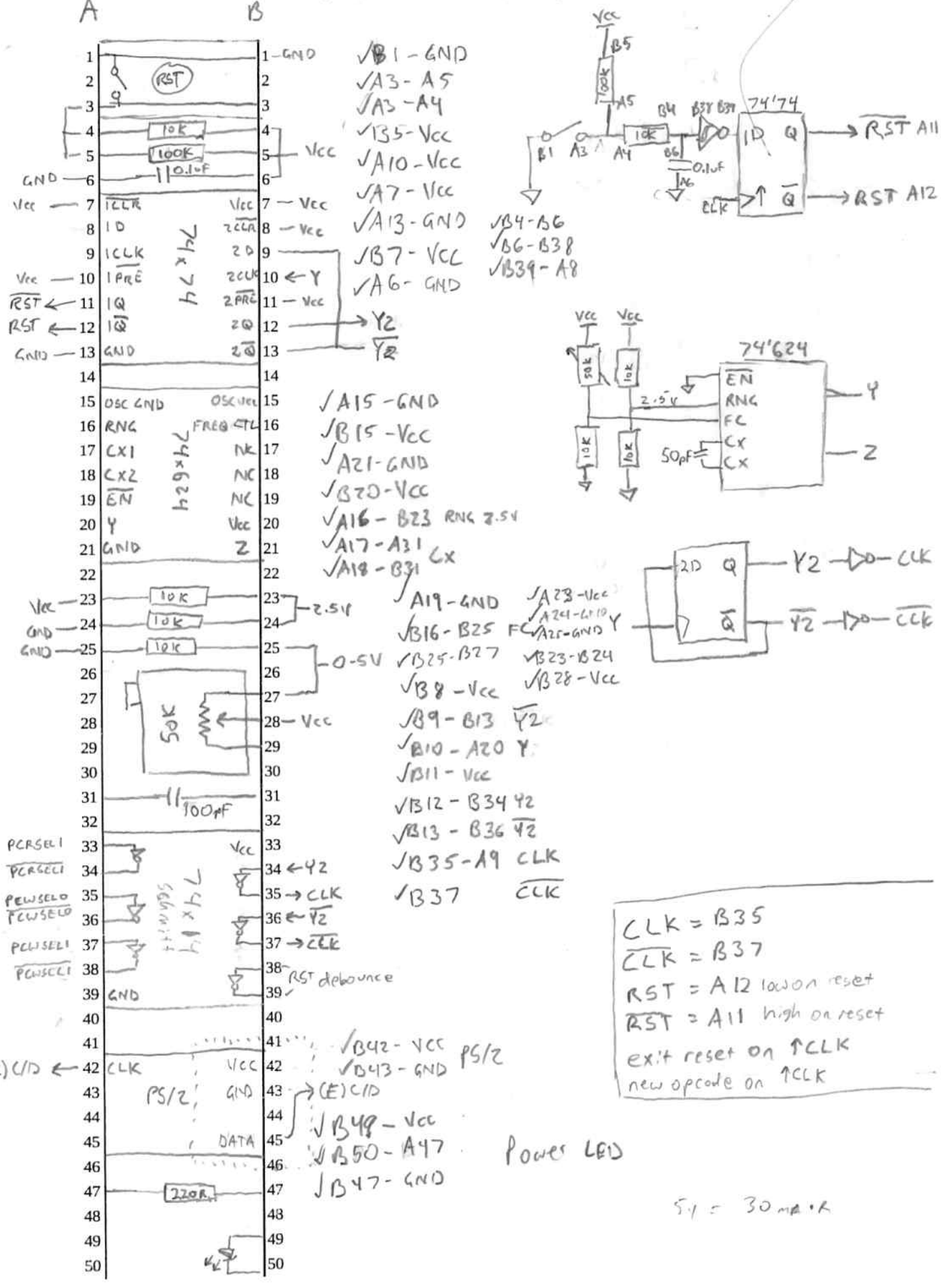
Group A

Reset + Clock Generation



CLK = B35

$\overline{CLK}$ = B37

RST = A12 low on reset

$\overline{RST}$ = A11 high on reset

exit reset on ↑CLK

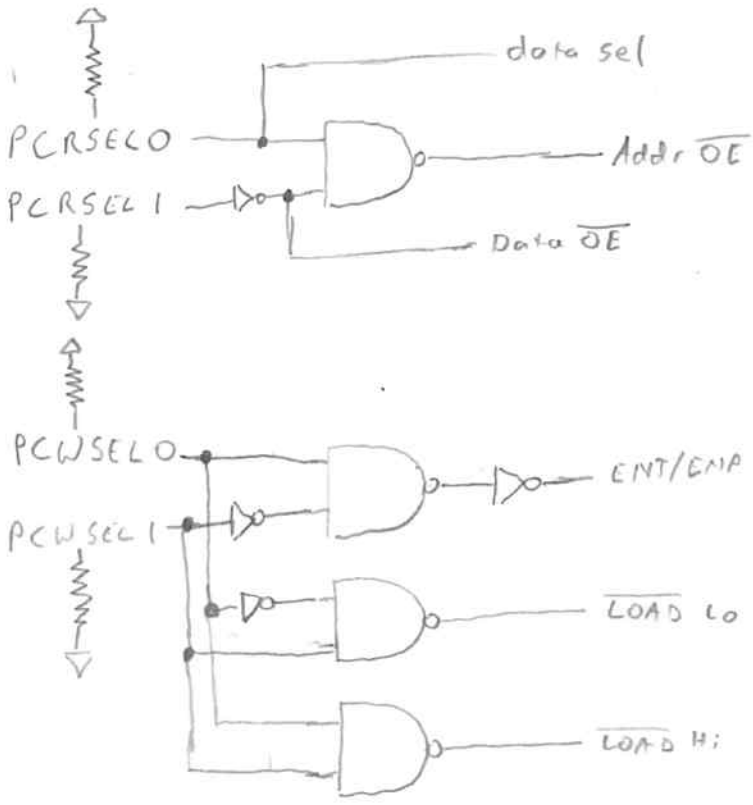
new opcode on ↑CLK

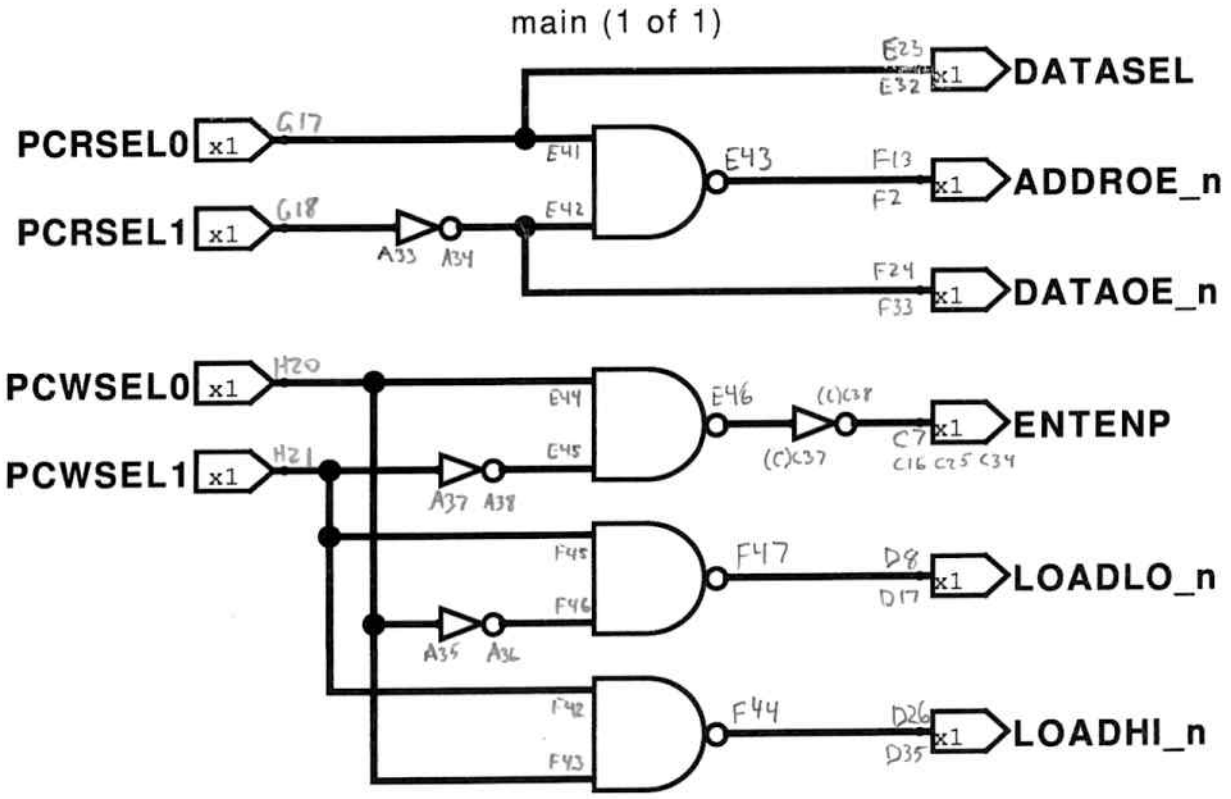
5V = 30mA

Program Counter control Logic

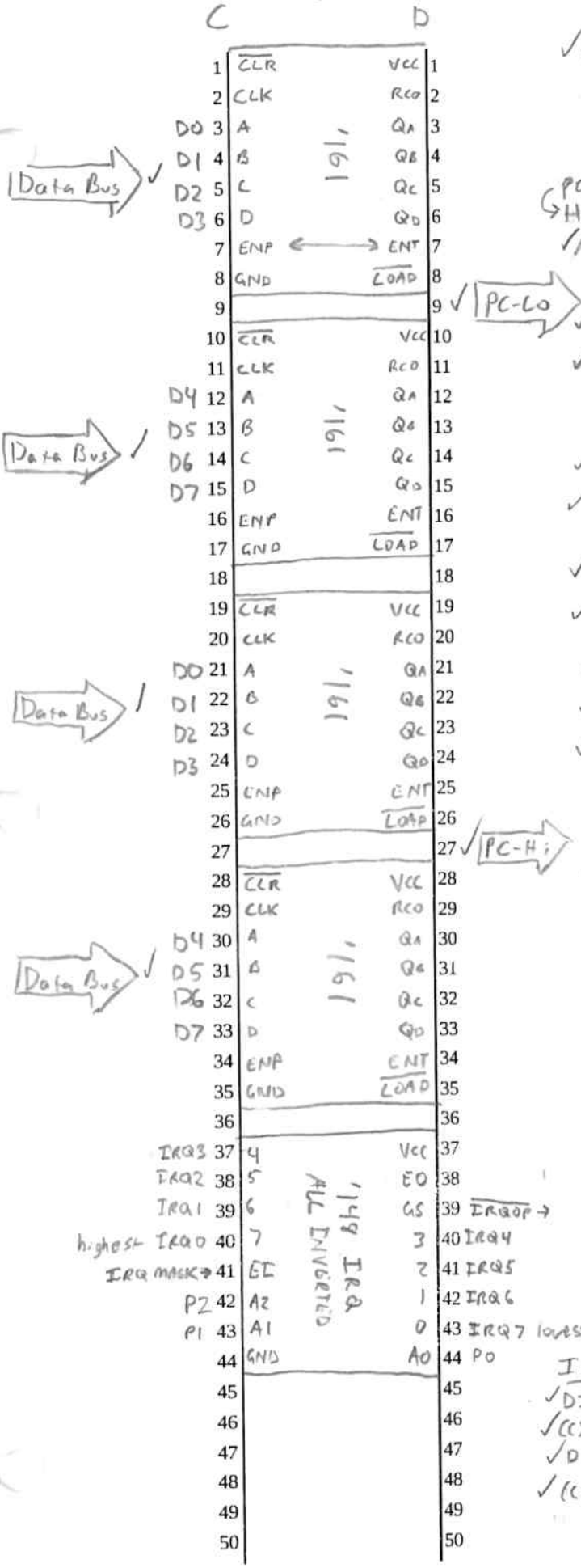
PCRSEL		Addr $\overline{OE}$	data sel	data $\overline{OE}$
0 0	do nothing	1	X	1
0 1	PC to addr bus	0	X	1
1 0	PCL to data bus	1	0	0
1 1	PCH to data bus	1	1	0

PCWSEL		$\overline{LOAD}$ Hi	$\overline{LOAD}$ Lo	ENT ENA
0 0	do nothing	1	1	0
0 1	increment PC	1	1	1
1 0	write PCL	1	0	0
1 1	write PCH	0	1	0




main (1 of 1)
PCRSEL0
PCRSEL1
PCWSEL0
PCWSEL1
x1
G17
G18
H20
H21
A33
A34
A37
A38
A35
A36
E41
E42
E43
E44
E45
E46
F45
F46
F47
F42
F43
F44
E23
E32
F13
F2
F24
F33
C7
C16
C25
C34
D8
D17
D26
D35
DATASEL
ADDROE_n
DATAOE_n
ENTENP
LOADLO_n
LOADHI_n

Program Counter 1/2



✓C28-C19 ✓C10-C1 ✓C19-C10
✓C1-A12 RST

PCWSEL1
↳ H21-A37✓ ✓E46-(C)C37 ✓(C)C38-C7 ✓A38-E45 ✓H20-E44 ✓C7-D7 } PC inc

✓D1-Vcc ✓C6-GND ✓D10-Vcc ✓C17-GND
✓D19-Vcc ✓C26-GND ✓D28-Vcc ✓C35-GND

✓C24-C19 ✓C19-C10 ✓C10-C1 ✓C1-A11 ← RST
✓C29-C20 ✓C20-C11 ✓C11-C2 ✓C2-B35 ← CLK

✓C7-C16 ✓C16-C25 ✓C25-C34 ← ENP

✓D2-D16 ✓D11-D25 ✓D20-D34 ← RCO-ENT

✓D3-E24 ✓D4-E27 ✓D5-F28
✓D6-F25 ✓D12-E33 ✓D13-E36
✓D14-F37 ✓D15-F34 } Data LO out

✓D21-E25 ✓D22-E28 ✓D23-F29
✓D24-F26 ✓D30-E34 ✓D31-E37
✓D32-F38 ✓D33-F35 } Data HI out

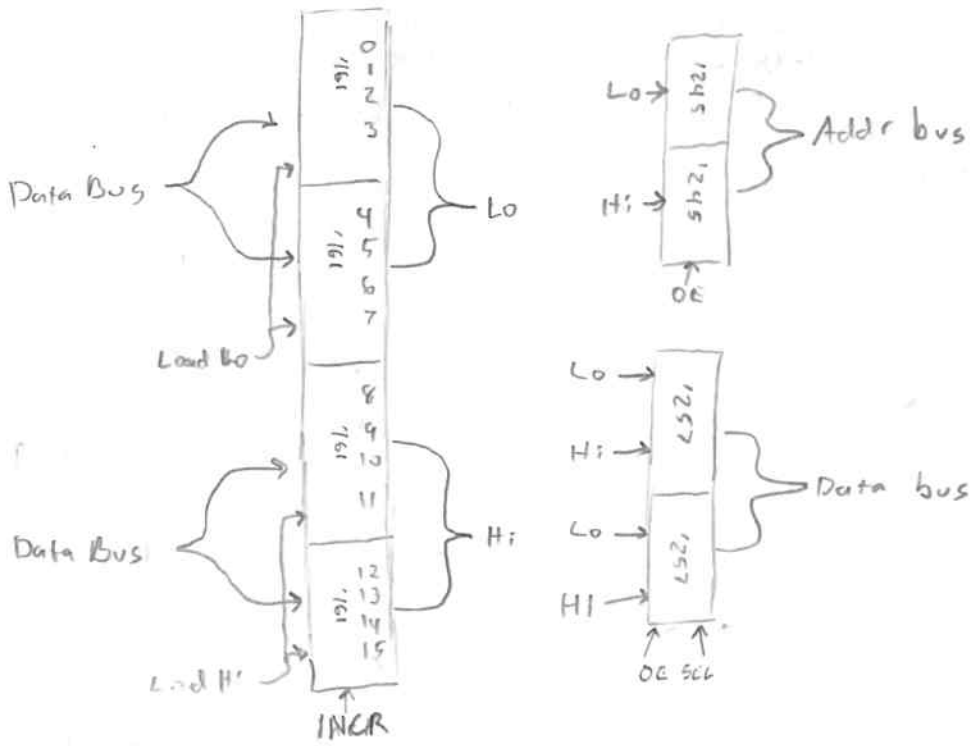
✓D3-E2 ✓D4-E3 ✓D5-E4
✓D6-E5 ✓D12-E6 ✓D13-E7
✓D14-E8 ✓D15-E9 } Addr LO out

✓D21-E13 ✓D22-E14 ✓D23-E15
✓D24-E16 ✓D30-E17 ✓D31-E18
✓D32-E19 ✓D33-E20 } Addr HI out

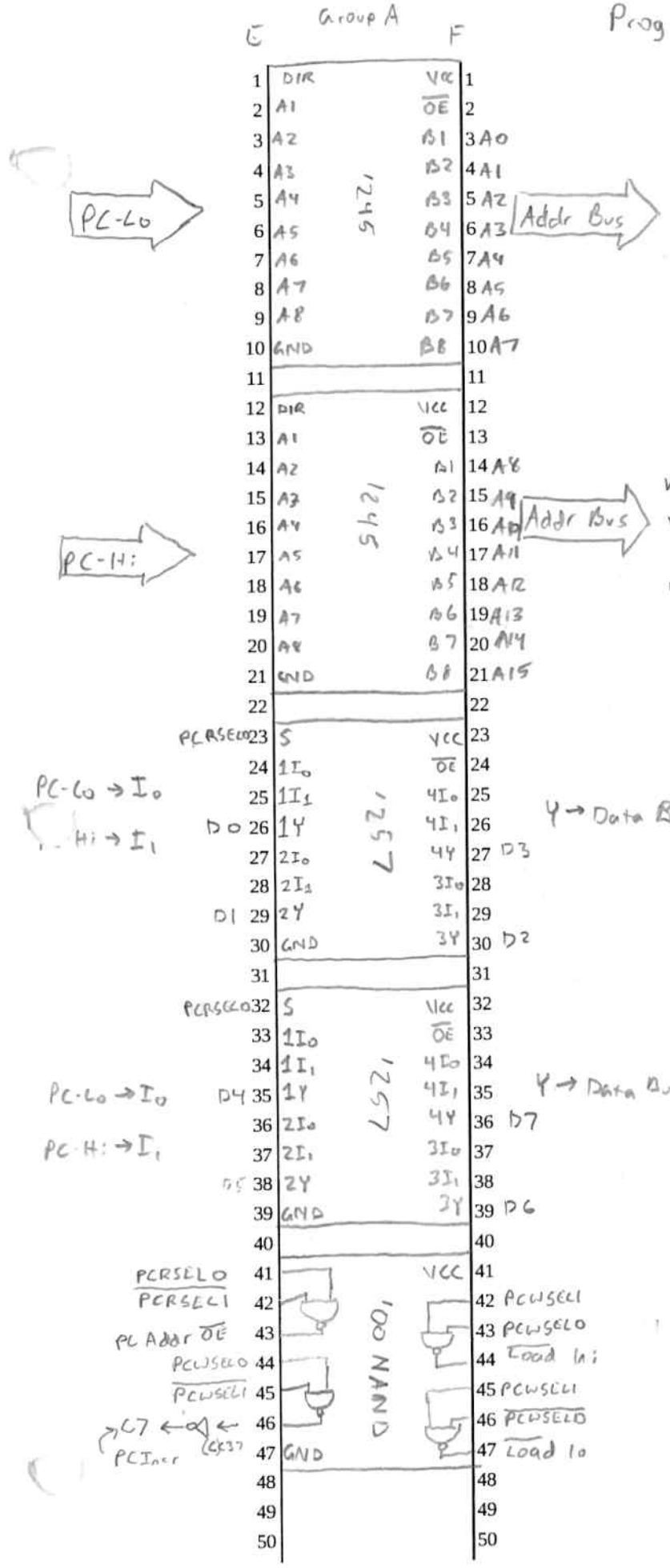
✓C3-C21 ✓C4-C22 ✓C5-C23 ✓C6-C24
✓C12-C30 ✓C13-C31 ✓C14-C32 ✓C15-C33
← Data IN ganging

IRQ Handler
✓D37-Vcc ✓C44-GND
✓(C)C27-C41 IRQ MASK
✓D39-(C)C35 invert
✓(C)C36-(B)E41 IRQOP

IRQ Pull-up - Green
✓IRQ0 C40-(B)G21 ✓(B)H21-Vcc
✓IRQ1 C39-(B)G22
✓IRQ2 C38-(B)G23
✓IRQ3 C37-(B)G24
✓IRQ4 D40-(B)G25
✓IRQ5 D41-(B)G26
✓IRQ6 D42-(B)G27
✓IRQ7 D43-(B)H27


Data Bus
Load Lo
Lo
Data Bus
Load Hi
Hi
INCR
Lo
Hi
Addr bus
OE
Lo
Hi
Lo
HI
Data bus
OE
SCL

Program Counter 2/2



PCRSEL1
✓G18 - A33

PCRSELO
✓G17 - E41
✓A34 - E42
✓E41 - E32
✓E42 - F33

✓F13 - F2, ✓E43 - F13 } Addr OE
✓E23 - 32 DataSel
✓F33 - F24 } Data OE

PCWSEL1
✓H20 - A35

PCWSELO
✓H21 - F45
✓A36 - F46
✓F42 - F45
✓F43 - E44

✓F47 - D8, ✓D8 - D17 } Load Lo
✓F44 - D26, ✓D26 - D35 } Load Hi

PCWSELO

✓F1 - VCC ✓E10 - GND ✓F12 - VCC ✓E21 - GND
✓F23 - VCC ✓E30 - GND ✓F32 - VCC ✓C39 - GND
✓F41 - VCC ✓E47 - GND

✓E1 - VCC ✓E12 - VCC ←Dir = 1

✓F3 - G5 A0
✓F4 - H6 A1
✓F5 - G6 A2
✓F6 - H7 A3
✓F7 - G7 A4
✓F8 - H10 A5
✓F9 - G10 A6
✓F10 - H11 A7

✓F14 - G11 A8
✓F15 - H12 A9
✓F16 - G12 A10
✓F17 - H13 A11
✓F18 - G13 A12
✓F19 - H14 A13
✓F20 - G14 A14
✓F21 - H15 A15

← Addr Bus Out →

Group B

✓E26 - C49 D0
✓E29 - C48 D1
✓F30 - C47 D2
✓F27 - C46 D3

Group B

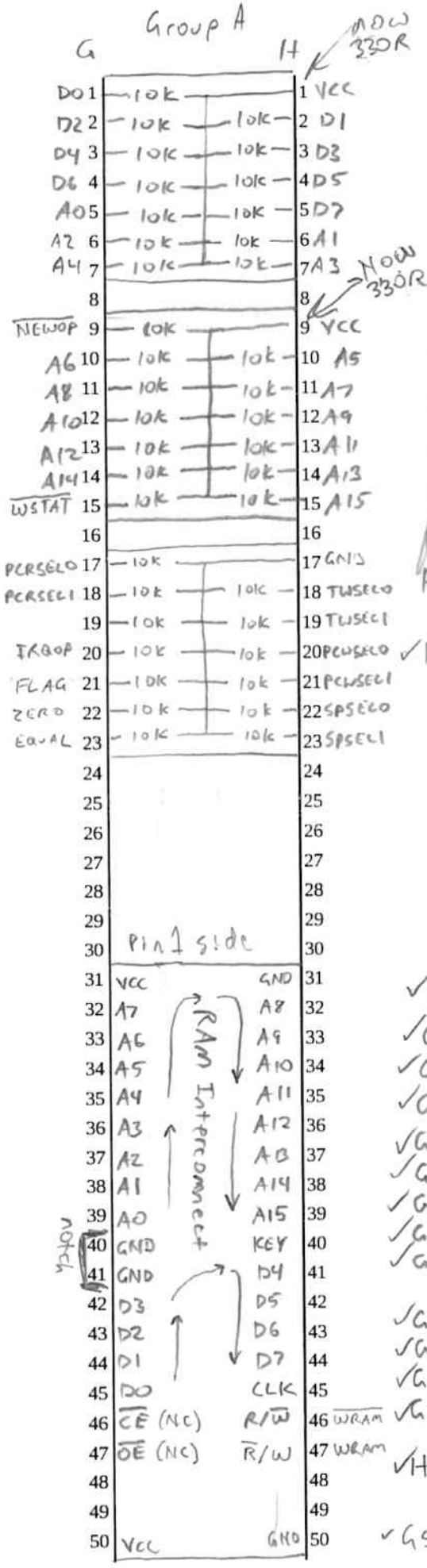
✓E35 - C45 D4
✓E38 - C44 D5
✓F39 - C43 D6
✓F36 - C42 D7

← Data Bus Out →

← Data Bus Into PC →

✓E26 - C21 D0
✓E29 - C22 D1
✓F30 - C23 D2
✓F27 - C24 D3

✓E35 - C30 D4
✓E38 - C31 D5
✓F39 - C32 D6
✓F36 - C33 D7



✓H1 - VCC ✓H9 - VCC

~~C3 - G1 D0~~ ~~C4 - H2 D1~~
~~C5 - G2 D2~~ ~~C6 - H3 D3~~
~~C12 - G3 D4~~ ~~C13 - H4 D5~~
~~C14 - G4 D6~~ ~~C15 - H5 D7~~

Data Pull UP ↑ From PC ↓ Address Pull UP

~~F3 - G5 A0~~ ~~F4 - H6 A1~~
~~F5 - G6 A2~~ ~~F6 - H7 A3~~
~~F7 - G7 A4~~ ~~F8 - H10 A5~~
~~F9 - G10 A6~~ ~~F10 - H11 A7~~
~~F11 - G11 A8~~ ~~F15 - H12 A9~~
~~F16 - G12 A10~~ ~~F17 - H13 A11~~
~~F18 - G13 A12~~ ~~F19 - H14 A13~~
~~F14 - G14 A14~~ ~~F20 - H15 A15~~

✓H17 - GND

✓G31 - VCC ✓H31 - GND ✓G40 - G41 ✓GND - G40

✓G39 - G5 A0 ✓H32 - G11 A8
✓G38 - H6 A1 ✓H33 - H12 A9
✓G37 - G6 A2 ✓H34 - G12 A10
✓G36 - H7 A3 ✓H35 - H13 A11
✓G35 - G7 A4 ✓H36 - G13 A12
✓G34 - H10 A5 ✓H37 - H14 A13
✓G33 - G10 A6 ✓H38 - G14 A14
✓G32 - H11 A7 ✓H39 - H15 A15

Address to RAM Interconnect

✓G45 - G1 D0 ✓H41 - G3 D4
✓G44 - H2 D1 ✓H42 - H4 D5
✓G43 - G2 D2 ✓H43 - G4 D6
✓G42 - H3 D3 ✓H44 - H5 D7

✓H45 - B35 CLK ✓H47 - (F)D41 WRAM

✓G50 - VCC ✓H50 - GND

VGA board modifications:

0000 - 3fff Rom
4000 - bfff RAm

✓A14 - disconnect from ground, add to interconnect
✓$\overline{CE}$, $\overline{OE}$ - disconnect from interconnect
✓add Rom
✓bits A15+A14+$\overline{WE}$ → $\overline{OE}$+$\overline{WE}$

A15	A14	$\overline{WE}$	Rom $\overline{OE}$	RAm $\overline{OE}$	RAm $\overline{WE}$
0	0	0	1	1	1
0	1	0	1	1	0
1	0	0	1	1	0
1	1	0	1	1	1
0	0	1	0	1	1
0	1	1	1	0	1
1	0	1	1	0	1
1	1	1	1	1	1

Data + Address Bus Probes,

Group A, J1 - notch up, align right (from top) - Pod (A1)

✓1 - GND
✓2 - CLK - C11
✓3 - (14) - RST - C1
✓4 - (12) PCRSEL0 - G17
✓5 - (10) $\overline{\text{NEWOP}}$ - G9
✓6 - (8) $\overline{\text{WSTAT}}$ - G15
✓7 - (6) D6 - G4
✓8 - (4) D4 - G3
✓9 - (2) D2 - G2
✓10 - (0) D0 - G1
11 -
12 -
13 -

✓14 - GND
✓15 - (15) - CLK - Z
✓16 - (13) PCRSEL1 - G18
✓17 - (11) $\overline{\text{SPADDR}}$ - (C) A33
✓18 - (9) IRQOP - G20
✓19 - (7) D7 - H5
✓20 - (5) D5 - H4
✓21 - (3) D3 - H3
✓22 - (1) D1 - H2
✓23 - GND
24 -
25 -
26 -

Data bus	Pod bit
CLK	15
RST	14
PCSELR	13 - 12
$\overline{\text{NEWOP}}$	10
IRQOP	9
$\overline{\text{WSTAT}}$	8
$\overline{\text{SPADDR}}$	11

Group A, J2 - notch up, align right (from top) - Pod (A3)

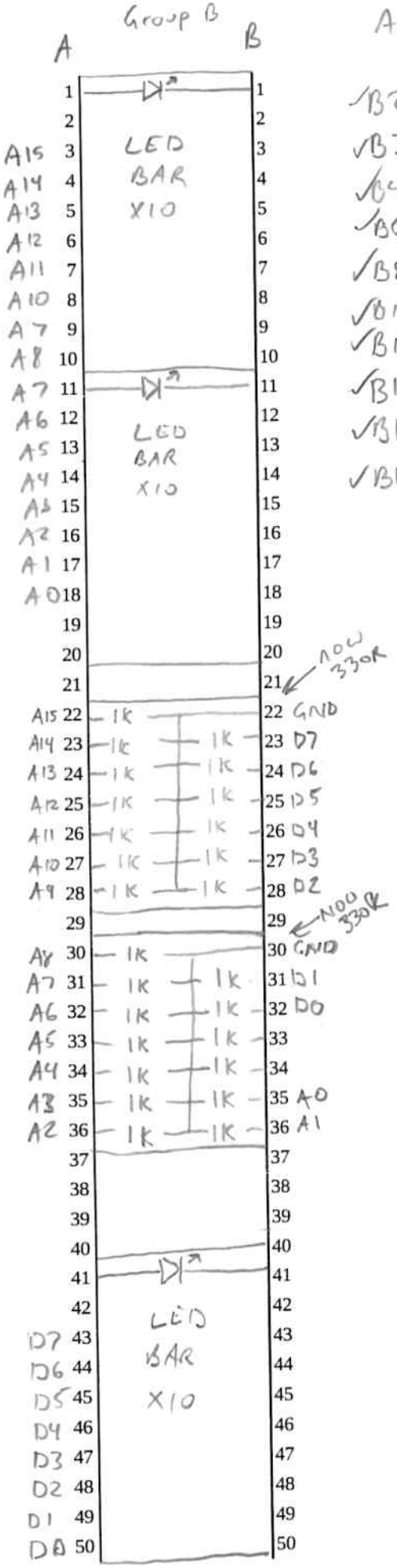
✓1 - GND
2 - CLK - NC
✓3 - (14) A14 - G14
✓4 - (12) A12 - G13
✓5 - (10) A10 - G12
✓6 - (8) A8 - G11
✓7 - (6) A6 - G10
✓8 - (4) A4 - G7
✓9 - (2) A2 - G6
✓10 - (0) A0 - G5
11 -
12 -
13 -

✓14 - GND
✓15 - (15) A15 - H15
✓16 - (13) A13 - H14
✓17 - (11) A11 - H13
✓18 - (9) A9 - H12
✓19 - (7) A7 - H11
✓20 - (5) A5 - H10
✓21 - (3) A3 - H7
✓22 - (1) A1 - H6
✓23 - GND
24 -
25 -
26 -

Address bus	Pod bit
	15 - 0

Address / Data Visualisation

Group B



✓B22 - GND

✓B30 - GND

✓B3 - A22 A15

✓B4 - A23 A14

✓B5 - A24 A13

✓B6 - A25 A12

✓B7 - A26 A11

✓B8 - A27 A10

✓B9 - A28 A9

✓B10 - A30 A8

✓B11 - A31 A7

✓B12 - A32 A6

✓B13 - A33 A5

✓B14 - A34 A4

✓B15 - A35 A3

✓B16 - A36 A2

✓B17 - B36 A1

✓B18 - B35 A0

✓B43 - B23 D7

✓B44 - B24 D6

✓B45 - B25 D5

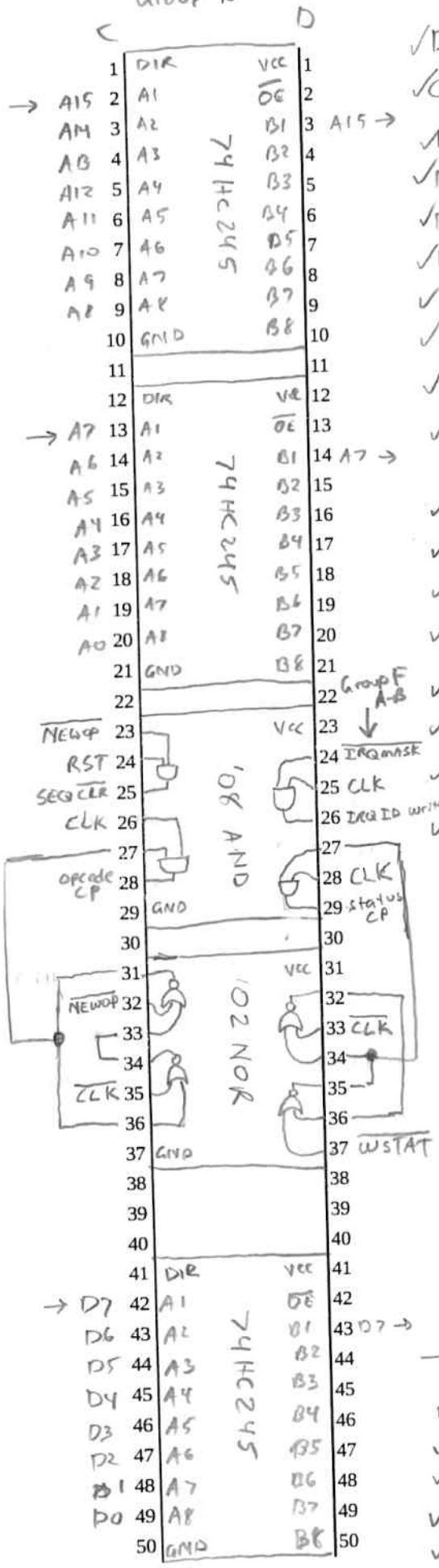
✓B46 - B26 D4

✓B47 - B27 D3

✓B48 - B28 D2

✓B49 - B31 D1

✓B50 - B32 D0



LED Drivers

✓D1 - VCC ✓C1 - D1 ✓D12 - VCC ✓C12 - D12
✓C10 - GND ✓GND - D2 ✓C21 - GND ✓GND - D13

✓D3 - A3 A15 ✓D4 - A4 A14
✓D5 - A5 A13 ✓D6 - A6 A12
✓D7 - A7 A11 ✓D8 - A8 A10
✓D9 - A9 A9 ✓D10 - A10 A8
✓D14 - A11 A7 ✓D15 - A12 A6
✓D16 - A13 A5 ✓D17 - A14 A4
✓D18 - A15 A3 ✓D19 - A16 A2
✓D20 - A17 A1 ✓D21 - A18 A0

Group A
✓H15 - C2 A15 ✓G14 - C3 A14
✓H14 - C4 A13 ✓G13 - C5 A12
✓H13 - C6 A11 ✓G12 - C7 A10
✓H12 - C8 A9 ✓G11 - C9 A8
✓H11 - C13 A7 ✓G10 - C14 A6
✓H10 - C15 A5 ✓G7 - C16 A4
✓H7 - C17 A3 ✓G6 - C18 A2
✓H6 - C19 A1 ✓G5 - C20 A0

✓D41 - VCC ✓C41 - D41 ✓C50 - GND ✓GND - D42

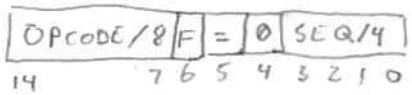
✓D43 - A43 D7 ✓D44 - A44 D6
✓D45 - A45 D5 ✓D46 - A46 D4
✓D47 - A47 D3 ✓D48 - A48 D2
✓D49 - A49 D1 ✓D50 - A50 D0

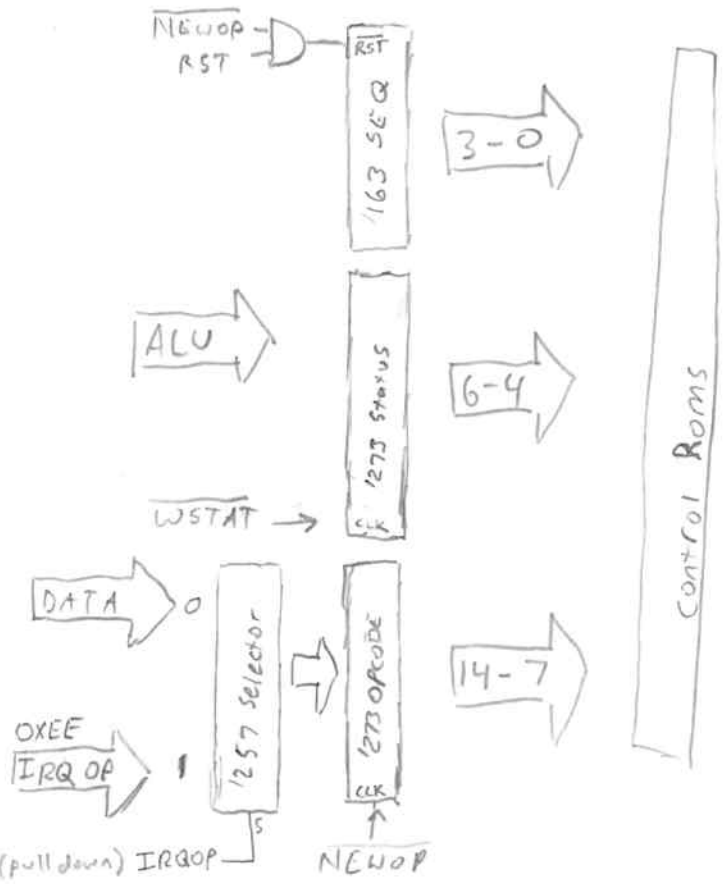
Group A
✓H44 - C42 D7 ✓H43 - C43 D6
✓H42 - C44 D5 ✓H41 - C45 D4
✓G42 - C46 D3 ✓G43 - C47 D2
✓G44 - C48 D1 ✓G45 - C49 D0

AND + NOR for instruction register

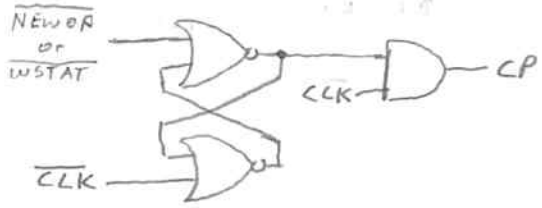
✓D23 - VCC ✓C29 - GND ✓D31 - VCC ✓C37 - GND
✓C32 - (A)G9 NEWOP pullup ✓C31 - C27 ✓D34 - D27
✓D37 - (A)G15 WSTAT pullup ✓C26 - D28 ✓C26 - (A)H45 CLK
✓C33 - C34 ✓C36 - C31 ✓C32 - C23 ✓E21 - C24 RST
✓D34 - D35 ✓D36 - D32
✓C35 - D33 ✓C35 - (A)B37 CLK

Instruction + Sequence Reg.





Register clock pulse debounce

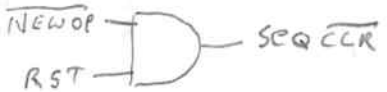


when input is low, CP matches CLK, on edges. when input is high, CP is low

Sequence Counter clear

low on reset

NEWOP	RST	SEQ CLR
0	0	0
0	1	0
1	0	0
1	1	1



✡ '163 is synch. clear, so will clear on rising clock edge

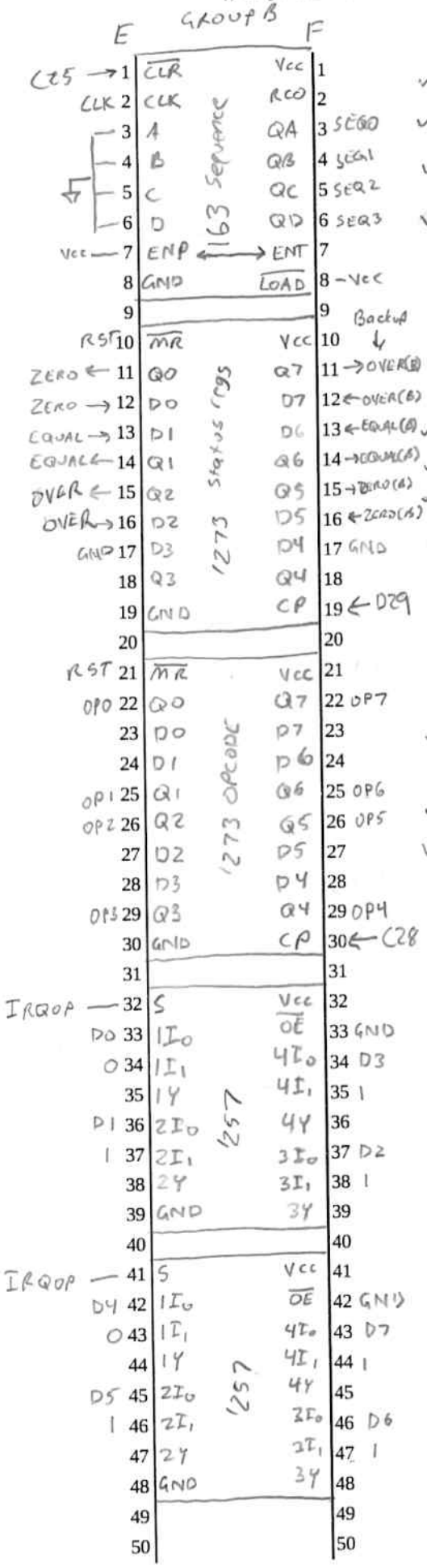
NEWOP - pulled high. when low, load new instruction from data bus (IRQOP=0) or fixed 0xEE (IRQOP=1), and reset sequence counter, on rising clock edge

WSTAT - pulled high. when low, load F, =, 0 flags from ALU into status register on rising clock edge

✡ First instruction in ROM will be ignored, NOP (0x00) will always execute first, since OPCODE is cleared on reset. consider adding AND gate to NEWOP + RST so actual first instruction is loaded

bit 6 of opcode stuck high?

Instruction + Sequence register



Sequence counter

always count

✓F1 - Vcc ✓E8 - GND ✓E7 - F7 ✓F7 - Vcc

✓E3 - E4 ✓E5 - E6 ✓E4 - E5 ✓E6 - GND

✓F8 - Vcc ✓D28 - E2 CLK

✓C25 - E1 SEQ $\overline{CLR}$

Status register

✓F10 - Vcc ✓E19 - GND ✓(A)C1 - E10 RST

✓D29 - F19 status reg clk

✓E12 - (A)G22 ZERO pull down

✓E13 - (A)G23 EQUAL pull down

✓E16 - (A)G21 EQAC pull down

✓E17 - GND

✓F17 - GND

See (F) A-B for status backup muxes and status inputs

Opcode register

✓F21 - Vcc ✓E30 - GND ✓E10 - E21 RST

✓C28 - F30 opcode reg clk

✓E23 - E35 0

✓E24 - E38 1

✓E27 - F39 2

✓E28 - F36 3

✓F28 - E44 4

✓F27 - E47 5

✓F24 - F48 6

✓F23 - F45 7

} from 1257s

Opcode select

✓F32 - Vcc ✓F41 - Vcc ✓E39 - GND ✓E48 - GND

✓E32 - E41 ✓E32 - (A)G20 IRQOP pull down

✓F33 - GND ✓F42 - GND

IRQ opcode: 0xEE 1110 1110

✓E34 - GND ✓E37 - F38 ✓F38 - F35 ✓F35 - Vcc

✓E43 - GND ✓E46 - F47 ✓F47 - F44 ✓F44 - Vcc

Data bus

✓C49 - E33 D0

✓C48 - E36 D1

✓C47 - F37 D2

✓C46 - F34 D3

✓C45 - E42 D4

✓C44 - E45 D5

✓C43 - F46 D6

✓C42 - F43 D7

Group B, J1 - notch up, align right - pod (A4) Instruction

✓1 - GND
2 - CLK - NC
✓3 - (14) - F22 OP7
✓4 - (12) - F26 OP5
✓5 - (10) - E29 OP3
✓6 - (8) - E25 OP1
✓7 - (6) - E15 FLAG
✓8 - (4) - E11 ZERO
✓9 - (2) - F5 SEQ2
✓10 - (0) - F3 SEQ0
11
12
13

✓14 - GND
15 - (15) - J41 $\overline{\text{ACUD}}$
✓16 - (13) - F25 OP6
✓17 - (11) - F29 OP4
✓18 - (9) - E26 OP2
✓19 - (7) - E22 OP0
✓20 - (5) - E14 EQUAL
✓21 - (3) - F6 SEQ3
✓22 - (1) - F4 SEQ1
✓23 - GND
24
25
26

Opcode
flags
sequence

Pod bits
14-7
6-4
3-0

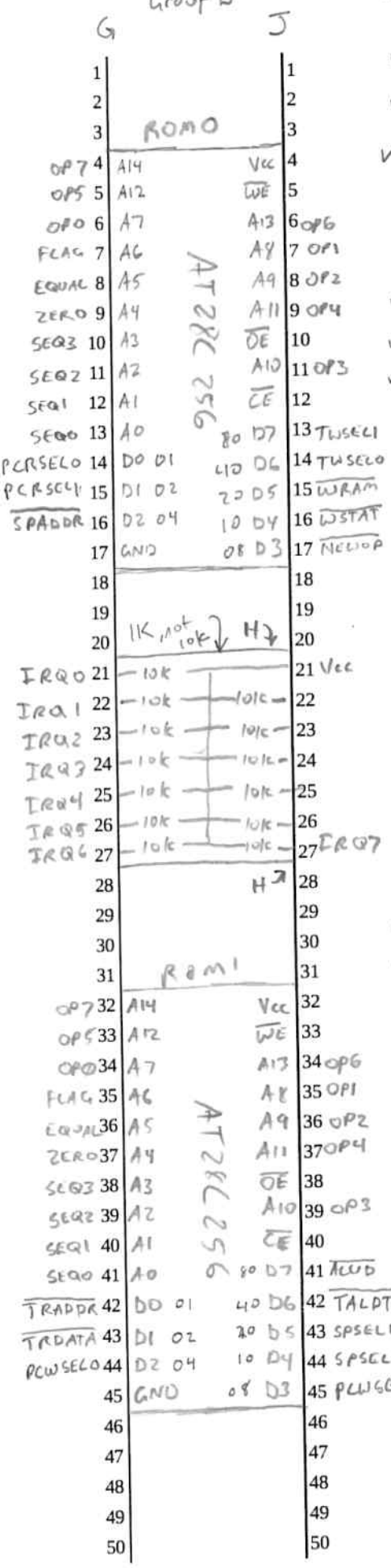
Group B, J2 - pod (A5) - PC

All remote pins group A

✓1 - GND
2 - CLK - NC
✓3 - (14) D32
✓4 - (12) D30
✓5 - (10) D23
✓6 - (8) D21
✓7 - (6) D14
✓8 - (4) D12
✓9 - (2) D5
✓10 - (0) D3
11
12
13

✓14 - GND
✓15 - (15) D33
✓16 - (13) D31
✓17 - (11) D24
✓18 - (9) D22
✓19 - (7) D15
✓20 - (5) D13
✓21 - (3) D6
✓22 - (1) D4
✓23 - GND
24
25
26

Control ROMs Bank 1



✓J4 - Vcc ✓G17 - GND ✓J32 - Vcc ✓G45 - GND
✓J5 - Vcc ✓J10 - GND ✓J12 - GND
✓J33 - Vcc ✓J38 - GND ✓J40 - GND

Address banding

✓G4 - G32 ✓G5 - G33 ✓G6 - G34
✓G7 - G35 ✓G8 - G36 ✓G9 - G37
✓G10 - G38 ✓G11 - G39 ✓G12 - G40
✓G13 - G41 ✓J6 - J34 ✓J7 - J35
✓J8 - J36 ✓J9 - J37 ✓J11 - J39

Instruction mapping

✓F3 - G13 SEQ0 ✓F4 - G12 SEQ1
✓F5 - G11 SEQ2 ✓F6 - G10 SEQ3
✓E11 - G9 ZERO ✓E14 - G8 EQUAL ✓E15 - G7 FLAG

✓E22 - G34 OP0 ✓E25 - J35 OP1
✓E26 - J36 OP2 ✓E29 - J39 OP3
✓F29 - J37 OP4 ✓F26 - G33 OP5
✓F25 - J34 OP6 ✓F22 - G32 OP7

ROM0 signals

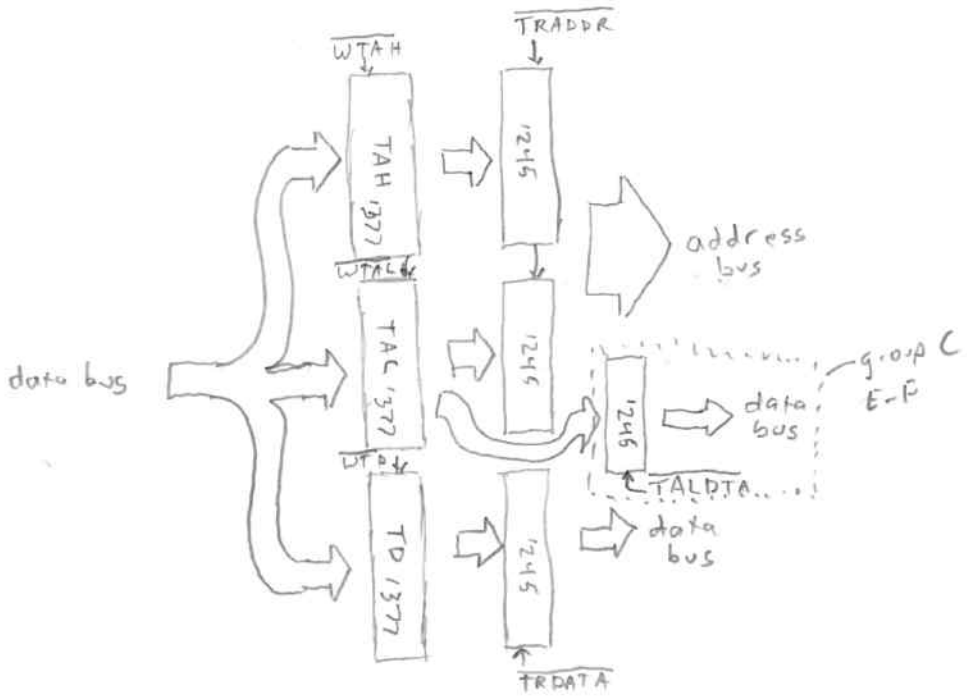
✓G14 D0 PCRSEL0 (A) G17 ✓J16 D4 WSTAT (A) G15
✓G15 D1 PCRSEL1 (A) G18 ✓J15 D5 WRAM (A) H46
G16 D2 SPADDR ✓J14 D6 TWSEL0 (C) A24
✓J17 D3 NEWOP (A) G9 ✓J13 D7 TWSEL1 (C) A25

ROM1 signals

✓G42 D0 TRADDR (C) B36 ✓J44 D4 SPSEL0
✓G43 D1 TR DATA (C) B37 ✓J43 D5 SPSEL1
✓G44 D2 PCWSEL0 (A) H20 ✓J42 D6 TALDTA
✓J45 D3 PCWSEL1 (A) H21 ✓J41 D7 ALUD

✓bypass caps @ H8 H40

Transfer Registers



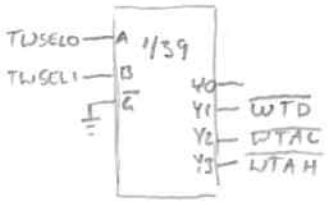
TWSEL - transfer write

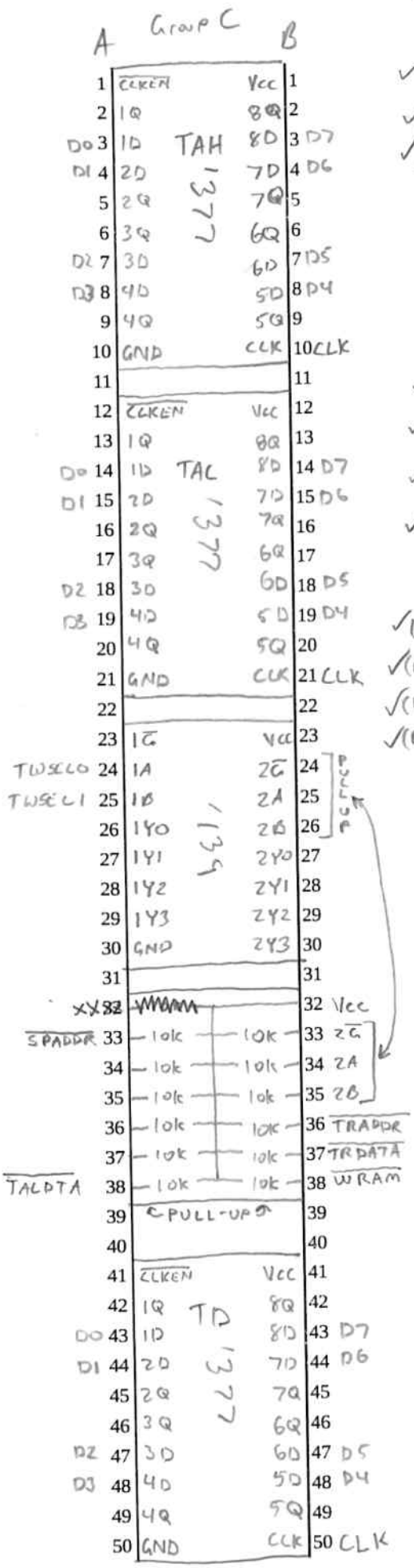
0 0	none
0 1	TD
1 0	TAL
1 1	TAH

$\overline{TRADDR}$ - TAL + TAH to addr bus

$\overline{TRDATA}$ - TD to data bus

$\overline{TALDTA}$ - TAL to data bus





Transfer registers

✓B1 - Vcc ✓A10 - GND ✓B12 - Vcc ✓A21 - GND
✓B23 - Vcc ✓A30 - GND ✓B32 - Vcc ✓B41 - Vcc
✓A50 - GND ✓B24 - B33 ✓B25 - B34 ✓B26 - B35

'439 2nd unit pull-up

Clocks

✓C34 - B10 ✓B10 - B21 ✓C34 - B50

Input ganging

✓A3 - A14 ✓A4 - A15 ✓A7 - A18 ✓A8 - A19
✓B3 - B14 ✓B4 - B15 ✓B7 - B18 ✓B8 - B19

✓A14 - A43 ✓A15 - A44 ✓A18 - A47 ✓A19 - A48
✓B14 - B43 ✓B15 - B44 ✓B18 - B47 ✓B19 - B48

Data bus input

✓(A) E33 - A43 D0
✓(A) E36 - A44 D1
✓(A) F37 - A47 D2
✓(A) F34 - A48 D3

✓(B) E42 - B48 D4
✓(B) E45 - B47 D5
✓(B) F46 - B44 D6
✓(B) F43 - B43 D7

TWSEL signals

✓A23 - GND
✓A24 - (B) J14 (control ROM) ✓A25 - (B) J13
✓A27 - A41 ✓A28 - A12 ✓A29 - A1
✓(A) H18 - (B) J14 ✓(A) H19 - (B) J13

TWSEL0 ← pull-downs → TWSEL1

WRAM pull-up

✓B38 - (B) J15

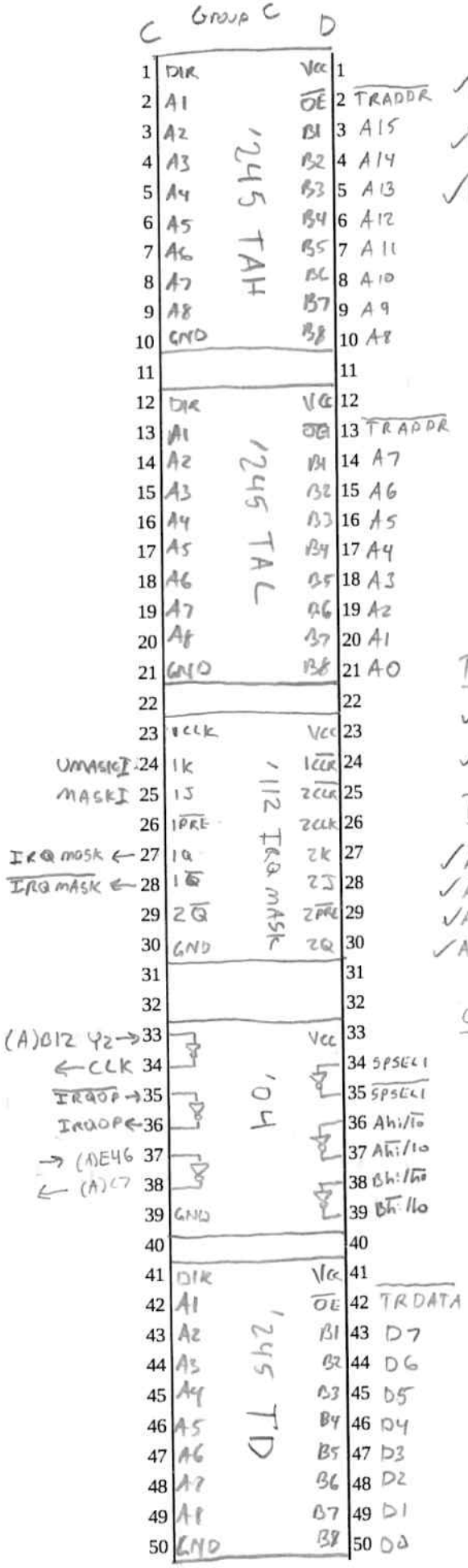
Group C, J1 - pod (A2) - more control signals

- ✓ 1 - GND
- 2 - CLK - NC
- 3 - (14)
- ✓✓ 4 - (12) $\overline{\text{BADDR}}$ - G15
- ✓✓ 5 - (10) $\overline{\text{ALUOPW}}$ - G16
- ✓✓ 6 - (8) SPSEL1
- stuck — ✓✓ 7 - (6) PCWSEL1 - H21(A)
- ✓✓ 8 - (4) $\overline{\text{TRDATA}}$ D42
- ✓✓ 9 - (2) TWSEL1 A25
- ✓✓ 10 - (0) $\overline{\text{WRAM}}$ (B) J15
- 11
- 12
- 13
- ✓ 14 - GND
- 15 - (15)
- 16 - (13)
- ✓✓ 17 - (11) $\overline{\text{AADDR}}$ - G14
- ✓✓ 18 - (9) $\overline{\text{TALDTA}}$ - F42
- ✓✓ 19 - (7) SPSEL0
- ✓✓ 20 - (5) PCWSEL0 - H20(A)
- ✓✓ 21 - (3) $\overline{\text{TRADDR}}$ D2
- ✓✓ 22 - (1) TWSEL0 A24
- ✓ 23 - GND
- 24
- 25
- 26

Group C, J2 - pod (A6)

Stack Pointer (7-0)
ROM2/3 ctrl signals (15-8)

- ✓ 1 - GND
- 2 - CLK - NC
- ✓✓ 3 - (14) $\overline{\text{BWH}}$ - J14
- ✓✓ 4 - (12) $\overline{\text{AWH}}$ - J16
- ✓ 5 - (10) CDSEL2 - G44
- ✓ 6 - (8) CDSEL0 - G42
- ✓ 7 - (6) SP6 - E14
- ✓ 8 - (4) SP4 - E16
- ✓ 9 - (2) SP2 - E18
- ✓ 10 - (0) SP0 - E20
- 11
- 12
- 13
- ✓ 14 - GND
- ✓✓ 15 - (15) $\overline{\text{BWL}}$ - J13
- ✓✓ 16 - (13) $\overline{\text{AWL}}$ - J15
- ✓ 17 - (11) CDSEL3 - J45
- ✓ 18 - (9) CDSEL1 - G43
- ✓ 19 - (7) SP7 - E13
- ✓ 20 - (5) SP5 - E15
- ✓ 21 - (3) SP3 - E17
- ✓ 22 - (1) SP1 - E19
- 23 - (GND)
- 24
- 25
- 26



Transfer register outputs

✓D1 - Vcc ✓C1 - Vcc ✓C10 - GND
✓D12 - Vcc ✓C12 - Vcc ✓C21 - GND
✓D41 - Vcc ✓C41 - Vcc ✓C50 - GND

OE hookups

✓D2 - D13
✓D13 - B36 (TRADDR (B)G42) pull up
✓D42 - B37 (TRDATA (6)G43) pull up

TAH inputs from registers

✓B2 - C2 ✓B5 - C3 ✓B6 - C4 ✓B9 - C5
✓A9 - C6 ✓A6 - C7 ✓A5 - C8 ✓A2 - C9

TAL inputs from register

✓B13 - C13 ✓B16 - C14 ✓B17 - C15 ✓B20 - C16
✓A20 - C17 ✓A17 - C18 ✓A16 - C19 ✓A13 - C20

TD inputs from register

✓B42 - C42 ✓B45 - C43 ✓B46 - C44 ✓B49 - C45
✓A49 - C46 ✓A46 - C47 ✓A45 - C48 ✓A42 - C49

TA to addr bus

Group B		Group A		Group A		Group B	
✓A15	C2 - D3	✓A11	C6 - D7	✓A7	C13 - D14	✓A3	C17 - D18
✓A14	C3 - D4	✓A10	C7 - D8	✓A6	C14 - D15	✓A2	C18 - D19
✓A13	C4 - D5	✓A9	C8 - D9	✓A5	C15 - D16	✓A1	C19 - D20
✓A12	C5 - D6	✓A8	C9 - D10	✓A4	C16 - D17	✓A0	C20 - D21

Clock repeater

✓D33 - Vcc ✓C39 - GND ✓(A)B12 - C33
✓(A)B13 - C35

IRQ mask ☆ negative edge trigger

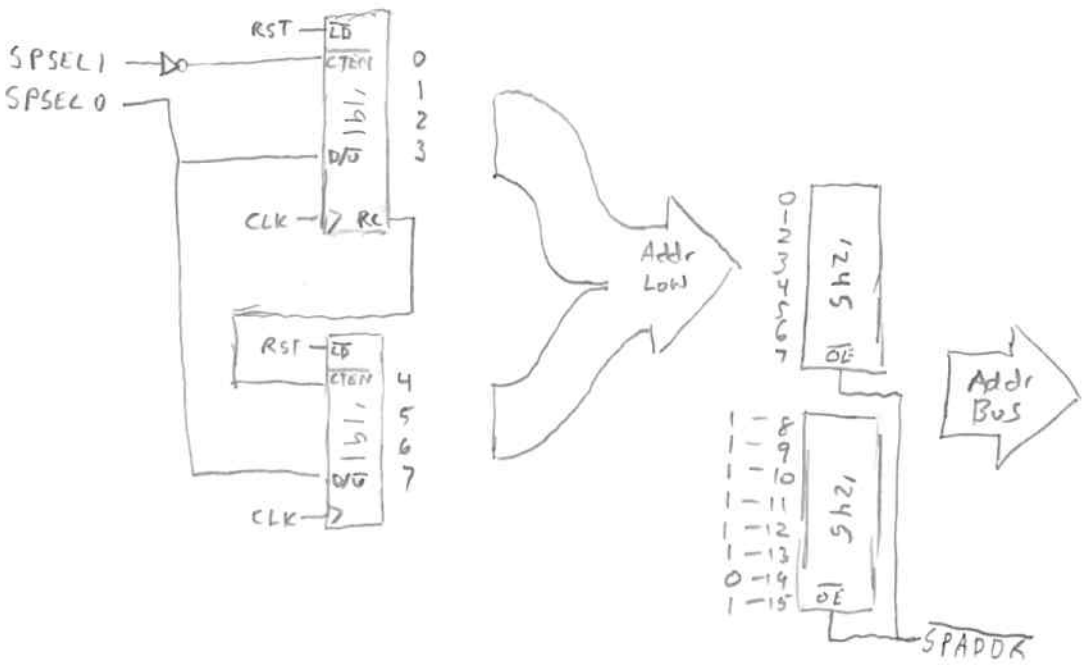
✓D23 - Vcc ✓C30 - GND ✓(A)B37 - C23 ✓C23 - D26 CLK
✓D25 - D23 ✓D29 - D25 2 CLR/PRE tie up
✓D28 - C30 ✓D27 - D24 2 J/K tie down
✓J44 - C25 MASKI ✓J43 - C24 UMASKI
✓(B)E21 - C26 RST - sets on reset
✓D24 - Vcc 1CLR tie up

TD to data bus

✓D7	B43 - D43	✓D3	A48 - D47
✓D6	B44 - D44	✓D2	A47 - D48
✓D5	B47 - D45	✓D1	A44 - D49
✓D4	B48 - D46	✓D0	A43 - D50

Stack Pointer design

Stack address: 0xbf00 - bfff



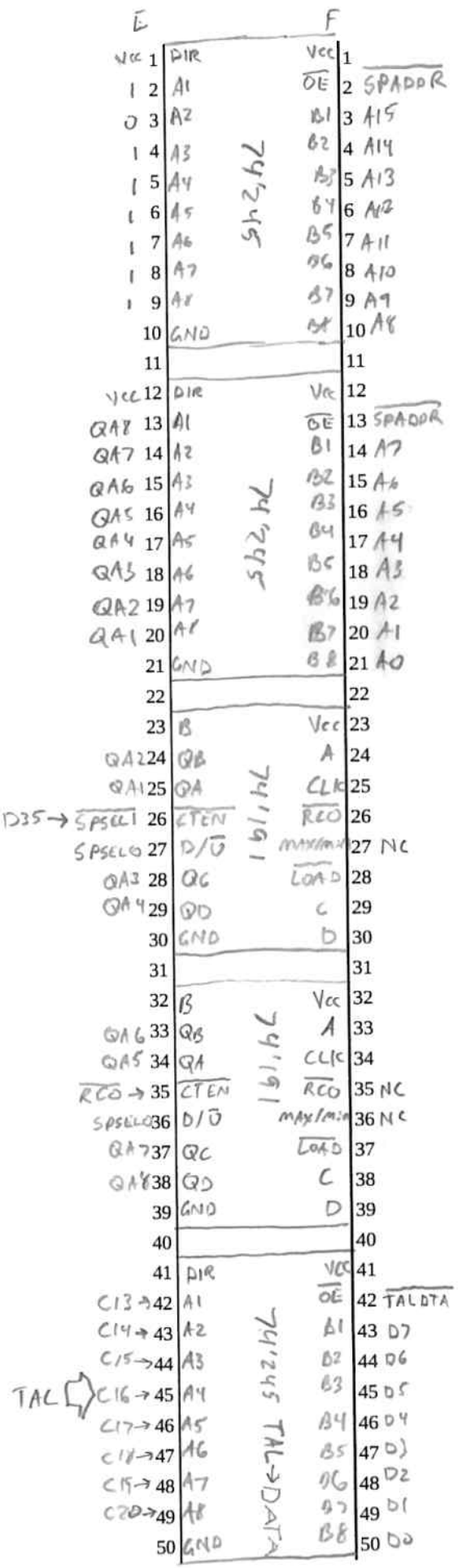
- Low pulse on RST loads all zeroes (reset)
- SPSEL:
  - 00 do nothing
  - 01 do nothing
  - 10 count up
  - 11 count down
- $\overline{SPADDR}$:
  - 0 SP to addr bus
  - 1 do nothing

TAL to data bus

$\overline{TALDTA}$

✓F41-Vcc	✓E50-GND	✓E41-Vcc	✓F42-A38 (pull-up)	✓A38-(B)J42 (Rom)
✓C13-E42	✓C14-E43	✓C15-E44	✓C16-E45	
✓C17-E46	✓C18-E47	✓C19-E48	✓C20-E49	
✓D43-F43	✓D44-F44	✓D45-F45	✓D46-F46	
✓D47-F47	✓D48-F48	✓D49-F49	✓D50-F50	

Group C



Stack pointer

✓F1-Vcc ✓E1-Vcc ✓E10-GND
✓F12-Vcc ✓E12-Vcc ✓E21-GND
✓F23-Vcc ✓E30-GND ✓F32-Vcc ✓E39-GND

Addr-H: ←0xBF
✓E1-E2 ✓E2-E4 ✓E4-E5 ✓E5-E6
✓E6-E7 ✓E7-E8 ✓E8-E9 ✓E3-GND

Addr-Lo ← Count
✓E13-E38 ✓E14-E37 ✓E15-E33
✓E16-E34 ✓E17-E29 ✓E18-E28
✓E19-E24 ✓E20-E25

Address Bus
✓D3-F3 ✓D4-F4 ✓D5-F5 ✓D6-F6
✓D7-F7 ✓D8-F8 ✓D9-F9 ✓D10-F10
✓D14-F14 ✓D15-F15 ✓D16-F16 ✓D17-F17
✓D18-F18 ✓D19-F19 ✓D20-F20 ✓D21-F21

Counter preload to zero
✓E23-F24 ✓F24-GND ✓F29-F30 ✓F30-GND
✓E32-F33 ✓F33-GND ✓F38-F39 ✓F39-GND

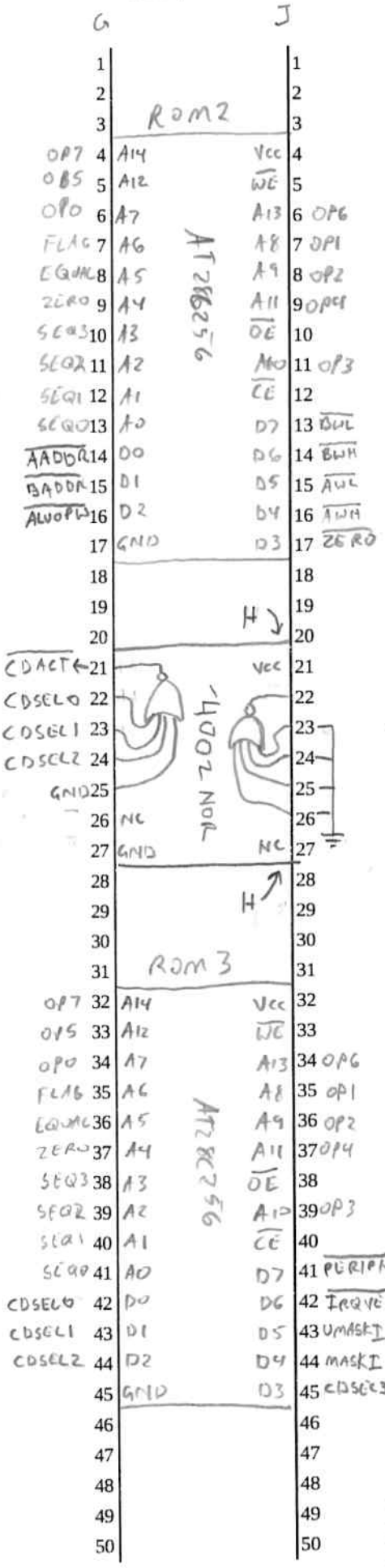
SPADDR
✓F2-F13 ✓F13-A33 (pull up)
✓A33-(A31-17) probe ✓(B)G16-A33 Rom

Clock
✓F25-F34 ✓C34-F34

COUNT LOGIC
✓F26-E35 RCO→CTEN
✓E27-E36 D/Ū ← SPSEL0
✓(A)H22-E27 SPSEL0 pulldown
✓(A)H23-D34 SPSEL1 pulldown ✓D35-E26 SPSEL1→CTEN
✓(B)J44-(A)H22 SPSEL0 ROM ✓(B)J44-(C)J1-19 > probe
✓(B)J43-(A)H23 SPSEL1 ROM ✓(B)J43-(C)J1-6

RST
✓F28-F37 ✓(B)E21-F28

Group C



Control ROMs Bank 2

✓J4 - Vcc ✓G17 - GND ✓J32 - Vcc ✓G45 - GND
✓J5 - Vcc ✓J10 - GND ✓J12 - GND
✓J33 - Vcc ✓J38 - GND ✓J40 - GND

Address bonding

✓G4 - G32 ✓G5 - G33 ✓G6 - G34
✓G7 - G35 ✓G8 - G36 ✓G9 - G37
✓G10 - G38 ✓G11 - G39 ✓G12 - G40
✓G13 - G41 ✓J6 - J34 ✓J7 - J35
✓J8 - J36 ✓J9 - J37 ✓J11 - J39

Instruction mapping from group B

All ties identical B → C

✓G32 ✓G33 ✓G34
✓G35 ✓G36 ✓G37
✓G38 ✓G39 ✓G40
✓G41 ✓J34 ✓J35
✓J36 ✓J37 ✓J39

ROM2 signals

Group C, J2 probe

✓G14 D0 $\overline{\text{AADDR}}$ (D)B13
✓G15 D1 $\overline{\text{BADDR}}$ (D)F13
✓G16 D2 $\overline{\text{ALUOPW}}$ (E)A41
J17 D3 $\overline{\text{STRES}}$

✓J16 D4 $\overline{\text{AWH}}$ (D)C1
✓J15 D5 $\overline{\text{AWL}}$ (D)C12
✓J14 D6 $\overline{\text{BWH}}$ (D)C23
✓J13 D7 $\overline{\text{BWL}}$ (D)C34

ROM3 signals

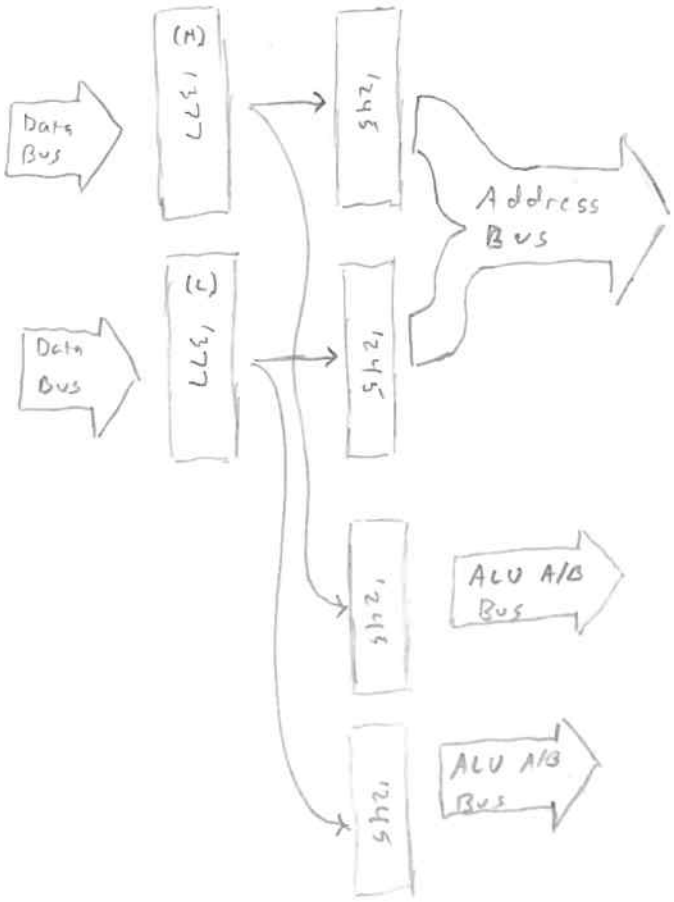
✓G42 D0 CDSEL0
✓G43 D1 CDSEL1
✓G44 D2 CDSEL2
✓J45 D3 CDSEL3

✓J44 D4 MASKINT
✓J43 D5 UMASKINT
J42 D6 CONSEL0
J41 D7 CONSEL1

NOR gates (for CDSEL Group E)

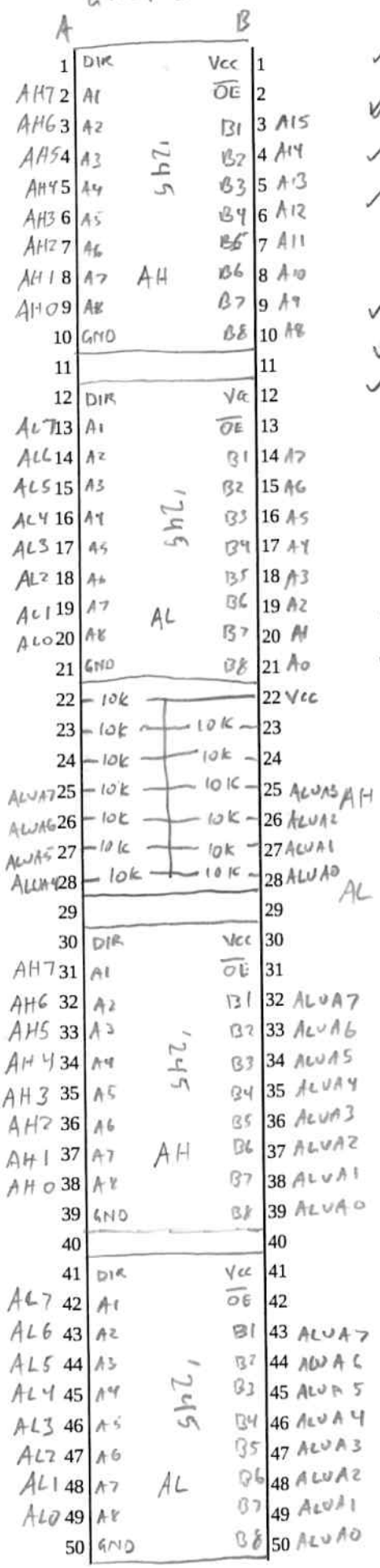
✓H21 - Vcc ✓G27 - GND
✓H23 - H24 ✓H25 - H26 ✓H24 - H25 ✓H26 - GND
✓G25 - GND
G21 = $\overline{\text{CDACT}}$
✓G42 - G22 CDSEL0
✓G43 - G23 CDSEL1
✓G44 - G24 CDSEL2

Registers A + B



$\overline{(A/B)WH}$ write high

$\overline{(A/B)WL}$ write Low



Register A outputs

✓B1 - Vcc ✓A1 - B1 ✓A10 - GND
✓B12 - Vcc ✓A12 - B12 ✓A21 - GND
✓B30 - Vcc ✓A30 - B30 ✓A39 - GND
✓B41 - Vcc ✓A41 - B41 ✓A50 - GND

Output enables

✓(C)G14 — B13 ✓B2 - B13 Addr $\overline{OE}$
✓(E) B45 — B42 ✓(C)D37 - B31 ALU $\overline{OE}$
✓(E) B45 - (C)D36 inv

Input Ganging

✓A2 - A31 ✓A3 - A32 ✓A4 - A33
✓A5 - A34 ✓A6 - A35 ✓A7 - A36
✓A8 - A37 ✓A9 - A38 ✓A13 - A42
✓A14 - A43 ✓A15 - A44 ✓A16 - A45
✓A17 - A46 ✓A18 - A47 ✓A19 - A48 ✓A20 - A49

Inputs

AH: ✓A2 - D2, ✓A3 - D5, ✓A4 - D6, ✓A5 - D9, ✓A6 - C9, ✓A7 - C6, ✓A8 - C5, ✓A9 - C2
AL: ✓A13 - D13, ✓A14 - D16, ✓A15 - D17, ✓A16 - D20, ✓A17 - C20, ✓A18 - C17, ✓A19 - C16, ✓A20 - C13

Address bus

✓(C)F3 - B3 ✓F4 - B4 ✓F5 - B5 ✓F6 - B6
✓(C)F7 - B7 ✓F8 - B8 ✓F9 - B9 ✓F10 - B10
✓F14 - B14 ✓F15 - B15 ✓F16 - B16 ✓F17 - B17
✓F18 - B18 ✓F19 - B19 ✓F20 - B20 ✓F21 - B21

ALU Bus ganging

✓B32 - B43 ✓B33 - B44 ✓B34 - B45
✓B35 - B46 ✓B36 - B47 ✓B37 - B48
✓B38 - B49 ✓B39 - B50

ALU-A pull up

✓B39 - B28 ✓B38 - B27 ✓B37 - B26 ✓B36 - B25
✓B35 - A28 ✓B34 - A27 ✓B33 - A26 ✓B32 - A25
✓B22 - Vcc

Group D, J1 - Register A Pod C2

✓1 - GND
2 - CLK - NC
✓3 - (14) - D5
✓4 - (12) - D9
✓5 - (10) - C6
✓6 - (8) - C2
✓7 - (6) - D16
✓8 - (4) - D20
✓9 - (2) - C17
✓10 - (0) - C13
11
12
13

✓14 - GND
✓15 - (15) - D2
✓16 - (13) - D6
✓17 - (11) - C9
✓18 - (9) - C5
✓19 - (7) - D13
✓20 - (5) - D17
✓21 - (3) - C20
✓22 - (1) - C16
✓23 - GND
24
25
26

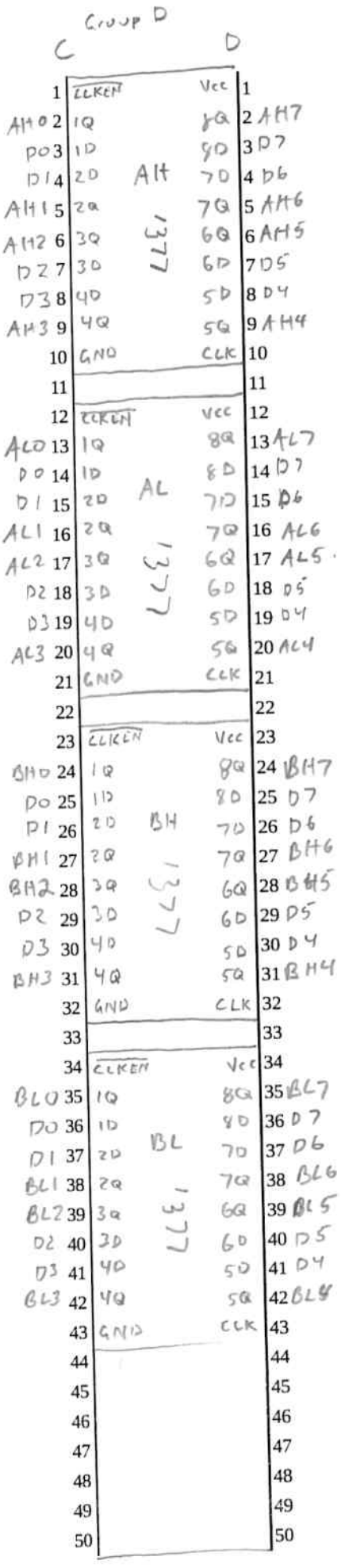
15-8 AH
7-0 AL

Group D, J2 - Register B Pod C3

✓1 - GND
2 - CLK - NC
✓3 - (14) - D27
✓4 - (12) - D31
✓5 - (10) - C28
✓6 - (8) - C24
✓7 - (6) - D38
✓8 - (4) - D42
✓9 - (2) - C39
✓10 - (0) - C35
11
12
13

✓14 - GND
✓15 - (15) - D24
✓16 - (13) - D28
✓17 - (11) - C31
✓18 - (9) - C27
✓19 - (7) - D35
✓20 - (5) - D39
✓21 - (3) - C42
✓22 - (1) - C38
✓23 - GND
24
25
26

15-8 BH
7-0 BL



Registers A/B

✓D1-Vcc ✓C10-GND ✓D12-Vcc ✓C21-GND
✓D23-Vcc ✓C32-GND ✓D34-Vcc ✓C43-GND

Data Bus Ganging

AH → AL
✓C3-C14 ✓C4-C15 ✓C7-C18 ✓C8-C19
✓D3-D14 ✓D4-D15 ✓D7-D18 ✓D8-D19

AL → BH
✓C14-C25 ✓C15-C26 ✓C18-C29 ✓C19-C30
✓D14-D25 ✓D15-D26 ✓D18-D29 ✓D19-D30

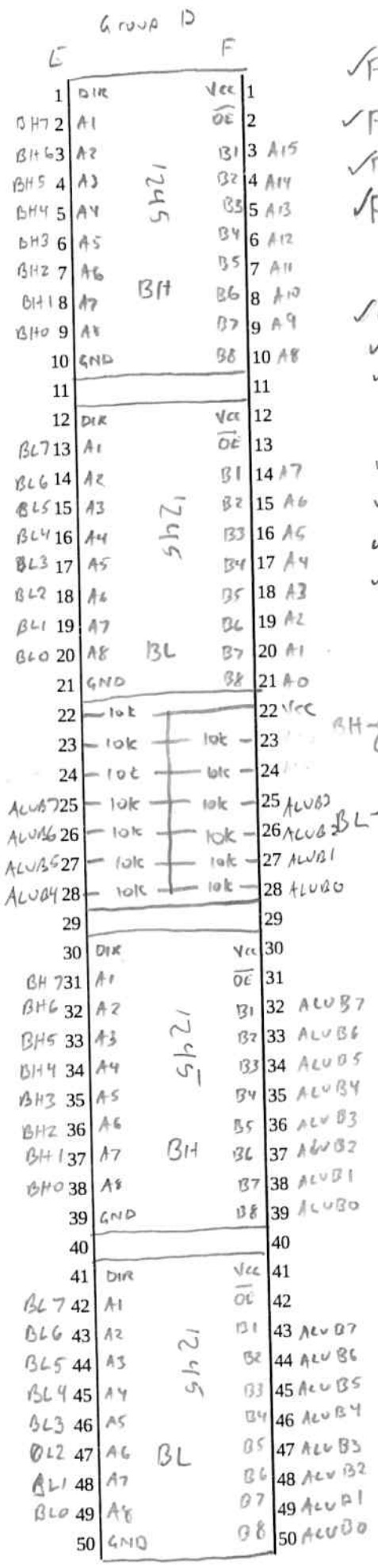
BH → BL
✓C25-C36 ✓C26-C37 ✓C29-C40 ✓C30-C41
✓D25-D36 ✓D26-D37 ✓D29-D40 ✓D30-D41

BL → Data Bus
✓(C)F43-D36 D7 ✓(C)F44-D37 D6
✓(C)F45-D40 D5 ✓(C)F46-D41 D4
✓(C)F47-C41 D3 ✓(C)F48-C40 D2
✓(C)F49-C37 D1 ✓(C)F50-C36 D0

Clock

✓(C)F34-D32 ✓D32-D43
✓D10-D21 ✓D21-D32

Write signals - see group C G-J



Register B outputs

✓F1-Vcc ✓E1-F1 ✓E10-GND
✓F12-Vcc ✓E12-F12 ✓E21-GND
✓F30-Vcc ✓E30-F30 ✓E39-GND
✓F41-Vcc ✓E41-F41 ✓E50-GND

Output enables

✓(C)G15 — F13 ✓F2-F13 Addr $\overline{OE}$
✓(E)B42 — F42 ✓(C)D39 — F31 ALU $\overline{OE}$
✓(E)B42 — (C)D38 inv

Input Ganging

✓E2-E31 ✓E3-E32 ✓E4-E33 ✓E5-E34
✓E6-E35 ✓E7-E36 ✓E8-E37 ✓E9-E38
✓E13-E42 ✓E14-E43 ✓E15-E44 ✓E16-E45
✓E17-E46 ✓E18-E47 ✓E19-E48 ✓E20-E49

Inputs from BH/BL outputs

BH: ✓E31-D24 ✓E32-D27 ✓E33-D28 ✓E34-D31 ✓E35-C31 ✓E36-C28 ✓E37-C27 ✓E38-C24
BL: ✓E42-D35 ✓E43-D38 ✓E44-D39 ✓E45-D42 ✓E46-C42 ✓E47-C39 ✓E48-C38 ✓E49-C35

Address Bus

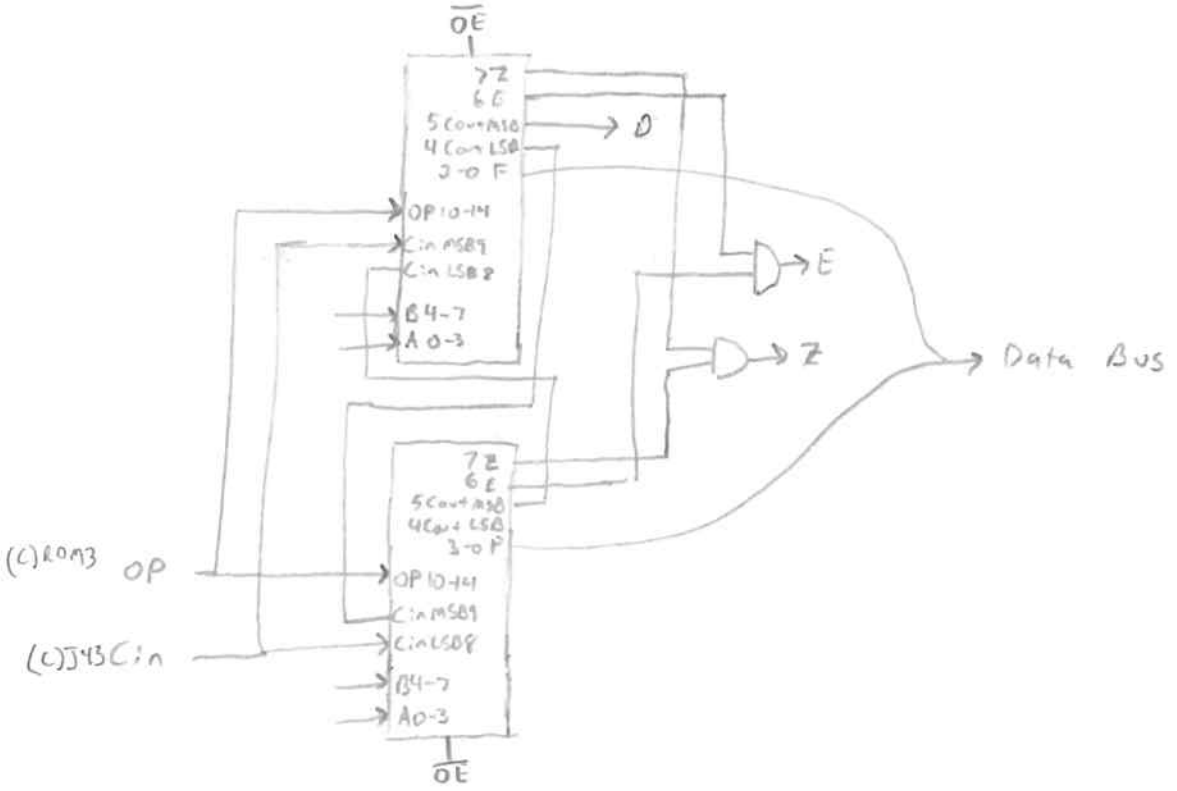
✓B3-F3 ✓B4-F4 ✓B5-F5 ✓B6-F6
✓B7-F7 ✓B8-F8 ✓B9-F9 ✓B10-F10
✓B14-F14 ✓B15-F15 ✓B16-F16 ✓B17-F17
✓B18-F18 ✓B19-F19 ✓B20-F20 ✓B21-F21

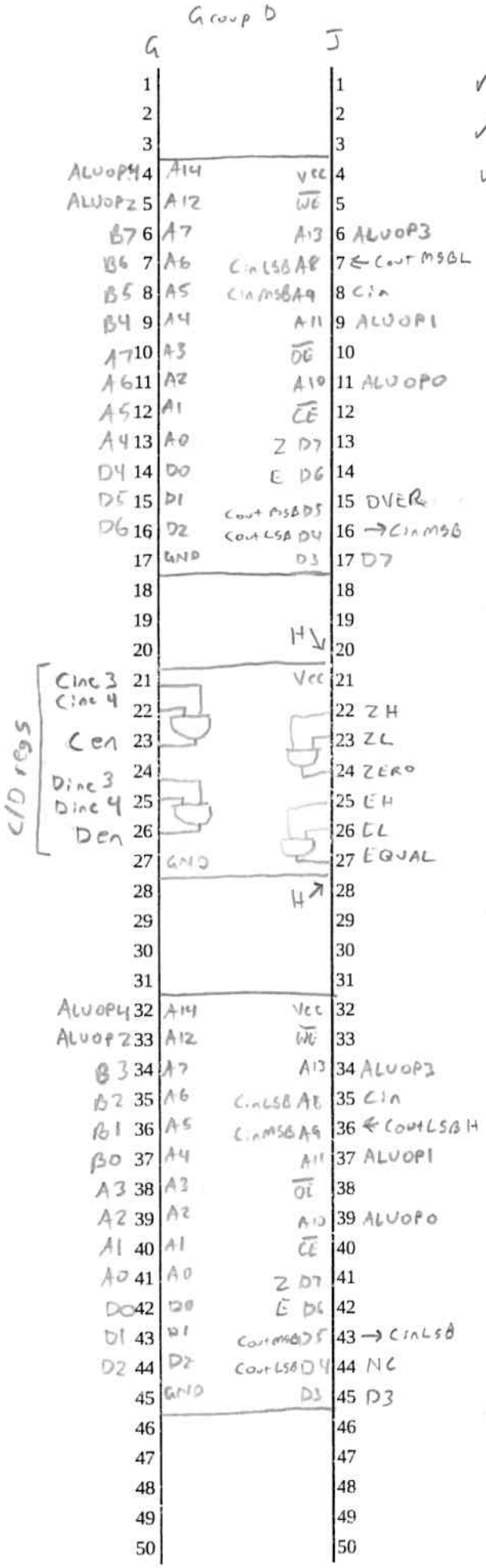
ALU Bus ganging

✓F32-F43 ✓F33-F44 ✓F34-F45
✓F35-F46 ✓F36-F47 ✓F37-F48
✓F38-F49 ✓F39-F50

ALU B pull-up

✓F39-F28 ✓F38-F27 ✓F37-F26 ✓F36-F25
✓F35-E26 ✓F34-E27 ✓F33-E28 ✓F32-E25
✓F22-Vcc


OE
7 Z
6 E
5 Cout MSB
4 Cout LSB
3-0 F
D
OP 10-14
Cin MSB
Cin LSB
B 4-7
A 0-3
E
Z
Data Bus
7 Z
6 E
5 Cout MSB
4 Cout LSB
3-0 F
(C)ROM3 OP
OP 10-14
Cin MSB
(C)J43 Cin
Cin LSB
B 4-7
A 0-3
OE



ALU

✓J4-Vcc ✓G17-GND ✓J32-Vcc ✓G45-GND
✓J12-GND ✓J40-GND ✓H21-Vcc
✓J5-Vcc ✓J33-Vcc ✓G27-GND

Output enable $\overline{ALUD}$

✓(B) J41 - J38 ✓J38 - J10

ALU OP - see group E A-B

✓J39 - (E) A42 ALU0 ✓J34 - (E) A49 ALU3
✓J37 - (E) A45 ALU1 ✓G32 - (E) B49 ALU4
✓G33 - (E) A46 ALU2 ✓J35 - (E) B46 Cin

Cin ganging: ✓J8 - J35 - see group C G-J

Daisy chain: ✓J7 - J43 ✓J16 - J36

Flags

✓J13 - H22 ✓J41 - H23
✓J14 - H25 ✓J42 - H26

see (F) A-B for flags handling

ALU A Bus In

✓B39 - G41 B38 - G40 ✓B37 - G39 ✓B36 - G38
✓B35 - G13 ✓B34 - G12 ✓B33 - G11 ✓B32 - G10

ALU B Bus In

✓F39 - G37 ✓F38 - G36 ✓F37 - G35 ✓F36 - G34
✓F35 - G9 ✓F34 - G8 ✓F33 - G7 ✓F32 - G6

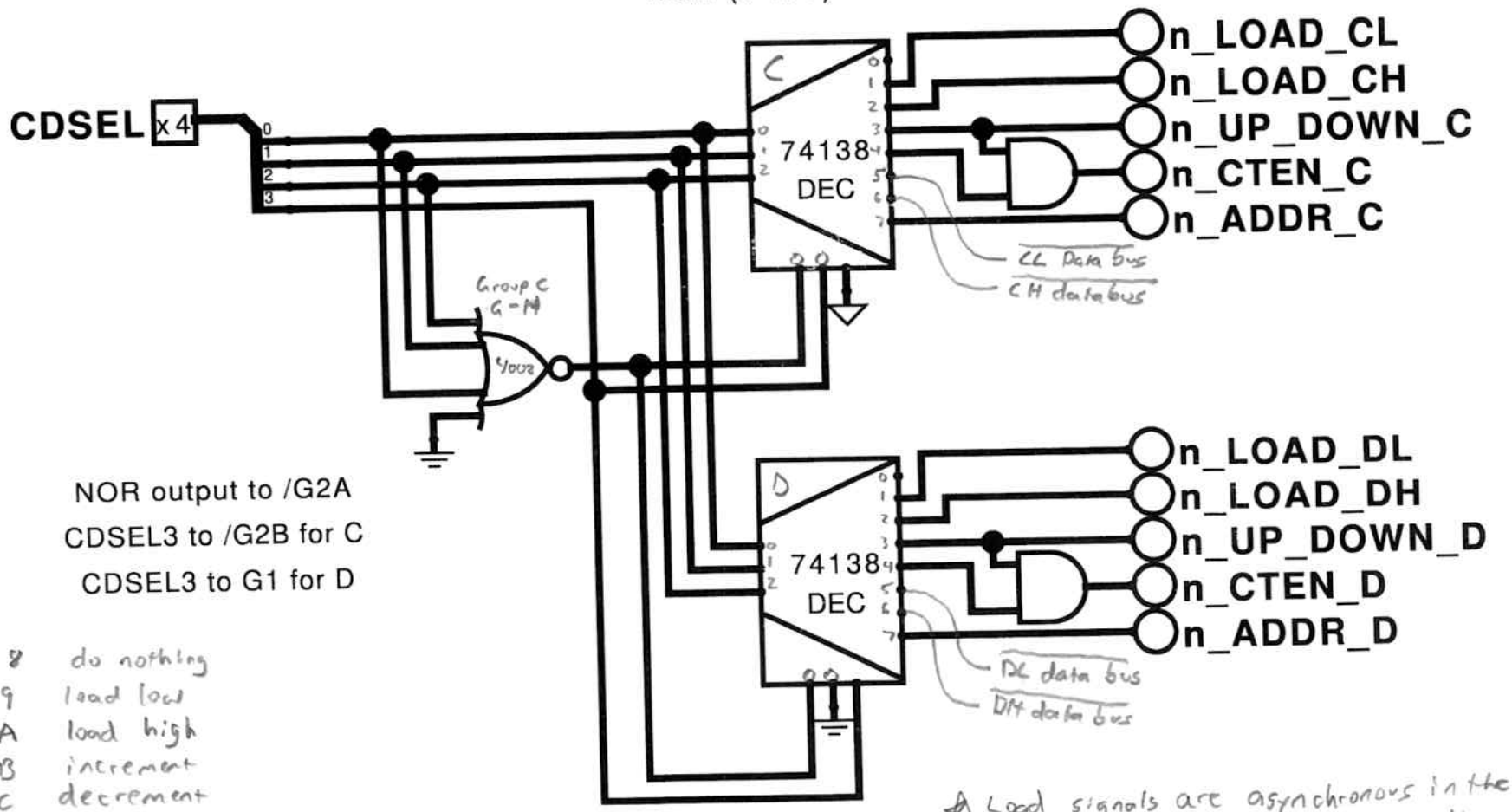
Data Bus Out

✓D0 G42 - (E) A43
✓D1 G43 - A44
✓D2 G44 - A47
✓D3 J45 - A48
✓D4 G14 - (E) B48
✓D5 G15 - B47
✓D6 G16 - B44
✓D7 J17 - B43

ALU OP ganging

✓G4 - G32 ✓G5 - G33 ✓J6 - J34
✓J9 - J37 ✓J11 - J39 ✓J8 - J35 Cin

main (1 of 1)



NOR output to /G2A

CDSEL3 to /G2B for C

CDSEL3 to G1 for D

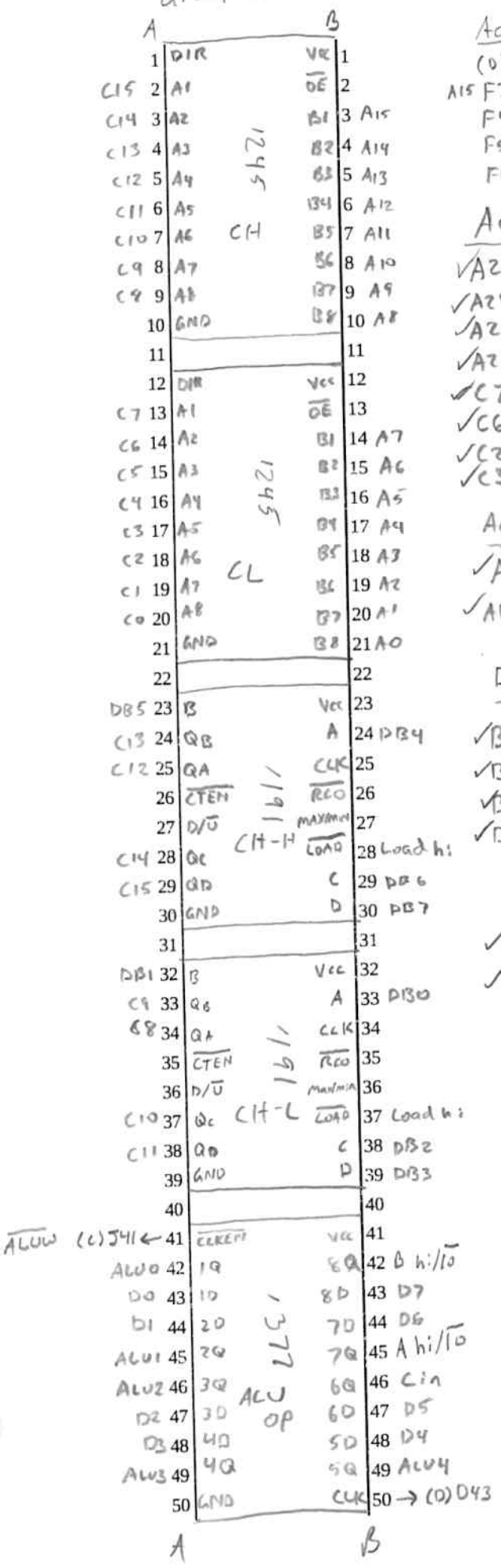
0/8	do nothing
1/9	load low
2/A	load high
3/B	increment
4/C	decrement
5/D	low to data bus
6/E	hi to data bus
7/F	addr bus

LD	CLK	LDO
0	0	0
0	1	1
1	0	1
1	1	1

* Load signals are asynchronous in the '191, so all load signals route through OR gates with CLK to ensure loading ends when CLK ↑

C/D regs

Group E



Address Bus - yellow

(D)
A15 F3 - B3 ✓
F4 - B4 ✓
F5 - B5 ✓
F6 - B6 ✓

(D)
A11 F7 - B7 ✓
F8 - B8 ✓
F9 - B9 ✓
F10 - B10 ✓

(D)
A7 F14 - B14 ✓
F15 - B15 ✓
F16 - B16
F17 - B17 ✓

(D)
A3 F18 - B18 ✓
F19 - B19 ✓
F20 - B20 ✓
F21 - B21 ✓

Address buffer input - C - green

✓A29 - A2 C15
✓A28 - A3 C14
✓A24 - A4 C13
✓A25 - A5 C12
✓C7 - A13 C7
✓C6 - A14 C6
✓C2 - A15 C5
✓C3 - A16 C4
✓A38 - A6 C11
✓A37 - A7 C10
✓A33 - A8 C9
✓A34 - A9 C8
✓C16 - A17 C3
✓C15 - A18 C2
✓C11 - A19 C1
✓C12 - A20 C0

Address Buffer control/Power

✓A1 - B1 ✓B1 - Vcc ✓A10 - GND
✓A12 - B12 ✓B12 - Vcc ✓A21 - GND

Data bus input - CH - red

✓B43 - B30 DB7
✓B44 - B29 DB6
✓B47 - A23 DB5
✓B48 - B24 DB4
✓A48 - B39 DB3
✓A47 - B38 DB2
✓A44 - A32 DB1
✓A43 - B33 DB0

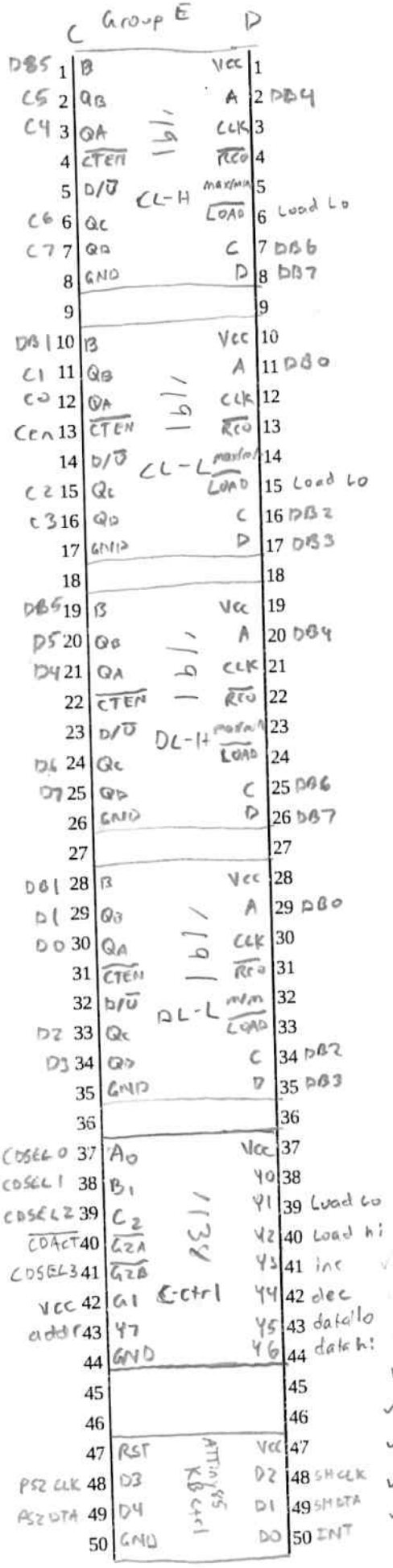
Counter Power

✓B23 - Vcc ✓A30 - GND
✓B32 - Vcc ✓A39 - GND

ALU OP register

✓B41 - Vcc ✓A50 - GND ✓B50 - (D) D43 CLK

Data bus
✓D0 - A43 - (D) C36
✓D1 - A44 - C37
✓D2 - A47 - C40
✓D3 - A48 - C41
✓D4 - B48 - (D) D41
✓D5 - B47 - D40
✓D6 - B44 - D37
✓D7 - B43 - D36



C/D regs

Power

✓D1-Vcc ✓C8-GND ✓D10-Vcc ✓C17-GND
✓D19-Vcc ✓C26-GND ✓D28-Vcc ✓C35-GND

PS2 controller

✓D47-Vcc ✓C50-GND ✓(C)F37-C47 RST
✓D50-(F)D10 D49-(F)F36 SH DATA D48-(F)F38 SH CLK
✓C48-(A)A42 PS2 CLK ✓C49-(A)B45 PS2 DATA

Data Bus Input - CL - red

✓D8-D26 DB7
✓D7-D25 DB6
✓C19-C1 DB5
✓D20-D2 DB4
✓D35-D17 DB3
✓D34-D16 DB2
✓C28-C10 DB1
✓D29-D11 DB0

Data Bus Input - DL - red

✓B30-D26 DB7
✓B29-D25 DB6
✓A23-C19 DB5
✓B24-D20 DB4
✓B39-D35 DB3
✓B38-D34 DB2
✓A32-C28 DB1
✓B33-D29 DB0

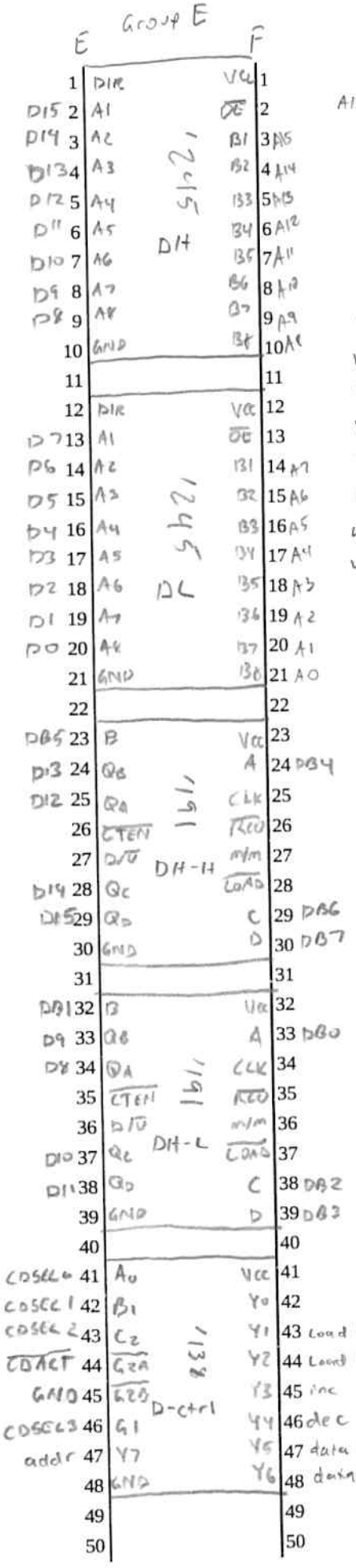
✓CDACT (C)G21-C40
✓CDSEL0 (C)G42-C37 ✓CDSEL2 (C)G44-C39
✓CDSEL1 (C)G43-C38 ✓CDSEL3 (C)J45-C41

C-ctrl logic

✓B37-Vcc ✓C44-GND ✓C42-Vcc

✓D6-D15 ✓(F)C26-D15-Load Lo out

✓B28-B37
✓A27-A36 ✓A27-C5 ✓C5-C14 ✓A36-D41-D/U
✓D41-(D)G21 ✓D42-(D)G22 ✓(D)G23-C13-Cen
✓B2-B13 ✓B13-C43 addr bus
✓D13-C4 ✓D4-A35 ✓B35-A26-RCO-CTEN
✓B25-B34 ✓B34-D30 ✓D30-D21 } CLK
✓D21-D12 ✓D12-D3 ✓B34-B50



C/D regs

Address Bus - yellow

A15 B3 - F3 ✓
B4 - F4 ✓
B5 - F5 ✓
B6 - F6 ✓
A11 B7 - F7 ✓
B8 - F8 ✓
B9 - F9 ✓
B10 - F10 ✓
A7 B14 - F14 ✓
B15 - F15 ✓
B16 - F16 ✓
B17 - F17 ✓
A3 B18 - F18 ✓
B19 - F19 ✓
B20 - F20 ✓
B21 - F21 ✓

Address Bus buffer input - D - green

✓E29 - E2 D15
✓E28 - E3 D14
✓E24 - E4 D13
✓E25 - E5 D12
✓C25 - E13 D7
✓C24 - E14 D6
✓C20 - E15 D5
✓C21 - E16 D4
✓E38 - E6 D11
✓E37 - E7 D10
✓E33 - E8 D9
✓E34 - E9 D8
✓C34 - E17 D3
✓C33 - E18 D2
✓C29 - E19 D1
✓C30 - E20 D0

Address buffer control/Power

✓E1 - F1
✓F1 - VCC
✓E10 - GND
✓E12 - F12
✓F12 - VCC
✓E21 - GND

Data Bus Input - D H - red

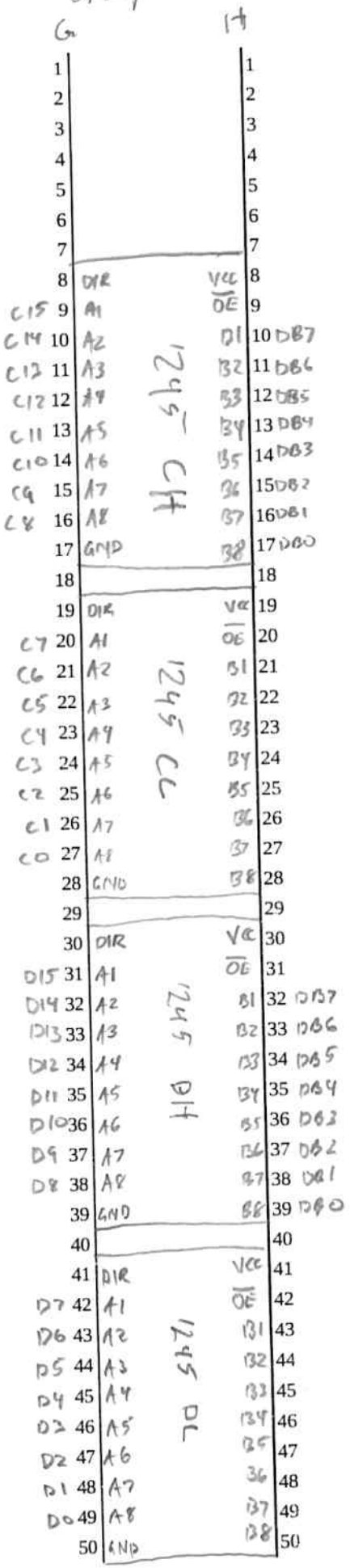
✓D26 - F30 DB7
✓D25 - F29 DB6
✓C19 - E23 DB5
✓D20 - F24 DB4
✓D35 - F39 DB3
✓D34 - F38 DB2
✓C28 - E32 DB1
✓D29 - F33 DB0

counter power

✓F23 - VCC
✓F32 - VCC
✓E30 - GND
✓E39 - GND

D-ctrl logic

✓F41 - VCC
✓E48 - GND
✓C37 - E41
✓C39 - E43
✓C41 - E46
✓C38 - E42
✓C40 - E44
} from C-ctrl
✓E45 - GND
✓D24 - D33 load lo
✓D33 - (F)D27 load lo
✓F28 - F37 load hi
F37 - (F)D30 load hi
✓C32 - C23 inc
✓C23 - E27
✓E27 - E36
✓E36 - F45 D/U
✓F45 - (D)G24 dec
✓F46 - (D)G25
✓(D)G26 - C31 CTEN
✓F2 - F13 data lo
✓F13 - E47 addr bus
✓D31 - C22 data hi
✓D22 - E35
✓F35 - E26 RCO - CTEN
✓D21 - F25
✓D30 - F34 CLK



C/D out to data bus

Power

✓H8 - Vcc ✓G8 - H8 ✓G17 - GND
✓H19 - Vcc ✓G19 - H19 ✓G28 - GND
✓H30 - Vcc ✓G30 - H30 ✓G39 - GND
✓H41 - Vcc ✓G41 - H41 ✓G50 - GND

Control - blue

✓D43 - H20 CL ✓B44 - H9 CH
✓F47 - H42 DL ✓F48 - H31 DH

Data bus gaging - red

✓H10 - H21 ✓H11 - H22 ✓H12 - H23 ✓H13 - H24
✓H14 - H25 ✓H15 - H26 ✓H16 - H27 ✓H17 - H28
✓H21 - H32 ✓H22 - H33 ✓H23 - H34 ✓H24 - H35
✓H25 - H36 ✓H26 - H37 ✓H27 - H38 ✓H28 - H39
✓H32 - H43 ✓H33 - H44 ✓H34 - H45 ✓H35 - H46
✓H36 - H47 ✓H37 - H48 ✓H38 - H49 ✓H39 - H50

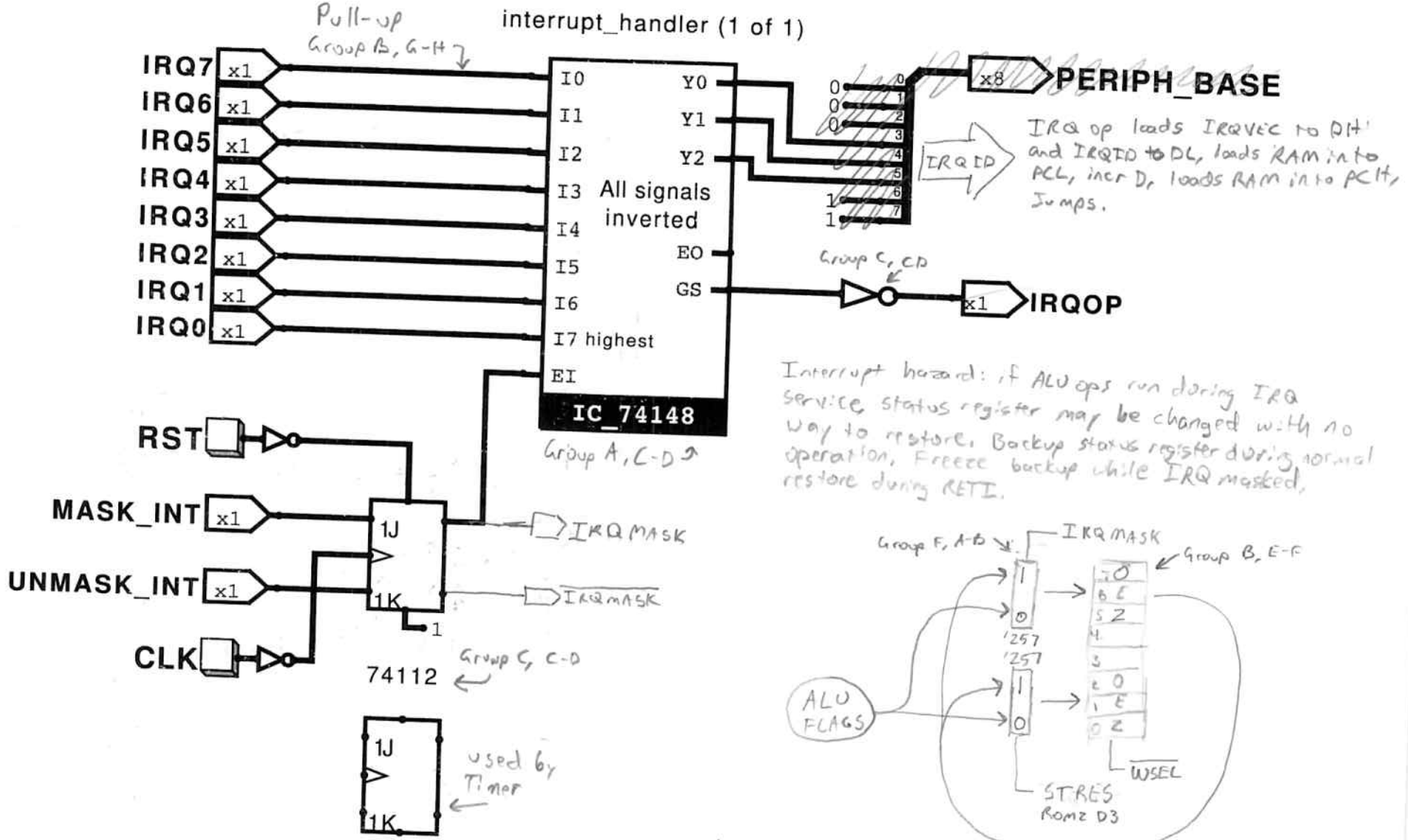
Data bus - red

✓DB7 F30 - H32 ✓DB3 F39 - H36
✓DB6 F29 - H33 ✓DB2 F38 - H37
✓DB5 E23 - H34 ✓DB1 E32 - H38
✓DB4 F24 - H35 ✓DB0 F33 - H39

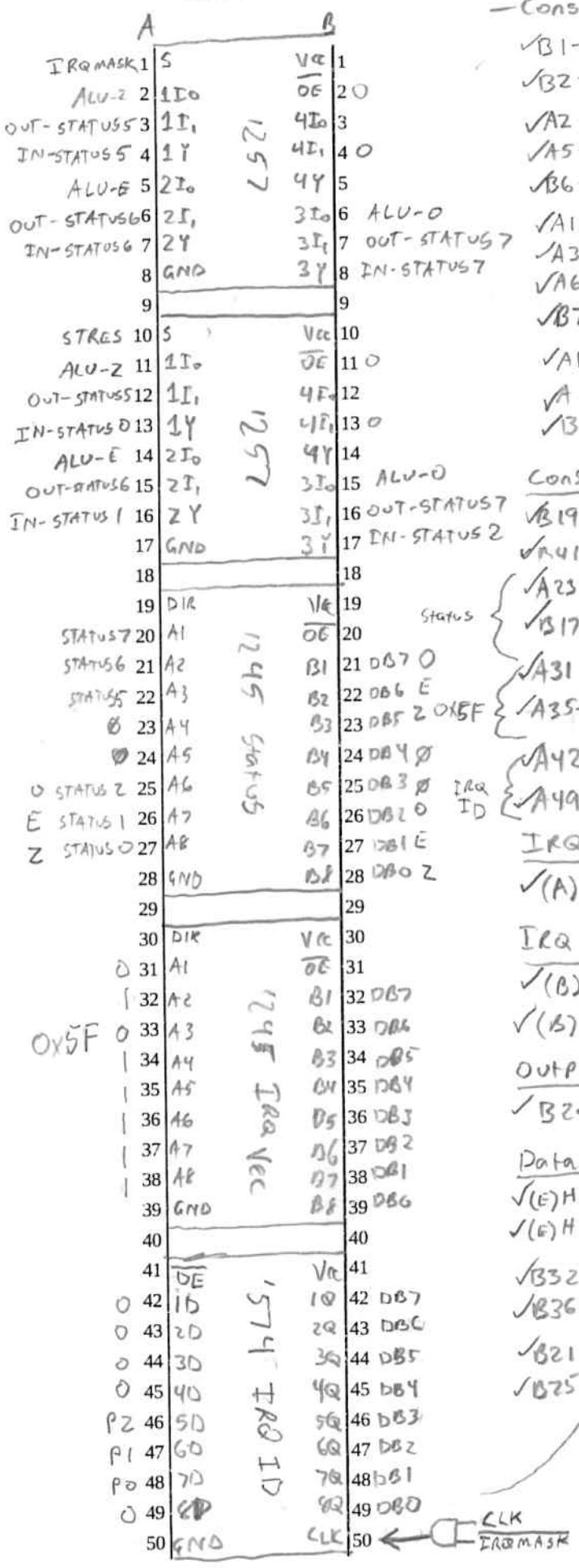
Reg to inputs - green

C15 A2 - G9 ✓ C7 A13 - G20 ✓
C14 A3 - G10 ✓ C6 A14 - G21 ✓
C13 A4 - G11 ✓ C5 A15 - G22 ✓
C12 A5 - G12 ✓ C4 A16 - G23 ✓
C11 A6 - G13 ✓ C3 A17 - G24 ✓
C10 A7 - G14 ✓ C2 A18 - G25 ✓
C9 A8 - G15 ✓ C1 A19 - G26 ✓
C4 A9 - G16 ✓ C0 A20 - G27 ✓

D15 E2 - G31 ✓ D7 E13 - G42 ✓
D14 E3 - G32 ✓ D6 E14 - G43 ✓
D13 E4 - G33 ✓ D5 E15 - G44 ✓
D12 E5 - G34 ✓ D4 E16 - G45 ✓
D11 E6 - G35 ✓ D3 E17 - G46 ✓
D10 E7 - G36 ✓ D2 E18 - G47 ✓
D9 E8 - G37 ✓ D1 E19 - G48 ✓
D8 E9 - G38 ✓ D0 E20 - G49 ✓


Pull-up
Group B, G-H
interrupt_handler (1 of 1)
IRQ7
IRQ6
IRQ5
IRQ4
IRQ3
IRQ2
IRQ1
IRQ0
x1
I0
I1
I2
I3
I4
I5
I6
I7 highest
EI
Y0
Y1
Y2
All signals inverted
EO
GS
IC_74148
Group A, C-D
x8
PERIPH_BASE
IRQ ID
IRQ op loads IRQVEC to PCH and IRQID to PCL, loads RAM into PCL, incr D, loads RAM into PCH, Jumps.
Group C, C-D
x1
IRQOP
RST
MASK_INT
UNMASK_INT
CLK
1J
1K
IRQMASK
74112
Group C, C-D
used by Timer
Interrupt hazard: if ALU ops run during IRQ service, status register may be changed with no way to restore. Backup status register during normal operation, freeze backup while IRQ masked, restore during RETI.
Group F, A-B
IRQMASK
Group B, E-F
'257
'257
ALU FLAGS
WSEL
STRES
Rom2 D3
normal operation:
IRQMASK=0
STRES=0
ALU ops store flags in both sides
during IRQ
IRQMASK=1
STRES=0
ALU ops only store flags in lower side,
END of IRQ (RETI)
IRQMASK=1
STRES=1
Pre-IRQ flags restored
not useful - ALU ops store status in backup, not main
IRQMASK=0
STRES=1

Group F



- IRQ Status Flag hazard protection
- Constants

✓B1-Vcc ✓A8-GND ✓B10-Vcc ✓A17-GND
✓B2 - B4 - GND ✓B11 - B13 - GND

✓A2 - A11 - (D) H24 ZERO
✓A5 - A14 - (D) H27 EQUAL } from ALU
✓B6 - B15 - (D) J15 OVER

✓A1 - (C) C27 IRQ MASK ✓A10 - (C) J17 STRES
✓A3 - A12 - (B) F15 ✓A4 - (B) F16 ZERO Backup
✓A6 - A15 - (B) F14 ✓A7 - (B) F13 EQUAL Backup
✓B7 - B16 - (B) F11 ✓B8 - (B) F12 OVER Backup

✓A13 - (B) E12 ZERO
✓A16 - (B) E13 EQUAL } out to status register
✓B17 - (B) E16 OVER

Constants Vcc/GND

✓B19-Vcc ✓A28-GND ✓B30-Vcc ✓A39-GND
✓A41-Vcc ✓A50-GND ✓A19-B19 ✓A30-B30
✓A23-A24 ✓A24-GND ✓A13-A27 ✓A16-A26
Status ✓B17-A25 ✓A4-A22 ✓A7-A21 ✓B8-A20
✓A31-A33 ✓A31-GND ✓A32-A34 ✓A34-A35
0x5F ✓A35-A36 ✓A36-A37 ✓A37-A38 ✓A38-Vcc
✓A42-A43 ✓A43-A44 ✓A44-A45 ✓A45-A49
IRQ ID ✓A49-GND ✓A30-B30

IRQ ID - Green

✓(A) D44-A48 ✓(A) C43-A47 ✓(A) C42-A46

IRQ ID Write - Blue (← CLK) (← $\overline{\text{IRQ MASK}}$)

✓(B) D28-(B) D25 ✓(B) D24-(C) C28
✓(B) D26-B50

Output Enable - Blue

✓B20-C47 ✓B31-C48 ✓A41-C49

Data Bus - Red

✓(E) H50-B49 ✓H49-B48 ✓H48-B47 ✓H47-B46
✓(E) H46-B45 ✓H45-B44 ✓H44-B43 ✓H43-B42

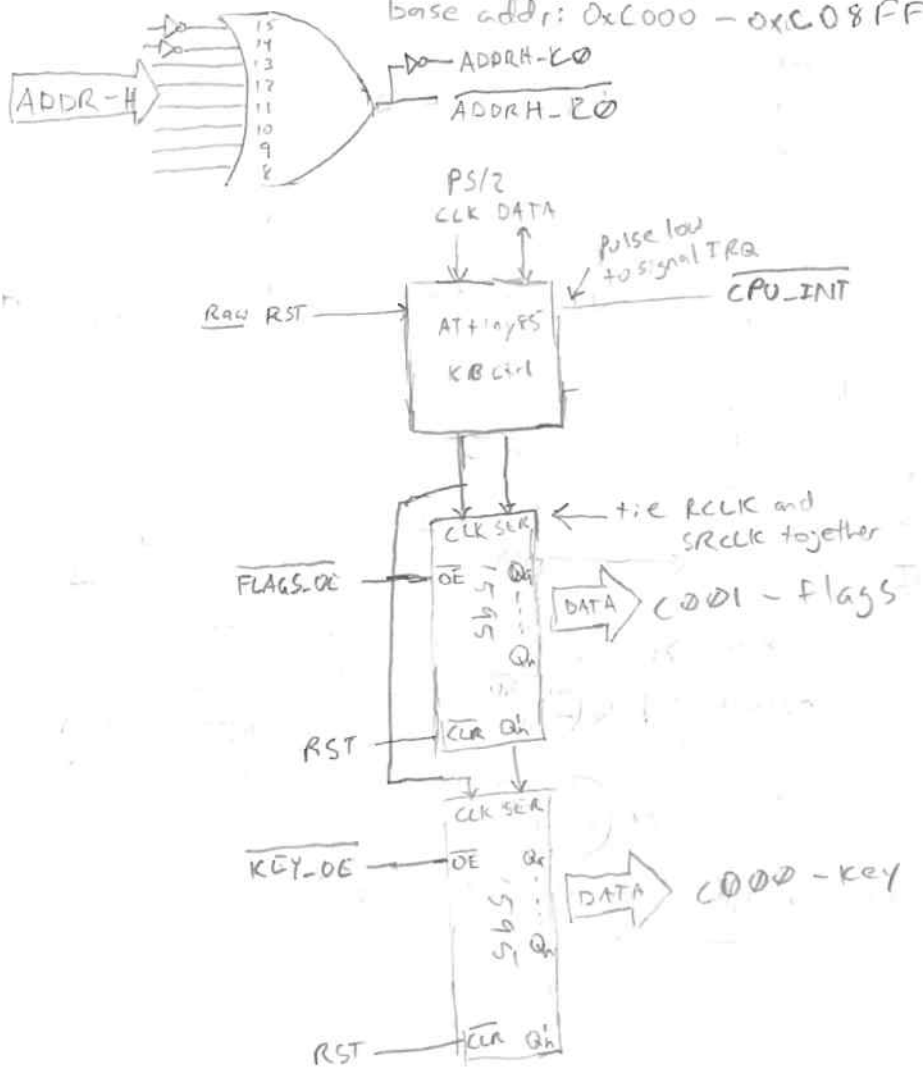
✓B32-B42 ✓B33-B43 ✓B34-B44 ✓B35-B45
✓B36-B46 ✓B37-B47 ✓B38-B48 ✓B39-B49
✓B21-B32 ✓B22-B33 ✓B23-B34 ✓B24-B35
✓B25-B36 ✓B26-B37 ✓B27-B38 ✓B28-B39

Write to IRQ ID when:
+IRQ not masked -
+CLK↑

IRQM	CLK	WRITE
0	0	0
0	1	0
1	0	0
1	1	1

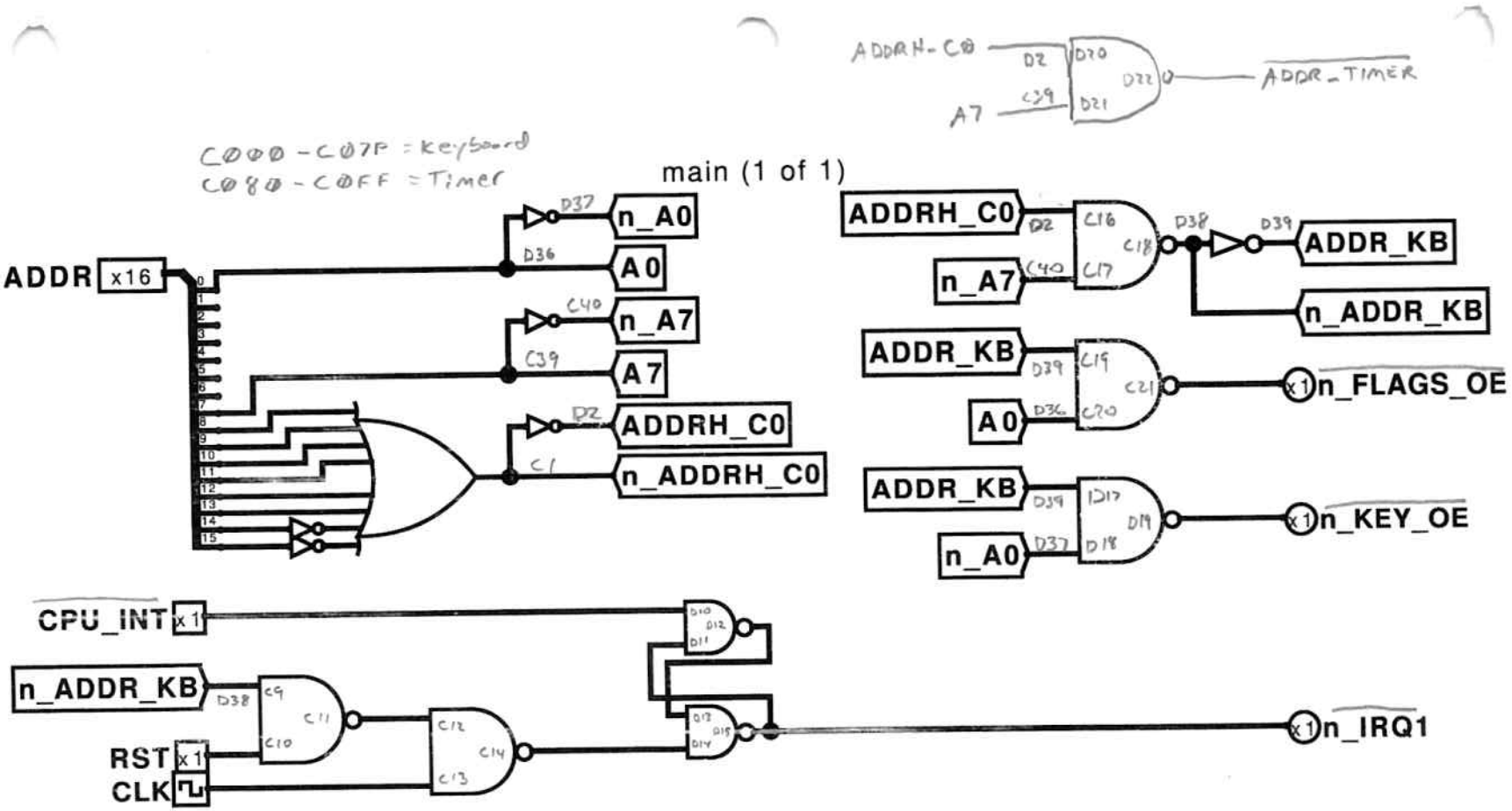
this prevents IRQ ID from changing value while handling an IRQ, should a higher priority interrupt arrive while in handler

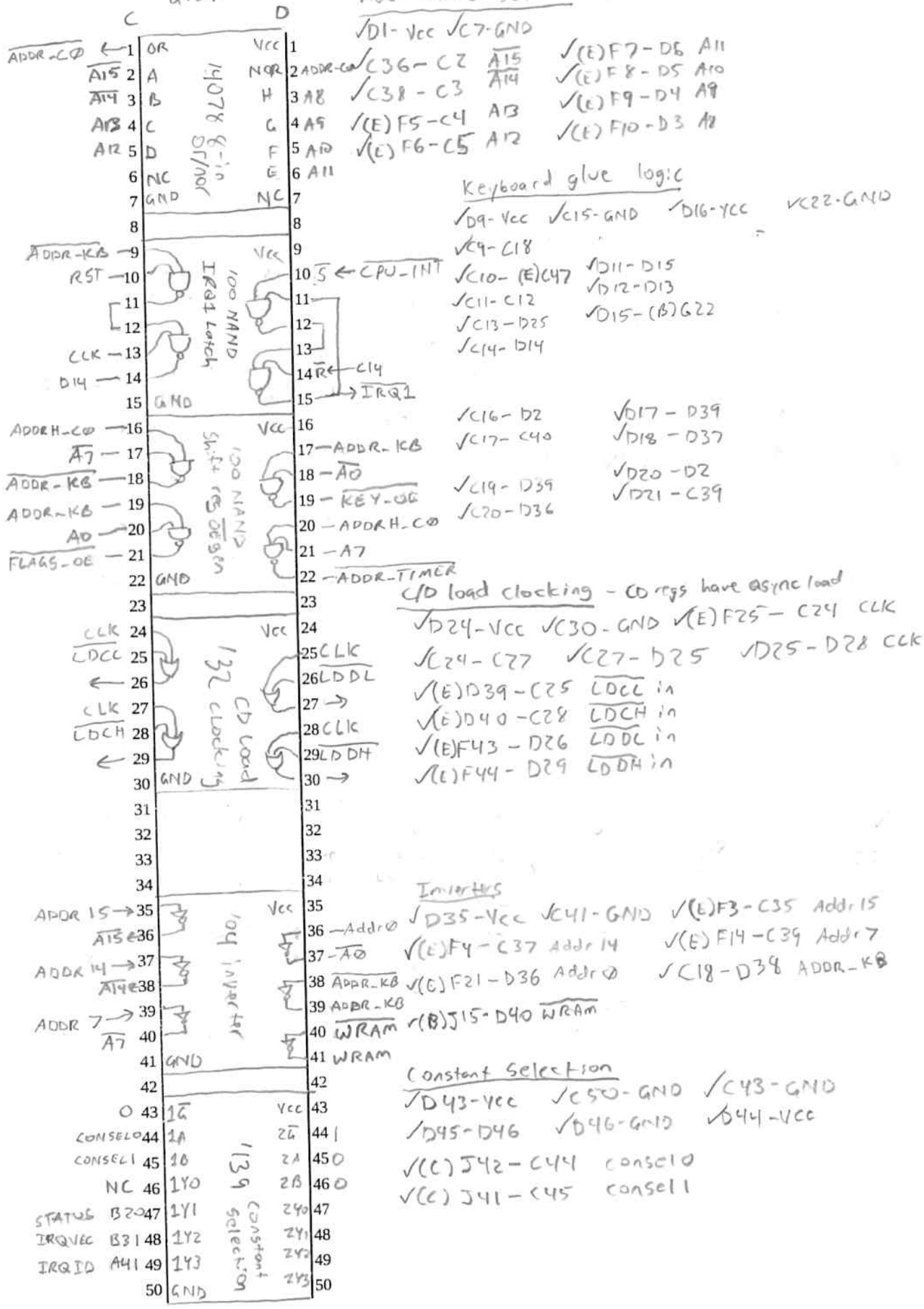
Peripheral 1: keyboard controller
base addr: 0xC000 – 0xC08FF



PS/2 adapter

11 bits sent on each event:
1 start bit (0)
8 data bits
1 parity bit
1 stop bit


ADDRH-C0
D2
A7
C39
D20
D21
D22
ADDR_TIMER
C000-C07F = keyboard
C080-C0FF = Timer
main (1 of 1)
ADDR
x16
n_A0
A0
n_A7
A7
ADDRH_C0
n_ADDRH_C0
D37
D36
C40
C39
D2
C1
ADDRH_C0
n_A7
ADDR_KB
n_ADDR_KB
ADDR_KB
A0
n_FLAGS_OE
ADDR_KB
n_A0
n_KEY_OE
CPU_INT
n_ADDR_KB
RST
CLK
n_IRQ1


Group F
C
D
Addr 0xC0 select - yellow
√D1 - Vcc √C7 - GND
√C36 - C2 A15 A14
√C38 - C3
√(E)F5 - C4 A13
√(E)F6 - C5 A12
√(E)F7 - D6 A11
√(E)F8 - D5 A10
√(E)F9 - D4 A9
√(E)F10 - D3 A8
Keyboard glue logic
√D9 - Vcc √C15 - GND √D16 - Vcc VC22 - GND
√C9 - C18
√C10 - (E)C47 √D11 - D15
√D12 - D13
√C11 - C12
√C13 - D25 √D15 - (B)G22
√C14 - D14
√C16 - D2 √D17 - D39
√C17 - C40 √D18 - D37
√D20 - D2
√C19 - D39 √D21 - C39
√C20 - D36
C/D load clocking - CD regs have async load
√D24 - Vcc √C30 - GND √(E)F25 - C24 CLK
√C24 - C27 √C27 - D25 √D25 - D28 CLK
√(E)D39 - C25 LDCL in
√(E)D40 - C28 LDCH in
√(E)F43 - D26 LDDL in
√(E)F44 - D29 LDDH in
Inverters
√D35 - Vcc √C41 - GND √(E)F3 - C35 Addr 15
√(E)F4 - C37 Addr 14 √(E)F14 - C39 Addr 7
√(E)F21 - D36 Addr 0 √C18 - D38 ADDR-KB
√(B)J15 - D40 WRAM
Constant selection
√D43 - Vcc √C50 - GND √C43 - GND
√D45 - D46 √D46 - GND √D44 - Vcc
√(C)J42 - C44 conselo
√(C)J41 - C45 consel1
ADDR-C0
A15
A14
A13
A12
NC
GND
OR
A
B
C
D
NC
Vcc
NOR
H
G
F
E
NC
74LS08 Quad 2-input AND
ADDR-KB
RST
CLK
D14
IRQ1 Latch
100 NAND
CPU-INT
CLR
PRE
D14
IRQ1
ADDRH-C0
A7
ADDR-KB
ADDR-KB
A0
FLAGS-OE
Shift reg OE gen
ADDR-KB
A0
KEY-OE
ADDRH-C0
A7
ADDR-TIMER
CLK
LDCL
CLK
LDCH
CD Load Clocking
CLK
LDDL
CLK
LDDH
ADDR 15
A15
ADDR 14
A14
ADDR 7
A7
104 Inverter
Addr0
A0
ADDR-KB
ADDR-KB
WRAM
WRAM
0
CONSELO
CONSEL1
NC
STATUS B20
IRQVEC B31
IRQID A41
1G
1A
1B
1Y0
1Y1
1Y2
1Y3
GND
Vcc
2G
2A
2B
2Y0
2Y1
2Y2
2Y3
139 Constant selection

Status as of 2022-11-06

KB interrupts seem to work fine. But controller takes a long time to start up, so a startup delay in the program is needed. Should replace this with a startup detection sequence, like the ay-3-8910 writing a known sequence of bytes and triggering a series of interrupts.

The IRQ and RCTI instructions are broken. The SP grows by 2 each interrupt. And the interrupt vector address gets overwritten, which sends the program into an undefined location (0x3F when testing). Probably related to SP leaking.

Status as of 2022-11-13

IRQ is fine. The STIG instruction blows away the D register, which was causing the weird behavior.

KB is working for basic chars. The controller gets fucky when shift or other flags get toggled.

Status as of 2022-11-24

KB controller seems to have all gremlins eliminated, but needs to be tested in situ

Status as of 2022-11-26

KB controller done, works great.
Timer installed, but test program does not call the interrupt routine. Oscillator/scope is running. Very weird signal on $\overline{WRAM}$. $\overline{OE}$ stays high.

2022-11-27

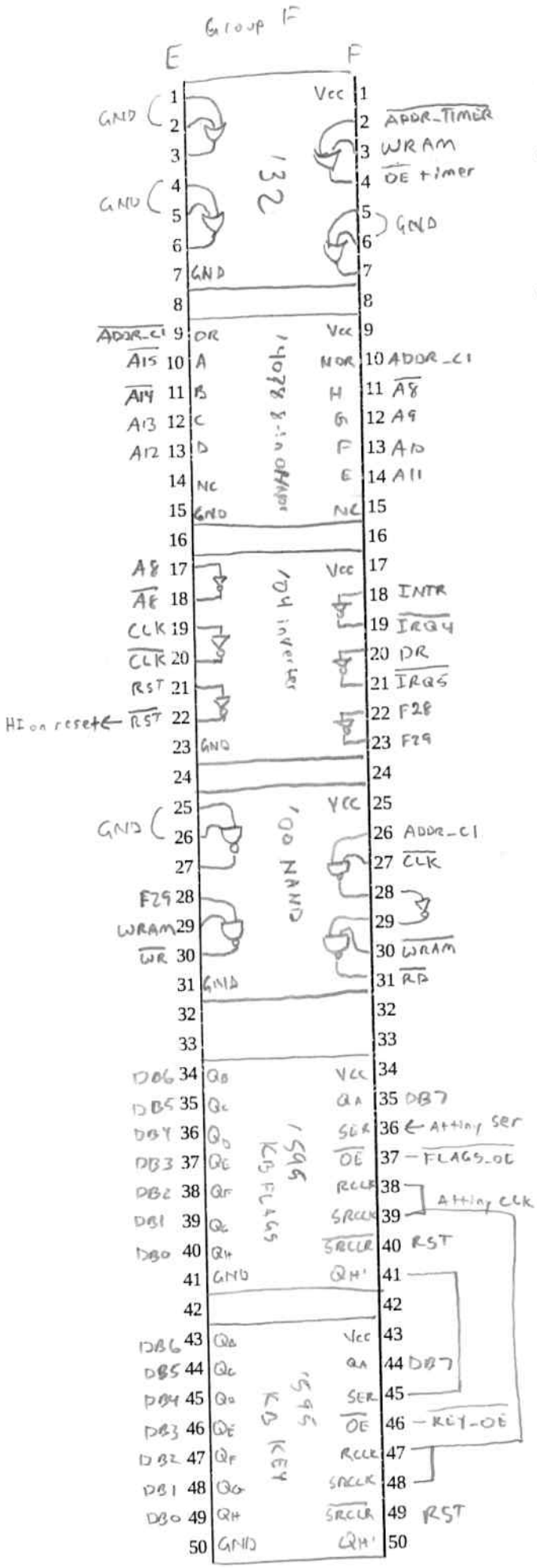
$\overline{WRAM}$ pin was mis-wired. Also borked the $\overline{OE}$ logic. KB IRQ was stuck low. Fixed all that and timer is now working!

2023-03-12

UART is wired up, but does not work yet. IRQ4 is constantly asserted, despite INTR line being low (and OR). Write cycle looks good on timing and $\overline{WR}$ asserts properly. But $\overline{RD}$ does not. So at least two problems: INTR → inv → IRQ4, and $\overline{RD}$ computation logic.

2023-04-30

INTR-INV was open, fixed that. $\overline{WRAM}$ was miswired, fixed. UART appears to be working after noting that IRQ4/5 were reversed in software. But RS232 adapter may be fried.



✓F1- Vcc ✓E7-GND
✓F2-D22 ADDR_TIMER
✓F3-D41 WRAM

UART glue Logic

✓E15-GND ✓F9-Vcc ✓E23-GND ✓F17-Vcc ✓E31-GND ✓E25-Vcc
✓C2-E10 A15 ✓C3-E11 A14 ✓C4-E12 A13
✓C5-E13 A12 ✓D6-F14 A11 ✓D5-F13 A10
✓D4-F12 A9 ✓D3-E17 A8 ✓E18-F11 A8

✓F10-F26 ADDR_C1 ✓D28-E19 CLK
✓E20-F27 CLK ✓F28-F22 ✓F23-F29
✓D40-F30 WRAM ✓F23-E28 ✓F3-E29 WRAM

✓C10-E21 RST ✓F19-(B)G25 IRQ4
✓F21-(B)G26 IRQ5

Keyboard registers

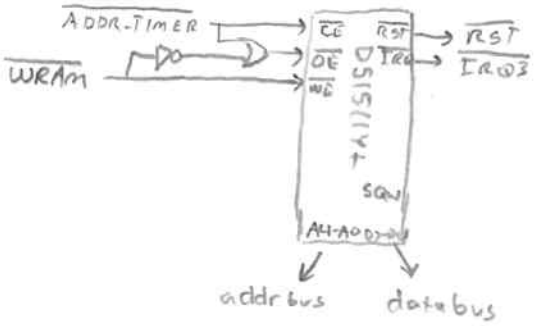
✓F34-Vcc ✓E41-GND ✓F43-Vcc ✓E50-GND

2 pairs:
✓E40-E49 ✓E39-E48 ✓E38-E47 ✓E37-E46
✓E36-E45 ✓E35-E44 ✓E34-E43 ✓F35-E44

2 pairs:
✓B49-E49 ✓B48-E48 ✓B47-E47 ✓B46-E46
✓B45-E45 ✓B44-E44 ✓B43-E43 ✓B42-F44

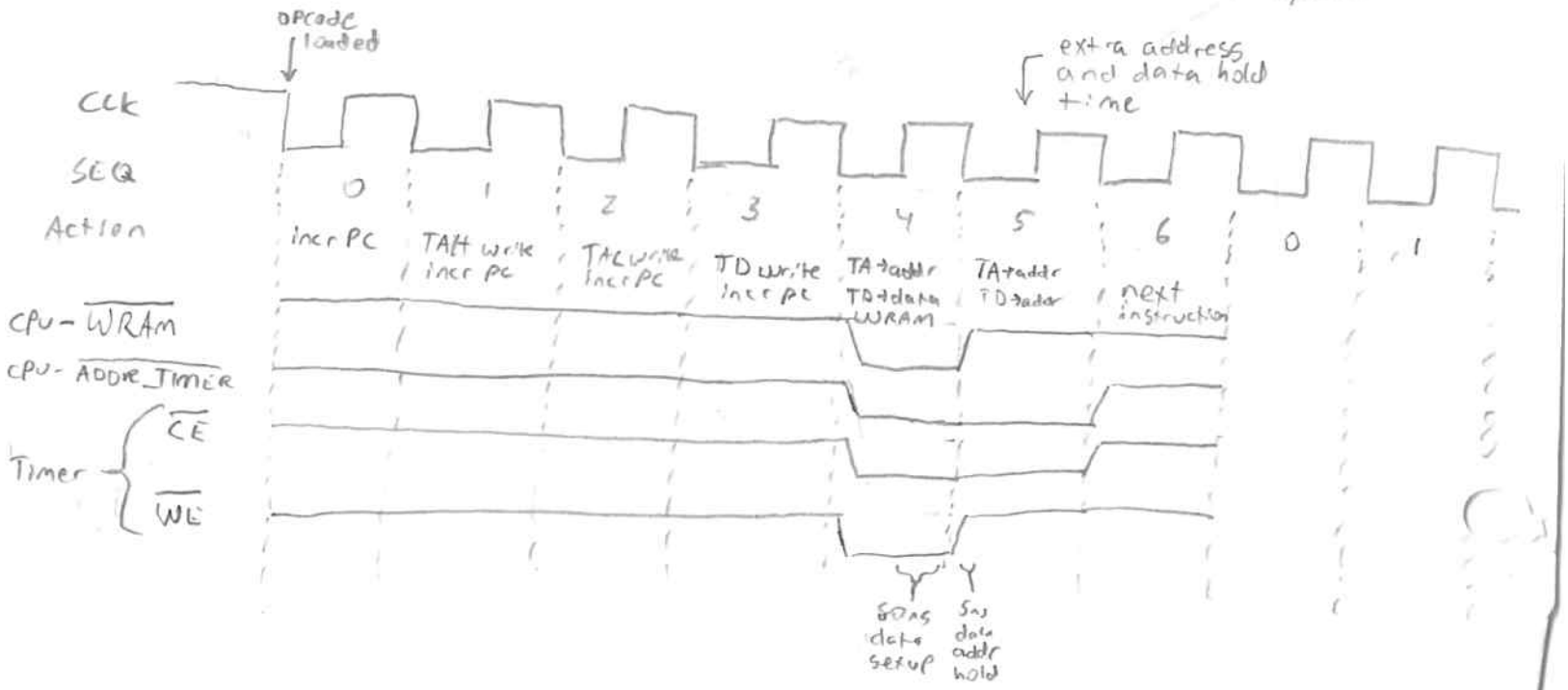
✓F36-(E)D49 SER
✓F38-F39 ✓F47-F48 ✓F39-F47
✓F38-(E)D48 SCLK
✓F41-F45 QH'-SER
✓C10-F40 ✓F40-F49 RST
✓C21-F37 FLAGS_OE
✓D19-F46 KEY_OE

0xC080 - 0xC09F = timer regs (00-1F) Timer Circuitry

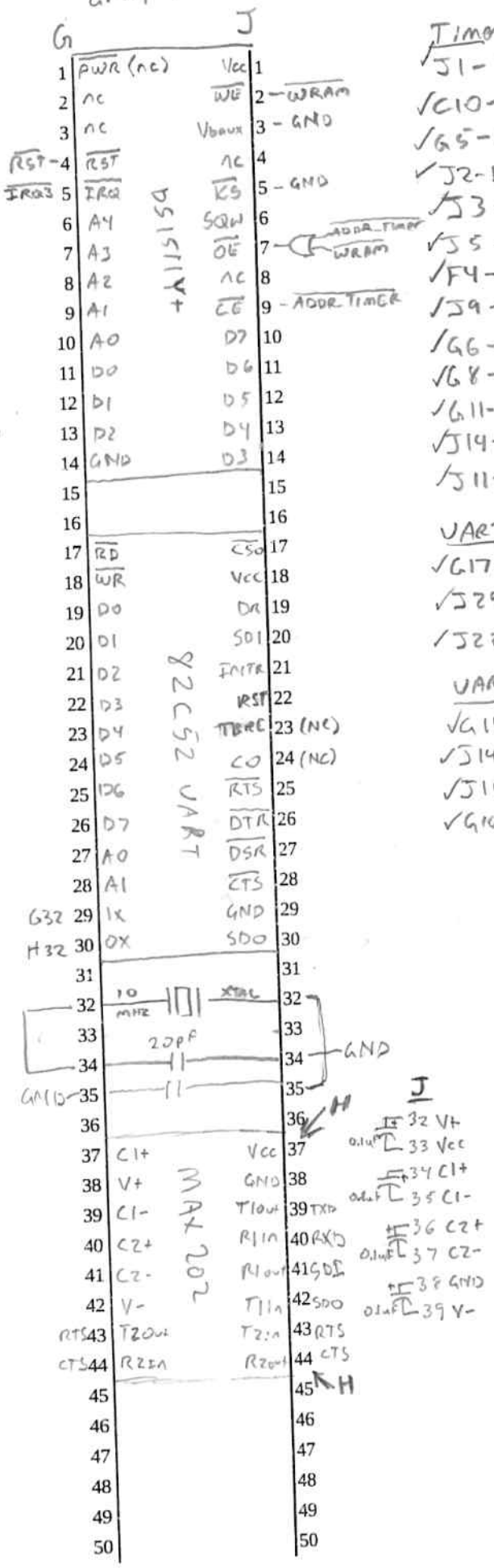


ADDR_TIMER	WRAM (inverted)	OE
0	0	0
0	1	1
1	0	1
1	1	1

Read and write timing has high latency. 120ns from address stable to data stable on reads. On writes, data must be stable for 80 ns then held for 5ns after $\overline{WE}$ goes high. At 2.5 MHz, each clock is 400ns (200ns between edges) and so reads should be ok. Writes are more challenging since the data and address have to be held after $\overline{WE}$ goes high. So a special write opcode will be needed*



Group F



Timer

✓J1 - Vcc ✓G14 - GND
✓C10 - G4 $\overline{RST}$
✓G5 - (B) G24 $\overline{IRQ3}$
✓J2 - D40 $\overline{WRAM}$
✓J3 - GND Vbaux not used
✓J5 - GND $\overline{CS}$ not used
✓F4 - J7 $\overline{OE}$
✓J9 - F2 ADDR_TIMER
✓G6 - (E) F17 A4 ✓G7 - (E) F18 A3
✓G8 - (E) F19 A2 ✓G9 - (E) F20 A1 ✓G10 - (E) F21 A0
✓G11 - (E) H17 D0 ✓G12 - (E) H16 D1 ✓G13 - (E) H15 D2
✓J14 - (E) H14 D3 ✓J13 - (E) H13 D4 ✓J12 - (E) H12 D5
✓J11 - (E) H11 D6 ✓J10 - (E) H10 D7

UART Control

✓G17 - F31 $\overline{RD}$ ✓G18 - E30 $\overline{WR}$ ✓J17 - E9 $\overline{CS0}$
✓J29 - GND ✓J18 - Vcc ✓J21 - F18 INTR
✓J22 - E22 RST ✓J19 - F20 DR ✓G29 - G32 ✓G30 - H32
✓H32 - H35 ✓H34 - GND ✓G32 - G34
✓G35 - GND

UART Addr/Data

✓G11 - G19 D0 ✓G12 - G20 D1 ✓G13 - G21 D2
✓J14 - G22 D3 ✓J13 - G23 D4 ✓J12 - G24 D5
✓J11 - G25 D6 ✓J10 - G26 D7
✓G10 - G27 A0 ✓G9 - G28 A1

MAX202 Charge Pumps

✓H37 - Vcc ✓H38 - GND ✓J33 - Vcc ✓J32 - G38
✓J34 - G37 ✓J35 - G39 ✓J36 - G40 ✓J37 - G41
✓J38 - GND ✓J39 - G42

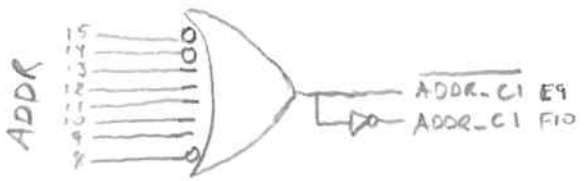
MAX202 - DB9 - UART

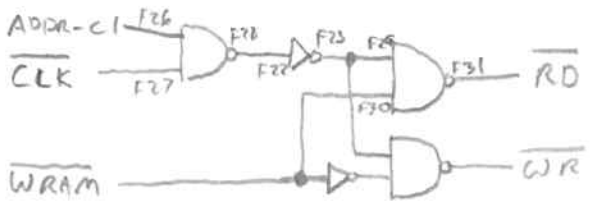
✓J30 - H42 SDO ✓H39 - (J1-6) TXD
✓J20 - H41 SDI ✓H40 - (J1-15) RXD
✓J25 - H43 RTS ✓G43 - (J1-2) RTS
✓J28 - H44 CTS ✓G44 - (J1-3) CTS

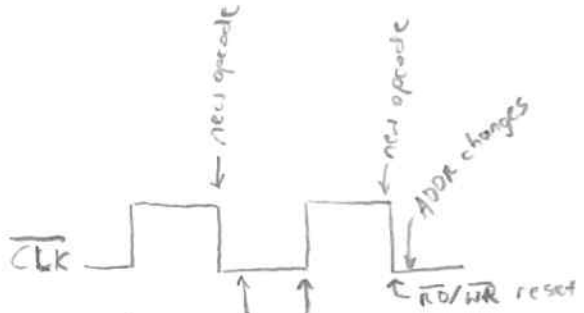
3-Wire Serial

GND - GND - GND
SDO → T1in T1out → TXD
SDI ← R1out R1in ← RXD
RTS → T2in T2out → RTS
CTS ← R2out R2in ← CTS

UART glue logic







Note: DATA must stay valid for 20 ns after RD/WR reset. CS0 must stay valid for 50 ns after RD/WR reset.

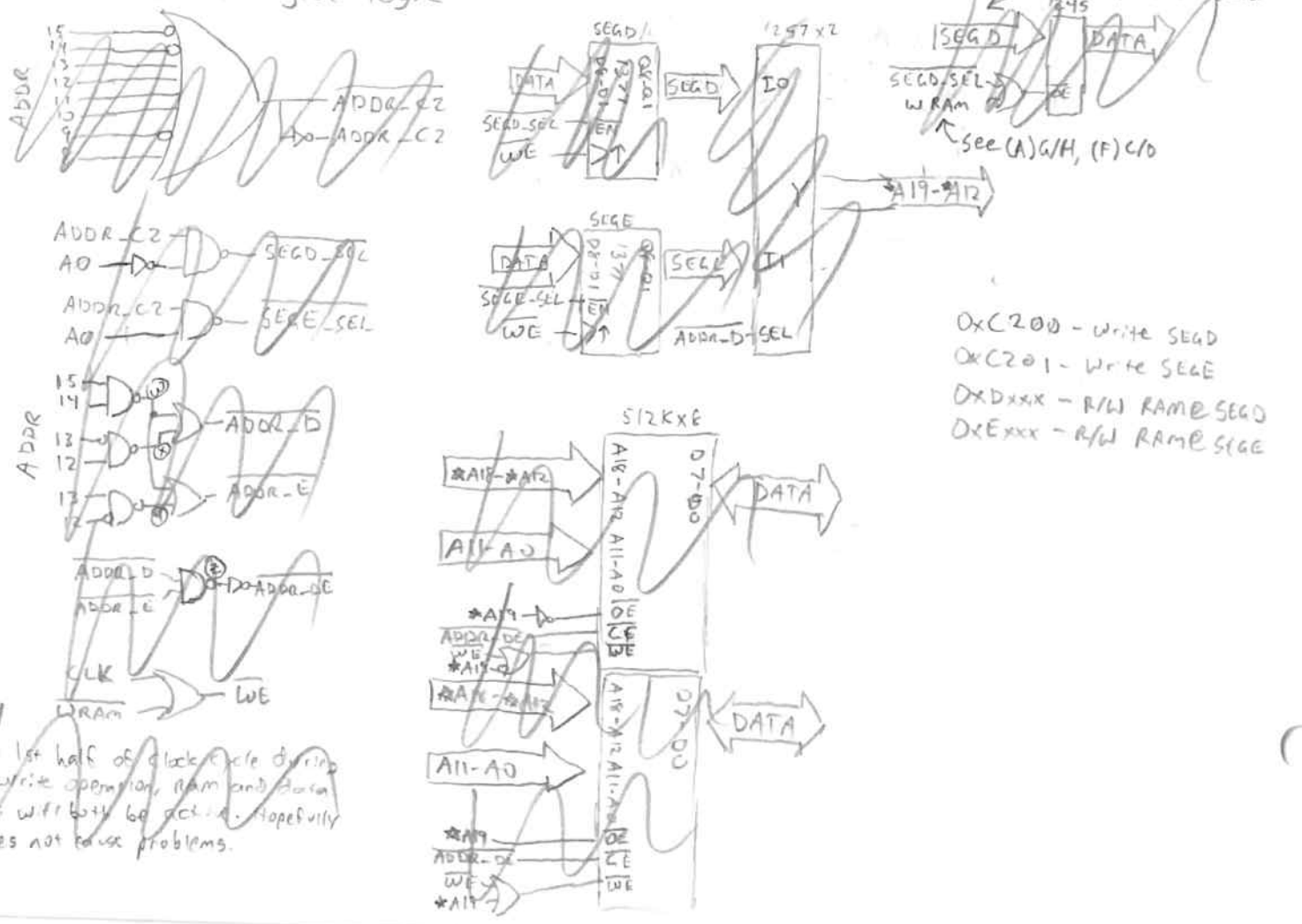
10 mHz XTAL for baud rate generator
1.73% error for each:

baud	prescaler	divisor
2400	4	64
4800	4	32
9600	3	22
19200	4	8
38400	1	16
115200	1	16/3

F-J1 RS-232 Port

J1	DB9	J1	DB9
1 - DSR	(6)	14 - DCD	(1)
2 - RTS	(7)	15 - RXD	(2)
3 - CTS	(8)	16 - TXD	(3)
4 - RI	(9)	17 - DTR	(4)
5 - NC		18 - GND	(5)

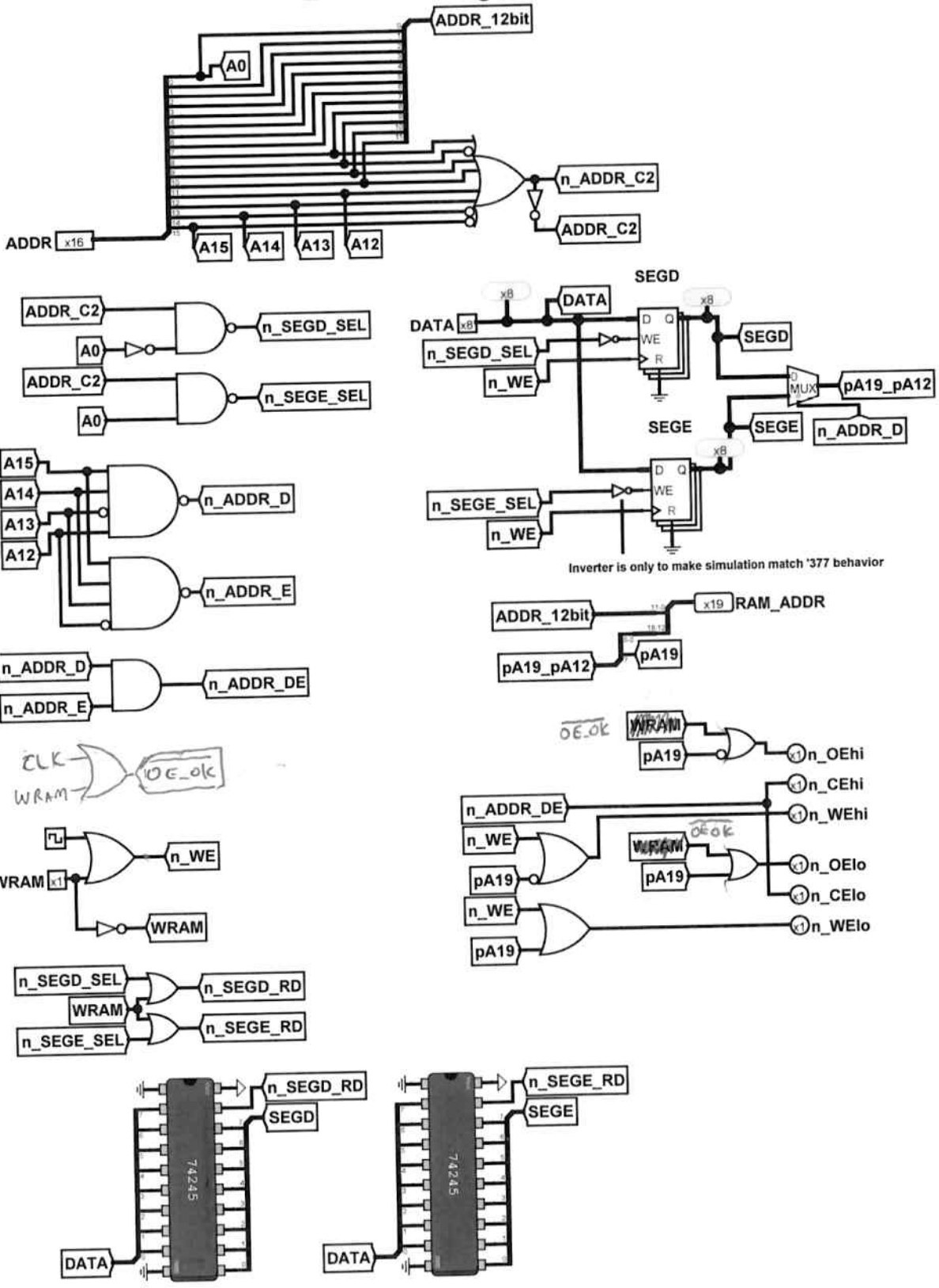
Extended RAM glue logic



0xC200 - Write SEGD
0xC201 - Write SEGE
0xDxxx - R/W RAM@SEGD
0xExxx - R/W RAM@SEGE

For 1st half of clock cycle during a write operation, RAM and flash will both be active. Hopefully does not cause problems.

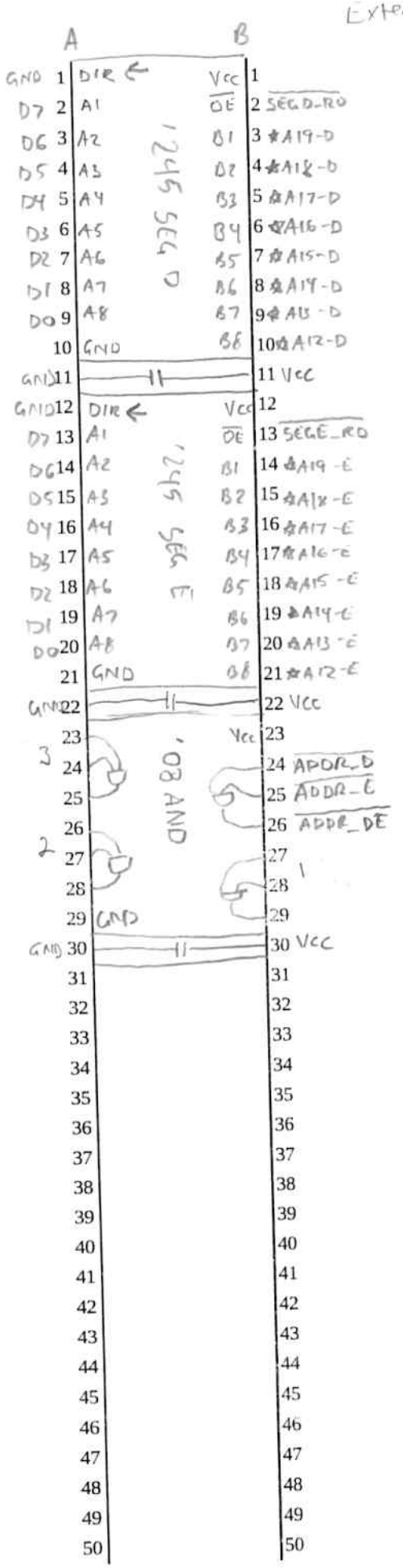
Extended RAM glue logic



Auxiliary Board bus interface

View from bottom (pin side)

Extended RAM Read Buffers



✓A1-GND ✓A10-GND ✓B1-B11 ✓B11-Vcc
✓A12-GND ✓A21-GND ✓B12-B11 ✓A11-GND

✓A2-A13 ✓A3-A14 ✓A4-A15 ✓A5-A16
✓A6-A17 ✓A7-A18 ✓A8-A19 ✓A9-A20

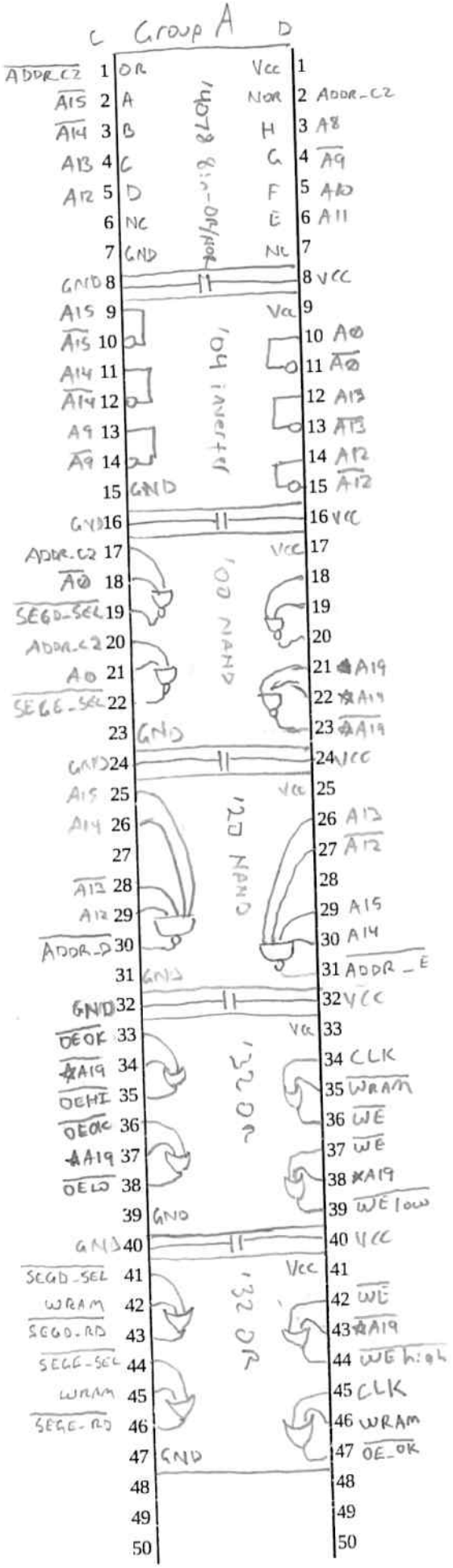
✓A2-[D7] ✓A3-[D6] ✓A4-[D5] ✓A5-[D4]
✓A6-[D3] ✓A7-[D2] ✓A8-[D1] ✓A9-[D0]

✓B3-F2 ✓B4-F5 ✓B5-F6 ✓B6-F9
✓B7-E9 ✓B8-E6 ✓B9-E5 ✓B10-E2

✓B14-F13 ✓B15-F16 ✓B16-F17 ✓B17-F20
✓B18-E20 ✓B19-E17 ✓B20-E16 ✓B21-E13

✓B2-C43 ✓B13-C46

✓B22-Vcc ✓B23-B22 ✓A23-A22 ✓A22-GND
✓B24-C30 ✓B25-D31



Extended RAM Glue Logic

ADDR_C2 generation

✓D1-Vcc ✓C7-GND ✓D8-Vcc ✓D1-D8 ✓C8-GND
✓C2-C10 $\overline{A15}$ ✓C3-C12 $\overline{A14}$ ✓C4-D12 A13
✓C5-D14 A12 ✓D6-[A11] ✓D5-[A10]
✓D4-C14 $\overline{A9}$ ✓D3-[A8]

Address inverters

✓D8 Vcc C16-GND ✓D8-D9
✓C9-[A15] ✓C11-[A14] ✓C13-[A9]
✓D14-[A12] ✓D12-[A13] ✓D10-[A0]

✓C39-C40
✓C7-C8 ✓C15-C16 ✓C23-C24 ✓C31-C32 ✓D40-Vcc

Glue logic

✓D16-Vcc ✓D24-Vcc ✓D32-Vcc
✓D17-D16 ✓D25-D24 ✓D33-D32 ✓D41-D40

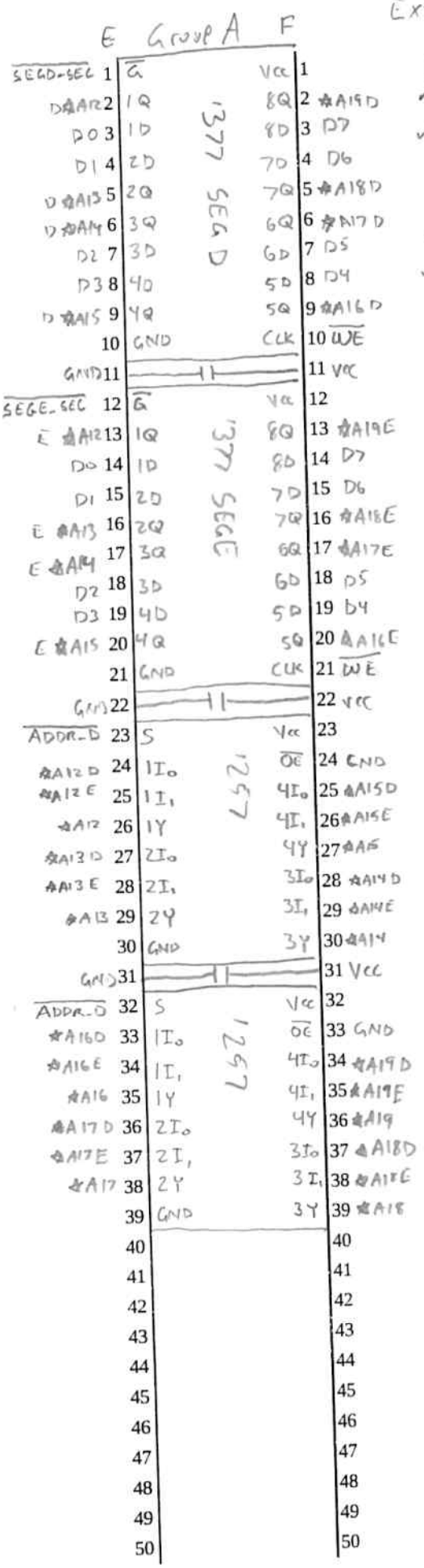
✓D2-C17 ✓D11-C18 ✓C17-C20 ✓D10-C21
✓D21-C37 ✓D23-C34 ✓D21-D22 ✓D21-F36
✓C16-GND ✓C24-GND ✓C32-GND ✓C40-GND

✓C25-C9 ✓C26-C11 ✓C28-D13 ✓C29-D14
✓D26-D12 ✓D27-D15 ✓C25-D29 ✓C26-D30

✓D34-[CLK] ✓D35-[$\overline{WRAM}$] ✓D36-D37
✓D38-D22 ✓C33-C36
✓C42-[WRAM]
✓C41-C19 ✓C42-C36 ✓C47-C40
✓C44-C22 ✓C45-C42
✓D42-D36 ✓D43-D23

✓D34-D45 ✓C42-D46 C36-D47

Extended RAM Segment Registers



Power

✓F1-F11 ✓E10-GND ✓F12-F11 ✓E21-GND

✓F23-F22 ✓E30-GND ✓F32-F31 ✓E39-GND

SEG Registers

✓E1-C19 ✓F10-D36 ✓E12-C22 ✓P21-F10

✓E23-E32 ✓F24-GND ✓E32-C30 ✓F33-GND

Data bus

✓E3-[D0] ✓E4-[D1] ✓E7-[D2] ✓E8-[D3]

✓F8-[D4] ✓F7-[D5] ✓F4-[D6] ✓F3-[D7]

✓E3-E14 ✓E4-E15 ✓E7-E18 ✓E8-E19

✓F8-F19 ✓F7-F18 ✓F4-F15 ✓F3-F14

Segregister to mux

✓E2-E24 ✓E5-E27 ✓E6-F28 ✓E9-F25

✓F9-E33 ✓F6-E36 ✓F5-F37 ✓F2-F34

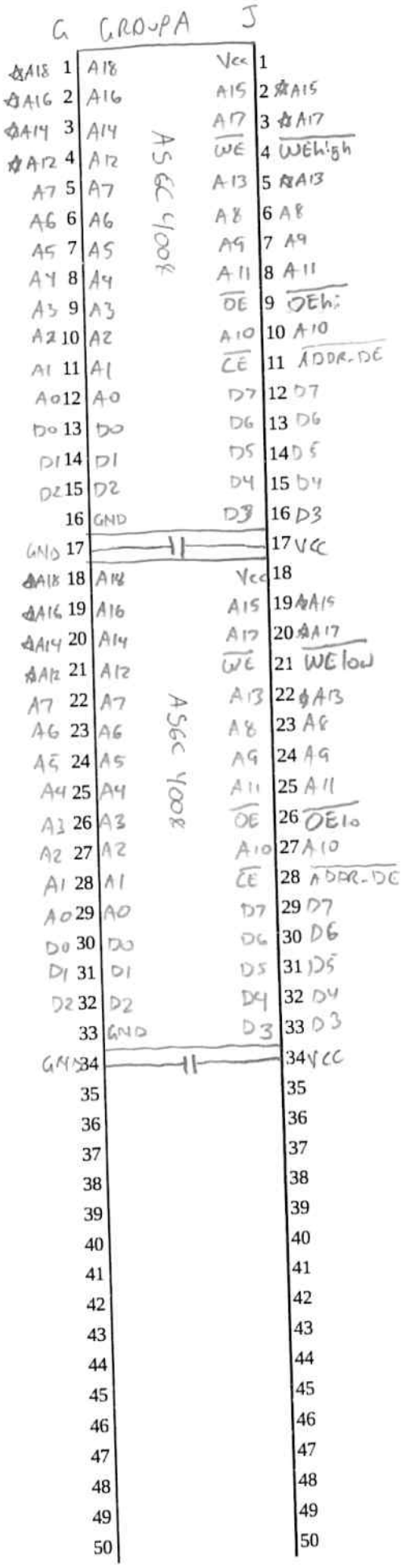
✓E13-E25 ✓E16-E28 ✓E17-F29 ✓E20-F26

✓F20-E34 ✓F17-E37 ✓F16-F38 ✓F13-F35

Capacitors

✓F11-Vcc ✓F22-Vcc ✓F31-Vcc

✓E11-GND ✓E22-GND ✓E31-GND



Extended RAM

✓J1-J17 ✓G16-G17 ✓J18-J37 ✓G39-G17
✓J17-Vcc ✓G17-GND

control signals

✓J4-D44 ✓J9-C35 ✓J28-J11
✓J21-D39 ✓J26-C38 ✓J28-B26

Data Bus Gang

✓G13-G30 ✓G14-G31 ✓G15-G32 ✓J16-J33
✓J15-J32 ✓J14-J31 ✓J13-J30 ✓J12-J29

Data BUS

✓G13-[D0] ✓G14-[D1] ✓G15-[D2] ✓J16-[D3]
✓J15-[D4] ✓J14-[D5] ✓J13-[D6] J12-[D7]

Address Gang

✓G1-G18 ✓G2-G19 ✓G3-G20 ✓G4-G21
✓G5-G22 ✓G6-G23 ✓G7-G24 ✓G8-G25
✓G9-G26 ✓G10-G27 ✓G11-G28 ✓G12-G29
✓J2-J19 ✓J3-J20 ✓J5-J22 ✓J6-J23
✓J7-J24 ✓J8-J25 ✓J10-J27

Ext address

✓G1-F39 ✓J3-E38 ✓G2-E35 ✓J2-F27
✓G3-F30 ✓J5-E29 ✓G4-E26

Address

✓J8-[A11] ✓J10-[A10] ✓J7-[A9] ✓J6-[A8]
✓G5-[A7] ✓G6-[A6] ✓G7-[A5] ✓G8-[A4]
✓G9-[A3] ✓G10-[A2] ✓G11-[A1] ✓G12-[A0]