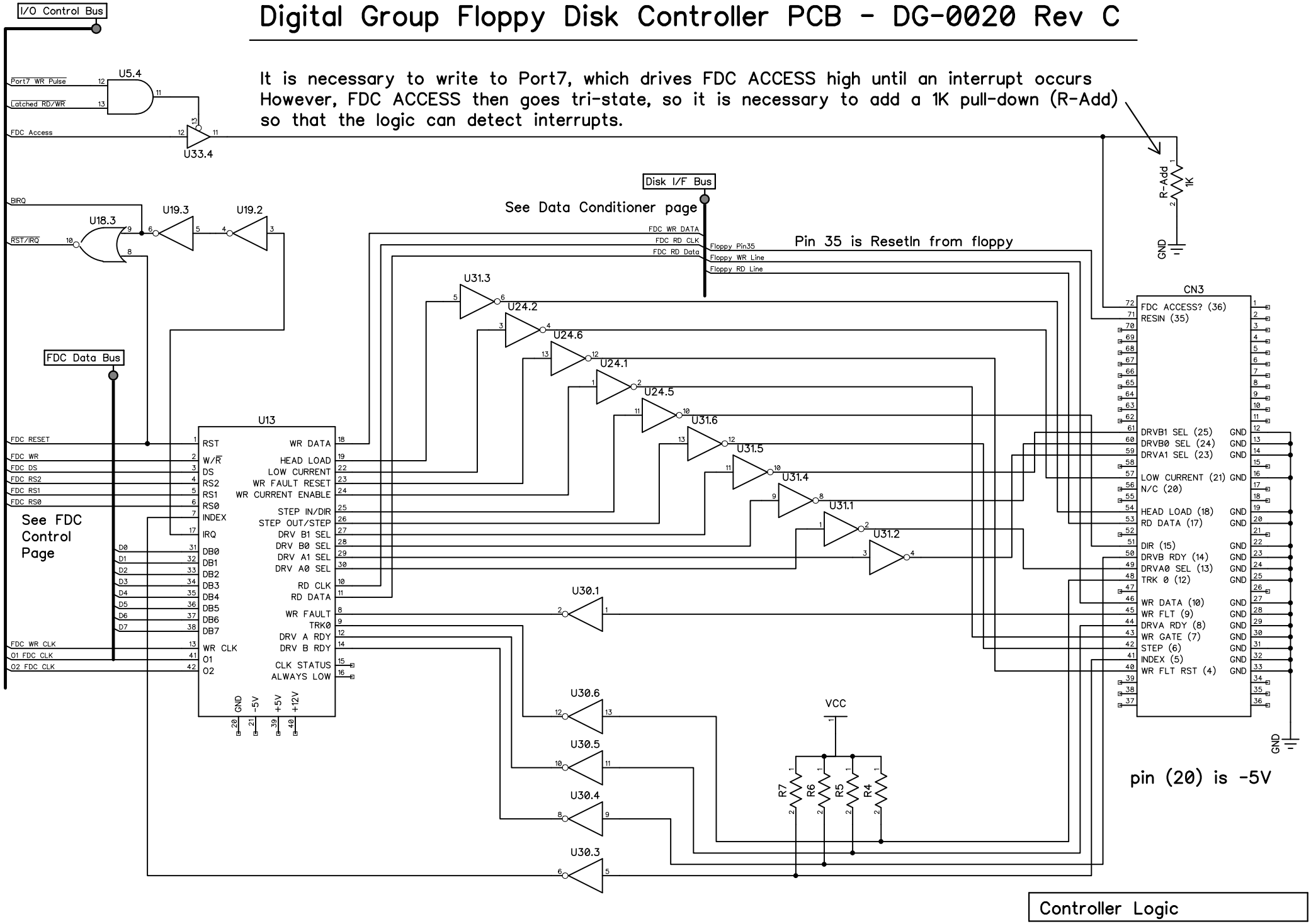
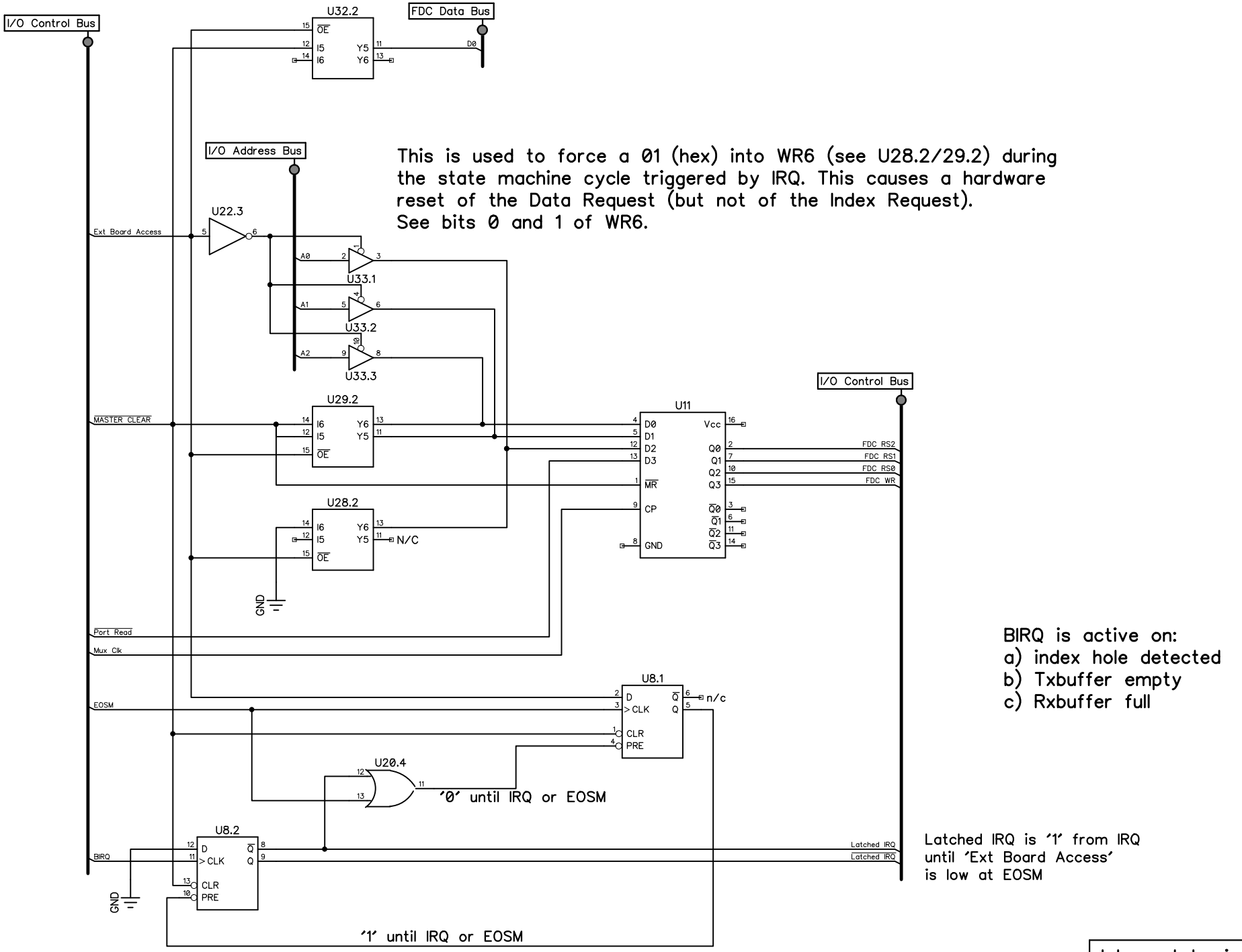
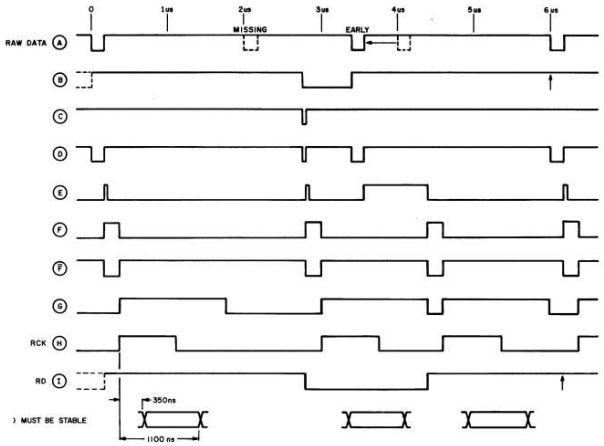
Digital Group Floppy Disk Controller PCB - DG-0020 Rev C




I/O Control Bus
U32.2
FDC Data Bus
D0
I/O Address Bus
This is used to force a 01 (hex) into WR6 (see U28.2/29.2) during the state machine cycle triggered by IRQ. This causes a hardware reset of the Data Request (but not of the Index Request). See bits 0 and 1 of WR6.
U22.3
Ext Board Access
A0
U33.1
A1
U33.2
A2
U33.3
I/O Control Bus
U29.2
MASTER CLEAR
U11
FDC RS2
FDC RS1
FDC RS0
FDC WR
U28.2
N/C
GND
Port Read
Mux Clk
U8.1
EOSM
n/c
U20.4
'0' until IRQ or EOSM
U8.2
BIRQ
Latched IRQ
Latched IRQ
GND
'1' until IRQ or EOSM
BIRQ is active on:
a) index hole detected
b) Txbuffer empty
c) Rxbuffer full
Latched IRQ is '1' from IRQ until 'Ext Board Access' is low at EOSM
Interrupt Logic



Notes about 'Floppy RD/WR Line's
Data is shifted MSB first, Clock-Data-Clock-Data...
Clock patterns: CBD 11000111, CBI 11010111, CBN 11111111 (MSB first)
How Marks appear on logic analyzer:
Data Mark (0xFB w/CBD) -> 1111010101101111 (MSB first)
ID Address Mark (0xFE w/CBD) -> 1111010101111110 (MSB first)
ASCII 'A' (0x41 w/CBN) -> 1011101010101011 (MSB first)

Disk I/F Bus

FDC RD Data

FDC RD CLK

I/O Control Bus

MASTER CLEAR

8224 RESIN

Disk I/F Bus

Floppy RD Line

U1.3

Cut

RC Omitted

U3.1

R10 220

C5 200pF

GND

U16.1

U5.1

U4.1

U5.2

U15.1

RC Omitted

U4.2

U2.2

RC Omitted

U3.2

IMPORTANT: the Floppy RD Line on the Shugart 801 is inverted from what this expects, so you need to cut U1.3 pin 8 trace, and jumper it U1.3 pin 10.

U2.1

RC Omitted

FDC WR DATA

nc

U14

nc

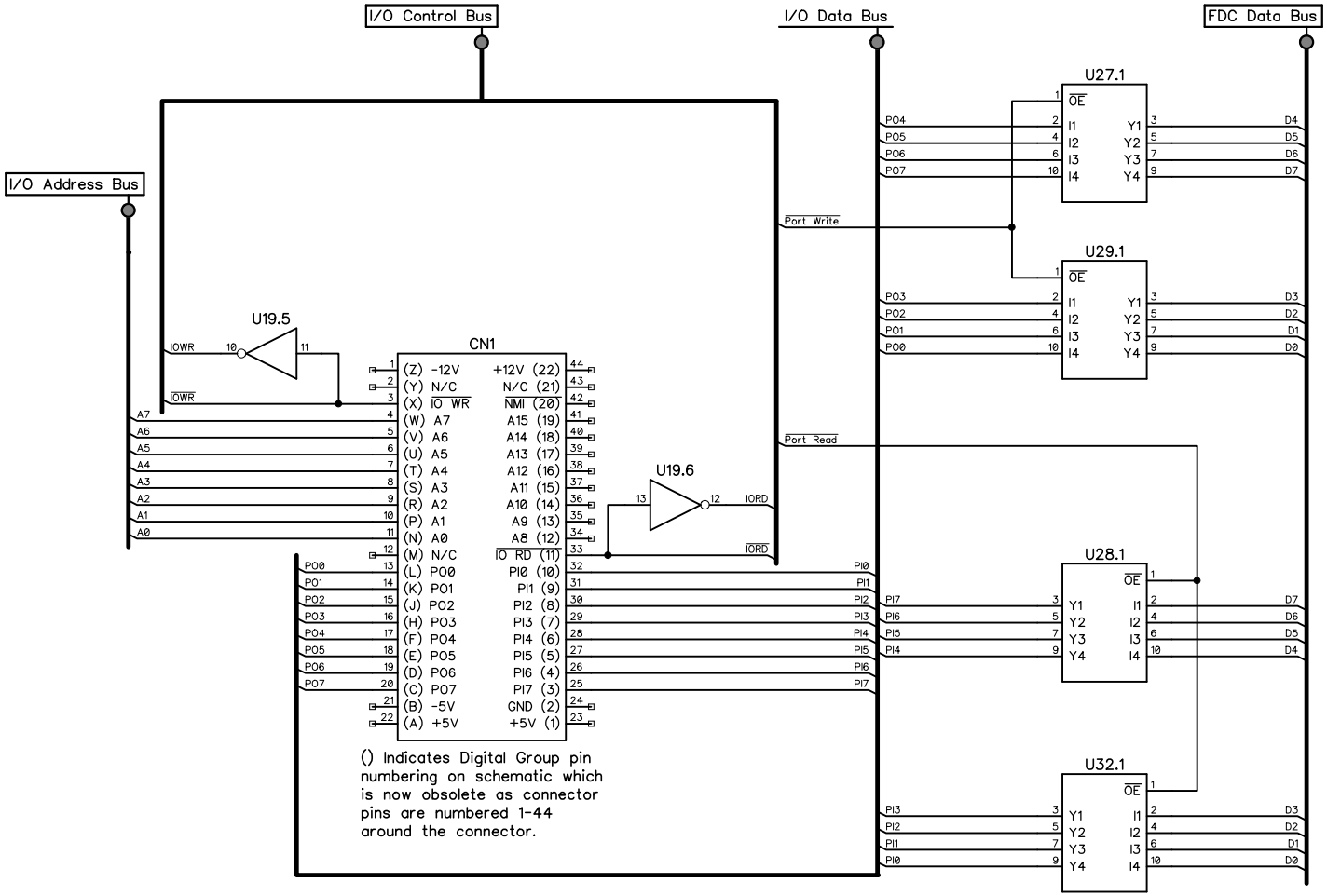
nc

nc

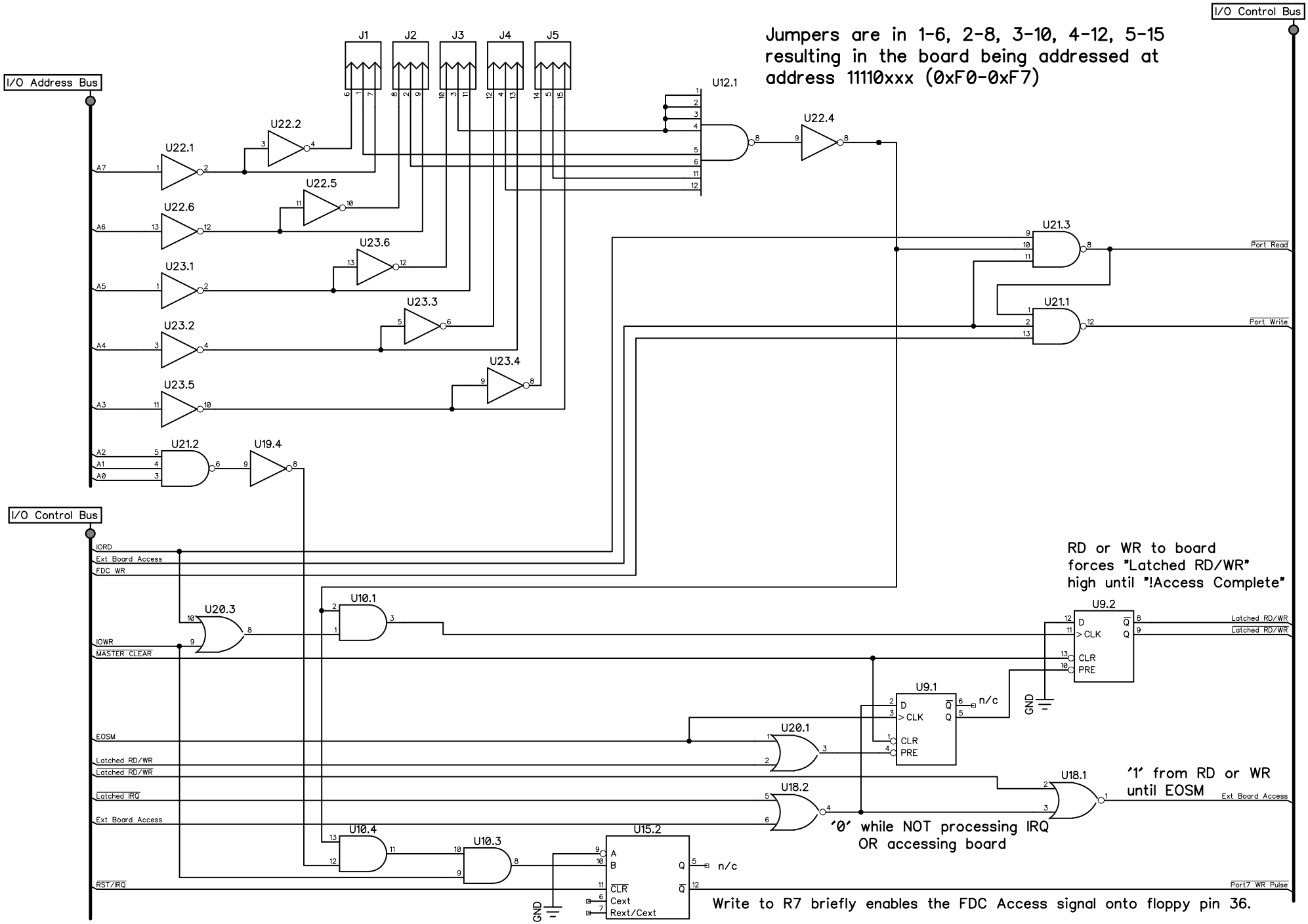
Floppy WR Line

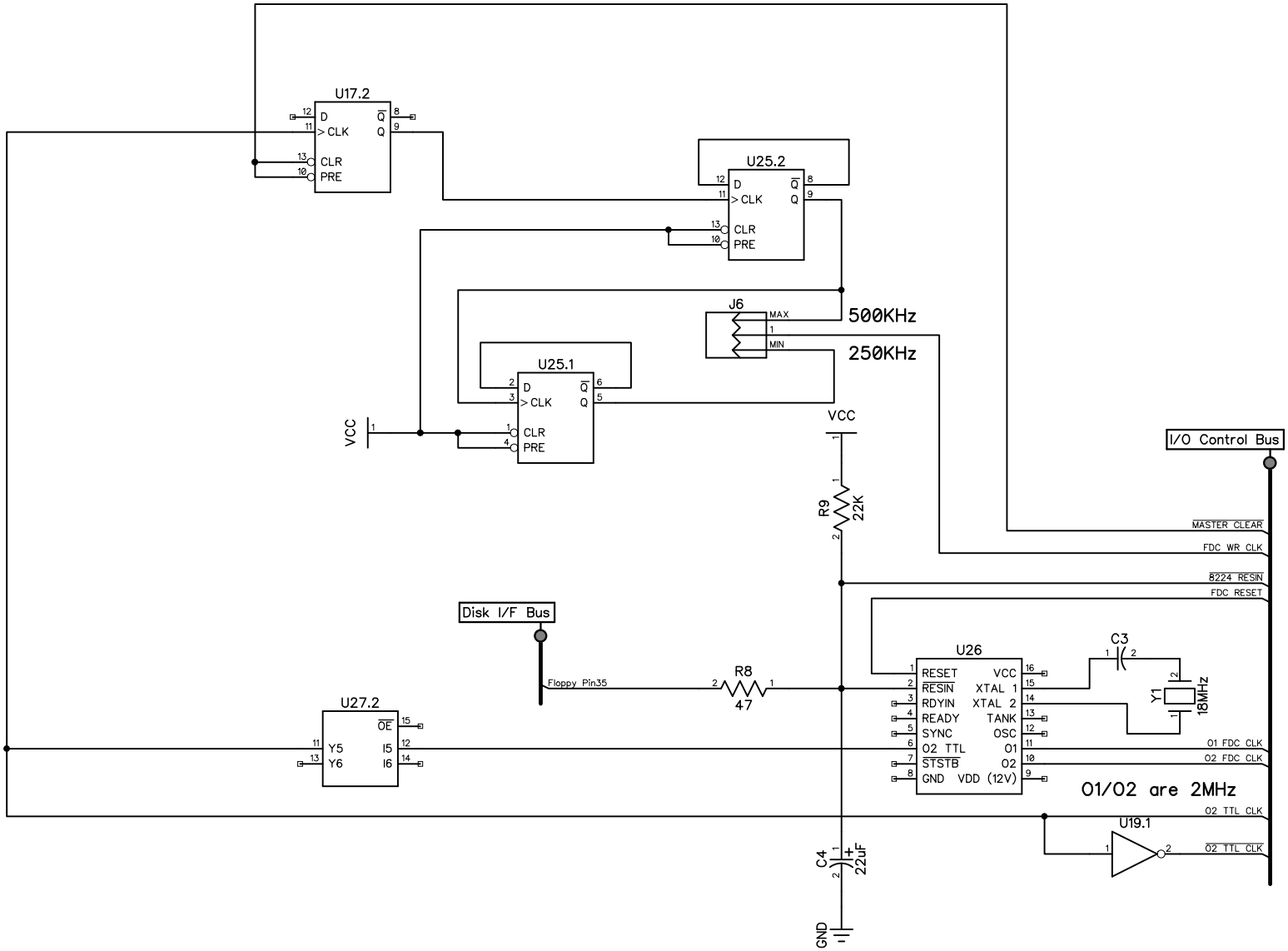
GND

Data Conditioner

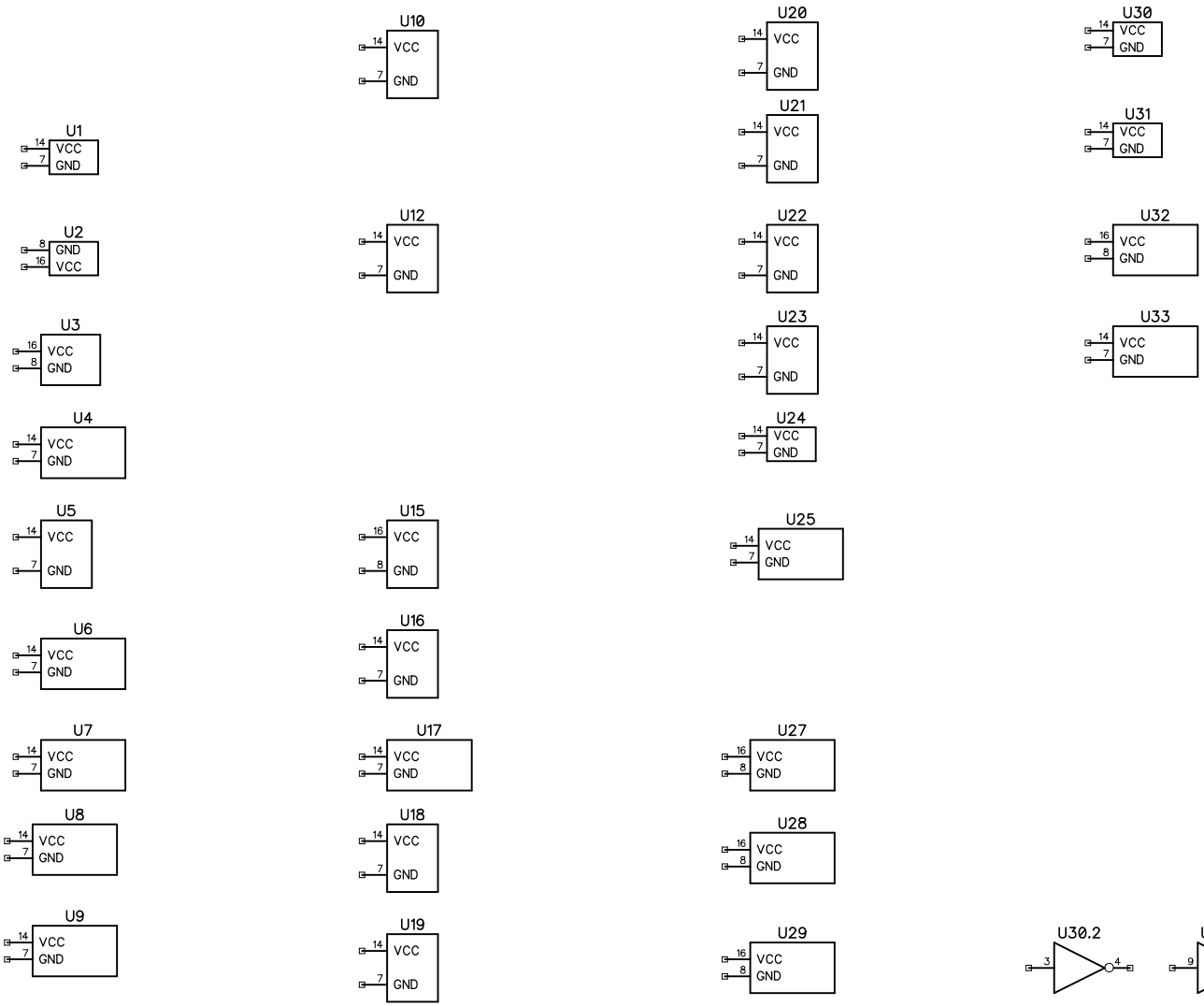


CPU Interface Logic

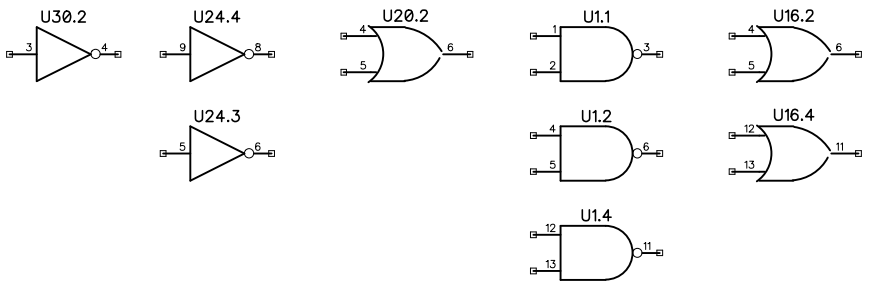

I/O Control Bus
Jumpers are in 1-6, 2-8, 3-10, 4-12, 5-15
resulting in the board being addressed at
address 11110xxx (0xF0-0xF7)
I/O Address Bus
J1
J2
J3
J4
J5
U12.1
U22.1
U22.2
U22.4
U22.5
U22.6
U23.1
U23.2
U23.3
U23.4
U23.5
U23.6
U21.1
U21.2
U21.3
U19.4
A7
A6
A5
A4
A3
A2
A1
A0
Port Read
Port Write
I/O Control Bus
IORD
Ext Board Access
FDC WR
IOWR
MASTER CLEAR
EOSM
Latched RD/WR
Latched RD/WR
Latched IRQ
Ext Board Access
RST/IRQ
U20.3
U10.1
U10.4
U10.3
U15.2
U9.1
U9.2
U20.1
U18.2
U18.1
A
B
CLR
Cext
Rext/Cext
Q
n/c
D
CLK
PRE
GND
RD or WR to board
forces "Latched RD/WR"
high until "!Access Complete"
'1' from RD or WR
until EOSM
'0' while NOT processing IRQ
OR accessing board
Write to R7 briefly enables the FDC Access signal onto floppy pin 36.
Port7 WR Pulse
Address Select Logic


U17.2
D
> CLK
CLR
PRE
Q
U25.2
U25.1
J6
MAX
MIN
500KHz
250KHz
VCC
R9
22K
I/O Control Bus
MASTER CLEAR
FDC WR CLK
8224 RESIN
FDC RESET
Disk I/F Bus
Floppy Pin35
R8
47
U26
RESET
RESIN
RDYIN
READY
SYNC
O2 TTL
STSTB
GND
VDD (12V)
XTAL 1
XTAL 2
TANK
OSC
O1
O2
C3
Y1
18MHz
O1 FDC CLK
O2 FDC CLK
O1/O2 are 2MHz
O2 TTL CLK
U19.1
C4
22uF
GND
U27.2
OE
Y5
Y6
I5
I6

Clock Logic

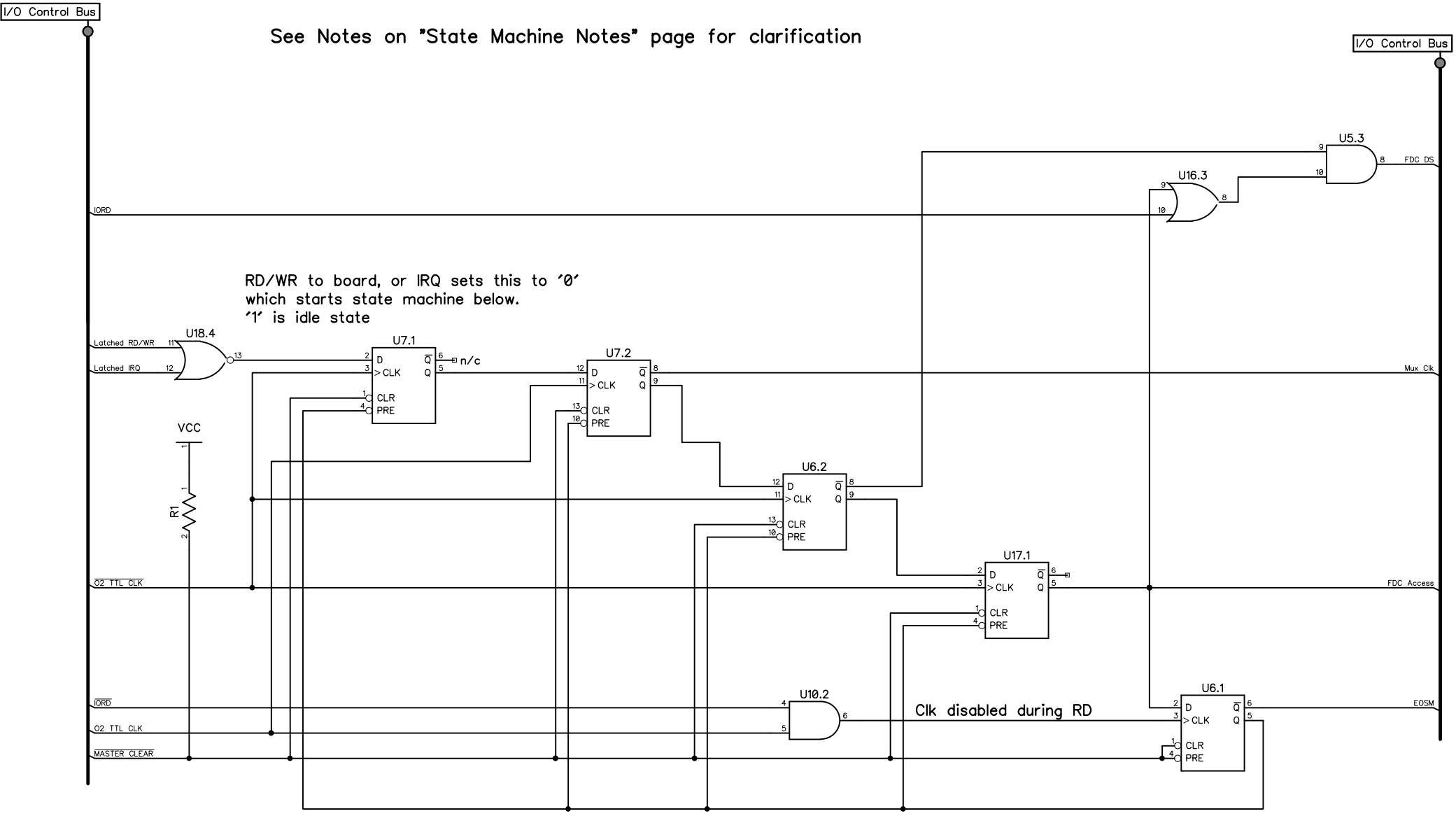


Unused Gates



Power Logic

It does not appear that anything drives !MASTER CLEAR
so it is always high from it's pull-up resistor.



State machine that walks a '0' bit through the 5 stages and when it gets to the last stage
it causes the state machine to reset to all '1's

State Machine Logic

NOTE: The logic condition which places uPD372 READ Register information on the Data Bus is DS·$\overline{W/R}$·$\overline{RS2}$. If these three signals are allowed to change asynchronously with respect to each other, care must be taken to insure that this condition does not become true inadvertantly. The simplest method is to require that W/R and RS2 must not change state while DS is a logic one.

NOTE: RS0 and RS1 must not change state during the period from 150ns before until 10ns after the trailing edge of Ø1 or else register contents and DISK DRIVE COMMANDS may be modified. A simple method to accomplish this is to use a dual flip-flop to synchronize changes in RS0 and RS1 with the leading edge of Ø2(TTL) as shown in Figure 10.

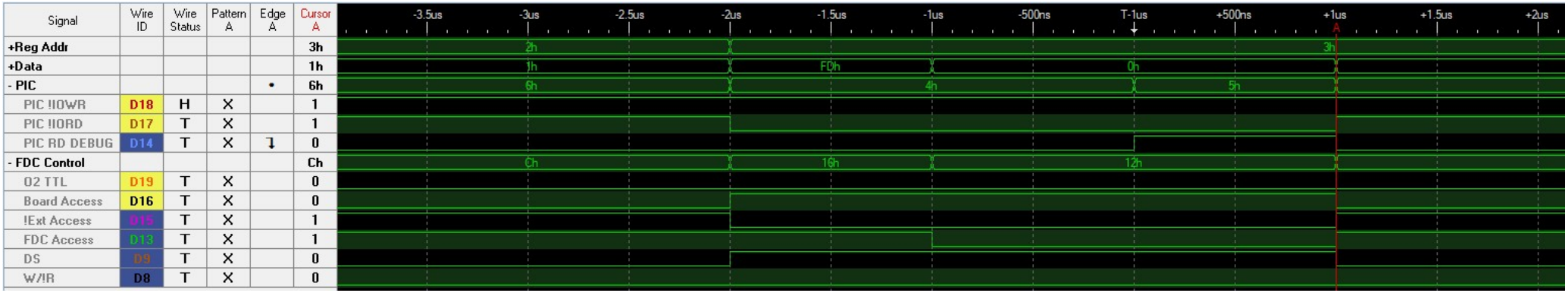
This state machine is needed to satisfy the following uPD372 timing requirements (extracted from the datasheet.)

When reading FDC registers, the host should:
a) setup address
b) drive IORD line low
b) wait until FDC Acess goes low
c) read the value from the data bus
d) drive IORD line high

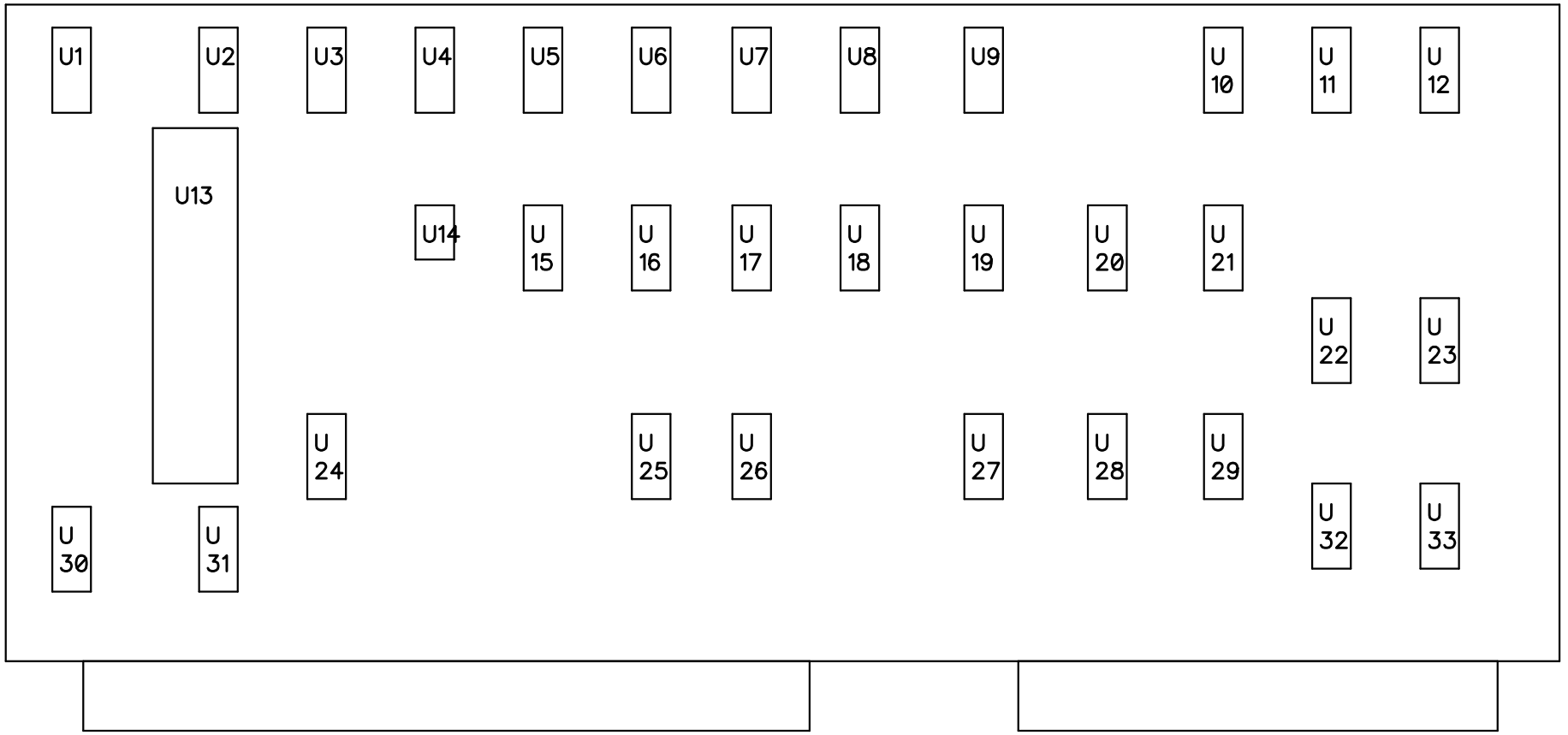
When checking for interrupts, the host should:
a) setup address to Reg 7
b) drive IOWR low
c) check FDC Acess line (high = interrupt)
d) drive IOWR high

When writing FDC registers, the host should:
a) setup address & data
b) drive IOWR low
c) wait until FDC Access goes low
d) drive IOWR high

Example Read Cycle Timing



State Machine Notes



Assembly Drawing

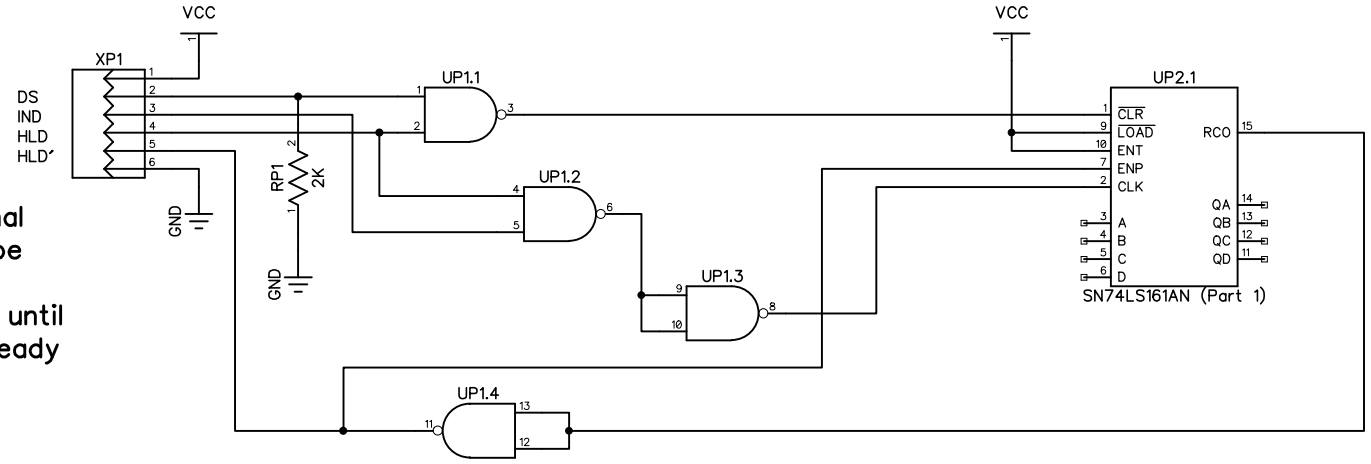
Prototype wire-wrap board attached to back of Digital Group FDC controller PCB

Wired into various points
on DG FDC PCB

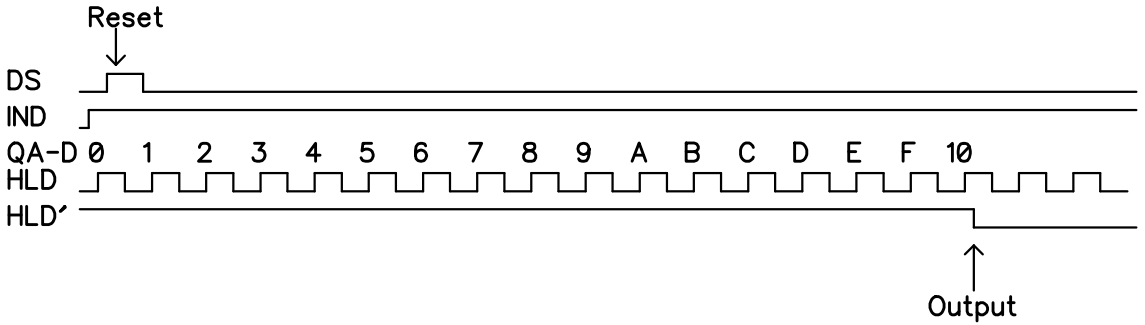
Likely used to hold Z80 in WAIT until FDC read data was ready, OR something with Head Load - since it uses "HLD" as a signal name, and there is evidence on the FDC PCB that the following traces were cut: Head Load (1 cut) & IORD (2 cuts)

This was likely used as a way to insert wait states into the Z80 access cycle.

DS is probably the FDC DS signal
IND is probably the INData strobe
HLD may be the phase 2 clock
HLD′ may hold the Z80 in wait until the FDC data register is ready to be read



HLD′ used to be labelled "WAIT" (under 2nd label)



DS appears to reset this feature, and must go low before count can proceed
IND appears to be used to 'gate' the HLD pulses
HLD appears to be some clock source
HLD′ appears to be the modified version of the clock source

NB: HLD′ stays low until DS re-asserted (it does NOT keep cycling)

Sockets, but not populated and no connections