

```

--
-- TeC7 VHDL Source Code
--   Tokuyama kousen Educational Computer Ver.7
--
-- Copyright (C) 2002-2011 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 RAM
--
library IEEE;
use std.textio.all;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_textio.all;

entity TEC_RAM is
  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); -- ロータリースイッチの位置
    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) -- 0,1:通常, 2:デモ1, 3:デモ2
  );
-- デモ1 : 電子オルゴールプログラム入力済
-- デモ2 : 電子オルゴールプログラムとデータが入力済
end TEC_RAM;

architecture BEHAVE of TEC_RAM is
  subtype word is std_logic_vector(7 downto 0);
  type memory is array(0 to 1023) of word;
  function read_file (fname : in string) return memory is
    file data_in : text is in fname;
    variable line_in: line;
    variable ram : memory;
  begin
    for i in 0 to 1023 loop
      readline (data_in, line_in);
      read(line_in, ram(i));
    end loop;
    return ram;
  end function;
  signal mem : memory := read_file("tec_ram.txt");

  signal deca   : std_logic; -- CPU のアドレスデコード結果
  signal wea    : std_logic; -- CPU 側書き込み信号

```

```

signal addr10a: std_logic_vector(9 downto 0); -- CPU 側アドレス
signal decb    : std_logic;                  -- パネルのアドレスデコード結果
signal web     : std_logic;                  -- パネル側書き込み信号
signal addr10b: std_logic_vector(9 downto 0); -- パネル側アドレス

begin
    -- アドレスをラッチする(BLOCK RAMになる)
    process(P_CLK)
    begin
        if (P_CLK'event and P_CLK='0') then
            addr10a <= P_MODE & P_ADDR;
            addr10b <= P_MODE & P_PNA;
        end if;
    end process;

    -- 読み出し制御
    P_DOUT <= mem( conv_integer(addr10a) );
    P_MMD  <= mem( conv_integer(addr10b) );

    -- 書き込み制御
    process(P_CLK)
    begin
        if (P_CLK'event and P_CLK='0') then
            if (wea='1') then
                mem( conv_integer(addr10a) ) <= P_DIN;
            elsif (web='1') then
                mem( conv_integer(addr10b) ) <= P_PND;
            end if;
        end if;
    end process;

    -- MODE=0,1 の時は E0H~FFH が書き込み不可
    -- MODE=2   の時は、加えて 80H~BFH が書き込み不可
    -- MODE=3   の時は、更に加えて 00H~7FH が書き込み不可

    -- CPU からの書き込み制御
    wea <= P_MR and P_RW and deca;
    process(P_MODE, addr10a(7), addr10a(6), addr10a(5))
    begin
        case P_MODE is
            when "00" =>
                deca <= not addr10a(7) or not addr10a(6) or not addr10a(5);
            when "01" =>
                deca <= not addr10a(7) or not addr10a(6) or not addr10a(5);
            when "10" =>
                deca <= not addr10a(7) or (addr10a(6) and not addr10a(5));
            when others =>
                deca <= addr10a(7) and addr10a(6) and not addr10a(5);
        end case;
    end process;

    -- パネルからの書き込み制御
    web <= P_WRITE and P_SEL(2) and not P_SEL(1) and P_SEL(0) and decb;
    process(P_MODE, addr10b(7), addr10b(6), addr10b(5))
    begin
        case P_MODE is
            when "00" =>
                decb <= not addr10b(7) or not addr10b(6) or not addr10b(5);
            when "01" =>
                decb <= not addr10b(7) or not addr10b(6) or not addr10b(5);
            when "10" =>
                decb <= not addr10b(7) or (addr10b(6) and not addr10b(5));
            when others =>
                decb <= addr10b(7) and addr10b(6) and not addr10b(5);
        end case;
    end process;
end process;

```

```
end BEHAVE;
```