```
TITLE "PROGRAPE-1 CONTROL CPLD";
%           _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
          _/_/_/_/_/                  ctrl.tdf        _/_/_/_/_/_/_/_/_/_/ |
        _/_/_/_/                                          _/_/_/_/_/_/_/_/  /
      _/_/_/_/       Controllers for PROGRAPE-1 Board       _/_/_/_/_/_/  / |
    _/_/_/_/_/_/   Copyright(c) 1997 by Tsuyoshi Hamada     _/_/_/_/_/  / |
  _/_/_/_/_/_/_/        July. 97 First Version        _/_/_/_/_/_/_/  / |
_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/  / |
|______________________________________________________________|/ |
 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |               %

FUNCTION dropdiff (clk ,in)     -- differential circuit  (react to dropping edge)
RETURNS (out);

FUNCTION delay (in)             -- asynchronous delay generator (about 20.7ns)
RETURNS (out);

CONSTANT        DWIDTH = 32;
CONSTANT        AWIDTH = 19;

SUBDESIGN ctrl
(
        clk, rst                        : INPUT;
        /io_req                         : INPUT;
        hdata[DWIDTH-1..0]              : BIDIR;
        /dma_ack                        : INPUT;
        /d_read                         : OUTPUT;
        data[DWIDTH-1..0]               : BIDIR;
        /dma_r                          : OUTPUT;
        /dma_w                          : OUTPUT;
        index[AWIDTH-6..0]              : OUTPUT;
        adrs[7..0]                      : OUTPUT;
        mwe[3..0]                       : OUTPUT;       -- Memory Write Enable
        /pwe                            : OUTPUT;       -- Pipeline Write Enable
        /pcs[15..0]                     : OUTPUT;       -- Pipeline Chip Select
        prun                            : OUTPUT;
        /d_write                        : OUTPUT;
)
VARIABLE
        gclk,grst                       : NODE;         -- GLOBAL
        pdata[DWIDTH-1..0]              : DFF;
        pdatad[DWIDTH-1..0]             : DFF;
        result[DWIDTH-1..0]             : DFF;
        phase:  MACHINE WITH STATES (crt, dmar, dmaw, run);
        p_y[1..0]                       : DFF;
        p_z[1..0]                       : NODE;
        /pipcs[15..0]                   : DFF;
        /pcs[15..0]                     : DFF;
        hibtri[DWIDTH-1..0]             : TRI;
        piptri[DWIDTH-1..0] : TRI;
        tri_oe                          : DFF; -- GND:HIB => PRO, VCC:PRO =>HIB
        index[AWIDTH-6..0]              : DFF;
        adrs[7..0]                      : DFF;
-- COMMAND READ TIMING
        /d_read                 : DFF;
        drop_req                : dropdiff;
        get                     : dropdiff;
        getd                    : DFF;
        getdd                   : DFF;
        end_crt                 : dropdiff;
        drop_reqd               : DFF;
        comnd[DWIDTH-1..0]      : DFF;
-- DMA_READ
        /dma_r                  : DFF;
        /dma_ackd               : DFF;
        hwords[11..0]           : DFF;
```

```
        end_dmar                : DFF;
        nset[15..0]             : DFF;
        /pipwe                  : DFF;
        /pwe                    : DFF;
        mwe[3..0]               : DFF;
        dclk                    : delay;
-- DMA_WRITE
        drop_ack                : dropdiff;
        /dma_w                  : DFF;
        start_w                 : DFF;
        pwords[7..0]            : DFF;
        end_dmaw                : DFF;
        /pd_write[5..0]         : DFF;
-- RUN
        rn_count[15..0]         : DFF;
        end_run                 : DFF;
        rn_cd[15..0]            : DFF;
        prun                    : DFF;
        pprun                   : DFF;
        ppprun                  : DFF;

        BEGIN
        DEFAULTS
                /pd_write[] = VCC;
        END DEFAULTS;
        gclk = GLOBAL(clk);
        grst = GLOBAL(rst);
        phase.clk       = gclk;
        phase.reset     = !grst;
        p_y[].clk       = gclk;
        p_y[].clrn      = grst;
        index[].clk = gclk;
        index[].clrn = grst;
        adrs[].clk  = gclk;
        adrs[].clrn = grst;
        pdata[].clk     = gclk;
        pdata[].clrn = grst;
        pdatad[].clk    = gclk;
        pdatad[].clrn   = grst;
        result[].clk = gclk;
        result[].clrn = grst;

        -- only used in DMA_READ and DMA_WRITE
        /dma_ackd.clk   = gclk;     % _|_| in common between DMA_READ and DMA_WEITE. _|_|%
        /dma_ackd.prn   = grst;     % _|_| in common between DMA_READ and DMA_WEITE. _|_|%
        /dma_ackd.d  = /dma_ack;    % _|_| in common between DMA_READ and DMA_WEITE. _|_|%

        /pipcs[].clk    = gclk;
        IF !((p_z[] == 1) # (p_z[] == 2)) THEN
                /pipcs[].prn    = grst;
                /pipcs[].d = VCC;
        END IF;
        -- Tri State Buffer control --|
        tri_oe.clk      = gclk;     --|
        tri_oe.clrn     = grst;     --|
        IF !/dma_w.q THEN           --|
                tri_oe.d = VCC;     --|
        ELSE                        --|
                tri_oe.d = GND;     --|
        END IF;                     --|
        ----------------------------|
        -- Tri State Buffer connection --|
        hibtri[].oe = tri_oe.q;        --|
        piptri[].oe = !tri_oe.q;       --|
                                       --|
        pdata[].d = hdata[];           --|
```

```
        pdatad[].d = pdata[].q;         --|
        piptri[].in = pdatad[].q;       --|
        data[] = piptri[].out;          --|
        result[].d = data[];            --|
        hibtri[].in = result[].q;       --|
        hdata[] = hibtri[].out;         --|
        ----------------------------------|

% _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
 _/_/_/_/          Phase transition          _/_/_/_/_/
_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/  %
        TABLE
                phase,   p_y[].q        =>  p_z[],              phase;
                crt,            0               =>              0,              crt;
                crt,            1               =>              1,              dmar;
                crt,            2               =>              2,              dmaw;
                dmar,           0               =>              1,              dmar;
                dmar,           1               =>              0,              crt;
                dmar,           2               =>              3,              run;
                run,            0               =>              3,              run;
                run,            1               =>              0,              crt;
                dmaw,           0               =>              2,              dmaw;
                dmaw,           1               =>              0,              crt;
        END TABLE;
% _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
 _/_/_/_/  COMMAND READ TIMING (P_Z==0:crt)  _/_/_/_/_/
_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/  %
        /d_read.clk     = gclk;
        /d_read.prn     = grst;
        drop_req.clk = gclk;
        drop_req.in  = /io_req;
        get.clk          = gclk;
        get.in           = !/d_read;
        getd.clk         = gclk;
        getd.d           = get.out;
        getdd.clk        = gclk;
        getdd.d          = getd.q;
        end_crt.in       = getd.q;
        end_crt.clk      = gclk;
        drop_reqd.clk   = gclk;
        drop_reqd.d             = drop_req.out;
        comnd[].clk             = gclk;
        comnd[].clrn    = grst;

IF p_z[] == 0 THEN

        /d_read.d = !(drop_req.out # drop_reqd.q);
        IF getdd.q THEN
                comnd[].d = pdata[].q;
        ELSE
                comnd[].d = comnd[].q;
        END IF;
        TABLE
                end_crt.out,    comnd[DWIDTH-1].q       =>      p_y[].d;
                        0,                                      X                       =>              0;
                        1,                                      1                       =>              1;
                        1,                                      0                       =>              2;
        END TABLE;

ELSE
        comnd[].d = comnd[].q;
        /d_read.d = /d_read.q;
END IF;


% _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
```

```
 _/_/_/_/       DMA_READ        (P_Z == 1)   _/_/_/_/_/
_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/  %
        /dma_r.clk              = gclk;
        /dma_r.prn      = grst;
        hwords[].clk    = gclk;
        end_dmar.clk    = gclk;
        end_dmar.clrn   = grst;
        nset[].clk              = gclk;
        nset[].clrn             = grst;
        /pipwe.clk              = gclk;
        /pipwe.prn              = grst;
        dclk.in                 = gclk;
        mwe[].clk               = gclk;
        mwe[].prn               = grst;
        /pwe.clk                = gclk;
        /pwe.prn                = grst;
        /pcs[].clk              = gclk;
        /pcs[].prn              = grst;
        /pwe.d  = /pipwe.q;
        /pcs[].d = /pipcs[].q;

IF p_z[] == 1 THEN
        hwords[].clrn = VCC; -- counter is available
        /dma_r.d = GND;
        ------------------------------------------------------------| XXXXXXXX  C A U T I O N  ! XXXXXXXX
        % requirement of transition to phase 'crt'.%             --| X  If HIB send a extra low signal X
        end_dmar.d = (hwords[].q == comnd[11..0]+1);         --| X of /dma_ack to PROGRAPE-1,      X
        IF end_dmar.q   THEN                                    --| X PROGRAPE-1 writes a wrong data  X
                p_y[].d = 1;                                    --| X into own Memory.                X
        ELSE                                                    --| XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                IF !/dma_ackd.q THEN                            --|
                        hwords[].d = hwords[].q + 1;            --|
                ELSE                                            --|
                        hwords[].d = hwords[].q;                --|
                END IF;                                         --|
                IF comnd[DWIDTH-2..DWIDTH-6] == B"01110"  THEN  --|  --> RUN
                        p_y[].d = 2;                            --|
                ELSE                                            --|
                        p_y[].d = 0;                            --|
                END IF;                                         --|
        END IF;                                                 --|
        ------------------------------------------------------------|
        ------------------------------------------------------------------------|
        %=*=*=*=*=*=*=*=*=*=*=   command decode   =*=*=*=*=*=*=*=*=*=*=*=*=%   --|
        IF      comnd[DWIDTH-2..DWIDTH-6] == B"00000"   THEN  -- 0000..03FFF --|
                mwe0.d = /dma_ackd.q;                                           --|
        ELSE                                                                    --|
                mwe0.d = VCC;                                                   --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"00001" THEN  -- 04000..07FFF          --|
                mwe1.d = /dma_ackd.q;                                           --|
        ELSE                                                                    --|
                mwe1.d = VCC;                                                   --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"00010" THEN  -- 08000..0BFFF          --|
                mwe2.d = /dma_ackd.q;                                           --|
        ELSE                                                                    --|
                mwe2.d = VCC;                                                   --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"00011" THEN  -- 0C000..0FFFF          --|
                mwe3.d = /dma_ackd.q;                                           --|
        ELSE                                                                    --|
                mwe3.d = VCC;                                                   --|
        END IF;                                                                 --|
        IF  (comnd[DWIDTH-2] == B"1")  THEN                                     --|
                /pipwe.d = /dma_ack;    --  Write Enable of ALL Pipelines    --|
```

```
        ELSE                                                                --|
                /pipwe.d= VCC;                                              --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10000" THEN  -- pipline0 CS       --|
                /pipcs0.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs0.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10001" THEN  -- pipline1 CS       --|
                /pipcs1.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs1.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10010" THEN  -- pipline2 CS       --|
                /pipcs2.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs2.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10011" THEN  -- pipline3 CS       --|
                /pipcs3.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs3.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10100" THEN  -- pipline4 CS       --|
                /pipcs4.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs4.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10101" THEN  -- pipline5 CS       --|
                /pipcs5.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs5.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10110" THEN  -- pipline6 CS       --|
                /pipcs6.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs6.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10111" THEN  -- pipline7 CS       --|
                /pipcs7.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs7.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11000" THEN  -- pipline8 CS       --|
                /pipcs8.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs8.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11001" THEN  -- pipline9 CS       --|
                /pipcs9.d = /dma_ack;                                       --|
        ELSE                                                                --|
                /pipcs9.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11010" THEN  -- pipline10 CS      --|
                /pipcs10.d = /dma_ack;                                      --|
        ELSE                                                                --|
                /pipcs10.d = VCC;                                           --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11011" THEN  -- pipline11 CS      --|
                /pipcs11.d = /dma_ack;                                      --|
        ELSE                                                                --|
                /pipcs11.d = VCC;                                           --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11100" THEN  -- pipline12 CS      --|
                /pipcs12.d = /dma_ack;                                      --|
        ELSE                                                                --|
```

```
                /pipcs12.d = VCC;                                               --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11101" THEN  -- pipline13 CS          --|
                /pipcs13.d = /dma_ack;                                          --|
        ELSE                                                                    --|
                /pipcs13.d = VCC;                                               --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11110" THEN  -- pipline14 CS          --|
                /pipcs14.d = /dma_ack;                                          --|
        ELSE                                                                    --|
                /pipcs14.d = VCC;                                               --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11111" THEN  -- pipline15 CS          --|
                /pipcs15.d = /dma_ack;                                          --|
        ELSE                                                                    --|
                /pipcs15.d = VCC;                                               --|
        END IF;                                                                 --|
        IF      comnd[DWIDTH-2..DWIDTH-6] == B"01100"   THEN  -- set N          --|
                IF !/dma_ackd.q THEN                                            --|
                        nset[].d = pdata[15..0].q;                              --|
                ELSE                                                            --|
                        nset[].d = nset[].q;                                    --|
                END IF;                                                         --|
        ELSE                                                                    --|
                nset[].d = nset[].q;                                            --|
        END IF;                                                                 --|
        ------------------------------------------------------------------------|
        index[] = (comnd[25..24],hwords[].q);
        adrs[]  = hwords[7..0].q;
ELSE
        /dma_r.d = VCC;
        mwe[]    = grst;
        hwords[].clrn = GND;  -- counter is disabailable
        nset[].d = nset[].q;  -- This N is constant during other state.
        /pipwe.d    = VCC;
END IF;
% _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
  _/_/_/_/          DMA_WRITE (P_Z==2:dmaw)          _/_/_/_/_/
_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/  %
        drop_ack.clk = gclk;
        /dma_w.clk      = gclk;
        /dma_w.prn      = grst;
        start_w.clk             = gclk;
        start_w.clrn    = grst;
        pwords[].clk    = gclk;
        end_dmaw.clk    = gclk;
        end_dmaw.clrn   = grst;
        /pd_write[].clk  = gclk;
        /pd_write1.d = /pd_write0.q;
        /pd_write2.d = /pd_write1.q;
        /pd_write3.d = /pd_write2.q;
        /pd_write4.d = /pd_write3.q;
        /pd_write5.d = /pd_write4.q;
        /d_write = /pd_write5.q;

IF p_z[] == 2 THEN
  --------------------------
  /dma_w.d = GND;
  drop_ack.in = /dma_ack;
  IF drop_ack.out THEN
                start_w.d = VCC;
  ELSE
                start_w.d = start_w.q;
  END IF;
  IF start_w.q THEN  -- ######################################################
  --------------------------
```

```
        pwords[].clrn = VCC; -- counter is available
        end_dmaw.d = (pwords[].q == comnd[7..0] + 1);
        IF end_dmaw.q THEN
                p_y[].d = 1;
        ELSE
                pwords[].d = pwords[].q + 1;
        END IF;
        adrs[].d = pwords[].q - 1;

        ------------------------------------------------------------------------|
  IF !end_dmaw.q THEN                                                           --|
        /pipcs[].prn = VCC;                                                     --|
        /pd_write0.prn = VCC;                                                   --|
        %=*=*=*=*=*=*=*=*=*=*=   command decode   =*=*=*=*=*=*=*=*=*=*=*=%   --|
        IF comnd[DWIDTH-2] == B"1" THEN  -- /d_write pulse                      --|
                /pd_write0.d = GND;                                             --|
        ELSE                                                                    --|
                /pd_write0.d = VCC;                                             --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10000" THEN  -- pipline0 CS           --|
                /pipcs0.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs0.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10001" THEN  -- pipline1 CS           --|
                /pipcs1.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs1.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10010" THEN  -- pipline2 CS           --|
                /pipcs2.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs2.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10011" THEN  -- pipline3 CS           --|
                /pipcs3.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs3.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10100" THEN  -- pipline4 CS           --|
                /pipcs4.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs4.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10101" THEN  -- pipline5 CS           --|
                /pipcs5.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs5.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10110" THEN  -- pipline6 CS           --|
                /pipcs6.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs6.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"10111" THEN  -- pipline7 CS           --|
                /pipcs7.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs7.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11000" THEN  -- pipline8 CS           --|
                /pipcs8.d = GND;                                                --|
        ELSE                                                                    --|
                /pipcs8.d = VCC;                                                --|
        END IF;                                                                 --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11001" THEN  -- pipline9 CS           --|
                /pipcs9.d = GND;                                                --|
```

```
        ELSE                                                                --|
                /pipcs9.d = VCC;                                            --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11010" THEN  -- pipline10 CS      --|
                /pipcs10.d = GND;                                           --|
        ELSE                                                                --|
                /pipcs10.d = VCC;                                           --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11011" THEN  -- pipline11 CS      --|
                /pipcs11.d = GND;                                           --|
        ELSE                                                                --|
                /pipcs11.d = VCC;                                           --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11100" THEN  -- pipline12 CS      --|
                /pipcs12.d = GND;                                           --|
        ELSE                                                                --|
                /pipcs12.d = VCC;                                           --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11101" THEN  -- pipline13 CS      --|
                /pipcs13.d = GND;                                           --|
        ELSE                                                                --|
                /pipcs13.d = VCC;                                           --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11110" THEN  -- pipline14 CS      --|
                /pipcs14.d = GND;                                           --|
        ELSE                                                                --|
                /pipcs14.d = VCC;                                           --|
        END IF;                                                             --|
        IF comnd[DWIDTH-2..DWIDTH-6] == B"11111" THEN  -- pipline15 CS      --|
                /pipcs15.d = GND;                                           --|
        ELSE                                                                --|
                /pipcs15.d = VCC;                                           --|
        END IF;                                                             --|
  ELSE                                                                      --|
        /pipcs[].prn = GND;                                                 --|
        /pd_write0.prn = GND;                                               --|
  END IF;                                                                   --|
        ------------------------------------------------------------------------|
  END IF; --#############################################################
ELSE
        !/dma_w.d = !/pd_write2.q # !/pd_write3.q # !/pd_write4.q;
        /pipcs[].prn = VCC;
        start_w.d = GND;
        pwords[].clrn = GND;  -- counter is disabailable
END IF;
% _/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/
  _/_/_/_/         RUN   (P_Z==3:run)          _/_/_/_/_/
_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/  %
        rn_count[].clk  = gclk;
        rn_count[].clrn = grst;
        end_run.clk      = gclk;
        end_run.clrn = grst;
        rn_cd[].clk  = gclk;
        rn_cd[].clrn = grst;
        prun.clk = gclk;
        prun.prn = grst;
        prun.d   = pprun.q;
        pprun.clk = gclk;
        pprun.prn = grst;
        ppprun.clk = gclk;
        ppprun.prn = grst;
        pprun.d   = ppprun.q;

IF p_z[] == 3 THEN
        ppprun.d = VCC;
        pprun.clrn = VCC;
```

```
        end_run.d = (rn_count[].q == 0);
        IF end_run.q   THEN
                p_y[].d = 1;
        ELSE
                rn_count[].d = rn_count[].q - 1;
                p_y[].d = 0;
        END IF;
        -------------------------
        rn_cd[].d = rn_count[].q;
        index[].d = rn_cd[13..0].q - 1;

ELSE
        rn_count[].d = nset[].q;
        ppprun.d = GND;
        pprun.clrn = GND;
END IF;


%---------------%
END;
```