Dongyun Lee

ID: 112794190

PreLab8: Simple Sequential Circuit

ESE382-L01

Bench #4

File: X:\ESE382-Lab\Lab8\prelab8\prelab8\src\d_structural.vhd

```
1   ------------------------------------------------------------------------
    ----
2   --
3   -- Title       : d_structural
4   -- Design      : prelab8
5   -- Author      : Dongyun Lee
6   -- Company     : Stony Brook University
7   --
8   ------------------------------------------------------------------------
    ----
9   --
10  -- File        : X:\ESE382-Lab\Lab8\prelab8\prelab8\src\d_latch.vhd
11  -- Generated   : Mon Mar 25 21:59:07 2024
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ------------------------------------------------------------------------
    ----
16  --
17  -- Description :
18  --
19  ------------------------------------------------------------------------
    ----
20
21  ----- d latch -------
22  library ieee;
23  use ieee.std_logic_1164.all;
24
25
26  entity d_latch is
27      port(
28      d : in std_logic;       -- data input
29      le_bar : in std_logic;  -- latch enable input
30      ql  : out std_logic     --latch output
31      );
32  end d_latch;
33
34
35  architecture behavioral of d_latch is
36  begin
37      latch: process (d, le_bar)
38      begin
39          if le_bar = '0' then
40              ql <= d;    -- updates the output to the value of data(input)
41          end if;
42      end process;
43
44  end behavioral;
45
46  -------- flip flop ----
47  library ieee;
48  use ieee.std_logic_1164.all;
49
50
51
52  entity d_flip_flop is
53      port(
```

```
54          d : in std_logic;
55          clk : in std_logic;
56          qff : out std_logic
57      );
58  end d_flip_flop;
59
60
61  architecture behavioral of d_flip_flop is
62  begin
63      flipflop: process (clk)
64      begin
65          if clk'event and clk = '1' then
66              qff <= d;
67          end if;
68      end process;
69
70  end behavioral;
71
72  ------- top level entity -----
73  library ieee;
74  use ieee.std_logic_1164.all;
75  use work.all;
76
77  entity latch_vs_flip_flop is
78    port (
79      d   : in  std_logic;  -- data input
80      clk : in  std_logic;  -- clock input
81      ql  : out std_logic;  -- latch output
82      qff : out std_logic   -- flip-flop output
83    );
84  end latch_vs_flip_flop;
85
86  architecture structural of latch_vs_flip_flop is
87  begin
88      u1 : entity d_latch port map (d => d, le_bar => clk, ql => ql);
89      u2 : entity d_flip_flop port map (d => d, clk => clk, qff => qff);
90  end structural;
91
92
93
94
```

File: X:\ESE382-Lab\Lab8\prelab8\prelab8\src\dff_en.vhd

```
1   ------------------------------------------------------------------------
    ----
2   --
3   -- Title       : dff_en
4   -- Design      : prelab8
5   -- Author      : Dongyun Lee
6   -- Company     : Stony Brook University
7   --
8   ------------------------------------------------------------------------
    ----
9   --
10  -- File        : X:\ESE382-Lab\Lab8\prelab8\prelab8\src\dff_en.vhd
11  -- Generated   : Tue Mar 26 14:56:15 2024
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ------------------------------------------------------------------------
    ----
16  --
17  -- Description :
18  --
19  ------------------------------------------------------------------------
    ----
20
21  library ieee;
22  use ieee.std_logic_1164.all;
23
24  entity dff_en is
25    port (
26      d       : in  std_logic;  -- data input
27      clk     : in  std_logic;  -- clock input
28      en      : in  std_logic;  -- enable input
29      rst_bar : in  std_logic;  -- asynchronous reset
30      q       : out std_logic   -- output
31    );
32  end dff_en;
33
34  architecture behavioral of dff_en is
35  begin
36      flipflop : process (clk, rst_bar)
37      begin
38          if rst_bar = '0' then
39              q <= '0';
40          elsif rising_edge(clk) then
41              if en = '1' then
42                  q <= d;
43              end if;
44
45          end if;
46      end process;
47  end behavioral;
48
49
50
```

File: X:/ESE382-Lab/Lab8/prelab8/prelab8/src/right_shift_reg.vhd

```vhdl
1   ----------------------------------------------------------------------------
    -------
2   --
3   -- Title       : right_shift_reg
4   -- Design      : prelab8
5   -- Author      : Dongyun Lee
6   -- Company     : Stony Brook University
7   --
8   ----------------------------------------------------------------------------
    -------
9   --
10  -- File        :
    X:\ESE382-Lab\Lab8\prelab8\prelab8\src\right_shift_reg.vhd
11  -- Generated   : Tue Mar 26 15:31:56 2024
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ----------------------------------------------------------------------------
    -------
16  --
17  -- Description :
18  --
19  ----------------------------------------------------------------------------
    -------
20
21  library ieee;
22  use ieee.std_logic_1164.all;
23
24  entity right_shift_reg is
25    port (
26      d          : in std_logic_vector(3 downto 0);  -- parallel input
    data
27      load       : in std_logic;                     -- synchronous load
    parallel input
28      en_shift   : in std_logic;                     -- enable shift if
    load is unasserted
29      clk        : in std_logic;                     -- clk
30      rst_bar    : in std_logic;                     -- asynchronous reset
31      serial_out : out std_logic                     -- serial output
32    );
33  end right_shift_reg;
34
35
36  architecture behavior of right_shift_reg is
37  signal memory : std_logic_vector(3 downto 0);
38  begin
39      reg : process (clk, rst_bar)
40      begin
41          if rst_bar = '0' then
42              memory <= "0000";
43          elsif rising_edge(clk) then
44              if load = '1' then
45                  memory <= d;
46              elsif en_shift = '1' then
47                  memory <= memory(3 downto 1) & '0';
48              end if;
49          end if;
50      end process;
51
52      serial_out <= memory(0);
53
```

```
54  end behavior;
55
```

```
1   ----------------------------------------------------------------------
    -------
2   --
3   -- Title       : \3_to_8_decoder\
4   -- Design      : prelab8
5   -- Author      : Dongyun Lee
6   -- Company     : Stony Brook University
7   --
8   ----------------------------------------------------------------------
    -------
9   --
10  -- File        :
    X:\ESE382-Lab\Lab8\prelab8\prelab8\src\3_to_8_decoder.vhd
11  -- Generated   : Tue Mar 26 16:29:37 2024
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ----------------------------------------------------------------------
    -------
16  --
17  -- Description :
18  --
19  ----------------------------------------------------------------------
    -------
20  -------------input latch ------------------------------------
21  library ieee;
22  use ieee.std_logic_1164.all;
23
24  entity input_latch is
25      port(
26      a : in std_logic_vector(2 downto 0);
27      le_bar : in std_logic;
28      a_lat : out std_logic_vector(2 downto 0)
29      );
30  end input_latch;
31
32  architecture behavioral of input_latch is
33  begin
34      latch: process (a, le_bar)
35      begin
36          if le_bar = '0' then
37              a_lat <= a; -- updates the output to the value of
    data(input)
38          end if;
39      end process;
40  end behavioral;
41
42  -------------3 to 8 decoder ---------------------------------
43  library ieee;
44  use ieee.std_logic_1164.all;
45  use ieee.numeric_std.all;
46
47  entity decoder_3to8 is
48      port(
49      a_lat : in std_logic_vector(2 downto 0);
50      g : in std_logic;
51      y : out std_logic_vector(0 to 7)
52      );
53  end decoder_3to8;
54
55  architecture cond of decoder_3to8 is
```

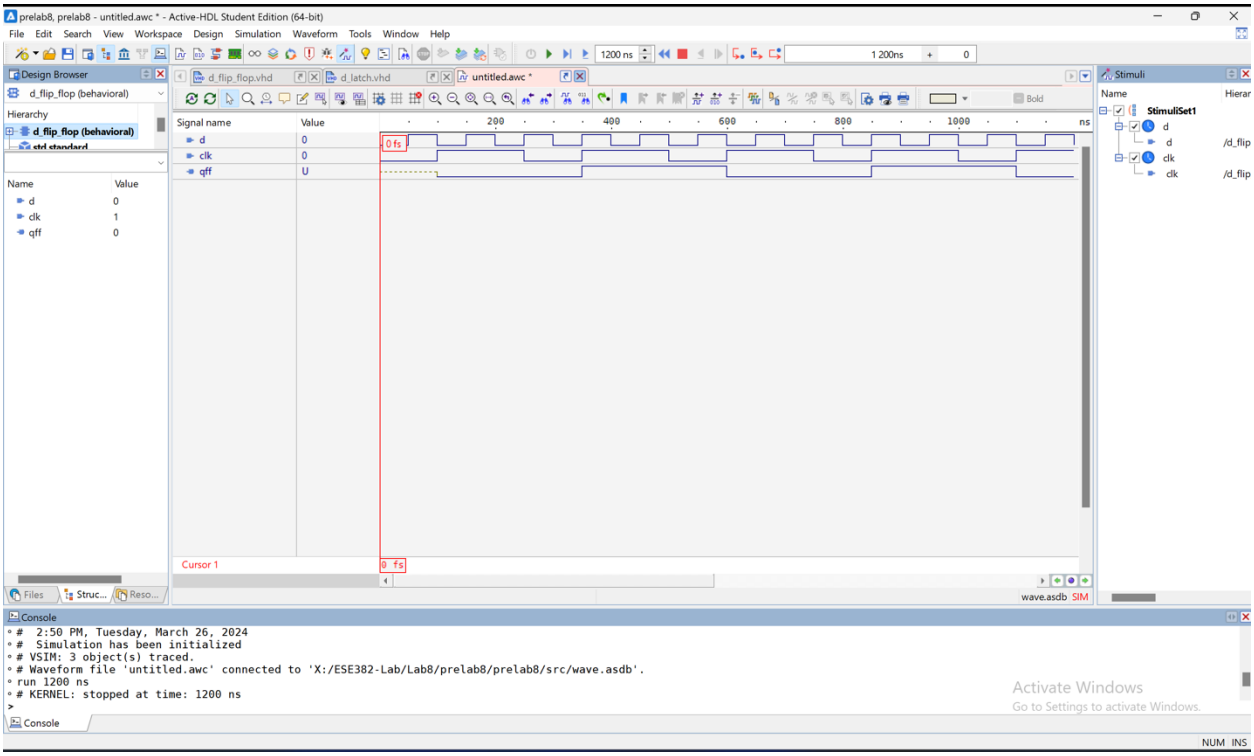
```
56  begin
57      decoder : process (a_lat, g)
58      begin
59          y <= "00000000";
60          if g = '1' then
61              for i in 0 to 7 loop
62                  if unsigned(a_lat) = to_unsigned(i,3) then
63                      y(i) <= '1';
64                  end if;
65                  end loop;
66          end if;
67      end process;
68  end cond;
69
70
71  --------------top level entity latched 3 to 8 decoder -----
72  library ieee;
73  use ieee.std_logic_1164.all;
74  use ieee.numeric_std.all;
75  use work.all;
76
77  entity latched_3to8_decoder is
78      port(
79      a   : in std_logic_vector(2 downto 0);
80      le_bar, e1_bar, e2 : in std_logic;
81      y   : out std_logic_vector(0 to 7)
82      );
83  end latched_3to8_decoder;
84
85  architecture structural of latched_3to8_decoder is
86  signal a_lat : std_logic_vector(2 downto 0);
87  signal g     : std_logic;
88  begin
89      u0 : entity input_latch port map (a => a, le_bar => le_bar, a_lat =>
    a_lat);
90      u1 : entity decoder_3to8 port map (a_lat => a_lat, g => g, y => y);

91      u2 : g <= '1' when (e1_bar = '0') and (e2 = '1') else '0';
92
93  end structural;
94
```

```
1   ----------------------------------------------------------------------------
    ----
2   --
3   -- Title       : latched_3to8_decoder_tb
4   -- Design      : prelab8
5   -- Author      : Dongyun Lee
6   -- Company     : Stony Brook University
7   --
8   ----------------------------------------------------------------------------
    ----
9   --
10  -- File        : Z:\Desktop\SBU 2024 Spring\ESE
    382\lab-backup\Lab8\prelab8\prelab8\src\latched_3to8_decoder_tb.vhd
11  -- Generated   : Sat Mar 30 16:39:02 2024
12  -- From        : interface description file
13  -- By          : Itf2Vhdl ver. 1.22
14  --
15  ----------------------------------------------------------------------------
    ----
16  --
17  -- Description : testbench for latched 3 to 8 decoder
18  --
19  ----------------------------------------------------------------------------
    ----
20
21  library ieee;
22  use ieee.std_logic_1164.all;
23  use ieee.numeric_std.all;
24  use work.all;
25
26  entity latched_3to8_decoder_tb is
27  end latched_3to8_decoder_tb;
28
29
30  architecture latched_3to8_decoder_tb of latched_3to8_decoder_tb is
31     signal a      : std_logic_vector(2 downto 0);
32     signal le_bar : std_logic;
33     signal e1_bar : std_logic;
34     signal e2     : std_logic;
35     signal y      : std_logic_vector(0 to 7);
36
37  type test_vector is record
38      le_bar : std_logic;
39      e1_bar : std_logic;
40      e2     : std_logic;
41      a      : std_logic_vector(2 downto 0);
42      y      : std_logic_vector(0 to 7);
43  end record;
44
45  type test_vectors is array (natural range <>) of test_vector;
46
47  constant test_table : test_vectors := (
48    -- el_   e1_  e2     a       y
49      ( '-', '1', '-', "---", "00000000"), -- When E1 is low, all outputs are
    low
50      ( '-', '-', '0', "---", "00000000"), -- When E2 is low, all outputs are
    low
```

```
51      -- Latch enabled and E1 is low, decoder is active
52      ( '0', '0', '1', "000", "10000000"), -- Input A is 0, Y0 is high
53      ( '0', '0', '1', "001", "01000000"), -- Input A is 1, Y1 is high
54      ( '0', '0', '1', "010", "00100000"), -- Input A is 2, Y2 is high
55      ( '0', '0', '1', "011", "00010000"), -- Input A is 3, Y3 is high
56          ( '1', '0', '1', "---", "--------"), -- When LE is high, outputs
    are stable
57      ( '0', '0', '1', "100", "00001000"), -- Input A is 4, Y4 is high
58      ( '0', '0', '1', "101", "00000100"), -- Input A is 5, Y5 is high
59      ( '0', '0', '1', "110", "00000010"), -- Input A is 6, Y6 is high
60      ( '0', '0', '1', "111", "00000001") -- Input A is 7, Y7 is high
61  );
62
63  begin
64      UUT : entity latched_3to8_decoder port map (
65            a => a,
66            le_bar => le_bar,
67            e1_bar => e1_bar,
68            e2 => e2,
69            y => y
70            );
71      tb : process
72      variable memory : std_logic_vector(7 downto 0);
73      begin
74
75          for i in test_table'range loop
76              a <= test_table(i).a;
77              le_bar <= test_table(i).le_bar;
78              e1_bar <= test_table(i).e1_bar;
79              e2 <= test_table(i).e2;
80              wait for 20ns;
81              if le_bar = '1' and e1_bar = '0' and e2 = '1' then
82                  assert y = memory;
83                  report "Error at le_bar = '1' and e1_bar = '0' and e2 =
    '1'. Output should be stable"
84                  severity error;
85              else
86                  memory := test_table(i).y;
87                  assert y = test_table(i).y
88                  report "Error at input a, le_bar, e1_bar, e2 : " &
    to_string(a) & to_string(le_bar) & to_string(e1_bar) & to_string(e2)
89                  severity error;
90              end if;
91
92
93          end loop;
94          std.env.finish;
95      end process;
96
97
98  end latched_3to8_decoder_tb;
99
```

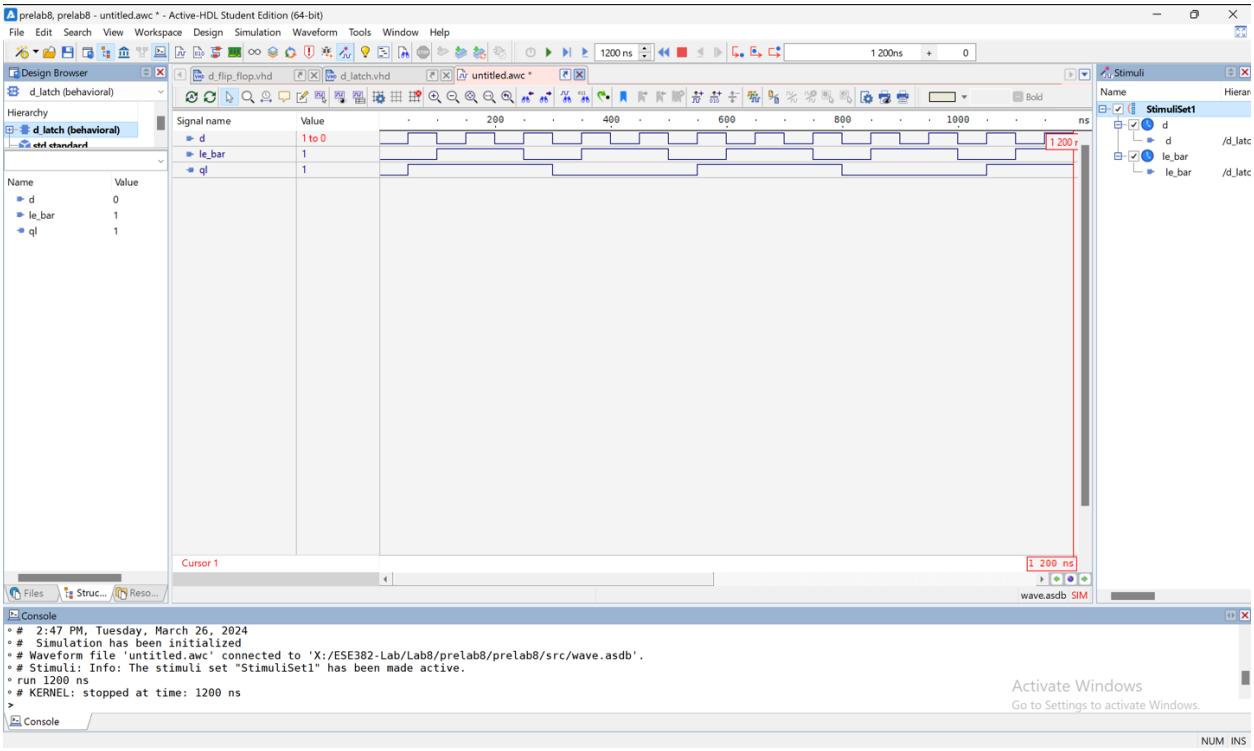
Design task 1

D-flip-flop waveform



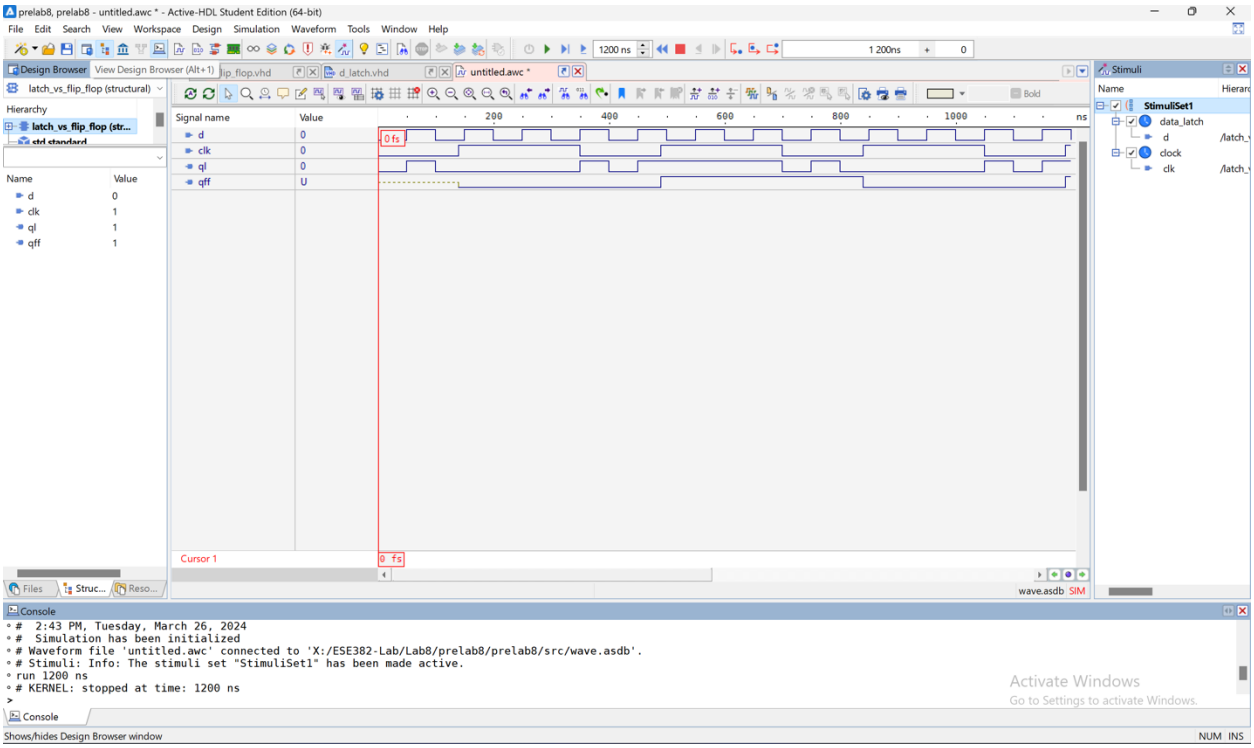
Qff copy d at rising clock edge.

D-latch waveform



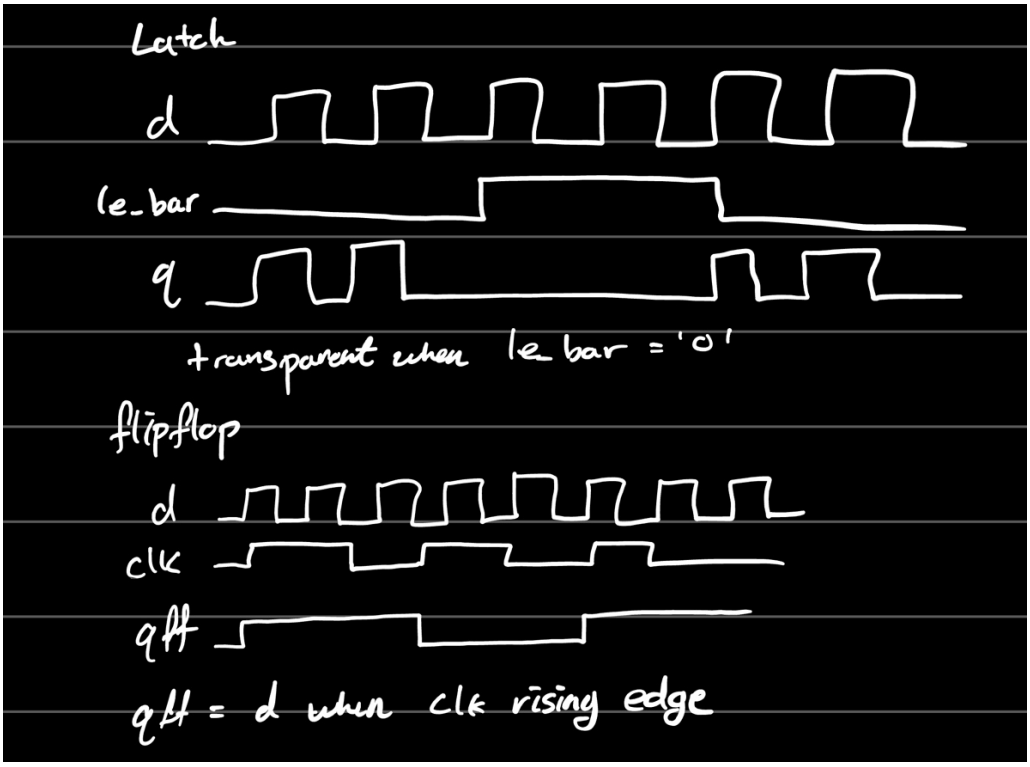
Ql copy d when le_bar = ‘0’.

Latch vs. flipflop waveform



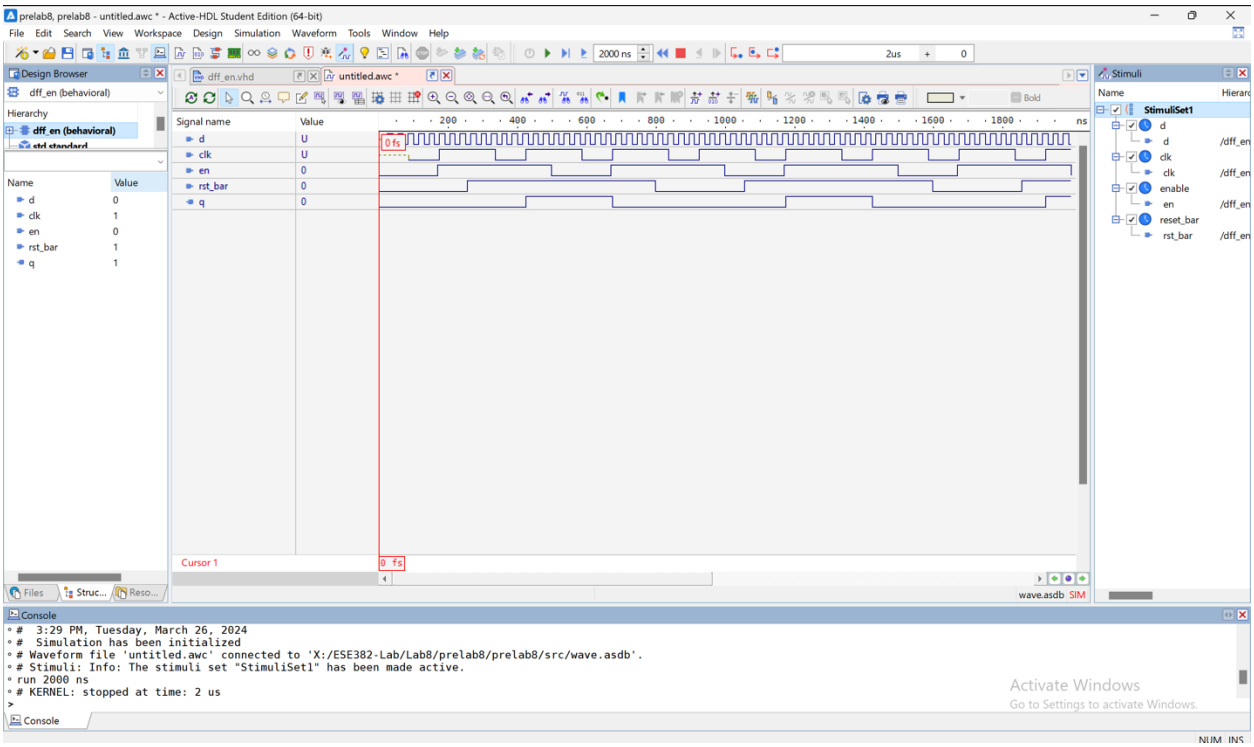
Qff copy d when rising clock edge and ql copy d when clock = ‘0’.

Sketch of the verification waveform



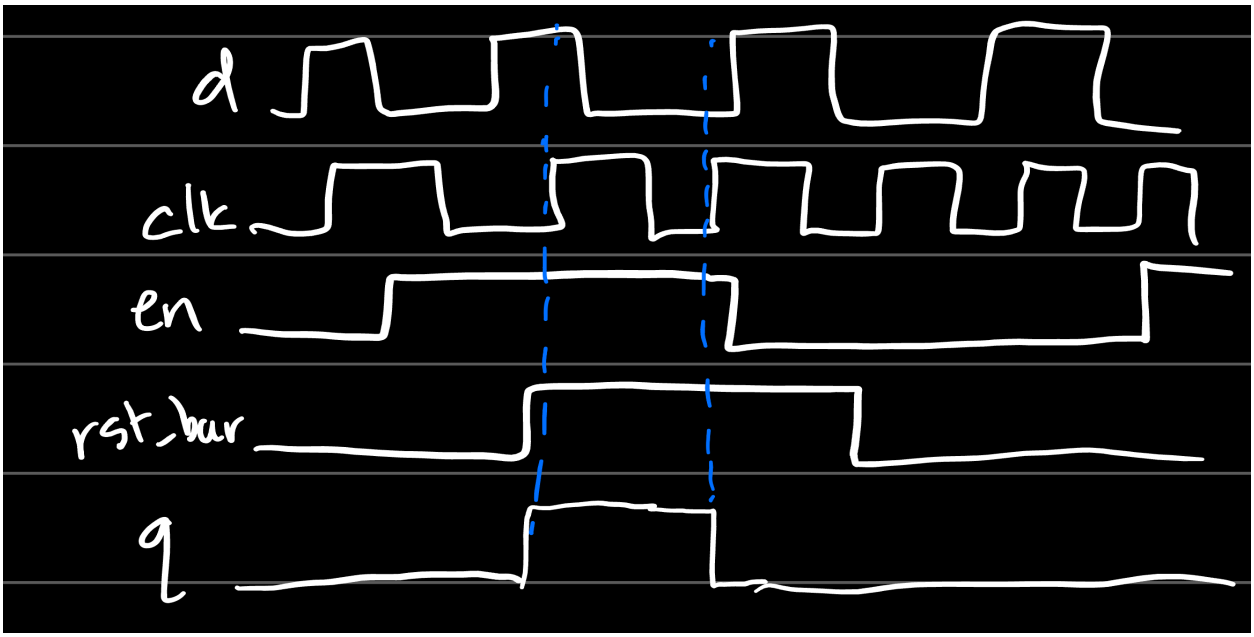
Design task 2

waveform



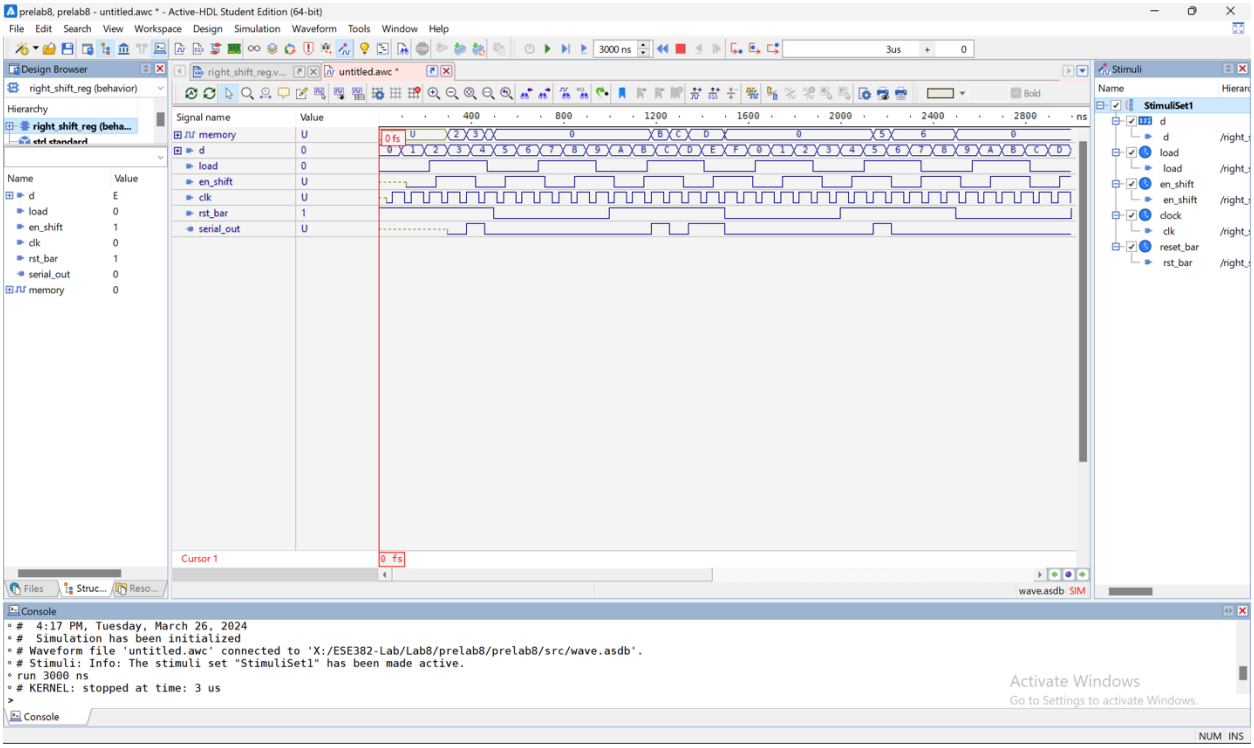
When en = ‘1’ and rst_bar = ‘1’ q copy d when rising clock edge.

Sketch of the verification waveform



Design task 3

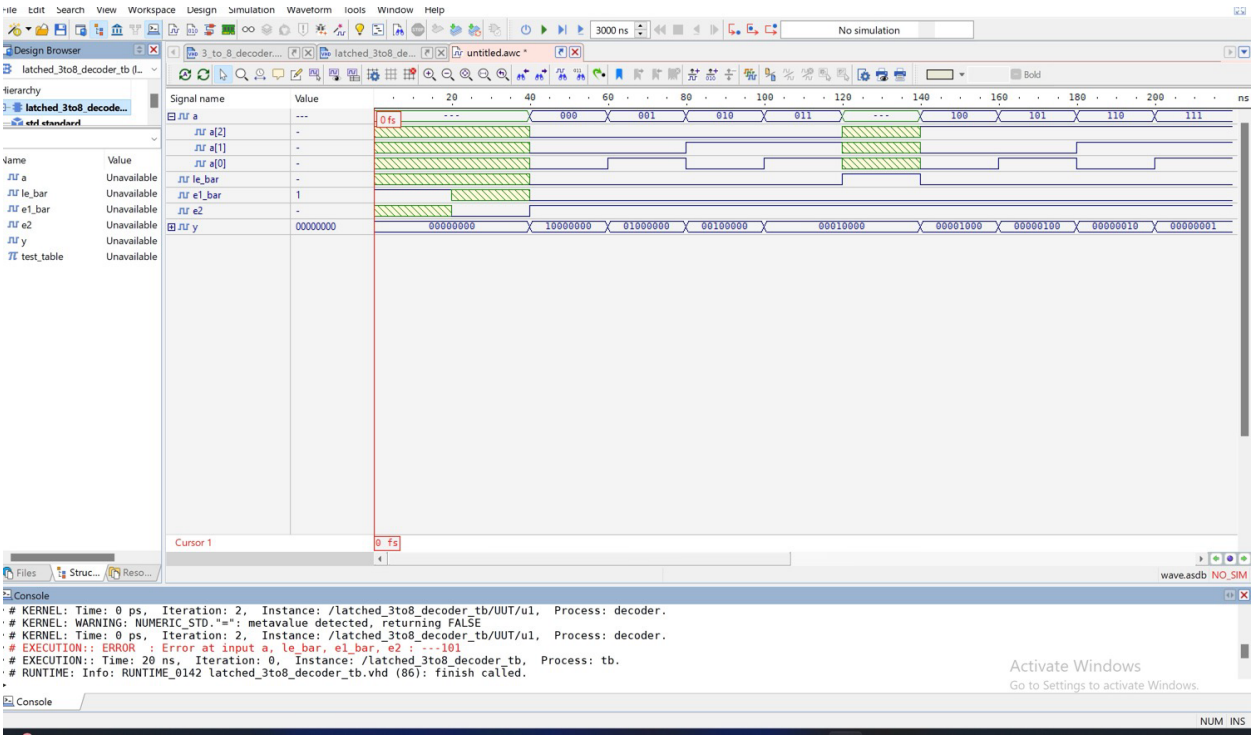
Waveform



Rst_bar is the only asynchronous input. When rst_bar = ‘0’ all the bits of the shift register are 0s. The output loads four bits of input on a rising clock edge when load = ‘1’.

Design task4

Waveform



When either e1_bar = ‘1’ and e2 = ‘0’ , the output must be all 0s. When le_bar = ‘0’ e1_bar = ‘0’ and e2 = ‘1’ then the output y is as expected on the truth table. When le_bar = ‘1’, the output stays the same (as it is shown for y = “00010000”.