

```

--
-- TeC7 VHDL Source Code
--   Tokuyama kousen Educational Computer Ver.7
--
-- Copyright (C) 2002-2012 by
--   Dept. of Computer Science and Electronic Engineering,
--   Tokuyama College of Technology, JAPAN
--
-- 上記著作権者は、Free Software Foundation によって公開されている GNU 一般公
-- 衆利用許諾契約書バージョン2に記述されている条件を満たす場合に限り、本ソース
-- コード(本ソースコードを改変したものを含む、以下同様)を使用・複製・改変・再配
-- 布することを無償で許諾する。
--
-- 本ソースコードは*全くの無保証*で提供されるものである。上記著作権者および
-- 関連機関・個人は本ソースコードに関して、その適用可能性も含めて、いかなる保証
-- も行わない。また、本ソースコードの利用により直接的または間接的に生じたいかな
-- る損害に関しても、その責任を負わない。
--
--
-- TeC/tec.vhd : TeC Top Level
--

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

library UNISIM;
use UNISIM.VComponents.all;

entity TEC is
  port(
    P_RESET      : in      std_logic;          -- reset(Negative)
    P_MODE       : in      std_logic_vector(1 downto 0); -- operation mode
    P_CLK        : in      std_logic;          -- 2.4576MHz

    -- CONSOLE(INPUT)
    P_DATA_SW    : in      std_logic_vector(7 downto 0); -- Data SW
    P_RESET_SW   : in      std_logic;
    P_SETA_SW    : in      std_logic;          -- SETA SW
    P_INCA_SW    : in      std_logic;          -- INCA SW
    P_DECA_SW    : in      std_logic;          -- DECA SW
    P_WRITE_SW   : in      std_logic;          -- WRITE SW
    P_STEP_SW    : in      std_logic;          -- STEP SW
    P_BREAK_SW   : in      std_logic;          -- BREAK SW
    P_STOP_SW    : in      std_logic;          -- STOP SW
    P_RUN_SW     : in      std_logic;          -- RUN SW
    P_RCW_SW     : in      std_logic;          -- Rotate SW(CW)
    P_RCCW_SW    : in      std_logic;          -- Rotate SW(CCW)

    -- CONSOLE(OUTPUT)
    P_A_LED      : out     std_logic_vector(7 downto 0); -- Address LED
    P_D_LED      : out     std_logic_vector(7 downto 0); -- Data LED
    P_R_LED      : out     std_logic;          -- RUN LED
    P_C_LED      : out     std_logic;          -- Carry LED
    P_S_LED      : out     std_logic;          -- Sing LED
    P_Z_LED      : out     std_logic;          -- Zero LED
    P_G0_LED     : out     std_logic;          -- G0 LED
    P_G1_LED     : out     std_logic;          -- G1 LED
    P_G2_LED     : out     std_logic;          -- G2 LED
    P_SP_LED     : out     std_logic;          -- SP LED
    P_PC_LED     : out     std_logic;          -- PC LED
    P_MM_LED     : out     std_logic;          -- MM LED
    P_BUZ        : out     std_logic;          -- BUZZER OUT

    -- SIO

```

```

        P_SIO_RXD  : in      std_logic;          -- SIO Receive
        P_SIO_TXD  : out     std_logic;          -- SIO Transmit

        -- PIO
        P_EXT_IN   : in      std_logic_vector (7 downto 0);
        P_ADC_REF  : out     std_logic_vector (7 downto 0);
        P_EXT_OUT  : out     std_logic_vector (7 downto 0)
    );
end TEC;

architecture RTL of TEC is

    -- clock
    signal I_CNT      : std_logic_vector(19 downto 0); -- 分周用バイナリカウンタ
    signal I_2_4kHz   : std_logic;                   -- ブザー等の音源用(2.4kHz)
    signal I_75Hz     : std_logic;                   -- インターバルタイマ用(75Hz)
    signal I_18_75Hz  : std_logic;                   -- SW サンプリング用(18.75Hz)
    signal I_2_3Hz    : std_logic;                   -- LED 点滅用(2.3Hz)

    -- 割り込みコントローラ関係
    signal I_VECT      : std_logic_vector(1 downto 0); -- 割り込み番号
    signal I_INTR      : std_logic;                   -- CPU への割り込み

    -- Address BUS
    signal I_ADDR      : std_logic_vector(7 downto 0); -- アドレスバス(CPUの出力)

    -- Data BUS
    signal I_DOUT_CPU : std_logic_vector(7 downto 0); -- データバス(CPUの出力)
    signal I_DIN_CPU  : std_logic_vector(7 downto 0); -- データバス(CPUの入力)
    signal I_DOUT_RAM : std_logic_vector(7 downto 0); -- データバス(RAMの出力)
    signal I_DOUT_IO  : std_logic_vector(7 downto 0); -- データバス(IOの出力)

    -- Control BUS
    signal I_RESET     : std_logic;                   -- クロック同期済みのRESET
    signal I_LI        : std_logic;                   -- 命令フェッチ(CPUの出力)
    signal I_HL        : std_logic;                   -- HALT命令実行(CPUの出力)
    signal I_ER        : std_logic;                   -- 不正命令実行(CPUの出力)
    signal I_RW        : std_logic;                   -- READ/WRITE(CPUの出力)
    signal I_MR        : std_logic;                   -- メモリ要求(CPUの出力)
    signal I_IR        : std_logic;                   -- 入出力要求(CPUの出力)
    signal I_STOP      : std_logic;                   -- CPU 停止(パネルの出力)
    signal I_INT0       : std_logic;                   -- タイマー割り込み
    signal I_INT1       : std_logic;                   -- SIO 受信割り込み
    signal I_INT2       : std_logic;                   -- SIO 送信割り込み
    signal I_INT3       : std_logic;                   -- コンソール割り込み SW

    -- パネル関係の配線
    signal I_RS_SEL    : std_logic_vector(2 downto 0); -- ロータリースイッチの位置
    signal I_RS_DEC    : std_logic_vector(5 downto 0); -- ロータリースイッチの LED
    signal I_A_LED     : std_logic_vector(7 downto 0); -- アドレス LED の値
    signal I_WRITE     : std_logic;                   -- WRITE SW が押された
    signal I_PINT      : std_logic;                   -- コンソール割り込み
    signal I_G0D       : std_logic_vector(7 downto 0); -- CPU から G0 の値を出力
    signal I_G1D       : std_logic_vector(7 downto 0); -- CPU から G1 の値を出力
    signal I_G2D       : std_logic_vector(7 downto 0); -- CPU から G2 の値を出力
    signal I_SPD       : std_logic_vector(7 downto 0); -- CPU から SP の値を出力
    signal I_PCD       : std_logic_vector(7 downto 0); -- CPU から PC の値を出力
    signal I_MMD       : std_logic_vector(7 downto 0); -- RAM からの値出力

    -- 内部配線
    signal I_SPK_I      : std_logic;                   -- I/O からスピーカーポート
    signal I_SPK_P      : std_logic;                   -- PANEL からスピーカーポート

    component TEC_PANEL
        port ( P_CLK      : in      std_logic;          -- Clock
              P_2_4kHz    : in      std_logic;          -- 2.4kHz

```

```

P_18_75Hz : in std_logic; -- 18.75Hz
P_2_3Hz : in std_logic; -- 2.3Hz
P_RESET : out std_logic; -- Reset Out Put
P_AIN : in std_logic_vector(7 downto 0); -- ADDR BUS
P_LI : in std_logic; -- Instruction Fetch
P_HL : in std_logic; -- Halt Request
P_ER : in std_logic; -- Error
P_MR : in std_logic; -- Memory Request
P_STOP : out std_logic; -- Stop
P_INT : out std_logic; -- Interrupt SW

-- パネルのスイッチ入力
P_DATA_SW : in std_logic_vector(7 downto 0); -- Data SW
P_RESET_SW : in std_logic; -- Reset SW
P_SETA_SW : in std_logic; -- SETA SW
P_INCA_SW : in std_logic; -- INCA SW
P_DECA_SW : in std_logic; -- DECA SW
P_WRITE_SW : in std_logic; -- WRITE SW
P_STEP_SW : in std_logic; -- STEP SW
P_BREAK_SW : in std_logic; -- BREAK SW
P_STOP_SW : in std_logic; -- STOP SW
P_RUN_SW : in std_logic; -- RUN SW
P_RCW_SW : in std_logic; -- Rotate SW(CW)
P_RCCW_SW : in std_logic; -- Rotate SW(CCW)

-- パネルへの出力
P_R_LED : out std_logic; -- Run LED
P_SPK : out std_logic; -- 操作音の出力
P_A_LED : out std_logic_vector(7 downto 0); -- Address LED
P_SEL : out std_logic_vector(2 downto 0); -- Rotate SW(Output)
P_WRITE : out std_logic; -- WRITEスイッチの操作
);
end component;

component TEC_INTC
port ( P_CLK : in std_logic; -- Clock
P_RESET : in std_logic; -- Reset
P_LI : in std_logic; -- Instruction fetch
P_MR : in std_logic; -- Memory access

P_INT0 : in std_logic; -- INT0 (Timer)
P_INT1 : in std_logic; -- INT1 (SIO RXD)
P_INT2 : in std_logic; -- INT2 (SIO TXD)
P_INT3 : in std_logic; -- INT3 (Console)

P_INTR : out std_logic; -- Interrupt
P_VECT : out std_logic_vector(1 downto 0); -- 割り込み番号
);
end component;

component TEC_CPU
port ( P_CLK : in std_logic; -- Clock
P_RESET : in std_logic; -- Reset
P_ADDR : out std_logic_vector(7 downto 0); -- ADDRESS BUS
P_DIN : in std_logic_vector(7 downto 0); -- DATA BUS
P_DOUT : out std_logic_vector(7 downto 0); -- DATA BUS
P_LI : out std_logic; -- Instruction Fetch
P_HL : out std_logic; -- Halt Request
P_ER : out std_logic; -- Decode Error
P_RW : out std_logic; -- Read/Write
P_MR : out std_logic; -- Memory Request
P_IR : out std_logic; -- I/O Request
P_INTR : in std_logic; -- Interrupt
P_STOP : in std_logic; -- Stop

P_WRITE : in std_logic; -- Panel Write

```

```

P_SEL   : in  std_logic_vector(2 downto 0);    -- Panel RotarySW Pos
P_PND   : in  std_logic_vector(7 downto 0);    -- Panel Data
P_C      : out std_logic;                      -- Carry   Flag
P_S      : out std_logic;                      -- Sign    Flag
P_Z      : out std_logic;                      -- Zero    Flag
P_G0D    : out std_logic_vector(7 downto 0);   -- G0 out
P_G1D    : out std_logic_vector(7 downto 0);   -- G1 out
P_G2D    : out std_logic_vector(7 downto 0);   -- G2 out
P_SPD    : out std_logic_vector(7 downto 0);   -- SP out
P_PCD    : out std_logic_vector(7 downto 0);   -- PC out

P_MODE   : in std_logic                      -- DEMO MODE
);
end component;

component TEC_IO
port ( P_CLK      : in  std_logic;              -- CLK
      P_2_4kHz    : in  std_logic;              -- Pi!
      P_75Hz      : in  std_logic;              -- 75Hz(タイマー用)
      P_RESET     : in  std_logic;              -- Reset
      P_RW        : in  std_logic;
      P_IR        : in  std_logic;
      P_ADDR      : in  std_logic_vector(3 downto 0);
      P_DOUT      : out std_logic_vector(7 downto 0);
      P_DIN       : in  std_logic_vector(7 downto 0);
      P_INT_TXD   : out std_logic;              -- SIO 送信割り込み
      P_INT_RXD   : out std_logic;              -- SIO 受信割り込み
      P_INT_TMR   : out std_logic;              -- タイマー割り込み
      P_INT_CON   : out std_logic;              -- コンソール割り込み

      P_INT_SW    : in  std_logic;              -- コンソール割り込みs

      P_DATA_SW   : in  std_logic_vector(7 downto 0);
      P_SPK       : out std_logic;
      P_RXD       : in  std_logic;
      P_TXD       : out std_logic;
      P_EXT_IN    : in  std_logic_vector(7 downto 0);
      P_ADC_REF   : out std_logic_vector(7 downto 0);
      P_EXT_OUT   : out std_logic_vector(7 downto 0)
);
end component;

component TEC_RAM
port ( P_CLK      : in  std_logic;
      P_ADDR      : in  std_logic_vector(7 downto 0);
      P_DOUT      : out std_logic_vector(7 downto 0);
      P_DIN       : in  std_logic_vector(7 downto 0);
      P_RW        : in  std_logic;
      P_MR        : in  std_logic;

      P_PNA       : in  std_logic_vector(7 downto 0); -- パネルアドレス
      P_PND       : in  std_logic_vector(7 downto 0); -- パネル用データ入力
      P_SEL       : in  std_logic_vector(2 downto 0); -- ロータリーSWの位置
      P_WRITE     : in  std_logic;                  -- パネル書き込み信号
      P_MMD       : out std_logic_vector(7 downto 0); -- パネル用データ出力

      P_MODE      : in  std_logic_vector(1 downto 0)
);
end component;

begin
-- クロックを作る
I_2_4kHz  <= I_CNT(9);          -- 2.4kHz tac_panel.pdf
I_75Hz    <= I_CNT(14);         -- 75Hz
I_18_75Hz <= I_CNT(15);         -- 18.75Hz
I_2_3Hz   <= I_CNT(19);         -- 2.3Hz

```

```

process(P_CLK, P_RESET)
begin
    if (P_RESET='0') then
        I_CNT <= "00000000000000000000";
    elsif (P_CLK'event and P_CLK='1') then
        I_CNT <= I_CNT + 1;
    end if;
end process;

-- I/O とパネルのスピーカ出力を合成する
P_BUZ <= I_SPK_I xor I_SPK_P;

-- パネル
P_A_LED <= I_A_LED;
panel0 : TEC_PANEL
port map ( P_CLK      => P_CLK,
           P_2_4kHz    => I_2_4kHz,
           P_18_75Hz  => I_18_75Hz,
           P_2_3Hz    => I_2_3Hz,
           P_RESET    => I_RESET,
           P_AIN       => I_ADDR,
           P_LI        => I_LI,
           P_HL        => I_HL,
           P_ER        => I_ER,
           P_MR        => I_MR,
           P_STOP      => I_STOP,

           P_RESET_SW  => P_RESET_SW,
           P_DATA_SW   => P_DATA_SW,
           P_SETA_SW   => P_SETA_SW,
           P_INCA_SW   => P_INCA_SW,
           P_DECA_SW   => P_DECA_SW,
           P_WRITE_SW  => P_WRITE_SW,
           P_STEP_SW   => P_STEP_SW,
           P_BREAK_SW  => P_BREAK_SW,
           P_STOP_SW   => P_STOP_SW,
           P_RUN_SW    => P_RUN_SW,
           P_RCW_SW    => P_RCW_SW,
           P_RCCW_SW   => P_RCCW_SW,

           P_R_LED     => P_R_LED,
           P_SPK       => I_SPK_P,
           P_A_LED     => I_A_LED,
           P_SEL       => I_RS_SEL,
           P_WRITE     => I_WRITE,
           P_INT       => I_PINT
        );

-- 割込みコントローラ
intr0 : TEC_INTC
port map ( P_CLK      => P_CLK,
           P_RESET    => I_RESET,
           P_LI       => I_LI,
           P_MR       => I_MR,

           P_INT0     => I_INT0,
           P_INT1     => I_INT1,
           P_INT2     => I_INT2,
           P_INT3     => I_INT3,

           P_INTR     => I_INTR,
           P_VECT     => I_VECT
        );

-- CPU
cpu0 : TEC_CPU

```

```

port map ( P_CLK    => P_CLK,
           P_RESET  => I_RESET,
           P_ADDR   => I_ADDR,
           P_DIN    => I_DIN_CPU,
           P_DOUT   => I_DOUT_CPU,
           P_LI     => I_LI,
           P_HL     => I_HL,
           P_ER     => I_ER,
           P_RW     => I_RW,
           P_MR     => I_MR,
           P_IR     => I_IR,
           P_INTR   => I_INTR,
           P_STOP   => I_STOP,

           P_WRITE  => I_WRITE,
           P_SEL    => I_RS_SEL,
           P_PND    => P_DATA_SW,
           P_C      => P_C_LED,
           P_S      => P_S_LED,
           P_Z      => P_Z_LED,
           P_G0D    => I_G0D,
           P_G1D    => I_G1D,
           P_G2D    => I_G2D,
           P_SPD    => I_SPD,
           P_PCD    => I_PCD,

           P_MODE   => P_MODE(1)
        );

-- 主記憶
ram0: TEC_RAM
port map ( P_CLK    => P_CLK,
           P_ADDR   => I_ADDR,
           P_DOUT   => I_DOUT_RAM,
           P_DIN    => I_DOUT_CPU,
           P_RW     => I_RW,
           P_MR     => I_MR,

           P_PNA    => I_A_LED,
           P_PND    => P_DATA_SW,
           P_SEL    => I_RS_SEL,
           P_WRITE  => I_WRITE,
           P_MMD    => I_MMD,

           P_MODE   => P_MODE
        );

-- 周辺回路
io0: TEC_IO
port map ( P_CLK    => P_CLK,
           P_2_4kHz => I_2_4kHz,
           P_75Hz  => I_75Hz,
           P_RESET => I_RESET,
           P_RW    => I_RW,
           P_IR    => I_IR,
           P_ADDR  => I_ADDR(3 downto 0),
           P_DOUT  => I_DOUT_IO,
           P_DIN   => I_DOUT_CPU,
           P_INT_TXD => I_INT2,
           P_INT_RXD => I_INT1,
           P_INT_TMR => I_INT0,
           P_INT_CON => I_INT3,

           P_INT_SW  => I_PINT,
           P_DATA_SW => P_DATA_SW,
           P_SPK     => I_SPK_I,

```

```

        P_TXD      => P_SIO_TXD,
        P_RXD      => P_SIO_RXD,
        P_EXT_IN   => P_EXT_IN,
        P_ADC_REF  => P_ADC_REF,
        P_EXT_OUT  => P_EXT_OUT
    );

-- データバスでCPUの入力を決定する部分
I_DIN_CPU <= I_DOUT_RAM      when (I_MR='1') else -- RAM
             "110111" & I_VECT when (I_LI='1') else -- Vector Read
             I_DOUT_IO;      -- I/O

-- データLED
with I_RS_SEL select
    P_D_LED <= I_G0D when "000",      -- G0
               I_G1D when "001",      -- G1
               I_G2D when "010",      -- G2
               I_SPD when "011",      -- SP
               I_PCD when "100",      -- PC
               I_MMD when others;     -- MM

-- ロータリースイッチの表示
with I_RS_SEL select
    I_RS_DEC <= "100000" when "000",  -- G0
               "010000" when "001",  -- G1
               "001000" when "010",  -- G2
               "000100" when "011",  -- SP
               "000010" when "100",  -- PC
               "000001" when others; -- MM

P_G0_LED <= I_RS_DEC(5);
P_G1_LED <= I_RS_DEC(4);
P_G2_LED <= I_RS_DEC(3);
P_SP_LED <= I_RS_DEC(2);
P_PC_LED <= I_RS_DEC(1);
P_MM_LED <= I_RS_DEC(0);

end RTL;

```