Dandelion Central Processor
with
Extended Control Store and DES Encryption

R e v i s i o n W

Notes:

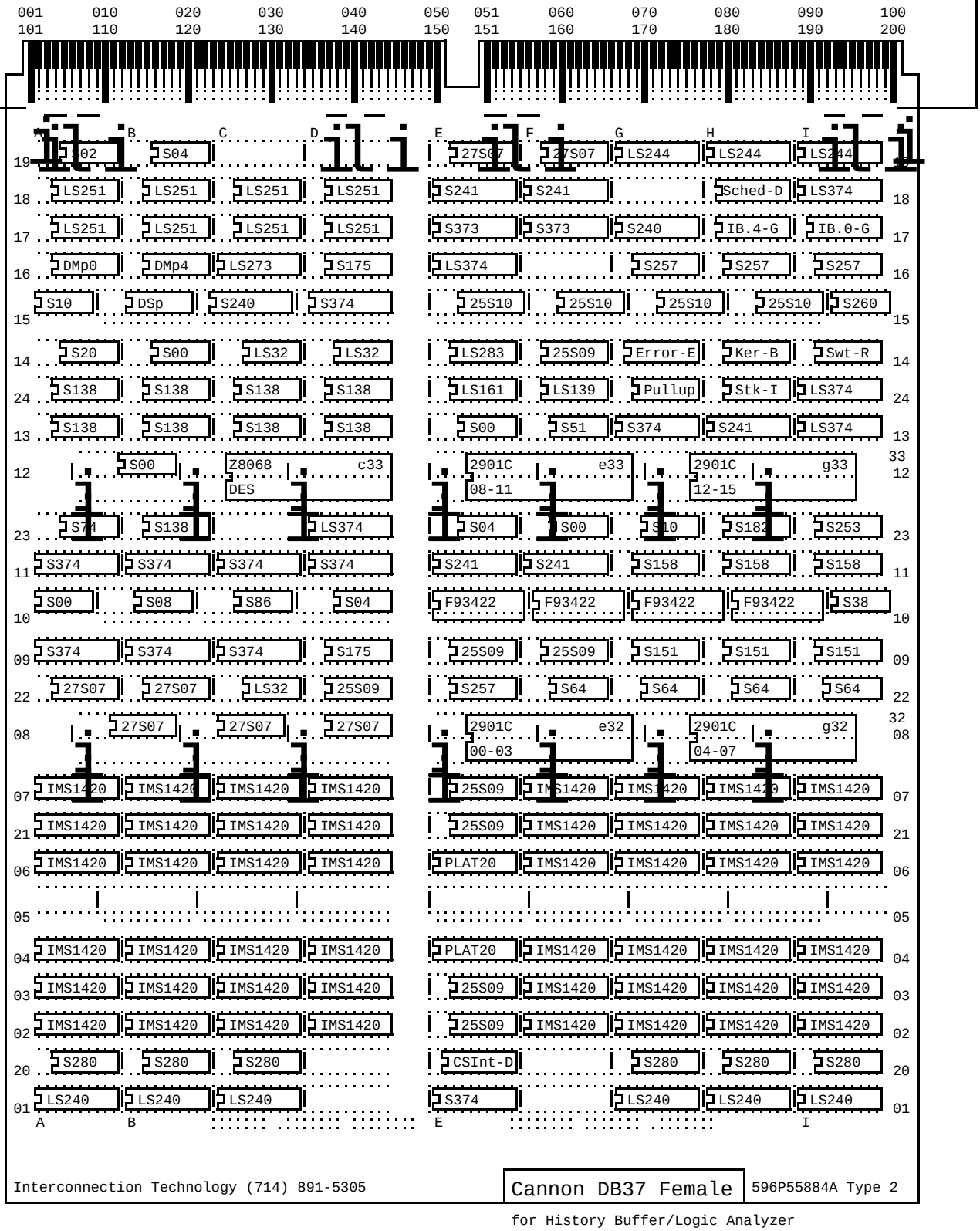
The drawings are filed in[Indigo]<Dandelion>CPE>Drawings>sCPE*.sil
and in: [Indigo]<Dandelion>CPE>Archive>sCPE-W-DrawingsOnly.dmsil
Print press file is in [Indigo]<Dandelion>CPE>Archive>sCPE-W.press

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.lb5 from ED/SDD
6 = Sil.lb6 from ED/SDD
7 = Sil.lb7 from ED/SDD
8 = CAS.lb8 from ED/SDD
9 = Sil.lb9 from ED/SDD

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



for History Buffer/Logic Analyzer

Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project	Reference	File	Designer	Rev	Date	Page
	CPE	High-Density Stitchweld Layout	sCPE01.sil	TonyWest.PA	W	1/24/83	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	S04	SPARE	SPARE		27S07	27S07	LS244	LS244	LS244
18	LS251	LS251	LS251	LS251		S241	S241	SPARE	ScheduleProm	LS374
17	LS251	LS251	LS251	LS251		S373	S373	S240	IBProm 4	IBProm 0
16	DesMpProm.0	DesMpProm.4	LS273	S175		LS374	SPARE	S257	S257	S257
15	S10	DesSpProm	S240	S374		25S10	25S10	25S10	25S10	S260
14	S20	S00	LS32	LS32		LS283	25S09	Error Prom	KernelPCProm	Switch Prom
24	S138	S138	S138	S138		LS161	LS139	Pullups	Stack Prom	LS374
13	S138	S138	S138	S138		S00	S51	S374	S241	LS374
12	S00		Z8068 Des Processor	c33		2901C	e33		2901C	g33
23	S74	S138	SPARE	LS374		S04	S00	S10	S182	S253
11	S374	S374	S374	S374		S241	S241	S158	S158	S158
10	S00	S08	S86	S04		F93422	F93422	F93422	F93422	S38
09	S374	S374	S374	S175		25S09	25S09	S151	S151	S151
22	27S07	27S07	LS32	25S09		S257	S64	S64	S64	S64
08	27S07	27S07	27S07			2901C	e32		2901C	g32
07	1420	1420	1420	1420		25S09	1420	1420	1420	1420
21	1420	1420	1420	1420		25S09	1420	1420	1420	1420
06	1420	1420	1420	1420		Platform 20	1420	1420	1420	1420
05	SPARE	SPARE	SPARE	SPARE		SPARE	SPARE	SPARE	SPARE	SPARE
04	1420	1420	1420	1420		Platform 20	1420	1420	1420	1420
03	1420	1420	1420	1420		25S09	1420	1420	1420	1420
02	1420	1420	1420	1420		25S09	1420	1420	1420	1420
20	S280	S280	S280	SPARE		CSInt Prom	SPARE	S280	S280	S280
01	LS240	LS240	LS240	SPARE		S374	SPARE	LS240	LS240	LS240

Cannon DB-37 Female

XEROX PARC-CSL	Project	Reference	File	Designer	Rev	Date	Page
	CPE	Dandelion CP - Functional Layout	sCPE02.sil	TonyWest.PA	W	1/24/83	02

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

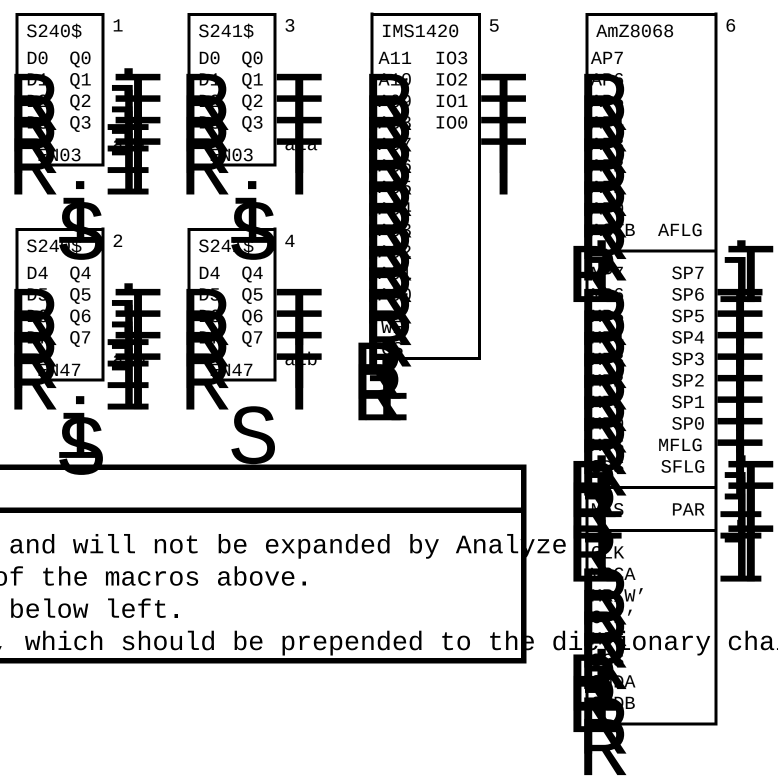
Prom files are stored on [Indigo]<Dandelion>CPE>Proms>*

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

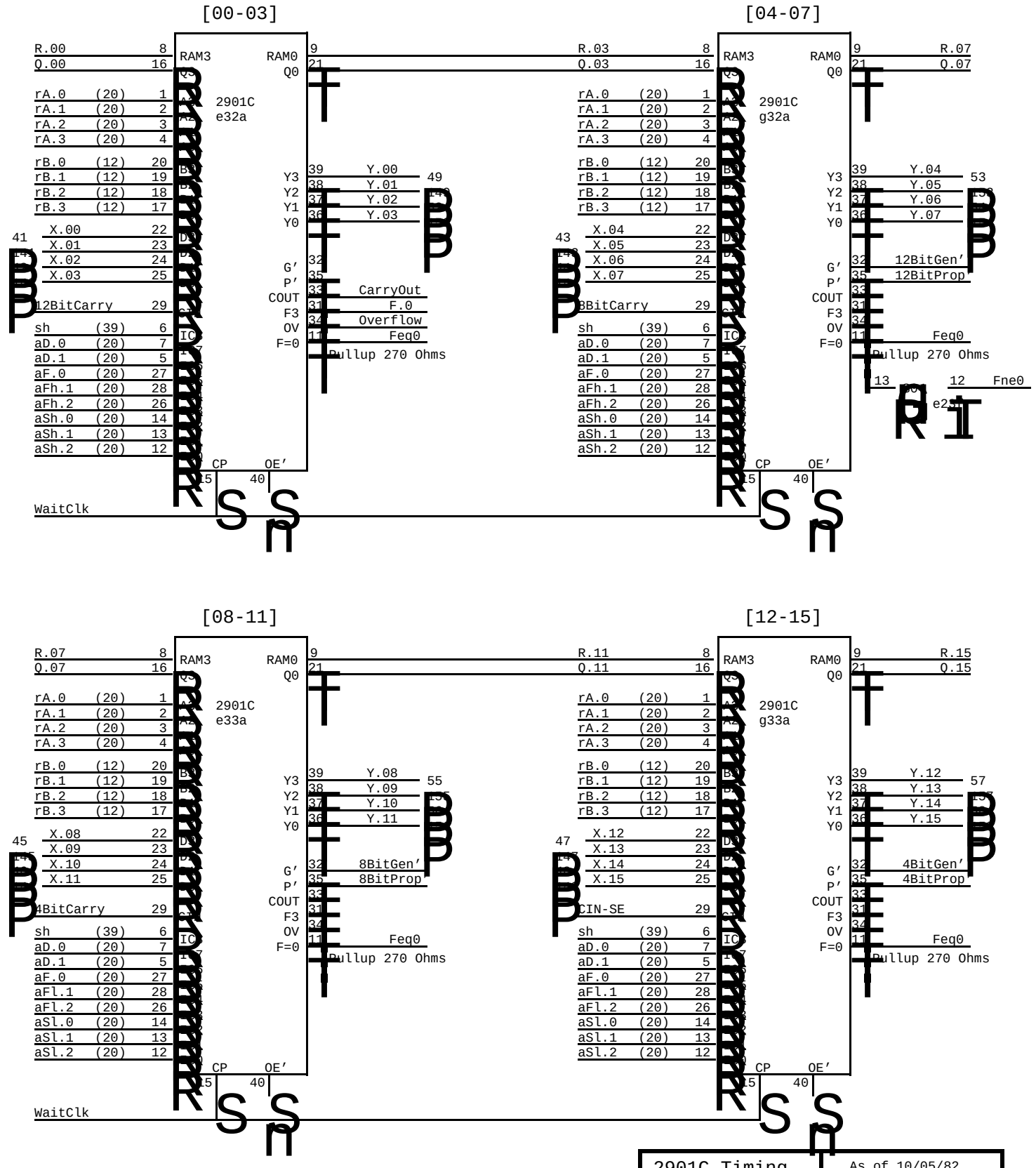


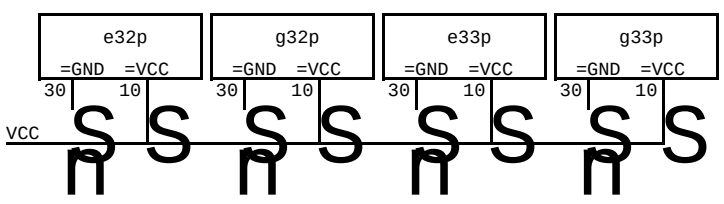
Important Notes:

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!

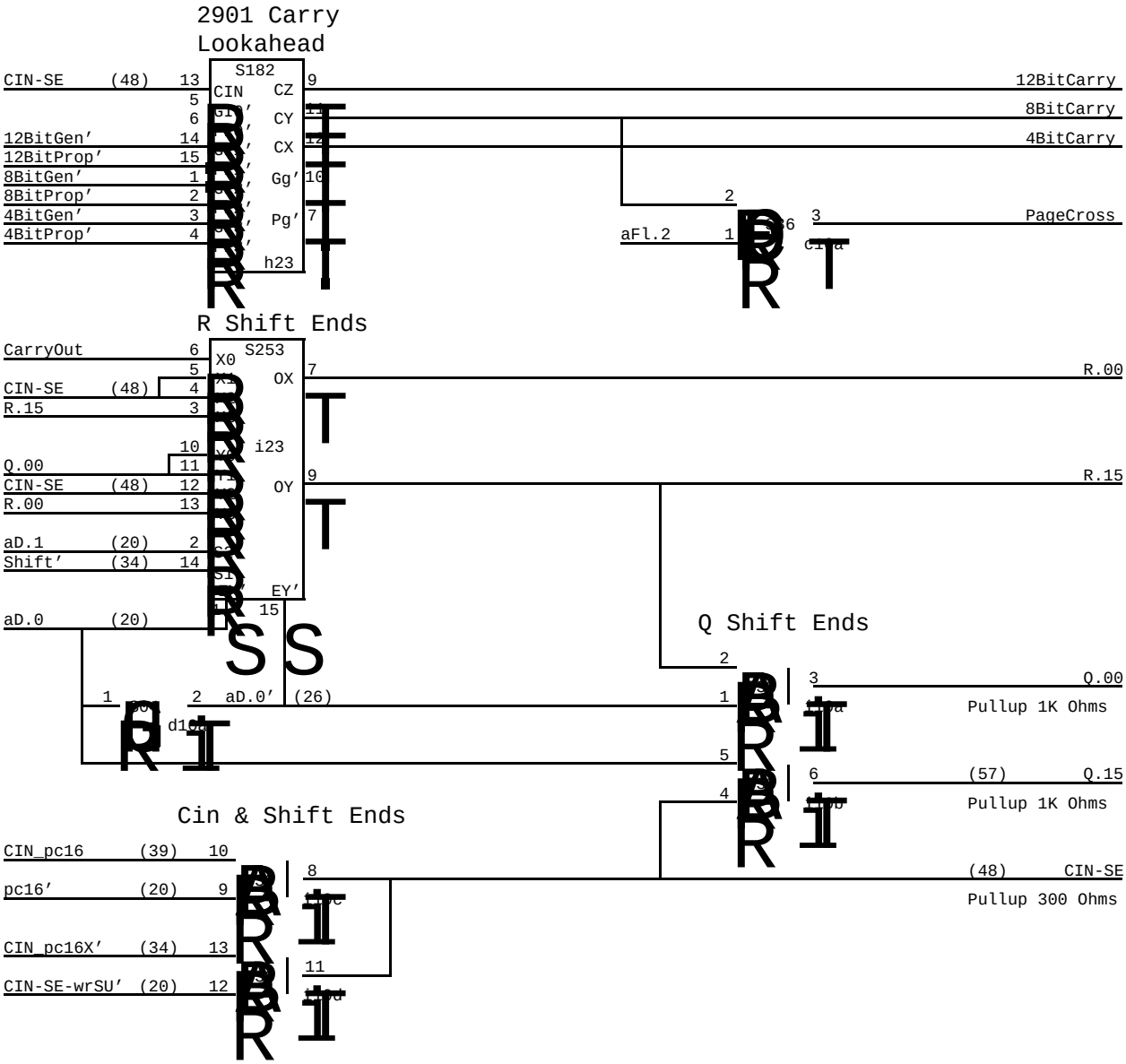
XEROX PARC-CSL	Project	Reference	File	Designer	Rev	Date	Page
	CPE	Prom Index & Font 4 Macros	sCPE03.sil	TonyWest.PA	W	1/24/83	03





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

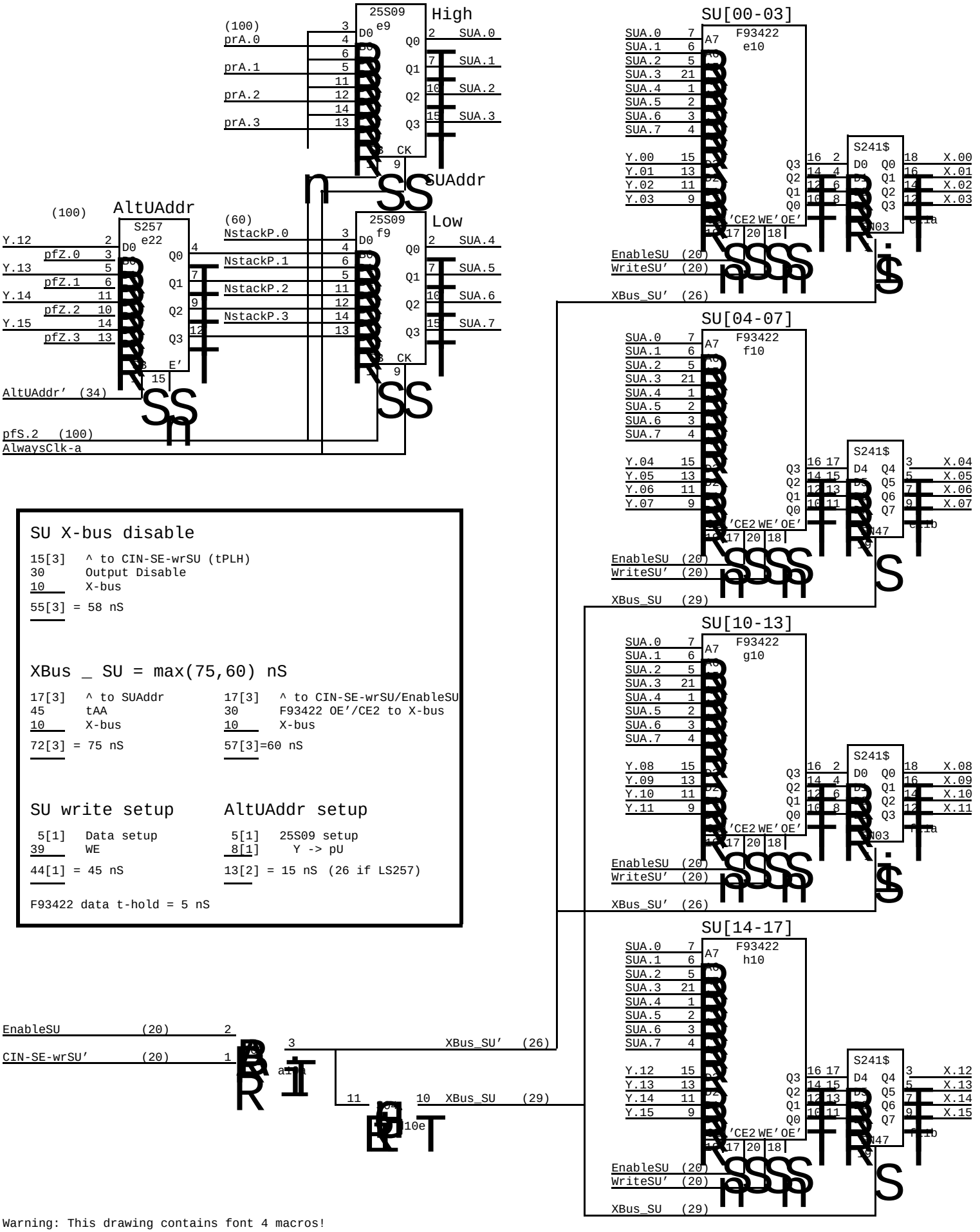
XEROX PARC-CSL	Project		File	Designer	Rev	Date	Page
	CPE	2901 ALU	sCPE04.sil	Garner.PA	W	1/14/83	04



aD.1	Shift	Shift configuration
0	0	Cout — R — ~ — Q — ~Cin
0	1	Cin — R — ~ — Q — ~Cin
1	0	Cin — R — Cin
1	1	R

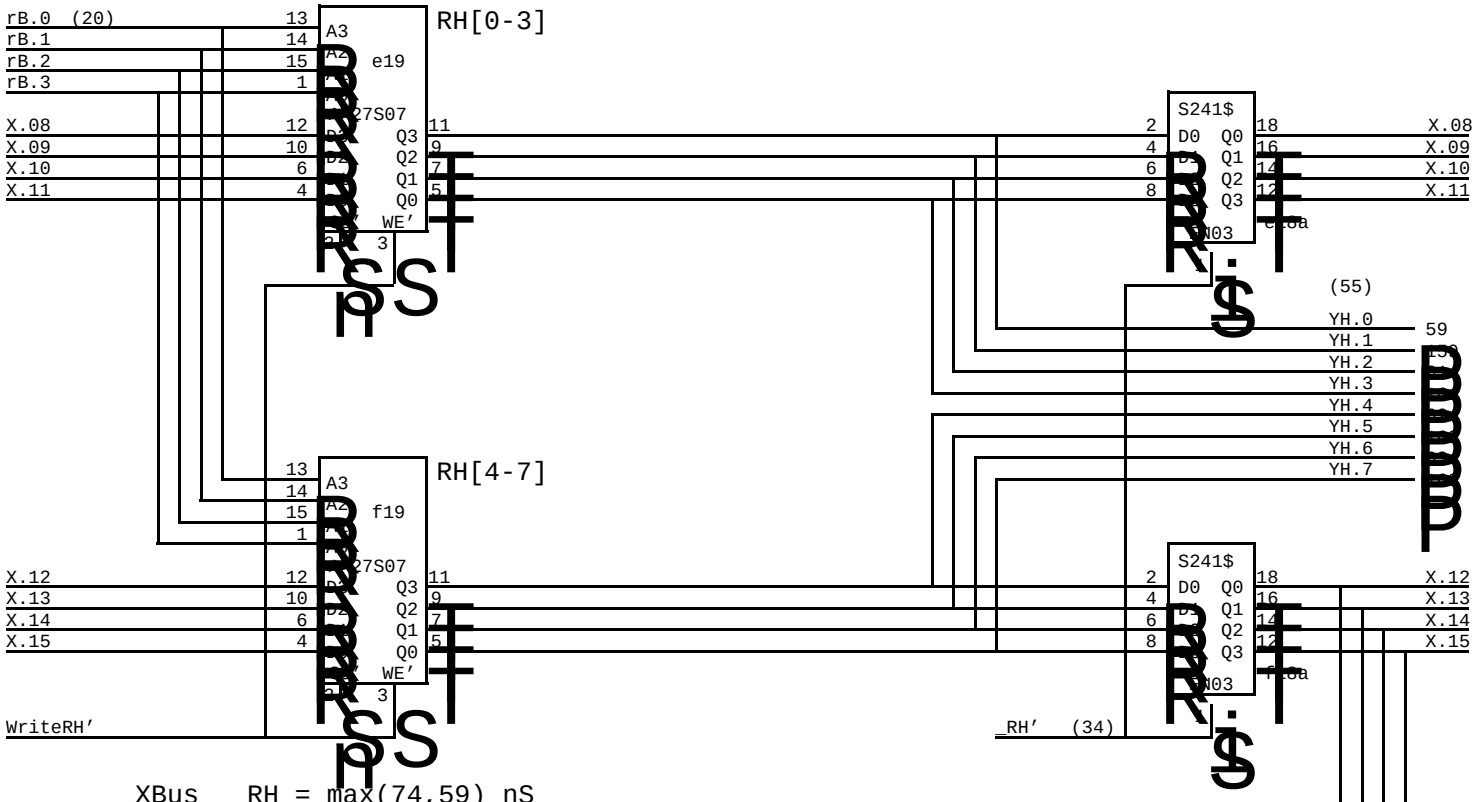
aD.0 = 0 implies right shift

XEROX PARC-CSL	Project		File	Designer	Rev	Date	Page
	CPE	Lookahead, Shift Ends, Cin	sCPE05.sil	Garner.PA	W	1/14/83	05



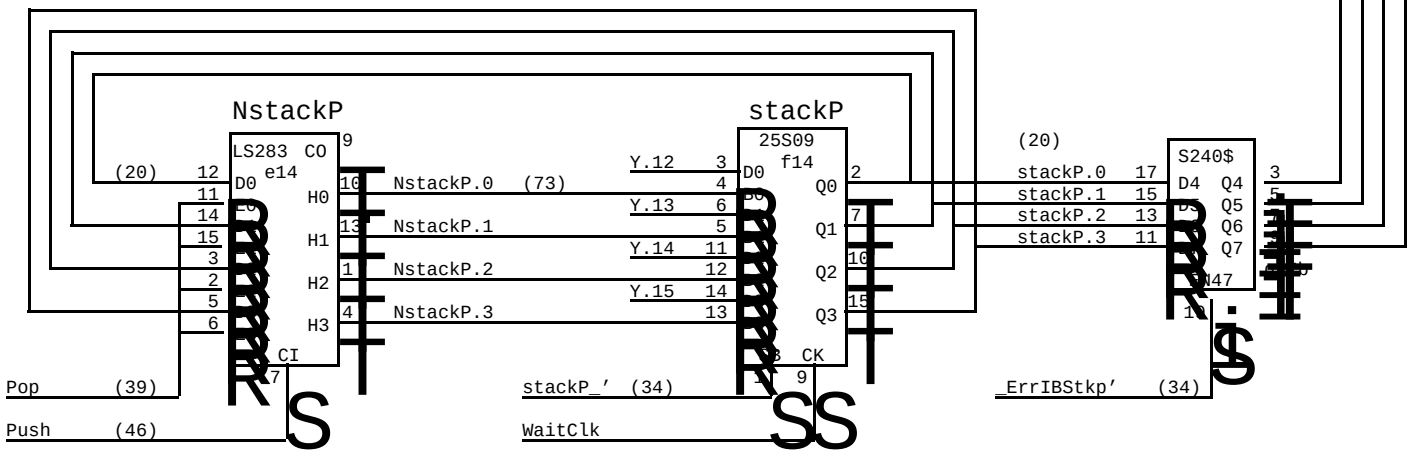
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project CPE	SU Registers	File sCPE06.sil	Designer TonyWest.PA	Rev W	Date 1/24/83	Page 06



XBus _ RH = max(74,59) nS

17[3]	^ to rB'	34	^ to _RH'
35	S189 tAA (wr recovery = 35 nS)	15	S241 EN' to X-bus
9	YH to X	10	X-bus
10	X-bus	59 nS	
71[3] = 74 nS			



Push Timing

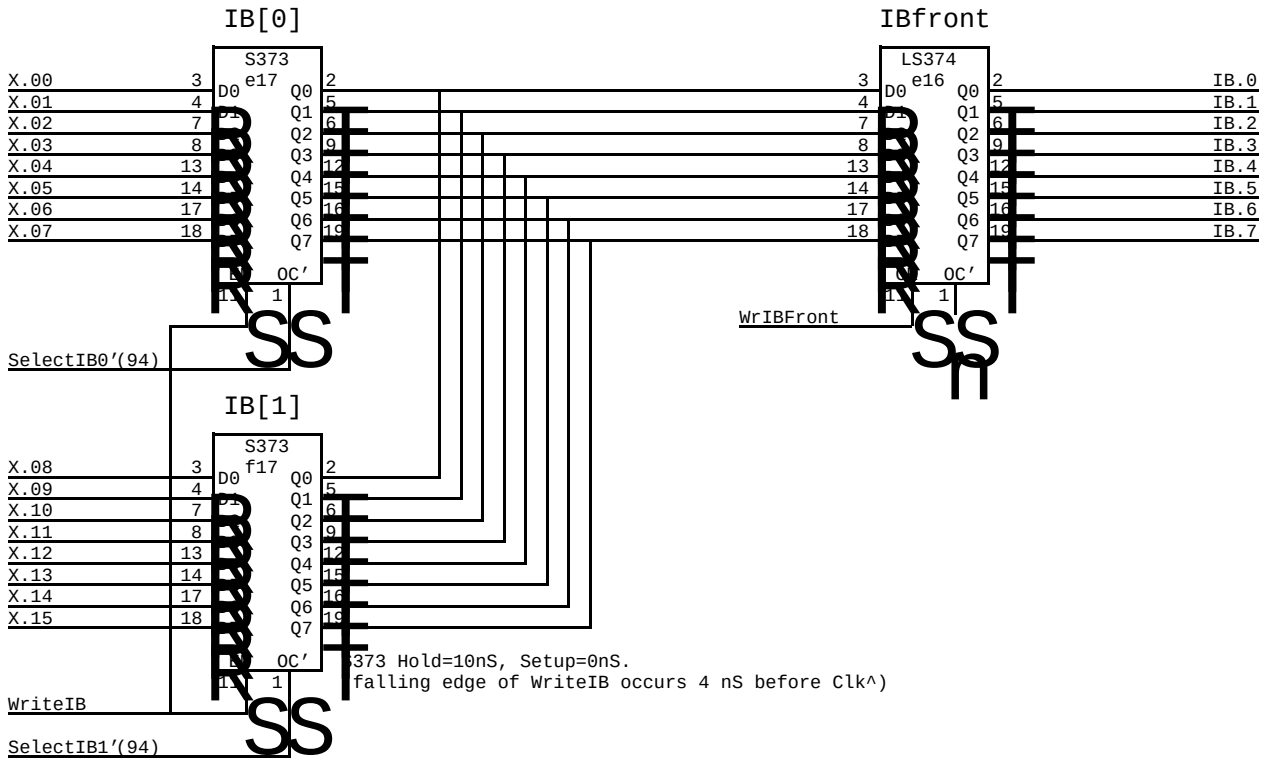
46	^ to Push
24[3]	Push to NstackP
5[1]	25S09 setup
75[4] = 79 nS	

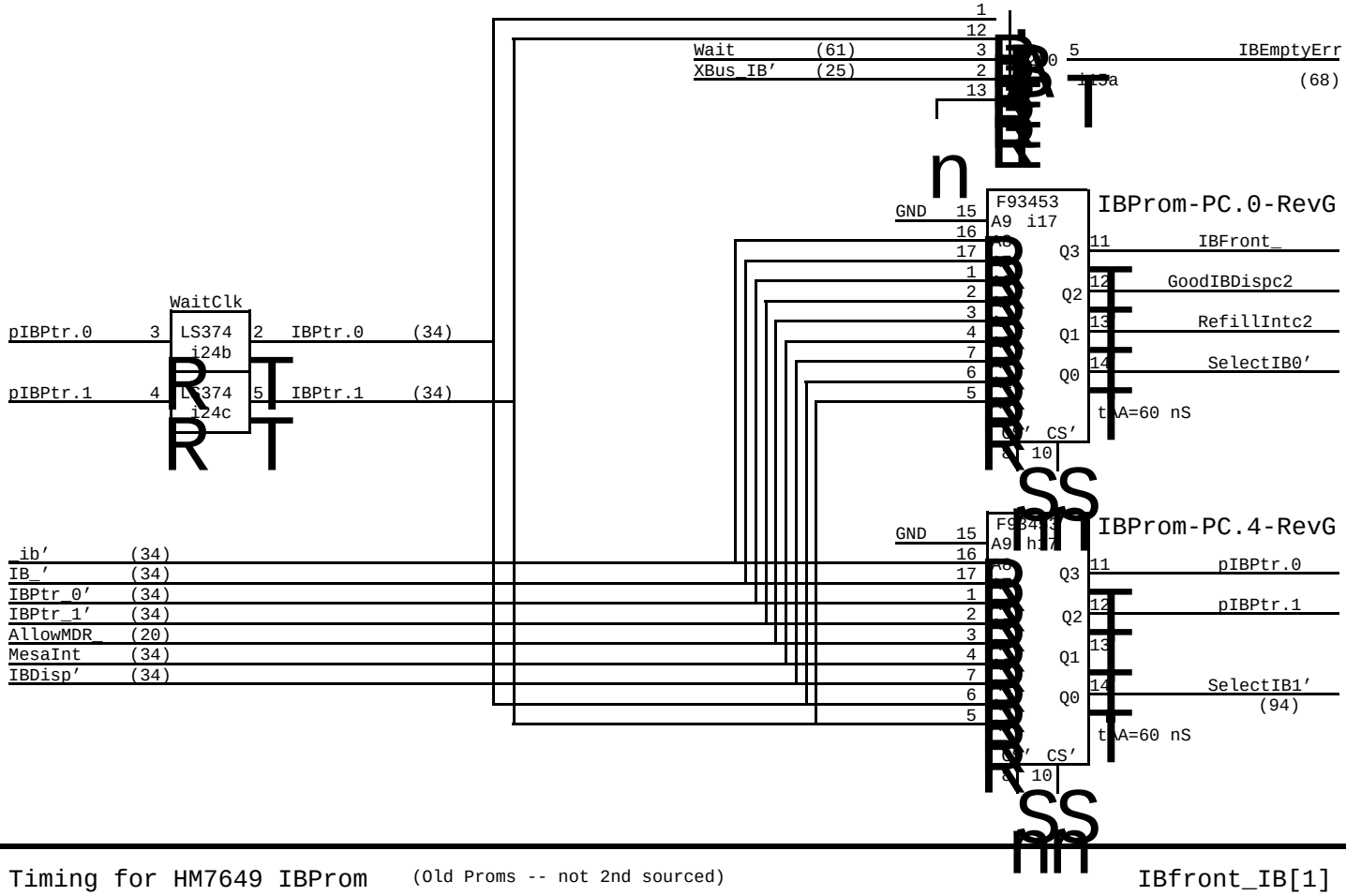
XBus _ stackP = max(59, 38) nS

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 W	Date 1/14/83	Page 07





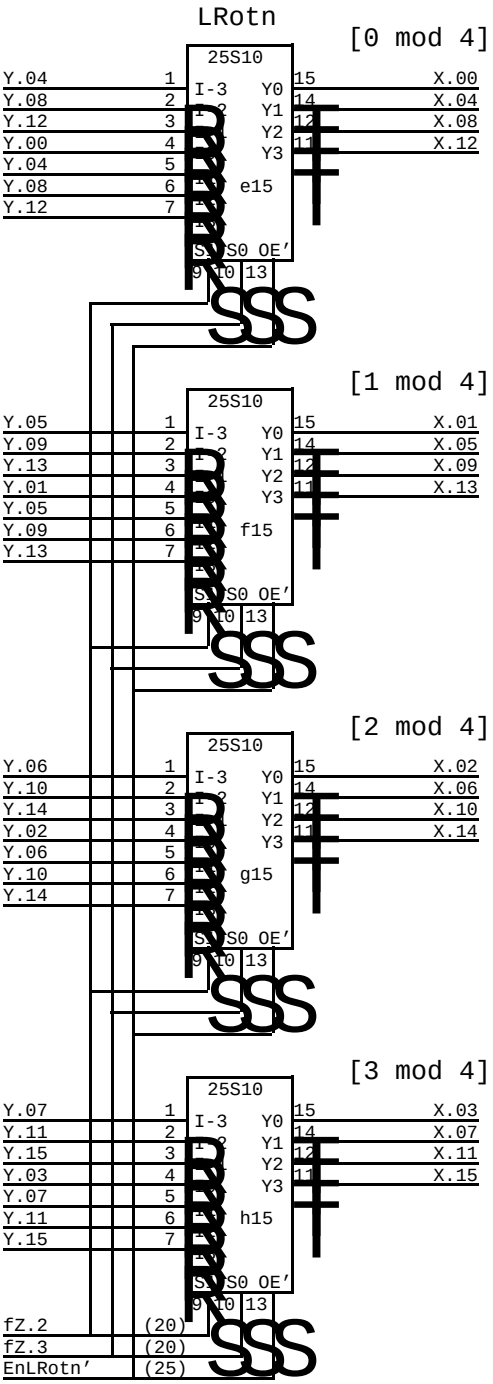
Timing for HM7649 IBProm (Old Proms -- not 2nd sourced)

IBfront_IB[1]

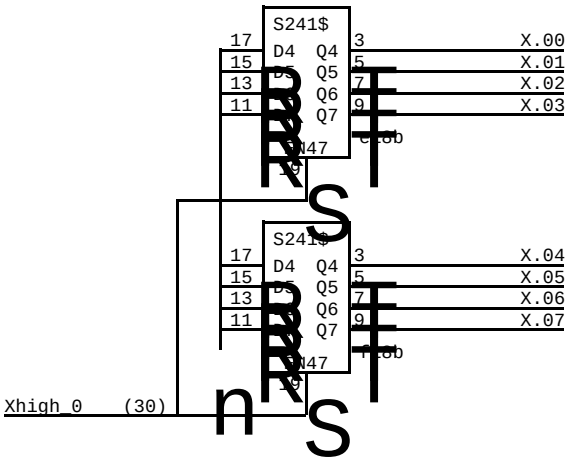
IBFront _ Xbus = (x+37, x+36) nS

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	tAA
- 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	

XEROX PARC-CSL	Project CPE	Instruction Buffer	File sCPE08.sil	Designer Garner.PA	Rev W	Date 1/14/83	Page 08



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	

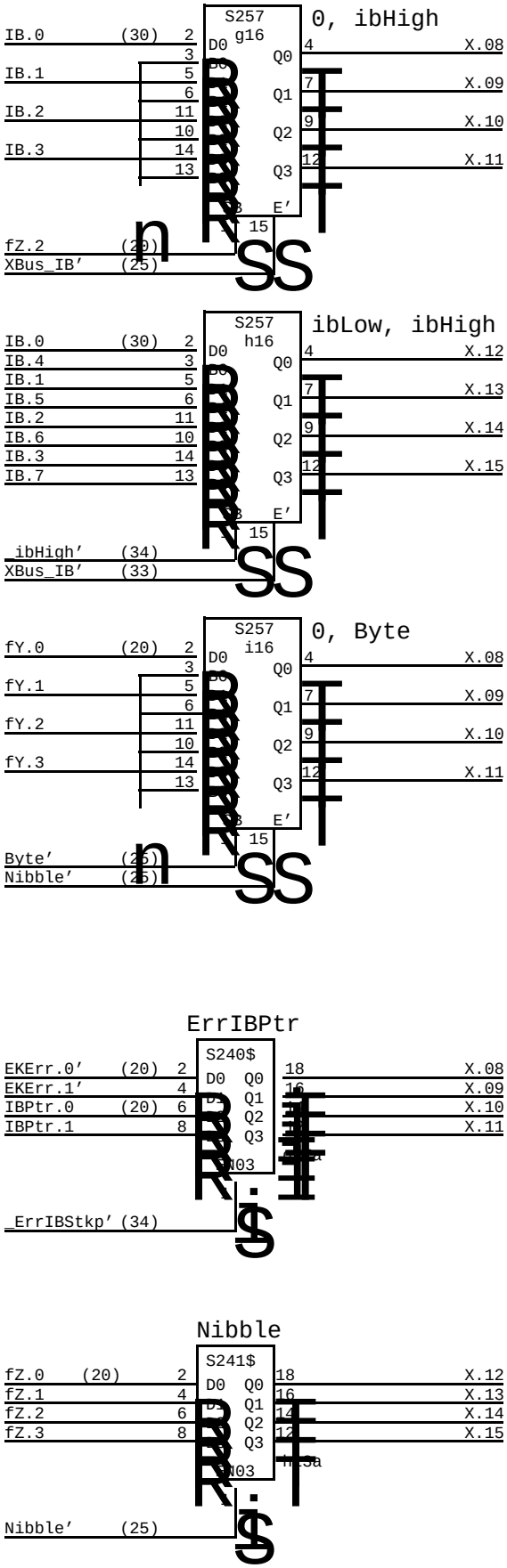
20	^ to fZ.2
20	25S10 Select to X-bus
10	X-bus
50 nS	

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!

XEROX PARC-CSL	Project		File	Designer	Rev	Date	Page
	CPE	X Bus: LRotn, RH, ZeroHigh X	sCPE09.sil	Garner.PA	W	1/14/83	09



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

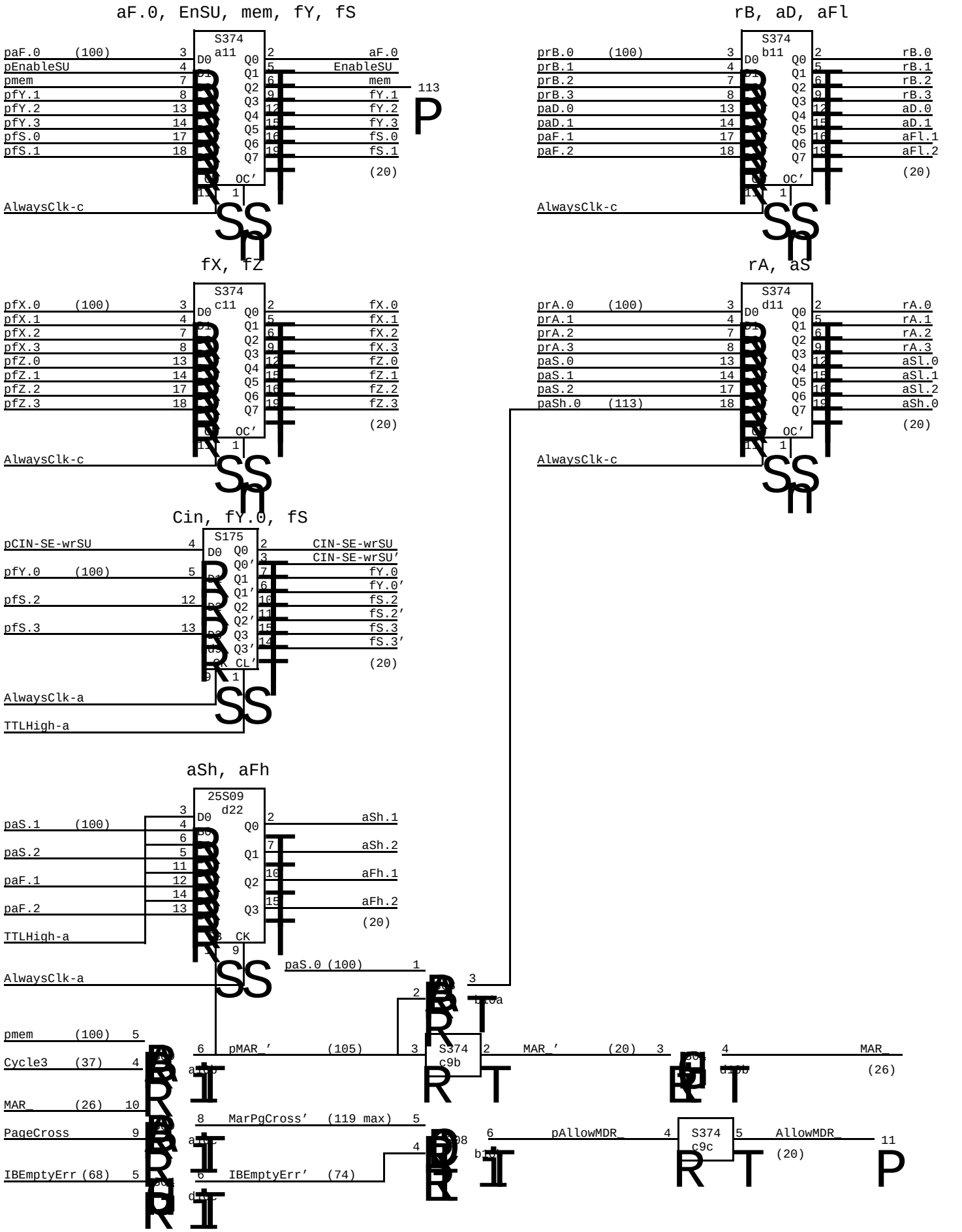
25 ^ to Byte'
15 S257 SB to Xbus
10 X-bus

50 nS

See stackP timings for ErrIBPtr

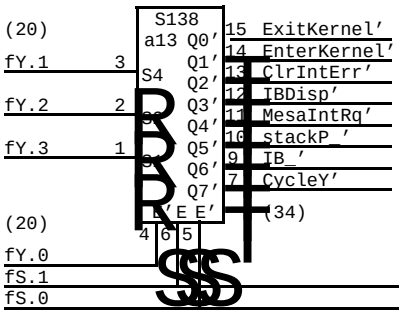
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project		File	Designer	Rev	Date	Page
	CPE	X Bus: IB, Constants, ErrIntstackP	CPE10.sil	Garner.PA	W	1/14/83	10

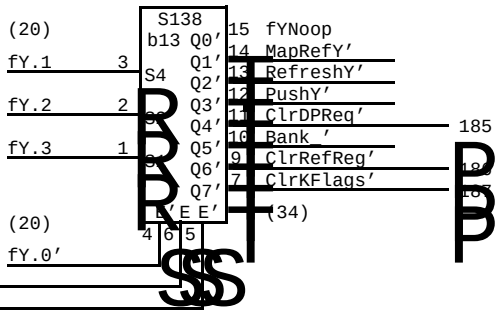


XEROX PARC-CSL	Project CPE	Microinstruction Register	File sCPE11.sil	Designer Garner.PA	Rev W	Date 1/14/83	Page 11

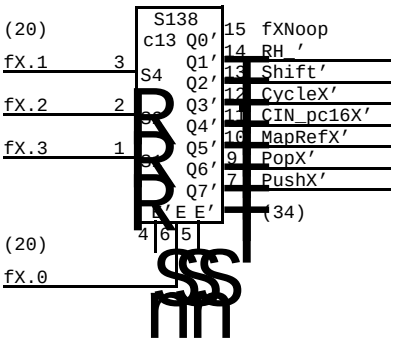
fYNorm - Low



fYNorm - High



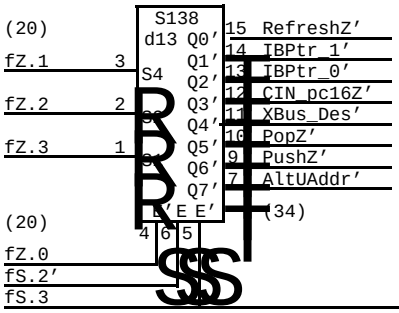
fX - High



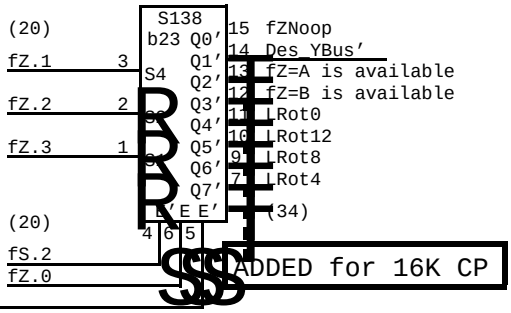
Bank_' replaces ClrIOPReq', which was connected to 184 on backplane

fX - Low is pCall/pRet

fZNorm - Low



fZNorm - High



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 ClrIOPReq', which is now obsolete. ClrIOPReq' was also connected to backplane pin 184.

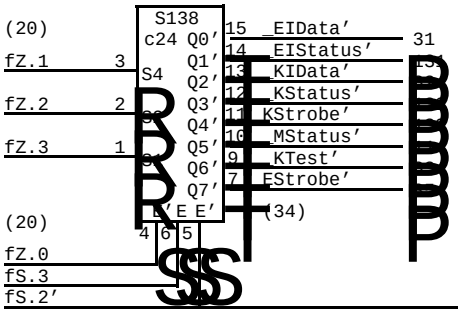
The meaning of Des_YBus' depends on which cycle it is activated in:

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

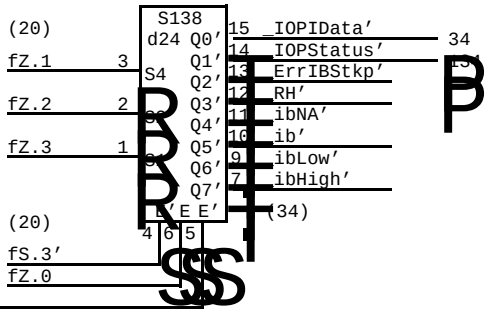
XBus_Des' can be activated in any cycle
See page sCPE31 for details of DES logic

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

IOXIn - Low



IOXIn - High



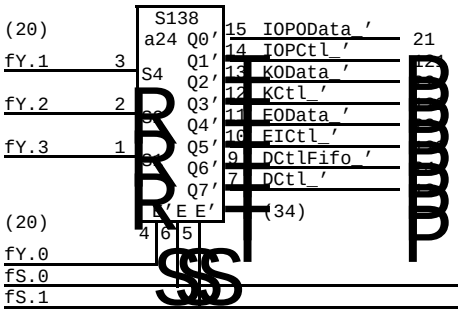
S138 Timing:

Propagation delays

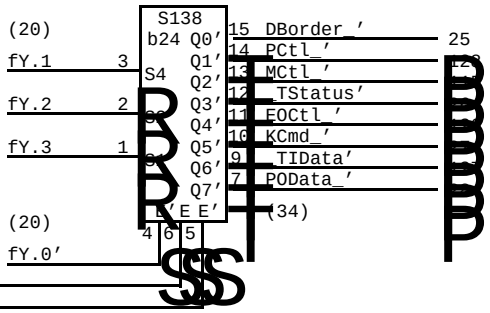
from Selects to Q' 14nS
from Enables to Q' 13nS

These timings are very conservative!

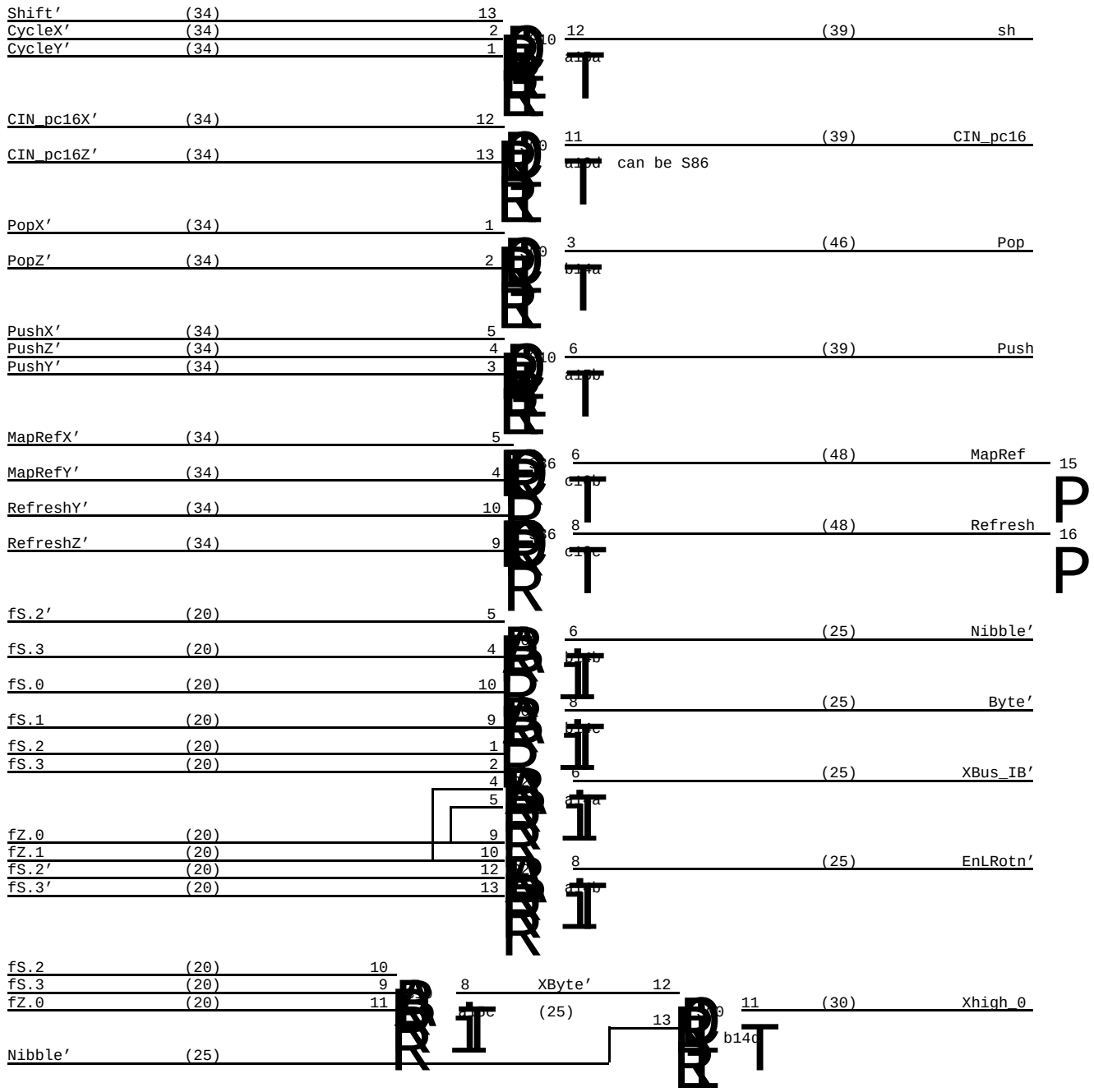
IOOut - Low



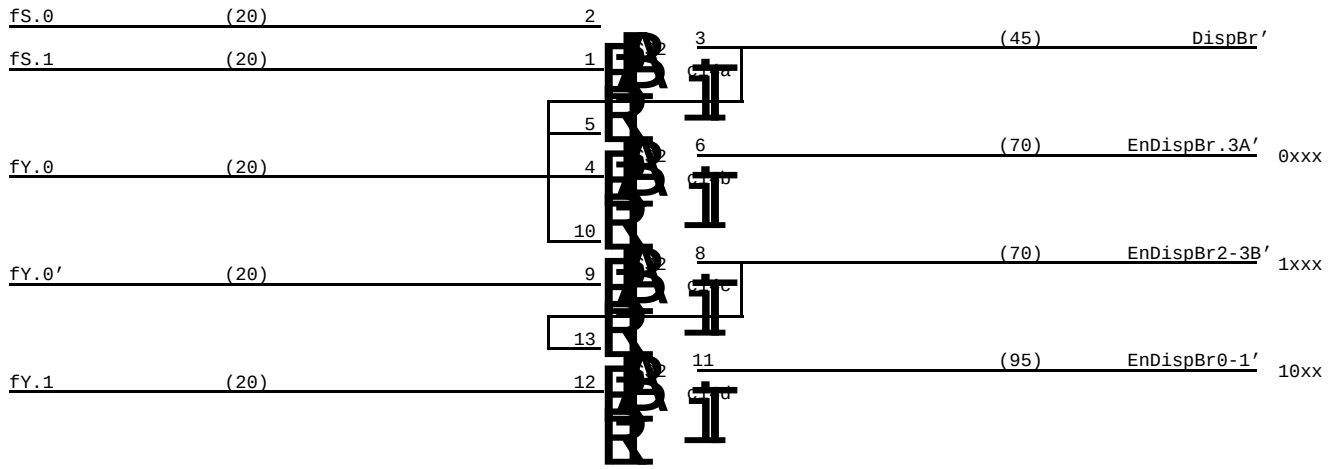
IOOut - High



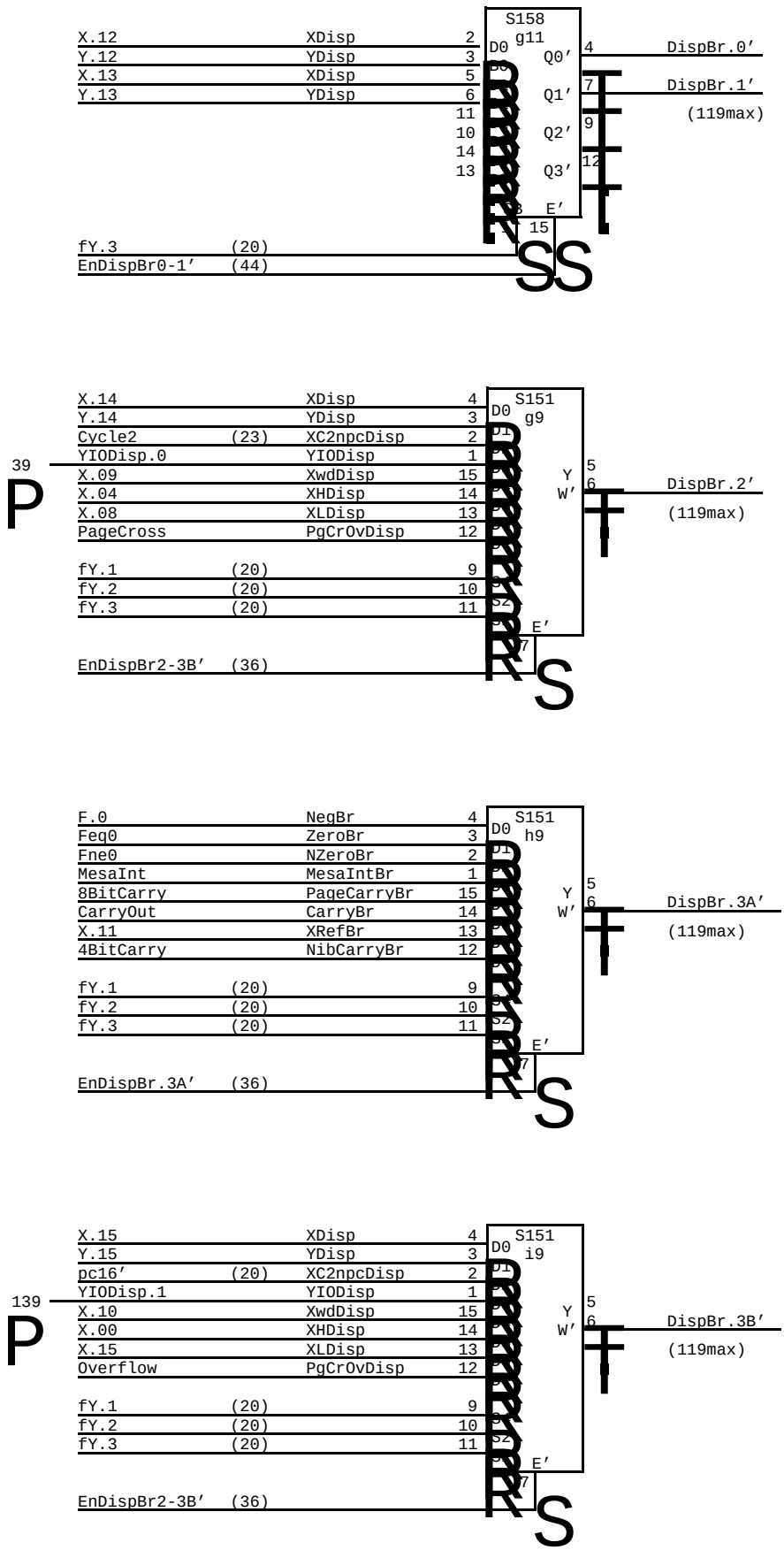
XEROX PARC-CSL	Project CPE	Microinstruction Decoding I	File sCPE12.sil	Designer TonyWest.PA	Rev W	Date 1/31/83	Page 12



EnDispBr



XEROX PARC-CSL	Project CPE	Microinstruction Decoding II	File sCPE13.sil	Designer Garner.PA	Rev W	Date 1/14/83	Page 13



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
```

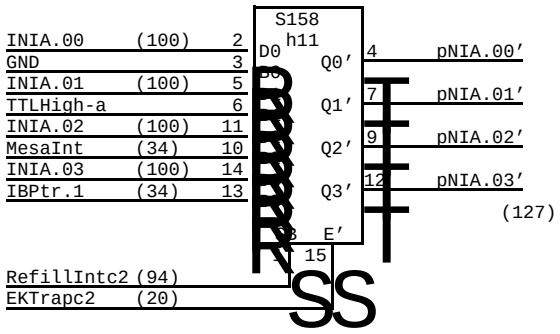
DispBr Setup

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

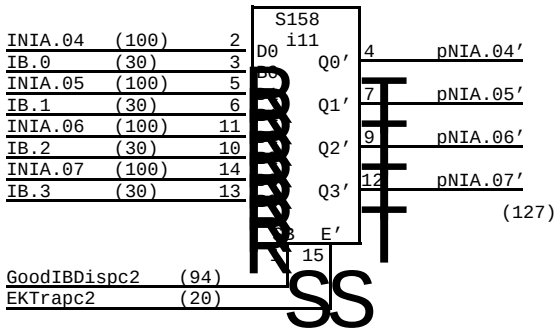
DispBr[2-3]=max(c+26,55,103)

```
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[0-3]



pNIA[4-7]

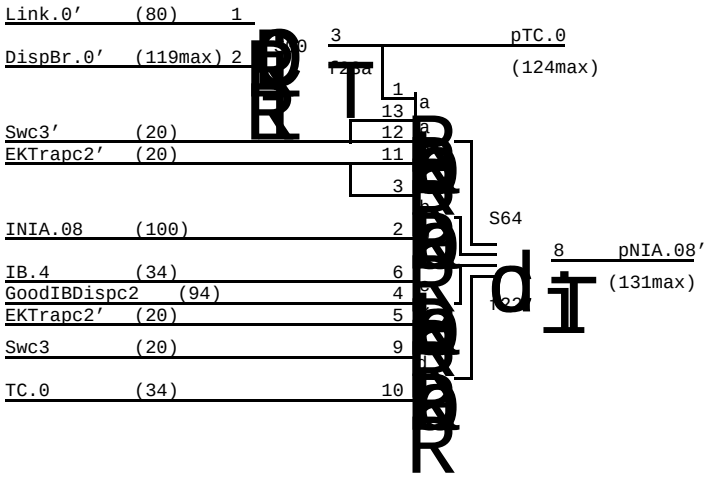


pNIA[0-7]=max(127, 120, 46) nS

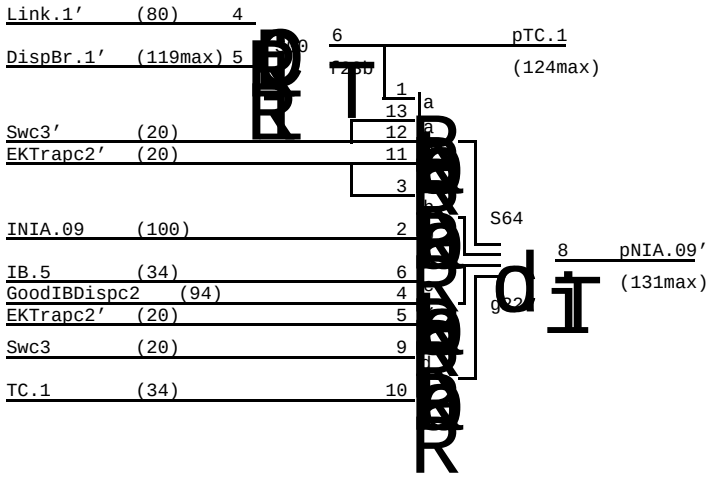
94	^ to RefillIntc2	100	^ to INIA	20	^ to EKErrc2
24[3]	LS158 SB to pNIA′	12[2]	LS158 data to pNIA′	18[2]	LS158 E′ to pNIA′
5[1]	25S09/S374 setup	5[1]	25S09/S374 setup	5[1]	25S09/S374 setup
123[4]=127 nS		117[3]=120 nS		43[3]=46nS	

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

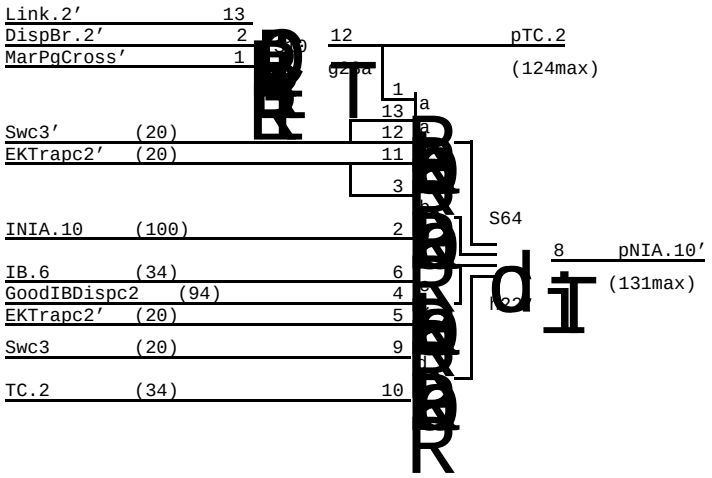
pNIA.8



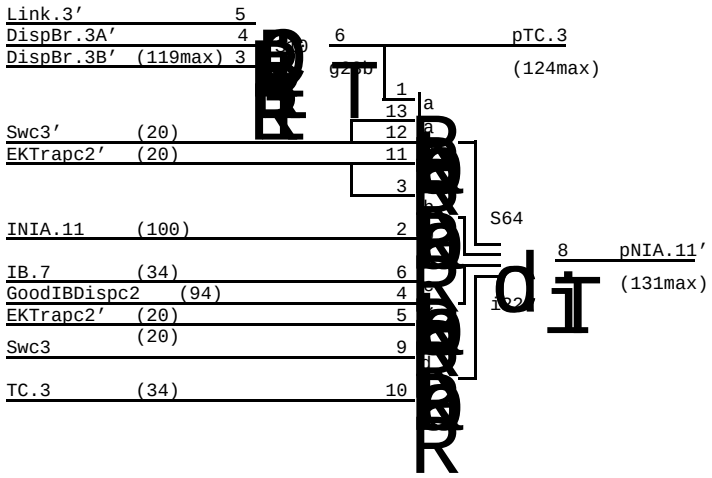
pNIA.9



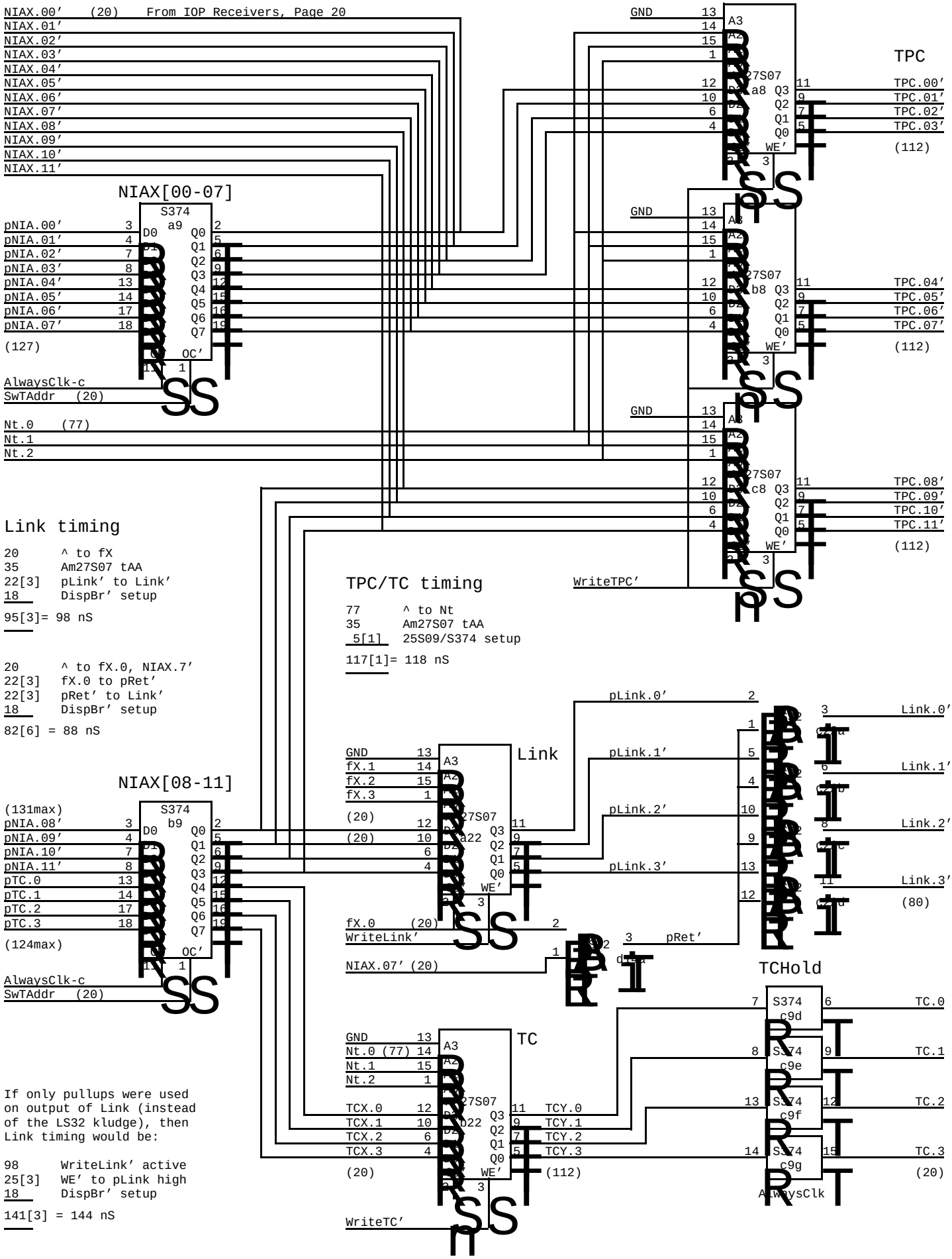
pNIA.10



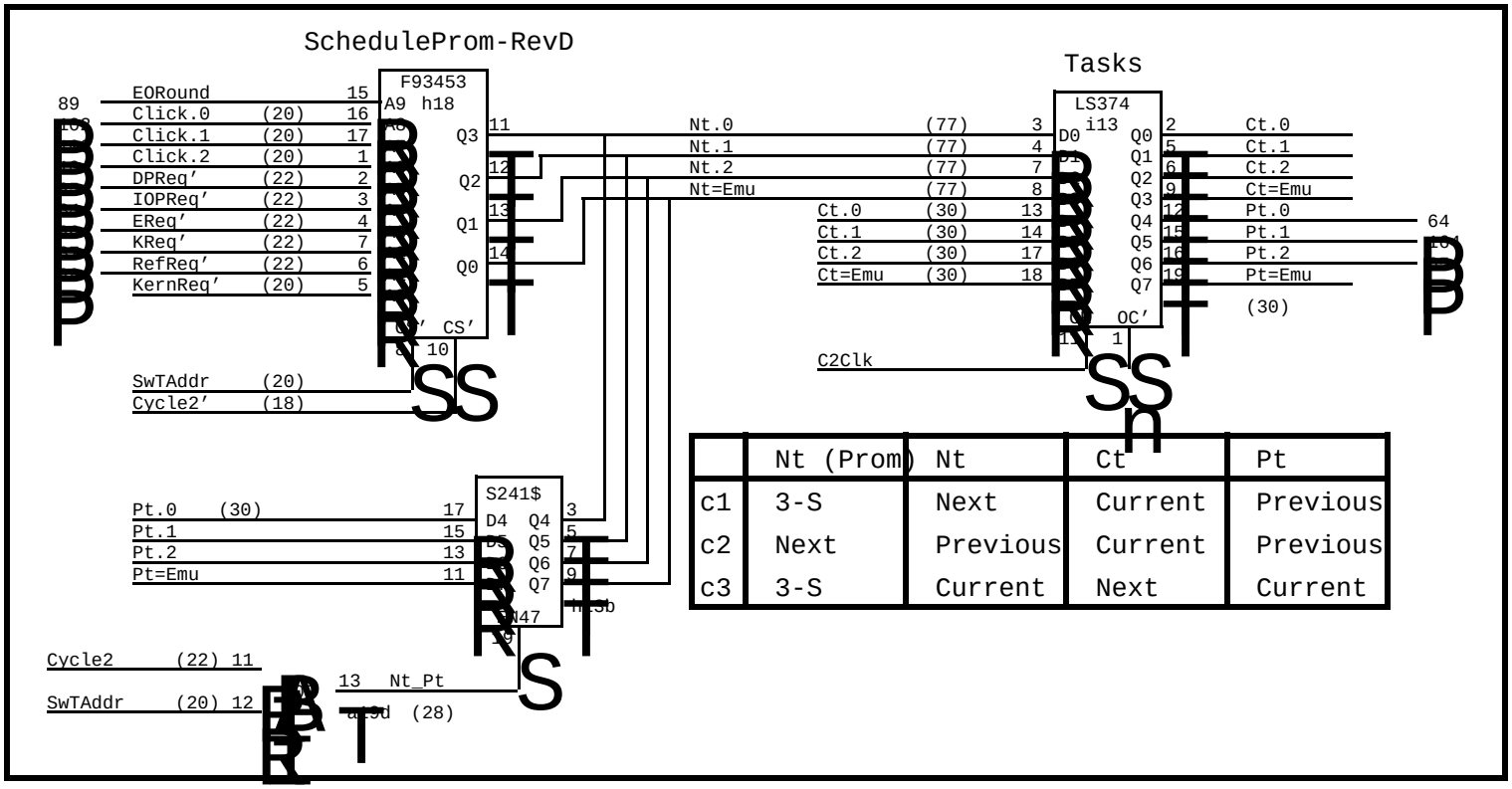
pNIA.11



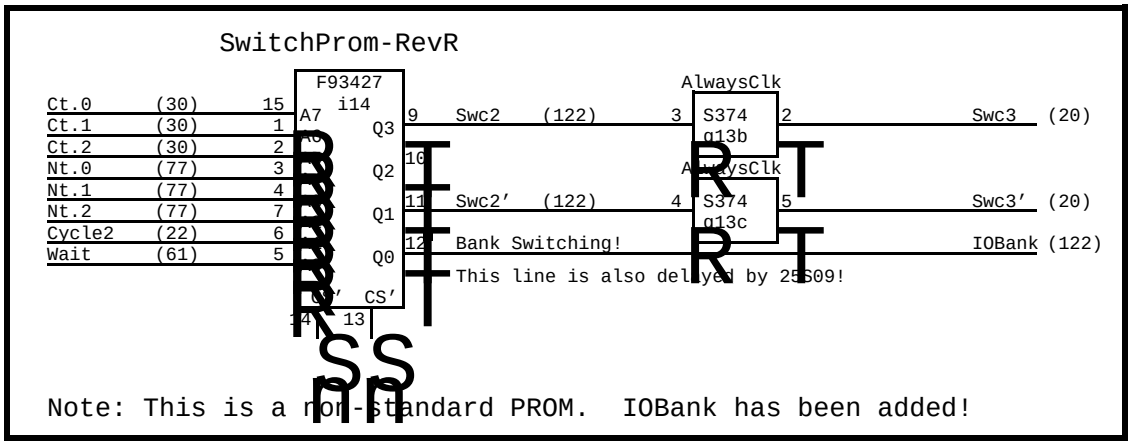
XEROX PARC-CSL	Project		File	Designer	Rev	Date	Page
	CPE	pNIA, pTC (Branching)	sCPE15.sil	Garner.PA	W	1/14/83	15



XEROX PARC-CSL	Project CPE	TPC, TC, & Link	File sCPE16.sil	Designer Garner.PA	Rev W	Date 1/24/83	Page 16



	Nt (Prom)	Nt	Ct	Pt
c1	3-S	Next	Current	Previous
c2	Next	Previous	Current	Previous
c3	3-S	Current	Next	Current



Task Numbers

0	Emulator
1	Display/LSEP
2	Ethernet
3	Refresh
4	Disk
5	IOP
6	Control Store R/W
7	Kernel

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
10[1]	25S09 SB setup	10[1]	25S09 SB setup	10[1]	25S09 SB setup
132[1]=133 nS		100[1]=101 nS		98[3]=101 nS	

Click Assignment

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

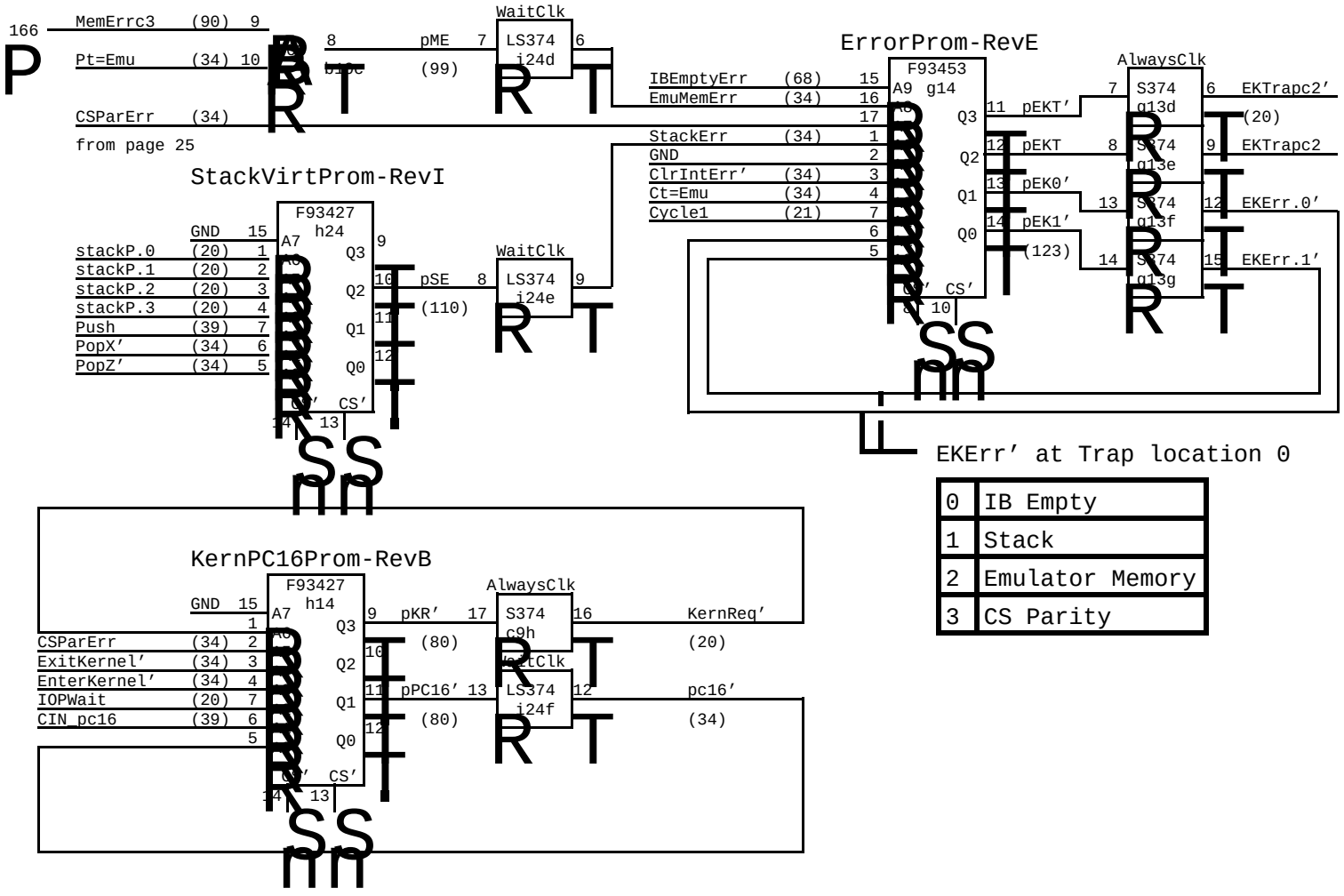
Notes:

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 roun

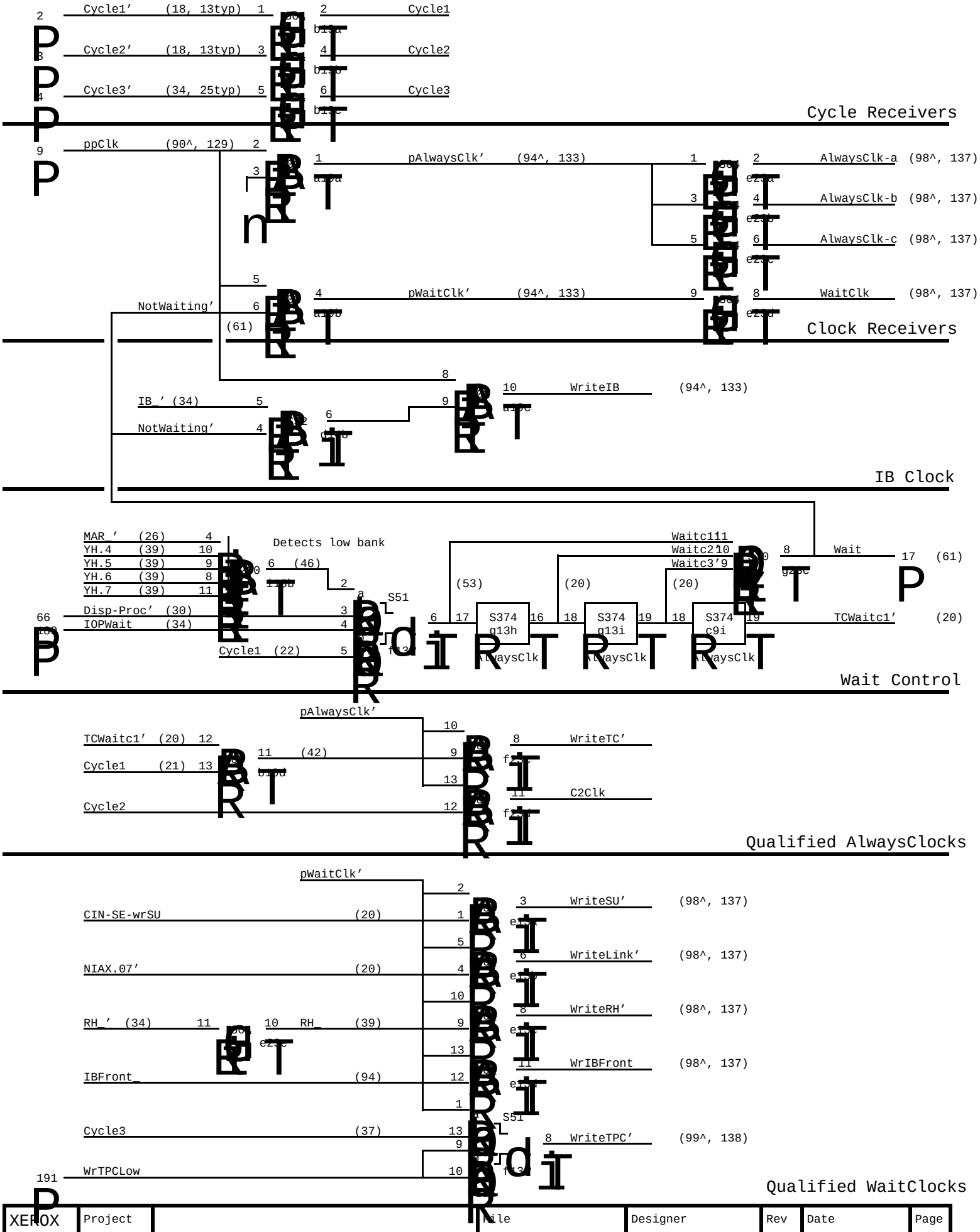
The Display & LSEP-refresh tasks never both use Click 4

Warning: This drawing contains font 4 macros!

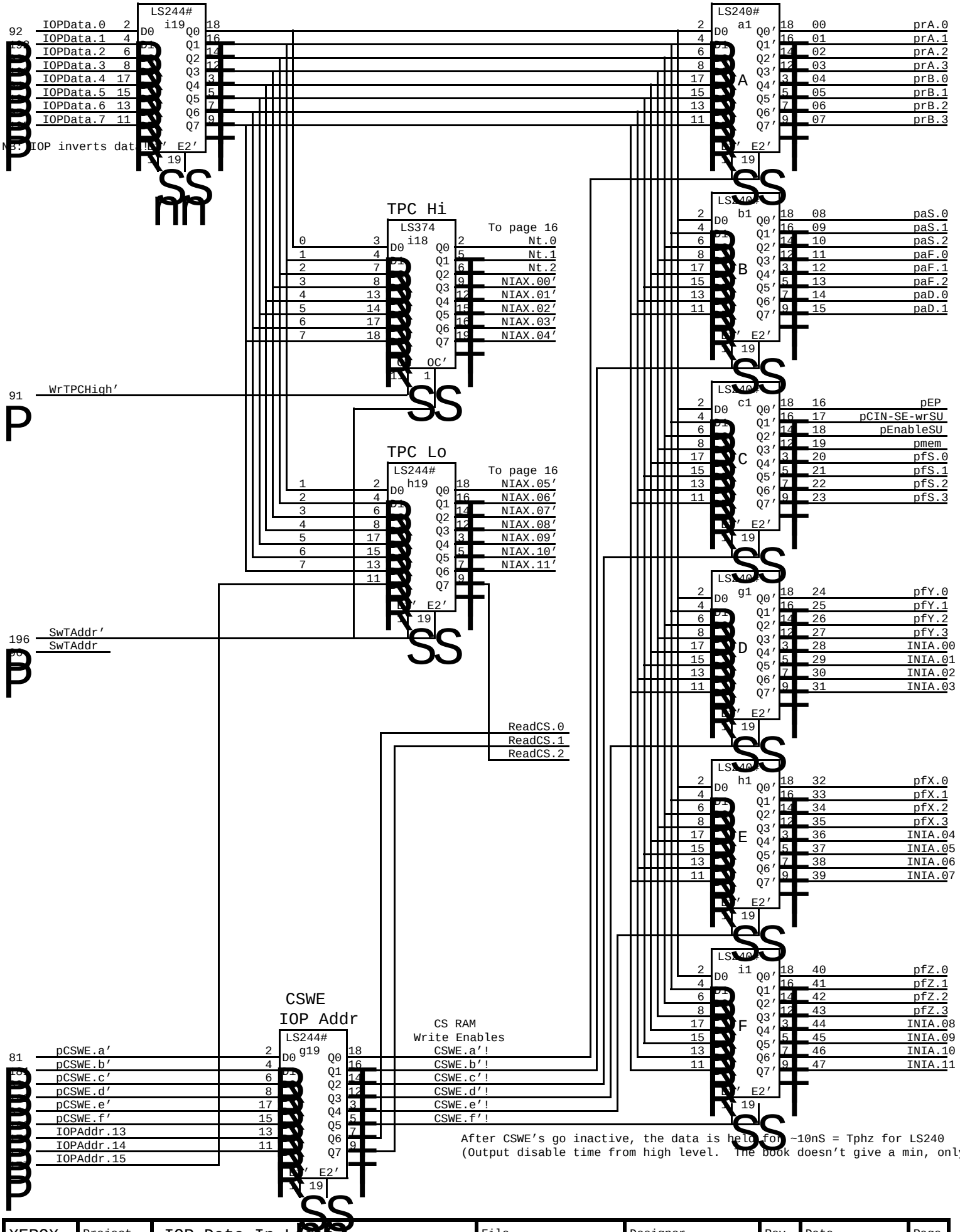
XEROX PARC-CSL	Project CPE	Schedule, Switch, & Tasks	File sCPE17.sil	Designer Garner.PA	Rev W	Date 1/24/83	Page 17



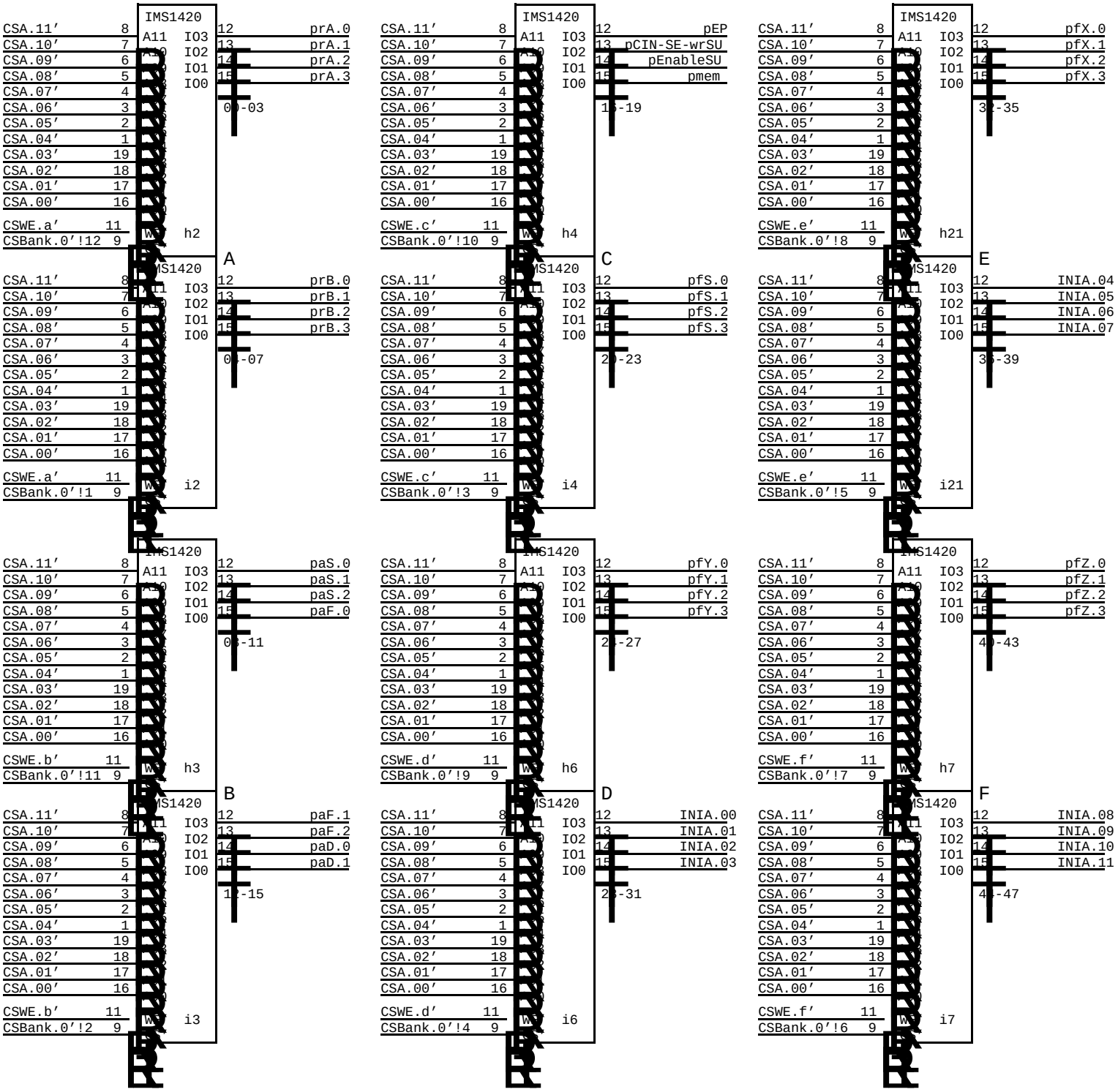
XEROX PARC-CSL	Project CPE	Error, Emulator, & Kernel Proms	File sCPE18.sil	Designer Garner.PA	Rev W	Date 1/14/83	Page 18



XEROX PARC-CSL	Project		File	Designer	Rev	Date	Page
	CPE	Clocks, Wait	sCPE19.sil	Garner.PA	W	1/14/83	19



XEROX PARC-CSL	Project	IOP Data In Logic Control Store Write Logic	File	Designer	Rev	Date	Page
	CPE		sCPE20.sil	TonyWest.PA	W	1/24/83	20

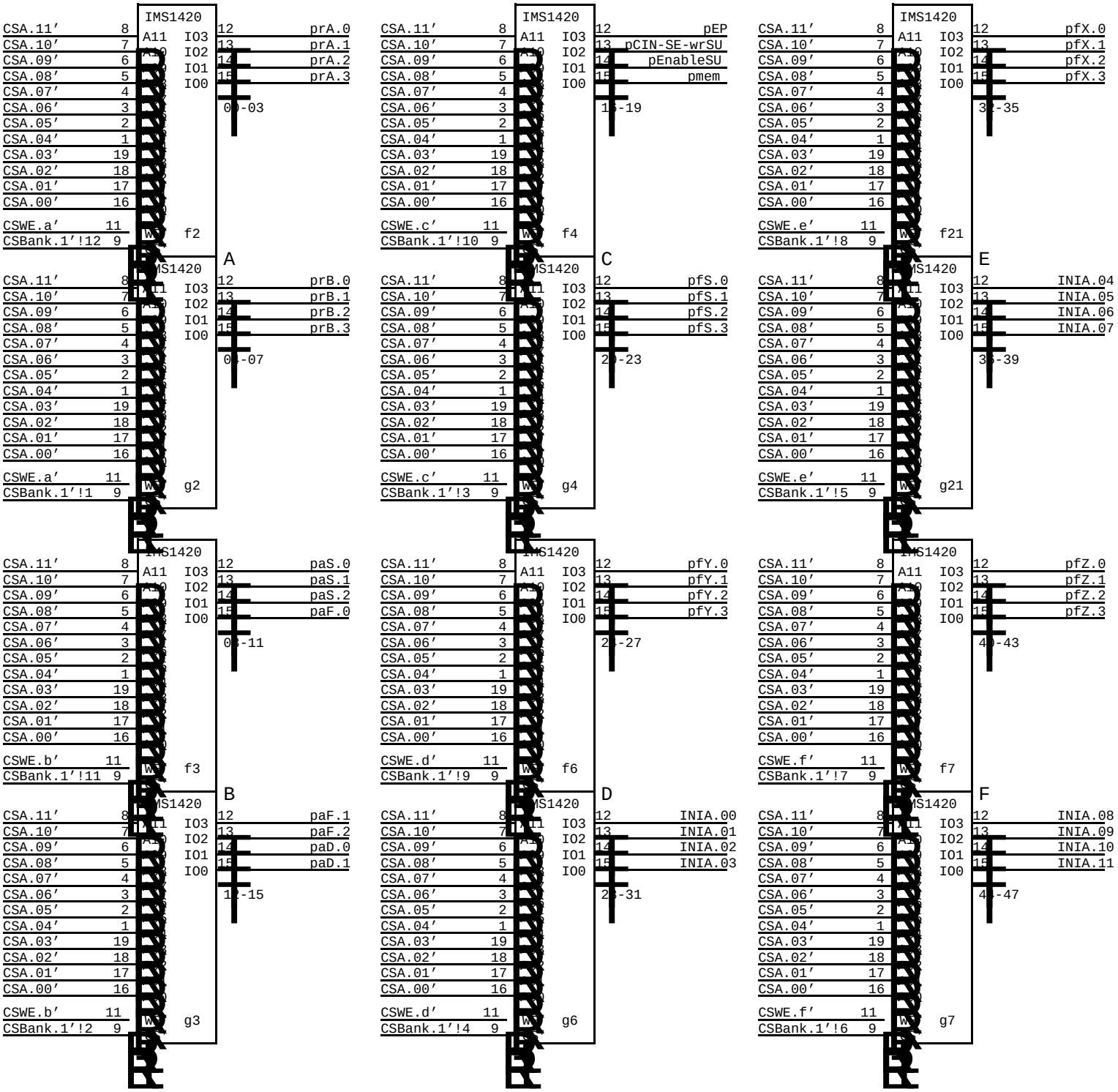


READ		WRITE - Data Hold	
Clock to CSA′ valid	17		
Transmission Delay	13	tPLH for LS240	12
tAA for IMS 1420-55	50	tPZ for LS244	10
CS Data valid at	80		22

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

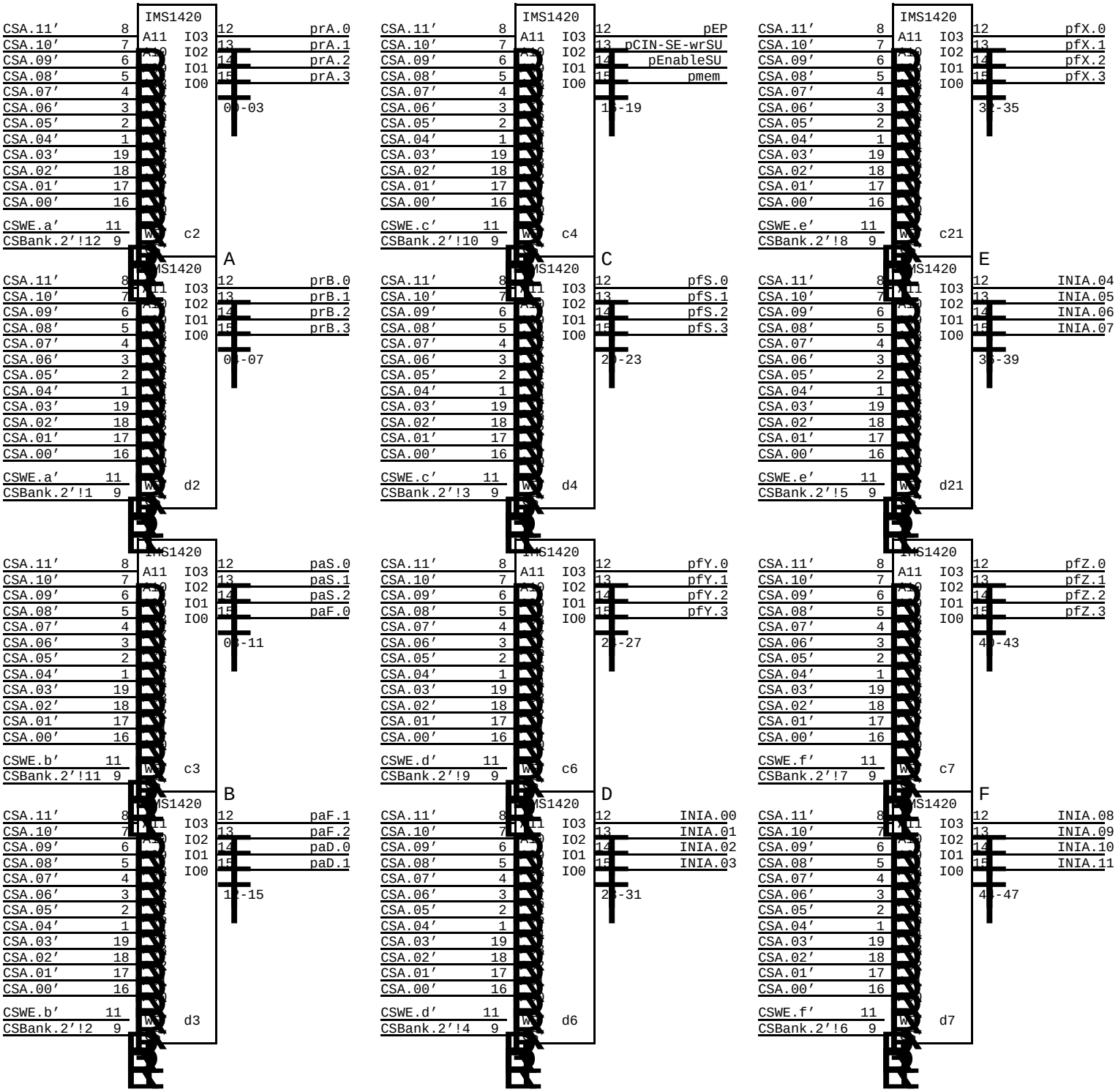
Warning: This drawing contains font 4 macros!

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



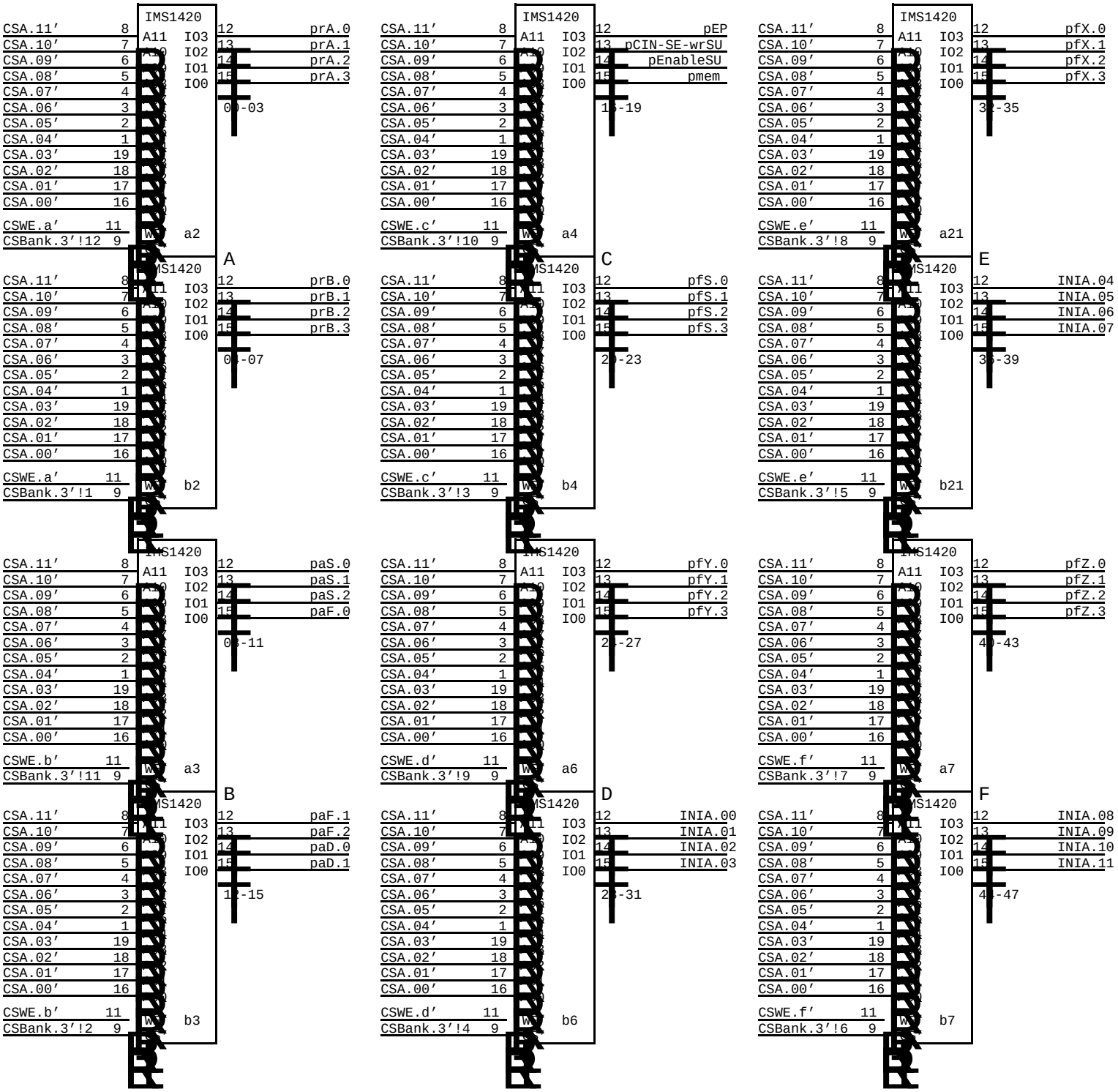
Warning: This drawing contains font 4 macros!

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



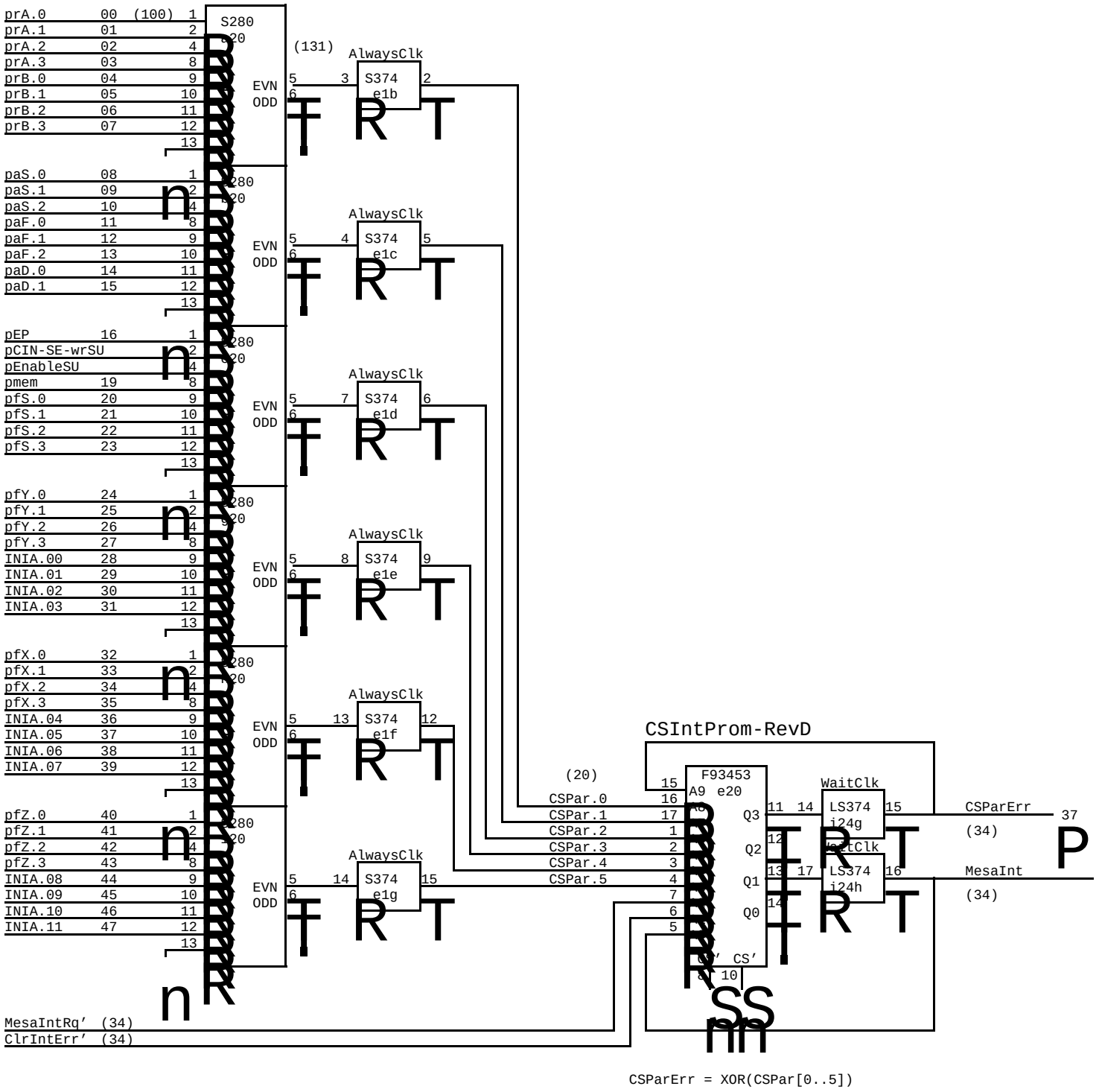
Warning: This drawing contains font 4 macros!

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



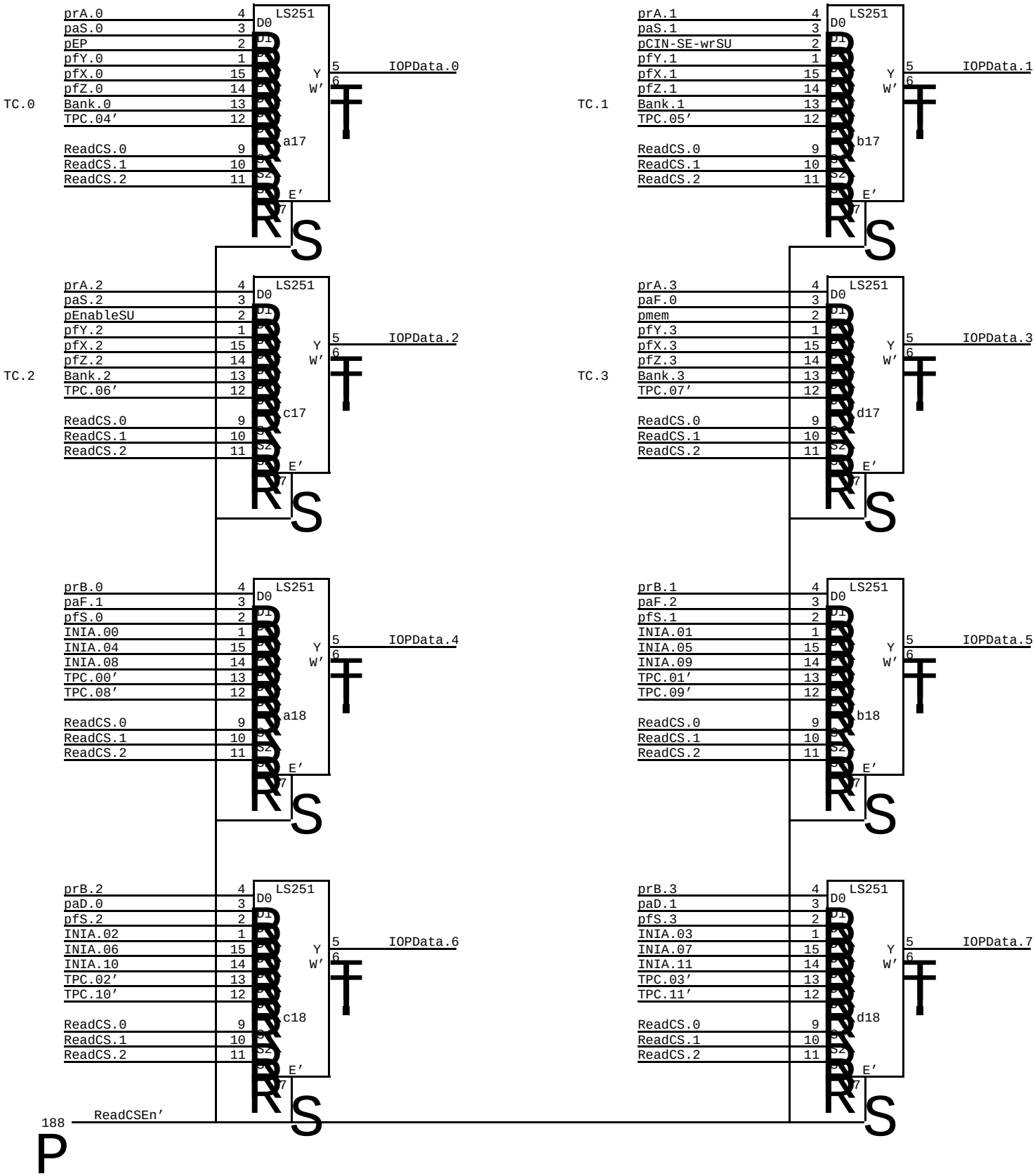
Warning: This drawing contains font 4 macros!

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

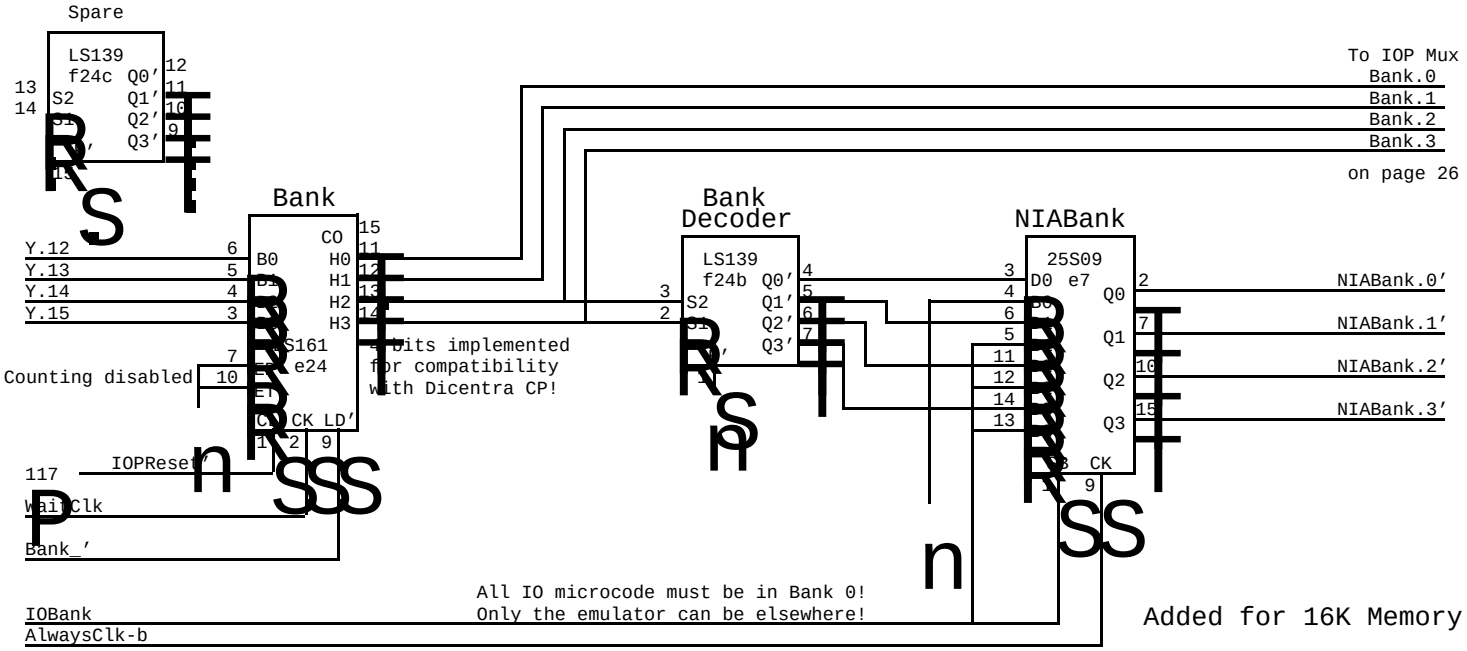


XEROX PARC-CSL	Project CPE	Control Store Parity Checker	File sCPE25.sil	Designer Garner.PA	Rev W	Date 1/14/83	Page 25

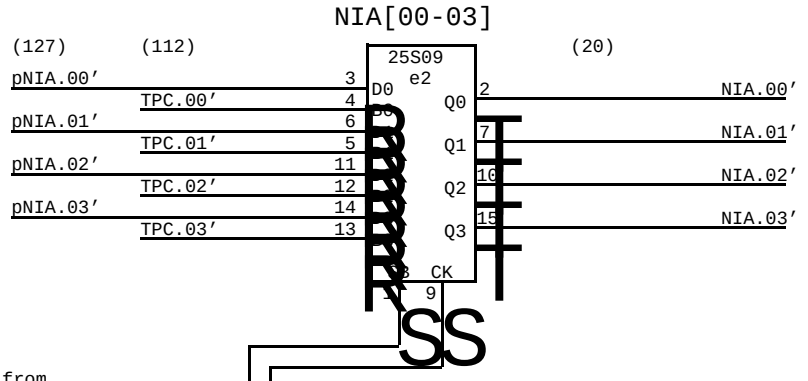
NB: TC[0-3] have been replaced by Bank[0-3].



XEROX PARC-CSL	Project		File	Designer	Rev	Date	Page
	CPE	IOP Data Out from CP	sCPE26.sil	Garner.PA	W	1/14/83	26



This section is standard



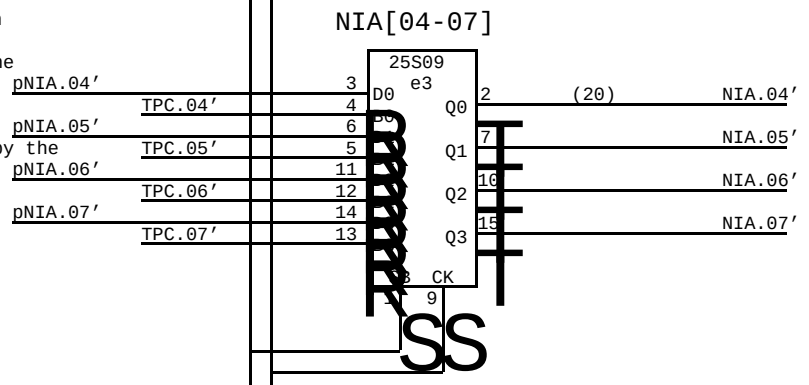
NOTE on Control Store Addresses

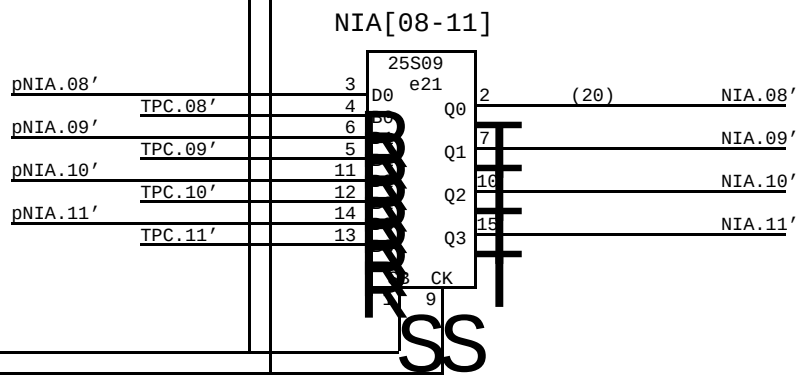
The next instruction address for the control store comes from one of two basic places:

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

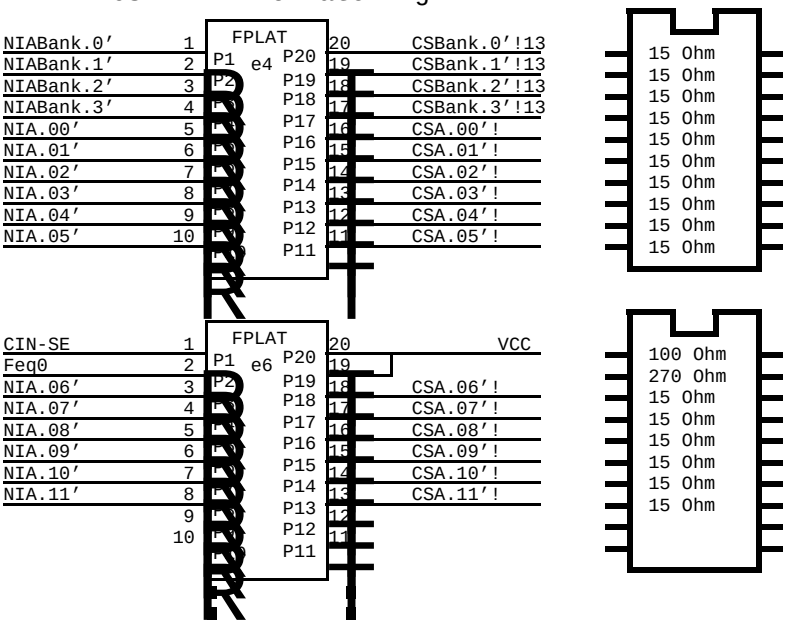
In the case of (1), task 6's TPC registers are used by the IOP to provide the address when the IOP wants to read or write data into the control store.

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

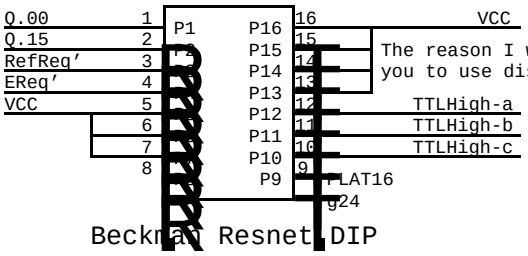




CS NIA Line Matching



1 KOhm Pullups



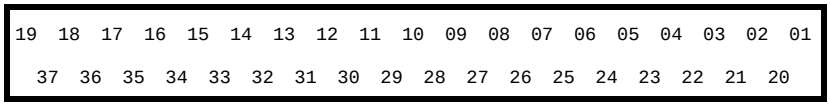
The reason I wired it like this is to permit you to use discrete resistors if you want.

Beckman Resnet DIP
898-1-1K

XEROX PARC-CSL	Project CPE	Resistors	File sCPE28.sil	Designer TonyWest.PA	Rev W	Date 1/24/83	Page 28

51	GND	19	
52	EKErr.1′	18	
53	EKErr.0′	17	
54	ReadCSEn′	16	The read strobe for the IOP multiplexor
55	CSWE.a′	15	One of the CS byte write strobes
56	SwTAddr	14	
57	WrTPCHigh′	13	
58	DesError	12	This signal indicates that one or both of the DES FSM’s detected an error. Once set, it rem
59	IOPWait	11	
60	Swc2	10	If true, this means there will be a task switch next click
61	IOBank	09	Coerces CS References to come from Bank 0 on next clock
62	Bank_′	08	Bank register load enable
63	Cycle3	07	
64	Cycle2	06	
65	Cycle1	05	
66	Wait	04	Stops WaitClk for a whole number of clicks when display active, IOP in control, or Burdock r
67	GoodIBDispc2	03	Says when you are about to dispatch on a Mesa Opcode Byte.
68	WaitClk	02	
69	VCC	01	
70	Nt.2	37	In c1, Nt = Next Task Number In c2, Nt = Previous Task Number In c3, Nt = Current Task Number
71	Nt.1	36	
72	Nt.0	35	
73	AlwaysClk-b	34	
74	Bank.3	33	Only the bottom two bits are needed
75	Bank.2	32	Only the bottom two bits are needed
76	CSA.11′	31	
77	CSA.10′	30	
78	CSA.09′	29	
79	CSA.08′	28	
80	CSA.07′	27	
81	CSA.06′	26	
82	CSA.05′	25	
83	CSA.04′	24	
84	CSA.03′	23	
85	CSA.02′	22	
86	CSA.01′	21	
87	CSA.00′	20	

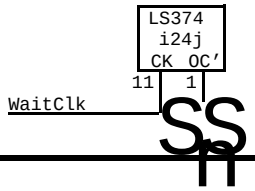
Bottom Connector
Female - End View



XEROX PARC-CSL	Project	Cannon DB37 Female Logic Analyzer Connector	File	Designer	Rev	Date	Page
	CPE		sCPE29.sil	TonyWest.PA	W	1/24/83	29

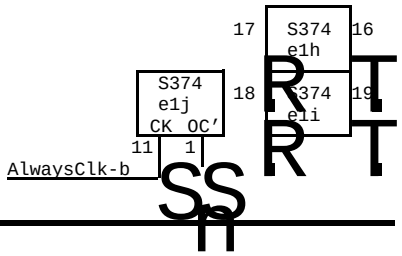
LS374 i24

b	IBPtr.0	08
c	IBPtr.1	08
d	EmuMemErr	18
e	StackErr	18
f	pc16’	18
g	CSParErr	25
h	MesaInt	25
i	SFLG’	33



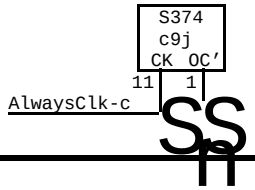
S374 e1

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		
i		



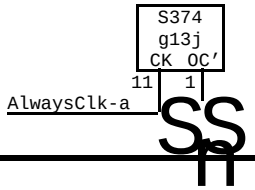
S374 c9

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



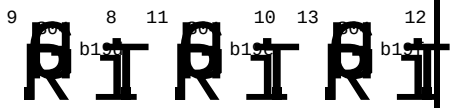
S374 g13

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



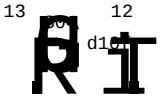
S04 b19

a	Cycle1	19
b	Cycle2	19
c	Cycle3	19
d		
e		
f		



S04 d10

a	aD.0’	05
b	MAR_	11
c	IBEmptyErr’	11
d	Des_YBus	31
e	XBus_SU	06
f		

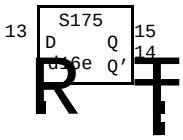


S04 e23

a	AlwaysClk-a	19
b	AlwaysClk-b	19
c	AlwaysClk-c	19
d	WaitClk	19
e	RH_	19
f	Fne0	04

S175 d16

b	MAS’	33
c	MDS’	33
d	SDS’	33
e		30
f	*anon*	33



S00 a10

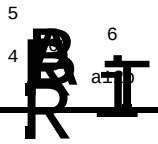
a	XBus_SU’	06
b	pMAR_’	11
c	MarPgCross’	11
d	CIN_pc16	13

S00 b14

a	Pop	13
b	Nibble’	13
c	Byte’	13
d	Xhigh_0	13

S00 a12

a	*anon*	31
b		
c	*anon*	31
d	DesClkDisable	32



S00 e13

a	WriteSU’	19
b	WriteLink’	19
c	WriteRH’	19
d	WrIBFront	19

S00 f23

a	pTC.0	15
b	pTC.1	15
c	WriteTC’	19
d	C2Clk	19

S02 a19

a	pAlwaysCLK’	19
b	pWaitCLK’	19
c	WriteIB	19
d	Nt_Pt	17

S08 b10

a	paSh.0	11
b	pAllowMDR_	11
c	pME	18
d	*anon*	19

S10 a15

a	sh	13
b	Push	13
c	XByte’	13

S10 g23

a	pTC.2	15
b	pTC.3	15
c	Wait	19

S20

a	XBus_IB’	13
b	EnLRotn’	13

LS32 c14

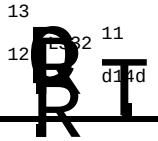
a	DispBr’	13
b	EnDispBr.3A’	13
c	EnDispBr2-3B’	13
d	EnDispBr0-1’	13

LS32 c22

a	Link.0’	16
b	Link.1’	16
c	Link.2’	16
d	Link.3’	16

LS32 d14

a	pRet’	16
b	*anon*	19
c	DesError	33
d		



S38 i10

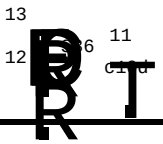
a	Q.00	05
b	Q.15	05
c	CarryIn	05
d	CarryIn	05

S51 f13

a	Waitc1’	19
b	WriteTPC’	19

S86 c10

a	PageCross	05
b	MapRef	13
c	Refresh	13
d		



S260 i15

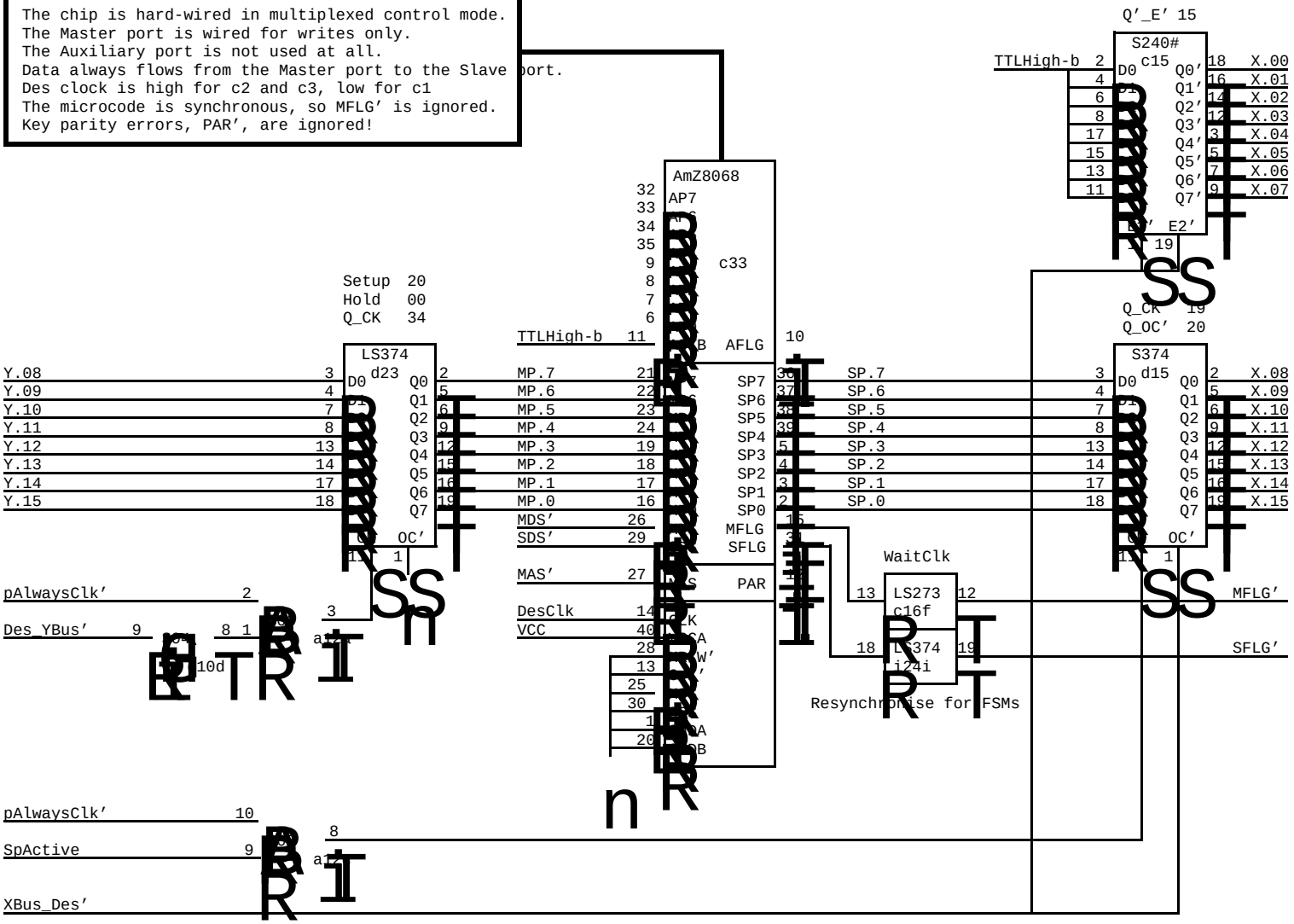
a	IBEmptyErr	08
b	*anon*	19

S175

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



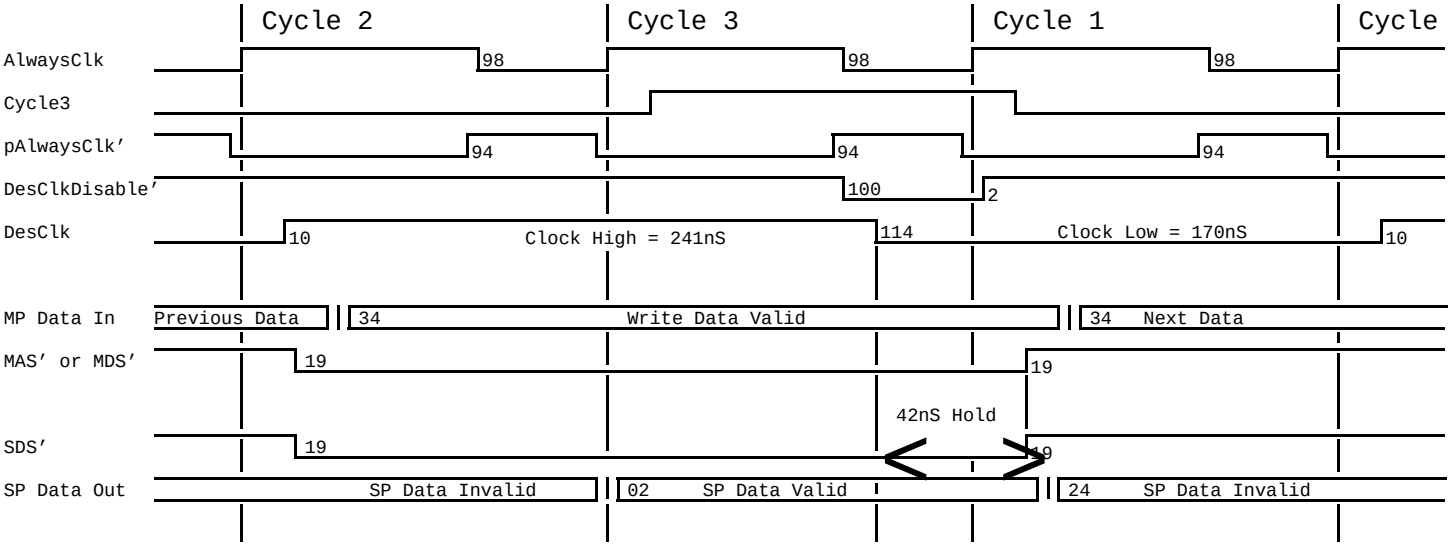
Warning: This drawing contains font 4 macros!

XEROX PARC-CSL	Project		File	Designer	Rev	Date	Page
	CPE	DES Encryption Hardware	sCPE31.sil	TonyWest.PA	W	1/24/83	31

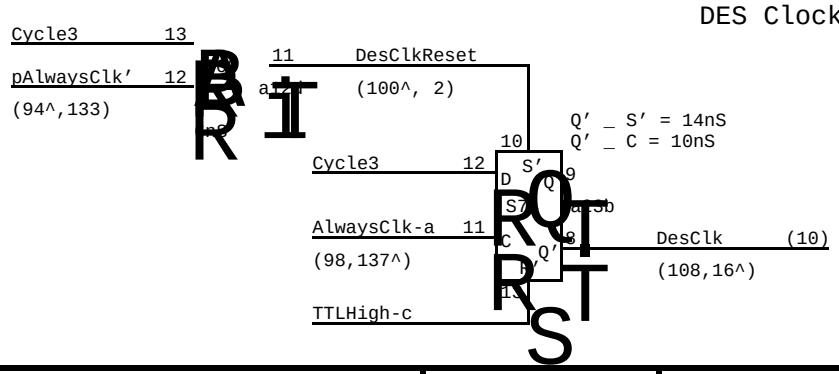
		AMD #	min.	max.	actual used	Notes
Clock & Reset						
Clock width HIGH		1	115		241	
Clock width LOW		2	115		170	
Clock Cycle		3	250		411	
Clock High to MAS'&MDS' High	Reset Hold	6	0	50	19	
MP and SP Strobe Times						
MAS' falling to MAS' rising (address)	MAS width Low	32	80		274	
MDS' falling to MDS' rising (data)	MDS width Low	44a	125	1000	274	Can't exceed 1000, so have to watch out for WaitClk
MDS' rising to MDS' falling	MDS Recovery	46	125		137	
SDS' falling to SDS' rising (data read)	SDS width Low	44a	125	1000	274	Can't exceed 1000, so have to watch out for WaitClk
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		
SDS rising to Data Invalid	SP Data Hold	50	5			
SDS falling to SFLG rising	SP Flag	51		125		for last byte read

DES Clock Generator Timing

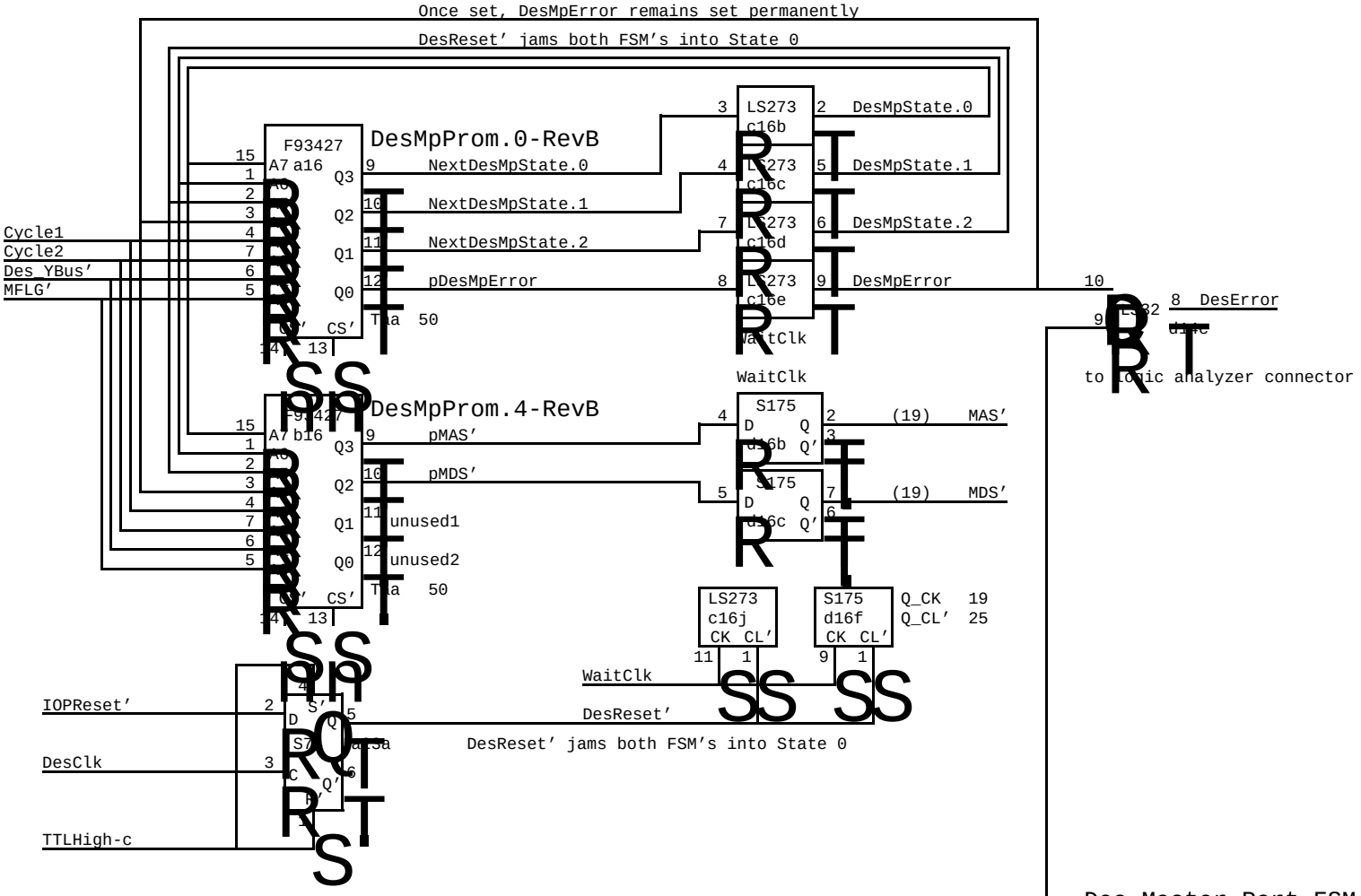
Note: Because of the requirement to hold MDS' and SDS' for 20 to 70 nanoseconds after DesClk falling, we bring DesClk down early in Cy
MAS', MDS' and SDS' follow at the end of Cycle 3.



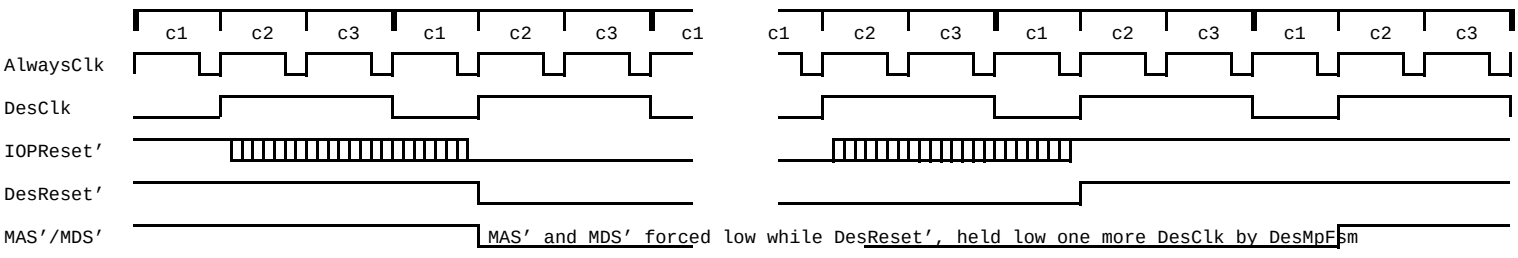
DES Clock Generator



XEROX PARC-CSL	Project	File	Designer	Rev	Date	Page
DES Chip and Clock Timings Des Clock Generator	CPE	sCPE32.sil	TonyWest.PA	W	1/24/83	32



Reset Des Chip and FSM's with IOPReset'

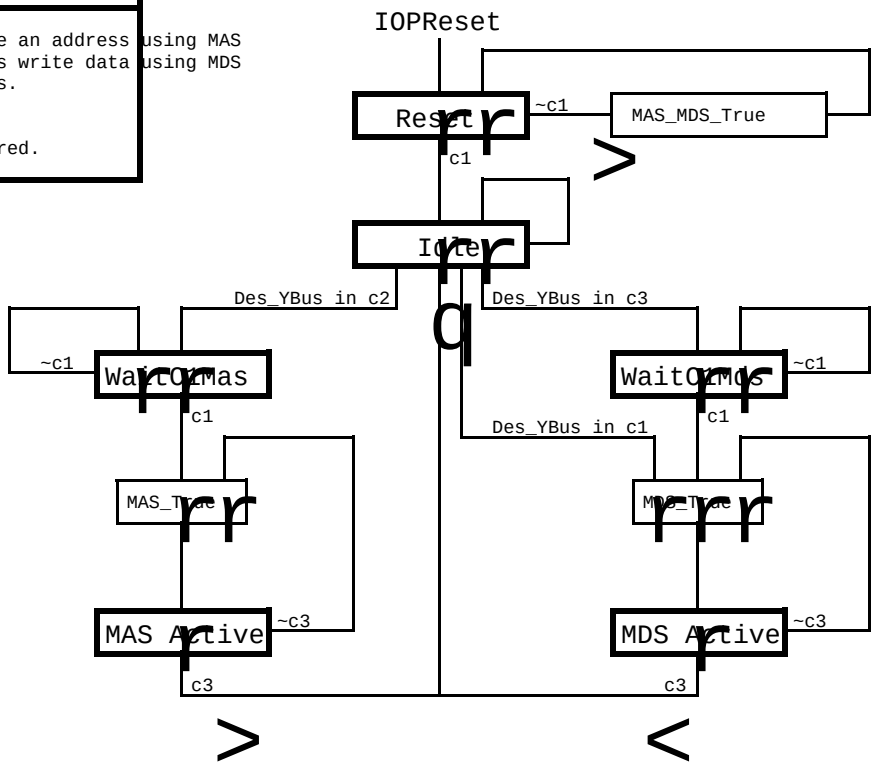


XEROX PARC-CSL	Project CPE	DES Finite-State Machines	File sCPE33.sil	Designer TonyWest.PA	Rev W	Date 1/24/83	Page 33

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.

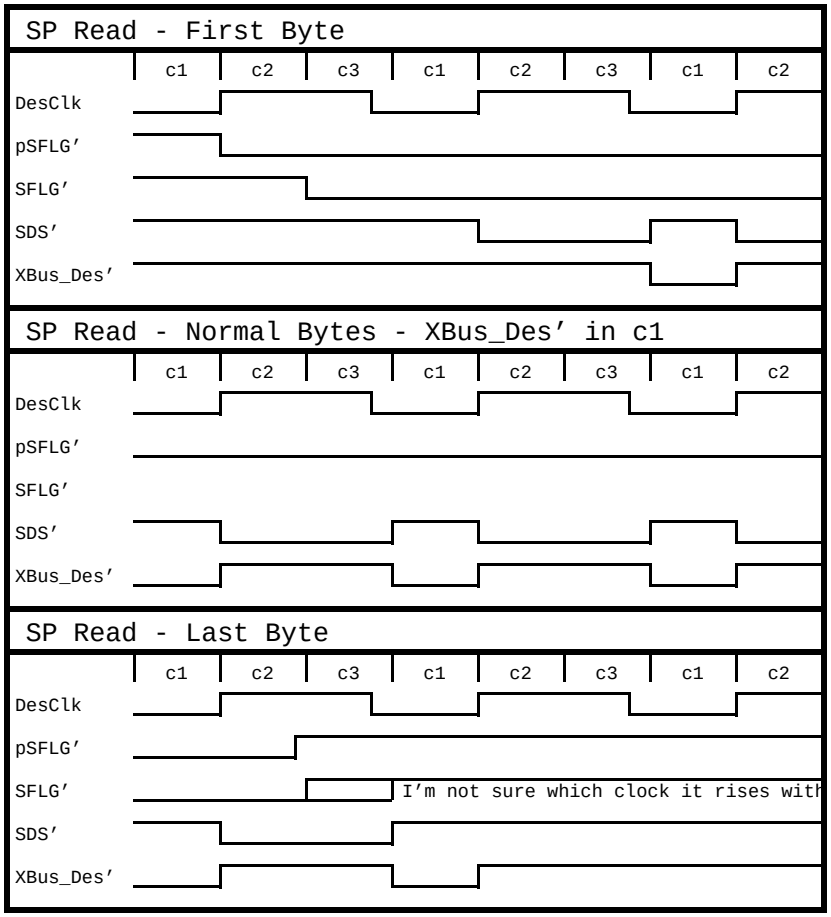
The signals are shown logical-true.
The implementation below inverts signals as required.

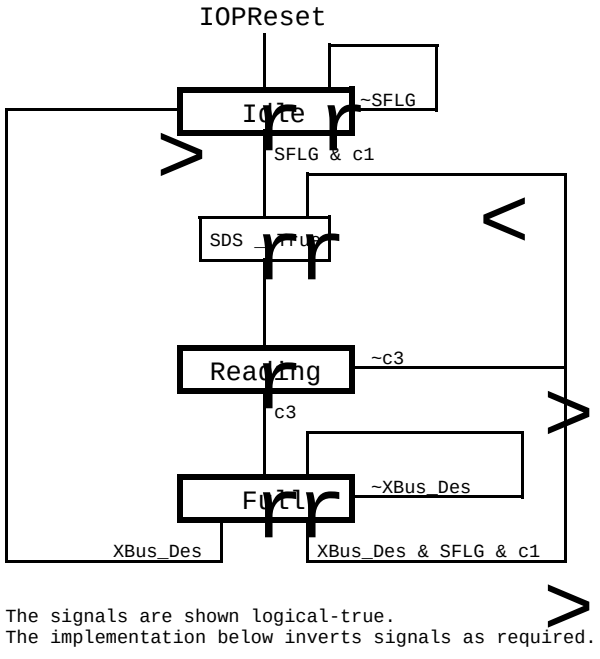


Master Port Finite-State Machine - Error handling of DesMpError signal is not shown

---

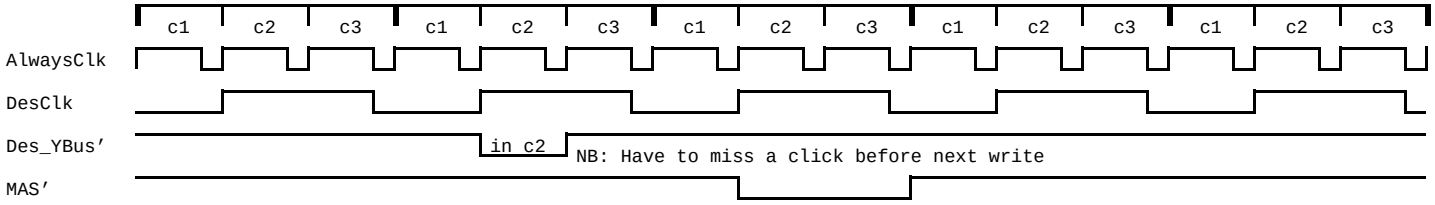
Slave Port Finite-State Machine - Error handling of DesSpError signal is not shown



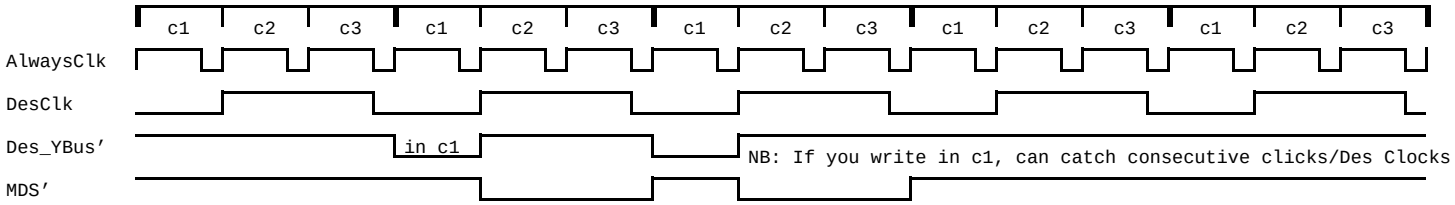


The signals are shown logical-true.
The implementation below inverts signals as required.

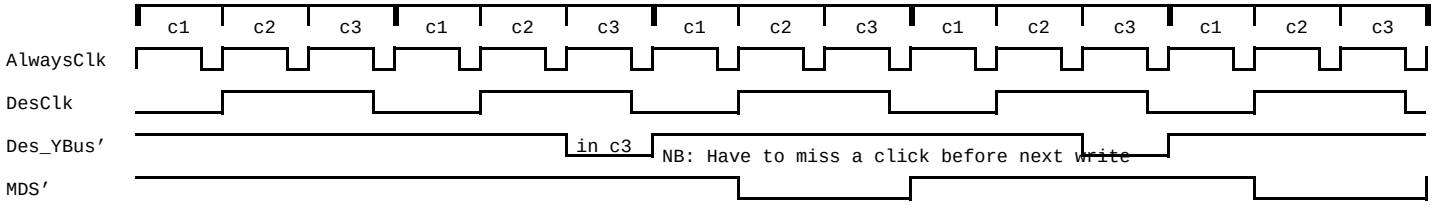
Write address into Des Master Port in C2



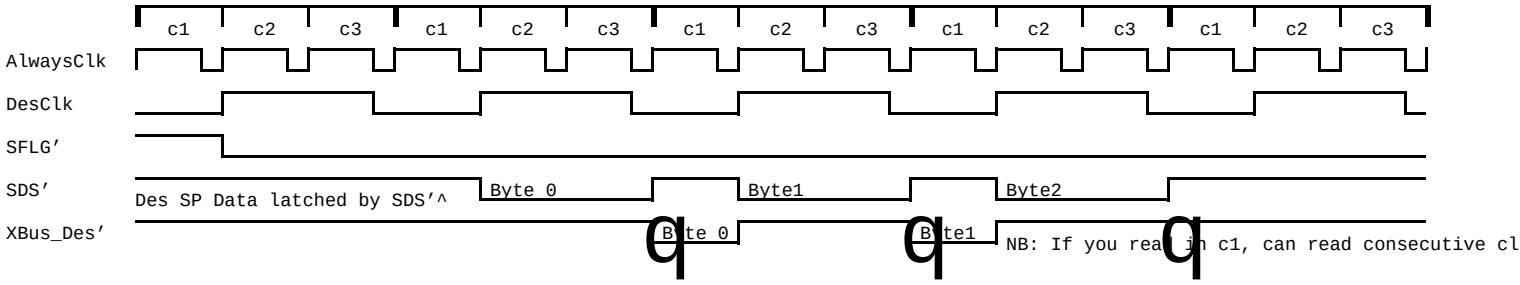
Write data into Des Master Port in C1



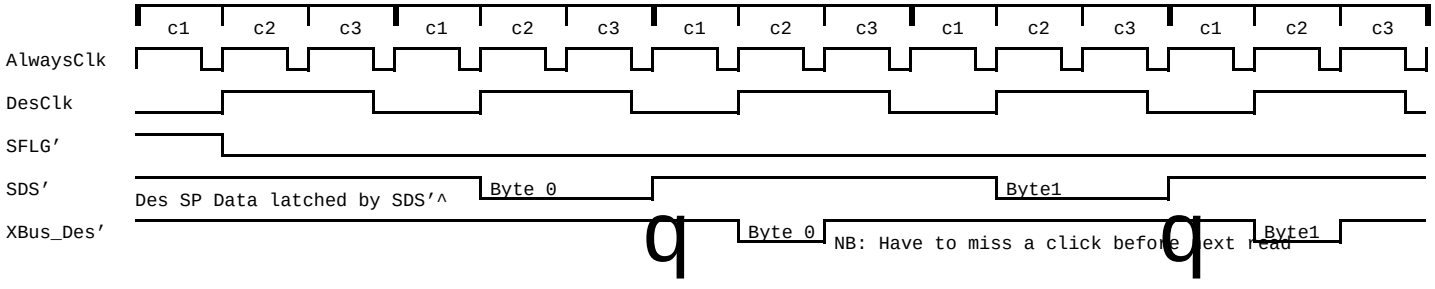
Write data into Des Master Port in C3



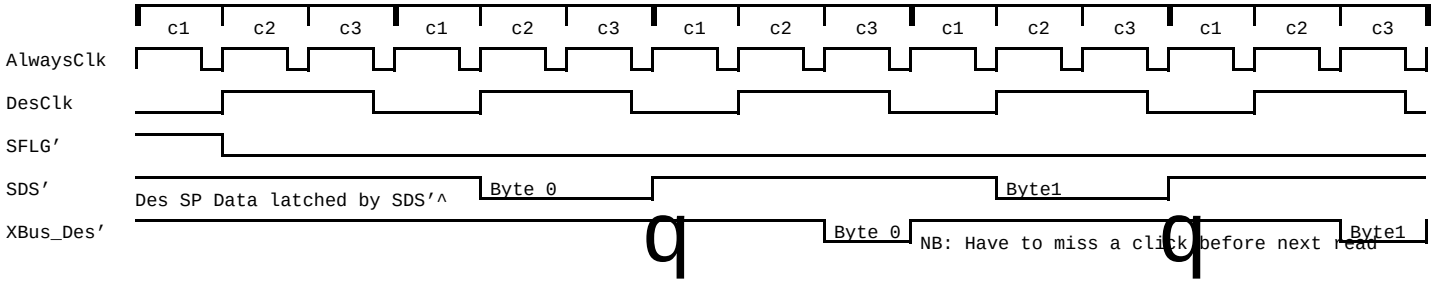
Read Data from Des Slave Port in C1



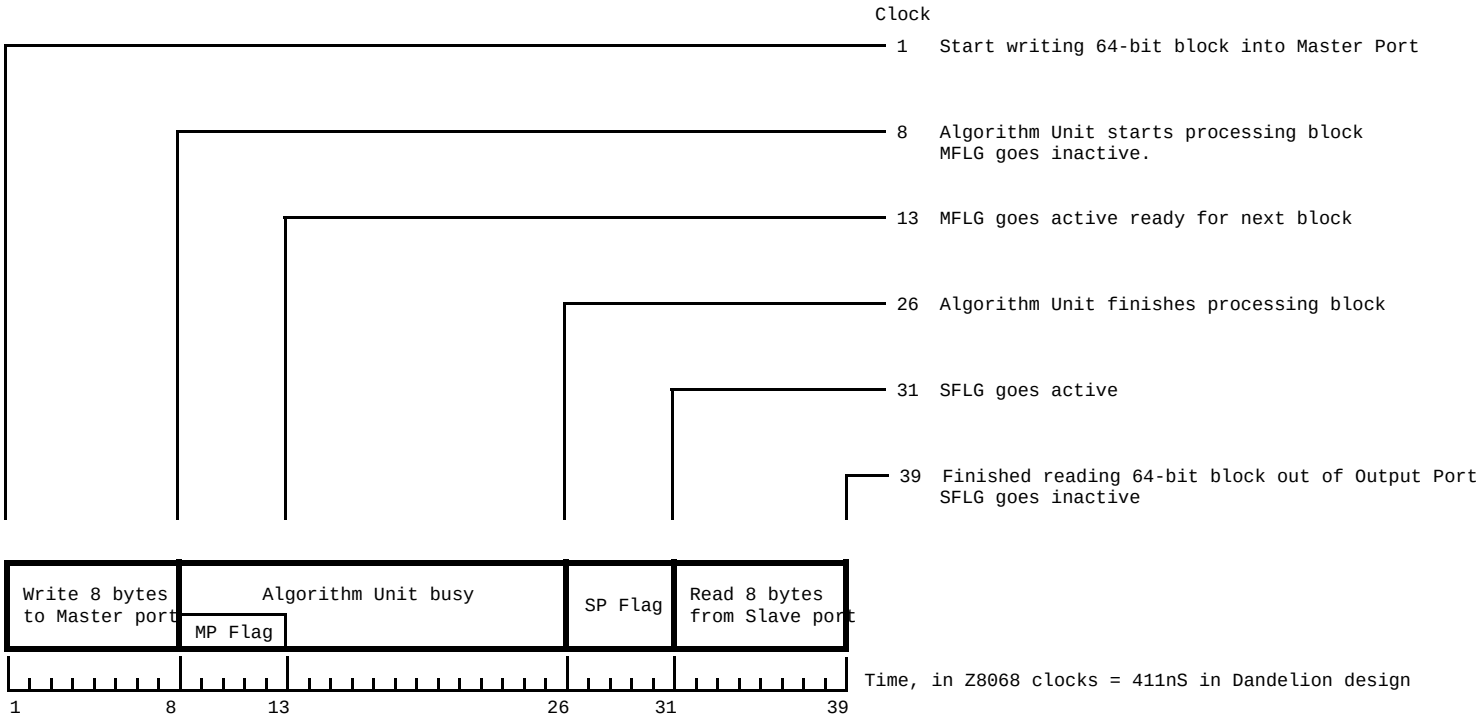
Read Data from Des Slave Port in C2



Read Data from Des Slave Port in C3



XEROX PARC-CSL	Project	Reference	File	Designer	Rev	Date	Page
	CPE	DES Timing Diagrams	sCPE35.sil	TonyWest.PA	W	1/31/83	35



WARNING! This data is not guaranteed to be correct!

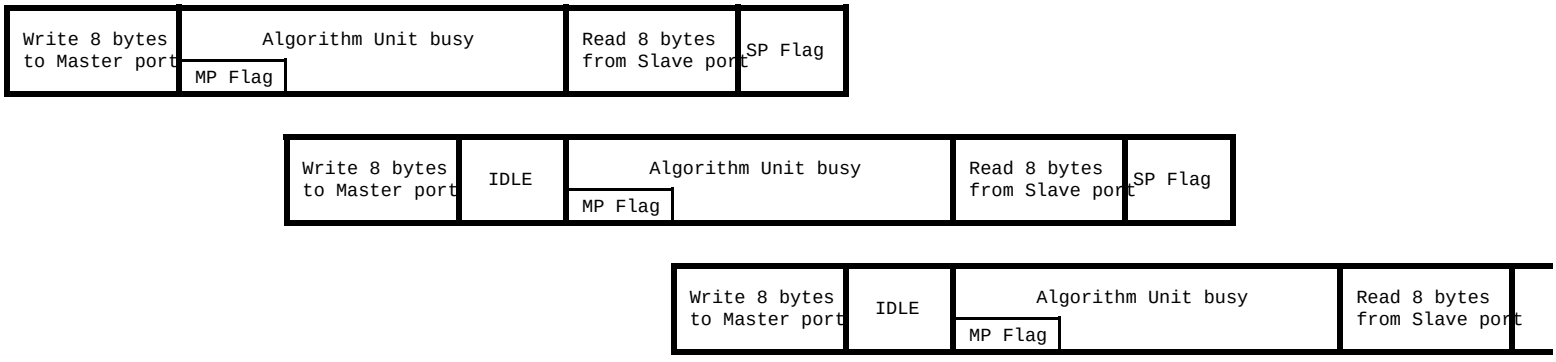
NOTES:

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!