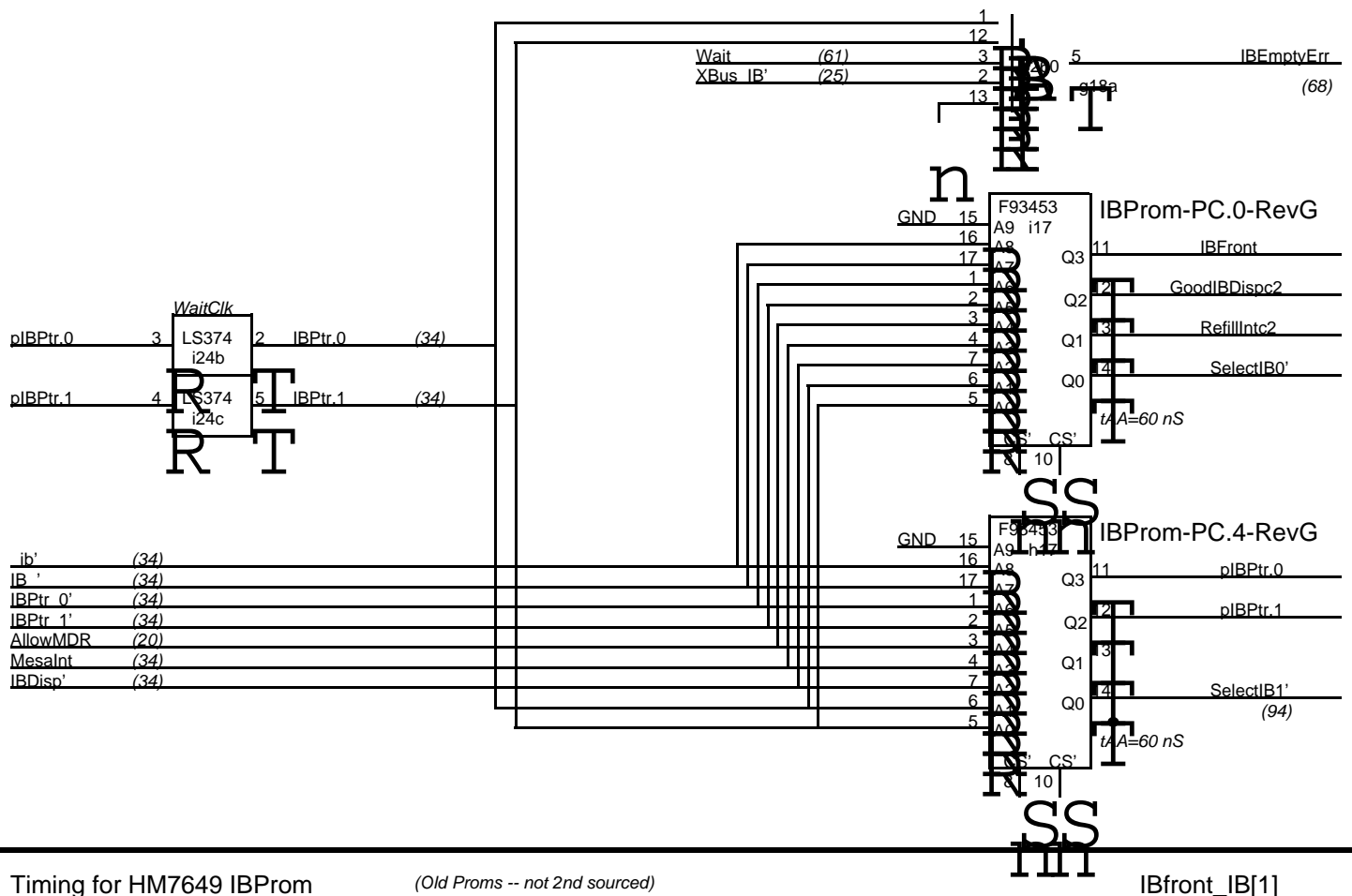
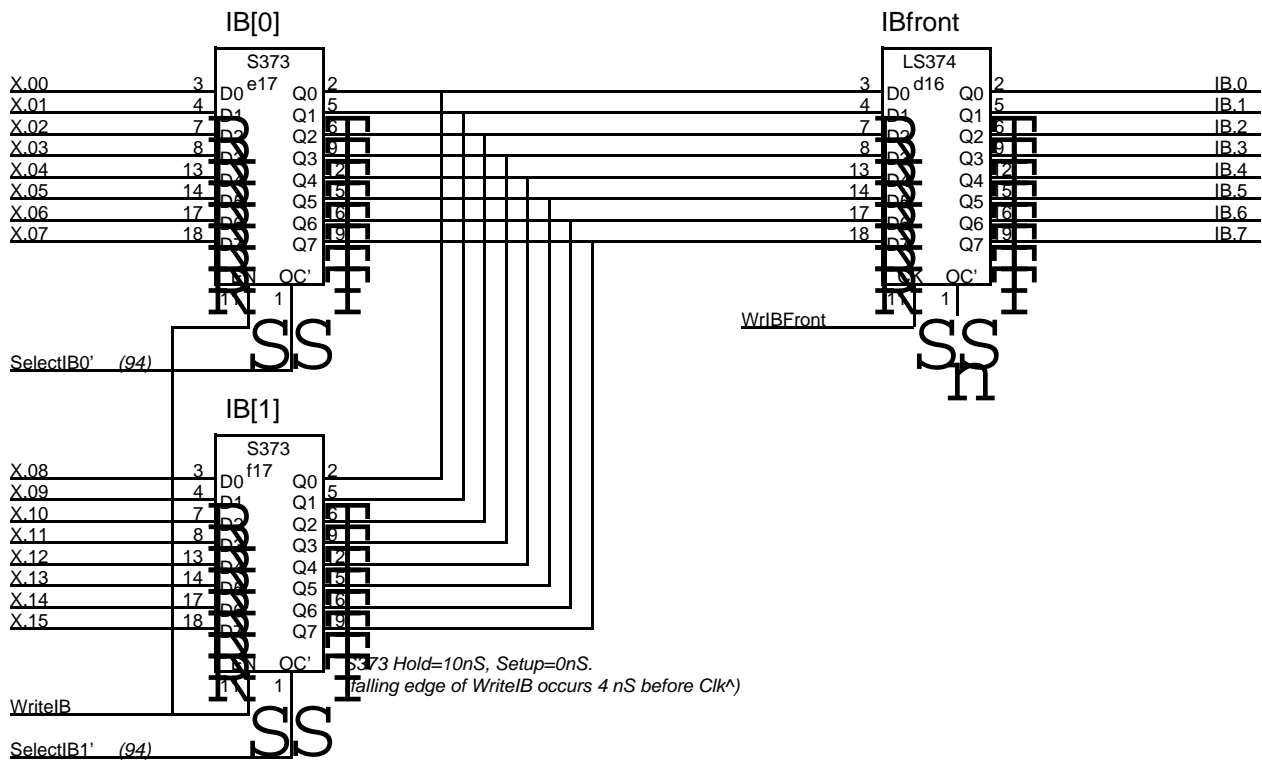
46 ^ to Push
 24[3] Push to NstackP
 5[1] 25S09 setup
 75[4] = 79 nS

XBus _ stackP = max(59, 38) nS

17[3] ^ to stackP 34 ^ to _ErrIntstackP'
 7 S240 data to X-bus 15 S240 EN' to X-bus
 10 X-bus 10 X-bus
 34[3] = 38 nS 59 nS

Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project CPE	RH, stackP	File sCPE07.sil	Designer Garner.PA	Rev X	Date 1/14/83	Page 07
-------------------	----------------	------------	--------------------	-----------------------	----------	-----------------	------------

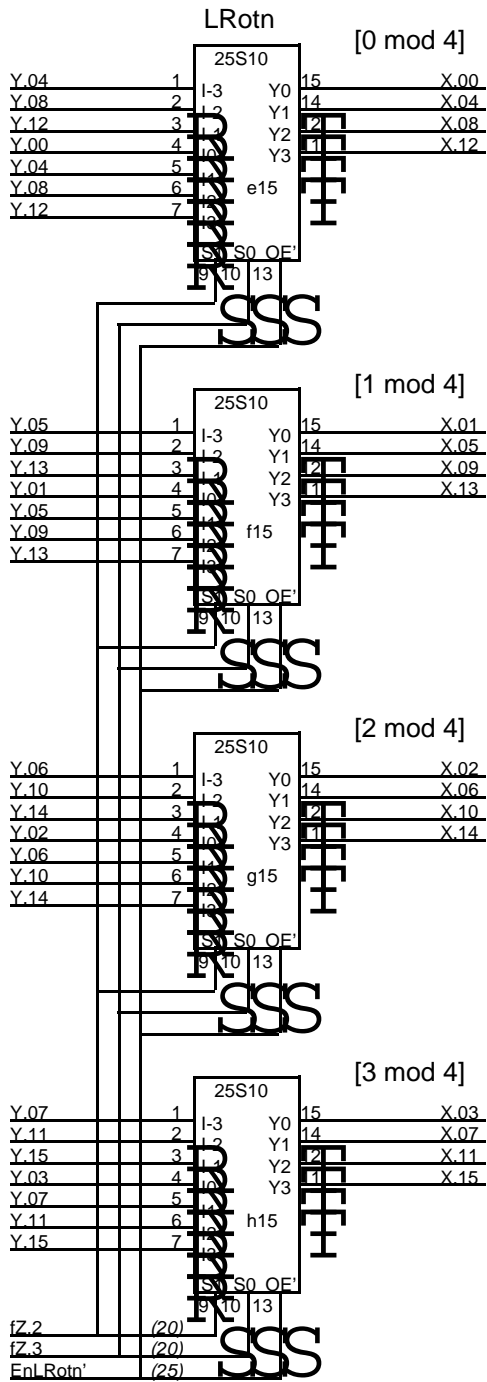


Timing for HM7649 IBProm

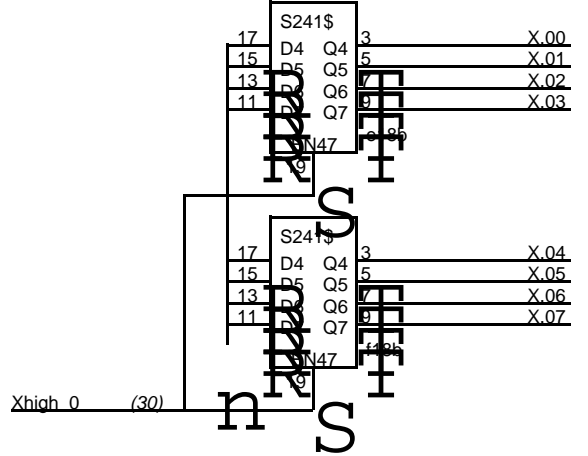
(Old Proms -- not 2nd sourced)

IBfront _ Xbus = (x+37, x+36) nS

x	Xbus to IB	x	Xbus to IB	94	WriteIB rises	34	^ to IBPtr_1'
43	WriteIB rises 43 nS before end of cycle	13[1]	S373 Data to NB	18[2]	S373 EN to NB	60	t _{AA}
-6	Difference between S373 "EN to Q" and "Data to Q" =	20[2]	LS374 setup	20[2]	LS374 setup	18[2]	SelectIB1' to NB
x+37 nS	18[2] - 13[1] = 6 nS. Data can arrive 6 nS after WriteIB goes high.	x+36 nS		132[4]=136 nS		20[2]	LS374 setup
						132[4]=136 nS	



fZ.2	fZ.3	Rotate
0	0	Left 0
0	1	Left 12
1	0	Left 8
1	1	Left 4



Zero disable X-bus

- 30 ^ to Xhigh_0
- 15 S241 EN to X-bus
- 10 X-bus
- 55 nS

Xbus[0-7] _ 0

- 30 ^ to Xhigh_0
- 15 S241 OE
- 10 X-bus
- 55 nS

Xbus _ Y LRotn = max(y+22, 56, 50) nS

- y ^ to Y bus
- 12 25S10 data in to out
- 10 X-bus
- y + 22 nS

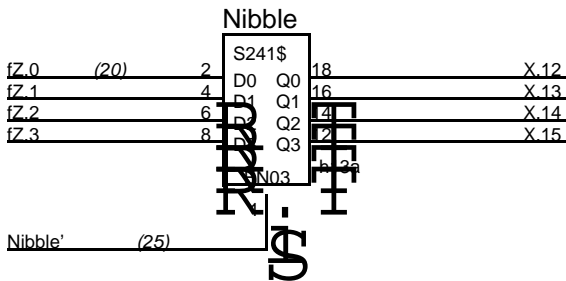
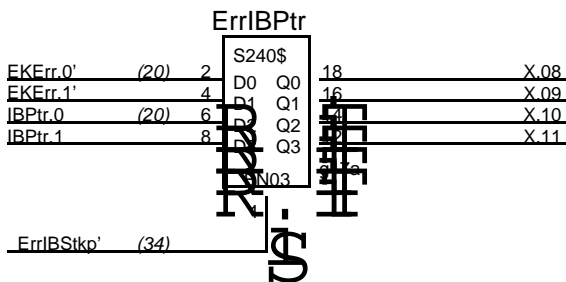
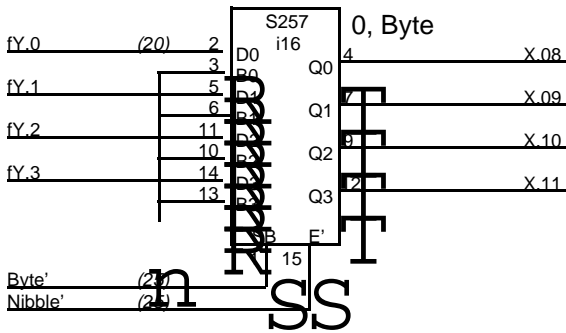
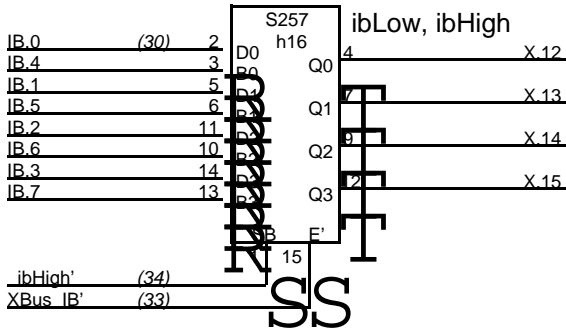
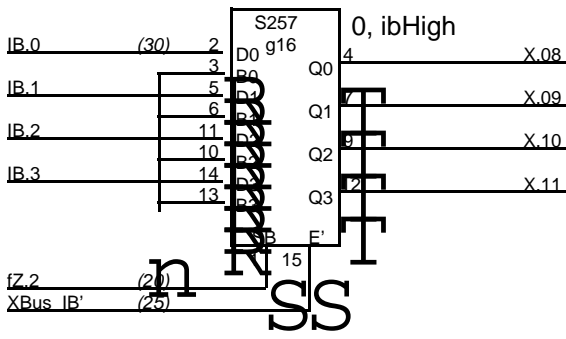
- 25 ^ to EnLRotn'
- 21 25S10 OE
- 10 X-bus
- 56 nS

- 20 ^ to fZ.2
- 20 25S10 Select to X-bus
- 10 X-bus
- 50 nS

LRotn disable X-bus

- 25 ^ to EnLRotn'
- 15 25S10 OE' to X-bus
- 10 X-bus
- 50 nS

Warning: This drawing contains font 4 macros!



IB disable X-bus

25 ^ to XBus_IB'
 14 S257 E' to X-bus
 10 X-bus
 49 nS

Byte disable X-bus

25 ^ to Nibble'
 14 S257 E' to X-bus
 10 X-bus
 49 nS

Nibble disable X-bus

25 ^ to Nibble'
 15 S241 EN' to X-bus
 10 X-bus
 50 nS

Xbus_IB=max(56,56,59) nS

34[4] ^ to IB
 8 S257 data to Xbus
 10 X-bus
 52[4] = 56 nS

25 ^ to Xbus_IB'
 21 S257 E' to Xbus
 10 X-bus
 56 nS

34 ^ to _ibHigh'
 15 S257 SB to Xbus
 10 X-bus
 59 nS

Xbus _ Nibble = max(39, 50) nS

20 ^ to fZ
 9 S241 data to X-bus
 10 X-bus
 39 nS

25 ^ to Nibble'
 15 S241 EN' to X-bus
 10 X-bus
 50 nS

Xbus _ Byte = max(38, 56,50) nS

20 ^ to fY
 8 S257 data to X-bus
 10 X-bus
 38 nS

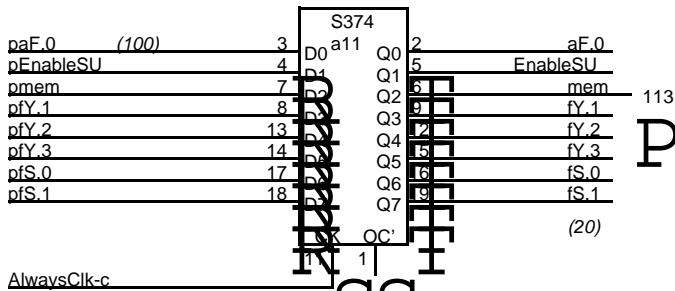
25 ^ to Nibble'
 21 S257 E' to X-bus
 10 X-bus
 56 nS

25 ^ to Byte'
 15 S257 SB to Xbus
 10 X-bus
 50 nS

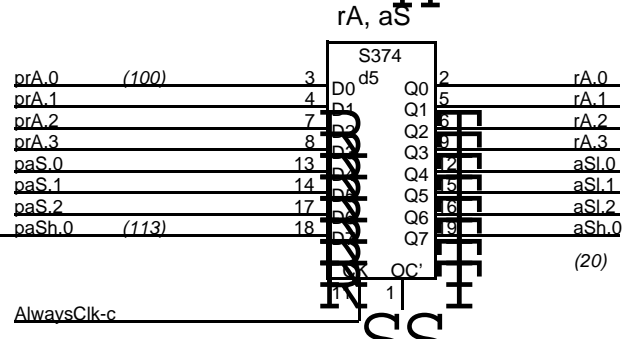
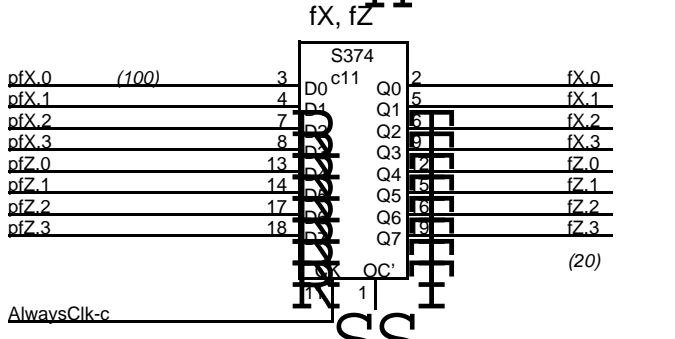
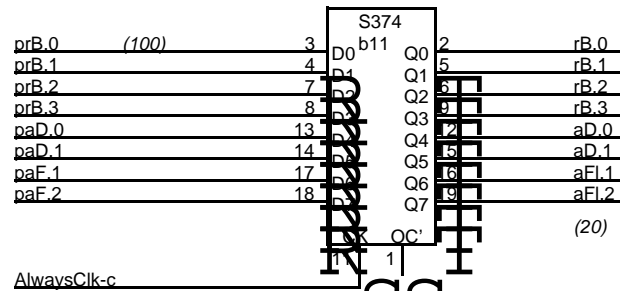
See stackP timings for ErrIBPtr

Warning: This drawing contains font 4 macros!

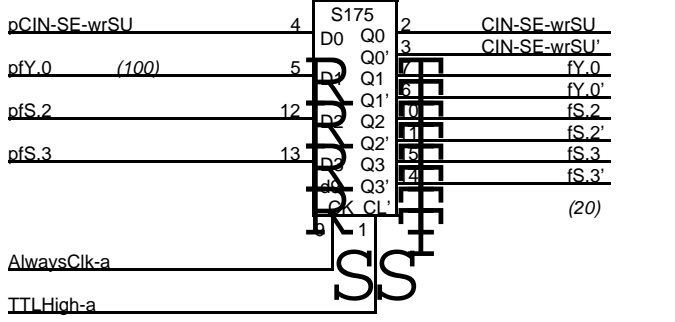
aF.0, EnSU, mem, fY, fS



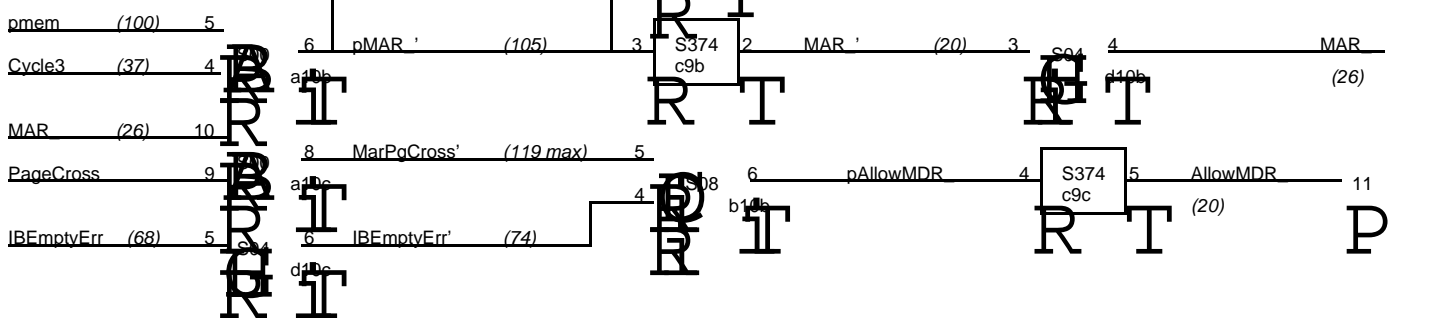
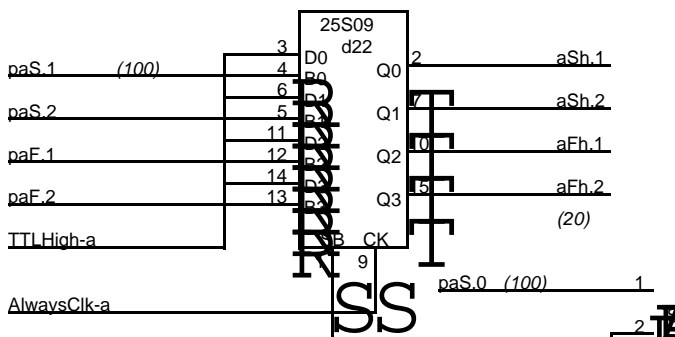
rB, aD, aFI

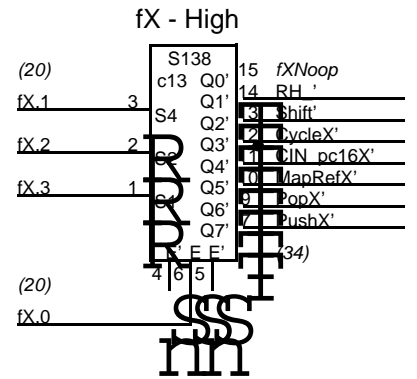
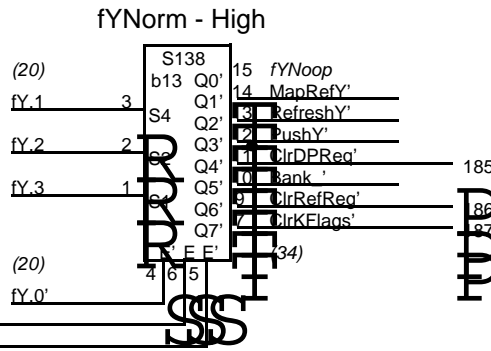
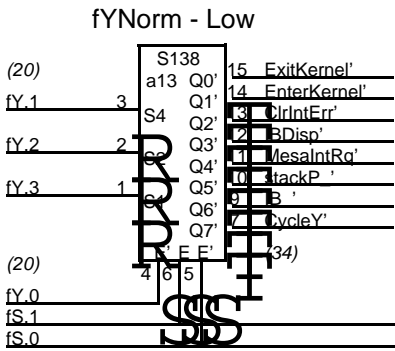


Cin, fY.0, fS



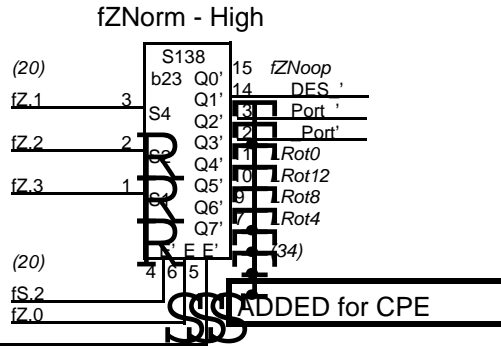
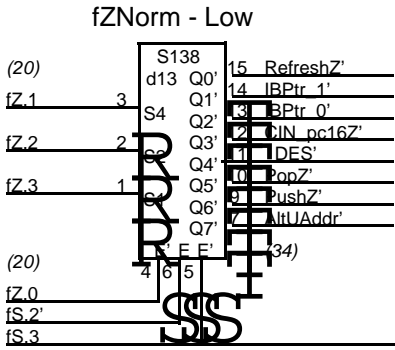
aSh, aFh





Bank_ replaces CrlrIOPReq', which was connected to 184 on backplane

fX - Low is pCall/pRet



There are 2 spare fZ decodes available for future expansion

Notes on 16K CP additions:

Note that Bank_ is fY=D, not fZ=4, as stated in the Dandelion Hardware Manual!

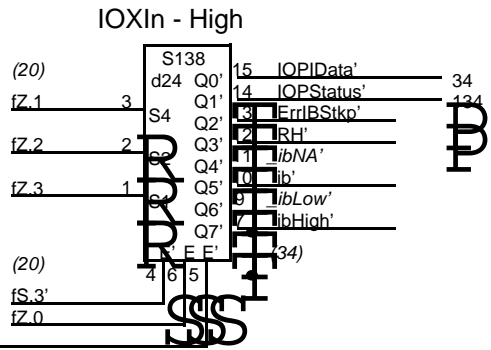
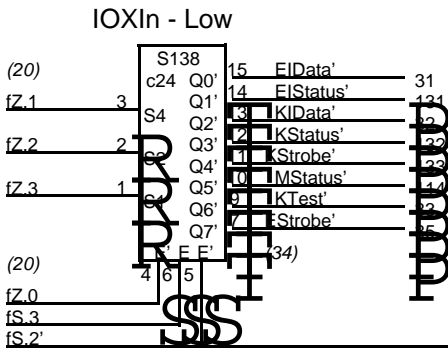
Bank_ replaces CrlrIOPReq', which is now obsolete. CrlrIOPReq' was also connected to backplane pin 184.

The meaning of Des_YBus' depends on which cycle it is activated in:

Des_YBus' in C2 means Write Des Address
Des_YBus' in C1 or C3 means Write Des Data

XBus_Des' can be activated in any cycle
See page sCPE31 for details of DES logic

The fZNorm-High decoder has been added in the 16K CP to derive the DES decodes.

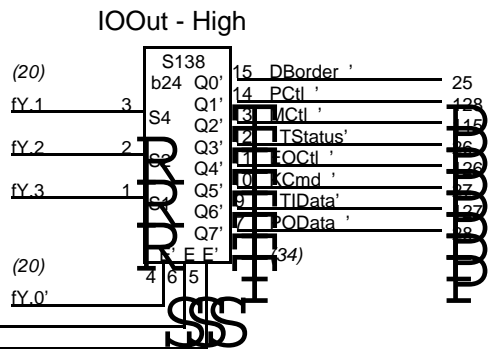
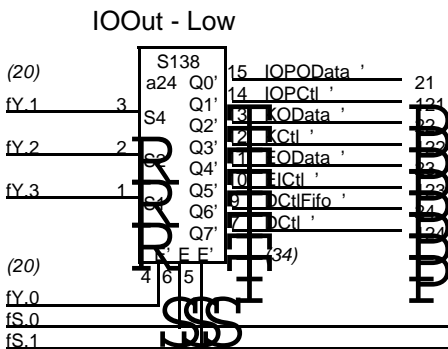


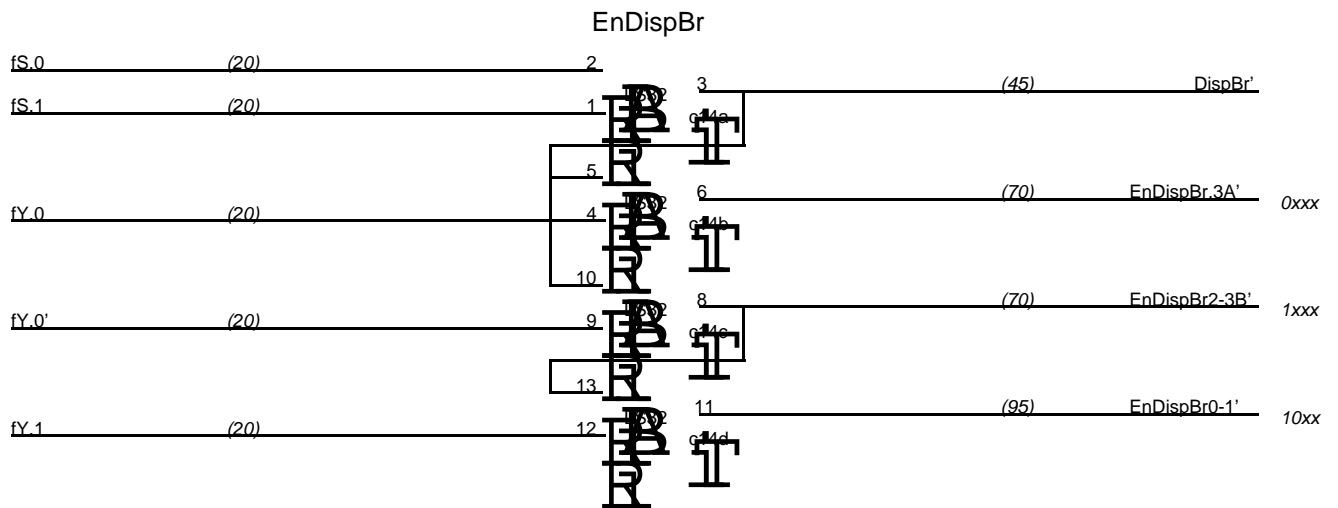
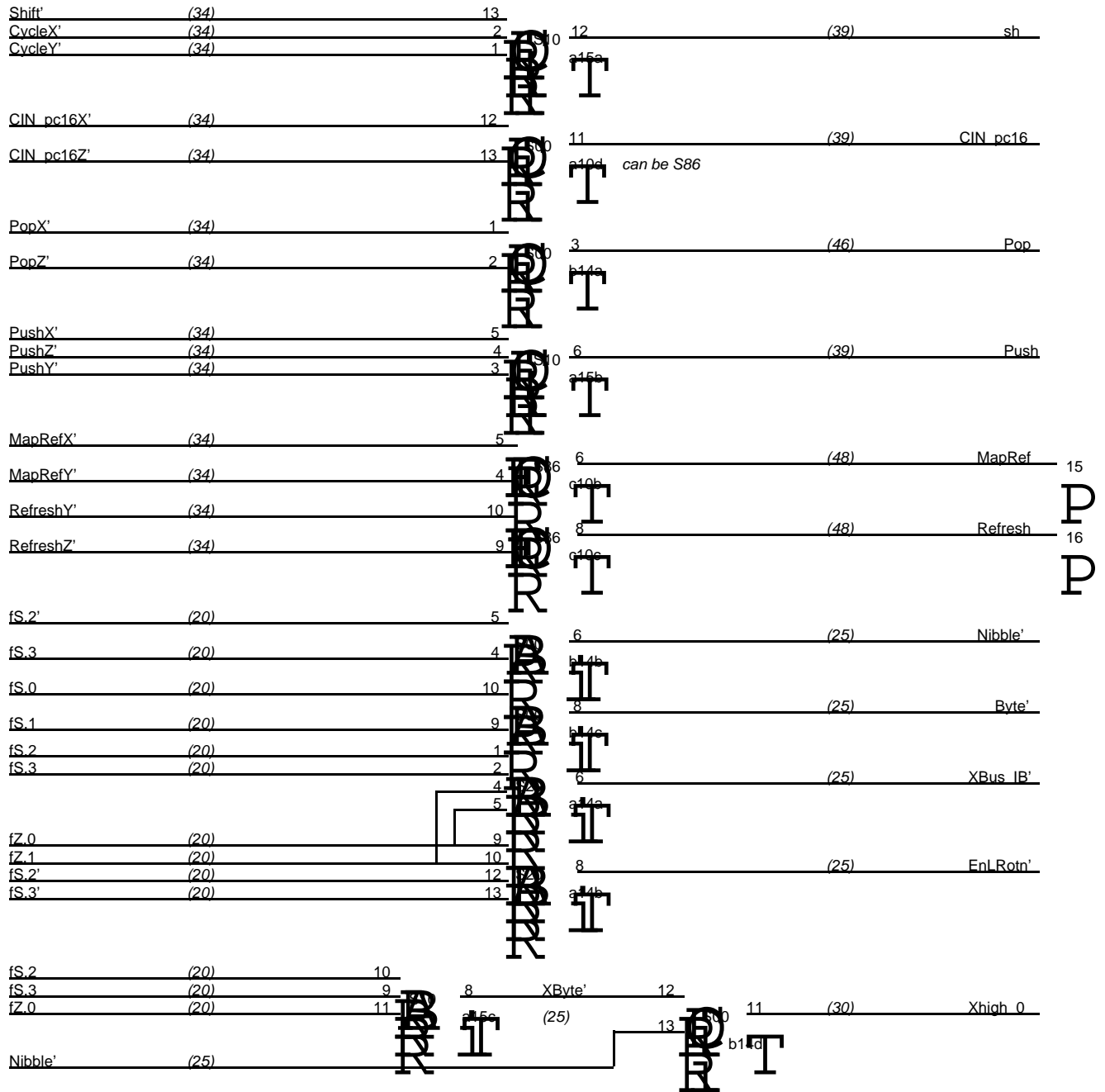
S138 Timing:

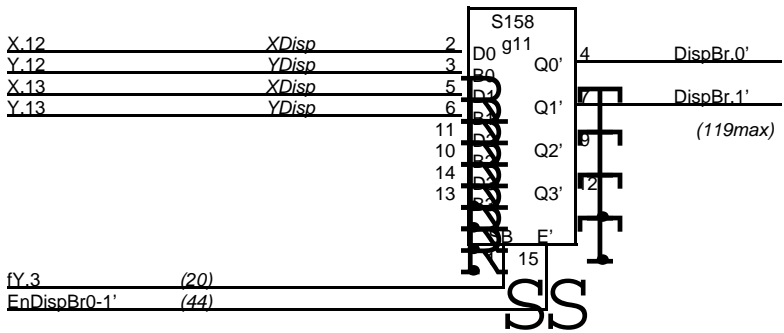
Propagation delays

from Selects to Q' 14nS
from Enables to Q' 13nS

These timings are very conservative!





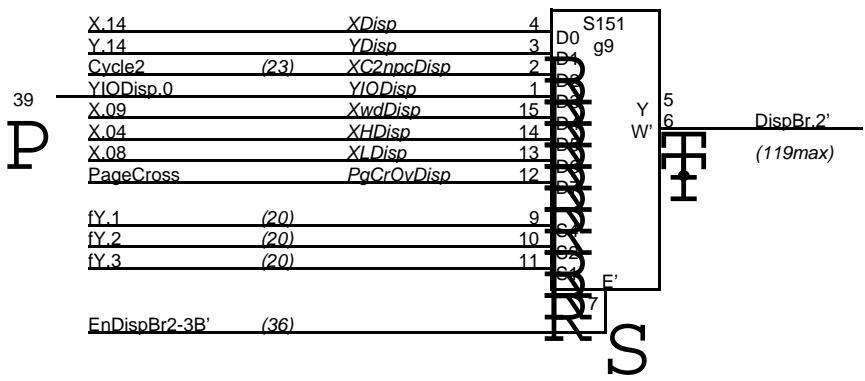


DispBr[0-1] = max(c+32,69,133)

20 ^ to fY
 24[3] S151 select to DispBr
 18 DispBr' setup
 64[3]=69

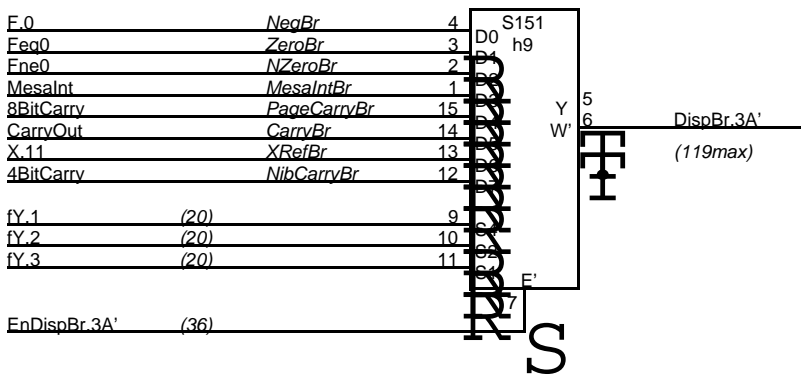
95 ^ to EnDispBr0-1'
 18[2] S151 E' to DispBr
 18 DispBr' setup
 131[2] = 133 nS

c condition source
 12[2] S151 data to DispBr
 18 DispBr' setup
 c+30[2]= c+32



DispBr Setup

5 S00 in to pTC
 6[1] S64 in to pNIA
 5[1] 25S09/S374 setup
 18 nS

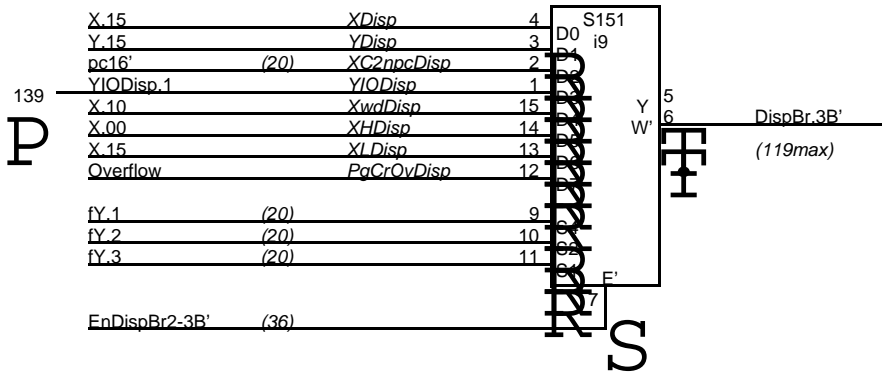


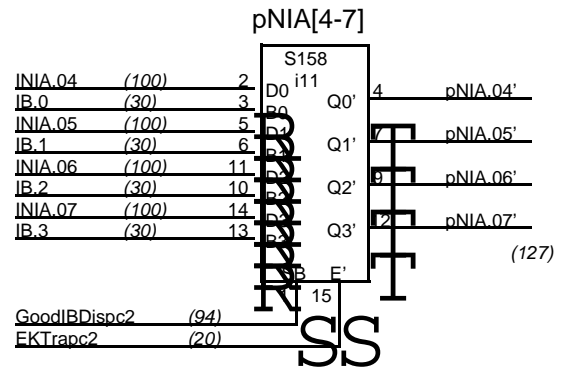
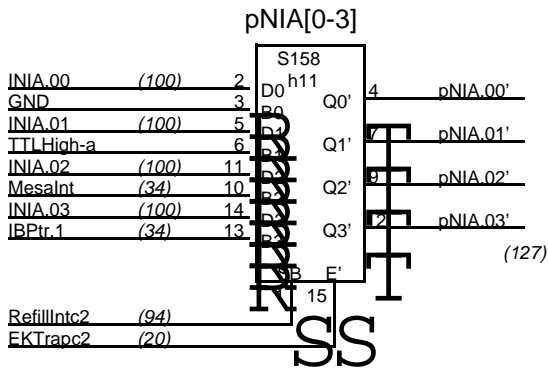
DispBr[2-3]=max(c+26,55,103)

20 ^ to fY
 15[2] S151 select to DispBr
 18 DispBr' setup
 51[4]=55 nS

70 ^ to EnDispBr.3A'
 13[2] S151 E' to DispBr
 18 DispBr' setup
 101[2] = 103 nS

c condition source
 7[1] S151 data to DispBr
 18 DispBr' setup
 c+23[3]=c+26 nS





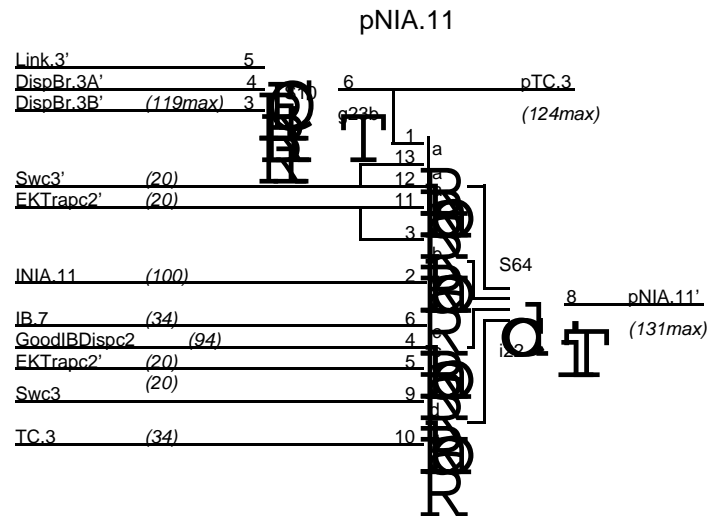
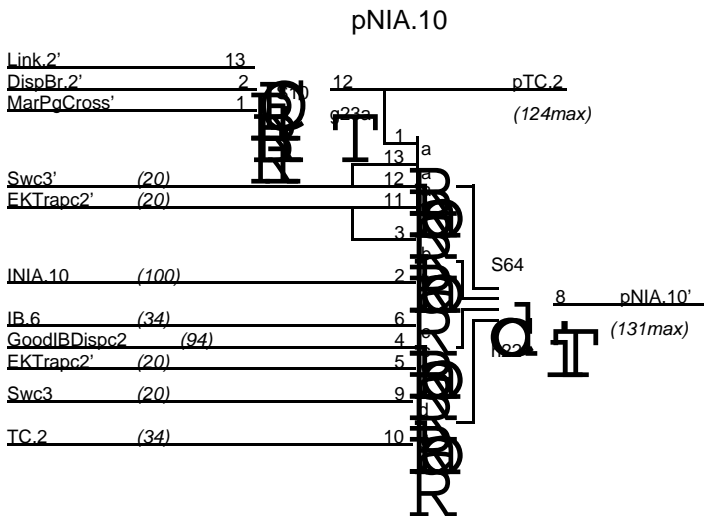
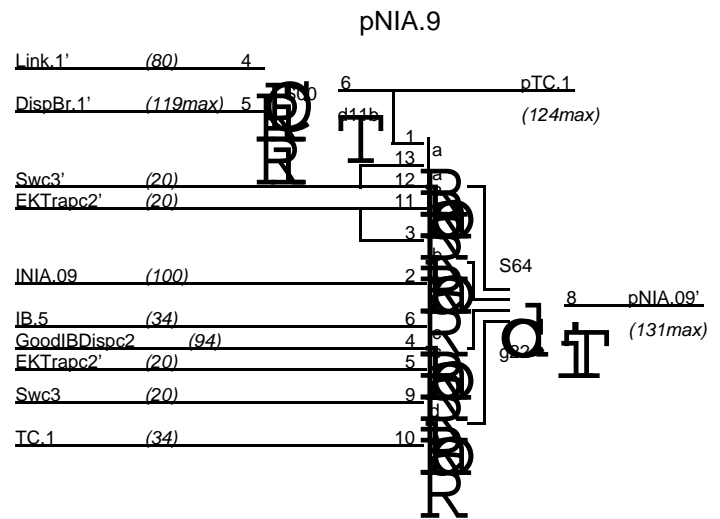
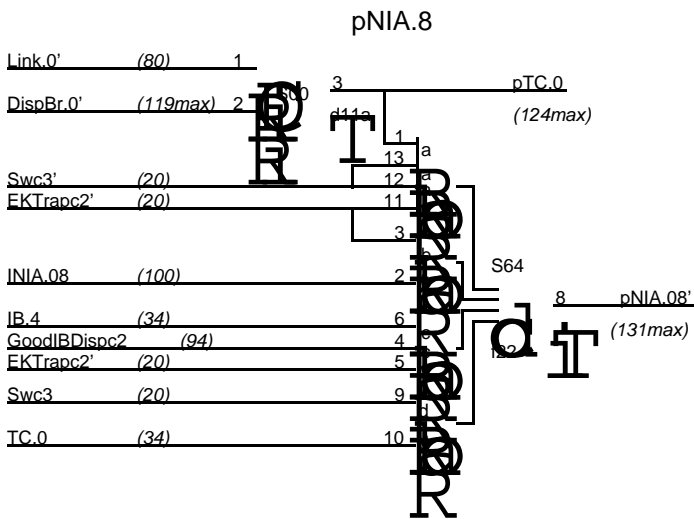
pNIA[0-7]=max(127, 120, 46) nS

94 ^ to RefillIntc2
 24[3] LS158 SB to pNIA'
 5[1] 25S09/S374 setup
 123[4]=127 nS

100 ^ to INIA
 12[2] LS158 data to pNIA'
 5[1] 25S09/S374 setup
 117[3]=120 nS

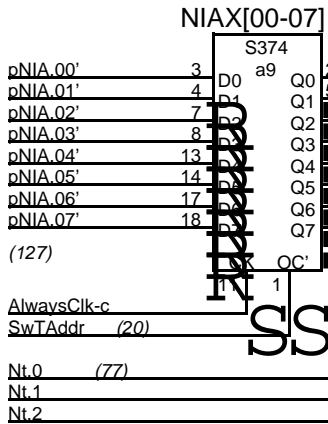
20 ^ to EKErrc2
 18[2] LS158 E' to pNIA'
 5[1] 25S09/S374 setup
 43[3]=46nS

(See page 11 for pNIA[8-11] timing)



NIAX.00' (20) From IOP Receivers. Page 20

NIAX.01'
NIAX.02'
NIAX.03'
NIAX.04'
NIAX.05'
NIAX.06'
NIAX.07'
NIAX.08'
NIAX.09'
NIAX.10'
NIAX.11'



pNIA.00' 3
pNIA.01' 4
pNIA.02' 7
pNIA.03' 8
pNIA.04' 13
pNIA.05' 14
pNIA.06' 17
pNIA.07' 18
(127)
AlwaysClk-c
SwTAddr (20)
Nt.0 (77)
Nt.1
Nt.2

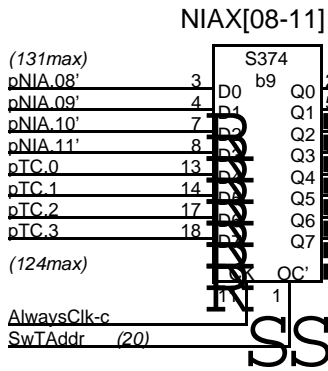
Link timing

20 ^ to fX
35 Am27S07 tAA
22[3] pLink' to Link'
18 DispBr' setup
95[3]= 98 nS

20 ^ to fX.0, NIAx.7'
22[3] fX.0 to pRet'
22[3] pRet' to Link'
18 DispBr' setup
82[6] = 88 nS

TPC/TC timing

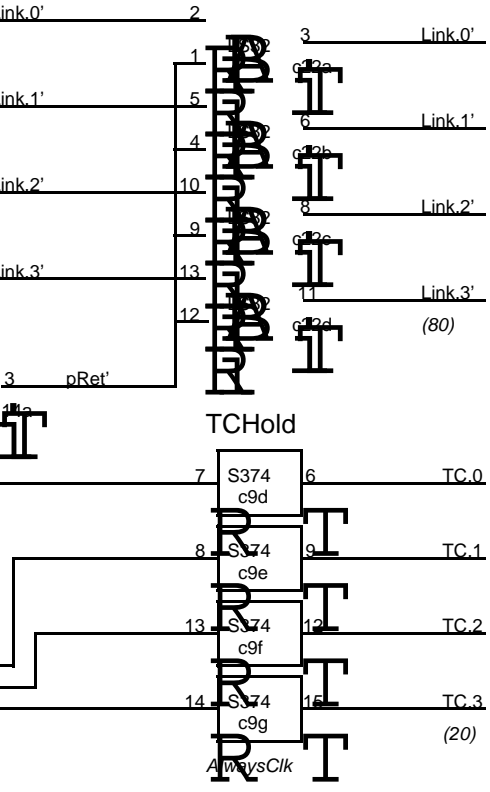
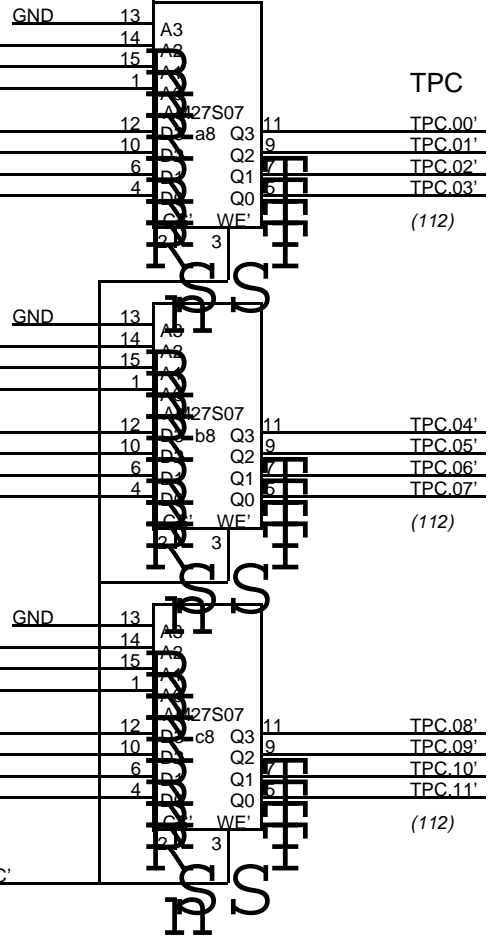
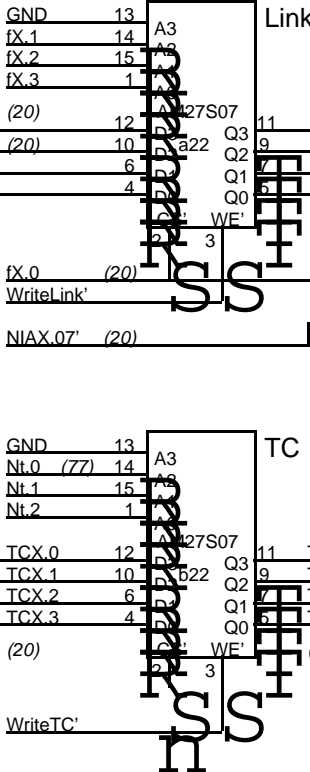
77 ^ to Nt
35 Am27S07 tAA
5[1] 25S09/S374 setup
117[1]= 118 nS

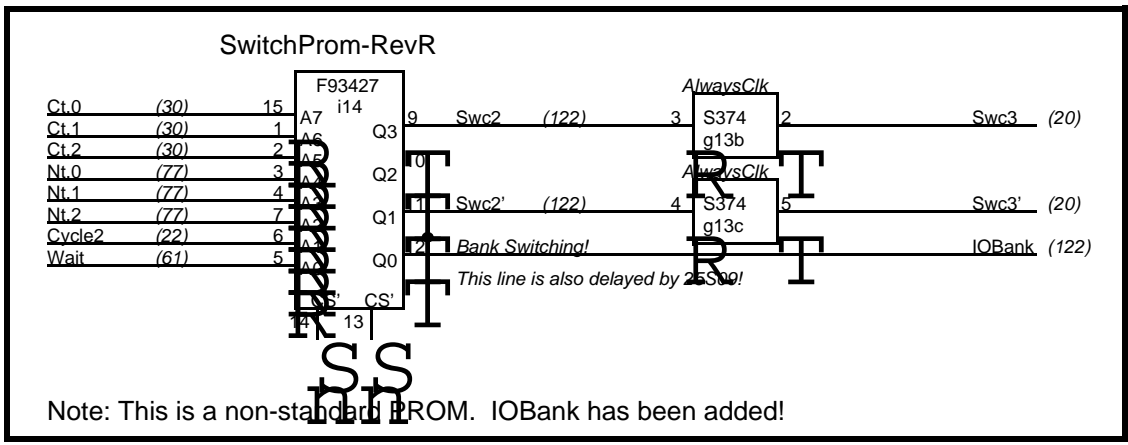
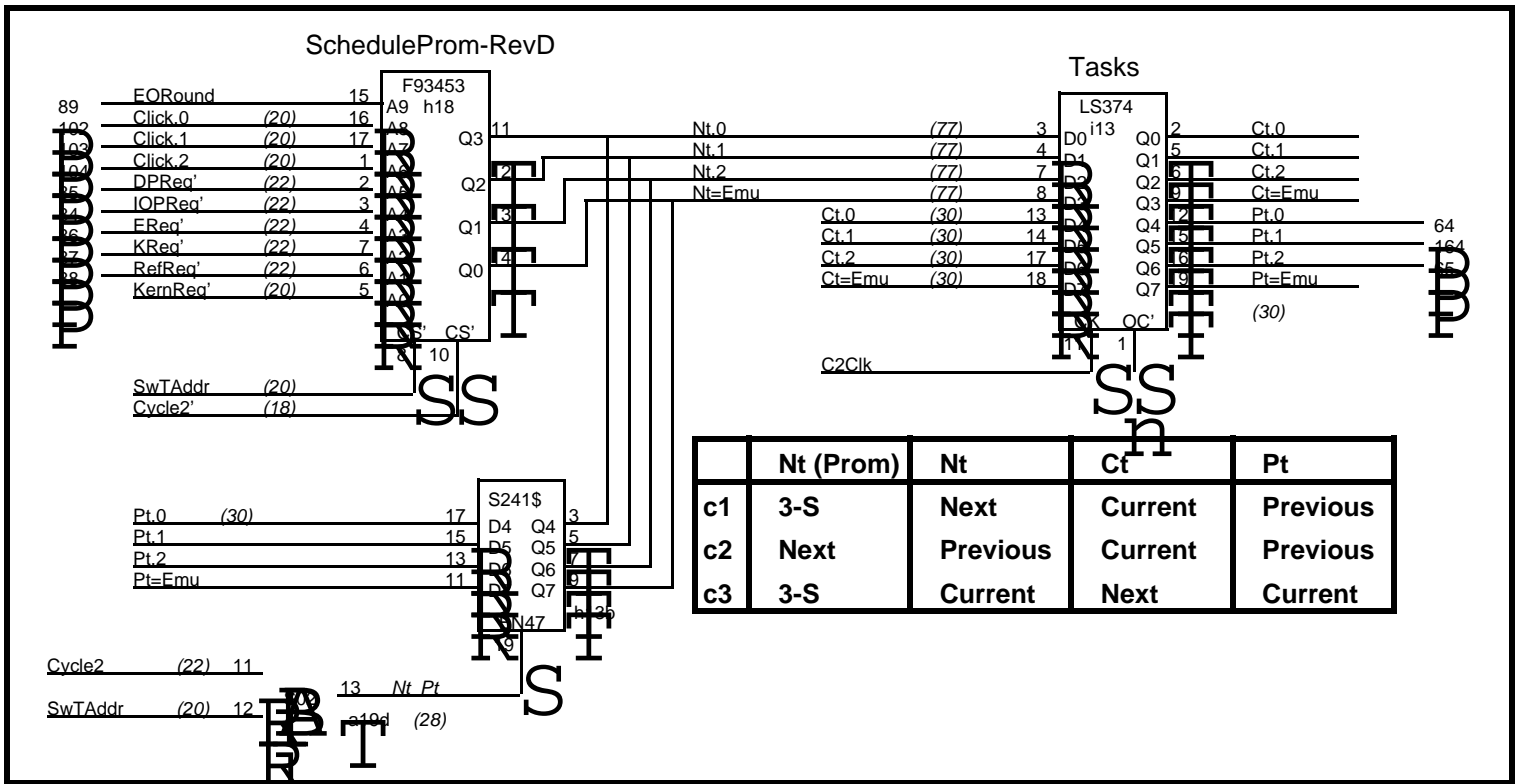


(131max)
pNIA.08' 3
pNIA.09' 4
pNIA.10' 7
pNIA.11' 8
pTC.0 13
pTC.1 14
pTC.2 17
pTC.3 18
(124max)
AlwaysClk-c
SwTAddr (20)

If only pullups were used on output of Link (instead of the LS32 kludge), then Link timing would be:

98 WriteLink' active
25[3] WE' to pLink high
18 DispBr' setup
141[3] = 144 nS





Swc2 timing=max(133,101,101)

22 ^ to Kreq'	20 ^ to SwTAddr	28 ^ to Nt_Pt
55 F93453 addr to Nt	25 F93453 CS' to Nt	15[2] S241 EN to Nt
45 F93427 addr to Swc2	45 F93427 addr to Swc2	45 F93427 addr to Swc2
<u>10[1]</u> 25S09 SB setup	<u>10[1]</u> 25S09 SB setup	<u>10[1]</u> 25S09 SB setup
<u>132[1]</u> =133 nS	<u>100[1]</u> =101 nS	<u>98[3]</u> =101 nS

Click Assignment

0	Ethernet
1	Disk
2	IOP
3	Ethernet/Disk
4	Display/LSEP/Rfrsh

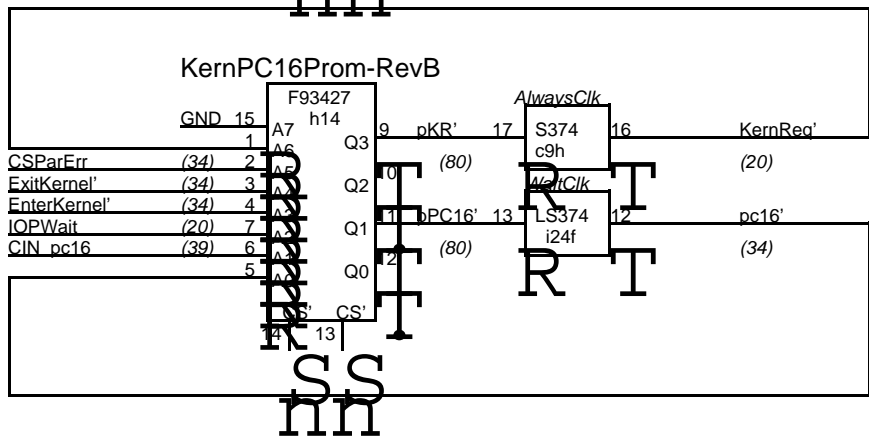
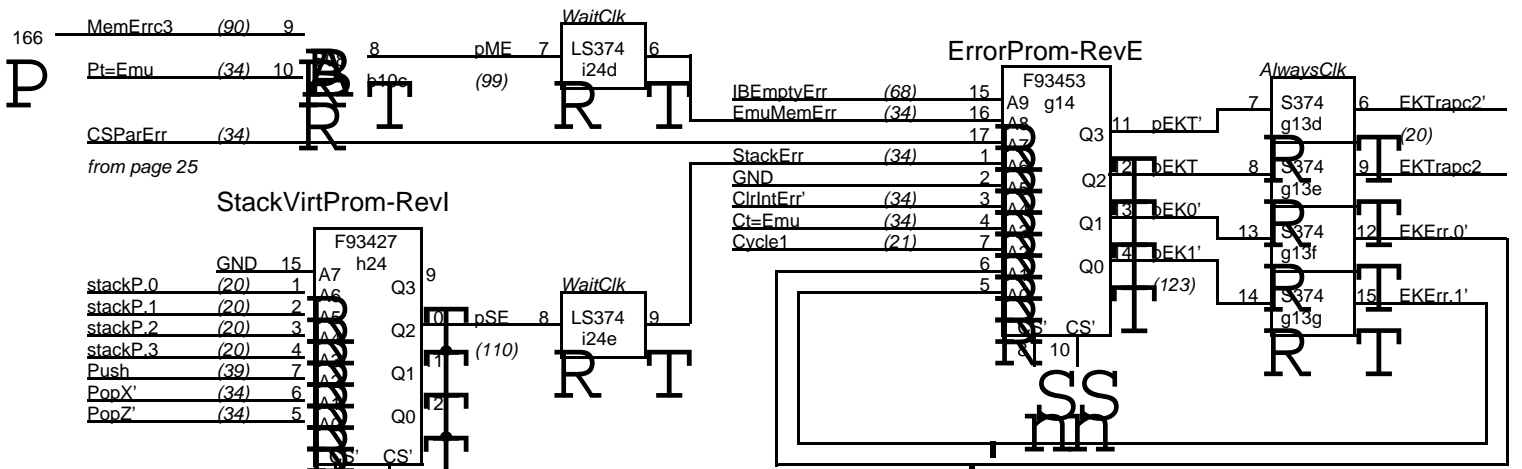
Notes:

When Disk = SA4000, Click 3 is Ethernet only.

When Disk = Trident, Click 3 is Ethernet on even rounds, Trident on Odd rounds (ie, 10-click round)

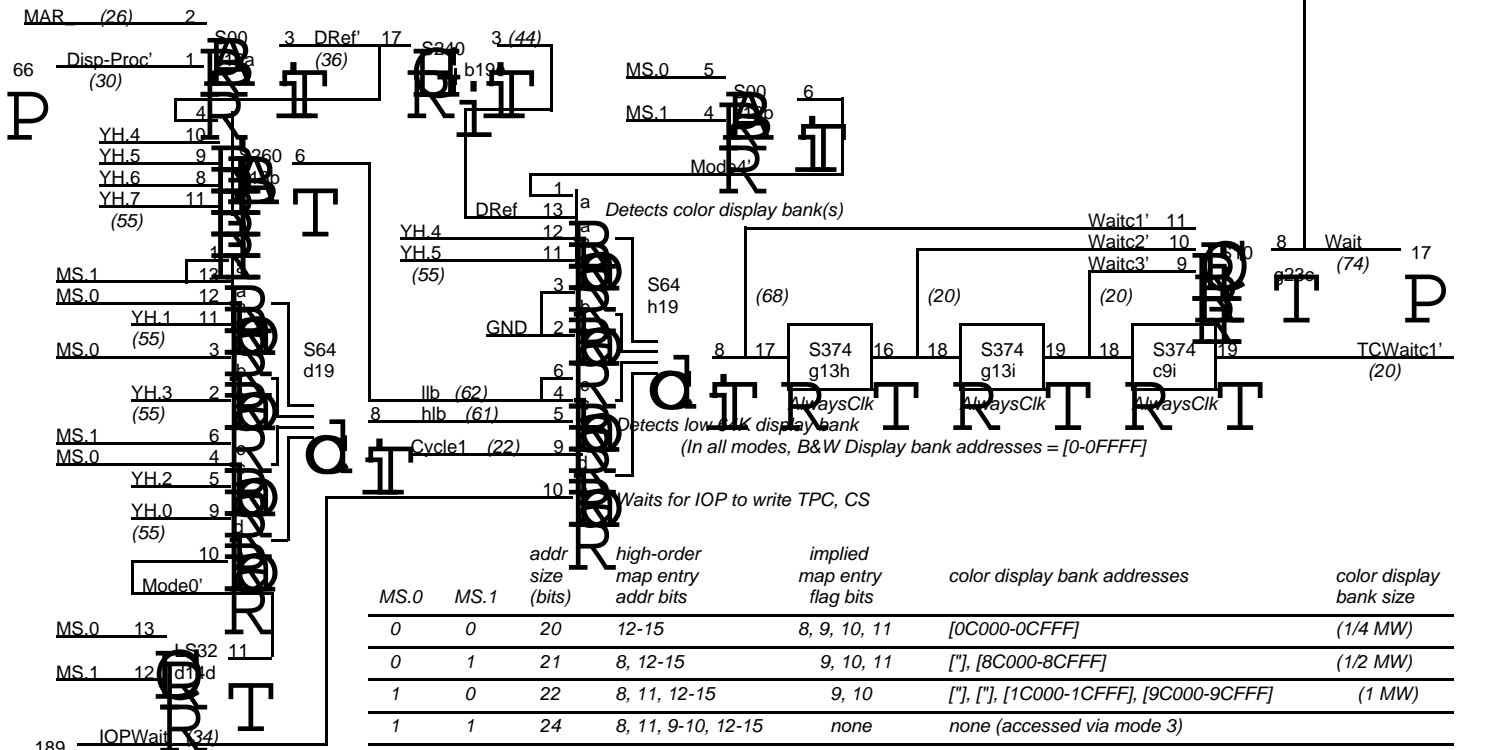
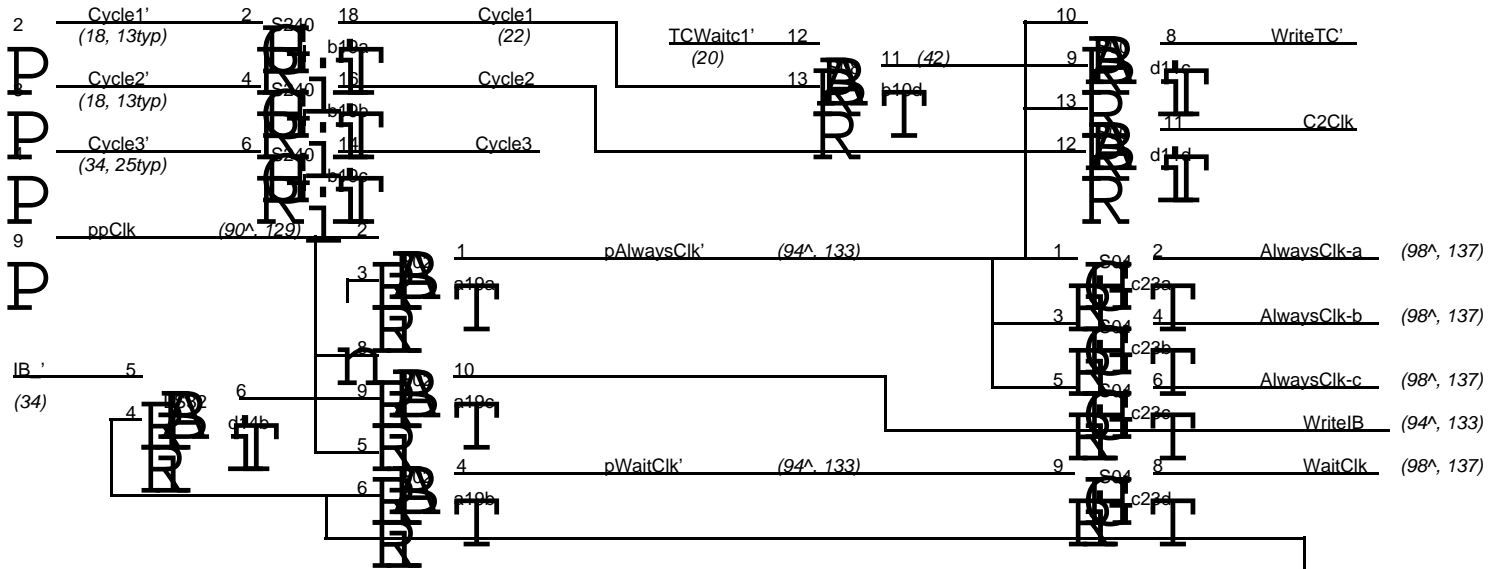
The Display & LSEP-refresh tasks never both use Click 4

Warning: This drawing contains font 4 macros!



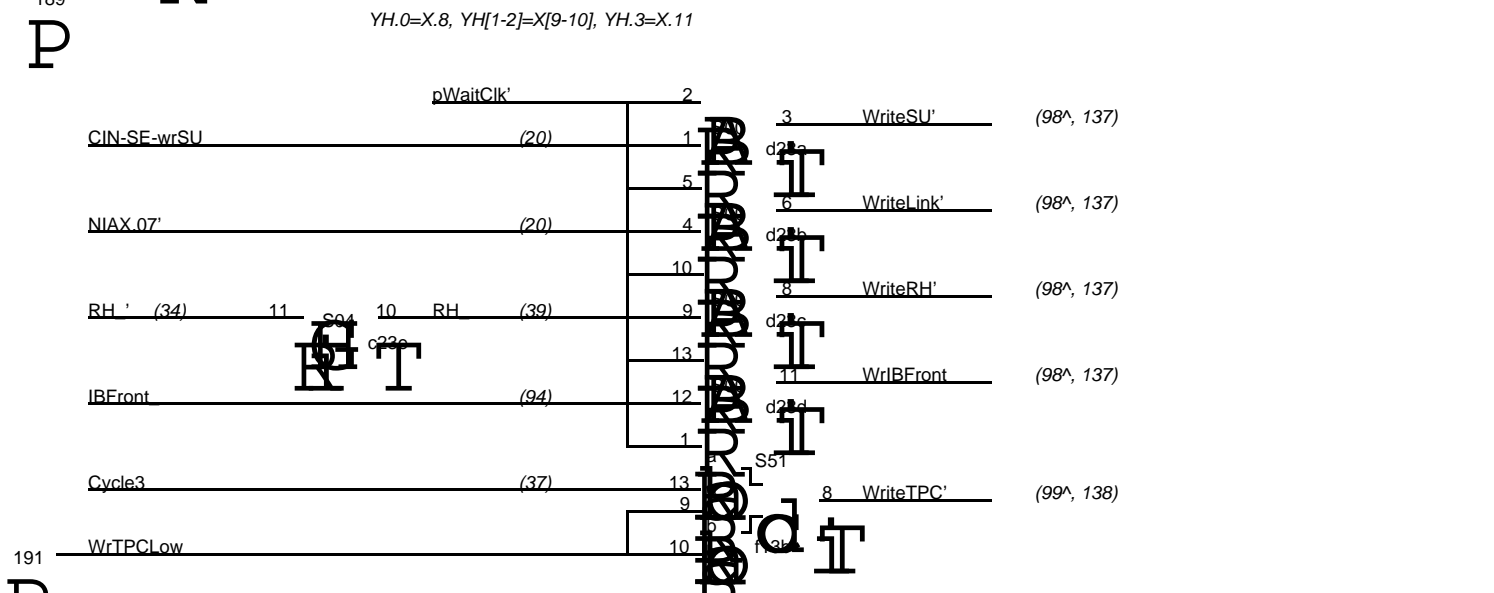
EKErr' at Trap location 0

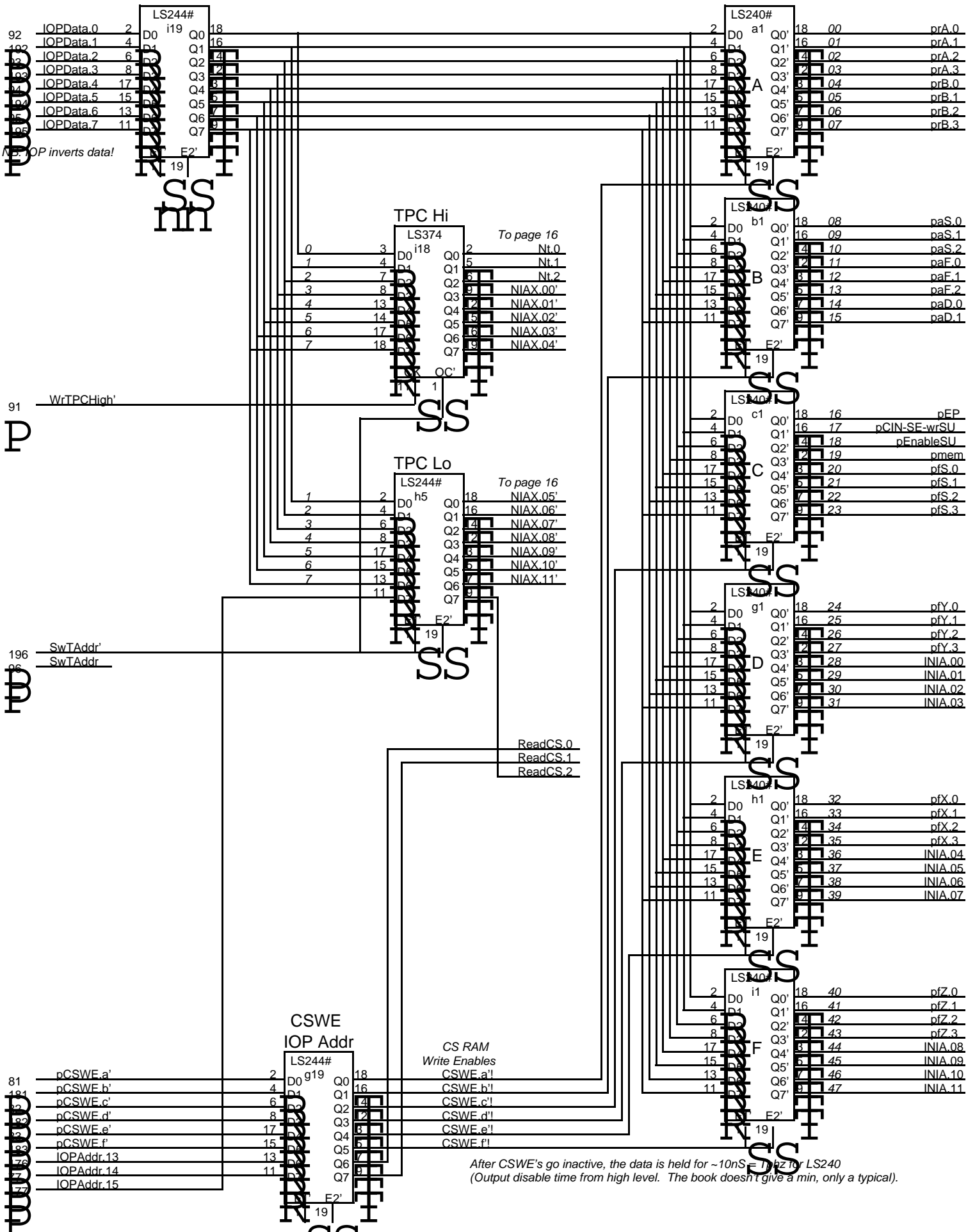
0	IB Empty
1	Stack
2	Emulator Memory
3	CS Parity

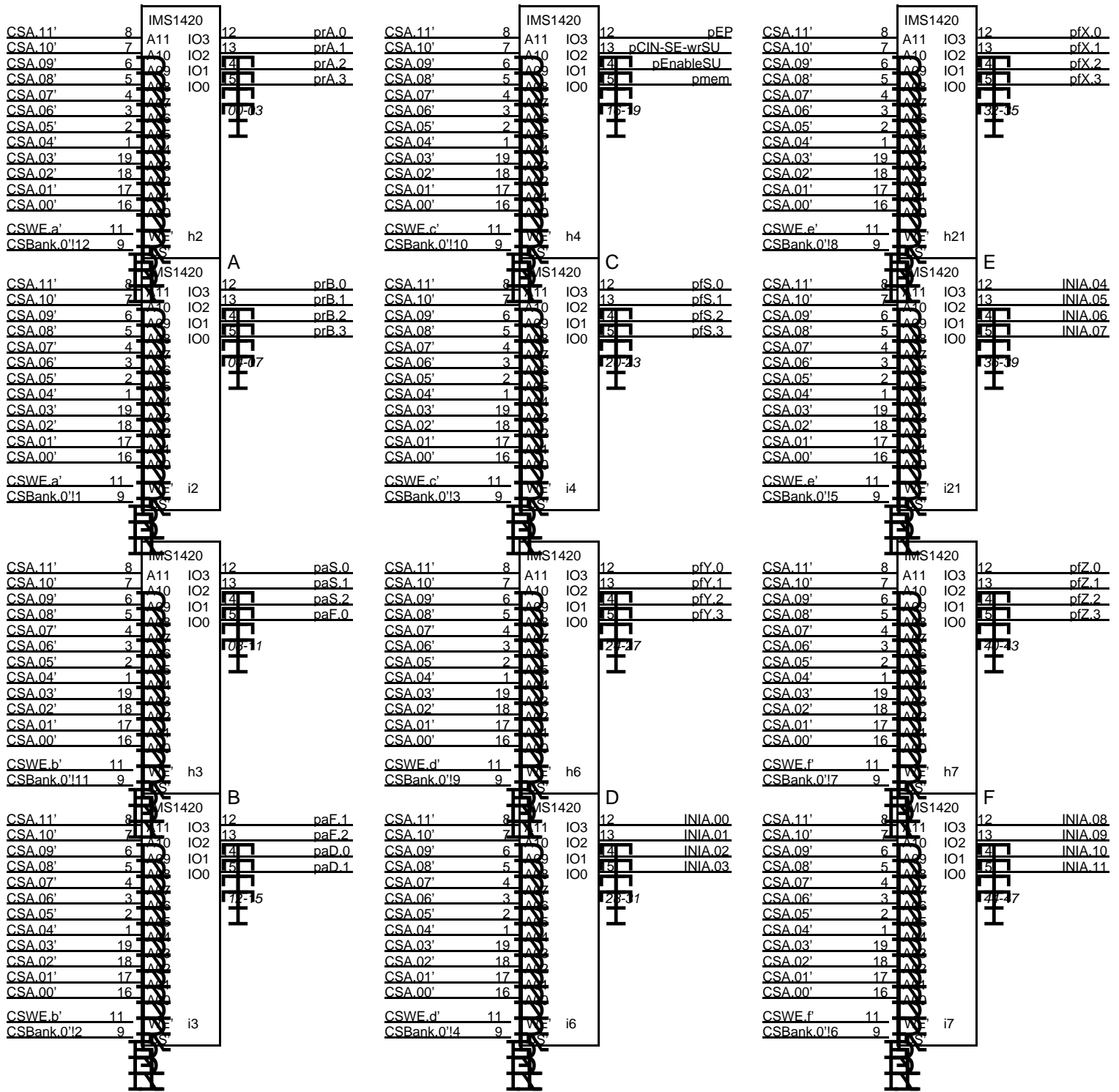


MS.0	MS.1	addr size (bits)	high-order map entry addr bits	implied map entry flag bits	color display bank addresses	color display bank size
0	0	20	12-15	8, 9, 10, 11	[0C000-0CFFF]	(1/4 MW)
0	1	21	8, 12-15	9, 10, 11	[*], [8C000-8CFFF]	(1/2 MW)
1	0	22	8, 11, 12-15	9, 10	[*], [*], [1C000-1CFFF], [9C000-9CFFF]	(1 MW)
1	1	24	8, 11, 9-10, 12-15	none	none (accessed via mode 3)	

YH.0=X.8, YH[1-2]=X[9-10], YH.3=X.11





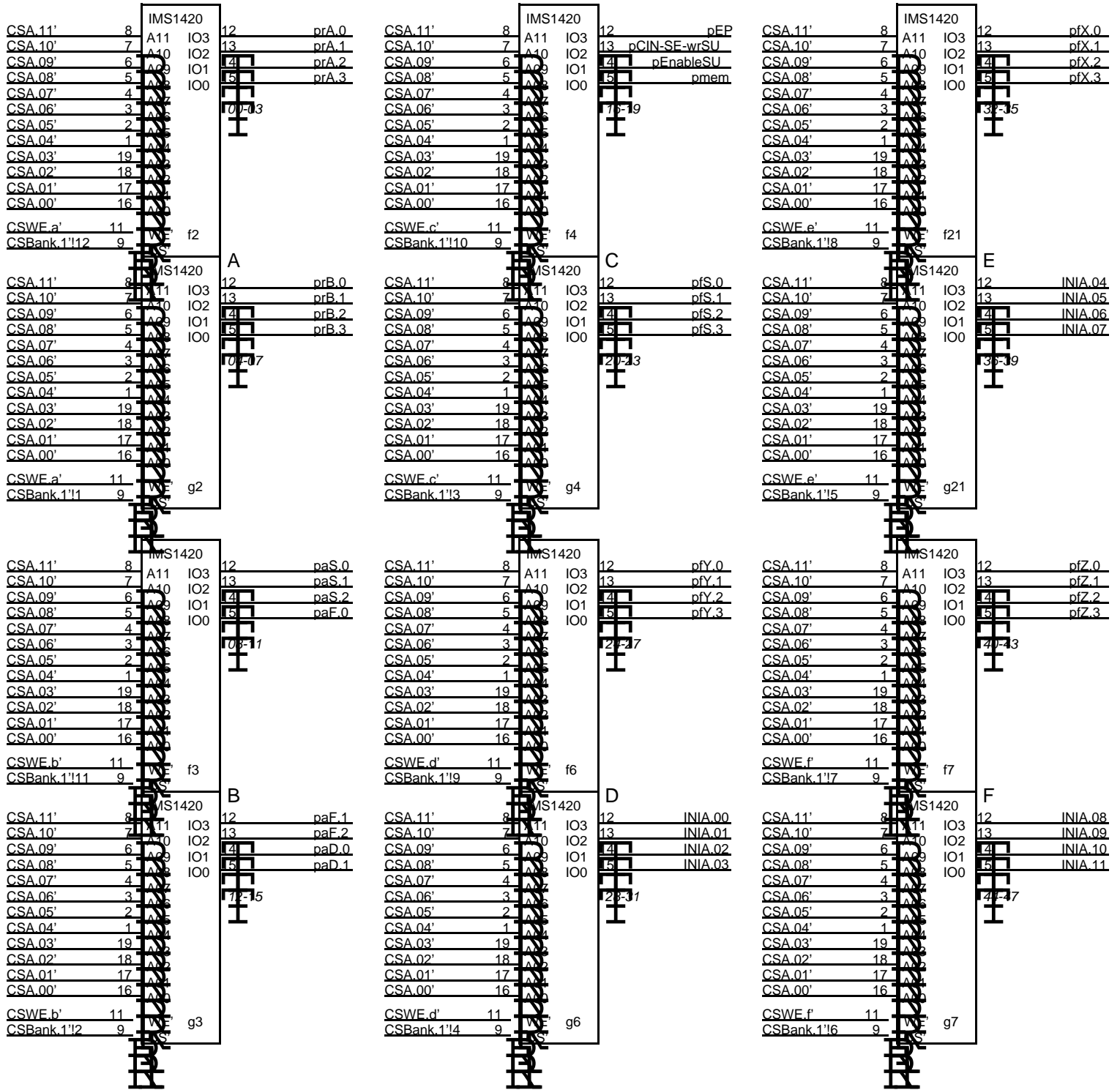


READ		WRITE - Data Hold	
Clock to CSA' valid	17	tPLH for LS240	12
Transmission Delay	13	tPZ for LS244	10
tAA for IMS 1420-55	50		
CS Data valid at	80		22

This suggests that IMS 1420-70 would also work without any trouble.

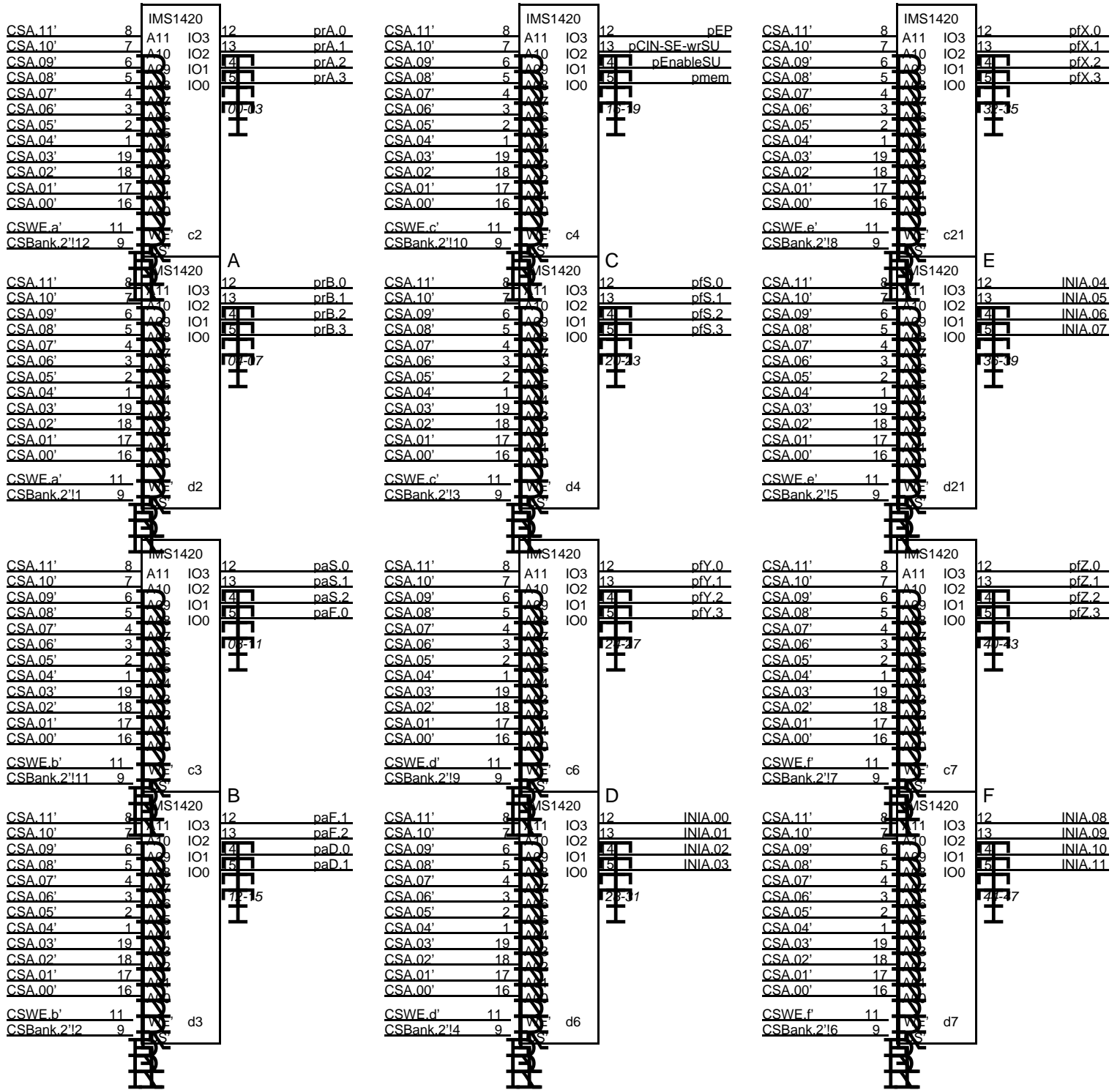
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project CPE	File Control Store Bank 0: 0000-0FFF	Designer sCPE21.sil	Rev TonyWest.PA	Date X	Page 1/14/83	Page 21
-------------------	----------------	---	------------------------	--------------------	-----------	-----------------	------------



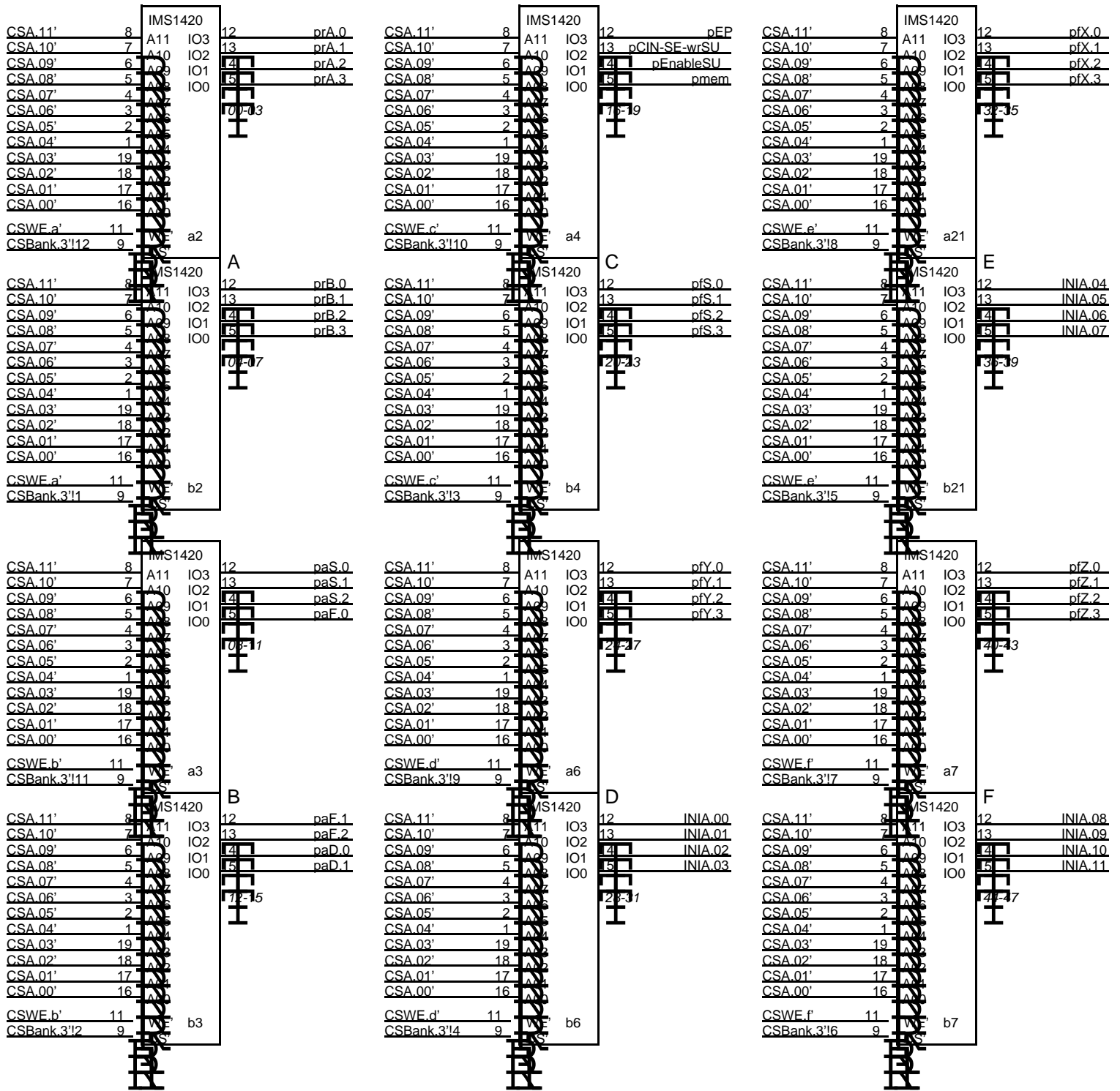
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project CPE	Control Store Bank 1: 1000-1FFF	File sCPE22.sil	Designer TonyWest.PA	Rev X	Date 1/14/83	Page 22
-------------------	----------------	---------------------------------	--------------------	-------------------------	----------	-----------------	------------



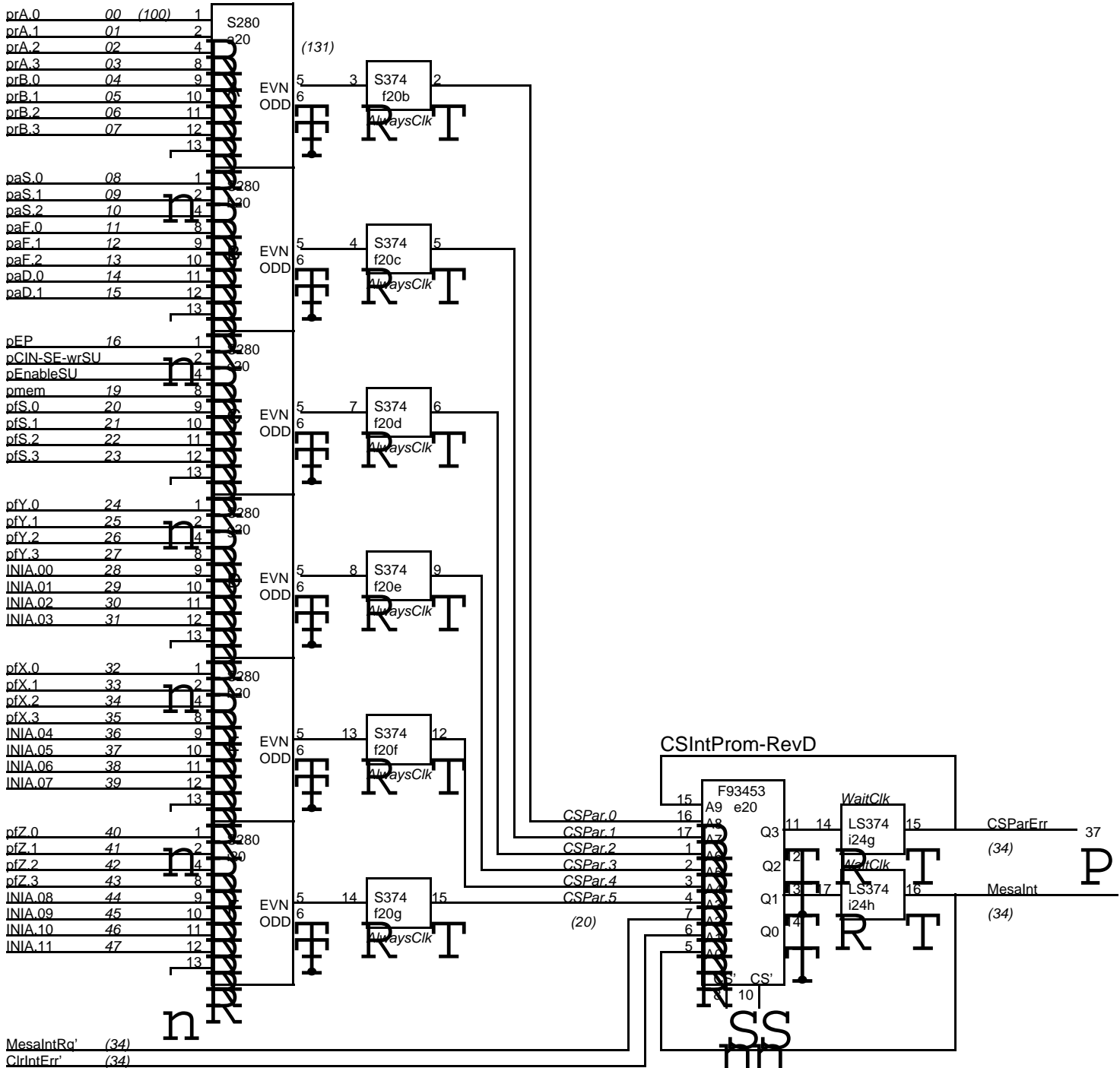
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project CPE	File Control Store Bank 2: 2000-2FFF	Designer sCPE23.sil	Rev TonyWest.PA	Date X	Page 1/14/83	Page 23
-------------------	----------------	---	------------------------	--------------------	-----------	-----------------	------------



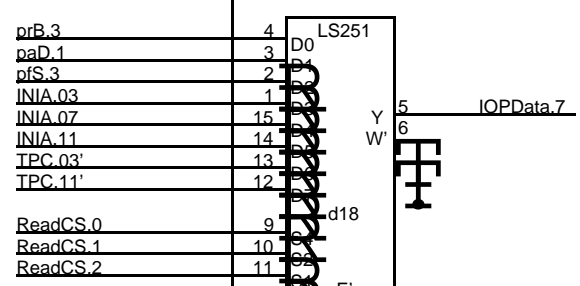
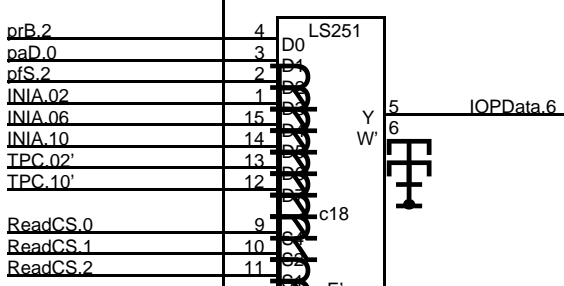
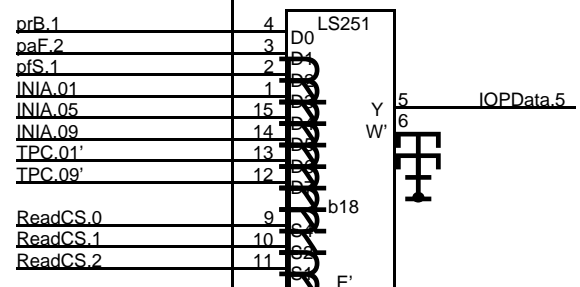
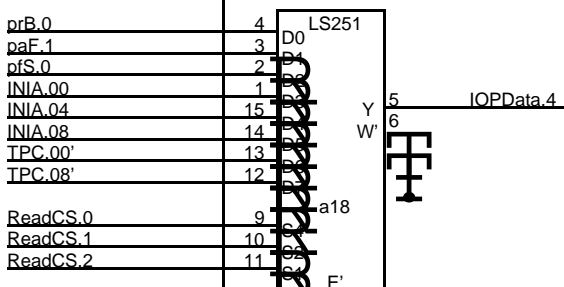
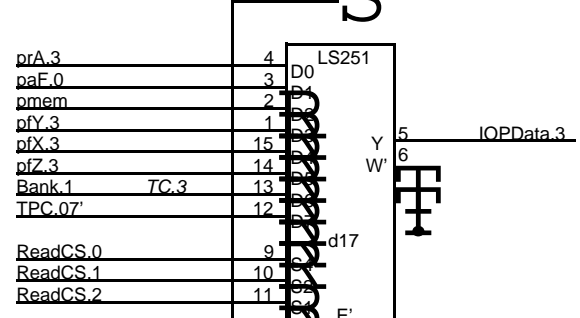
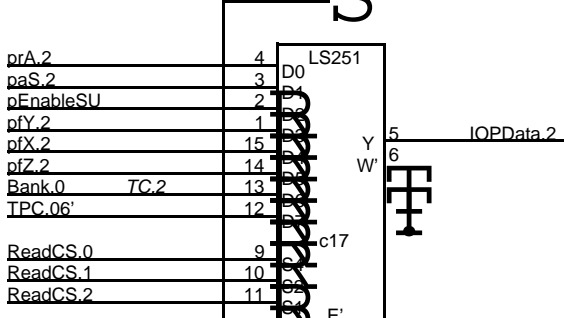
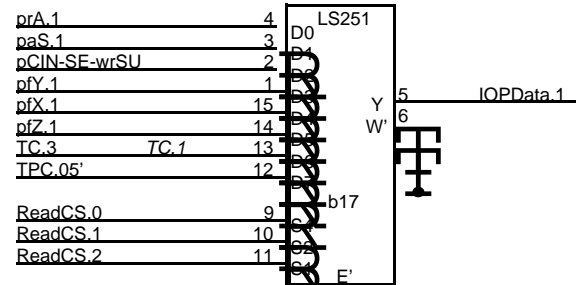
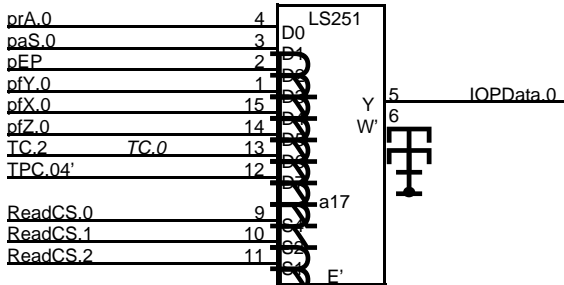
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project CPE	File Control Store Bank 3: 3000-3FFF	Designer sCPE24.sil	Rev TonyWest.PA	Date X	Page 1/14/83	Page 24
-------------------	----------------	---	------------------------	--------------------	-----------	-----------------	------------



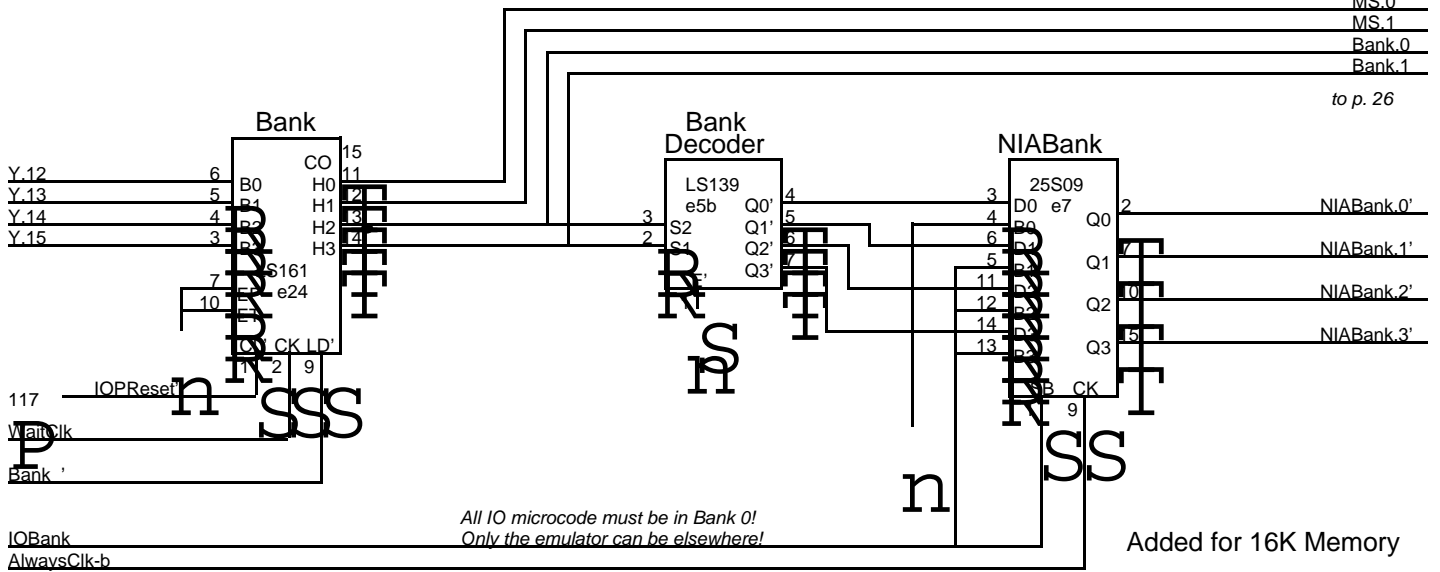
$CSParErr = XOR(CSPar[0..5])$

NB: TC[0-3] have been replaced by Bank[0-3].

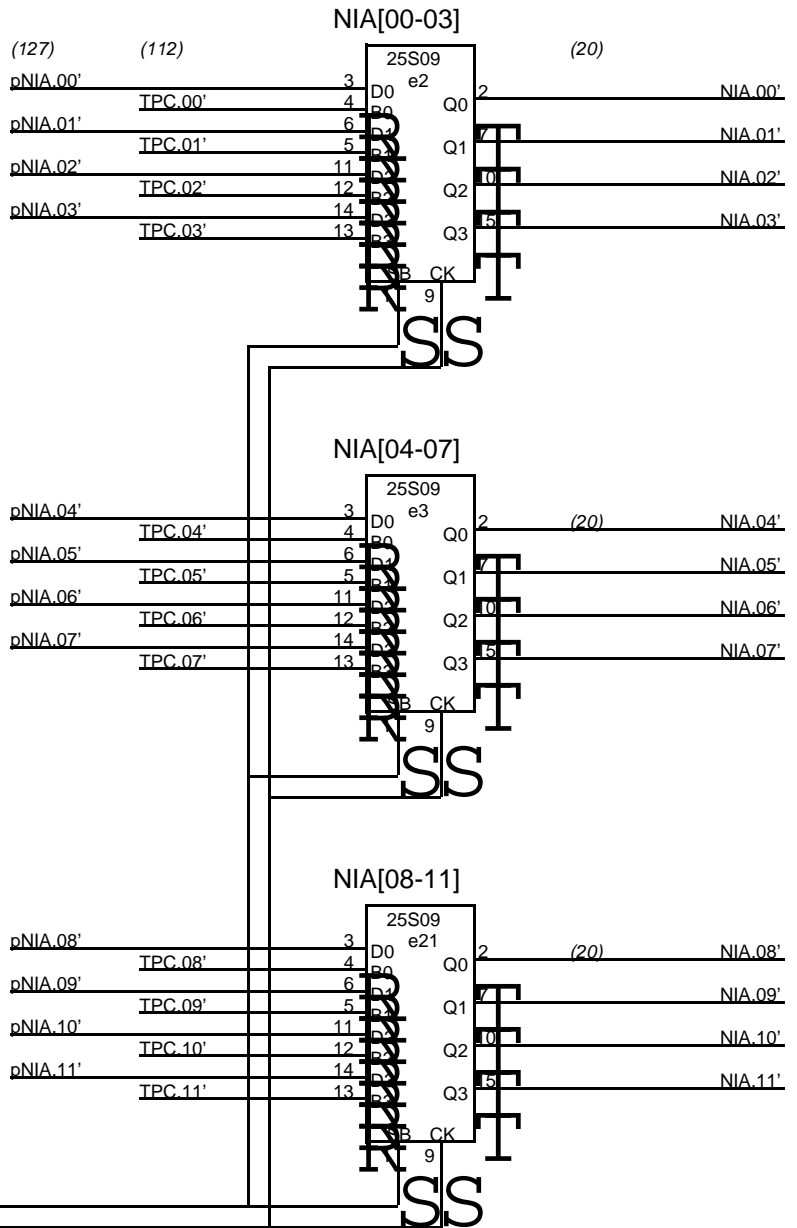


188
P

ReadCSEn'



This section is standard



NOTE on Control Store Addresses

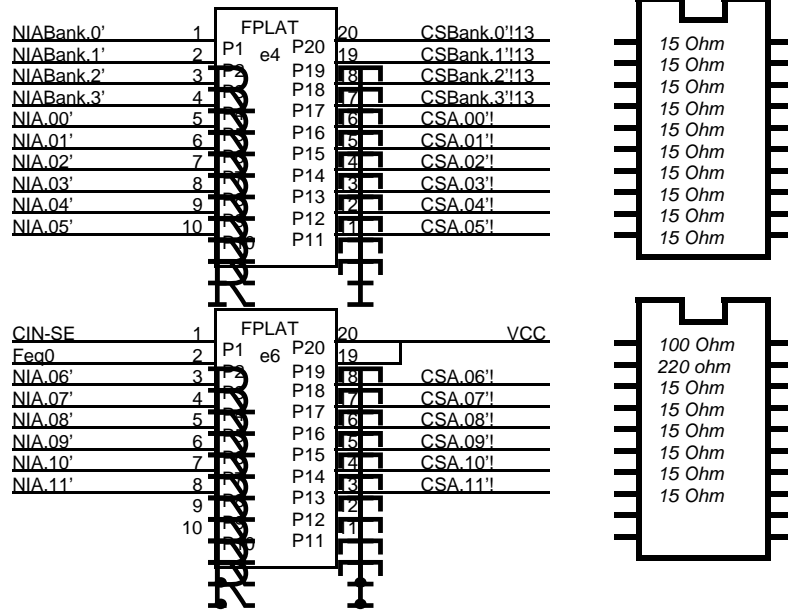
The next instruction address for the control store comes from one of two basic places:

1. TPC registers if switching tasks
2. From the INIA field of the previous microinstruction

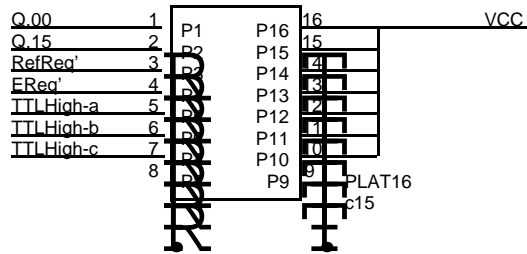
In the case of (1), task 6's TPC registers are used by the IOP to provide the address when the IOP wants to read or write data into the control store.

In the case of (2), the INIA field is suitably modified by the trap and conditional branch logic on page 16

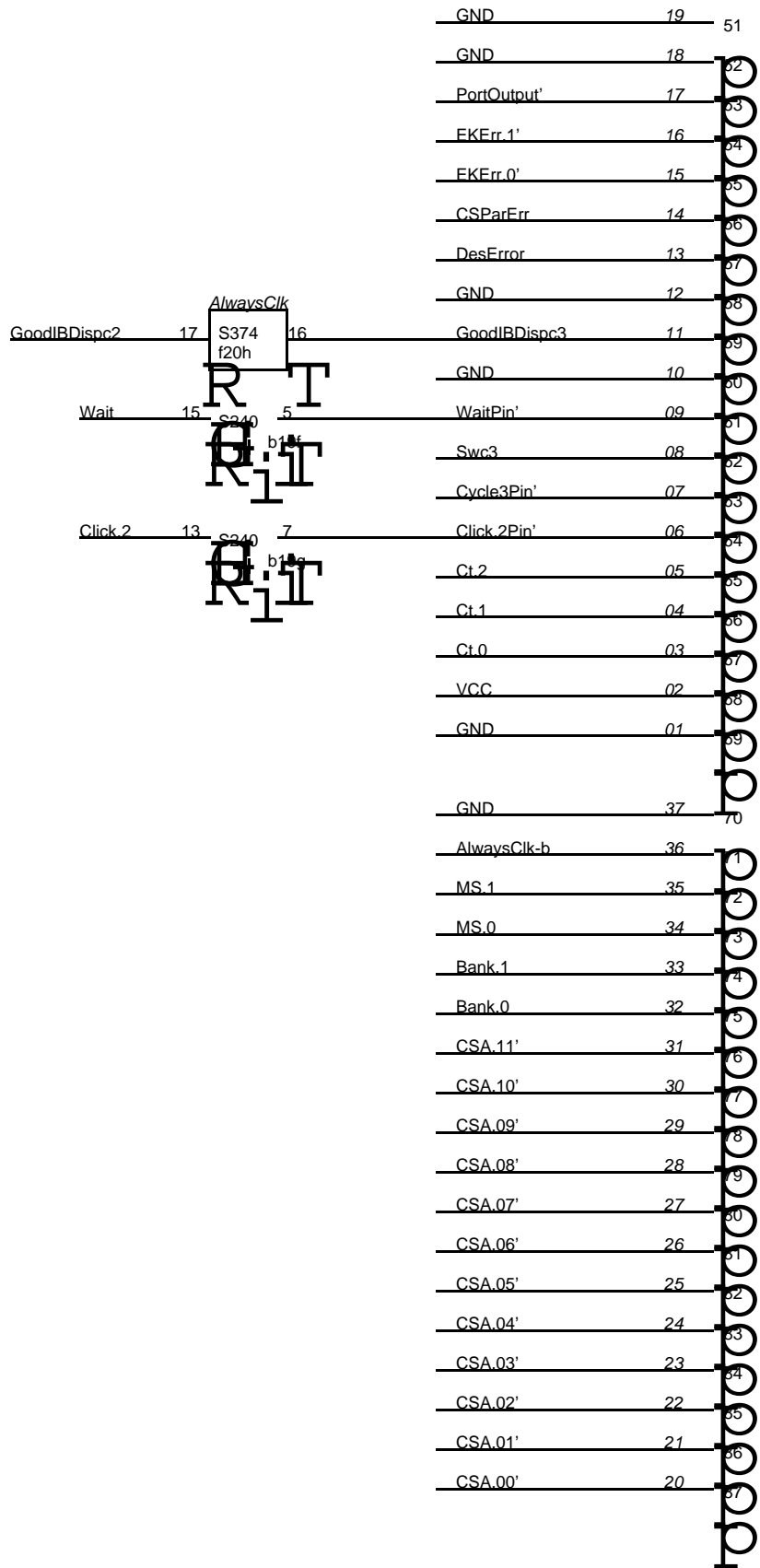
CS NIA Line Matching



1 KOhm Pullups

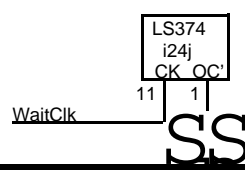
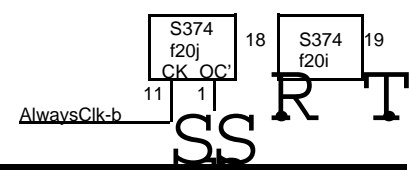
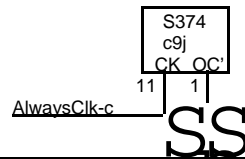
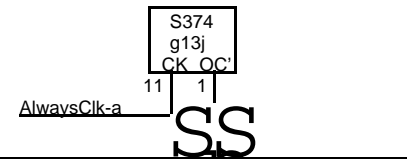
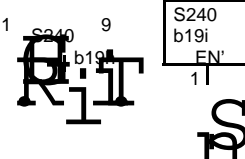
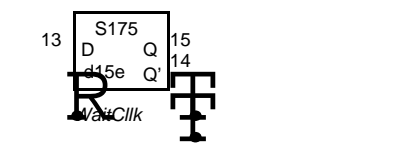
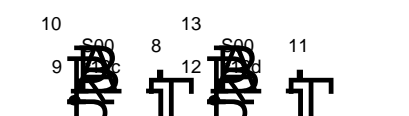
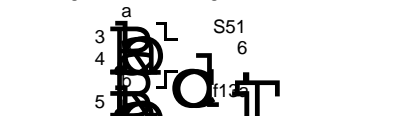
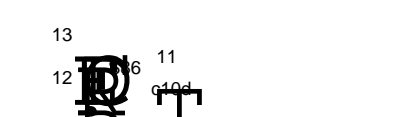
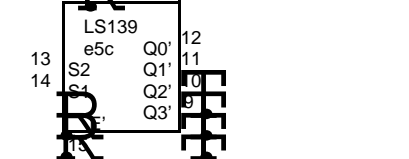


Beckman Resnet DIP
898-1-1K



Bottom Connector
Female - End View

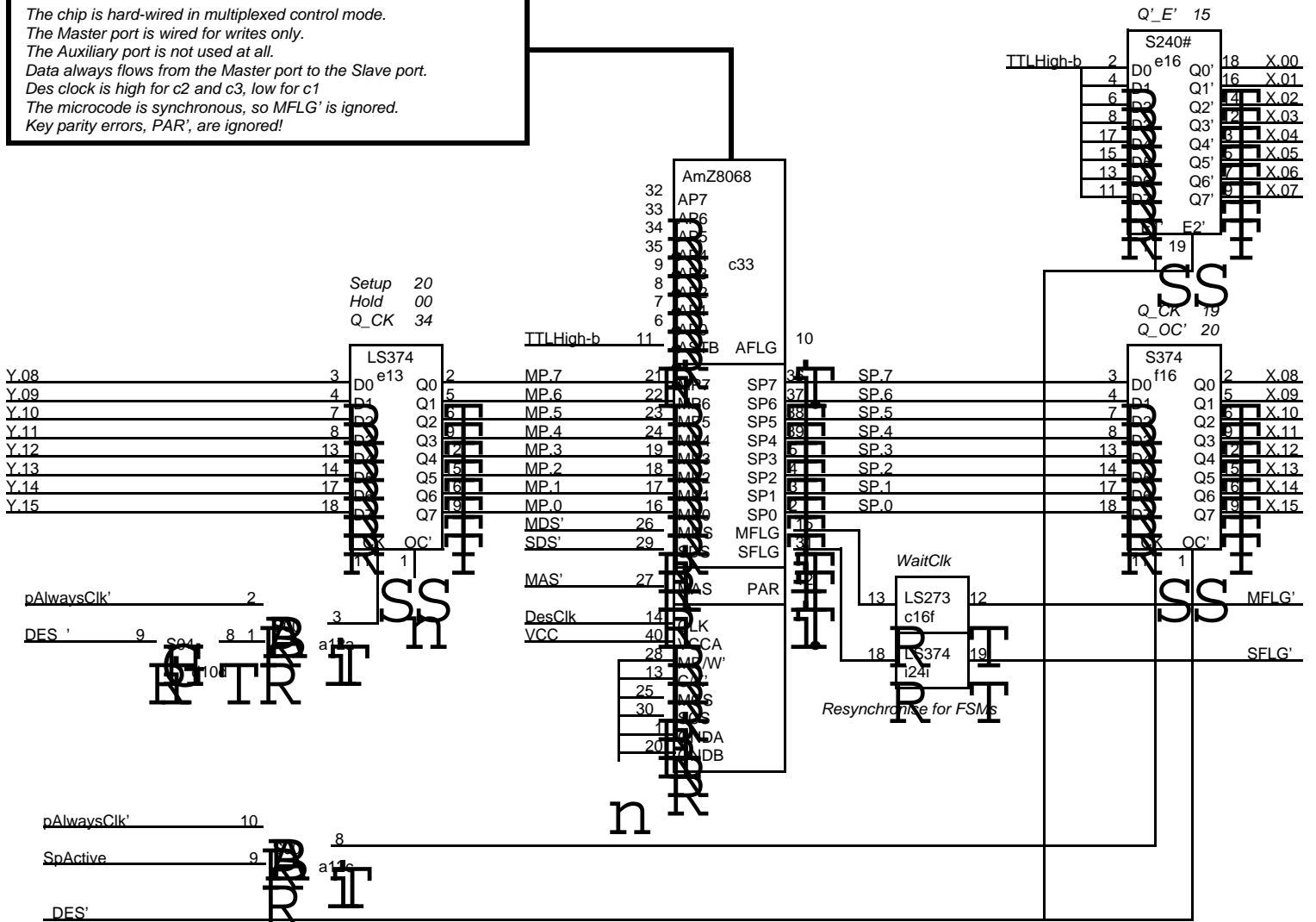
19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01
37	36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	

<p>LS374 i24</p> <p>b IBPtr.0 08 c IBPtr.1 08 d EmuMemErr 18 e StackErr 18 f pc16' 18 g CSParErr 25 h Mesalnt 25 i SFLG' 33</p> 		<p>S374 f20</p> <p>b CSPar.0 25 c CSPar.1 25 d CSPar.2 25 e CSPar.3 25 f CSPar.4 25 g CSPar.5 25 h Good/BDIspec3 29 i</p> 
<p>S374 c9</p> <p>b MAR_ 11 c AllowMDR_ 11 d TC.0 16 e TC.1 16 f TC.2 16 g TC.3 16 h KernReq' 18 i TCWaitc1' 19</p> 		<p>S374 g13</p> <p>b Swc3 17 c Swc3' 17 d EKTrapc2' 18 e EKTrapc2 18 f EKErr.0' 18 g EKErr.1' 18 h Waitc2' 19 i Waitc3' 19</p> 
<p>S240 b19</p> <p>a Cycle1 19 b Cycle2 19 c Cycle3 19 d Cycle3' 37 e DRef 19 f WaitPin' 29 g Click.2Pin' 29</p> 		<p>S04 d10</p> <p>a aD.0' 05 b MAR_ 11 c IBEmptyErr' 11 d Des_YBus 31 e XBus_SU 06 f Port_ 06</p>
<p>S04 e23</p> <p>a AlwaysClk-a 19 b AlwaysClk-b 19 c AlwaysClk-c 19 d WaitClk 19 e RH_ 19 f Fne0 04</p>		<p>S175 d15</p> <p>b MAS' 33 c MDS' 33 d SDS' 33 e 30 f *anon* 33</p> 
<p>S00 a10</p> <p>a XBus_SU' 06 b pMAR_ 11 c MarPgCross' 11 d CIN_pc16 13</p>	<p>S00 b14</p> <p>a Pop 13 b Nibble' 13 c Byte' 13 d Xhigh_0 13</p>	<p>S00 a12</p> <p>a *anon* 31 b WPort 19 c *anon* 31 d DesClkDisable 32</p>
<p>S00 e13</p> <p>a WriteSU' 19 b WriteLink' 19 c WriteRH' 19 d WrlBFront 19</p>	<p>S00 f23</p> <p>a pTC.0 15 b pTC.1 15 c WriteTC' 19 d C2Clk 19</p>	<p>S00 c19</p> <p>a DRef' 19 b Mode4' 19 c d</p> 
<p>S02 a19</p> <p>a pAlwaysCLK' 19 b pWaitCLK' 19 c WritelB 19 d Nt_Pt 17</p>	<p>S08 b10</p> <p>a paSh.0 11 b pAllowMDR_ 11 c pME 18 d *anon* 19</p>	<p>S51 f13</p> <p>a b WrTPC 19</p> 
<p>S10 a15</p> <p>a sh 13 b Push 13 c XByte' 13</p>	<p>S10 g23</p> <p>a pTC.2 15 b pTC.3 15 c Wait 19</p>	<p>S20</p> <p>a XBus_IB' 13 b EnLRotn' 13</p>
<p>LS32 c14</p> <p>a DispBr' 13 b EnDispBr.3A' 13 c EnDispBr2-3B' 13 d EnDispBr0-1' 13</p>	<p>LS32 c22</p> <p>a Link.0' 16 b Link.1' 16 c Link.2' 16 d Link.3' 16</p>	<p>LS32 d14</p> <p>a pRet' 16 b *anon* 19 c DesError 33 d M01 27</p>
<p>S38 i10</p> <p>a Q.00 05 b Q.15 05 c CarryIn 05 d CarryIn 05</p>	<p>S51 f13</p> <p>a Waitc1' 19 b WriteTPC' 19</p>	<p>S86 c10</p> <p>a PageCross 05 b MapRef 13 c Refresh 13 d</p> 
<p>S260 i15</p> <p>a IBEmptyErr 08 b *anon* 19</p>		<p>LS139 e5</p> <p>a b bank decode 27</p> 

DES Hardware Configuration Information

The chip is hard-wired in multiplexed control mode.
 The Master port is wired for writes only.
 The Auxiliary port is not used at all.
 Data always flows from the Master port to the Slave port.
 Des clock is high for c2 and c3, low for c1
 The microcode is synchronous, so MFLG' is ignored.
 Key parity errors, PAR', are ignored!

Zero out the high X bus when reading DES



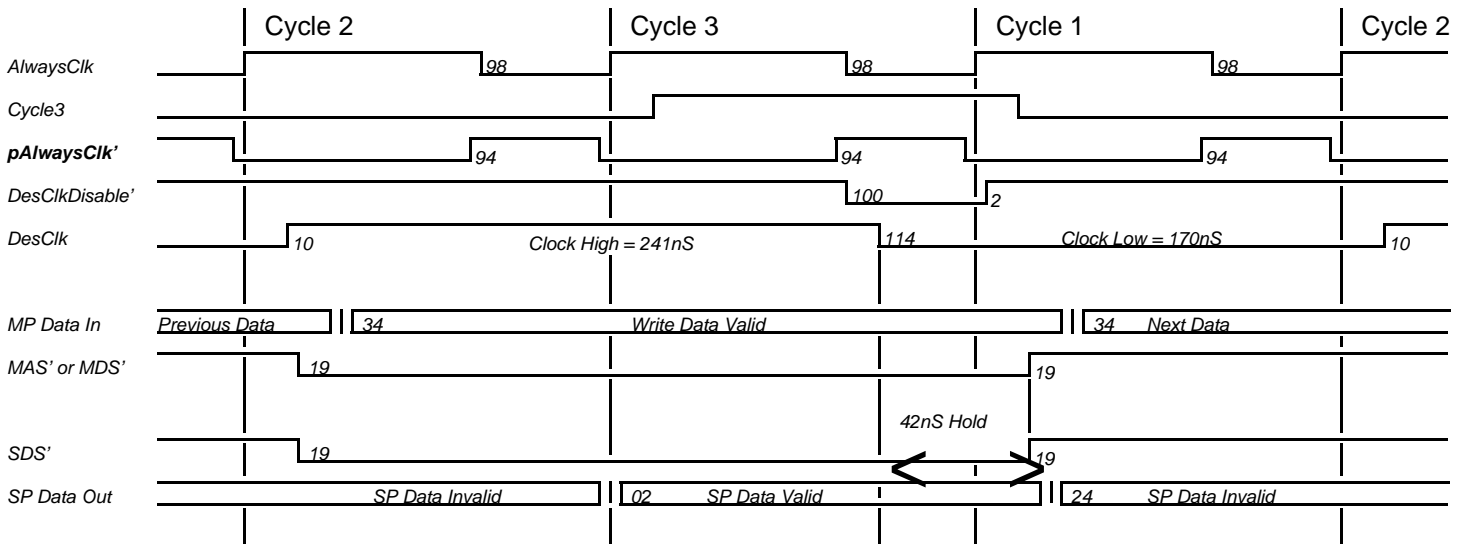
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project CPE	DES Encryption Hardware	File sCPE31.sil	Designer TonyWest.PA	Rev X	Date 4/8/83	Page 31
-------------------	----------------	-------------------------	--------------------	-------------------------	----------	----------------	------------

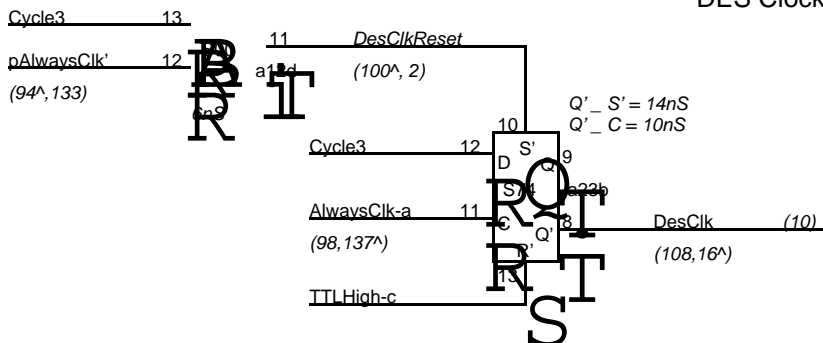
		AMD #	min.	max.	actual used	Notes
Clock & Reset						
Clock width HIGH		1	115		241	
Clock width LOW		2	115		170	
Clock Cycle		3	250		411	
Clock High to MAS'&MDS' High	Reset Hold	6	0	50	19	
MP and SP Strobe Times						
MAS' falling to MAS' rising (address)	MAS width Low	32	80		274	Can't exceed 1000, so have to watch out for WaitClk
MDS' falling to MDS' rising (data)	MDS width Low	44a	125	1000	274	
MDS' rising to MDS' falling	MDS Recovery	46	125		137	Can't exceed 1000, so have to watch out for WaitClk
SDS' falling to SDS' rising (data read)	SDS width Low	44a	125	1000	274	
SDS' rising to SDS' falling	SDS Recovery	46	125		137	
Clk falling to MDS' rising	MDS Hold	45	20	70	42	This is the difficult bit! See circuitry below.
Clk falling to SDS' rising	SDS Hold	46	20	70	42	This is the difficult bit! See circuitry below.
MAS Write into Master Port						
Data Valid to MAS' rising	Address Setup	36	55		268	
Data Hold after MAS' rising	Address Hold	37	60		243	
MDS Write into Master Port						
Data Valid to MDS' rising	Data Setup	47b	125		268	
Data Hold after MDS' rising	Data Hold	48	80		243	
SDS Read from Slave Port						
SDS falling to Data Valid	SP Access	49b		120		for last byte read
SDS rising to Data Invalid	SP Data Hold	50	5			
SDS falling to SFLG rising	SP Flag	51		125		

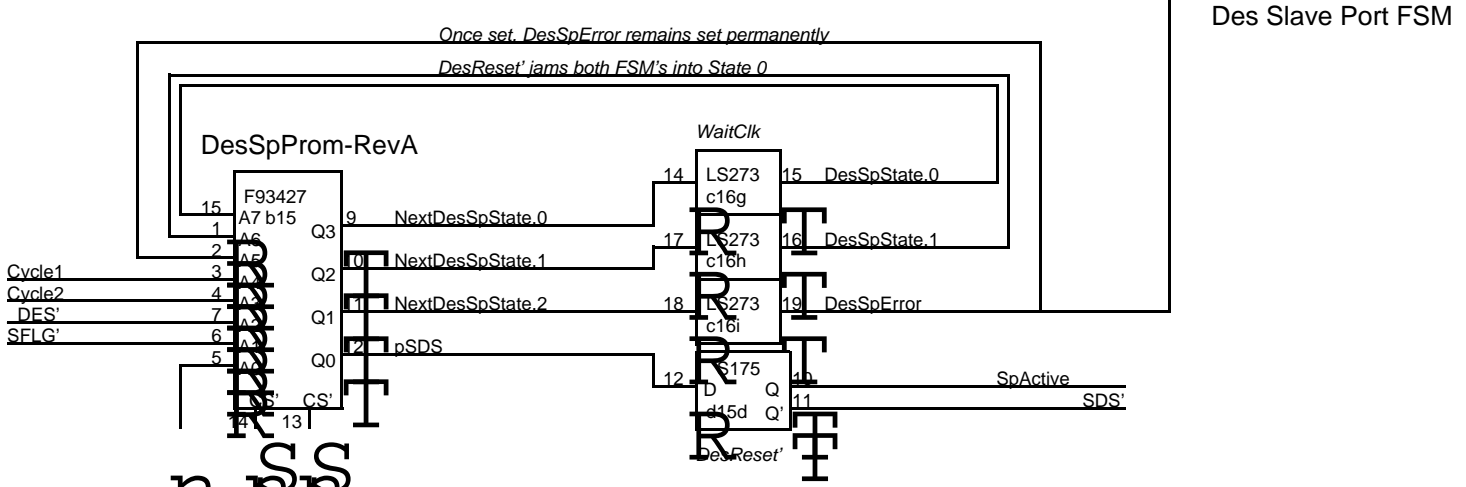
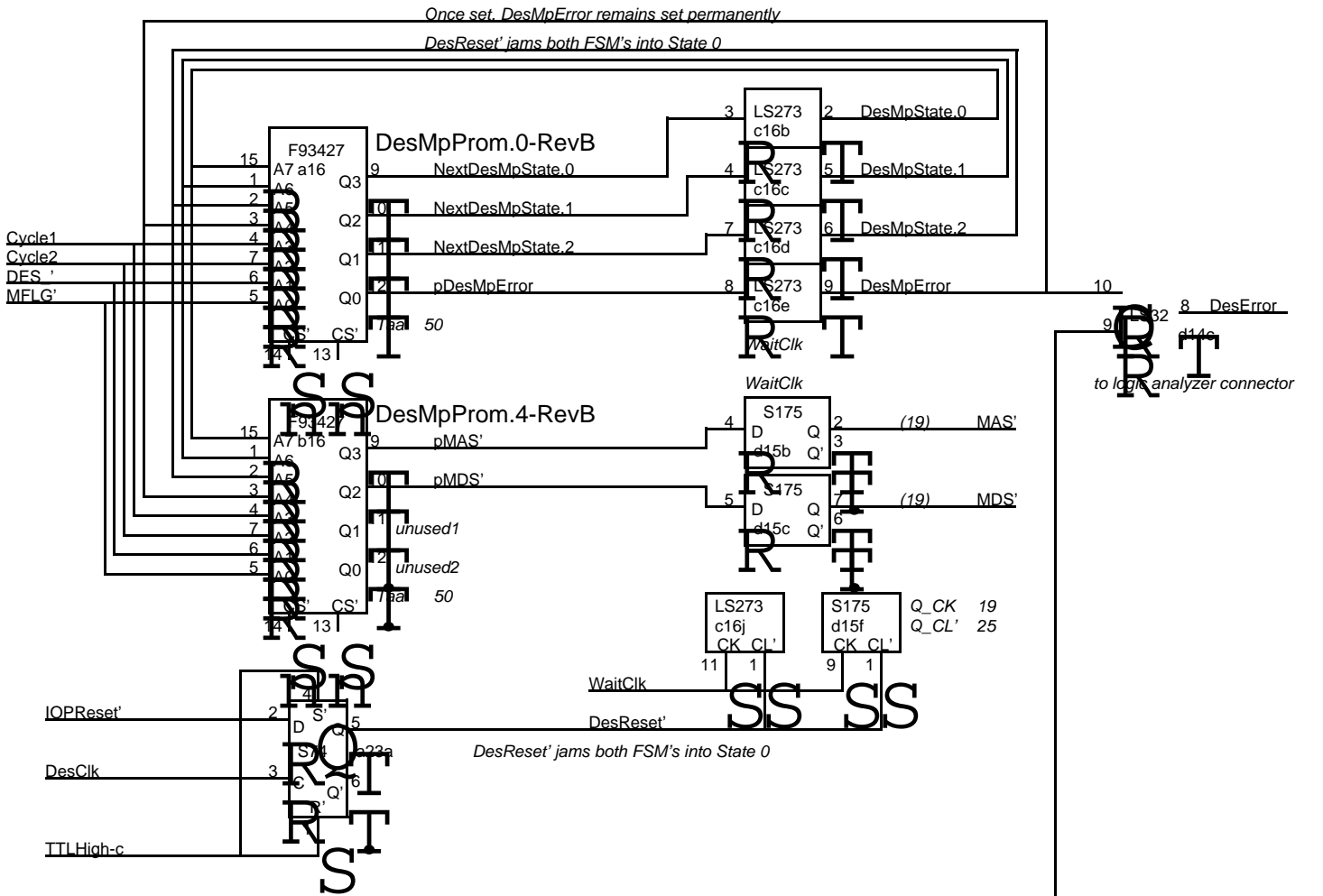
DES Clock Generator Timing

Note: Because of the requirement to hold MDS' and SDS' for 20 to 70 nanoseconds after DesClk falling, we bring DesClk down early in Cycle 3. MAS', MDS' and SDS' follow at the end of Cycle 3.

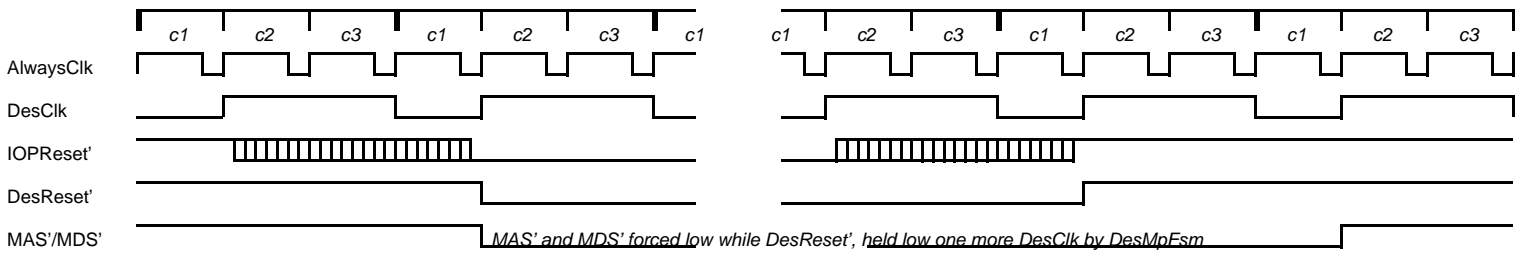


DES Clock Generator





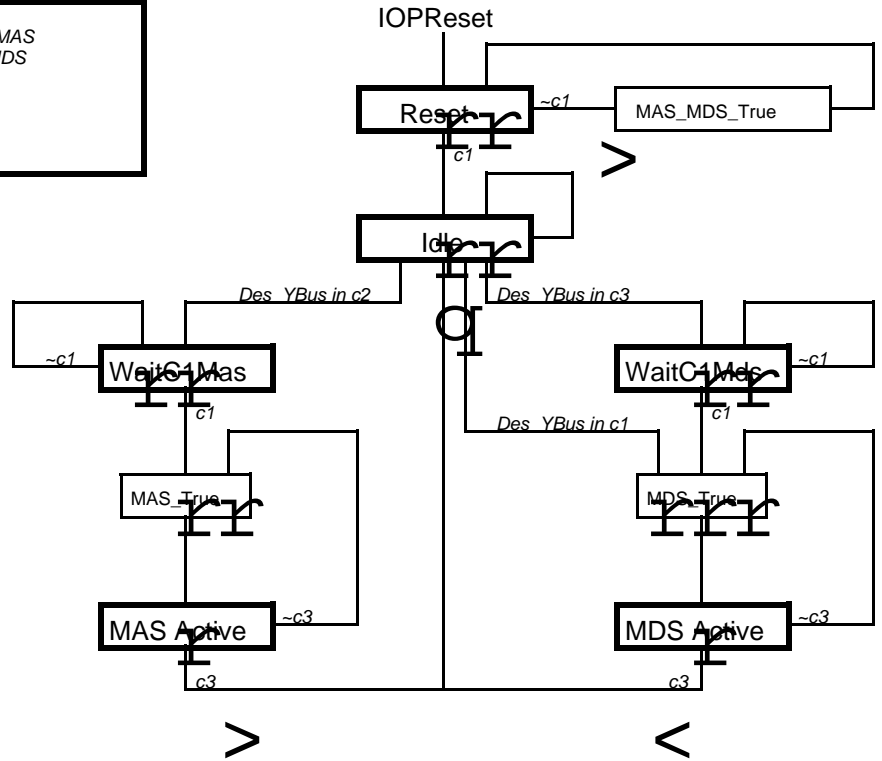
Reset Des Chip and FSM's with IOPReset'



Note on semantics of Master Port Writes

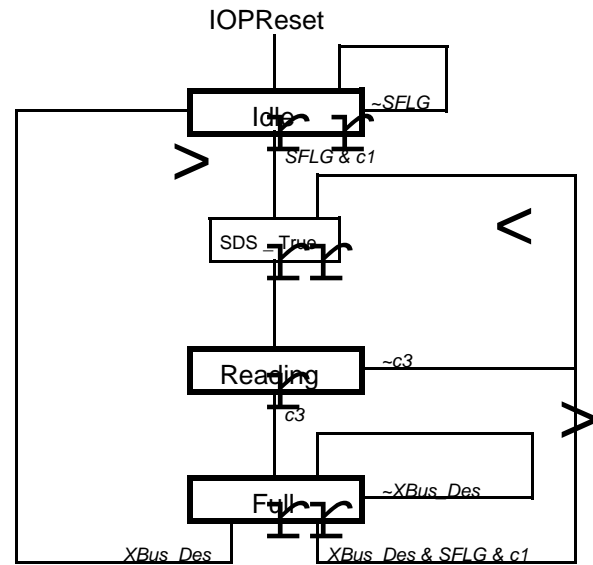
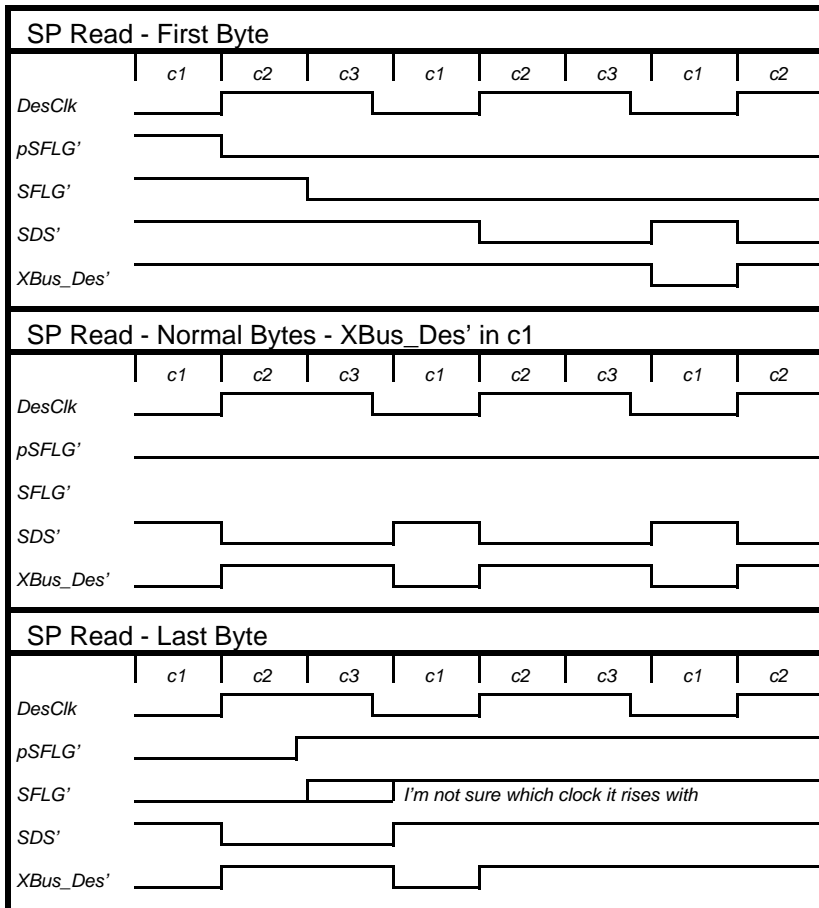
If you write to the Des chip in c2, it means write an address using MAS
 If you write to the Des chip in c1 or c3, it means write data using MDS
 You may have to wait for c1 in some of these cases.

The signals are shown logical-true.
 The implementation below inverts signals as required.



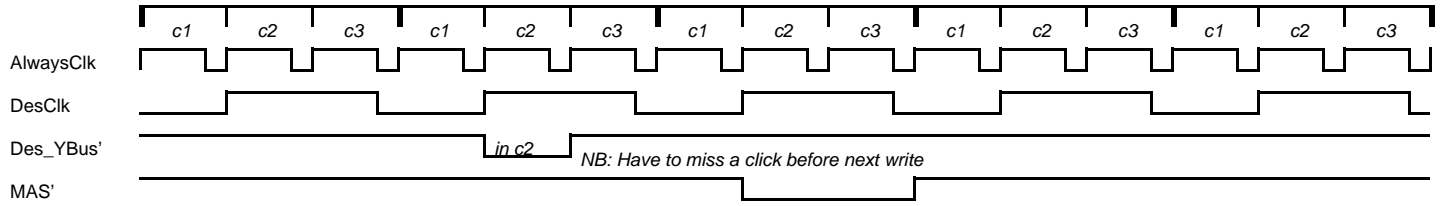
Master Port Finite-State Machine - Error handling of DesMpError signal is not shown

Slave Port Finite-State Machine - Error handling of DesSpError signal is not shown

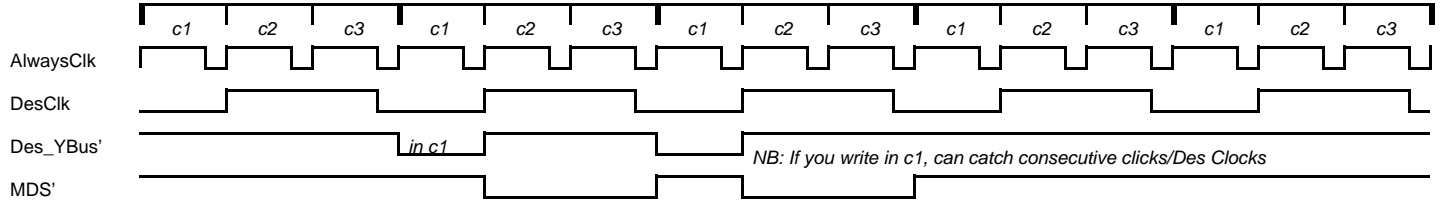


The signals are shown logical-true.
 The implementation below inverts signals as required.

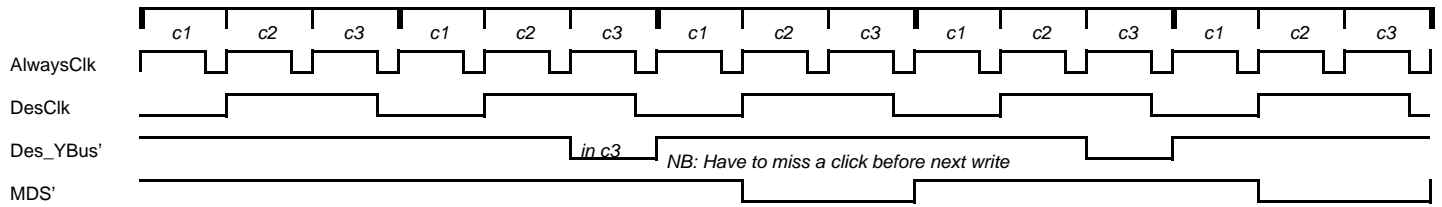
Write address into Des Master Port in C2



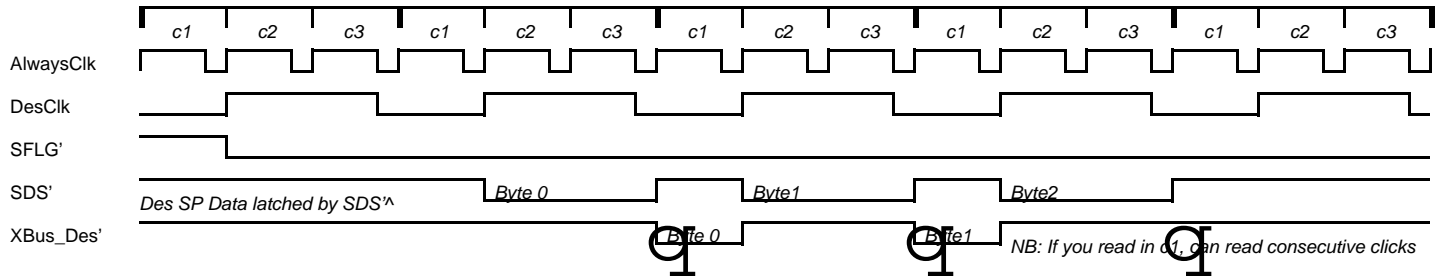
Write data into Des Master Port in C1



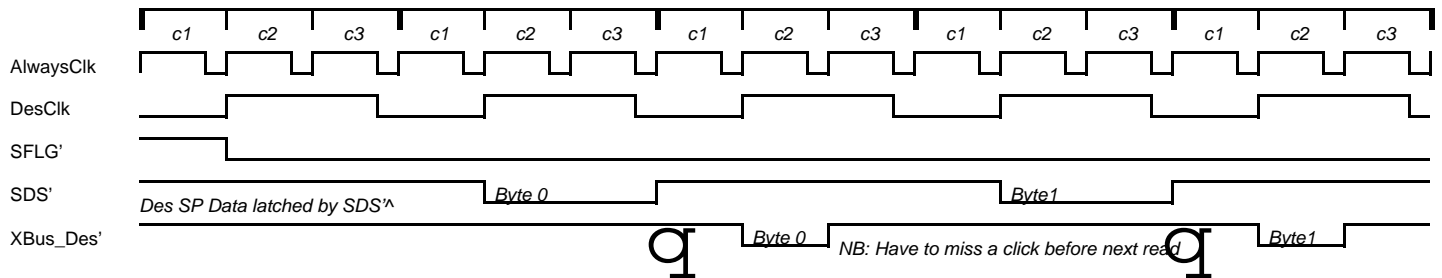
Write data into Des Master Port in C3



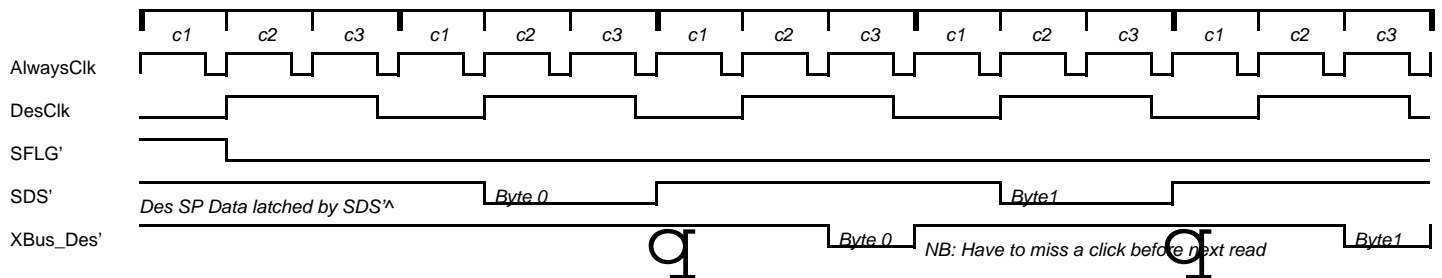
Read Data from Des Slave Port in C1

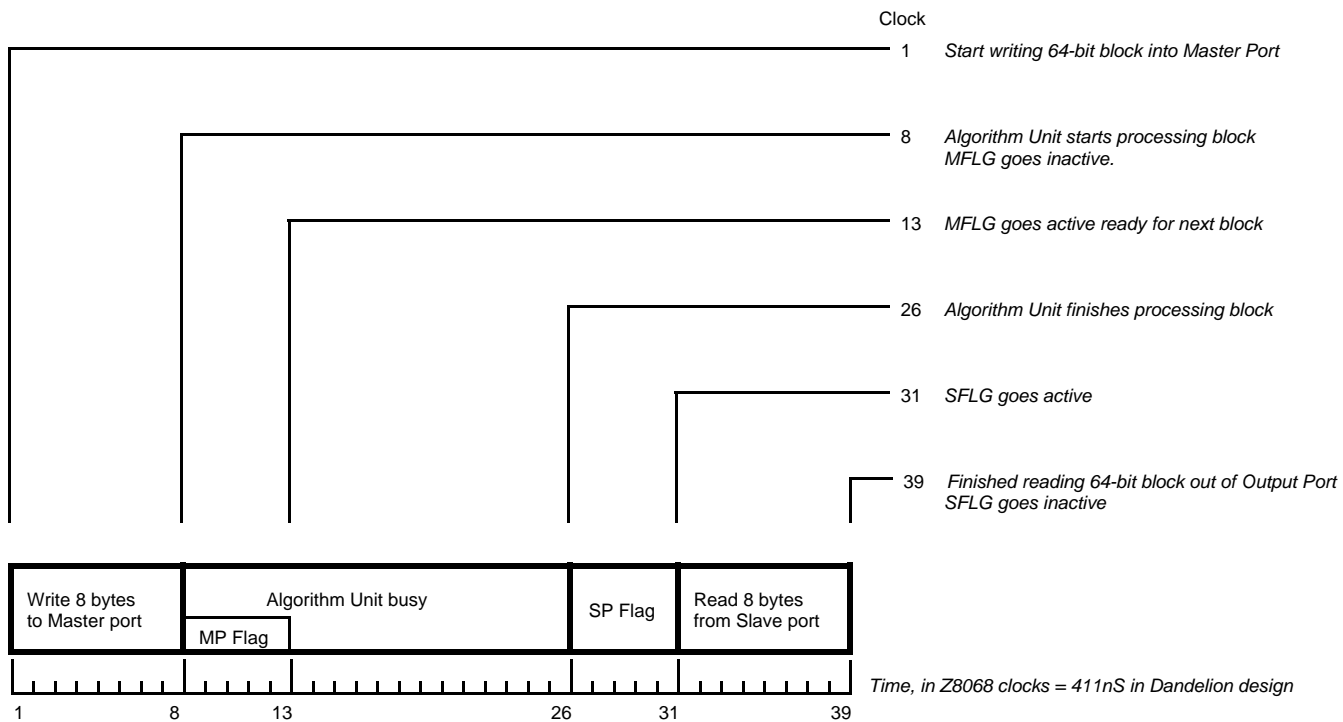


Read Data from Des Slave Port in C2



Read Data from Des Slave Port in C3





WARNING! This data is not guaranteed to be correct!

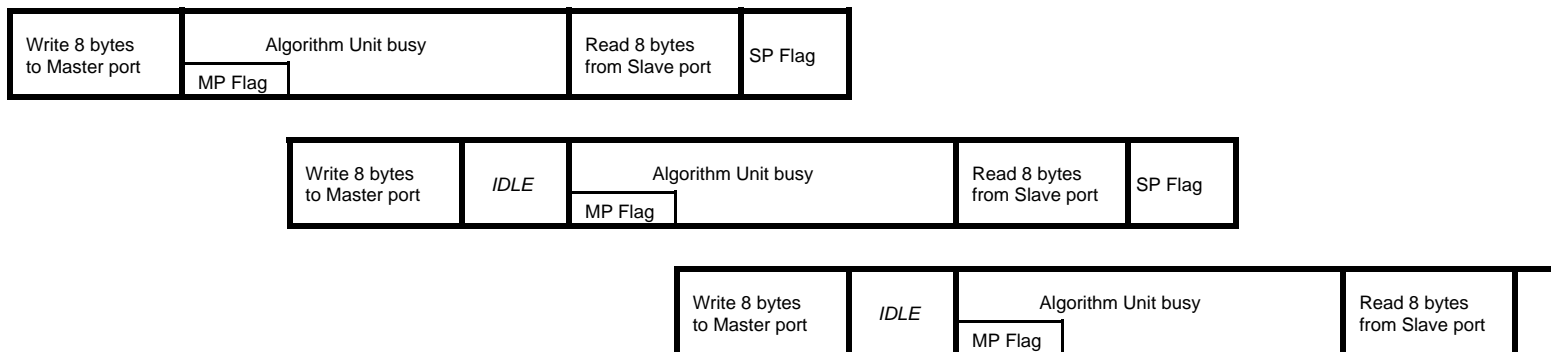
NOTES:

The longest operation in encrypting a block is the time it takes to get the data through the algorithm unit, 18 clocks. Therefore, this is the bottleneck in the pipelining scheme, and the software must aim to keep the Algorithm unit fully busy.

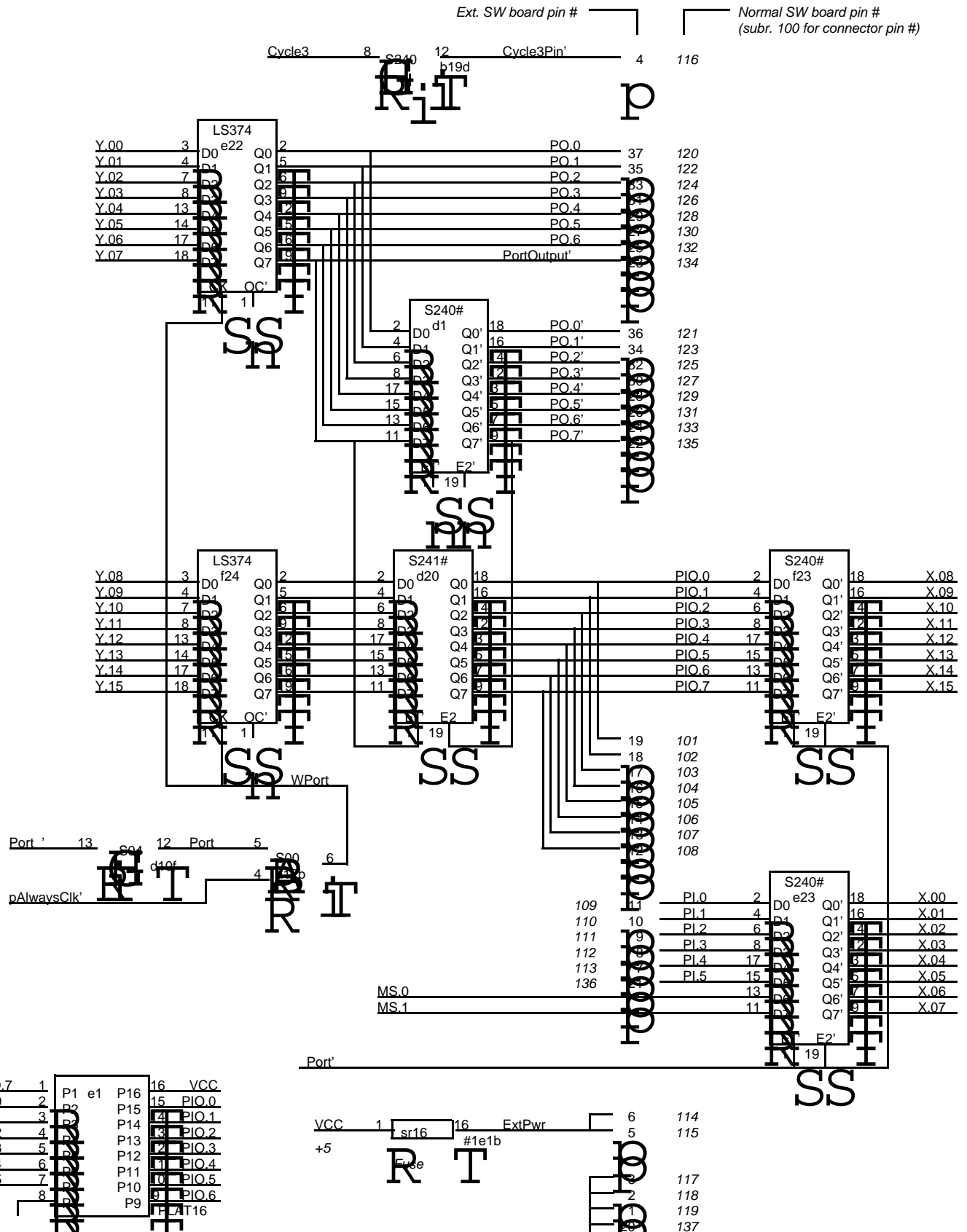
Apart from the first and last blocks, the time taken to encrypt the middle blocks is 18 clocks.

One possible pipelining scheme

WARNING! This data is not guaranteed to be correct!



WARNING! This data is not guaranteed to be correct!



77 Female Connector

Connection compatible with Dolphin interface:
 except for pin 116 (N.C. in D0) & pin 136 (GND in D0)

Board I/O pin numbers were changed from the normal D0/D1ion Stichweld board (for some unknown reason)