Homework Assignment 1
(Due Jan. 29th at the beginning of the class)

***** Submission policy: Please zip your source code and waveform screenshots into a single file and send it to daehyun@eecs.wsu.edu. The file name should be *firstname_lastname*.zip (or .tar.gz or .tar …)

(1) **[VHDL, 10 points]** Make a VHDL code for a 4:1 mux and test it with the following spec.

- Input ports: i0, i1, i2, i3, s1, s0
- Output port: z
- Function: z=i0 when (s1s0=00), z=i1 when (s1s0=01), z=i2 when (s1s0=10), z=i3 when (s1s0=11)
- Test input vectors (i3,i2,i1,i0,s1,s0)
  - 000000 → 000100 → 000101 → 000110 → 000111 → 000100 → 000000 → 001000 → 001001 → 001010 → 001011 → 001000 → 000000 → 010000 → 010001 → 010010 → 010011 → 010000 → 000000 → 100000 → 100001 → 100010 → 100011 → 100000
- [**Submit**] Source code + test waveform (inputs + outputs)

*** Source code for a 4:1 MUX**

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

ENTITY vMUX4 IS
  PORT ( i3, i2, i1, i0, s1, s0 : IN std_logic;
         z : OUT std_logic );
END vMUX4;

ARCHITECTURE vMUX4_arch OF vMUX4 IS
BEGIN
  PROCESS (i0, i1, i2, i3, s0, s1)
  BEGIN
    IF (s1 = '0') AND (s0 = '0') THEN
      z <= i0;
    ELSIF (s1 = '0') AND (s0 = '1') THEN
```

```
      z <= i1;
    ELSIF (s1 = '1') AND (s0 = '0') THEN
      z <= i2;
    ELSIF (s1 = '1') AND (s0 = '1') THEN
      z <= i3;
    ELSE  -- exception handling
      z <= 'X';
    END IF;
  END PROCESS;
END vMUX4_arch;
```

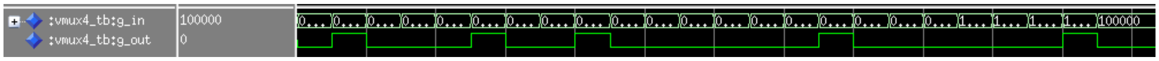
* Testbench

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

ENTITY vMUX4_tb IS
END vMUX4_tb;

ARCHITECTURE vMUX4_tb_arch OF vMUX4_tb IS
  COMPONENT vMUX4 PORT ( i3, i2, i1, i0, s1, s0 : IN std_logic; z : OUT std_logic ); END COMPONENT;
  SIGNAL g_in : std_logic_vector (5 downto 0);
  SIGNAL g_out : std_logic;
BEGIN
  u1 : vMUX4 PORT MAP (g_in(5), g_in(4), g_in(3), g_in(2), g_in(1), g_in(0), g_out);

  PROCESS
  BEGIN
    g_in <= "000000";  -- z: 0
    WAIT FOR 0.1 ns;
    g_in <= "000100";  -- z: 1
    WAIT FOR 0.1 ns;
    g_in <= "000101";  -- z: 0
    WAIT FOR 0.1 ns;
    g_in <= "000110";  -- z: 0
    WAIT FOR 0.1 ns;
```

```
g_in <= "000111";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "000100";  -- z: 1
WAIT FOR 0.1 ns;
g_in <= "000000";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "001000";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "001001";  -- z: 1
WAIT FOR 0.1 ns;
g_in <= "001010";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "001011";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "001000";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "000000";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "010000";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "010001";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "010010";  -- z: 1
WAIT FOR 0.1 ns;
g_in <= "010011";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "010000";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "000000";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "100000";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "100001";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "100010";  -- z: 0
WAIT FOR 0.1 ns;
g_in <= "100011";  -- z: 1
```

```
    WAIT FOR 0.1 ns;
    g_in <= "100000";  -- z: 0
    WAIT FOR 100 ns;
  END PROCESS;
END vMUX4_tb_arch;
```

* Waveform



(2) **[VHDL, 10 points]** Make a VHDL code for a D-latch and test it with the following spec.

- Input ports: D, Clk
- Output ports: Q, QN
- Function: If Clk=0, Q=Q_{prev}, QN=QN_{prev}. If Clk=1, Q=D, QN=$\overline{D}$.
- Test input vectors (D, Clk)
    - o 00 → 01 → 00 → 10 → 00 → 01 → 11 → 01 → 11 → 10 → 00 → 10 → 00 → 01 → 00 → 10 → 00
- [**Submit**] Source code + test waveform (inputs + outputs)

* Source code for a 4:1 MUX

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

ENTITY vDLatch IS
  PORT ( D, Clk : IN std_logic;
         Q, QN : OUT std_logic );
END vDLatch;

ARCHITECTURE vDLatch_arch OF vDLatch IS
BEGIN
  PROCESS (D, Clk)
  BEGIN
    IF (Clk = '1') THEN
```

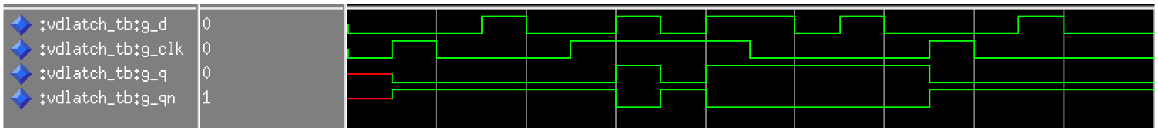
```
      Q <= D;
      QN <= NOT D;
    END IF;
  END PROCESS;
END vDLatch_arch;
```

* Testbench

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

ENTITY vDLatch_tb IS
END vDLatch_tb;

ARCHITECTURE vDLatch_tb_arch OF vDLatch_tb IS
  COMPONENT vDLatch PORT ( D, Clk : IN std_logic; Q, QN : OUT std_logic ); END COMPONENT;
  SIGNAL g_D, g_Clk, g_Q, g_QN : std_logic;
BEGIN
  u1 : vDLatch PORT MAP (g_D, g_Clk, g_Q, g_QN);

  PROCESS
  BEGIN
    g_D <= '0';
    g_Clk <= '0';  -- Q: U, QN: U
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '1';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '0';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '1';
    g_Clk <= '0';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '0';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '1';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '1';
    g_Clk <= '1';  -- Q: 1, QN: 0
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '1';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '1';
    g_Clk <= '1';  -- Q: 1, QN: 0
    WAIT FOR 0.1 ns;
    g_D <= '1';
    g_Clk <= '0';  -- Q: 1, QN: 0
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '0';  -- Q: 1, QN: 0
    WAIT FOR 0.1 ns;
    g_D <= '1';
    g_Clk <= '0';  -- Q: 1, QN: 0
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '0';  -- Q: 1, QN: 0
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '1';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '0';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '1';
    g_Clk <= '0';  -- Q: 0, QN: 1
    WAIT FOR 0.1 ns;
    g_D <= '0';
    g_Clk <= '0';  -- Q: 0, QN: 1
    WAIT FOR 10 ns;
  END PROCESS;
```

```
END vDLatch_tb_arch;
```

* Waveform



```
:vdlatch_tb:g_d    0
:vdlatch_tb:g_clk  0
:vdlatch_tb:g_q    0
:vdlatch_tb:g_qn   1
```