Comp 331 – Introduction to Hardware

De-bounce Circuits
Clean Clock Pulse Circuit
Random Number Generator

Pushbutton Bounce Problem

De-bounce Circuit: inp = ‘1’ for 3 clock pulses

See Listing 7.12 page 184-185

architecture debounce4 of debounce4 is
signal delay1, delay2, delay3 : STD_VECTOR_VECTOR(3 downto 0);
beg
begin
process(cclk, clr, inp) – inp on sensitivity list?
begin
if clr = '1' then
  delay1 <= "0000";
  delay2 <= "0000";
  delay3 <= "0000";
elsif cclk'event and cclk = '1' then
  delay1 <= inp;
  delay2 <= delay1;
  delay3 <= delay2;
  end if;
end process;
outp <= delay1 and delay2 and delay 3;
end debounce4;

Exercise 1: Implement and Simulate (see Figure 7.27)
the debounce4.vhd file given on pages 184-85

See if you can “duplicate” Figure 7.27

Clock Pulse: Input produces “clean pulse” no matter how long input button is pressed

See Listing 7.13 page 186

4-Bit Pseudo-Random Generator

There is some “tricky timing” going on here

Note – This is Set
architecture rand4 of rand4 is
signal qs: STD_LOGIC_VECTOR(3 downto 0);
begin
  process(clk, clr)
  begin
    if clr = '1' then
      qs <= "0001";
    elsif clk'event and clk = '1' then
      qs(3) <= qs(0) xor qs(3);
      qs(2 downto 0) <= qs(3 downto 1);
    end if;
  end process;
  q <= qs;
end rand4;

Exercise 2: Design and implement on the BASYS-2 board
rand8_top.vhd, the 8-bit pseudo random number generator
that displays 8-bit pseudo-random numbers on the two right-
most digits of the 7-segment display. See previous slide for
details.

Use the clk3 output from the clkdiv module to advance the
pseudo-random number generator component rand8.

The x7seg.vhd component (leading zeros are blank) to display
your digits (see Listing 5.23 page 97 – Example 15b from
9/25/12). Alternately use x7seg.vhd