Home work for this Friday

1. Design a 32 bit Adder.
2. Here is D_latch, Dff, JKFF, and 8 divided counter. We set different period clock to get the wave form as following. You can try design the test files to get the wave form.
   First, get understand the truth tables.
   Then, run the code that we have provided to you.
   Finally, try to design your test files (if you have time).

   You need to hand in the report next Friday.

1. D-Latch: Dlatch.vhd

   +-----+-----+-----+-----+  
   |     |     |     |     |  
   +-----+-----+-----+-----+  
   | D   | Q   | D   | Q   |  
   +-----+-----+-----+-----+  
   | Enable | Enable | Enable | Enable |  
   +-----+-----+-----+-----+  
   | 1/0  | 1   | D   | not D |  

   library ieee;
   use ieee.std_logic_1164.all;

   entity D_latch is
     port( D: in std_logic;
           enable: in std_logic;
           Q: out std_logic;
           NQ: out std_logic);
   end D_latch;

   architecture behv of D_latch is
     begin
       process(D, enable)
         begin
           if (enable='1') then
             Q <= D;
             NQ <= not D;
           end if;
       end process;
   end behv;
library ieee;
use ieee.std_logic_1164.all;
entity DFF is
port(  D: in std_logic;
       clock: in std_logic;
       Q: out std_logic;
       NQ: out std_logic
);
end DFF;
architecture behv of DFF is
begin
  process(D, clock)
  begin
    if (clock='1' and clock'event) then
      Q <= D;
      NQ <= not D;
    end if;
  end process;
end behv;
3. J_K flip flop (jkff.vhd)

library ieee;
use ieee.std_logic_1164.all;
entity JKFF is
port (clock: in std_logic;
      J, K: in std_logic;
      reset: in std_logic;
      Q, NQ: out std_logic
);
end JKFF;
architecture behv of JKFF is
signal ss: std_logic;
begin
end behv;
begin
process(clock, reset) is
begin
if (reset='1') then
    ss <= '0';
elsif (rising_edge(clock)) then
    if(j='1' and k='1') then
        ss <= not ss;
    elsif(j='1' and k='0') then
        ss <= '1';
    elsif(j='0' and k='1') then
        ss <= '0';
    else
        null;
    end if;
end if;
end if;
end process;
Q <= ss;
NQ <= not ss;
end behv;
4. Divide-by-8 counter: (using 3 J_KFF)

```vhdl
library ieee;
use ieee.std_logic_1164.all;
use work.jkff;
entity counter8 is
  port( clock8: in std_logic;
       reset8: in std_logic;
       Q8:     out std_logic);
end counter8;
architecture struct of counter8 is
  component JKFF is
    port (clock:     in std_logic;
           J, K:     in std_logic;
           reset:    in std_logic;
           Q, NQ:    out std_logic);
  end component;
  signal ss1: std_logic;
  signal ss2: std_logic;
  begin
    Gate1: JKFF port map (clock=>clock8,J=>'1',K=>'1',reset=>reset8,Q=>ss1);
    Gate2: JKFF port map (clock=>ss1,J=>'1',K=>'1',reset=>reset8,Q=>ss2);
    Gate3: JKFF port map (clock=>ss2,J=>'1',K=>'1',reset=>reset8,Q=>Q8);
  end struct;
```