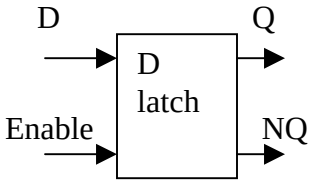
Home work for this Friday

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).

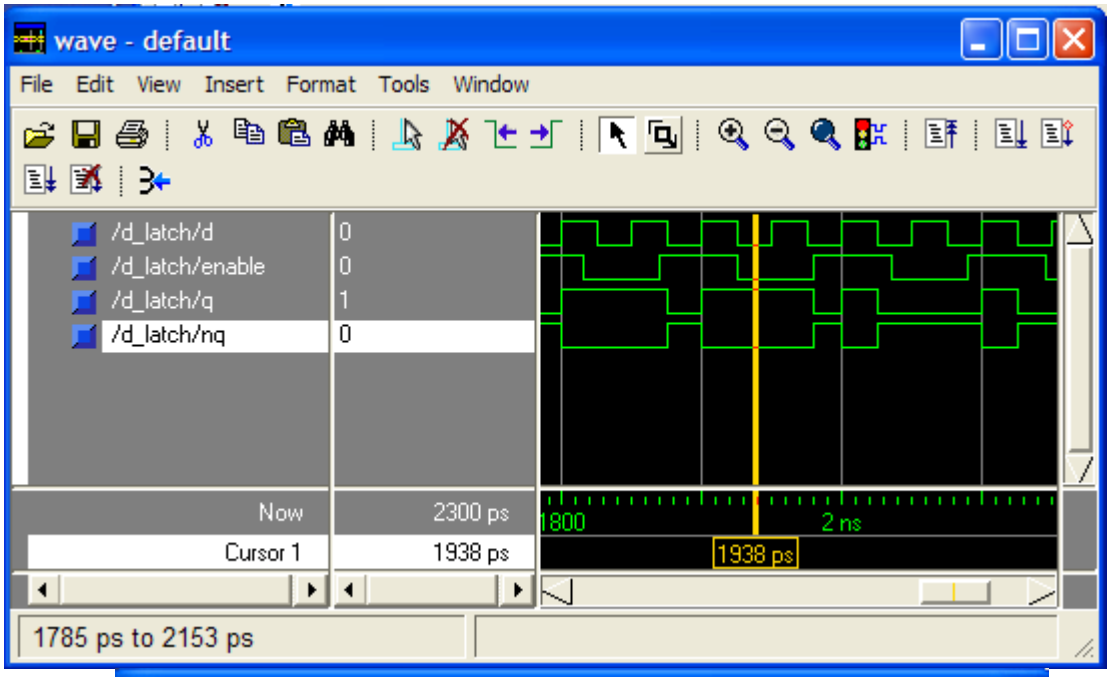
   You need to hand in the report next Friday.

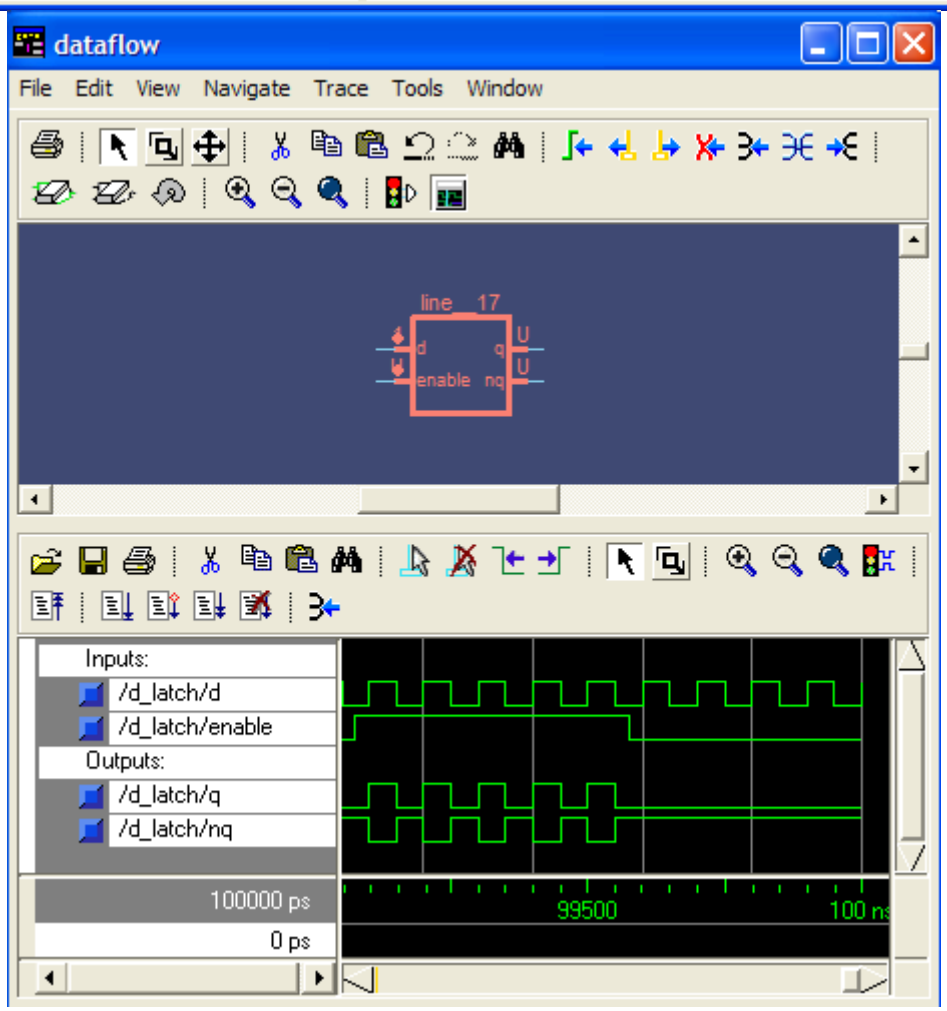
1. D-Latch: Dlatch.vhd



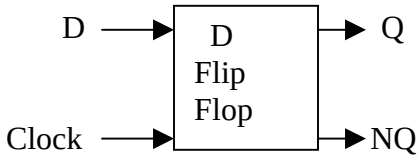
D	enable	Q	NQ
1/0	1	D	not D

```
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;
```


wave - default
File Edit View Insert Format Tools Window
/d_latch/d
/d_latch/enable
/d_latch/q
/d_latch/nq
0
0
1
0
Now
2300 ps
Cursor 1
1938 ps
1800
2 ns
1938 ps
1785 ps to 2153 ps

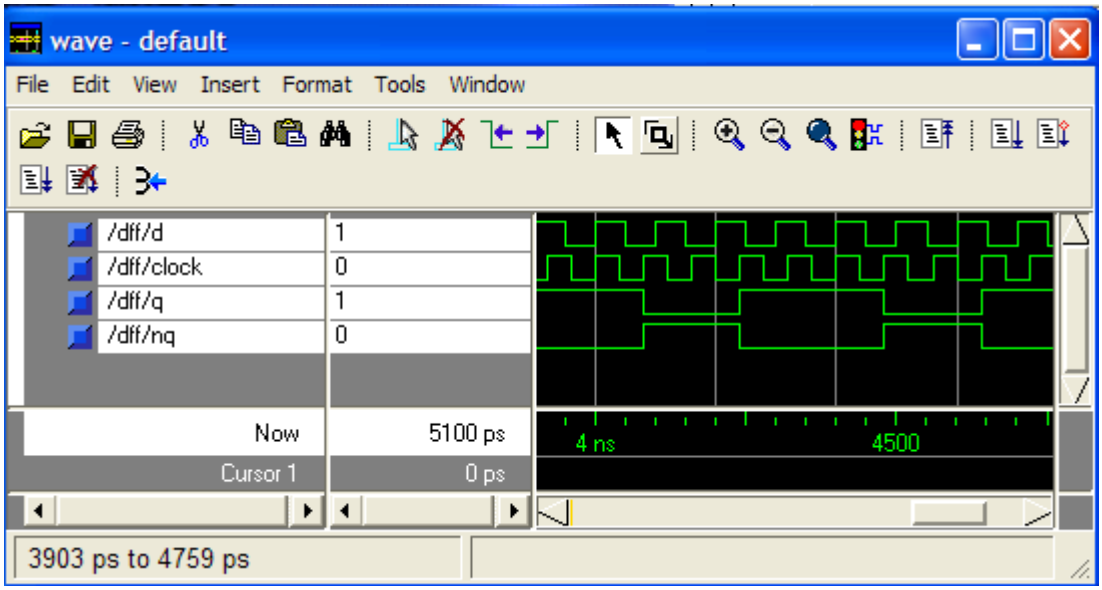

dataflow
File Edit View Navigate Trace Tools Window
line__17
d
enable
q
nq
U
U
Inputs:
/d_latch/d
/d_latch/enable
Outputs:
/d_latch/q
/d_latch/nq
100000 ps
0 ps
99500
100 ns

2. D_flip flop: DFF.vhd

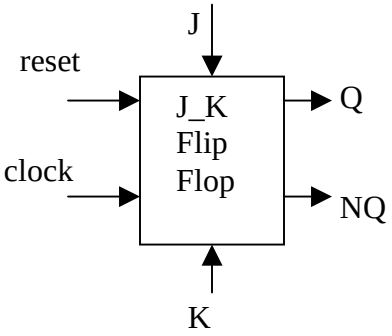


D	clock	Q	NQ
1/0	↑	D	not D

```
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;
```



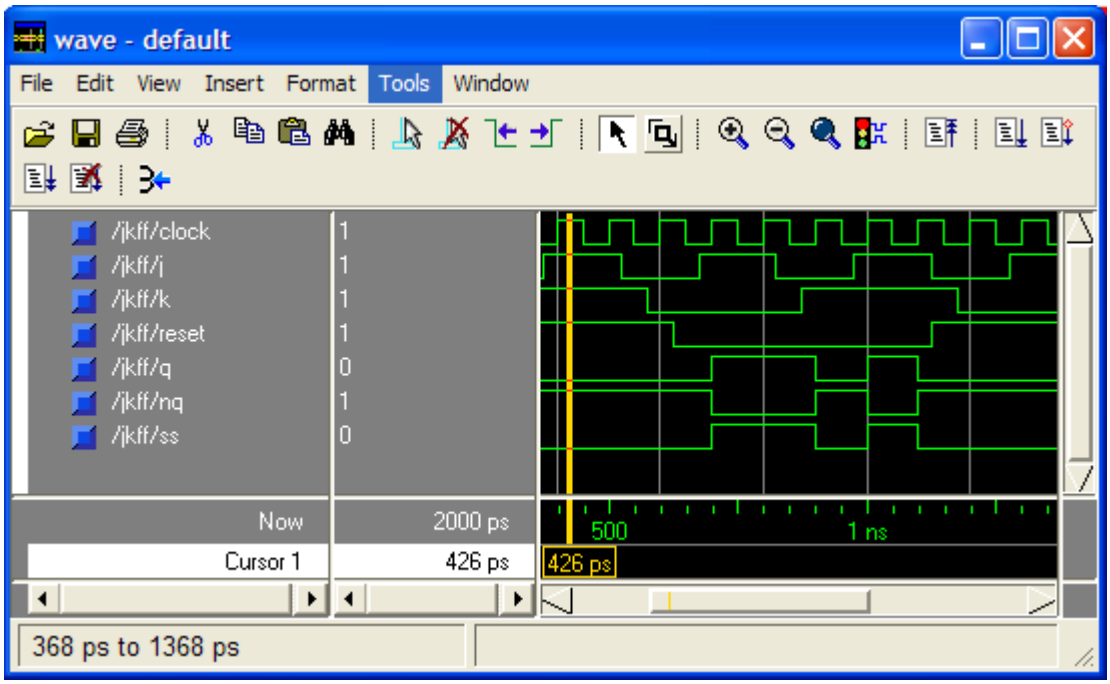
3. J_K flip flop (jkff.vhd)



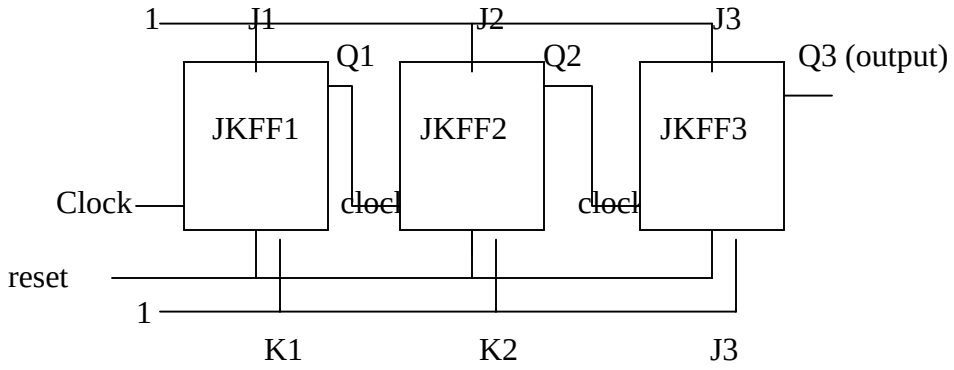
reset	clock	J	K	Q_n	NQ_n
1	X	X	X	0	1
0	↑	0	0	Q_n	NQ_n
		0	1	0	1
		1	0	1	0
		1	1	NQ_n	Q_n

```
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;
```

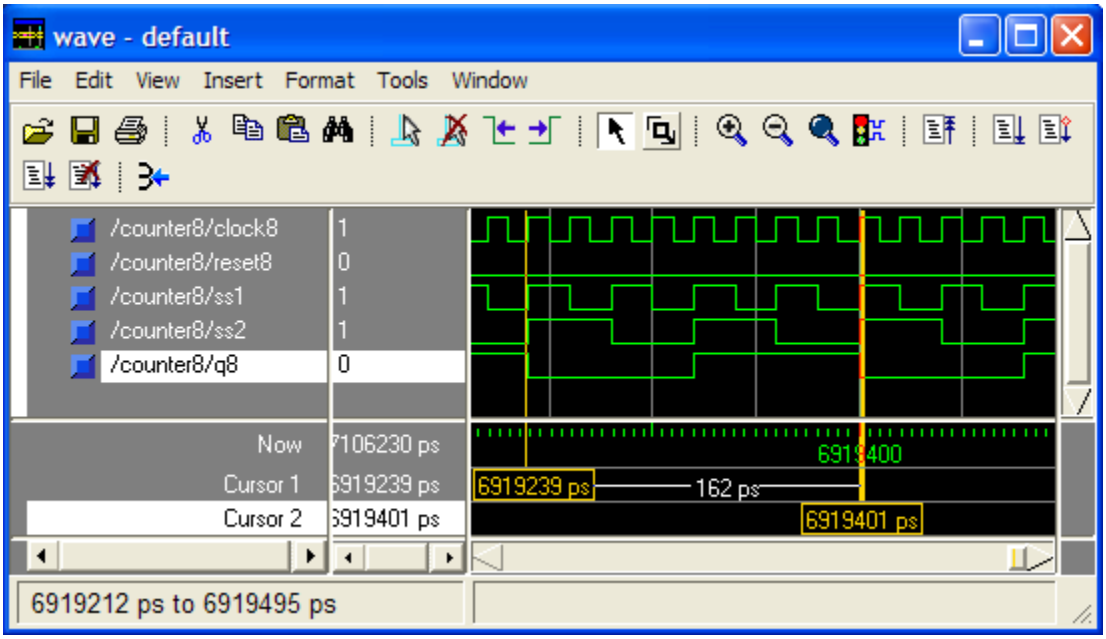
```
    begin
  process(clock, reset) is
  begin
       if (reset='1') then
                    ss <= '0';
       elsif (rising_edge(clock)) then
         if(j='1'and k='1')then
                   ss <= not ss;
         elsif(j='1'and k='0') then
                        ss <= '1';
         elsif(j='0'and k='1') then
                        ss <= '0';
         else
                         null;
       end if;
  end if;
 end process;
  Q <= ss;
  NQ <= not ss;
end behv;
```



4. Divide-by-8 counter: (using 3 J_KFF)




```
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;
```


wave - default
File Edit View Insert Format Tools Window
/counter8/clock8
/counter8/reset8
/counter8/ss1
/counter8/ss2
/counter8/q8
1
0
1
1
0
Now
Cursor 1
Cursor 2
7106230 ps
6919239 ps
6919401 ps
6919400
6919239 ps
162 ps
6919401 ps
6919212 ps to 6919495 ps