```
----- pic.vhd -----
--*****************************************
--
--  Interrupt Controller for SH4  ver 1.0
--
--  programmed by kaneko  Mar.27 2004
--
--*****************************************
library ieee;
use ieee.std_logic_1164.all;

--=========================================
--
--  entity
--
--=========================================
entity pic is
    port(
        CLK     : in    std_logic;                      -- clock
        RSTn    : in    std_logic;                      -- reset
        CS1n    : in    std_logic;                      -- chip select 1
        CS2n    : in    std_logic;                      -- chip select 2
        RDn     : in    std_logic;                      -- read pulse
        WRn     : in    std_logic;                      -- write pulse
        DB      : inout std_logic_vector(7 downto 0);   -- data bus
        IRL     : out   std_logic_vector(3 downto 0);   -- IRL
        IRQ     : in    std_logic_vector(7 downto 0)    -- IRQ
        );
end pic;

--=========================================
--
--  architecture
--
--=========================================
architecture RTL of pic is
signal  MREG    : std_logic_vector(7 downto 0);         -- mask register
signal  INT     : std_logic_vector(7 downto 0);         -- interrupts
signal  RDi     : std_logic;                            -- selected read pulse
signal  IRLi    : std_logic_vector(3 downto 0);         -- internal IRL
begin
    ----- simple -----
--  IRL(0) <= IRQ(0);
--  IRL(1) <= IRQ(1);
--  IRL(2) <= IRQ(2);
--  IRL(3) <= IRQ(4);

    ----- make INT with priority -----
    INT( 0) <= IRQ( 0) or MREG( 0);
    INT( 1) <= IRQ( 1) or MREG( 1);
    INT( 2) <= IRQ( 2) or MREG( 2);
    INT( 3) <= IRQ( 3) or MREG( 3);
    INT( 4) <= IRQ( 4) or MREG( 4);
    INT( 5) <= IRQ( 5) or MREG( 5);
    INT( 6) <= IRQ( 6) or MREG( 6);
    INT( 7) <= IRQ( 7) or MREG( 7);

    ----- make IRL by INT -----
    IRLi <= "0111" when INT(7) = '0' else
            "0110" when INT(6) = '0' else
            "0101" when INT(5) = '0' else
            "0100" when INT(4) = '0' else
            "0011" when INT(3) = '0' else
            "0010" when INT(2) = '0' else
            "0001" when INT(1) = '0' else
            "0000" when INT(0) = '0' else
            "1111";

    process(CLK)
    begin
        if(CLK'event and CLK = '0') then
            IRL <= IRLi;
        end if;
    end process;

    ----- make mask register -----
    process(RSTn, CS1n, WRn, CS2n)
    begin
        if(RSTn = '0') then
            MREG <= "11111111";
        elsif(WRn'event and WRn = '1') then
            if(CS1n = '0' and CS2n = '0') then
                MREG <= DB;
            end if;
        end if;
    end process;

    ----- make DB -----
```

```
    RDi <= CS1n or CS2n or RDn;
    DB <= MREG when RDi = '0' else (others => 'Z');

end RTL;
```