ELE 3510 HW5 Name_____

1 – Create the 2 test processes described below and provide a simulation. Assume an existing clk process running with period PER and a resetb process that releases reset on the 2nd falling clock edge 50pts

You can use any existing design as the top level block for compiling the code

Periodic input “foo” with period 4 times the clock period and a 25% duty cycle that runs for 7 “foo” periods then stops. Synchronized to the falling clock edge.

8 bit signal “boo” that increments every 6th cycle and repeats infinitely

ELE 3510 HW5 Name_____

1 – Create the 2 test processes described below and provide a simulation.

running with period PER and a reset b

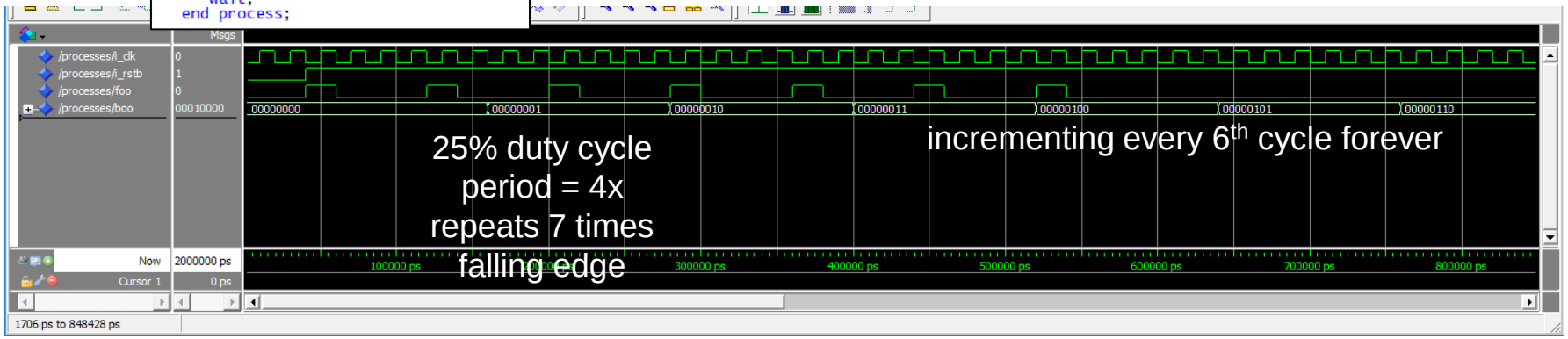
ne 2nd

Periodic input “foo” with period 4 times the clock period and a 25% duty cycle that runs for 7 “foo” periods then stops. Synchronized to the falling clock edge.

8 bit signal “boo” that increments every 6th cycle and repeats infinitely

```
-- foo process
foo_proc: process
begin
   -- initialize value
   foo <= '0';

   -- wait for reset
   wait for 2*PER;

   -- already on falling clock edge

   -- create a 7 cycle loop
   for i in 1 to 7 loop
      foo <= '1';
      wait for 1*PER;
      foo <= '0';
      wait for 3*PER;
   end loop;

   -- cause it to stop
   wait;
end process;
```

```
boo_proc: process
begin
   -- initialize value
   boo <= (others => '0');

   -- wait for reset
   wait for 2*PER;

   -- no clock edge specified - use falling

   -- create infinite process
   loop
      wait for 6*PER;
      boo <= std_logic_vector(unsigned(boo) + 1);
   end loop;
end process;
```



ELE 3510 HW5 Name_____

2 – Create the testbench and simulation for the JK Flip-Flop from problem 2 of HW 4. Be sure you can clearly see and identify each mode of operation 50pts

```
------------------------------
--
-- ff_jk_special_tb.vhdl
--
-- created: 9/6/18
-- by: johnsontimoj
-- rev: 0
--
-- testbench for ff_jk_special
--
------------------------------
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity ff_jk_special_tb is
   -- no entry - testbench
end entity;

architecture testbench of ff_jk_special_tb is
   signal   CLK:  std_logic;
   signal   RSTB: std_logic;
   signal   SET:  std_logic;
   signal   J:    std_logic;
   signal   K:    std_logic;
   signal   Q:    std_logic;
   signal   Qb:   std_logic;

   constant PER:  time:= 20 ns;

   ---------------------------------
   -- Component prototype
   ---------------------------------
COMPONENT ff_jk_special
    port (
         i_clk :  in std_logic;
         i_rstb : in std_logic;
         i_set :  in std_logic;
         i_J :    in std_logic;
         i_K :    in std_logic;

         o_Q :    out std_logic;
         o_Qb :   out std_logic
   );
END COMPONENT;
   -------------------------------------
```

```
begin

   -------------------------------------
   -- Device under test (DUT)
   -------------------------------------
   DUT: ff_jk_special
      port map(
               i_clk    => CLK,
               i_rstb   => RSTB,
               i_set    => SET,
               i_J      => J,
               i_K      => K,
               o_Q      => Q,
               o_Qb     => Qb
      );


   -------------------------------------
   -- Test processes
   -------------------------------------

   -- Clock process
   clock: process    -- note - no sensitivity
      begin
         CLK <= '0';
         wait for PER/2;
         infinite: loop
            CLK <= not CLK; wait for PER/2;
         end loop;
   end process clock;
```

```
   -- Set process
   set_process: process    -- note - no sensitiv
   begin
      SET <= '1'; wait for 2*PER;
      SET <= '0'; wait;
   end process set_process;

   -- Run Processes
   run: process
   begin
      -- initialize inputs
      RSTB <= '1';
      J <= '0';
      K <= '0';
      -- wait for reset
      wait for 2*PER;

      --00 case
      wait for 3*PER;

      --10 case
      J <= '0';
      K <= '1';
      wait for 3*PER;

      --01 case
      J <= '1';
      K <= '0';
      wait for 3*PER;

      --11 case
      J <= '1';
      K <= '1';
      wait for 2*PER;

      -- rstb case
      RSTB <= '0';
      wait for 3*PER;

   end process;

   -----------------------------------------
   -- End test processes
   -----------------------------------------

end architecture;
```

ELE 3510 HW5 Name_____

2 – Create the testbench and simulation for the JK Flip-Flop from problem 2 of HW 4 50pts