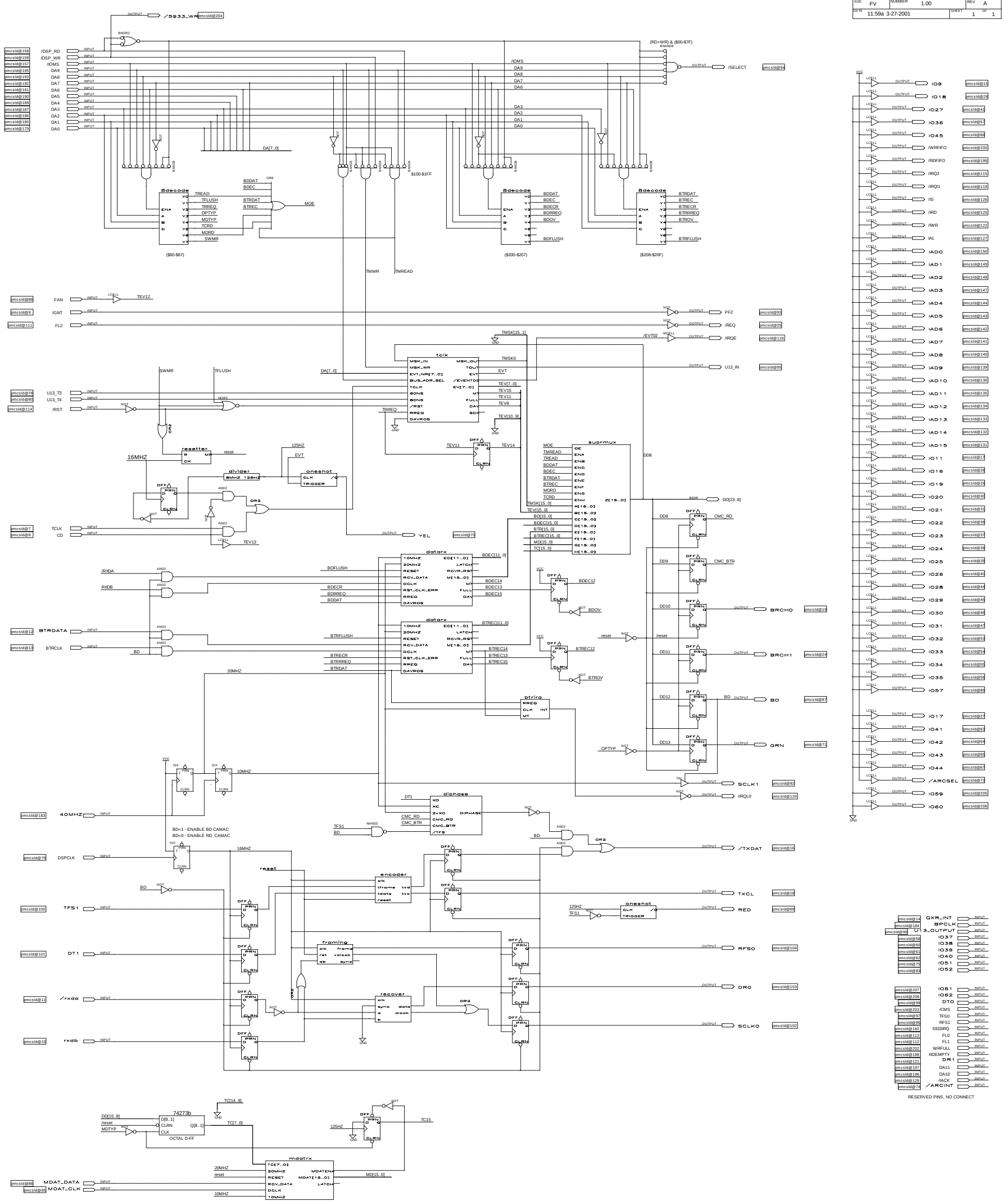
TITLE	UNIVERSAL CAMAC FOR REV C NNSLD
COMPANY	Fermilab
DESIGNER	Paul Kasley
SIZE FV	NUMBER 1.00 REV A
DATE 11:59a 3-27-2001	SHEET 1 OF 1

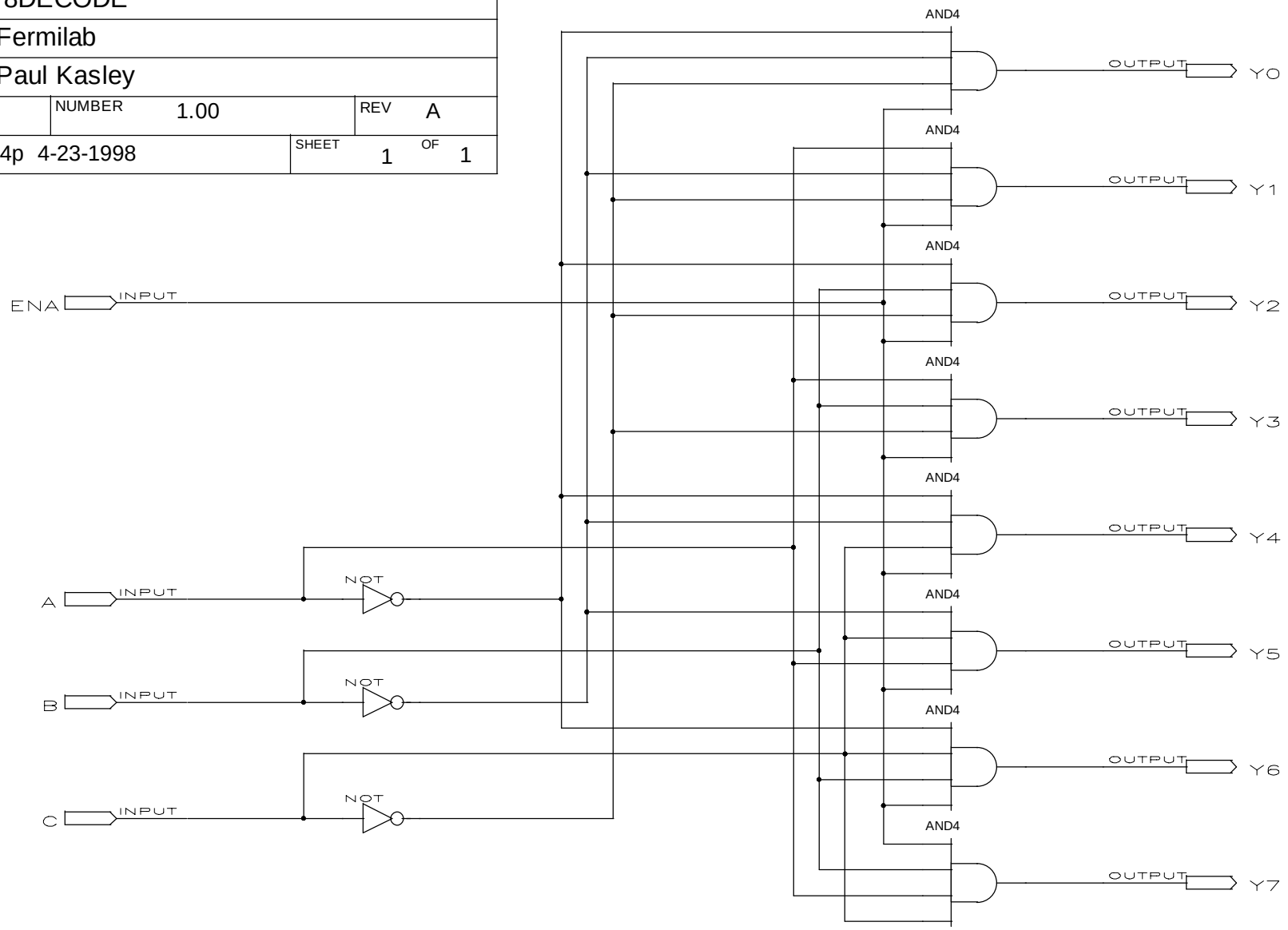


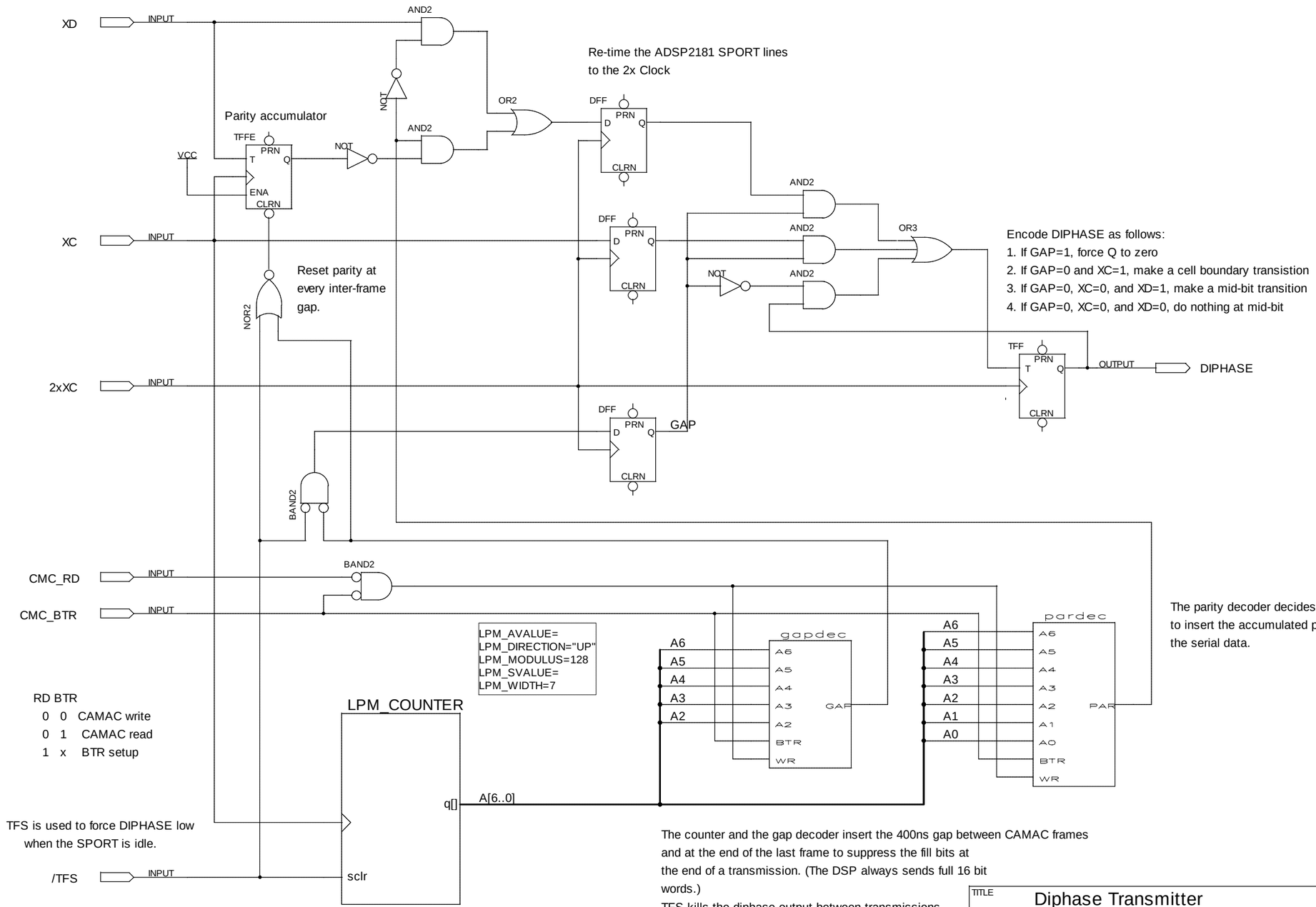
015478B1 8/2/99 Initial Release

0154D1D5 9/13/00 Repaired accumulated parity problem in diphase.gdf. PIOX parity is now reset at every interframe gap.

0154EB83 3/27/01 Recompiled with Quartus Fitter and Global Logic Synthesis Style="FAST" to eliminate PIOX and BTR errors at nominal operating temp. No other design changes.

TITLE	8DECODE		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE B	NUMBER 1.00	REV A	
DATE 2:24p 4-23-1998	SHEET 1	OF 1	

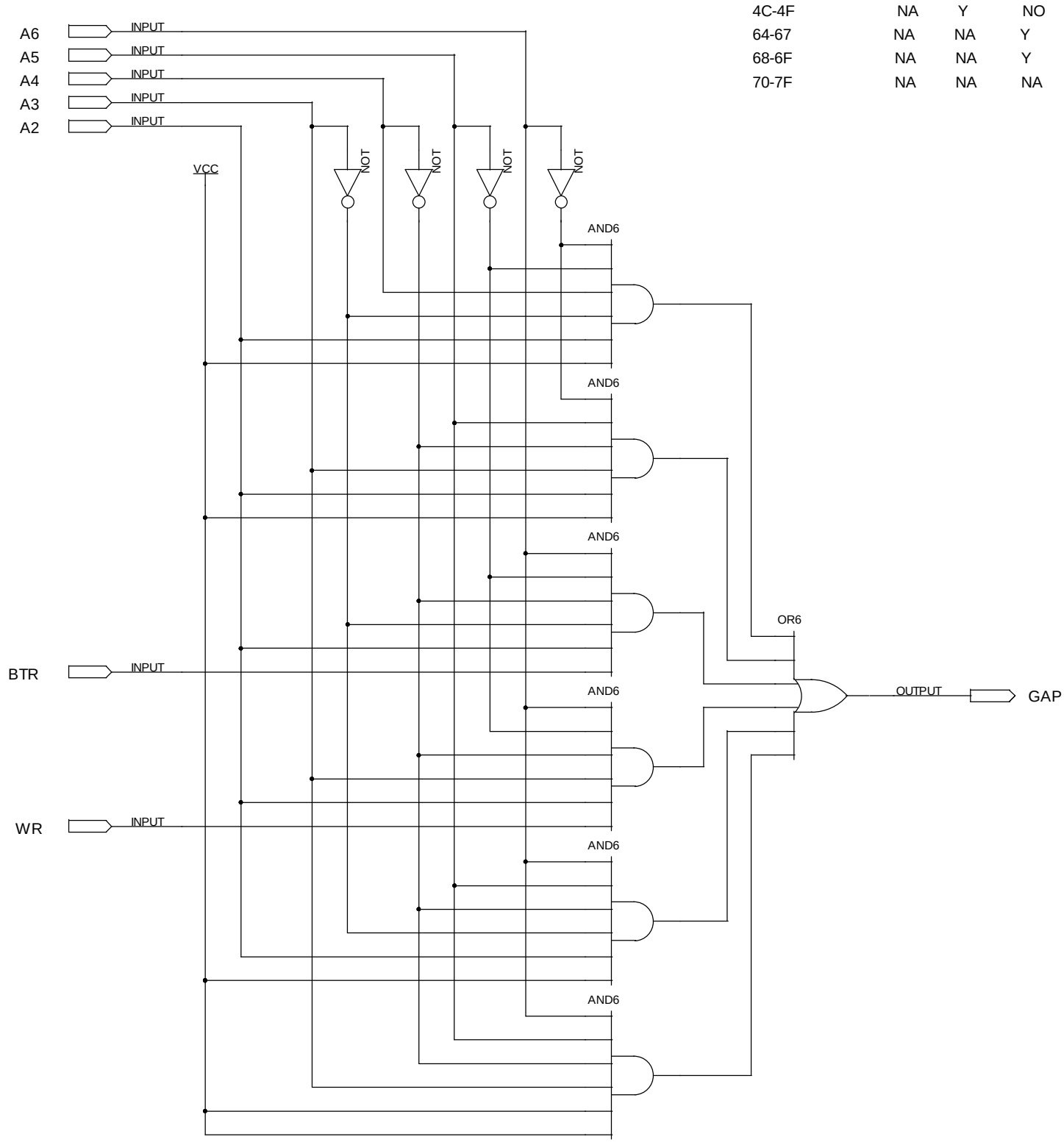




TITLE	Diphase Transmitter		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE C	NUMBER 1.00	REV A	
DATE 2:02p 9-13-2000		SHEET 1	OF 1

This is used to insert interframe gaps into PIOX transmissions. Each gap is four bits wide. The location of the gaps is determined by the type of transmission.

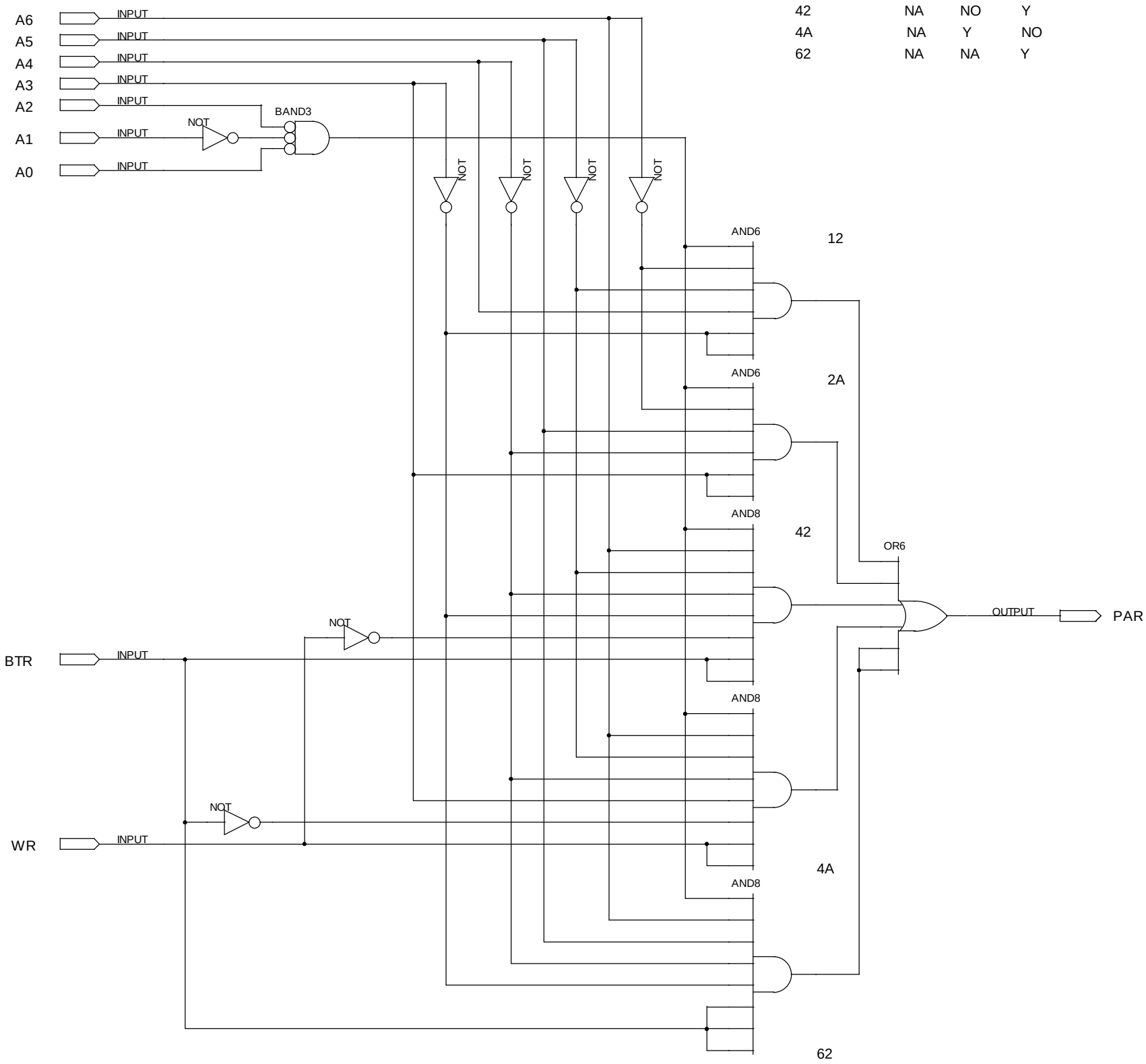
Bit	Read	Write	BTR	
14-17	Y	Y	Y	/6 /5 4 /3 2
2C-2F	Y	Y	Y	/6 5 /4 3 2
44-47	NA	NO	Y	6 /5 /4 /3 2 BTR
4C-4F	NA	Y	NO	6 /5 /4 3 2 WR
64-67	NA	NA	Y	6 5 /4 /3 2
68-6F	NA	NA	Y	6 5 /4 3
70-7F	NA	NA	NA	



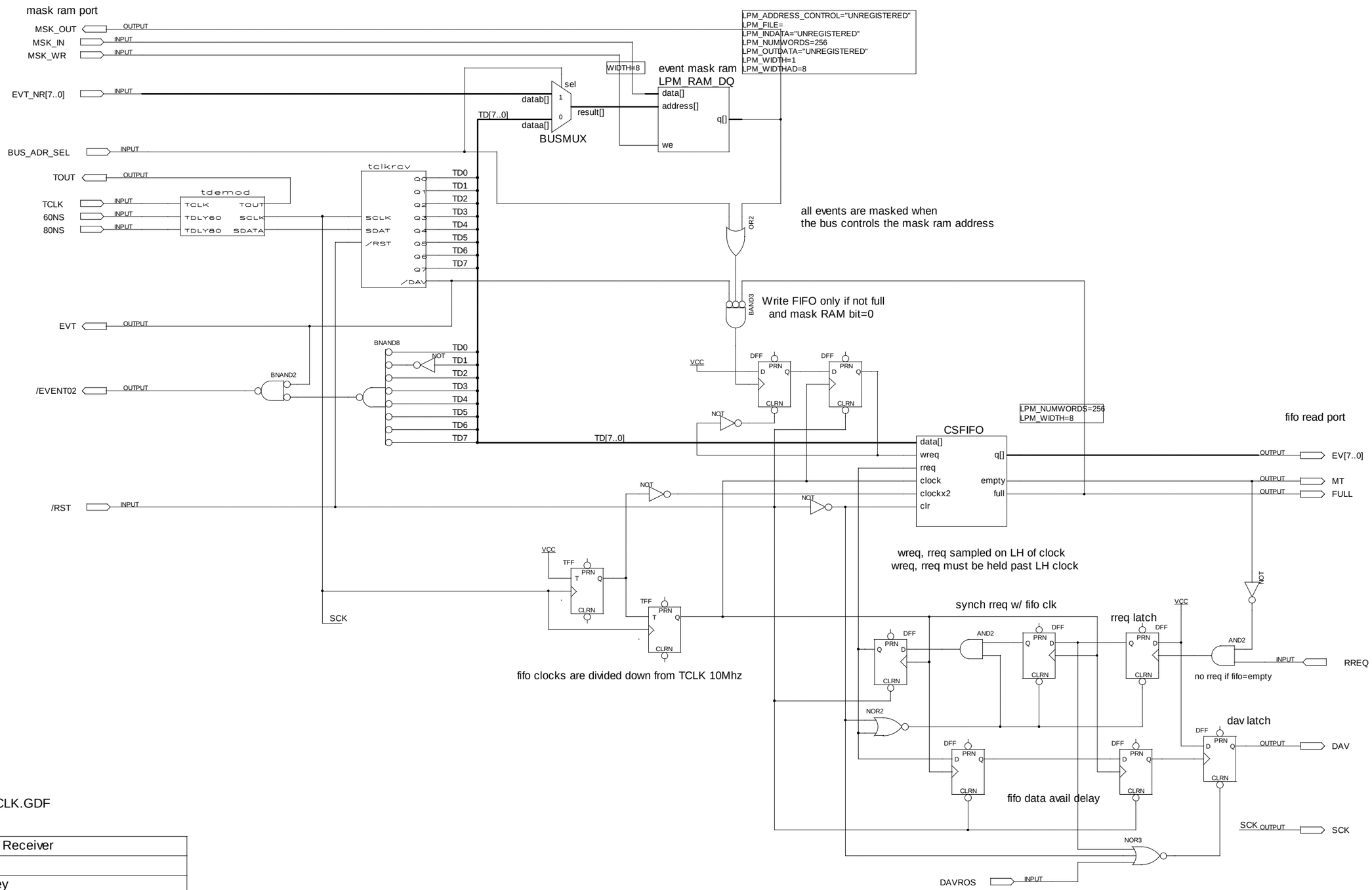
TITLE	GAP DECODER		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE CV	NUMBER 1.00	REV A	
DATE 2:00p 4-17-1998	SHEET 1	OF 1	

This is used to insert parity into PIOX transmissions.
The location of parity is determined by the type of transmission.

Bit	Read	Write	BTR
12	Y	Y	Y
2A	Y	Y	Y
42	NA	NO	Y
4A	NA	Y	NO
62	NA	NA	Y

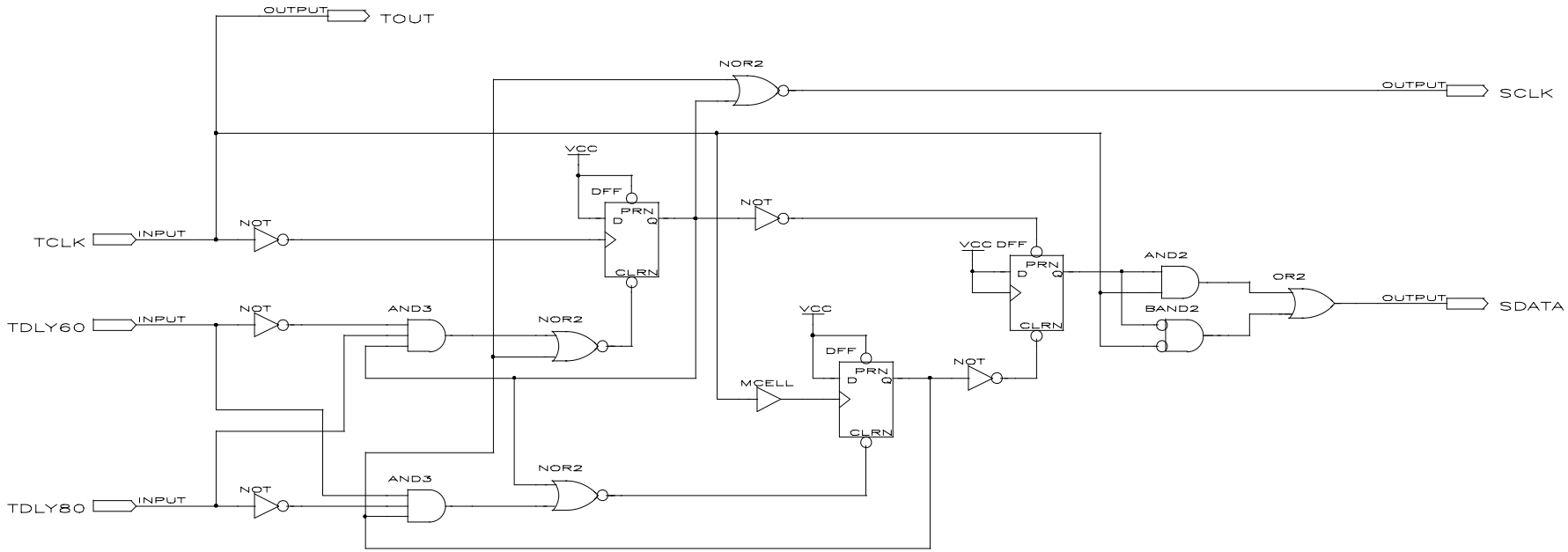


TITLE	PARITY DECODER		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE CV	NUMBER 1.00	REV A	
DATE 10:49a 5-26-1998		SHEET 1	OF 1



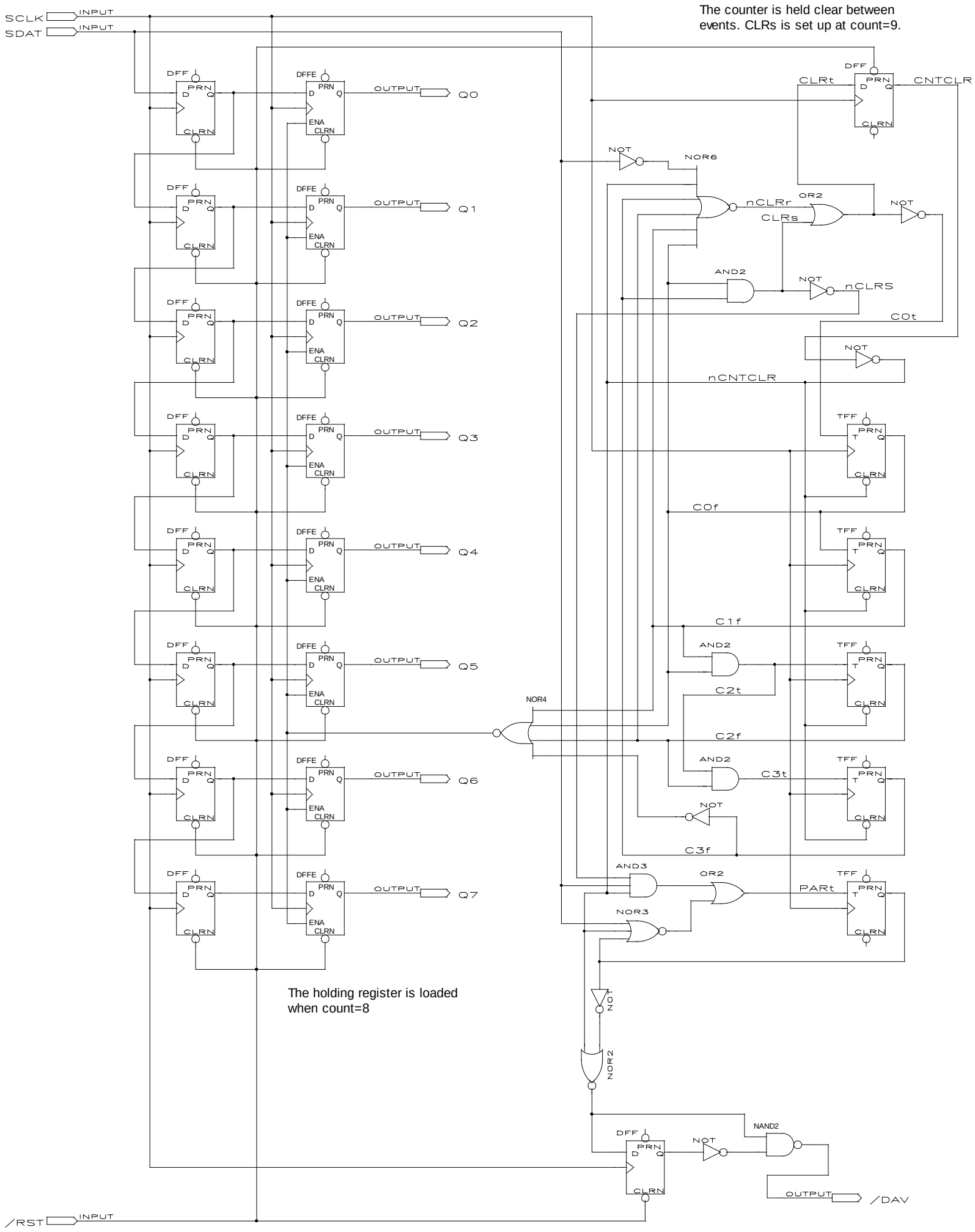
TCLK.GDF

TITLE	TeV Clock Receiver		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE D	NUMBER 1.00	REV A	
DATE 3:13p 4-21-1998		SHEET 1	OF 1



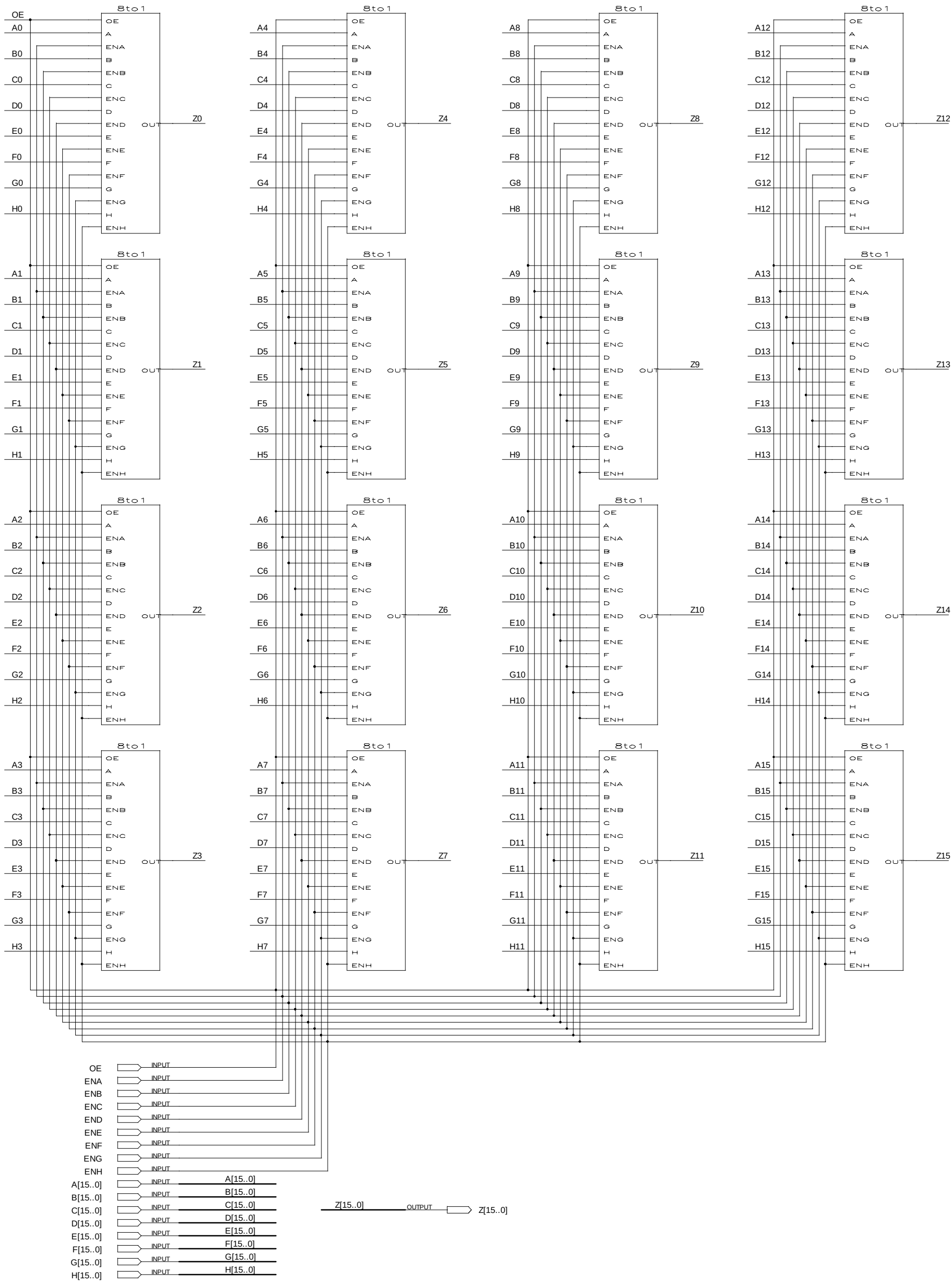
TDEMOD.GDF

TITLE TCLK DEMODULATOR		
COMPANY Fermilab		
DESIGNER PAK		
SIZE C	NUMBER 1.00	REV A
DATE 12:04p 8-13-1997	SHEET 1 OF 1	

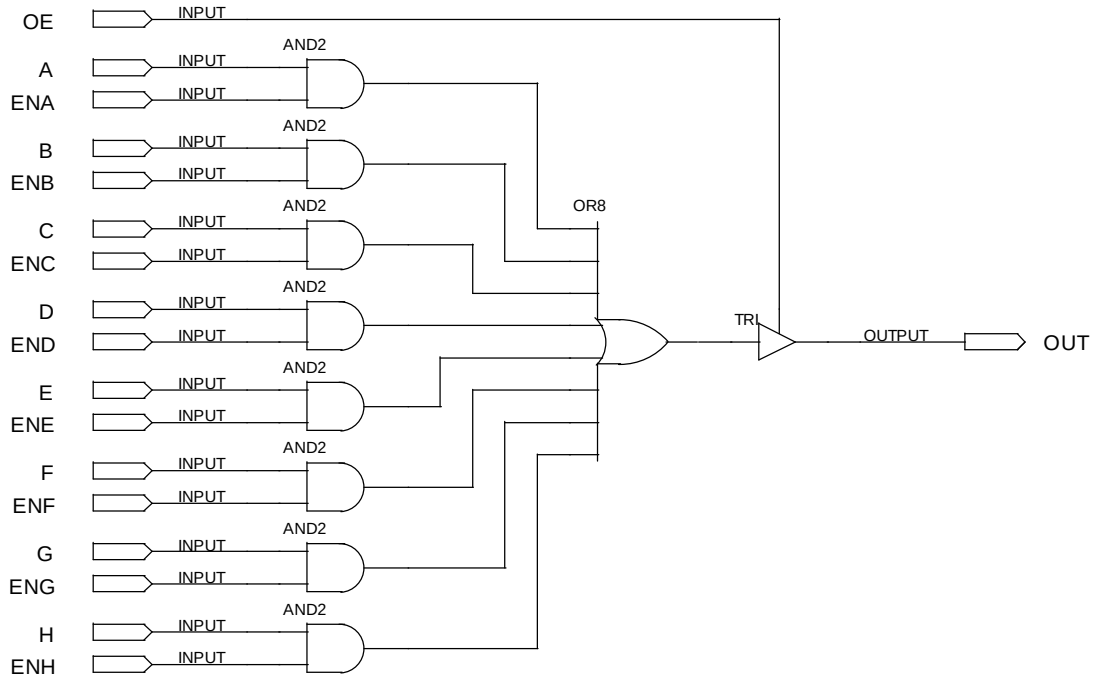


/DAV is 100ns low pulse that occurs after data is transferred to the holding register and if the received parity is good.

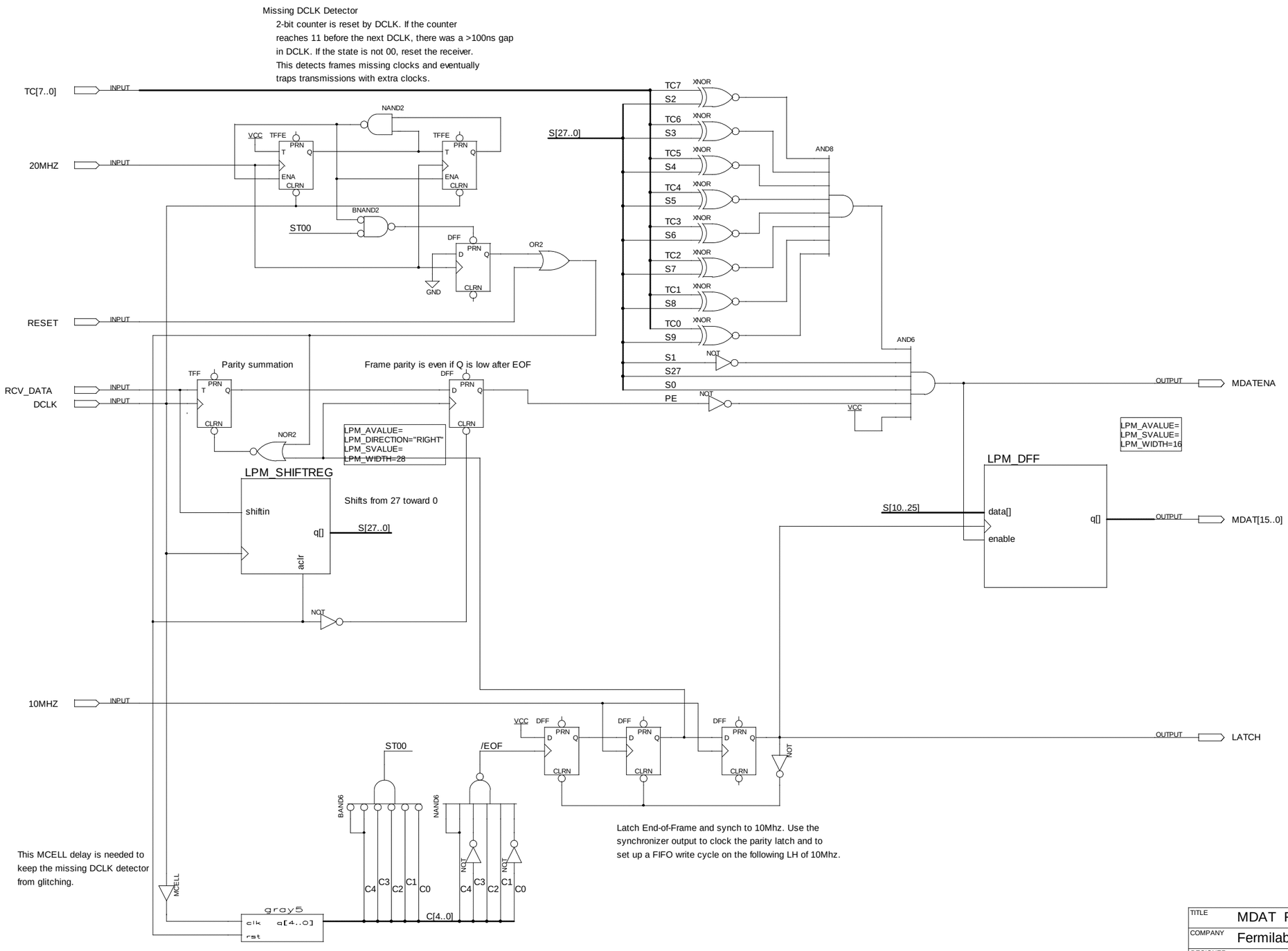
TITLE TCLK RCVR		
COMPANY FERMILAB		
DESIGNER PAK		
SIZE DV	NUMBER 1.00	REV A
DATE 11:59a 4-24-1997	SHEET 1 OF 1	



TITLE	SUPRMUX		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE DV	NUMBER 1.00	REV A	
DATE 1:40p 4-17-1998		SHEET 1	OF 1



TITLE	8TO1				
COMPANY	Fermilab				
DESIGNER	Paul Kasley				
SIZE B	NUMBER 1.00			REV A	
DATE 1:36p 4-17-1998			SHEET 1	OF 1	

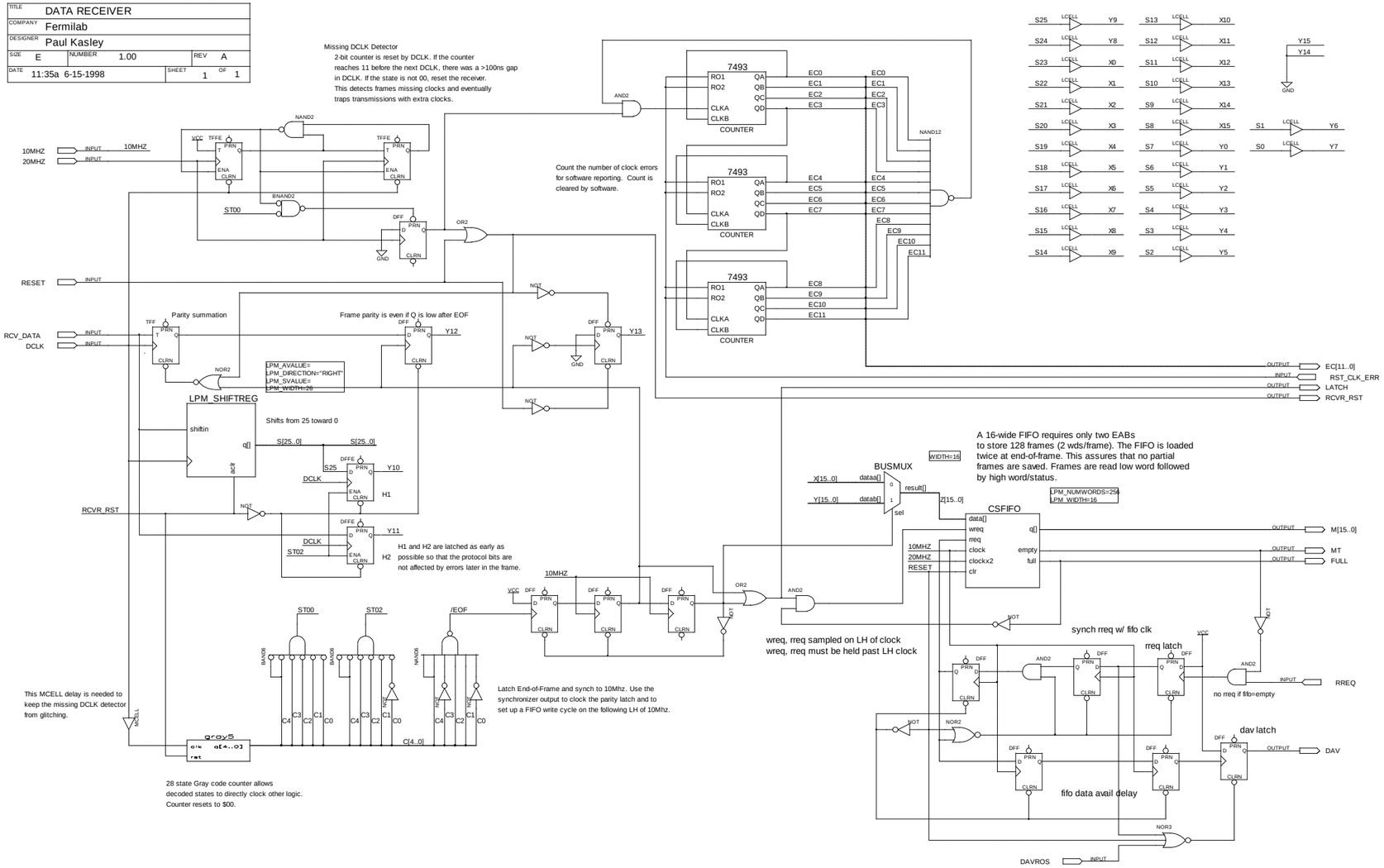


TITLE	MDAT RECEIVER		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE D	NUMBER 1.00		REV A
DATE 10:15a 2-02-1999		SHEET 1	OF 1

```
%*****************************************************************************
5 bit Gray code counter
28 Jan 98 pak
*****************************************************************************%

TITLE "Gray5";

SUBDESIGN Gray5
(clk        : INPUT;        % clock %
 rst        : INPUT;        % reset %
 q[4..0]    : OUTPUT;)

VARIABLE
gray5: MACHINE OF BITS (q[4..0])
    WITH STATES (
        x00=B"00000",
        x01=B"00001",
        x02=B"00010",
        x03=B"00011",
        x04=B"00100",
        x05=B"00101",
        x06=B"00110",
        x07=B"00111",
        x08=B"01000",
        x09=B"01001",
        x0a=B"01010",
        x0b=B"01011",
        x0c=B"01100",
        x0d=B"01101",
        x0e=B"01110",
        x0f=B"01111",
        x10=B"10000",
        x11=B"10001",
        x12=B"10010",
        x13=B"10011",
        x14=B"10100",
        x15=B"10101",
        x16=B"10110",
        x17=B"10111",
        x18=B"11000",
        x19=B"11001",
        x1a=B"11010",
        x1b=B"11011",
        x1c=B"11100",
        x1d=B"11101",
        x1e=B"11110",
        x1f=B"11111");

BEGIN

%set up the clock and reset%
gray5.clk=clk;
gray5.reset=rst;

% state transitions %
TABLE
    gray5   =>  gray5;

    x00 =>  x01;
    x01 =>  x03;
    x03 =>  x02;
    x02 =>  x06;
    x06 =>  x07;
```

```
    x07 =>  x05;
    x05 =>  x04;
    x04 =>  x0c;
    x0c =>  x0d;
    x0d =>  x0f;
    x0f =>  x0e;
    x0e =>  x0a;
    x0a =>  x0b;
    x0b =>  x09;
    x09 =>  x08;
    x08 =>  x18;
    x18 =>  x19;
    x19 =>  x1b;
    x1b =>  x1a;
    x1a =>  x1e;
    x1e =>  x1f;
    x1f =>  x1d;
    x1d =>  x1c;
    x1c =>  x14;
    x14 =>  x15;
    x15 =>  x17;
    x17 =>  x16;
    x16 =>  x00;
    x12 =>  x13;
    x13 =>  x11;
    x11 =>  x10;
    x10 =>  x00;

END TABLE;

END;
```

TITLE	DATA RECEIVER
COMPANY	Fermilab
DESIGNER	Paul Kasley
SIZE E / NUMBER 1.00 / REV A	
DATE 11:35a 6-15-1998 / SHEET 1 OF 1	



```
%*****************************************************************************
CAMAC SERIAL LINK CLOCK AND DATA RECOVERY

1 Aug 1997 pak
ISSUE X

Convert pulse streams on RXDA and RXDB lines to clock and synchronous data

"0" Cell from LMUX:
RXDA LLLL LLLL HHHH LLLL
RXDB HHHH LLLL LLLL LLLL

"1" Cell from LMUX:
RXDA HHHH LLLL LLLL LLLL
RXDB LLLL LLLL HHHH LLLL

When idle, RXDA and RXDB are both low. One full cell of idle will be interpreted
as a pause between words.
*****************************************************************************%

TITLE "CAMAC Serial Link Clock and Data Recovery";

SUBDESIGN recover
(clk        : INPUT;        %16 Mhz system clock%
 a          : INPUT;        %link data A%
 b          : INPUT;        %link data B%
 sync       : INPUT;        %reset pulse from framing logic%
 data       : OUTPUT;       %recovered data to ADSP%
 clock      : OUTPUT;       %recovered clock to ADSP%
)

VARIABLE
recov: MACHINE OF BITS (p[2..0])
    WITH STATES (idlezero,idleone,one,zero,zerowait,onewait);

BEGIN

%set up the clock and reset%
recov.clk=clk;
recov.reset=sync;

TABLE
recov, a, b => data, clock, recov ;

%RECOV      A   B   =>  DAT CLK RECOV %
idlezero,   1,  0   =>  1,  0,  one         ;
idlezero,   0,  1   =>  0,  0,  zero        ;
idlezero,   0,  0   =>  0,  0,  idlezero    ;
idlezero,   1,  1   =>  0,  0,  idlezero    ;

idleone,    1,  0   =>  1,  0,  one         ;
idleone,    0,  1   =>  0,  0,  zero        ;
idleone,    0,  0   =>  1,  0,  idleone     ;
idleone,    1,  1   =>  1,  0,  idleone     ;

zero,       1,  0   =>  0,  1,  zerowait    ;
zero,       1,  1   =>  0,  1,  zero        ;
zero,       0,  x   =>  0,  0,  zero        ;

zerowait,   0,  0   =>  0,  0,  idlezero    ;
zerowait,   0,  1   =>  0,  1,  zerowait    ;
zerowait,   1,  x   =>  0,  1,  zerowait    ;
```

```
one,        0,  1   =>  1,  1,  onewait     ;
one,        0,  0   =>  1,  0,  one         ;
one,        1,  x   =>  1,  0,  one         ;

onewait,    0,  0   =>  1,  0,  idleone     ;
onewait,    0,  1   =>  1,  1,  onewait     ;
onewait,    1,  x   =>  1,  1,  onewait     ;

END TABLE;
END;
```

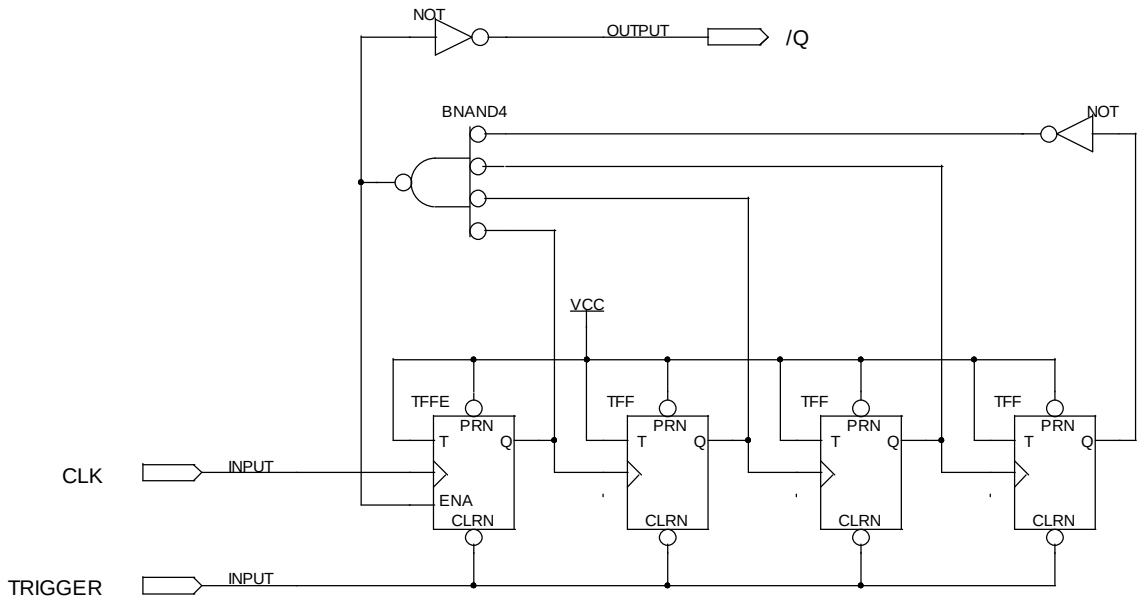
```
%*****************************************************************************
CAMAC SERIAL LINK RECEIVE FRAMING RECOVERY

1 Aug 1997 pak
ISSUE X

Monitor pulse streams on RXDA and RXDB lines to generate word framing for ADSP

"0" Cell from LMUX:
RXDA LLLL LLLL HHHH LLLL
RXDB HHHH LLLL LLLL LLLL

"1" Cell from LMUX:
RXDA HHHH LLLL LLLL LLLL
RXDB LLLL LLLL HHHH LLLL

When idle, RXDA and RXDB are both low. One full cell of idle will be interpreted
as a pause between words.
*****************************************************************************%

TITLE "CAMAC Serial Link Receive Framing Recovery";

SUBDESIGN framing
(clk        : INPUT;        %16 Mhz system clock%
 rst        : INPUT;        %system reset%
 ab         : INPUT;        %link data A or'ed with link data B%
 frame      : OUTPUT;       %synthesized receive framing to ADSP%
 xclock     : OUTPUT;       %extra clock pulse to burp the ADSP%
 sync       : OUTPUT;       %reset pulse to data recovery box%
)

VARIABLE
framing: MACHINE OF BITS (p[4..0])
    WITH STATES (idle,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,
        a14,a15,i0,i1,i2,i3,i4,i5,i6,i7,i8,i9,i10,i11,i12,i13,
        i14,i15);

BEGIN

%set up the clock%
framing.clk=clk;
framing.reset=rst;

% Set up current state decoding %
xclock = i4 # i5 # i6 #i7;
frame = a1 # a2 # a3 # a4 # a5 # a6 # a7 # a8 # a9 # a10
     # a11 # a12 # a13;

% SYNC currently has no real purpose. It could be used to
    reset the clock and data recovery machine. %
sync = a15;

% state transitions %
TABLE
    framing,ab => framing;

%   PS      AB  =>  NS %
    idle,   0   =>  i0;
    idle,   1   =>  a1;

% The A states form a retriggerable one-shot. The one-shot is
    retriggered whenever ab is high. Framing is high as long
    as the one-shot is triggered. If ab is low for 12
```

```
    consecutive clocks, the one-shot expires and
    framing is taken low. %
    a1  ,   0   =>  a2;
    a1  ,   1   =>  a1;
    a2  ,   0   =>  a3;
    a2  ,   1   =>  a1;
    a3  ,   0   =>  a4;
    a3  ,   1   =>  a1;
    a4  ,   0   =>  a5;
    a4  ,   1   =>  a1;
    a5  ,   0   =>  a6;
    a5  ,   1   =>  a1;
    a6  ,   0   =>  a7;
    a6  ,   1   =>  a1;
    a7  ,   0   =>  a8;
    a7  ,   1   =>  a1;
    a8  ,   0   =>  a9;
    a8  ,   1   =>  a1;
    a9  ,   0   =>  a10;
    a9  ,   1   =>  a1;
    a10 ,   0   =>  a11;
    a10 ,   1   =>  a1;
    a11 ,   0   =>  a12;
    a11 ,   1   =>  a1;
    a12 ,   0   =>  a13;
    a12 ,   1   =>  a1;
    a13 ,   0   =>  a14;
    a13 ,   1   =>  a1;
    a14 ,   0   =>  a15;
    a14 ,   1   =>  a1;
    a15 ,   x   =>  idle;

% The I states are used to generate fake clocks while AB
    is inactive. As long as AB is inactive, we cycle thru
    the 16 I states. Fake clock is active for i4-i7. As
    soon as AB goes active, we go to the idle state. This
    takes fake clock low one tick before framing is asserted. %
%   PS      AB  =>  NS %
    i0  ,   0   =>  i1;
    i0  ,   1   =>  idle;
    i1  ,   0   =>  i2;
    i1  ,   1   =>  idle;
    i2  ,   0   =>  i3;
    i2  ,   1   =>  idle;
    i3  ,   0   =>  i4;
    i3  ,   1   =>  idle;

% Fake clock is active for these four states %
    i4  ,   0   =>  i5;
    i4  ,   1   =>  idle;
    i5  ,   0   =>  i6;
    i5  ,   1   =>  idle;
    i6  ,   0   =>  i7;
    i6  ,   1   =>  idle;
    i7  ,   0   =>  i8;
    i7  ,   1   =>  idle;

    i8  ,   0   =>  i9;
    i8  ,   1   =>  idle;
    i9  ,   0   =>  i10;
    i9  ,   1   =>  idle;
    i10 ,   0   =>  i11;
    i10 ,   1   =>  idle;
    i11 ,   0   =>  i12;
```

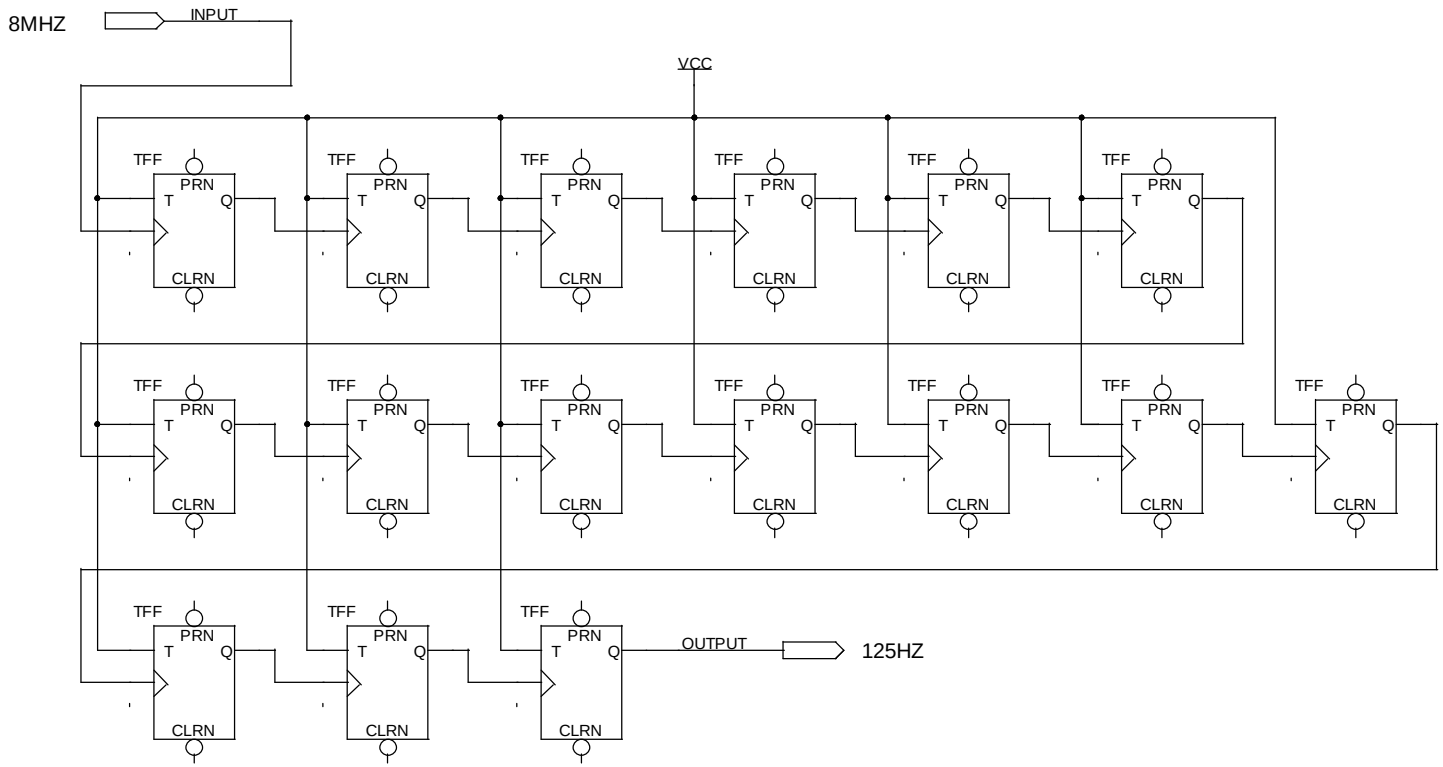
```
    i11 ,   1   =>  idle;
    i12 ,   0   =>  i13;
    i12 ,   1   =>  idle;
    i13 ,   0   =>  i14;
    i13 ,   1   =>  idle;
    i14 ,   0   =>  i15;
    i14 ,   1   =>  idle;
    i15 ,   0   =>  i0;
    i15 ,   1   =>  idle;


END TABLE;

END;
```

```
%*****************************************************************************
CAMAC SERIAL LINK CLOCK AND DATA ENCODER

1 Aug 1997 pak
ISSUE 0

Convert clock and synchronous data from the ADSP2111 SPORT to pulse streams
on TXD and /TXC lines.

The ADSP has clock, serial data, and framing outputs. Framing and data change on
the LH transition of clock. Framing is high for the entire 70 bit word. Clock is
continuous 1Mhz. Data is tri-stated between words.

This encoder is clocked at 16Mhz to eliminate phasing of a high frequency
clock with the data clock from the ADSP2111. The count is a Gray-code sequence
to eliminate glitches in the txd and txc decoding.

Both txd and txc are at DC vcc when framing is inactive.

0 Cell:
txd (+TXDT) LLLL LLLL HHHH HHHH
txc (-TXCL) LLLL HHHH LLLL HHHH

1 Cell:
txd (+TXDT) HHHH HHHH LLLL LLLL
txc (-TXCL) LLLL HHHH LLLL HHHH
*****************************************************************************%

TITLE "CAMAC Serial Link Clock and Data Encoder";

SUBDESIGN encoder
(clk        : INPUT;        %16Mhz clock%
 tframe     : INPUT;        %ADSP framing%
 tdata      : INPUT;        %ADSP serial data%
 reset      : INPUT;        %system reset input%
 txd        : OUTPUT;       %transmit data; same as +TXDT to LMUX%
 txc        : OUTPUT;       %transmit clock; same as -TXCL to LMUX%
)

VARIABLE
encoder: MACHINE OF BITS (z[5..0])
    WITH STATES (i, a0, b0, c0, d0, e0, f0, g0, h0, i0, j0, k0, l0,
     m0, n0, o0, p0, a1, b1, c1, d1, e1, f1, g1, h1, i1, j1, k1, l1,
     m1, n1, o1, p1);

BEGIN

%set up the clock and reset%
encoder.clk=clk;
encoder.reset=reset;

%output decoding%
txd = i # i0 # j0 # k0 # l0 # m0 # n0 # o0 # p0 # a1 # b1 # c1 # d1 #
     e1 # f1 # g1 # h1;
txc = i # e0 # f0 # g0 # h0 # m0 # n0 # o0 # p0 # e1 # f1 # g1 # h1 #
     m1 # n1 # o1 # p1;

TABLE
encoder, tframe, tdata => encoder;

%ENC    TFRM    TDAT    =>  ENC%
i,      1,      1       =>  a1;
i,      1,      0       =>  a0;
```

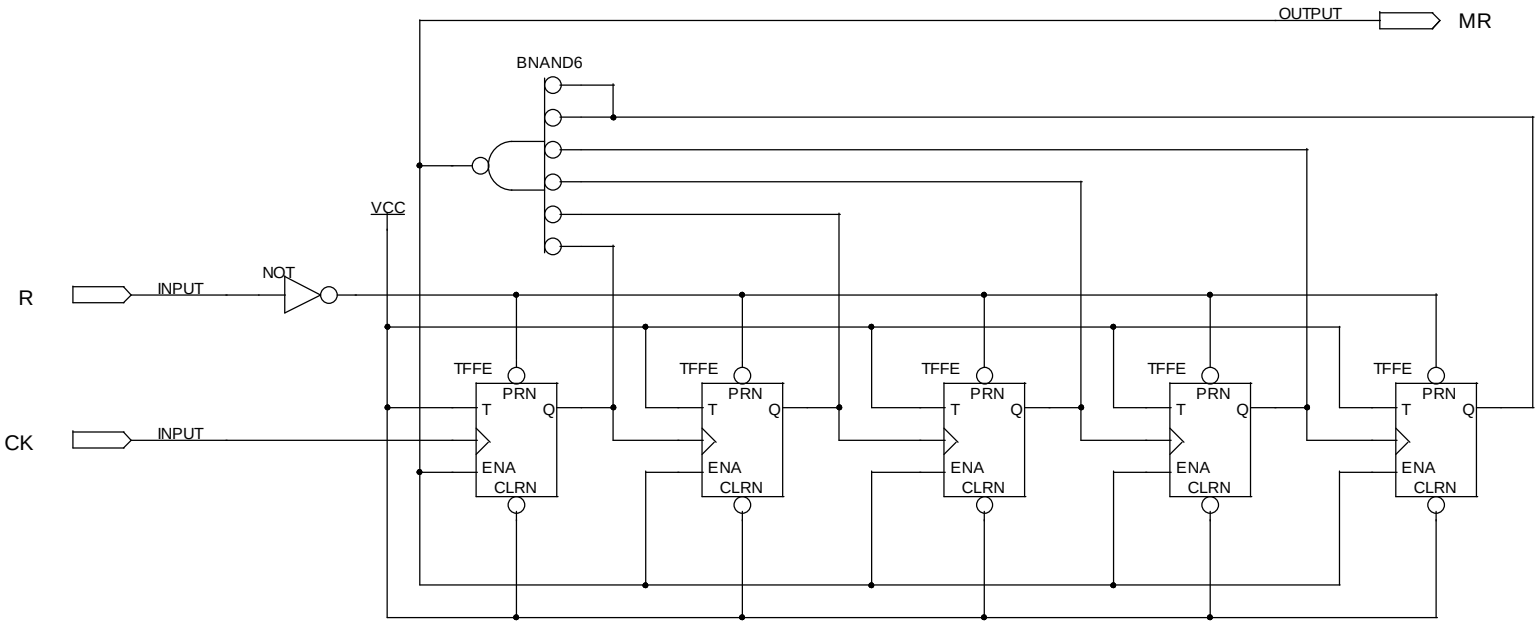
```
i,      0,      x       =>  i;

a0,     x,      x       =>  b0;
b0,     x,      x       =>  c0;
c0,     x,      x       =>  d0;
d0,     x,      x       =>  e0;
e0,     x,      x       =>  f0;
f0,     x,      x       =>  g0;
g0,     x,      x       =>  h0;
h0,     x,      x       =>  i0;
i0,     x,      x       =>  j0;
j0,     x,      x       =>  k0;
k0,     x,      x       =>  l0;
l0,     x,      x       =>  m0;
m0,     x,      x       =>  n0;
n0,     x,      x       =>  o0;
o0,     x,      x       =>  p0;

p0,     1,      1       =>  a1;
p0,     1,      0       =>  a0;
p0,     0,      x       =>  i;

a1,     x,      x       =>  b1;
b1,     x,      x       =>  c1;
c1,     x,      x       =>  d1;
d1,     x,      x       =>  e1;
e1,     x,      x       =>  f1;
f1,     x,      x       =>  g1;
g1,     x,      x       =>  h1;
h1,     x,      x       =>  i1;
i1,     x,      x       =>  j1;
j1,     x,      x       =>  k1;
k1,     x,      x       =>  l1;
l1,     x,      x       =>  m1;
m1,     x,      x       =>  n1;
n1,     x,      x       =>  o1;
o1,     x,      x       =>  p1;

p1,     1,      1       =>  a1;
p1,     1,      0       =>  a0;
p1,     0,      x       =>  i;


END TABLE;
END;
```



TITLE	Oneshot		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE B	NUMBER 1.00	REV A	
DATE 12:14p 9-15-1998	SHEET 1	OF 1	



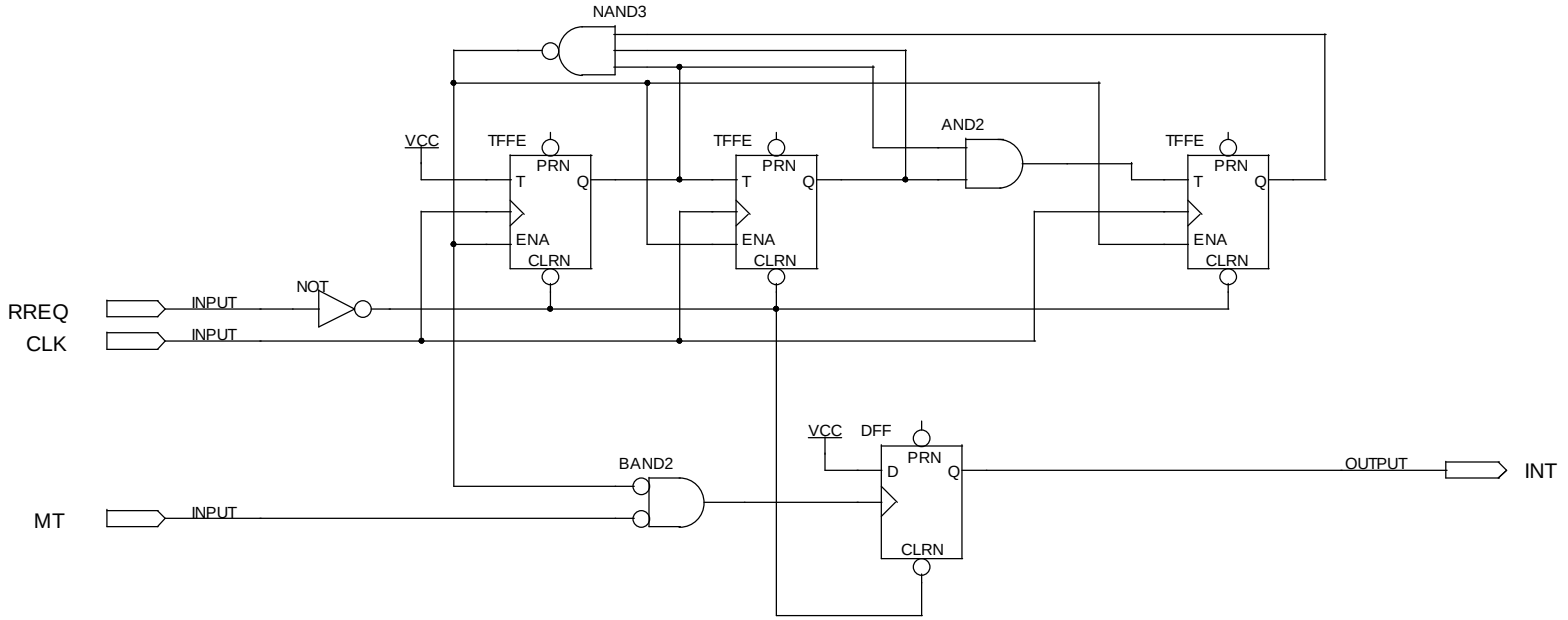
TITLE	DIVIDER		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE B	NUMBER 1.00	REV A	
DATE 10:13a 4-24-1998	SHEET 1	OF 1	



TITLE	RESETTER		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE B	NUMBER 1.00	REV A	
DATE 1:39p 4-17-1998	SHEET 1	OF 1	

RESETTER.GDF

BTRIRQ converts the MT signal from the BTR receiver into a level sensitive DSP interrupt. The counter inserts a delay between successive interrupts to ensure that lower priority DSP interrupts are serviced.



TITLE	BTRIRQ		
COMPANY	Fermilab		
DESIGNER	Paul Kasley		
SIZE B	NUMBER 1.00	REV A	
DATE 2:58p 6-15-1998	SHEET 1	OF 1	