Solutions to Assignment 7

Question 1

This design uses a `high_bit` circuit that is instantiated twice to obtain bits-map with the highest-priority bits in the ISR and IRR registers. The remaining code updates the IRR and ISR register contents based on the type of CPU cycle and generates the interrupt request and data bus (interrupt type) outputs.

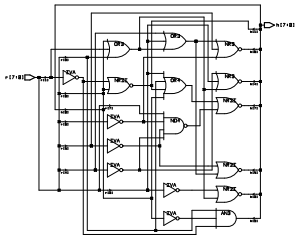
```
-- ELEC 379 Assignment 7
-- greatly simplified version of an 8259 priority
-- interrupt controller (PIC)
-- Ed Casas 98/3/26

-- compute a bit mask that has only the highest
-- priority bit set (bit 0 is highest priority)

library ieee ;
use ieee.std_logic_1164.all ;

entity high_bit is
   port ( r : in std_logic_vector (7 downto 0) ;
          h : out std_logic_vector (7 downto 0) ) ;
end high_bit ;

architecture rtl of high_bit is
begin
   h <=
      "00000001"  when r(0) = '1' else
      "00000010"  when r(1) = '1' else
      "00000100"  when r(2) = '1' else
      "00001000"  when r(3) = '1' else
      "00010000"  when r(4) = '1' else
      "00100000"  when r(5) = '1' else
      "01000000"  when r(6) = '1' else
      "10000000"  when r(7) = '1' else
      "00000000"  ;
end rtl ;


-- PIC component package

library ieee ;
use ieee.std_logic_1164.all ;

package pic_components is

   component high_bit
   port ( r : in std_logic_vector (7 downto 0) ;
          h : out std_logic_vector (7 downto 0) ) ;
   end component ;

end pic_components ;
```

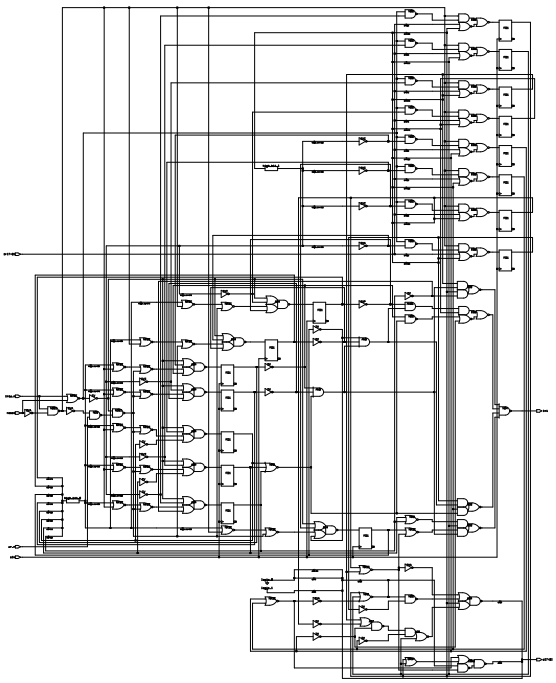
```
-- Simple version of 8259 PIC (priority interrupt
-- controller).  See Assignment 7 for details.

library ieee ;
use ieee.std_logic_1164.all ;
use work.pic_components.all ;

entity pic is
   port ( clk, reset, wr_n, inta_n : in std_logic ;
      ir : in std_logic_vector (7 downto 0) ;
      int : out std_logic ;
      d : out std_logic_vector (7 downto 0) ) ;
end pic ;

architecture rtl of pic is
   signal irr, isr : std_logic_vector (7 downto 0) ;
   signal high_irr, high_isr : std_logic_vector (7 downto 0) ;
   signal next_irr, next_isr : std_logic_vector (7 downto 0) ;
   signal irrmask : std_logic_vector (7 downto 0) ;

   for all : high_bit use entity work.high_bit(rtl) ;

begin

   -- highest bit of IRR
   h0: high_bit port map ( irr, high_irr ) ;

   -- highest bit of ISR
   h1: high_bit port map ( isr, high_isr ) ;

   -- irr set/reset
   next_irr <=
     "00000000" when reset = '1' else
     ( irr or ir ) and ( not high_irr ) when inta_n = '0' else
     irr or ir ;

   --- isr set/reset
   next_isr <=
     "00000000" when reset = '1' else
     isr or high_irr when inta_n = '0' else
     isr and ( not high_isr ) when wr_n = '0' else
     isr ;

   -- registers
   process(clk)
   begin
      if clk'event and clk = '1' then
         irr <= next_irr ;
         isr <= next_isr ;
      end if ;
   end process ;

   -- mask of interrupts allowed
   irrmask <=
      "00000000"  when isr(0) = '1' else
      "00000001"  when isr(1) = '1' else
      "00000011"  when isr(2) = '1' else
      "00000111"  when isr(3) = '1' else
      "00001111"  when isr(4) = '1' else
      "00011111"  when isr(5) = '1' else
```

```
      "00111111"  when isr(6) = '1' else
      "01111111"  when isr(7) = '1' else
      "11111111"  ;

   -- interrupt request
   int <=
      '0' when ( irrmask and irr ) = "00000000" else
      '1' ;

   -- data bus output (40H plus highest interrupt)
   d <=
      "01000000" when irr(0) = '1' else
      "01000001" when irr(1) = '1' else
      "01000010" when irr(2) = '1' else
      "01000011" when irr(3) = '1' else
      "01000100" when irr(4) = '1' else
      "01000101" when irr(5) = '1' else
      "01000110" when irr(6) = '1' else
      "01000111" ;

end rtl ;
```

Schematics

There are two schematics: one for the `high_bit` entity and one for the pic itself:





Simulation Log

```
# run
Test Input: 0 clk=0 reset=1 wr_n=1 inta_n=1 ir=00000000
    Output: int=1 d=01000111
  Required: int=0 d=00000000
Test Input: 1 clk=1 reset=1 wr_n=1 inta_n=1 ir=00000000
    Output: int=0 d=01000111
  Required: int=0 d=01000111
Test Input: 2 clk=0 reset=0 wr_n=1 inta_n=1 ir=00000000
    Output: int=0 d=01000111
  Required: int=0 d=01000111
Test Input: 3 clk=1 reset=0 wr_n=1 inta_n=1 ir=00000000
    Output: int=0 d=01000111
  Required: int=0 d=01000111
Test Input: 4 clk=0 reset=0 wr_n=1 inta_n=1 ir=00001000
    Output: int=0 d=01000111
  Required: int=0 d=01000111
Test Input: 5 clk=1 reset=0 wr_n=1 inta_n=1 ir=00001000
    Output: int=1 d=01000011
  Required: int=1 d=01000011
Test Input: 6 clk=0 reset=0 wr_n=1 inta_n=0 ir=00000000
    Output: int=1 d=01000011
  Required: int=1 d=01000011
Test Input: 7 clk=1 reset=0 wr_n=1 inta_n=0 ir=00000000
    Output: int=0 d=01000111
  Required: int=0 d=01000111
Test Input: 8 clk=0 reset=0 wr_n=1 inta_n=1 ir=01000001
    Output: int=0 d=01000111
  Required: int=0 d=01000111
Test Input: 9 clk=1 reset=0 wr_n=1 inta_n=1 ir=01000001
    Output: int=1 d=01000000
  Required: int=1 d=01000000
Test Input: 10 clk=0 reset=0 wr_n=1 inta_n=0 ir=00000000
    Output: int=1 d=01000000
  Required: int=1 d=01000000
Test Input: 11 clk=1 reset=0 wr_n=1 inta_n=0 ir=00000000
    Output: int=0 d=01000110
  Required: int=0 d=01000110
Test Input: 12 clk=0 reset=0 wr_n=1 inta_n=1 ir=00000000
    Output: int=0 d=01000110
  Required: int=0 d=01000110
Test Input: 13 clk=1 reset=0 wr_n=1 inta_n=1 ir=00000000
    Output: int=0 d=01000110
  Required: int=0 d=01000110
Test Input: 14 clk=0 reset=0 wr_n=0 inta_n=1 ir=00000000
    Output: int=0 d=01000110
  Required: int=0 d=01000110
Test Input: 15 clk=1 reset=0 wr_n=0 inta_n=1 ir=00000000
    Output: int=0 d=01000110
  Required: int=0 d=01000110
Test Input: 16 clk=0 reset=0 wr_n=0 inta_n=1 ir=00000000
    Output: int=0 d=01000110
  Required: int=0 d=01000110
Test Input: 17 clk=1 reset=0 wr_n=0 inta_n=1 ir=00000000
    Output: int=1 d=01000110
  Required: int=1 d=01000110
Test Input: 18 clk=0 reset=0 wr_n=1 inta_n=0 ir=00000000
    Output: int=1 d=01000110
  Required: int=1 d=01000110
Test Input: 19 clk=1 reset=0 wr_n=1 inta_n=0 ir=00000000
    Output: int=0 d=01000111
  Required: int=0 d=01000111
(vhdlsim): Simulation complete, time is 20000000000 FS.
```