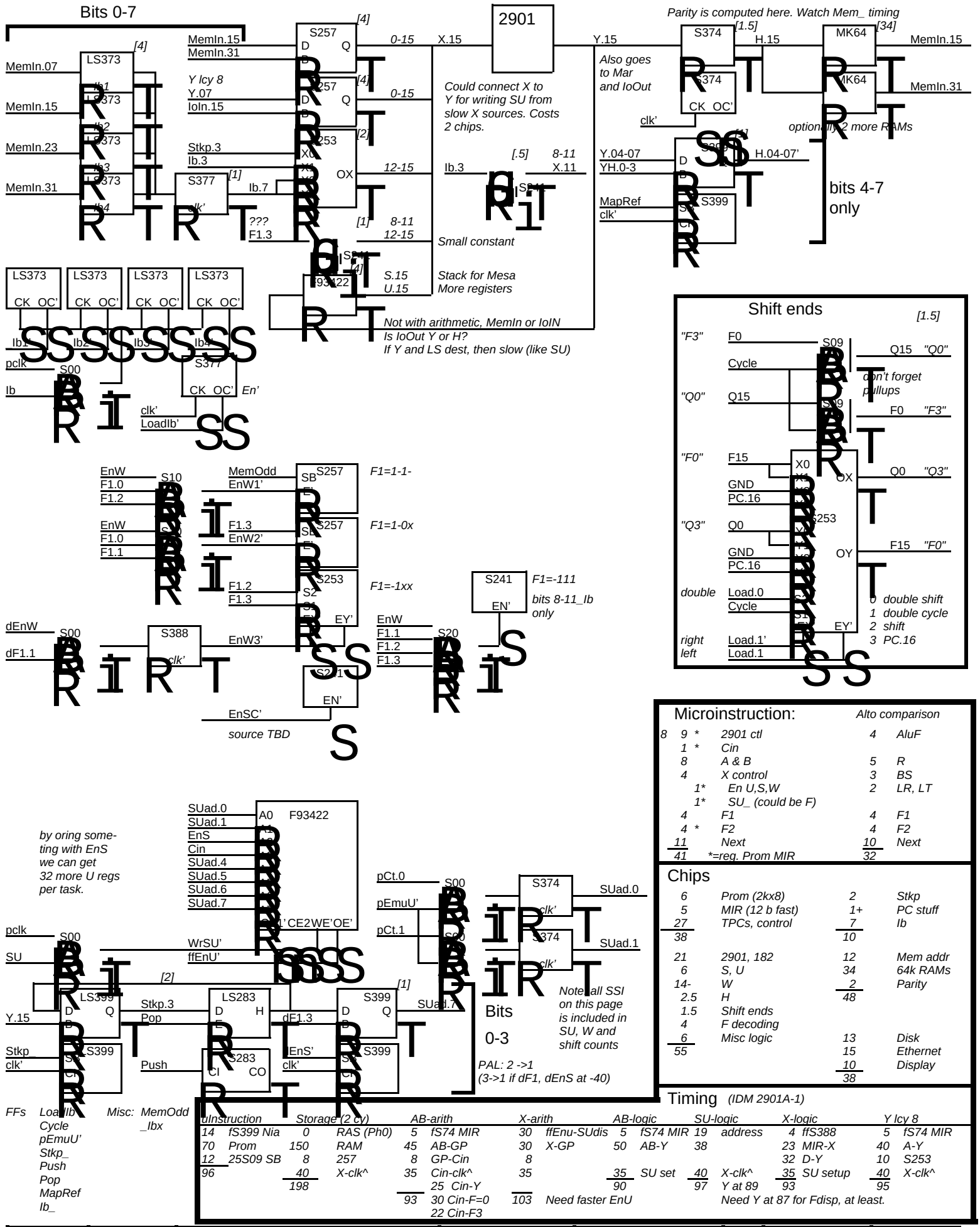


Microinstruction:

				Alto comparison	
8	9	*	2901 ctl	4	AluF
	1	*	Cin		
	8		A & B	5	R
	4		X control	3	BS
		1*	En U,S,W	2	LR, LT
		1*	SU_ (could be F)		
	4		F1	4	F1
	4	*	F2	4	F2
	11		Next	10	Next
	41		*=reg. Prom MIR	32	

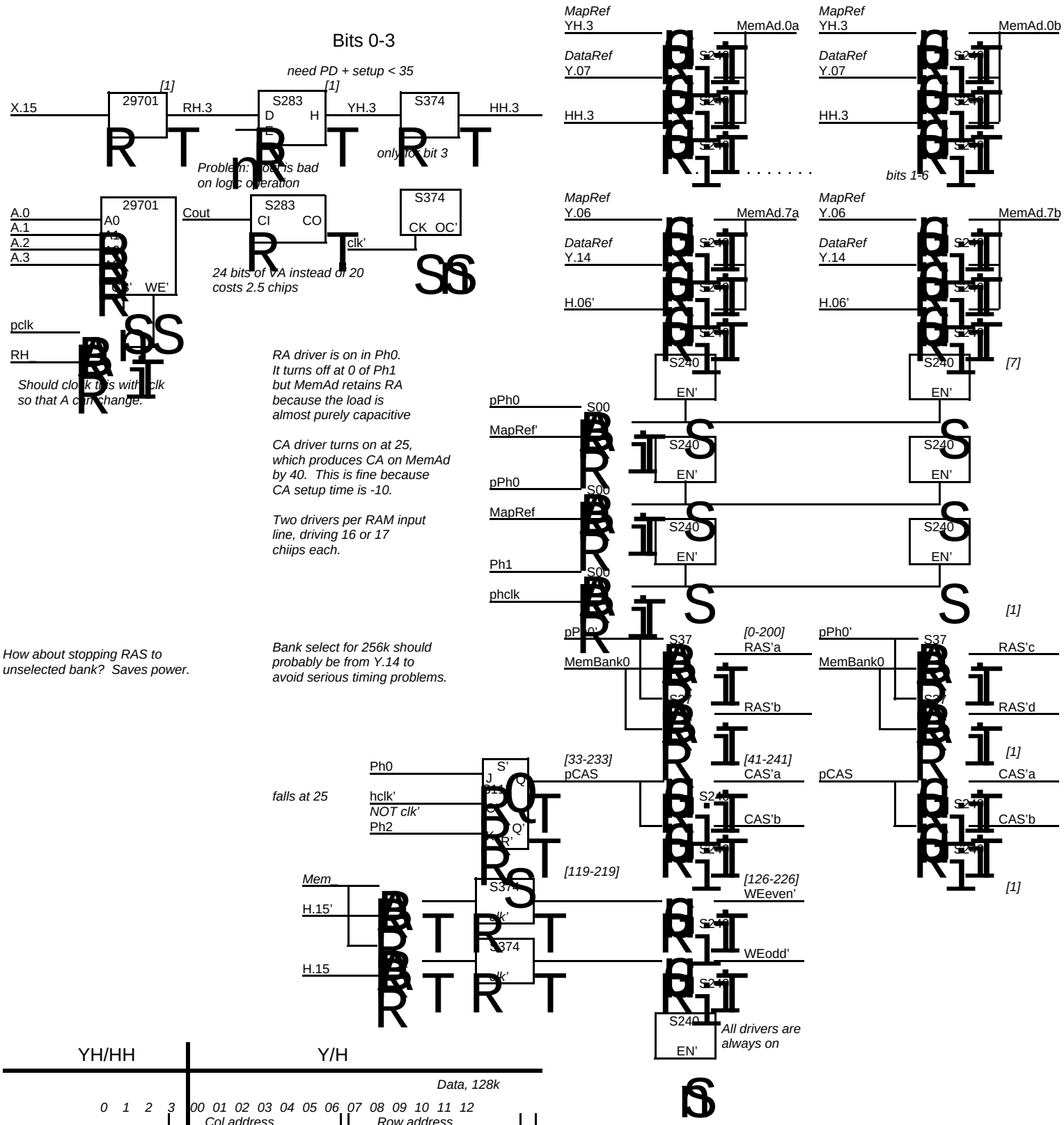
Chips

6	Prom (2kx8)	2	Stkp
5	MIR (12 b fast)	1+	PC stuff
27	TPCs, control	7	Ib
38		10	
21	2901, 182	12	Mem addr
6	S, U	34	64k RAMs
14-	W	2	Parity
2.5	H	48	
1.5	Shift ends		
4	F decoding		
6	Misc logic	13	Disk
55		15	Ethernet
		10	Display
		38	

Timing (IDM 2901A-1)

uInstruction		Storage (2 cy)		AB-arith		X-arith		AB-logic		SU-logic		X-logic		Y lcy 8	
14	fS399 Nia	0	RAS (Ph0)	5	fS74 MIR	30	ffEnu-SUdis	5	fS74 MIR	19	address	4	ffS388	5	fS74 MIR
70	Prom	150	RAM	45	AB-GP	30	X-GP	50	AB-Y	38		23	MIR-X	40	A-Y
12	25S09 SB	8	257	8	GP-Cin	8						32	D-Y	10	S253
96		40	X-clk^	35	Cin-clk^	35		35	SU set	40	X-clk^	35	SU setup	40	X-clk^
		198			25 Cin-Y			90		97	Y at 89	93		95	
				93	30 Cin-F=0	103	Need faster EnU				Need Y at 87 for Fdisp, at least.				
					22 Cin-F3										

XEROX PARC	Project	Drawing	File	Designer	Rev	Date	Page
	Wildflower	Main data paths	wf01.sil	Lampson		8/21/78	01



XEROX PARC	Project	Drawing	File	Designer	Rev	Date	Page
	Wildflower	Memory addressing	wf03.sil	Lampson		8/18/78	3

Ib transitions

Old IbPtr / Action	-1	0	1	2	3	4	5
NextIns	x	1 LG	2 LG	3 LG	3 R2	4 R1	0 R0
AlwaysNI	x	1 LG	2 LG	3 LG	-1 LG	0 G	x
Ib	x	1 L	2 L	3 L	4 L	5 L(0)	x
Ib_	0	1 L	2 L	3 L	4 L	x	x
IbPtr_0	x	0 . . .					
IbPtr_1	x	1 . . .					
IbPtr_2	x	2 . . .					
IbPtr_3	x	3 . . .					

refill only

*jump only also does Ib_*

IbVal is the contents of Ib
IbPtr is the state, normally =IbVal
IbSel is the Ibx byte selected (which is loaded into Ib is LoadIb)

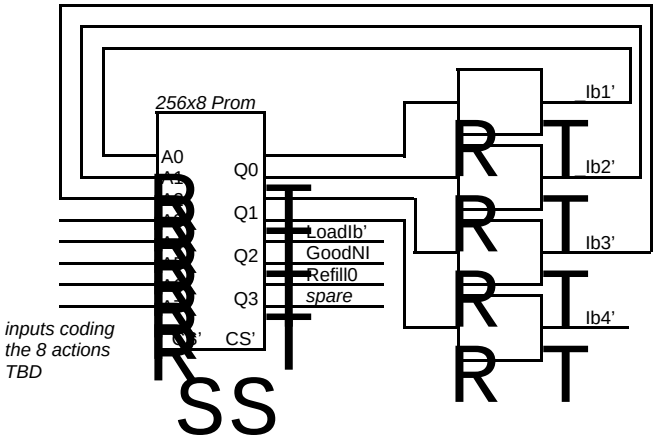
Notation
- *n New IbPtr. The next byte is the one selected*
- *L Load Ib*
- *G GoodNI*
- *Ri Refill dispatch*

Ib examples

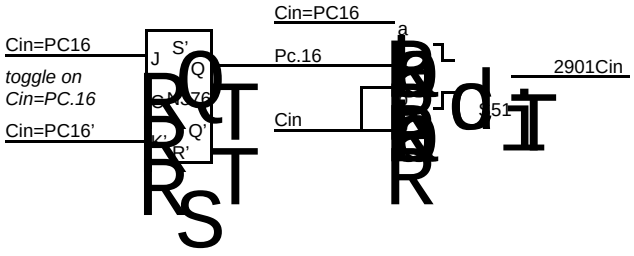
	Ib Val Ptr Sel	Val Ptr Sel	Val Ptr Sel	Val Ptr Sel
	No refill	Refill 2	Refill 1	Refill 0
_Ib	1 1 2	2 2 3	3 3 4	4 4 x
xxx, NextIns	2 2 3	3 3 4	4 4 x	x 5 x
xxx	3 3 4	3 3 4	4 4 x	x 0 1
Normal refill				
Mar_PC+1		3 3 4	4 4 x	
AlwaysNI		3 3 4	4 4 x	
Ib_MemIn		4 -1 x	4 0 1	
		4 0 1	1 1 2	
Empty refill				
Mar_PC+1				x 0 1
nop				x 0 1
Ib_MemIn				x 0 1
nop				1 1 2
AlwaysNI				1 1 2
nop				2 2 3
Jump to byte 2				
Mar_JumpAd	x x x			
nop	x x x			
IbPtr_1	x x x			
_Ib	x 1 2			
NextIns	2 2 3			
nop	3 3 4			
Jump to byte 3/4				
Mar_JumpAd	x x x	x x x		
nop	x x x	x x x		
IbPtr_2/3	x x x	x x x		
_Ib	x 2 3	x 3 4		
AlwaysNI	3 3 4	4 4 x		
Ib_MemIn	4 -1 x	4 0 1		
	4 0 1	1 1 2		

during this cycle

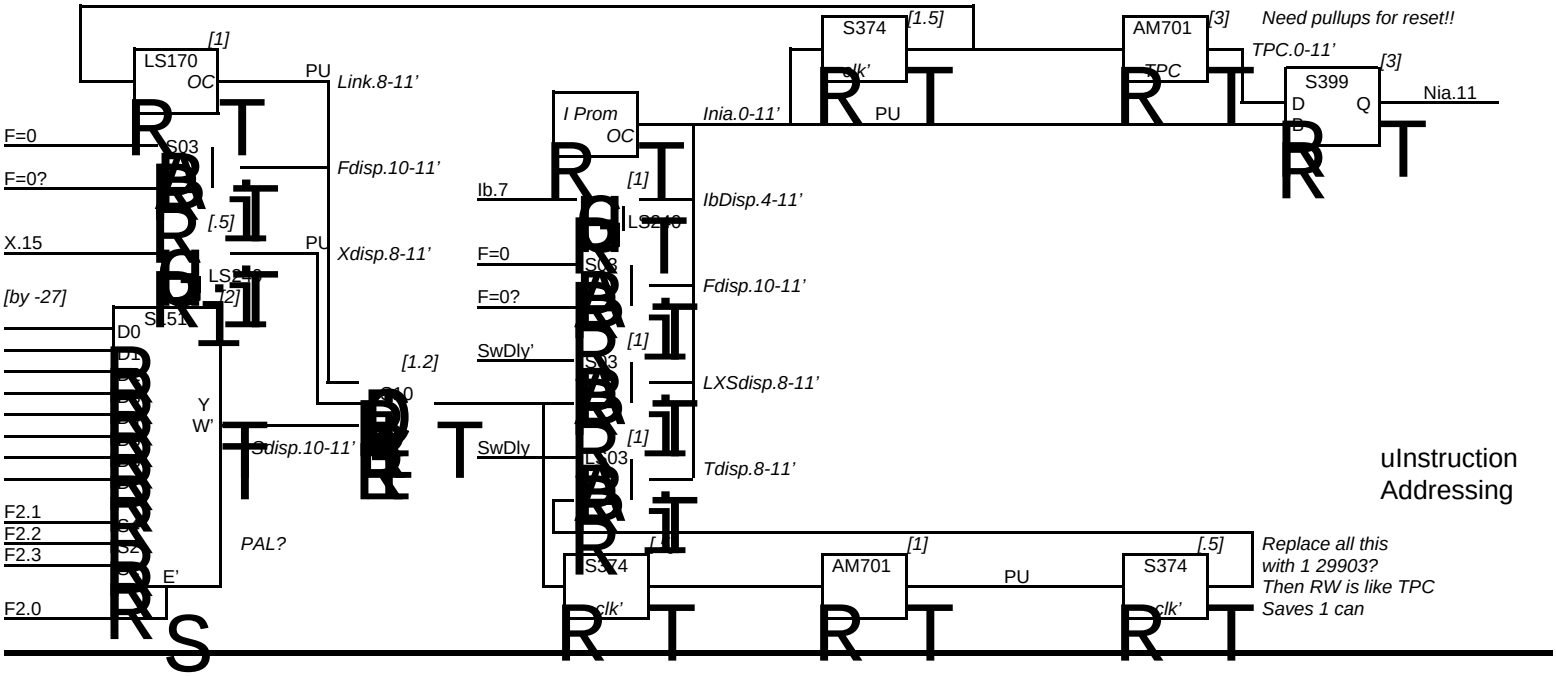
Code:	IbPtr	IbSel	Ib1234
	-1	x	100x
	0	1	1000
	1	2	0100
	2	3	0010
	3	4	0001
	4	x	110x
	5	x	111x

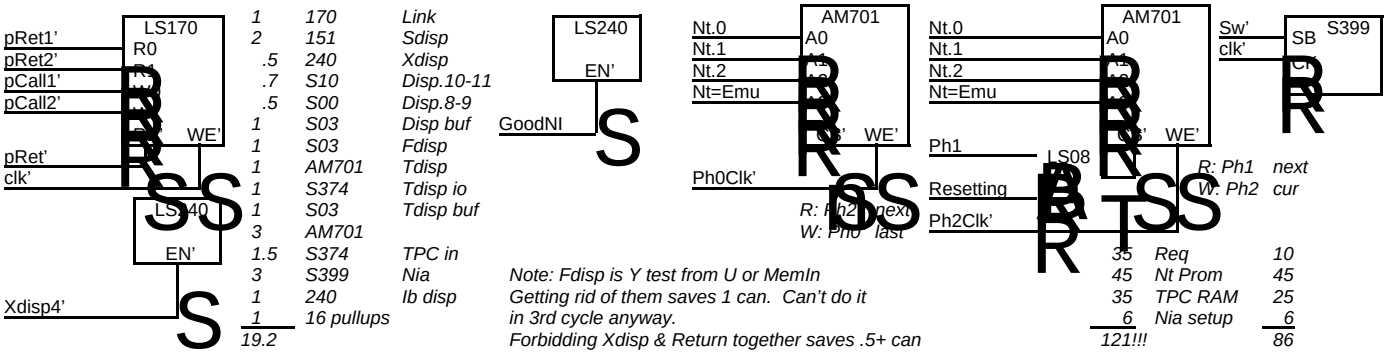


inputs coding the 8 actions TBD



XEROX PARC	Project	Drawing	File	Designer	Rev	Date	Page
	Wildflower	Ib control, PC.16	wf04.sil	Lampson		8/18/78	4

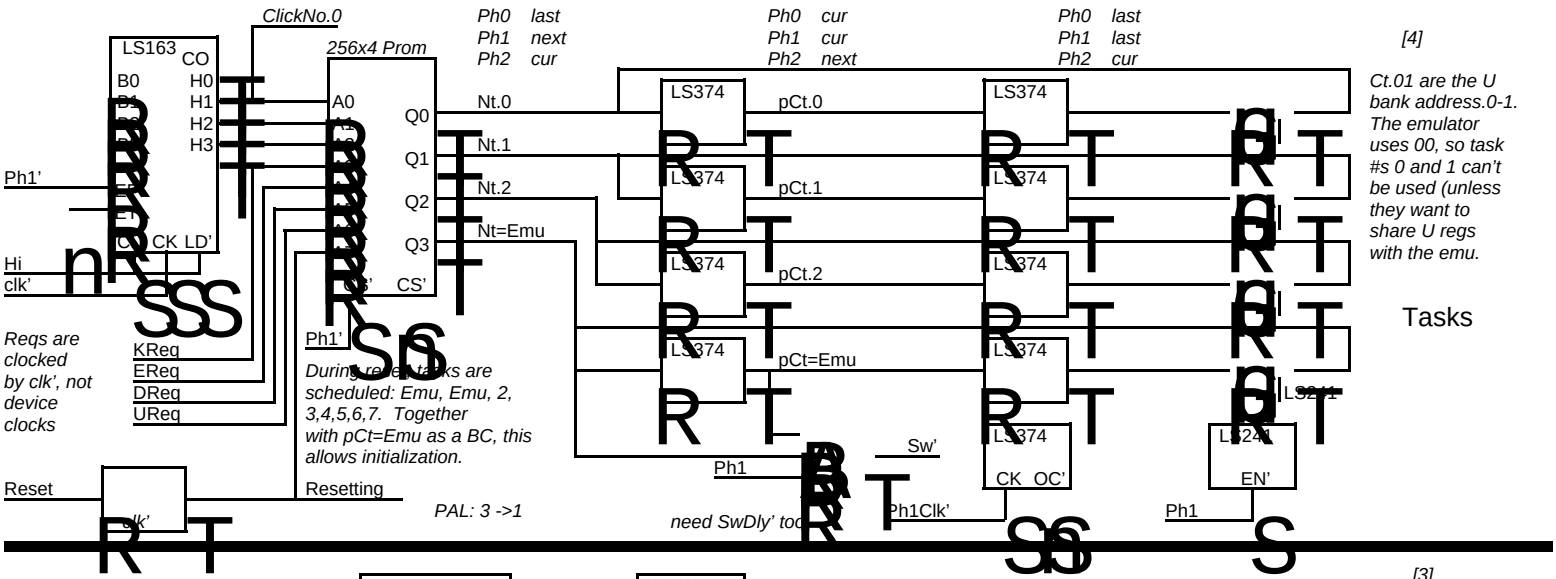


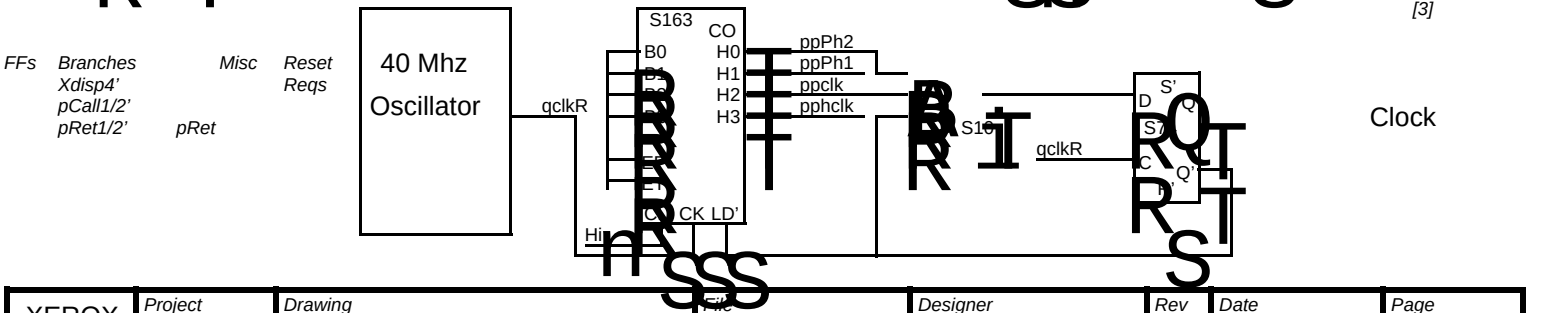


1	170	Link
2	151	Sdisp
.5	240	Xdisp
.7	S10	Disp.10-11
.5	S00	Disp.8-9
1	S03	Disp buf
1	S03	Fdisp
1	AM701	Tdisp
1	S374	Tdisp io
1	S03	Tdisp buf
3	AM701	
1.5	S374	TPC in
3	S399	Nia
1	240	Ib disp
1	16 pullups	
19.2		

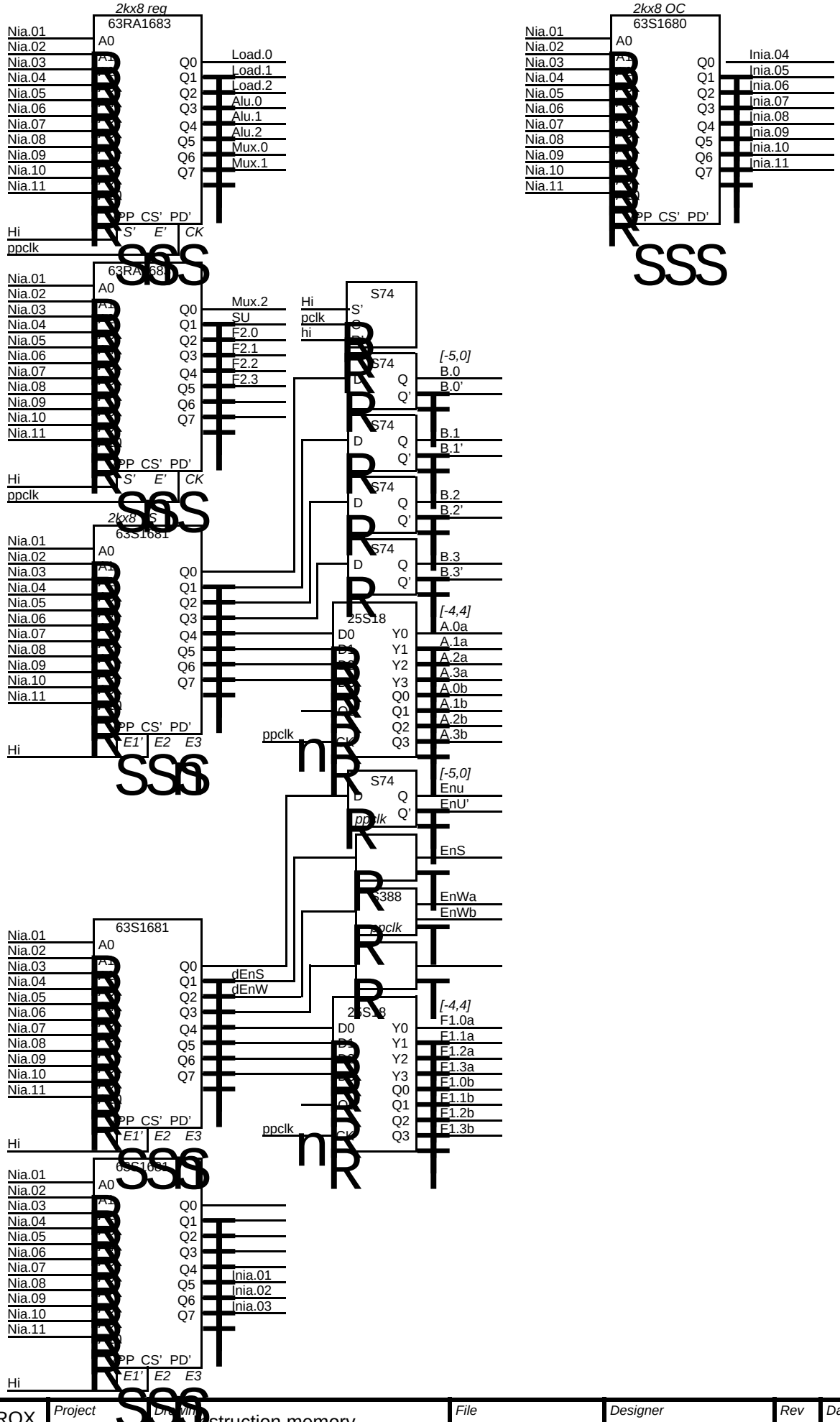
Note: Fdisp is Y test from U or MemIn
Getting rid of them saves 1 can. Can't do it
in 3rd cycle anyway.
Forbidding Xdisp & Return together saves .5+ can

35	Req	10
45	Nt Prom	45
35	TPC RAM	25
6	Nia setup	6
121!!!		86



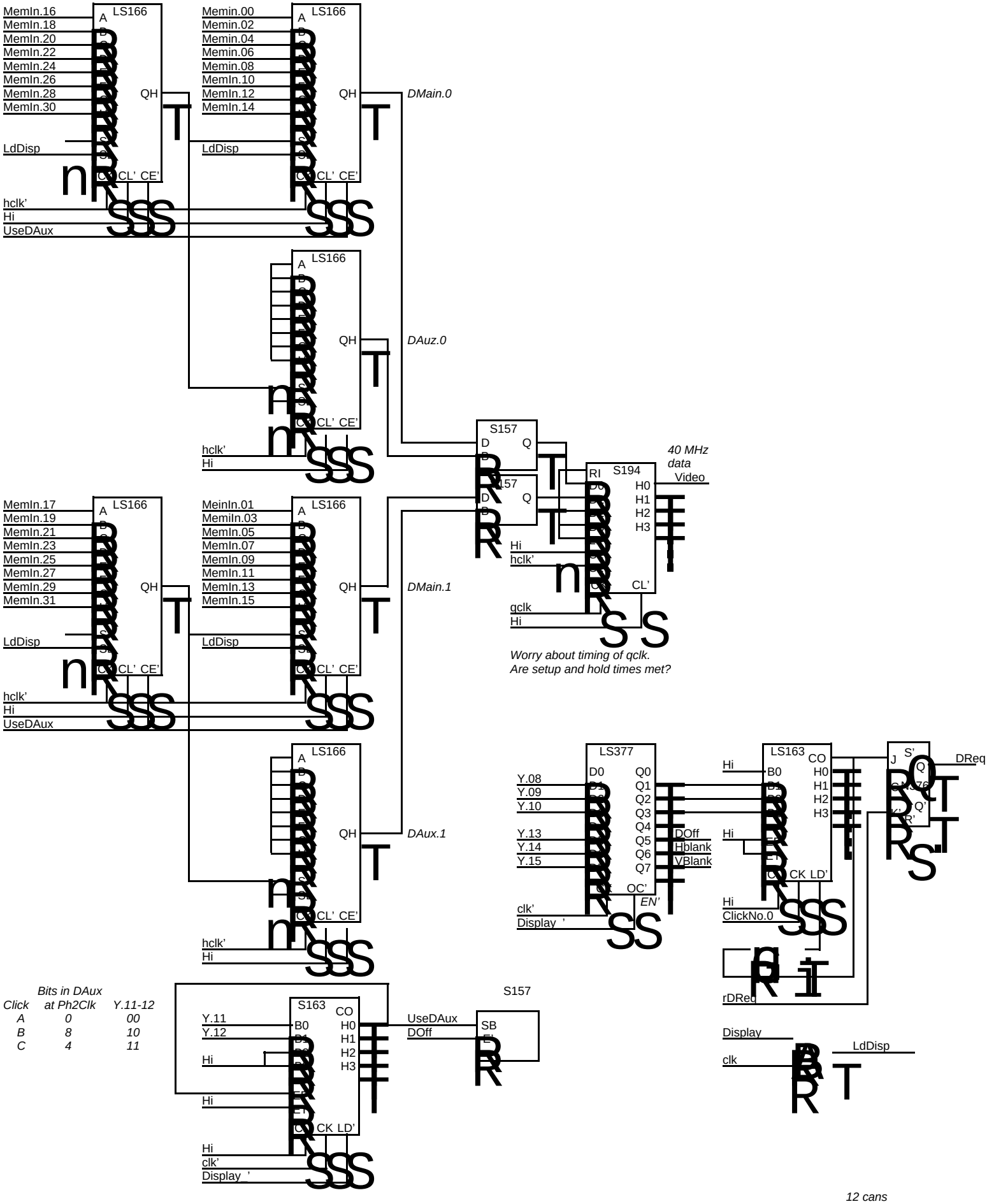


XEROX PARC	Project	Drawing	File	Designer	Rev	Date	Page
	Wildflower	Control	wf05.sil	Lampson		8/18/78	5



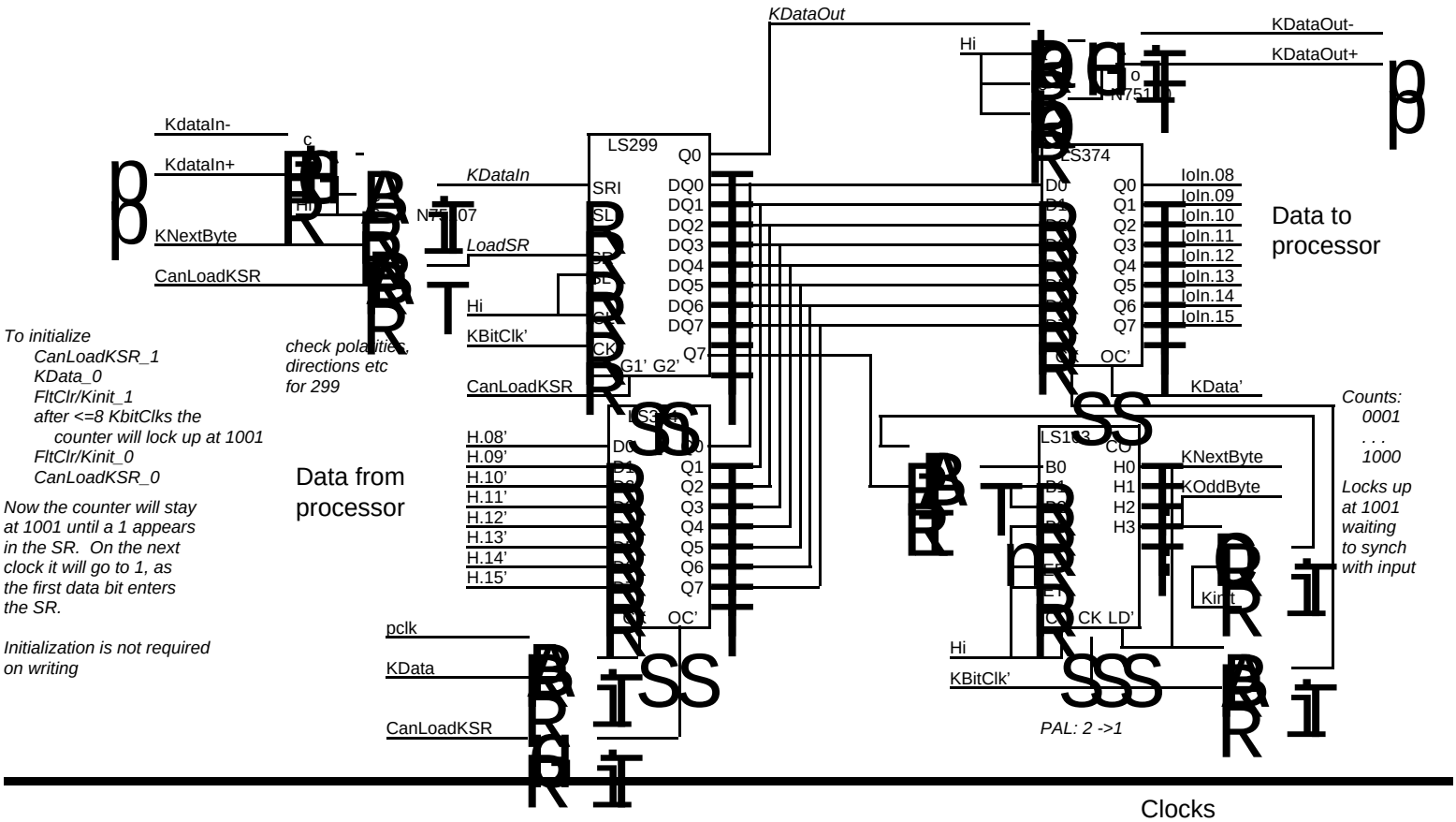
about 11 chips

XEROX PARC	Project: Wildflower	Drawing: Instruction memory MIR	File: wf06.sil	Designer: Lampson	Rev	Date: 8/21/78	Page: 6

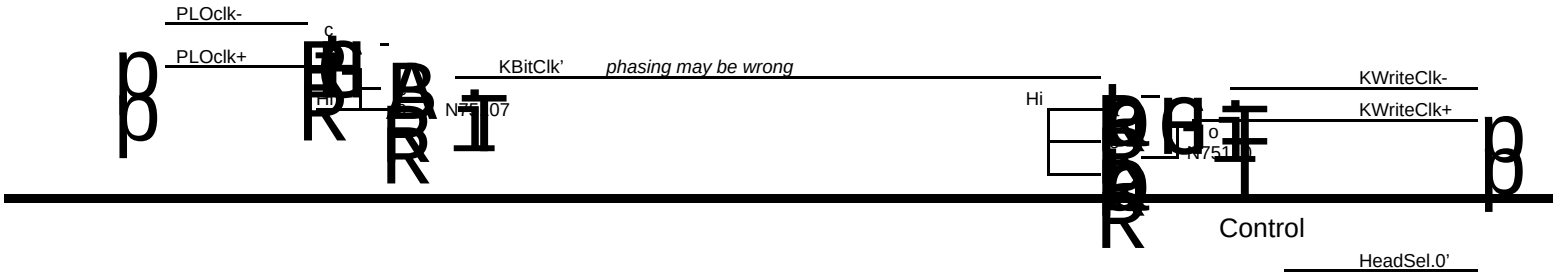


XEROX PARC	Project	Drawing	File	Designer	Rev	Date	Page
	Wildflower	Display	wf07.sil	Lampson		8/18/78	7

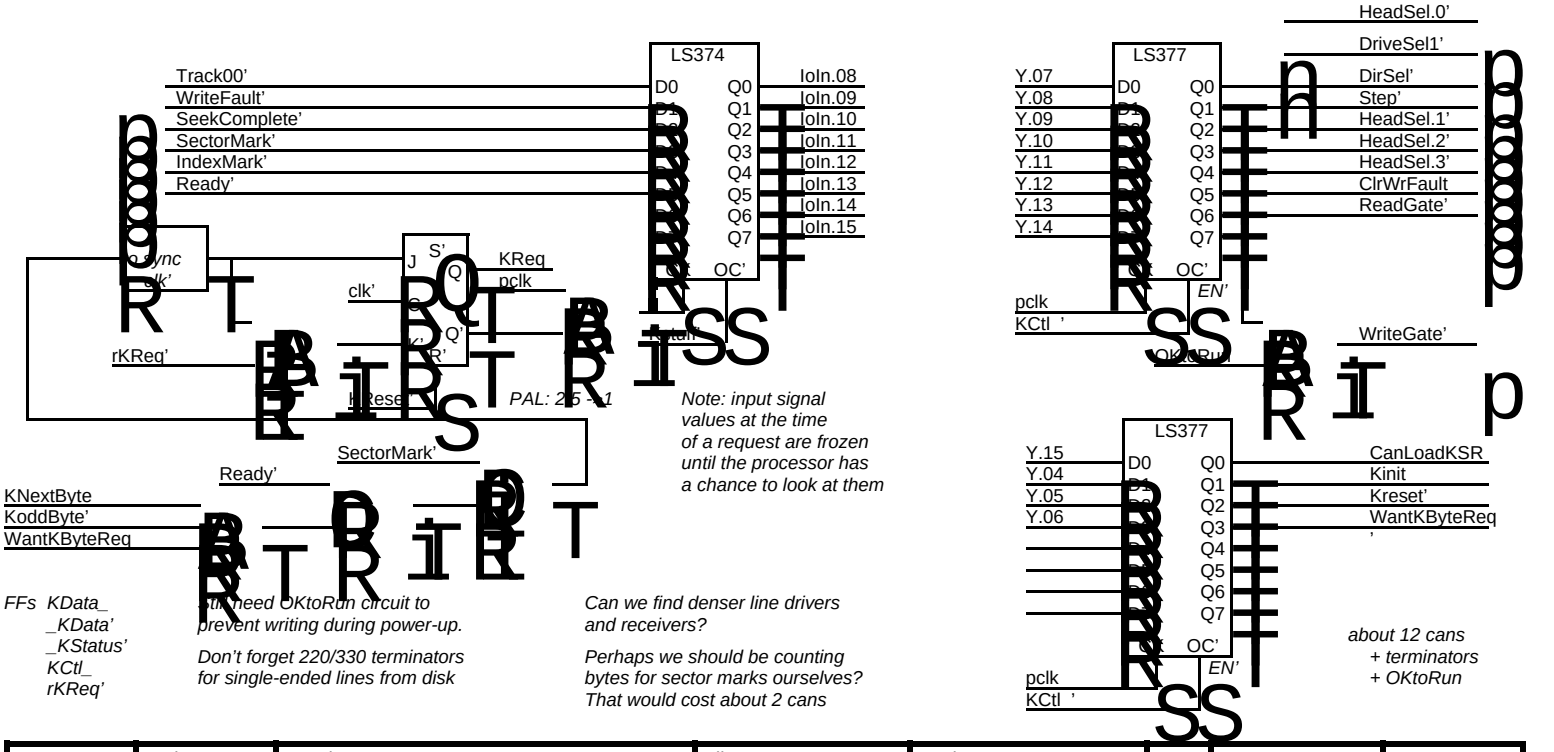
Data



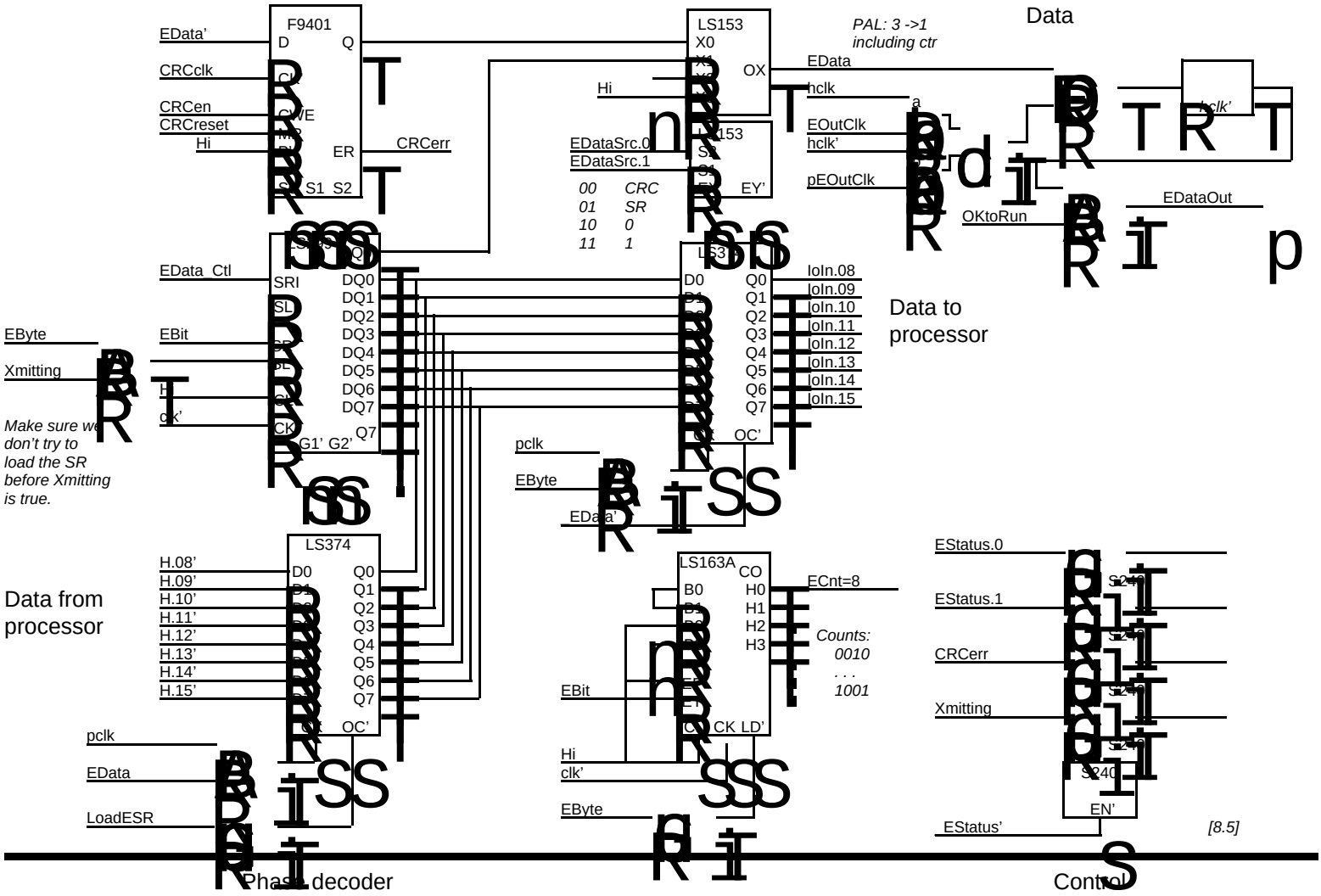
Clocks

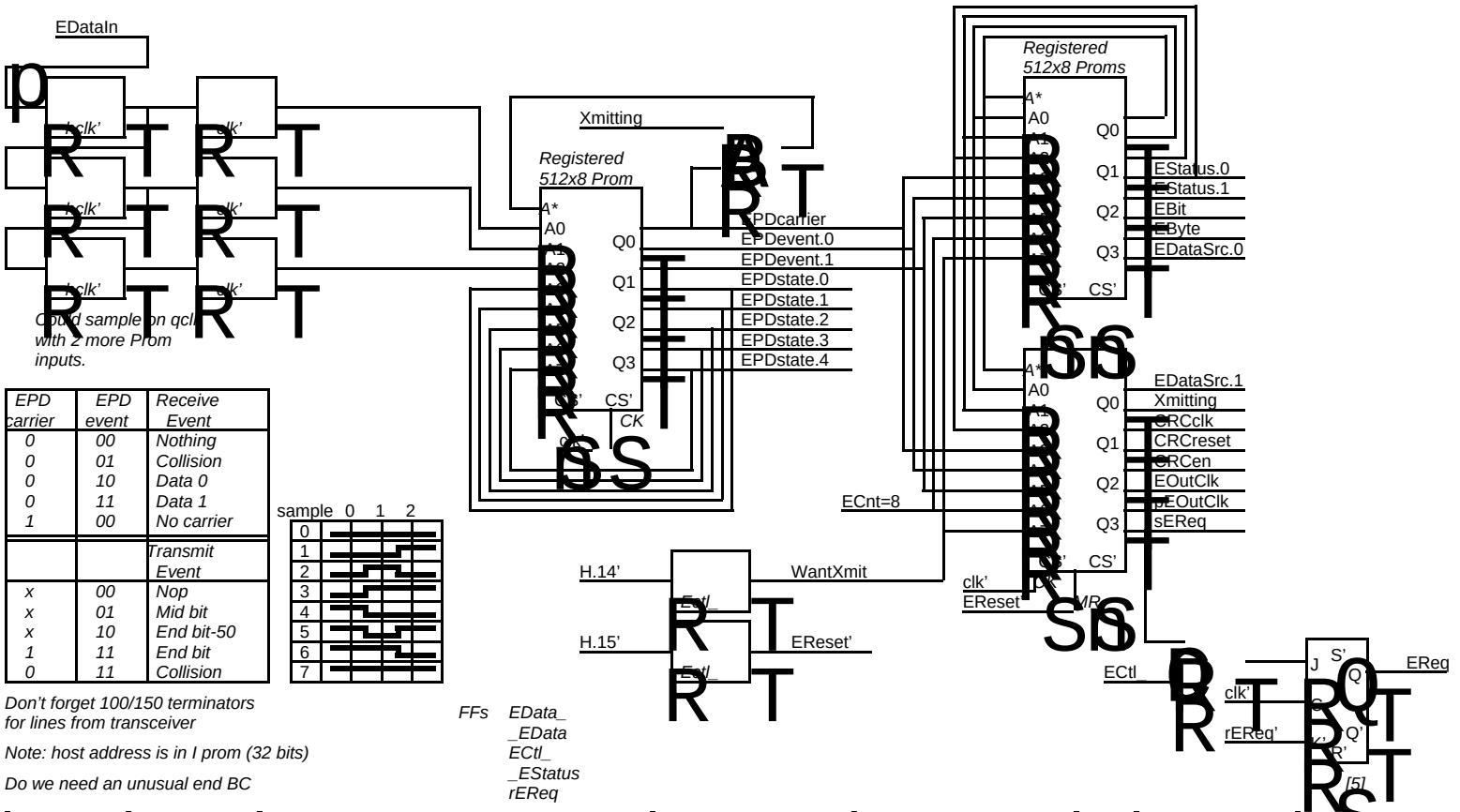


Control



XEROX PARC	Project	Drawing	File	Designer	Rev	Date	Page
	Wildflower	Disk controller	wf08.sil	Lampson		8/18/78	8





EPD carrier	EPD event	Receive Event
0	00	Nothing
0	01	Collision
0	10	Data 0
0	11	Data 1
1	00	No carrier
		Transmit Event
x	00	Nop
x	01	Mid bit
x	10	End bit-50
1	11	End bit
0	11	Collision

sample 0 1 2

0 1 2 3 4 5 6 7

Don't forget 100/150 terminators for lines from transceiver

Note: host address is in I prom (32 bits)

Do we need an unusual end BC

FFs EData_ _EData ECtl_ _EStatus rEReq

XEROX PARC	Project: Wildflower	Drawing: Ethernet controller	File: wf09.sil	Designer: Lampson	Rev	Date: 8/18/78	Page: 9

Phase decoder and encoder

Receiver states

1	no carrier
9	time since last data transition 1, 2, 3, 4, 5, 6, 7, 8, 9

Transmitter states

1	collision
35	7 states to count bit clocks X 5 times since last transition 1, 2-3, 4-5, 6-7, 8-9

Total states

49 out of 64

Constraints

Need an easy (1 bit) way to force R-idle into T-start
It would be nice if the first few R states were unaffected.

Need to code R: carrier vs no carrier
T: running vs collision
into one bit. Also, there should be one "starting transmission" state different in the 3 outputs from any R state, so that successful start of transmission can be distinguished from simultaneous start of reception.

In					Out					
	Receive		Transmit				Receive		Transmit			
State	Carrier	PDcnt	PEcnt	Tcnt	Sample	State	Carrier	PDcnt	PEcnt	Tcnt	Event	Note
	L	x			0 7		L	x			Nop	Idle
					2 4 5 6		H	1			Coll	Impossible
					1 3		H	1, 2			One	Start packet
	H	1			1 6, 3 4		H	1, 2			Coll	Too many
	H	1-7			0 7		H	+2			Nop	Active
	H	1-9			2 5		H	1			Coll	Too many
	H	2-4			3 4		H	+2			Nop	Setup
	H	2-3			1 6		H	+2			Nop	Setup transition
	H	5-9			3, 4		H	2			One, Zero	Data transition
	H	4-8			1, 6		H	1			One, Zero	Data transition
	H	9			1 6		H	1			Coll	Too few
	H	8-9			0		L	X			Nop	End of packet
	H	8-9			7		H	0			Coll	Jam
			x	1-11	2 5		H	1			Coll	Too many
			x	1	1 6, 3 4		H	1, 2			Coll	Too many
			x	9-11	0		H	9			Coll	Impossible
			x	9-11	7		H	0			Coll	Jam
			x	11	1 6, 3 4		H	1, 2			Coll	Impossible
			x	1-7	0 7				*	+2	depends	
			x	3	1 6 3 4				*	+2	on	
			x	5-7	1 6, 3 4				*	1, 3	PEcnt	
			2	x					4	*	Nop	
			4	x					6	*	Mid bit	
			6	x					8	*	Pre end bit	
			8	x					10	*	Nop	
			10	x					12	*	Mid bit	
			12	x					14	*	Nop	
			14	x					2	*	End bit	

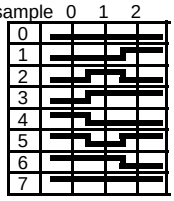
Receive

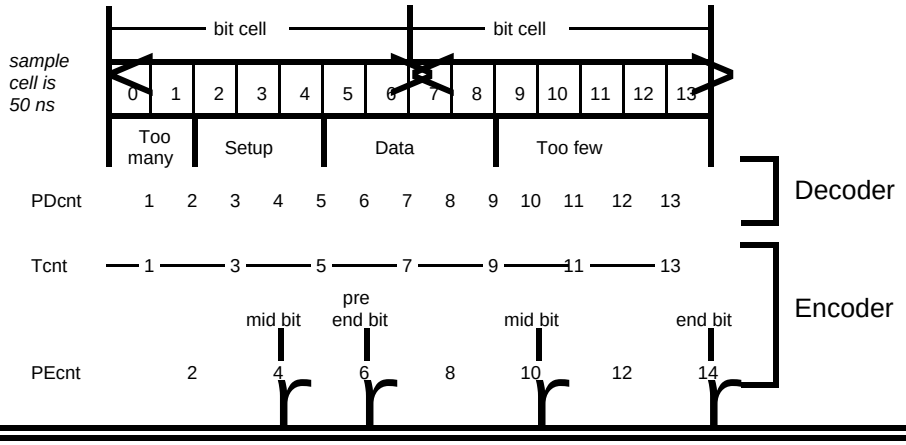
Transmit collision detection

Transmit bit cell counting

EPD carrier	*EPD event*	*Receive Event*
0	*00*	*Nothing*
0	*01*	*Collision*
0	*10*	*Data 0*
0	*11*	*Data 1*
1	*00*	*No carrier*
		Transmit Event
x	*00*	*Nop*
x	*01*	*Mid bit*
x	*10*	*End bit-50*
1	*11*	*End bit*
0	*11*	*Collision*

Cnts are from a previous transition (data trans for decoder) in cell 0. At cnt i we know that the time since the transition is between i-1 and i cells. When contemplating cnt=i, the Prom can also see the state of the line at i+1 and i+2.





Ethernet control

State	Meaning
0	reset - go to 1 if carrier off, else 2
1	idle (carrier off): start receiving if carrier, else start transmitting if WantXmit, else stay here
2	waiting for carrier off: stay here until carrier off, then go to 1
3-4	reporting results - lsb of state is part of status code. Stay here until reset
5-7	receive states: maybe, full, imip (?)
8-11	transmit states: mark, data, CRC1, CRC2
12-15	unused!

Note: to turn off EDataOut, Prom must make Edata 0 (by EdataSrc=10) and both EOutClk's=1

Part	Function	Usage						
		ALU	Control	IB	Disk	Ethernet	Display	Storage
	Prom, 2kx4		10					
	Prom, 512x8					3		
	Prom, 512x4		1					
	Prom, 256x8	2						
IDM 2901A-1 (National)	4 bit ALU slice	4						
74S182	Carry accelerator	1						
74S283	4-bit adder							1
74LS374	)ctal 3-S reg		4	5	4		1	
74S374	)ctal 3-S reg	2	4			5		1
AM25S09 (AMD)	2x-mux/reg	2	3					
74S299	8-bit 3-S shift reg			1	1 LS	1 LS		
74LS166	8-bit P-S shift reg						6	
74S194	4-bit shift reg						1	
74S373	Octal latch							2
74S74	Dual D flipflop		5					
93422 (Fairchild)	256x4 RAM	4						
74S189	16x4 RAM		3					1
	64k MOS RAM							34
74S169	4-bit up-down ctr	1						
74S163	4-bit counter		2		1	1 LS	1 LS	
74S251	8x mux	16	2					
74S153	4x mux	1				1 LS		
74S258	2x mux		3					
74S240	Octal 3-S buffer	1	2					5
74S138	3:8 decoder	4						
74S280	9-bit parity							2
Misc SSI		8	4	1	5	4	1	3
	40 MHz oscillator		1					
9401 (Fairchild)	CRC chip					1		

Parts list for 128k machine with display, disk and ethernet controllers

Since the design is not complete, it might be prudent to allow for 10 additional MSI parts (e.g. 74S374 or 74S258) and 10 additional SSI parts.