TB_RATEGEN.vhd

```
--============================================================--
--  A PECL Rate-Generator modeling for MTI/Vsys
--  This TEST BENCH EXAMPLE is designed for www Demo
--
--      by Yoshiaki Naruse   1998/9/10 Systems Workshop Inc.
-- ------------------------------------------------------------
--
-- Inputs : Clock(PLL Max400MHz/0.5% step programmable ),
--          Program values : Prog_Master(Rate),Prog_Slave(Witth)
--          External trigg : ExtStep when operating @ExtTrig
--          Mode Control   : Operating mode :ExtTrig  0:FreeRun
--
-- Outputs: Rate-signal    : nTC_Rate
--          Width-signal   : nTC_Slave
--
-- ======= Block descriptions ( val's in clock count ) ====--
--
--                    Prog_Master        Prog_Slave
--                   (Delay/Rate val)    (Width val)
--
--                           |                |
-- FreeRun/ExtTrig    +-----+-----+     +-----+----+
--     MModeLatch --->|           |     |          |
--                    | SubCkt    |     | SubCkt   |
--         ExtTrig --->|   Master  +---->+  Slave   +---->
--          (STEP)    |           |     |          |
--                    +-----------+     +----------+
--


-- ======= Waveform descriptions ( Tic's in clock count ) ====--
--
--  Master            <--  Rate value -->
--  SubCktRateGen   --_------------------_------------------_--
--
--
--  Slave              <-- Width-->
--  SubCktRateGen   ___------------_______------------_______--
--
--
```

TB_RATEGEN.vhd

```
        library ieee;
        use ieee.Std_Logic_1164.all;
        use ieee.Std_Logic_unsigned.all;
        use work.BasePack.all;
45      use work.PECL_Comp.all;
        -- ------------------------------------------------------------ --
        -- A package PECL_Comp is to simplify descriptions, and to show
        -- how to describe a "configuration"  --> see TB_config.vhd .

50      --=============================================================--
        Entity TestBench is
        Generic( ClockTic       : TIME := 1250 ps; -- 400 MHz Clock max
                 MaxMasterVAL   : integer := 255    );
        end;
55
        --=============================================================--
        Architecture TB of TestBench is

        -- component declarations done external to this archtecture
60
            -- Utility nets ----------------------------------------
            signal MasterVAL: integer;
            signal SlaveVAL : integer;
            signal NetVCC   : std_logic;
65          signal NetGND   : Std_Logic;
            -- Internal OnBoard wiring signals ------------------------
            signal CLOCK,nCLOCK : Std_Logic;
            signal INITCLR,INIT0: Std_Logic;
            signal CLR,RESET    : Std_Logic;
70          signal pLOAD        : Std_Logic;
            -- ReLoadable counter program inputs ----------------------
            signal Prog_Master  : BYTE;
            signal Prog_Slave   : BYTE;
            -- TerminalCount signal outputs ---------------------------
75          signal nTC_Master   : Std_Logic;
            signal nTC_Slave    : Std_Logic;
            -- Counter operation mode control signals -----------------
            signal MMode        : Std_Logic; -- 1:ExtTrig  0:FreeRun
            signal nMMode       : Std_Logic;
80          signal SMode        : Std_Logic; -- 1:Sync     0:Async
            signal nSMode       : Std_Logic;
            -- Wiring nets --------------------------------------------
            signal MModeLatch   : Std_Logic;
            signal SModeLatch   : Std_Logic;
85          signal pMasterSS    : Std_Logic;
            signal nMasterSS    : Std_Logic;
            signal pSlaveSS     : Std_Logic;
            signal nSlaveSS     : Std_Logic;
            signal ExtSTEP      : Std_Logic;
90          signal pSTEP        : Std_Logic;
            signal ExtStep0     : Std_Logic;
            signal ExtStep1     : Std_Logic;
            signal MMode0       : Std_Logic;
            signal MMode1       : Std_Logic;
95          signal SMode0       : Std_Logic;
            signal SMode1       : Std_Logic;
```

TB_RATEGEN.vhd

```
      --=============================================================--
      Begin
100   ----  Fixed inputs on PCBD wiring might be as follows -----
      NetVCC  <= PECL_PULLUP;
      NetGND  <= PECL_OPEN;

      ---- Initialize On-Board devices for simulation -----------

105   INITCLR <= '1','0'after 20 ns;
      INIT0   <= '1','0'after  2 ns;

      ---- Mode switch Master side ( FreeRun / External step )
      MMode <=    '0',    -- 1:ExtTrig  0:FreeRun
110               '1' after 1000 ns,  '0' after 2000 ns,
                  '1' after 3000 ns,  '0' after 4000 ns,
                  '1' after 5000 ns ;

      ---- Mode switch Slave side ( user's input ): Sync to Master
115   SMode <=    '1';        -- 1:Sync  0:Async

      nMMode <= not( MMode or INITCLR);    --  PCBD needs a gete here
      nSMode <= not( SMode or INITCLR);    -- or extrnl FPGA will do.

120   ------------------------------------------------------------------
      CLK_GEN : Process   -- You can modify ClockTic /w MTI DialogBox
      Begin   CLOCK<='0'; wait for ClockTic;
              CLOCK<='1'; wait for ClockTic;
      end Process CLK_GEN;
125
      nCLOCK  <= not CLOCK;

      ------------------------------------------------------------------
      DFFBuf: E141  -- E141 is used as a single-clk width pulse shaper
130    port map(  CLK=> CLOCK,
                  MR => INIT0,
                  DL => NetGND,   DR => NetGND,
                  SEL0 => NetGND, SEL1 => NetGND, -- "Load" mode E141

135               Da => INITCLR,  Qa => CLR,
                  Db => CLR,      Qb => RESET,
                  Dc => ExtSTEP,  Qc => ExtStep0, -- External Step In
                  Dd => ExtStep0, Qd => ExtStep1, -- External STEP
                  De => nSMode,   Qe => SMode0,
140               Df => SMode1,   Qf => SModeLatch,
                  Dg => nMMode,   Qg => MMode0,
                  Dh => MMode1,   Qh => MModeLatch    );

      MModeSwitch: EL58   -- Master Mode signal latch timing ----
145    port map(  SELIN=> nTC_Master, -- Switch @ nTC timing only
                  Da   => MModeLatch, -- Hold on current Mode'
                  Db   => MMode0,     -- @ nTC, get MMode0 FF
                  pQ   => MMode1,     -- put result to MMode1FF
                  nQ   => OPEN    );
150
      SModeSwitch: EL58   -- Slave Mode signal latch timing ----
       port map(  SELIN=> nTC_Slave,  -- Switch @ nTC timing only
                  Da   => SModeLatch, -- Hold on current Mode'
                  Db   => SMode0,     -- @ nTC, get MMode0 FF
155               pQ   => SMode1,     -- put result to MMode1FF
                  nQ   => OPEN    );
```

TB_RATEGEN.vhd

```
       ExtStepSS: EL58     -- External Step triggers MasterRateGenerator
        port map(  SELIN=> ExtStep0,    -- @ Cold-Start,MODE='0'
160                Da   => pLOAD,       -- During ExtStep0 ='1'
                   Db   => ExtStep1,    -- when   ExtStep0 ='0'
                   pQ   => pSTEP,       -- Trigger output
                   nQ   => OPEN    );

165    nLOAD_INV: EL58     -- Power On clear will initialize counters -
        port map(  SELIN=> CLR,
                   Da   => NetGND,      -- During CLR='1'
                   Db   => RESET,       -- when   CLR='0'
                   pQ   => pLOAD,       -- Initial Loading signal
170                nQ   => OPEN    );


       -- Two sub-circuit units are used in this design ---------------

175    Master: SubCktRateGen   -- Master unit of "Rate" Generator
       generic map( SbCktName=> "Mastr:" )
       port map( CLOCK     =>  CLOCK,
                 CLR       =>  INIT0,       -- Power On reset.
                 ModeLatch =>  MModeLatch,  -- Sync/Async control in
180              ProgValue =>  Prog_Master,-- RepRate(Delay) Value
                 pLOAD     =>  pLOAD,
                 pSTEP     =>  pSTEP,       -- External trigger input
                 nTCOUT    =>  nTC_Master,  -- Delay/Rep output
                 pTCSS     =>  pMasterSS,   -- 1 clock width Rate output
185              nTCSS     =>  nMasterSS  );

       Slave: SubCktRateGen    -- Slave unit "Width" signal generator
       generic map( SbCktName => "Slave:" )
       port map( CLOCK     =>  CLOCK,
190              CLR       =>  INIT0,       -- Power On reset.
                 ModeLatch =>  SModeLatch,  -- Sync/Async control in
                 ProgValue =>  Prog_Slave,  -- Width Value
                 pLOAD     =>  pLOAD,
                 pSTEP     =>  pMasterSS,   -- Slave triggerd by Master
195              nTCOUT    =>  nTC_Slave,   -- Width output
                 pTCSS     =>  pSlaveSS,    -- 1 clock width DONE output
                 nTCSS     =>  nSlaveSS  );
```

TB_RATEGEN.vhd

```
    ------------------------------------------------------------
200 ExternalStep:Process(CLOCK,CLR) -- External STEP generation
    ------------------------------------------------------------
        variable NUMCK  : integer;
    Begin
        if(CLR='1') then
205                 ExtSTEP <= '0'; NUMCK := 0;
        elsif(CLOCK'EVENT and CLOCK='1') then
            if( NUMCK > 60) then
                    ExtStep <= '0'; NUMCK := 0;
            else    ExtStep <= '1'; NUMCK := NUMCK + 1;
210         end if;
        end if;
    end Process ExternalStep;

    ------------------------------------------------------------
215 TestLoop:Process( CLOCK,CLR,nTC_Master )
    ------------------------------------------------------------
        variable Prev_nTC : Std_Logic :='1';
        variable TC_Event : Std_Logic :='1';
        variable RepCount : integer := 0;
220 Begin
      if(CLR='1') then
           MasterVAL    <= 1 ;
           SlaveVAL     <= 1 ;
           Prog_Master  <= not( Int2BYTE( MasterVAL) );
225        Prog_Slave   <= not( Int2BYTE( SlaveVAL ) );

      elsif(CLOCK'EVENT and CLOCK='1') then
         TC_Event := Prev_nTC and not( nTC_Master );
         Prev_nTC := nTC_Master;
230
        if( TC_Event ='1') then
            if( MasterVAL=MaxMasterVAL ) then
                Assert False        report "End of Test @ Max Value"

                severity FAILURE;
235         elsif(RepCount > 4) then
                RepCount := 0;
                MasterVAL   <= MasterVAL + 1;
                SlaveVAL    <= MasterVAL/2 + 1 ;
                Prog_Master <= not( Int2BYTE( MasterVAL) );
240             Prog_Slave  <= not( Int2BYTE( SlaveVAL ) );
            else
                RepCount := RepCount + 1;
            end if;
        end if;
245   end if;
        Assert Now < 5000 ns    report "End of Test  @ End Time"

        severity FAILURE;
    end Process TestLoop;
    end;
```