```
module _c1151u3  flag '-d83'

@message ' NOTICE: This file generates Intel Hex file called C1151U3.P83 '

title   'CAMAC function code decoder /Vector Generator
        Mike Kuplic
        17-Oct-1990
        Checksum: D1AF'

        c1151u3 device 'RA9P8';

" Modification history:
" 17-Oct-1990 MJK added F6A5 for Read_Module_Serial_Number

" Inputs
        " Function lines
                F1,F2,F4,F8,F16                 pin     5,16,17,18,19;
        " Address lines
                A1,A2,A4,A8                     pin     1,2,3,4;
        " Module Select
                N                               pin     15;
" Outputs
        " CAMAC function codes to 5032 (U?)
                CF0,CF1,CF2,CF3,CF4,CF5,CF6     pin     6,7,8,9,11,12,13;

        " CAMAC 'X' Response
                !X                              pin     14;

" Set Definitions (Define function and subaddress inputs as active low)
"       Define outputs as active low so default state is FF Hex

        func = [!F16,!F8,!F4,!F2,!F1];
        addr = [!A8,!A4,!A2,!A1];
        fcod = [.X.,!CF6,!CF5,!CF4,!CF3,!CF2,!CF1,!CF0];

" Macro Definitions
        vector macro (f,a,v) {[?f,?a] -> [1,?v]};

truth_table ( [func,addr] -> [X,fcod] )

" F8A0 and F9A0 are taken care of in hardware and do not generate
" an interrupt vector to the processor
" Vector 7F is not allowed, as this is the default state without interrupts

vector ( 8, 0,^h7E);
vector ( 9, 0,^h7D);

" Place all new interrupt vectors here, Using the following protocol:
"       Interrupts 80-BF: Read Functions
"       Interrupts C0-EF: Write Functions
"       Interrupts F0-FD: Control Functions
" Syntax of vector macro is: vector (function,subaddress,vector#);

" Define interrupt vectors for reading and writing 16 memory locations
" for the level and slope information
" F0A0 - F0A15 use interrupt vectors 00-0F
" F16A0 - F16A15 use vectors 40-4F

        @const temp=0;
        @repeat 16
                {
                vector ( 0,@expr temp;,@expr temp; + ^h80);
                vector (16,@expr temp;,@expr temp; + ^hC0);
                @const temp = temp + 1;
                }
```

```
" Read Status and A/D
        vector (1,0,^h90);
        vector (1,2,^h92);

" Read Current T-Time
        vector (1,3,^h98);

" Read module ID, software revision, & serial number
        vector (6,0,^h94);
        vector (6,1,^h95);
        vector (6,5,^h99);

" Enable/Disable LAM
        vector (26,15,^hF4);
        vector (24,15,^hF2);

" Write/Read LAM Register
        vector (19,14,^hD2);
        vector ( 6,14,^h93);

" Write/Read LAM Mask
        vector (19,13,^hD5);
        vector ( 6,13,^h9A);

" Clear LAM
        vector (10,0,^hF8);             " Clear all LAM bits
        vector (19,12,^hD6);            " Clear selected bits

" Power supply RESET, OFF, ON and POLarity relay functions
        vector (12,0,^hF0);
        vector (28,0,^hF5);
        vector (30,0,^hF6);
        vector (17,0,^hD0);

" RAMP Enable and Disable
        vector (24,0,^hF1);
        vector (26,0,^hF3);

" Write/Read Switches
        vector (17,1,^hD1);
        vector (1,1,^h91);

" FOP Code Stuff
        vector(6,3,^h96);
        vector(6,4,^h97);
        vector(19,2,^hD3);
        vector(19,3,^hD4);

" UNUSED VECTORS (Hex):
"       READ:    9B-BF
"       WRITE:   D7-EF
"       CONTROL: F9-FD

end _c1151u3
```

```
module _C1151U16
title 'VIKING I/O controller for C1151 module
mjk
8-1-89
Checksum: D436'


 C1151U16 device 'P22V10';

 X,VCC,GND                      = .X., 1, 0;

"Processor control lines
        IOSEL,IORD,IOWR         pin 2,3,4;

"Processor address and data lines
        D0,A3,A2,A1             pin 6,7,8,9;    Address = [A3,A2,A1,X];

"Control outputs for data buffer
        SBA,G,XDIR,SAB,CAB      pin 15,16,17,18,19;

"Control outputs for relays
        ON,OFF,RESET,POL        pin 23,22,21,20;

"Interrupt output just in case we ever need it
        IOINT                   pin 14;
" Write strobe into data buffer
        CBA                     pin 13;

" Use standard PALASM logical operators
@ALTERNATE

equations

/ON     = /IOSEL * /IOWR * (Address == 0) * D0
        + /ON *    /(/IOSEL * /IOWR * (Address == 0) * /D0);

/OFF    = /IOSEL * /IOWR * (Address == 2) * /D0
        + /OFF *   /(/IOSEL * /IOWR * (Address == 2) *  D0);

/RESET  = /IOSEL * /IOWR * (Address == 4) * D0
        + /RESET * /(/IOSEL * /IOWR * (Address == 4) * /D0);

/POL    = /IOSEL * /IOWR * (Address == 6) * D0
        + /POL *   /(/IOSEL * /IOWR * (Address == 6) * /D0);

/G      = /IOSEL * /IORD * (Address == 8);

XDIR    = GND;
SBA     = GND;
SAB     = GND;
CAB     = GND;

/IOINT  = /CBA;

end _C1151U16
```

```
module _C1151U22
title 'Tevetron Clock decode controller
mjk
8-1-89
Checksum: 4AB2'

 C1151U22 device 'P16V8R';

 X,C,L,H                = .X., .C., 0, 1;

" INPUTS:
        " Output clock
                SCLK pin 1;

        " I/O read and write strobes
                IORD,IOWR pin 7,6;

        " Select lines for FIFO and mask RAM from U10
                FIFORD,CLKCS pin 3,8;

        " Status lines related to TCLK decode
                CLKDAV,MSKOUT,PERR pin 5,2,4;

        " Data bit 0 (mask value)
                D0 pin 13;

        " FIFO Output Ready
                FIFOOR pin 9;

" OUTPUTS:
        " FIFO output enable, shift out, shift in, and shift in flip-flop
                CLKEN,RDCLK,FIFOSI,SIFF pin 12,14,15,18;

        " Mask RAM Write strobe, data buffer enable, and data bit (d0)
                MSKWR,MSKDBE,MSKIN pin 19,17,16;

" Use standard PALASM logical operators
@ALTERNATE

equations

ENABLE D0 = /CLKCS * /IORD;             "Enable D0 when reading Mask values"
/D0      = /MSKOUT;                     "No inversion, just buffering"
/MSKIN   = /D0;                         "Same here"

/CLKEN   = /FIFORD * /IORD * FIFOOR     "FIFO OE (Data of FF means no event)"
         + /CLKEN * /FIFORD * /IORD;

RDCLK    = /FIFORD * /IORD;             "FIFO SO"

FIFOSI   = CLKDAV * MSKOUT * SIFF;      "Write to FIFO on unmasked event"
/SIFF   := CLKDAV;                      " Use SCLK for SI pulse"

/MSKWR   = /CLKCS * /IOWR;              "Write to mask RAM"

" MSKDBE selects whether CPU Address lines or the decoded clock event"
" addresses the mask RAM"
" If the processor wants to write to the mask RAM when there is a valid event"
" decoded, we have a problem. Tune in tomorrow for the dramatic conclusion!!!"

/MSKDBE = /CLKCS;

fuses "User signature word
        [2056..2071] = 'C1';
        [2072..2087] = '15';
```

```
        [2088..2103] = '1U';
        [2104..2119] = '22';

end _C1151U22
```

```
module _C1151U36
title 'CAMAC Ramp Controller
wrk
8-1-89
Checksum: 7AFB'

C1151U36 device 'P22V10';

 X      = .X.;

" Inputs from CPU:
" Address AND Data Strobes
        AS,DS                   pin 1,2;
" processor Status Lines
        ST3,ST2,ST1,ST0         pin 9,8,7,6;    STAT = [ST3, ST2, ST1, ST0];
" data size control lines
        BW                      pin 5;
" bus cycle control
        RW,MEMREQ               pin 3,4;
" address lines
        A0                      pin 10;
" Z8036 Chip select (active HIGH)
" Disable data buffers because this talks directly to A/D lines
        TIMERCS                 pin 11;

" Outputs to buffers and memory
" byte write strobes
" WR0 is the MSB.
        WRL, WRH                pin 22,21;

" Data Bus Buffer Enable And Direction
        DBEN,DIR                pin 19,20;
" Vectored Interrupt Acknowledge
        VIACK                   pin 23;
" Memory Area select lines
        ROMREQ,RAMREQ,IOREQ     pin 17,16,18;

" valid memory cycles
" STAT = 1000B ; data memory request
" STAT = 1001B ; stack memory request
" STAT = 1100B ; program reference, nth word
" STAT = 1101B ; Instruction fetch, first word
" STAT = 0010B ; I/O reference
" STAT = 0111B ; Vectored interrupt acknowledge

        ROM_OP  = STAT == [1,1,0,X];
        RAM_OP  = STAT == [1,0,0,X];
        IO_OP   = STAT == [0,0,1,0];
        VIA_OP  = STAT == [0,1,1,1];

" Use standard PALASM logical operators
@ALTERNATE

equations

" Enable data buffers on any valid bus cycle
" Enable bus early on write cycle, and wait for Data Strobe on read
" Disable bus whenever CIO is selected or AS is low

/DBEN   = /MEMREQ * RAM_OP *  RW * /DS * AS
        + /MEMREQ * RAM_OP * /RW * AS
        + /MEMREQ * ROM_OP *  RW * /DS * AS
        + /MEMREQ * ROM_OP * /RW * AS
        + IO_OP * /TIMERCS *  RW * /DS * AS
        + IO_OP * /TIMERCS * /RW * AS;
```

```
" WRL & WRH control which byte(s) in memory are written to

/WRH    = /RW * /DS * /MEMREQ * RAM_OP *  BW * /A0
        + /RW * /DS * /MEMREQ * RAM_OP * /BW;

/WRL    = /RW * /DS * /MEMREQ * RAM_OP *  BW *  A0
        + /RW * /DS * /MEMREQ * RAM_OP * /BW;

/VIACK  = VIA_OP;

" Keep buffers pointing out except when reading through them

/DIR    = RW * /TIMERCS;

/ROMREQ = /MEMREQ * ROM_OP * RW;

/RAMREQ = /MEMREQ * RAM_OP;

/IOREQ  = IO_OP;

end _C1151U36
```

```
module _C1151U40
title '

CAMAC 1151
CAMAC Interrupt Controller
WAIT state generator
CAMAC RESET latch

mjk
8-1-89

Checksum: 3A7A
'

 C1151U40 device 'P16V8R';

" INPUTS:
        " Clock (10 MHz system clock)
              MHZ10                   pin 1;

        " Interrupt input
              CAMINT                  pin 3;

        " Interrupt acknowledge from processor
              VIACK                   pin 4;

        " IEO from 8036
              IEO                     pin 2;

        " Reset command from CAMAC and System reset (Command acknowledge)
              RESET,CPURES            pin 5,6;

        " WAIT input from address decoder (Inserts one wait state)
              WAIT                    pin 7;

        " CAMAC Module Select
              N                       pin 8;

        " Data Strobe (to make sure processor sees wait signal)
              DS                      pin 9;

        " CAMAC interrupt disable
              IDIS                    pin 12;

        " Pin 11 of 16V8 must be grounded when it is used as 16V8R
              gnd                     pin 11;

" OUTPUTS:
        " A/D bit 7
              AD7                     pin 19;

        " Interrupt Acknowledge output to 5032
              CAMIACK                 pin 18;

        " Interrupt line to processor
              VINT                    pin 17;

        " WAIT line to processor
              CPUWAIT                 pin 16;

        " WAIT timing flip flops
              WFF1,WFF2               pin 15,14;

        " RESET trigger to DS1232
              RESTRG                  pin 13;
```

```
" Constants:
        GND = 0;
        VCC = 1;

" Use standard PALASM logical operators
@ALTERNATE

equations

" CAMAC Interrupt Control

ENABLE AD7 = /CAMIACK;  " Pull AD7 High during Interrupt Acknowledge cycle "
AD7      = VCC;

" The CAMAC chip gets the VIACK if the 8036 doesn't want it "

/CAMIACK = /VIACK * IEO * /DS;

" Tri-State the CAMAC interrupt onto the VINT line where it is wire "
" ORed with the /INT line from the 8036 "

ENABLE VINT = /CAMINT * /IDIS;
VINT    = GND;

" Processor WAIT timing:

/CPUWAIT = /WAIT * WFF2         " Wait 1 State for slow IO  SET "
         + /N;                          " Wait during CAMAC cycle   "

/WFF1   := /WAIT * /DS;                 " Be sure CPUWAIT is low on /DS "
/WFF2   := /WAIT * /WFF1;

" Stretch F9A0 and BZ from 5032 long enough for DS1232 to see them "

/RESTRG  = /RESET                                  " Set  "
         + /RESTRG * CPURES;                       " Hold "

end _C1151U40
```

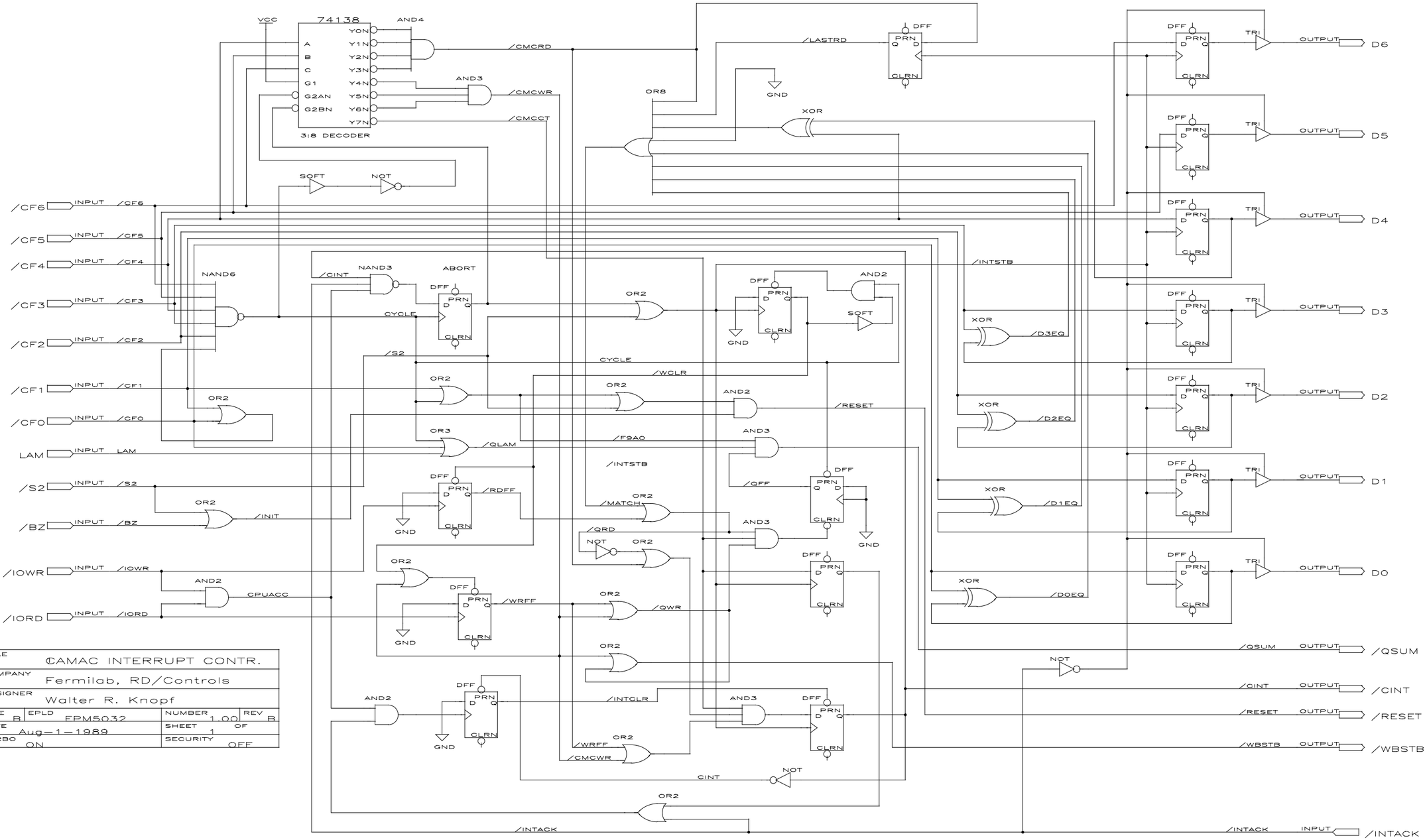
```
mjk
Fermilab
08-01-89
C1151 Address Decoder
U45 - EP900J

OPTIONS: TURBO=ON
PART:    EP900J

INPUTS:  A7@5,A8@4,A9@3,A10@43,A11@42,A12@41,A13@27,A14@28,A15@29
         10MHZ@2,IOREQ@19,RW@20,DS@21,RAMREQ@25,ROMREQ@26

OUTPUTS: ADCS@9,CMCWR@10,CMCRD@11,WAIT@12,ZCLK@13,TIMERCS@14,RAMRD@15
         RAMEN@16,ROMRD@18,IOSEL@30,NVRCS@31,WRDAC@32,DUART@33,FIFORD@34
         CLKCS@35,IORD@36,IOWR@37,RDLED@38,WRLED@40

NETWORK:

ADCS             = CONF (XADCS,)                % A/D CS'                %
CMCWR            = CONF (XCMCWR,)               % Write CAMAC Buffers    %
CMCRD            = CONF (XCMCRD,)               % Read CAMAC Buffers     %
WAIT             = CONF (XWAIT,)                % WAIT to U14            %
ZCLK             = TONF (VCC,10MHZ,,,)          % 5MHZ clock to Z8036    %
TIMERCS          = CONF (XTIMERCS,)             % Z8036 CS1              %
RAMRD            = CONF (XRAMRD,)               % RAM OE'                %
RAMEN            = CONF (XRAMEN,)               % RAM CS'                %
ROMRD            = CONF (XROMRD,)               % ROM OE'                %
IOSEL            = CONF (XIOSEL,)               % U35 Select             %
NVRCS            = CONF (XNVRCS,)               % NVRAM CS'              %
WRDAC            = CONF (XWRDAC,)               % CLK to DAC latches     %
DUART            = CONF (XDUART,)               % 2692 CS'               %
FIFORD           = CONF (XFIFORD,)              % TCLK FIFO OE'          %
CLKCS            = CONF (XCLKCS,)               % TCLK Mask RAM Enable   %
IORD             = CONF (XIORD,)                % I/O Read Strobe        %
IOWR             = CONF (XIOWR,)                % I/O Write Strobe       %
RDLED            = CONF (XRDLED,)               % CLK to LED latch       %
WRLED            = CONF (XWRLED,)               % OE' to LED latch       %
10MHZ            = INP (10MHZ)                  % 10MHz clock input      %
A7               = INP (A7)                     % Address Lines          %
A8               = INP (A8)
A9               = INP (A9)
A10              = INP (A10)
A11              = INP (A11)
A12              = INP (A12)
A13              = INP (A13)
A14              = INP (A14)
A15              = INP (A15)
DS               = INP (DS)                     % Z8K Data strobe        %
RW               = INP (RW)                     % Z8k R/W line           %
IOREQ            = INP (IOREQ)                  % I/O cycle              %
RAMREQ           = INP (RAMREQ)                 % Memory cycle           %
ROMREQ           = INP (ROMREQ)                 % Instruction cycle      %

EQUATIONS:

%       RAM and ROM Control Lines                                          %

XRAMRD           = /(/RAMREQ * RW * A15
                 +   /ROMREQ * RW * A15);
XRAMEN           = /(/RAMREQ *  A15
                 +   /ROMREQ * A15);
XROMRD           = /(/ROMREQ * /A15
                 +   /RAMREQ * /A15);

%       I/O Read and Write Strobes                                         %
```

```
XIORD           = /(/IOREQ *  RW);
XIOWR           = /(/IOREQ * /RW * /DS);

%       Z8K WAIT Line (There is an automatic 1 wait state on all I/O)   %
%       Insert one wait state for the Z8036 and for the 2692            %
%                                               Address: 0000 - 01FF    %

XWAIT = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * /A9);

%       Z8036 Select line (Active High)         Address: 0000 - 00FF    %

XTIMERCS =  (/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * /A9 * /A8);

%       2692 DUART Select line                  Address: 0100 - 01FF    %

XDUART   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * /A9 *  A8);

%       TCLK Mask RAM                           Address: 0200 - 03FF    %

XCLKCS   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * A9);

%       TCLK FIFO                               Address: 0400 - 047F    %

XFIFORD  = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 * /A8
            * /A7 *  RW);

%       Front Panel LED Read and Write lines    Address: 0480 - 04FF    %

XRDLED   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 * /A8
            *  A7 *  RW);

XWRLED   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 * /A8
            *  A7 * /RW * /DS);

%       D/A Write Strobe                        Address: 0500 - 057F    %

XWRDAC   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 *  A8
            * /A7 * /RW * /DS);

%       AD574 A/D Converter                     Address: 0580 - 05FF    %

XADCS    = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 *  A8
            *  A7);

%       Camac Buffer Read and Write Lines       Address: 0600 - 067F    %

XCMCRD   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 *  A9 * /A8
            * /A7 *  RW * /DS);

XCMCWR   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 *  A9 * /A8
            * /A7 * /RW * /DS);

%       I/O to Viking Connector                 Address: 0680 - 06FF    %

XIOSEL   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 *  A9 * /A8
            *  A7);

%       Non Volatile Ram                        Address: 1000 - 1FFF    %
XNVRCS   = /(/IOREQ * /A15 * /A14 * /A13 *  A12);


END$
```

```
XIORD           = /(/IOREQ *  RW);
XIOWR           = /(/IOREQ * /RW * /DS);

%       Z8K WAIT Line (There is an automatic 1 wait state on all I/O)   %
%       Insert one wait state for the Z8036 and for the 2692            %
%                                               Address: 0000 - 01FF    %

XWAIT = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * /A9);

%       Z8036 Select line (Active High)         Address: 0000 - 00FF    %

XTIMERCS =  (/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * /A9 * /A8);

%       2692 DUART Select line                  Address: 0100 - 01FF    %

XDUART   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * /A9 *  A8);

%       TCLK Mask RAM                           Address: 0200 - 03FF    %

XCLKCS   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 * /A10 * A9);

%       TCLK FIFO                               Address: 0400 - 047F    %

XFIFORD  = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 * /A8
            * /A7 *  RW);

%       Front Panel LED Read and Write lines    Address: 0480 - 04FF    %

XRDLED   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 * /A8
            *  A7 *  RW);

XWRLED   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 * /A8
            *  A7 * /RW * /DS);

%       D/A Write Strobe                        Address: 0500 - 057F    %

XWRDAC   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 *  A8
            * /A7 * /RW * /DS);

%       AD574 A/D Converter                     Address: 0580 - 05FF    %

XADCS    = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 * /A9 *  A8
            *  A7);

%       Camac Buffer Read and Write Lines       Address: 0600 - 067F    %

XCMCRD   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 *  A9 * /A8
            * /A7 *  RW * /DS);

XCMCWR   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 *  A9 * /A8
            * /A7 * /RW * /DS);

%       I/O to Viking Connector                 Address: 0680 - 06FF    %

XIOSEL   = /(/IOREQ * /A15 * /A14 * /A13 * /A12 * /A11 *  A10 *  A9 * /A8
            *  A7);

%       Non Volatile Ram                        Address: 1000 - 1FFF    %
XNVRCS   = /(/IOREQ * /A15 * /A14 * /A13 *  A12);


END$
```

```
W. R. Knopf
RD/Controls
17-APR-1988
1.00
A
EP600
Tevatron Clock Reg.
LogiCaps Schematic Capture Ver 1.6

OPTIONS: TURBO = ON, SECURITY = OFF
PART: EP600
INPUTS:
   SCLK1@1,CLKRD@11,SCLK2@13,SDATA@23

OUTPUTS:
   CNTCLR@3,DAV@9,Q7@22,Q6@21,Q5@20,Q4@19,Q3@18,
   Q2@17,Q1@16,Q0@15,PARFF@10,C3@7,C2@6,C1@5,C0@4

NETWORK:
nQ0 = NOT(Q0f) % SYM 1 %
C3t = AND(C2t,C2f) % SYM 2 %
C2t = AND(C1f,C0F) % SYM 3 %
C0t = NOT(CLRt) % SYM 4 %
nSDATA = NOT(SDATA) % SYM 5 %
nCLRs = NOT(CLRs) % SYM 6 %
nCLRr = NOR(nSDATA,nCNTCLR,C3f,C2f,C1f,C0F) % SYM 7 %
Q6s = AND(Q5f,nHOLD) % SYM 8 %
SCLK2 = INP(SCLK2) % SYM 9 %
nCNTCLR = NOT(CNTCLR) % SYM 10 %
DAV = NOR(..138180,nCNTCLR) % SYM 11 %
CLRt = OR(nCLRr,CLRs) % SYM 12 %
nQ5 = NOT(Q5f) % SYM 13 %
CNTCLR,CNTCLR = RORF(CLRt,SCLK1,,,) % SYM 14 %
Q6h = NOR(nHOLD,nQ6) % SYM 15 %
Q7t = OR(Q7s,Q7h) % SYM 16 %
Q6t = OR(Q6s,Q6h) % SYM 17 %
Q5t = OR(Q5s,Q5h) % SYM 18 %
Q4t = OR(Q4s,Q4h) % SYM 19 %
Q3t = OR(Q3s,Q3h) % SYM 20 %
Q2t = OR(Q2s,Q2h) % SYM 22 %
Q1t = OR(Q1s,Q1h) % SYM 23 %
RSt = NOR(SDATA,nCNTCLR,PAR) % SYM 24 %
SDATA = INP(SDATA) % SYM 25 %
Q7,Q7f = RORF(Q7t,SCLK2,,,CLKOE) % SYM 26 %
Q6,Q6f = RORF(Q6t,SCLK2,,,CLKOE) % SYM 27 %
Q5,Q5f = RORF(Q5t,SCLK2,,,CLKOE) % SYM 28 %
Q4,Q4f = RORF(Q4t,SCLK2,,,CLKOE) % SYM 29 %
Q3,Q3f = RORF(Q3t,SCLK2,,,CLKOE) % SYM 30 %
DAV = CONF(DAV,) % SYM 31 %
nQ7 = NOT(Q7f) % SYM 32 %
CLKOE = INP(CLKRD) % SYM 33 %
SCLK1 = INP(SCLK1) % SYM 34 %
Q2,Q2f = RORF(Q2t,SCLK2,,,CLKOE) % SYM 35 %
Q7h = NOR(nHOLD,nQ7) % SYM 36 %
Q5h = NOR(nHOLD,nQ5) % SYM 37 %
Q5s = AND(Q4f,nHOLD) % SYM 38 %
Q0h = NOR(nHOLD,nQ0) % SYM 39 %
PARFF,PAR = TOTF(PARt,SCLK1,,,) % SYM 40 %
C3,C3f = TOTF(C3t,SCLK1,CNTCLR,,) % SYM 41 %
C2,C2f = TOTF(C2t,SCLK1,CNTCLR,,) % SYM 42 %
C1,C1f = TOTF(C0F,SCLK1,CNTCLR,,) % SYM 43 %
C0,C0F = TOTF(C0t,SCLK1,CNTCLR,,) % SYM 44 %
Q0,Q0f = RORF(Q0t,SCLK2,,,CLKOE) % SYM 45 %
Q1,Q1f = RORF(Q1t,SCLK2,,,CLKOE) % SYM 46 %
Q7s = AND(Q6f,nHOLD) % SYM 47 %
```

```
Q1s = AND(Q0f,nHOLD) % SYM 48 %
nQ6 = NOT(Q6f) % SYM 49 %
Q1h = NOR(nHOLD,nQ1) % SYM 50 %
nQ1 = NOT(Q1f) % SYM 51 %
Q2s = AND(Q1f,nHOLD) % SYM 52 %
Q2h = NOR(nHOLD,nQ2) % SYM 53 %
nQ2 = NOT(Q2f) % SYM 54 %
Q3s = AND(Q2f,nHOLD) % SYM 55 %
Q3h = NOR(nHOLD,nQ3) % SYM 56 %
nQ3 = NOT(Q3f) % SYM 57 %
Q4s = AND(Q3f,nHOLD) % SYM 58 %
Q4h = NOR(nHOLD,nQ4) % SYM 59 %
nQ4 = NOT(Q4f) % SYM 60 %
nHOLD = NOR(CNTCLR,C3f) % SYM 61 %
Q0t = OR(Q0s,Q0h) % SYM 62 %
Q0s = AND(SDATA,nHOLD) % SYM 63 %
CLRs = AND(C0F,C3f) % SYM 64 %
DTt = AND(nCLRs,SDATA,nCNTCLR) % SYM 65 %
PARt = OR(DTt,RSt) % SYM 66 %
..138180 = NOT(PAR) % SYM 67 %

END$
```



TITLE	CAMAC INTERRUPT CONTR.		
COMPANY	Fermilab, RD/Controls		
DESIGNER	Walter R. Knopf		
SIZE B	EPLD FPM5032	NUMBER 1.00	REV B
DATE Aug—1—1989		SHEET 1	OF
TURBO ON		SECURITY OFF	