Using the FPGA Editor

Introduction

• This week we will have a part lecture/part tutorial on using the FPGA Editor
  – generate a design which cannot be represented in VHDL
Basic design

VHDL base code (34 bit version)

-- divide by 16 using a 4 bit counter

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity sr34 is
  port(clk: in std_logic; din1: in std_logic; dout: out std_logic);
end sr34;

architecture synthesis of sr34 is
begin

q <= (others => '0');
dout1 <= '0';
din2 <= '0';
dout2 <= '0';
force1 <= '0';

component ram16x1s
  PORT(D: IN std_logic; A0: IN std_logic; A1: IN std_logic; A2: IN std_logic; A3: IN std_logic; WCLK: IN std_logic; O: OUT std_logic);
end component;

-- architecture synthesis of sr34 is
signal q: std_logic_vector(3 downto 0);
signal dout1: std_logic;  
signal din2: std_logic;  
signal dout2: std_logic;  
signal force1: std_logic;

end synthesis;
VHDL base code cont’d

begin
  force1 <= '1';
  process(clk)
  begin
    if (clk'event and clk = '1') then
      q(3) <= q(2);
      q(2) <= q(1);
      q(1) <= q(0);
      q(0) <= (not(q(3) xor q(2))) xor (q(2) and q(1) and q(0));
    end if;
  end process;
end process;

mem1 : ram16x1s port map
  ( D => din1, A0 => q(0),
    A1 => q(1), A2 => q(2),
    A3 => q(3), WE => force1,
    WCLK => clk, O => dout1);
mem2 : ram16x1s port map
  ( D => din2, A0 => q(0),
    A1 => q(1), A2 => q(2),
    A3 => q(3), WE => force1,
    WCLK => clk, O => dout2);
end synthesis;

Modifications using EPIC

- Understand design
- Change pinout
- Swap component locations
- Remove nets, add nets
- Add components
- Modify designs
Things to do

• Trace a path using FPGA editor and the timing analyzer
• Swap CLBs and IOBs
• Change to use CLB FF instead of IOB

Conclusions

• Using EPIC we can directly view the resultant layout
  – understand the logic, placement and routing
  – make modifications to the design