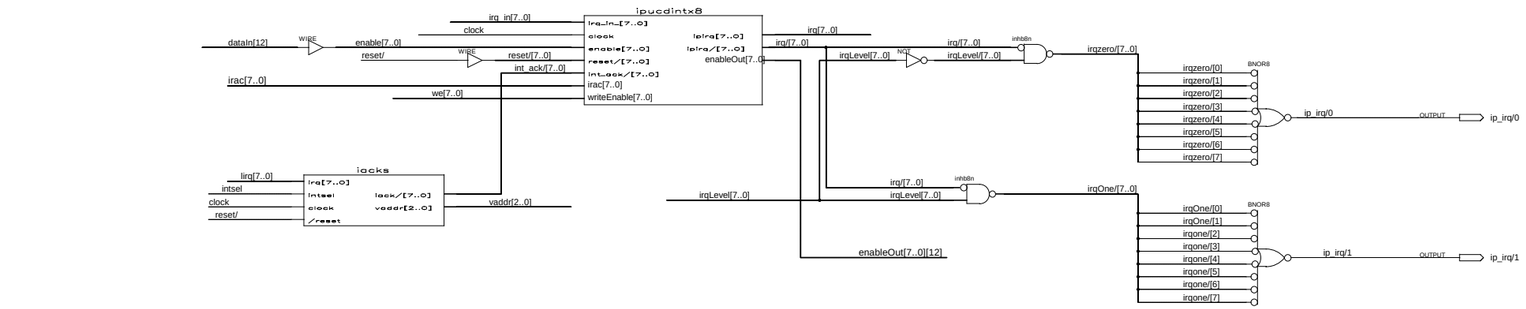
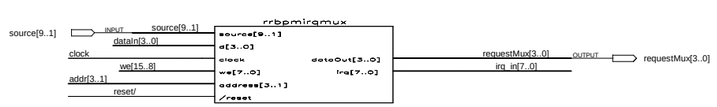
GPIP 8 Frame MDAT Transmitter

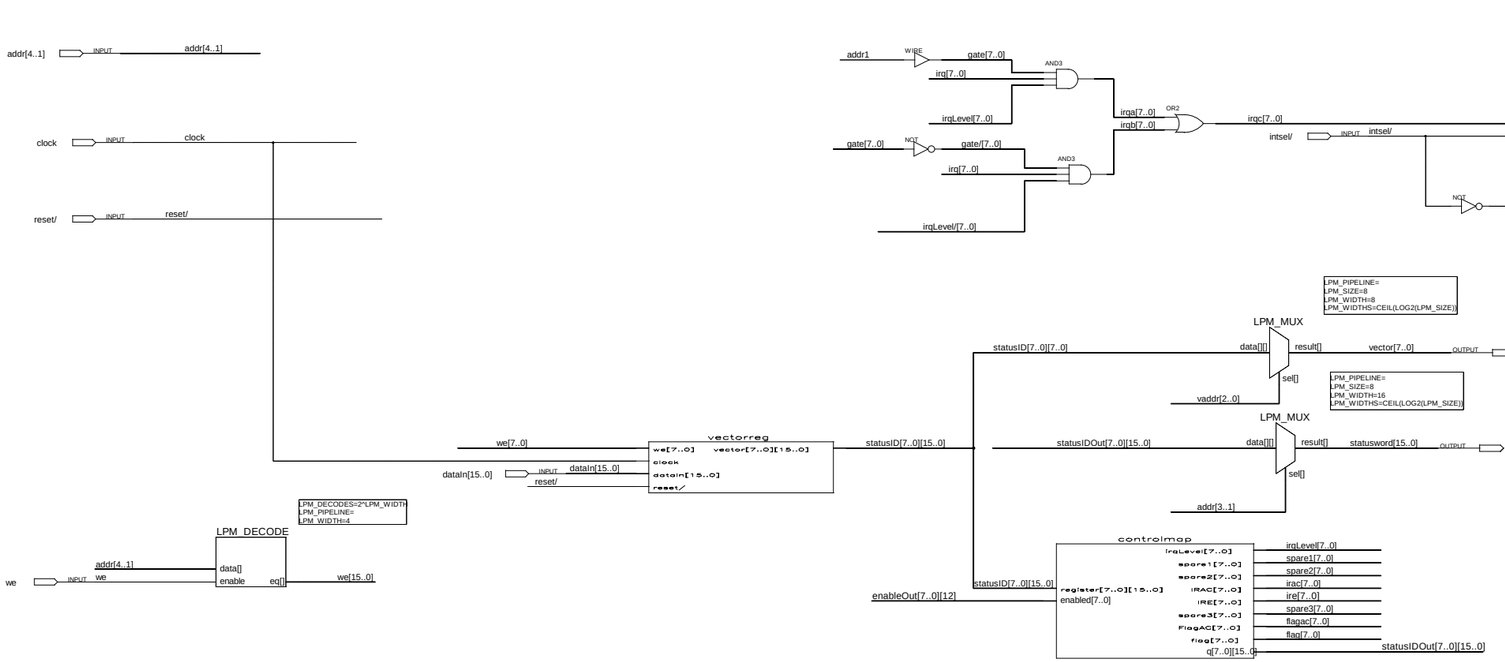
COMPANY	Fermilab BD/AS-Controls
DESIGNER	Mike Kuplic
SIZE	F
NUMBER	1.00
REV	A
DATE	11:35a 3-07-2000
SHEET	1 OF 1

```
TITLE "Decoder";

SUBDESIGN Decoder
(
    idsel/,
    iosel/,
    memsel/,
    intsel/,
    latchenable     : INPUT;

    idselL,
    ioselL,
    memselL,
    intselL,
    valid_access,
    Select[1..0] : OUTPUT;

)
VARIABLE

idselL,ioselL,memselL,intselL:  LATCH;
idsel,iosel,memsel,intsel:NODE;


BEGIN

idselL.ENA = latchenable;
ioselL.ENA = latchenable;
memselL.ENA = latchenable;
intselL.ENA = latchenable;

    TABLE
        idsel/,iosel/,memsel/,intsel/   =>  Select[1..0];


        0,  1,  1,  1       =>  b"00";
        1,  0,  1,  1       =>  b"01";

        1,  1,  0,  1       =>  b"10";

        1,  1,  1,  0       =>  b"11";

    END TABLE;

idsel =  !idsel/ AND iosel/ AND memsel/ AND intsel/;
iosel =  idsel/ AND !iosel/ AND memsel/ AND intsel/;
memsel =  idsel/ AND iosel/ AND !memsel/ AND intsel/;
intsel =  idsel/ AND iosel/ AND memsel/ AND !intsel/;

valid_access = idsel OR iosel OR memsel OR intsel;

idselL =  idsel;
ioselL =  iosel;
memselL =  memsel;
intselL =  intsel;

END;
```

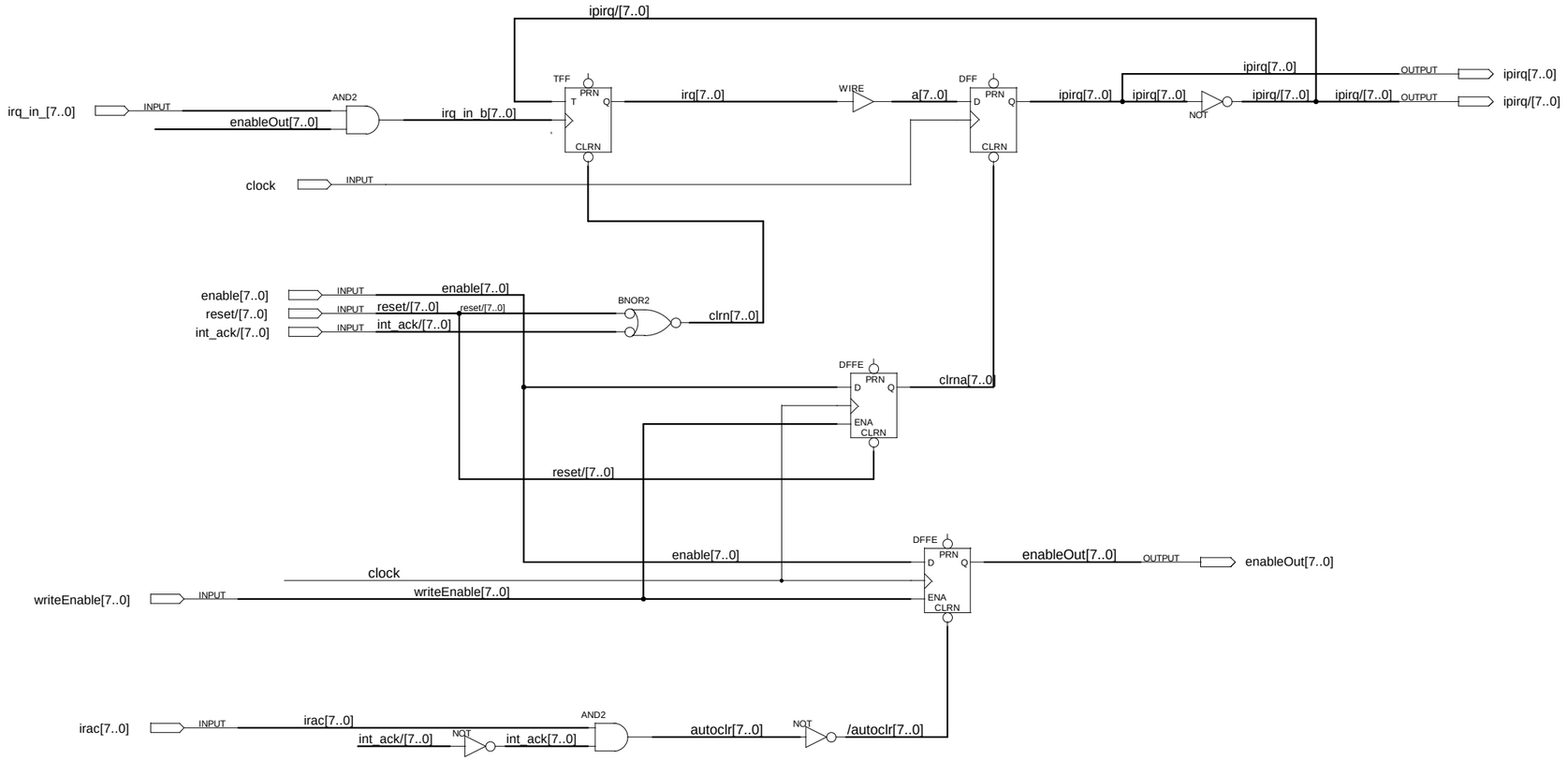
```
TITLE "Chip Selects for the IP UCD";

SUBDESIGN ipUcdCs
(
    idsel,
    iosel,
    memsel,
    intsel,
    Addr[18..1]
                            : INPUT;

    CS[14..0],
    OutputMuxSelect[3..0] : OUTPUT;

)

VARIABLE

CS[14..0]:      NODE;


BEGIN

CS0 =   iosel & (addr[6..1] >= 0 & addr[6..1] <= h"1F");        %8 Frame MDAT Transmitter%
CS1 =   iosel & addr[6..1] == h"20";                            %Serial data read%
CS2 =   iosel & addr[6..1] == h"21";                            %Status%
CS3 =   iosel & addr[6..1] == h"22";                            %%
CS4 =   iosel & addr[6..1] == h"23";                            %%
CS5 =   iosel & addr[6..1] == h"24";                            %%
CS6 =   iosel & addr[6..1] == h"26";                            %%
CS7 =   iosel & addr[6..1] == h"27";                            %%
CS8 =   iosel & addr[6..1] == h"28";                            %%
CS9 =   iosel & addr[6..1] == h"29" ;                           %%
CS10 =  iosel & (addr[6..1] >= h"30" & addr[6..1]<=h"37");      %Interrupt sontrol registers%
CS11 =  iosel & (addr[6..1] >= h"38" & addr[6..1]<=h"3f");      %Interrupt Source Mux%
CS12 =  memsel & addr[18..1]>= h"00000" & addr[18..1]<=h"3FFFF";%General Memory 128K words%
CS13 =  idsel;                                                  %Read ID ROM%
CS14 =  intsel;                                                 %Read interrupt vector%

TABLE
    CS[14..0]           =>  OutputMuxSelect[3..0];

    b"000000000000000"  => b"XXXX";
    b"000000000000001"  => h"0";
    b"000000000000010"  => h"1";
    b"000000000000100"  => h"2";
    b"000000000001000"  => h"3";
    b"000000000010000"  => h"4";
    b"000000000100000"  => h"5";
    b"000000001000000"  => h"6";
    b"000000010000000"  => h"7";
    b"000000100000000"  => h"8";
    b"000001000000000"  => h"9";
    b"000010000000000"  => h"A";
    b"000100000000000"  => h"B";
    b"001000000000000"  => h"C";
    b"010000000000000"  => h"D";
    b"100000000000000"  => h"E";
END TABLE;

END;
```



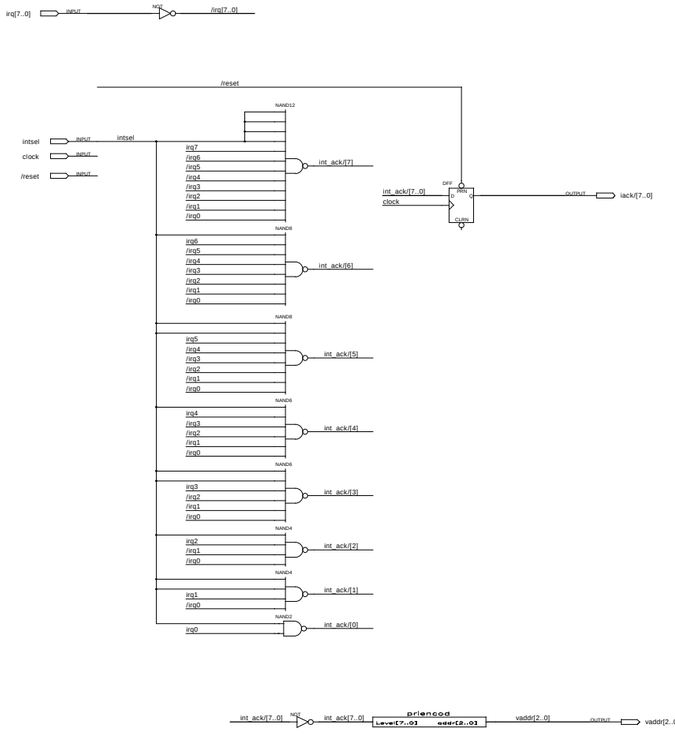


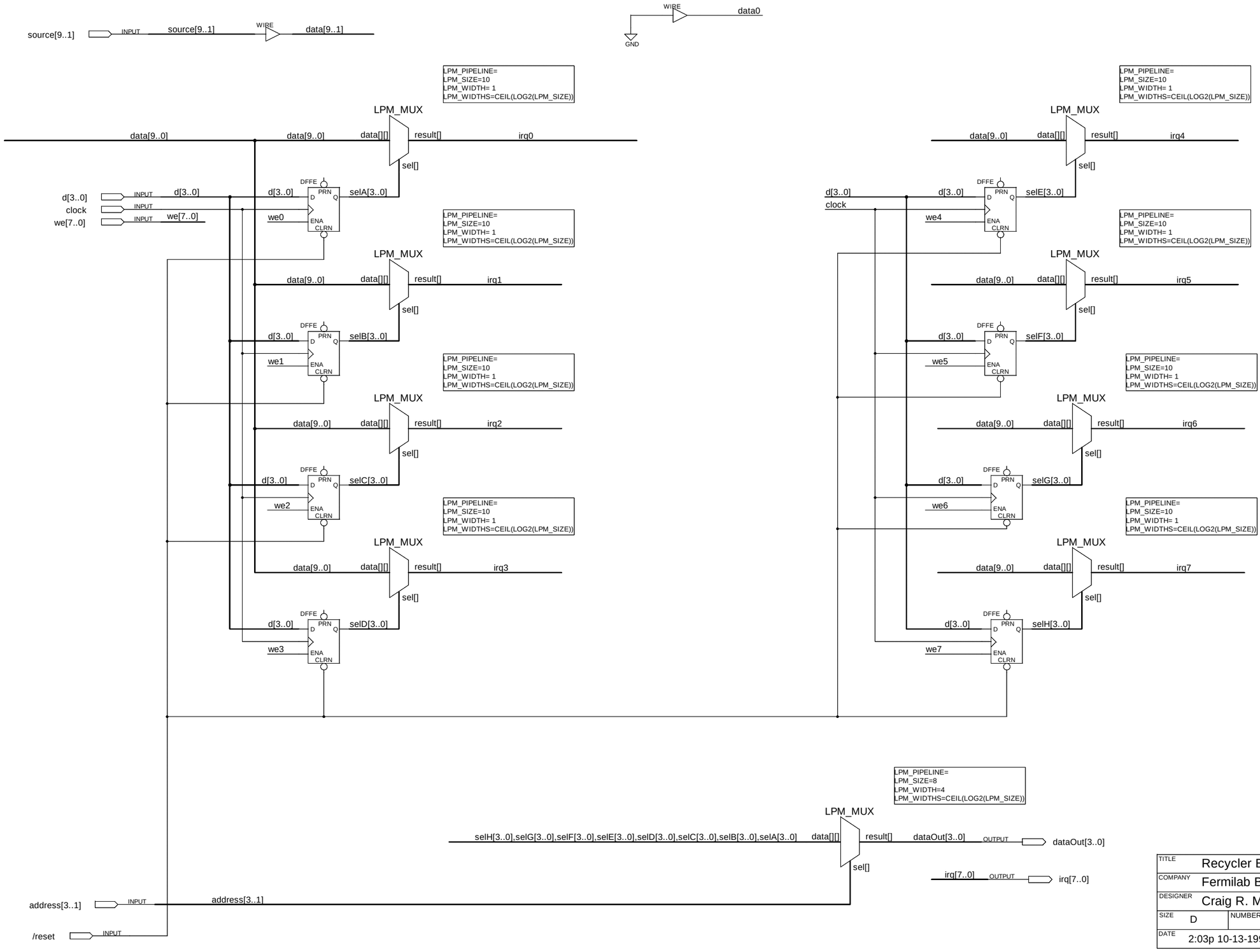


TITLE	IP UCD Interrrupts		
COMPANY	Fermilab BD/AS-Controls		
DESIGNER	Craig McClure		
SIZE E	NUMBER 1.00	REV A	
DATE 3:24p 2-10-1999	SHEET 1	OF 1	



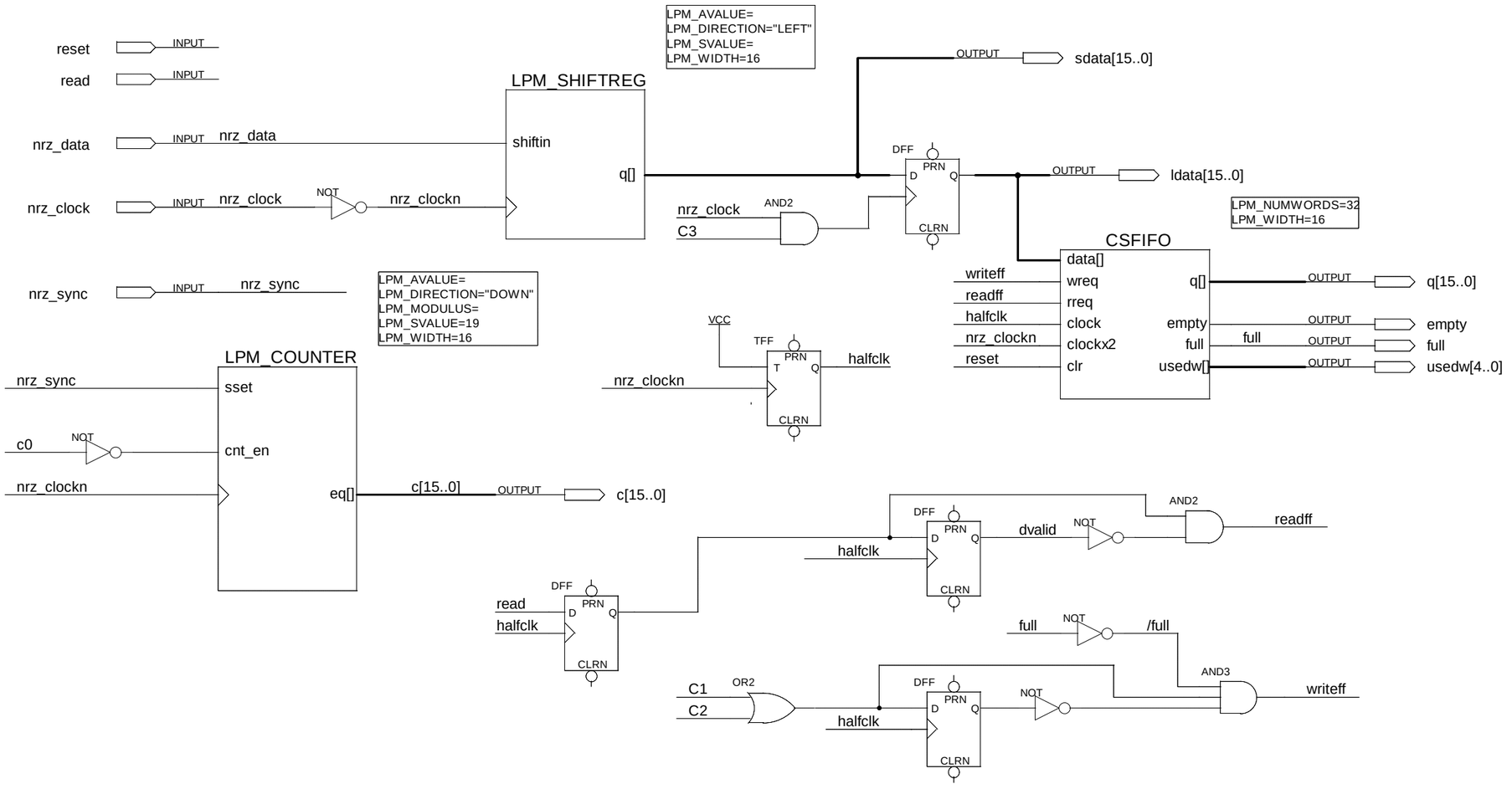
TITLE	IP Interrupts		
COMPANY	Fermilab BD/AS-Controls		
DESIGNER	Craig R. McClure		
SIZE D	NUMBER 1.00	REV A	
DATE 2:53p 12-03-1998	SHEET 1 OF 1		


irq[7..0]
input
NOT
/irq[7..0]
/reset
intsel
input
clock
input
/reset
input
intsel
NAND12
/irq7
/irq6
/irq5
/irq4
/irq3
/irq2
/irq1
/irq0
int_ack/[7]
NAND8
/irq6
/irq5
/irq4
/irq3
/irq2
/irq1
/irq0
int_ack/[6]
NAND8
/irq5
/irq4
/irq3
/irq2
/irq1
/irq0
int_ack/[5]
NAND5
/irq4
/irq3
/irq2
/irq1
/irq0
int_ack/[4]
NAND6
/irq3
/irq2
/irq1
/irq0
int_ack/[3]
NAND4
/irq2
/irq1
/irq0
int_ack/[2]
NAND4
/irq1
/irq0
int_ack/[1]
NAND2
/irq0
int_ack/[0]
int_ack/[7..0]
clock
DFF
PRN
D
Q
CLRN
output
iack/[7..0]
int_ack/[7..0]
NOT
int_ack[7..0]
priencod
Level7..0
eed<2..0
vaddr[2..0]
output
vaddr[2..0]

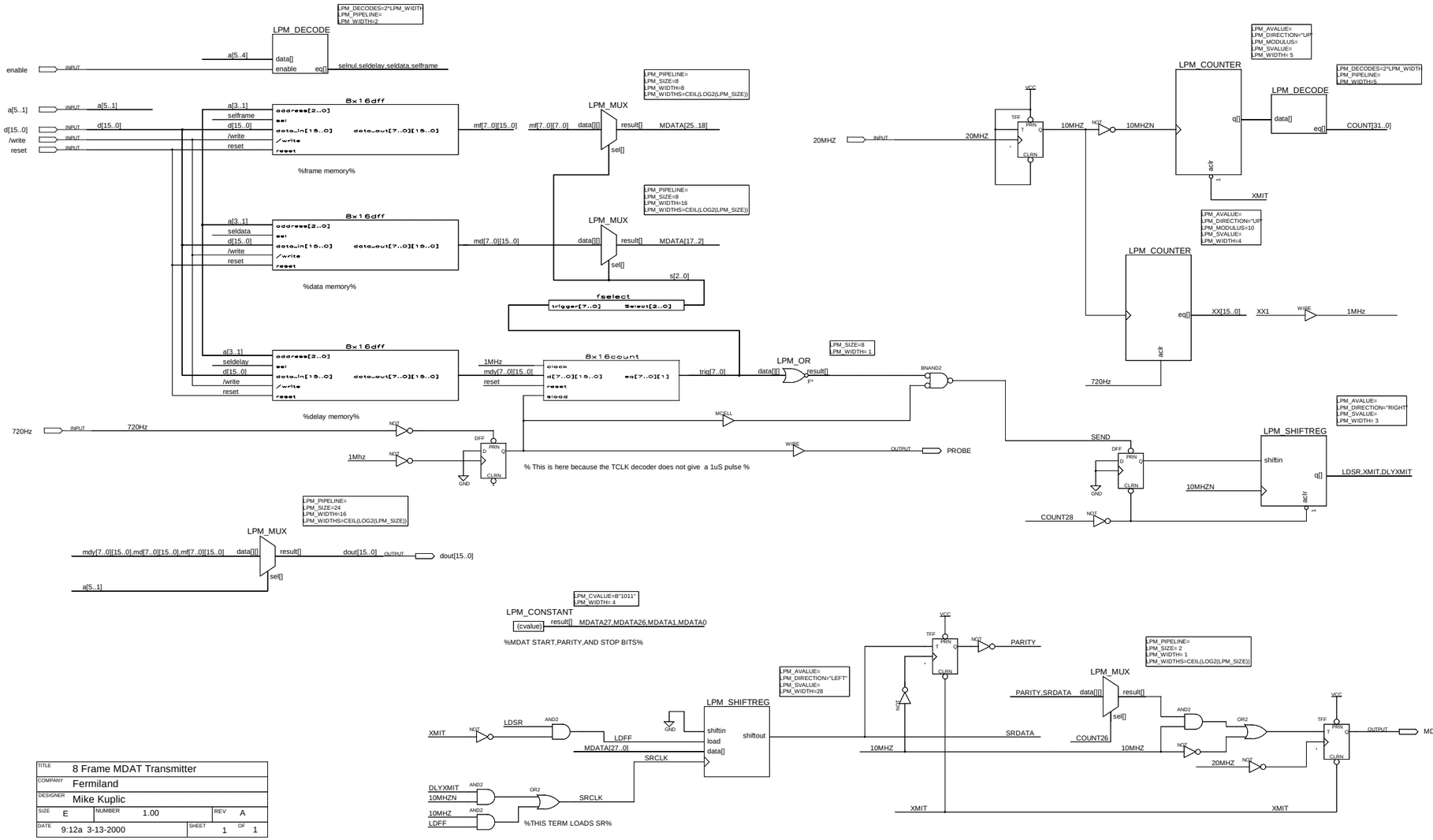


TITLE	Recycler BPM IP IRQ Mux		
COMPANY	Fermilab BD/AS-Controls		
DESIGNER	Craig R. McClure		
SIZE D	NUMBER 1.00	REV A	
DATE 2:03p 10-13-1998	SHEET 1 OF 1		

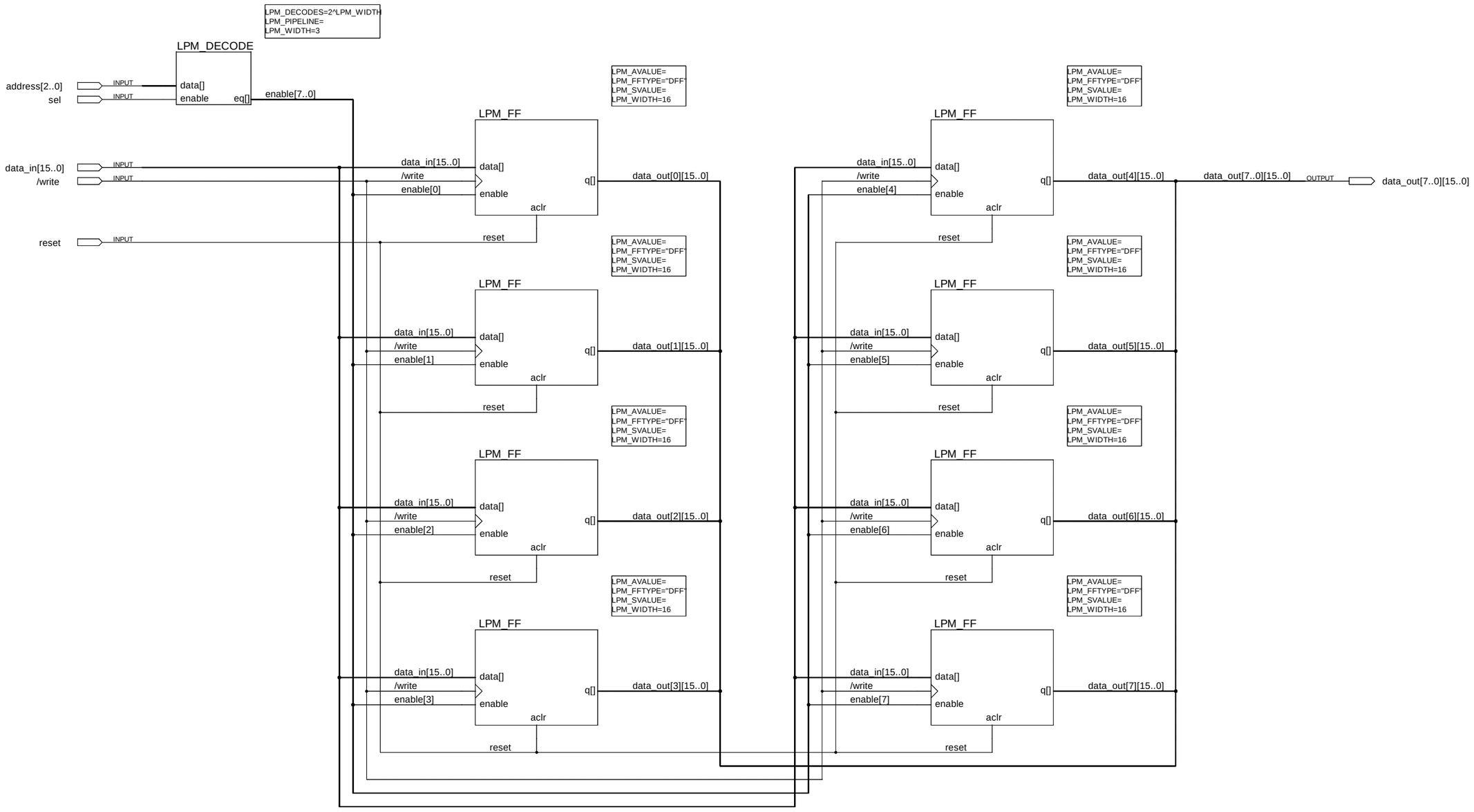
```
TITLE "Write condition";

SUBDESIGN write_stb
(
    iosel,
    rw/,
    ipclock,
    /ipreset                            : INPUT;

    write_strobe,
    write_gate: OUTPUT;

)

VARIABLE

state[1..0]:    dff;


BEGIN
state[].clk     = !ipclock;
state[].clrn    = /ipreset;

TABLE
iosel,  rw/,    state[1..0] => state[1..0];
0,      x,      b"00"       =>  b"00";
1,      1,      b"00"       =>  b"00";
1,      0,      b"00"       =>  b"01";
x,      x,      b"01"       =>  b"11";
x,      x,      b"11"       =>  b"10";
1,      x,      b"10"       =>  b"10";
0,      x,      b"10"       =>  b"00";
END TABLE;

write_strobe = state[1] & state[0];
write_gate = state[0];

END;
```

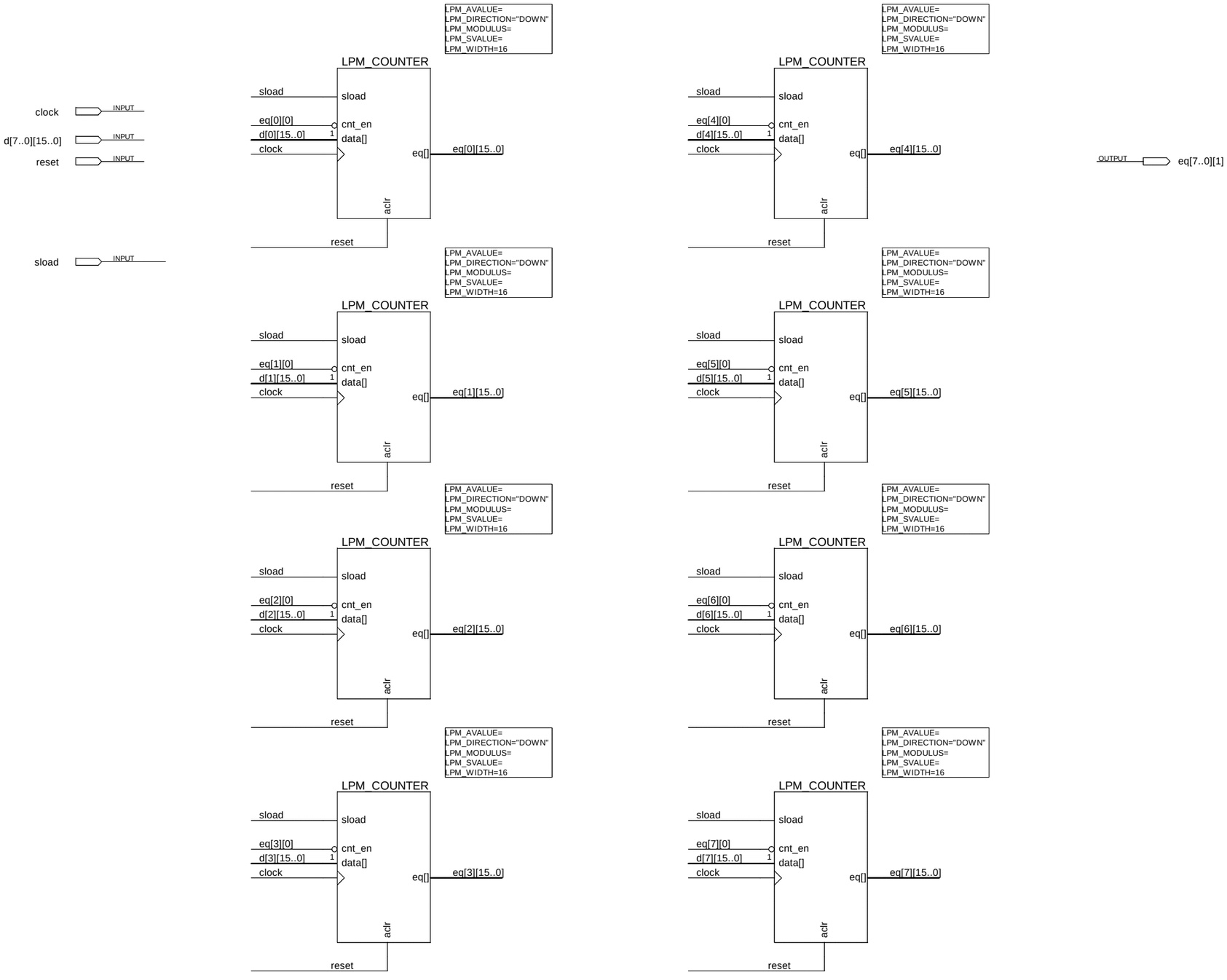


TITLE	NRZ Decoder		
COMPANY	Fermiland		
DESIGNER	Mike Kuplic		
SIZE C	NUMBER 1.00	REV A	
DATE 11:46a 9-24-1999	SHEET 1	OF 1	

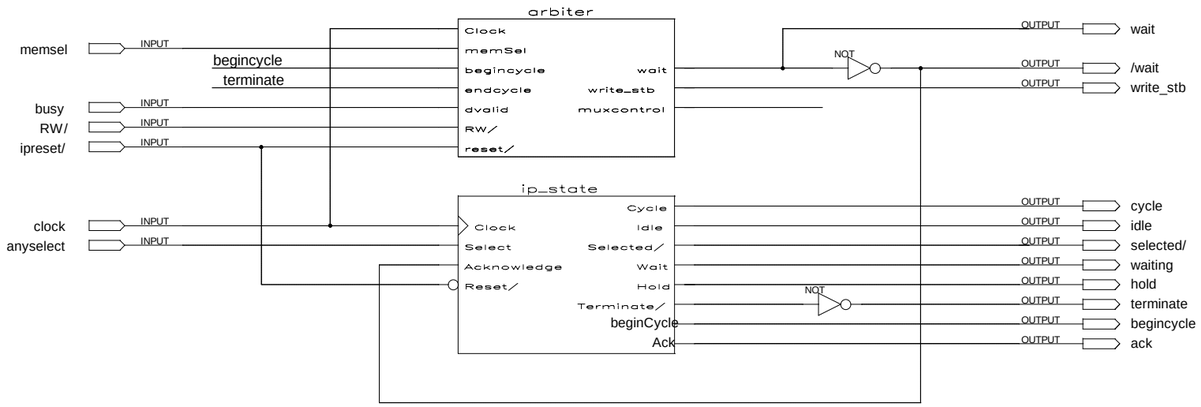


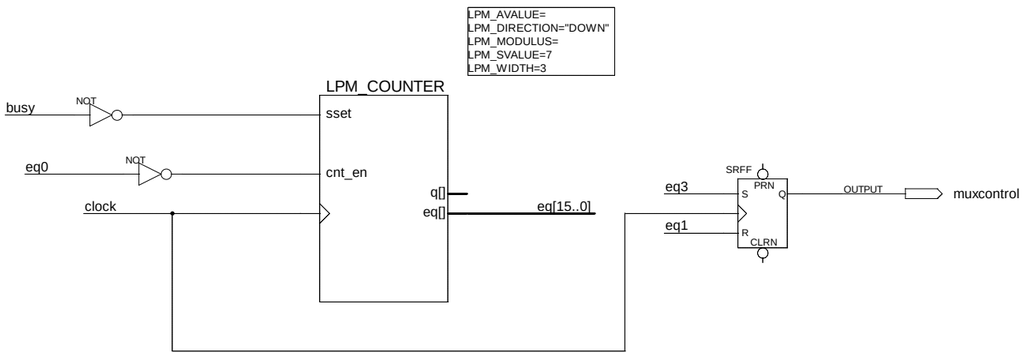
TITLE	8 Frame MDAT Transmitter			
COMPANY	Fermiland			
DESIGNER	Mike Kuplic			
SIZE	E	NUMBER 1.00	REV	A
DATE	9:12a 3-13-2000	SHEET 1	OF	1


LPM_DECODES=2^LPM_WIDTH
LPM_PIPELINE=
LPM_WIDTH=3
LPM_DECODE
address[2..0]
sel
INPUT
data[]
enable
eq[]
enable[7..0]
LPM_AVALUE=
LPM_FFTYPE="DFF"
LPM_SVALUE=
LPM_WIDTH=16
LPM_FF
data_in[15..0]
/write
reset
data[]
enable
aclr
q[]
enable[0]
data_out[0][15..0]
enable[1]
data_out[1][15..0]
enable[2]
data_out[2][15..0]
enable[3]
data_out[3][15..0]
enable[4]
data_out[4][15..0]
enable[5]
data_out[5][15..0]
enable[6]
data_out[6][15..0]
enable[7]
data_out[7][15..0]
data_out[7..0][15..0]
OUTPUT


clock
INPUT
d[7..0][15..0]
INPUT
reset
INPUT
sload
INPUT
LPM_COUNTER
sload
cnt_en
data[]
clock
eq[]
aclr
reset
eq[0][15..0]
eq[1][15..0]
eq[2][15..0]
eq[3][15..0]
eq[4][15..0]
eq[5][15..0]
eq[6][15..0]
eq[7][15..0]
eq[0][0]
eq[1][0]
eq[2][0]
eq[3][0]
eq[4][0]
eq[5][0]
eq[6][0]
eq[7][0]
d[0][15..0]
d[1][15..0]
d[2][15..0]
d[3][15..0]
d[4][15..0]
d[5][15..0]
d[6][15..0]
d[7][15..0]
1
LPM_AVALUE=
LPM_DIRECTION="DOWN"
LPM_MODULUS=
LPM_SVALUE=
LPM_WIDTH=16
OUTPUT
eq[7..0][1]

```
TITLE "Fselect";

SUBDESIGN Fselect
(
    trigger[7..0]   : INPUT;
    Select[2..0] : OUTPUT;
)

BEGIN

    TABLE

        trigger[7..0]   =>  Select[2..0];

        b"00000001"     => 0;
        b"00000010"     => 1;
        b"00000100"     => 2;
        b"00001000"     => 3;
        b"00010000"     => 4;
        b"00100000"     => 5;
        b"01000000"     => 6;
        b"10000000"     => 7;

    END TABLE;
END;
```


arbiter
Clock
memSel
begincycle
endcycle
dvalid
RW/
reset/
wait
write_stb
muxcontrol
memsel
INPUT
begincycle
terminate
busy
RW/
ipreset/
NOT
OUTPUT
wait
/wait
write_stb
ip_state
Cycle
Idle
Selected/
Wait
Hold
Terminate/
beginCycle
Ack
Clock
Select
Acknowledge
Reset/
clock
anyselect
cycle
idle
selected/
waiting
hold
terminate
begincycle
ack


LPM_AVALUE=
LPM_DIRECTION="DOWN"
LPM_MODULUS=
LPM_SVALUE=7
LPM_WIDTH=3
LPM_COUNTER
busy
NOT
sset
eq0
cnt_en
clock
q[]
eq[]
eq[15..0]
SRFF
eq3
eq1
PRN
S
R
CLRN
Q
OUTPUT
muxcontrol

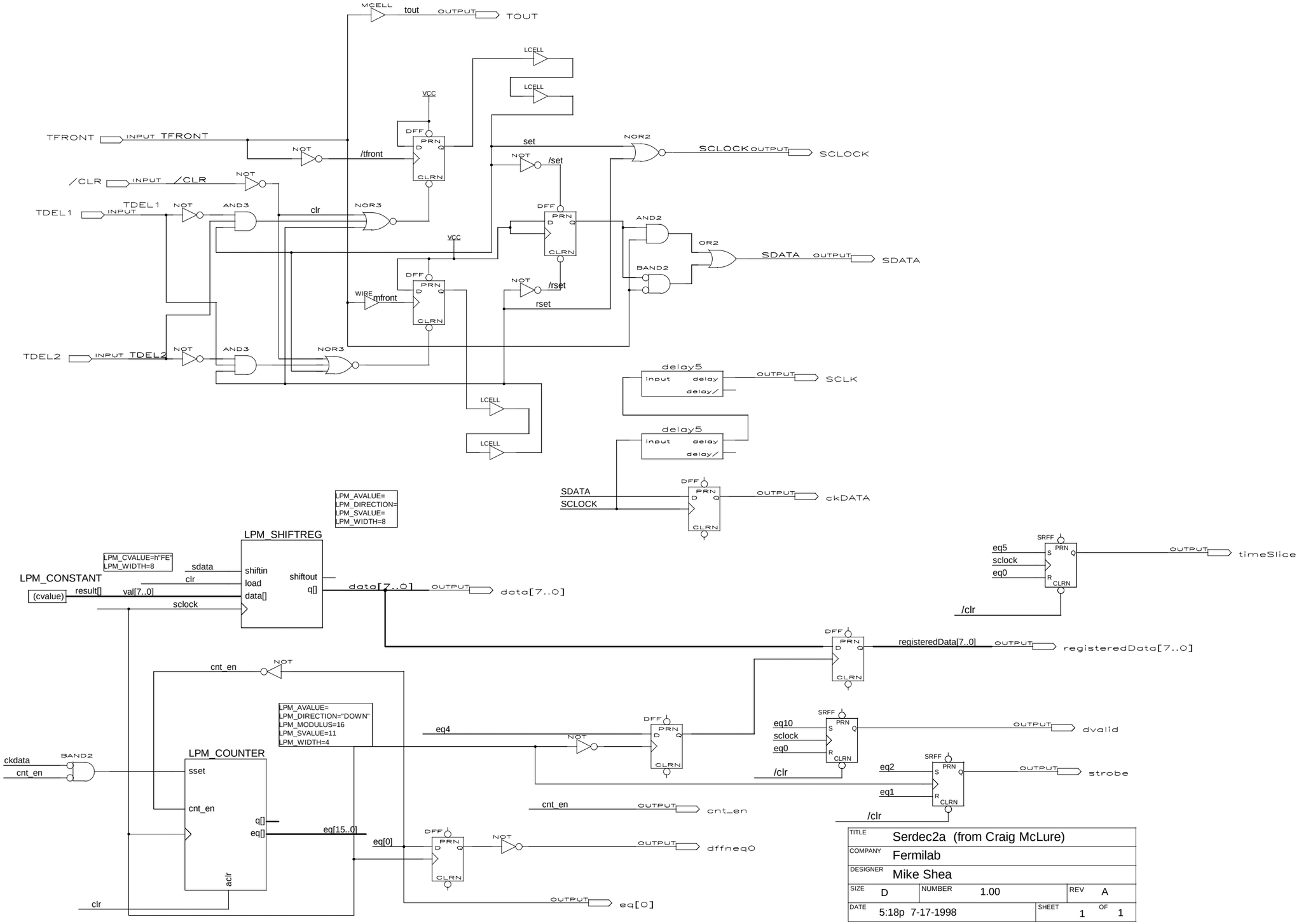
```
TITLE "Arbiter";

SUBDESIGN Arbiter
(
    Clock,
    memSel,
    begincycle,
    endcycle,
    dvalid,
    RW/,
    reset/      : INPUT;

wait,
write_stb,
muxcontrol          : OUTPUT;
)

VARIABLE
STATE[1..0]:    DFF;
count[3..0]:    DFF;
wait        :   SRFF;

BEGIN
STATE[].CLK= CLOCK;
STATE[].CLRN = reset/;
count[].clk= clock;
count[].clrn = reset/;
wait.clk = clock;
wait.clrn = reset/;


wait.s = begincycle;

    TABLE

begincycle, endcycle,   memsel, dvalid, state[] =>  state[];
0,          0,          X,      X,      h"0"    =>  H"0";
1,          0,          0,      X,      h"0"    =>  H"1";
1,          0,          1,      0,      h"0"    =>  H"1";
1,          0,          1,      1,      h"0"    =>  H"2";
0,          0,          x,      x,      h"1"    =>  H"1";
x,          x,          x,      x,      h"1"    =>  H"3";
x,          x,          x,      x,      h"2"    =>  H"3";
0,          0,          x,      x,      h"3"    =>  H"3";
0,          1,          x,      x,      h"3"    =>  H"0";

END TABLE;

If state[]==2 then
    count[] = 10;
ELSIF state[]==1 then
    count[] = 2;
ELSIF count[] > 0 then
    count[] = count[] -1;
ELSE count[] = 0;


END IF;

muxcontrol = count[] >= 0 and count[] <= 5;
wait.r = (count[] == 1);
write_stb = count[] == 2 AND !RW/;


END;
```

```
TITLE "Ip_State";

SUBDESIGN Ip_State
(
    Clock,
    Select,
    Acknowledge,
    Reset/      : INPUT;

    Cycle,
    Idle,
    Selected/,
    Wait,
    Hold,
    Terminate/,
    BeginCycle,
    Ack         : OUTPUT;
)

VARIABLE


Idle,
Wait,
Hold            : NODE;
State[2..0]     :DFF;
Cycle           :SRFF;
selected/,terminate/,BeginCycle,EndCycle        :LCELL;

BEGIN
    State[].clrn = Reset/;
    State[].clk = Clock;
    Cycle.CLRN = Reset/;
    Cycle.S = BeginCycle;
    Cycle.R = EndCycle;
    Cycle.CLK = Clock;

TABLE
    Select, Acknowledge,    State[2..0] =>  State[2..0],BeginCycle,EndCycle;
    0,      x,          b"000"  =>  b"000",0,0;
    1,      x,          b"000"  =>  b"001",1,0;
    0,      1,          b"001"  =>  b"110",0,0;
    1,      1,          b"001"  =>  b"111",0,0;
    x,      0,          b"001"  =>  b"101",0,0;
    0,      x,          b"110"  =>  b"000",0,1;
    1,      x,          b"110"  =>  b"001",0,1;

    x,      x,          b"010"  =>  b"110",0,0;

    1,      x,          b"111"  =>  b"111",0,0;
    0,      x,          b"111"  =>  b"110",0,0;
    x,      0,          b"100"  =>  b"101",0,0;
    0,      1,          b"100"  =>  b"110",0,0;
    1,      1,          b"100"  =>  b"111",0,0;
    x,      0,          b"101"  =>  b"101",0,0;
    0,      1,          b"101"  =>  b"110",0,0;
    1,      1,          b"101"  =>  b"111",0,0;
    x,      x,          b"011"  =>  b"111",0,0;

END TABLE;

Idle = (State[]==b"000");
!Selected/ = (State[]==b"001");
Wait = (State[2..1]==b"10");
Hold = (State[]==b"111");
!Terminate/ = (State[]==b"110");
```

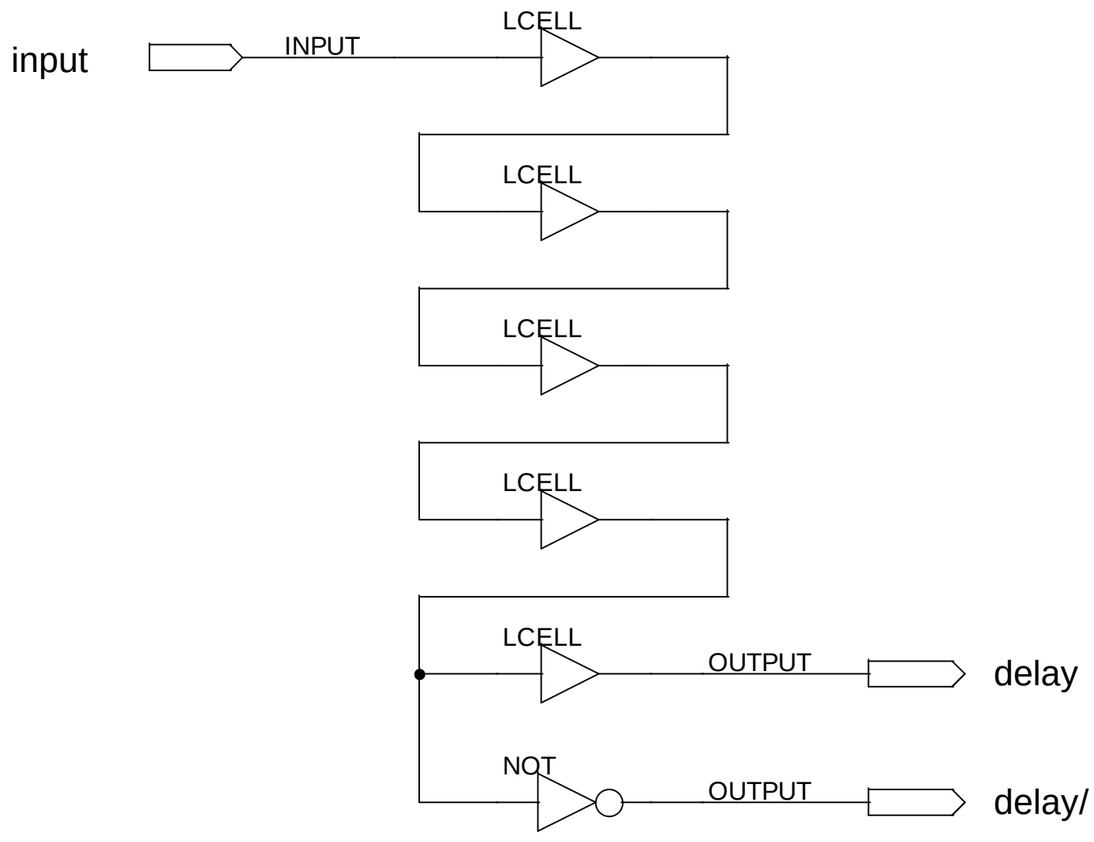
```
Ack = (State[1]==b"1");
END;
```

```
%TITLE " IDPROM FOR IP MODEL IPMDAT8 "%
% FILE NAME: ipmdat8.mif%
% 64 WORDS IN I/O SPACE, 128K WORDS IN MEMORY SPACE%
DEPTH = 64;
WIDTH = 8;
ADDRESS_RADIX = HEX;
DATA_RADIX = HEX;

CONTENT
 BEGIN
  0 : 00;
  1 : 49;       % I %
  2 : 00;
  3 : 50;       % P %
  4 : 00;
  5 : 41;       % A %
  6 : 00;
  7 : 43;       % C %
  8 : 00;
  9 : bb;   % Manufacturer ID 187 %
  A : 00;
  B : 18;   % Model Number 24 %
  C : 00;
  D : 01;       % Revision 1 %
  E : 00;
  F : 00;       % Reserved %
  10 : 00;
  11 : bb;      % Driver ID, low byte %
  12 : 00;
  13 : 18;      % Driver ID, high byte%
  14 : 00;
  15 : 20;      % Number of bytes used %
  16 : 00;
  17 : 00;      % CRC "0x00" %
  18 : 00;
  19 : 49;      % I %
  1A : 00;
  1B : 50;      % P %
  1C : 00;
  1D : 4D;      % M %
  1E : 00;
  1F : 44;      % D %
  20 : 00;
  21 : 41;      % A %
  22 : 00;
  23 : 54;      % T %
  24 : 00;
  25 : 38;      % 8 %
  26 : 00;
  27 : 20;      %   %
  28 : 00;
  29 : 52;      % R %
  30 : 00;
  31 : 45;      % E %
  32 : 00;
  33 : 56;      % V %
  34 : 00;
  35 : 30;      % 0 %
  36 : 00;
  37 : 00;      % NUL %
  38 : 00;
  39 : 00;      % NUL %
  3A : 00;
  3B : 4D;      % M %
  3C : 00;
  3D : 4A;      % J %
```

```
  3E : 00;
  3F : 4B;      % K %
END;
```



TITLE	Serdec2a (from Craig McLure)		
COMPANY	Fermilab		
DESIGNER	Mike Shea		
SIZE D	NUMBER 1.00	REV A	
DATE 5:18p 7-17-1998	SHEET 1	OF 1	


input
INPUT
LCELL
LCELL
LCELL
LCELL
LCELL
LCELL
OUTPUT
delay
NOT
OUTPUT
delay/