Homework Assignment 2
(Due Feb. 1st 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]** Design a positive-edge-triggered D-FF with synchronous reset and set. Use the following spec:

- Input ports: D, R, S, Clk
- Output port: Q
- Function:
  - Q = 0 if reset = 1 (regardless of D and set) when Clk ↑.
  - Q = 1 if reset = 0 and set = 1 (regardless of D) when Clk ↑.
  - Q = D if reset = 0 and set = 0 when Clk ↑.
- Create your own test input vectors to test all the following 16 combinations:
  - reset=0, set=0, D=0, Clk ↑ (when Q=0)
  - reset=0, set=0, D=0, Clk ↑ (when Q=1)
  - reset=0, set=0, D=1, Clk ↑ (when Q=0)
  - reset=0, set=0, D=1, Clk ↑ (when Q=1)
  - **…**
- [**Submit**] Source code + test waveform (inputs + outputs)

*** Source code (PDFF with synchronous RS)**

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

ENTITY vPDFF_SRS IS
  PORT ( D, R, S, Clk : IN std_logic;
       Q : OUT std_logic );
END vPDFF_SRS;

ARCHITECTURE vPDFF_SRS_arch OF vPDFF_SRS IS
BEGIN
  PROCESS (Clk)
```

```
 BEGIN
  IF rising_edge(Clk) THEN
   IF (R = '1') THEN
    Q <= '0';
   ELSIF (S = '1') THEN
    Q <= '1';
   ELSE
    Q <= D;
   END IF;
  END IF;
 END PROCESS;
END vPDFF_SRS_arch;
```

*** Source code (testbench)**

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

ENTITY vPDFF_SRS_tb IS
END vPDFF_SRS_tb;

ARCHITECTURE vPDFF_SRS_tb_arch OF vPDFF_SRS_tb IS
 COMPONENT vPDFF_SRS PORT ( D, R, S, Clk : IN std_logic; Q : OUT std_logic ); END
COMPONENT;
 SIGNAL g_in : std_logic_vector (3 downto 0);
 SIGNAL g_out : std_logic;
BEGIN
 u1 : vPDFF_SRS PORT MAP (g_in(1), g_in(3), g_in(2), g_in(0), g_out);

 PROCESS
 BEGIN
  g_in <= "0000";  -- R/S/D/Clk
  WAIT FOR 0.1 ns;
  g_in <= "0001";  -- Q = 0
  WAIT FOR 0.1 ns;
  g_in <= "0000";
  WAIT FOR 0.1 ns;
```

```
g_in <= "0001";  -- case 0: RSDQ=0000. Q: 0->0
WAIT FOR 0.1 ns;
g_in <= "0010";  -- this input transition is safe
WAIT FOR 0.1 ns;
g_in <= "0011";  -- case 2: RSDQ=0010. Q: 0->1
WAIT FOR 0.1 ns;
g_in <= "0000";
WAIT FOR 0.1 ns;
g_in <= "0001";  -- case 1: RSDQ=0001. Q: 1->0
WAIT FOR 0.1 ns;
g_in <= "0100";
WAIT FOR 0.1 ns;
g_in <= "0101";  -- case 4: RSDQ=0100. Q: 0->1
WAIT FOR 0.1 ns;
g_in <= "0010";
WAIT FOR 0.1 ns;
g_in <= "0011";  -- case 3: RSDQ=0011. Q: 1->1
WAIT FOR 0.1 ns;
g_in <= "0100";
WAIT FOR 0.1 ns;
g_in <= "0101";  -- case 5: RSDQ=0101. Q: 1->1
WAIT FOR 0.1 ns;
g_in <= "1000";
WAIT FOR 0.1 ns;
g_in <= "1001";  -- case 9: RSDQ=1001. Q: 1->0
WAIT FOR 0.1 ns;
g_in <= "0110";
WAIT FOR 0.1 ns;
g_in <= "0111";  -- case 6: RSDQ=0110. Q: 0->1
WAIT FOR 0.1 ns;
g_in <= "0110";
WAIT FOR 0.1 ns;
g_in <= "0111";  -- case 7: RSDQ=0111. Q: 1->1
WAIT FOR 0.1 ns;
g_in <= "1000";
WAIT FOR 0.1 ns;
g_in <= "1001";  -- Q: 0
```

```
WAIT FOR 0.1 ns;
g_in <= "1000";
WAIT FOR 0.1 ns;
g_in <= "1001";  -- case 8: RSDQ=1000. Q: 0->0
WAIT FOR 0.1 ns;
g_in <= "1010";
WAIT FOR 0.1 ns;
g_in <= "1011";  -- case 10: RSDQ=1010. Q: 0->0
WAIT FOR 0.1 ns;
g_in <= "0100";
WAIT FOR 0.1 ns;
g_in <= "0101";  -- Q = 1
WAIT FOR 0.1 ns;
g_in <= "1010";
WAIT FOR 0.1 ns;
g_in <= "1011";  -- case 11: RSDQ=1011. Q: 1->0
WAIT FOR 0.1 ns;
g_in <= "1100";
WAIT FOR 0.1 ns;
g_in <= "1101";  -- case 12: RSDQ=1100. Q: 0->0
WAIT FOR 0.1 ns;
g_in <= "0100";
WAIT FOR 0.1 ns;
g_in <= "0101";  -- Q = 1
WAIT FOR 0.1 ns;
g_in <= "1100";
WAIT FOR 0.1 ns;
g_in <= "1101";  -- case 13: RSDQ=1101. Q: 1->0
WAIT FOR 0.1 ns;
g_in <= "1110";
WAIT FOR 0.1 ns;
g_in <= "1111";  -- case 14: RSDQ=1110. Q: 0->0
WAIT FOR 0.1 ns;
g_in <= "0100";
WAIT FOR 0.1 ns;
g_in <= "0101";  -- Q = 1
WAIT FOR 0.1 ns;
```

```
   g_in <= "1110";
   WAIT FOR 0.1 ns;
   g_in <= "1111";  -- case 15: RSDQ=1111. Q: 1->0
   WAIT FOR 100 ns;
  END PROCESS;
END vPDFF_SRS_tb_arch;
```

*** Waveform**

(2) **[VHDL, 10 points]** Design a positive-edge-triggered D-FF with asynchronous reset and set. Use the following spec:

- Input ports: D, R, S, Clk
- Output ports: Q
- Function:
  - Q = 0 if reset = 1 (regardless of D, set, and Clk).
  - Q = 1 if reset = 0 and set is 1 (regardless of D and Clk).
  - Q = D if reset = 0 and set is 0 when Clk ↑.
  - Test your design using the same input vectors you created above.
- [**Submit**] Source code + test waveform (inputs + outputs)

*** Source code (PDFF with asynchronous RS)**

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

ENTITY vPDFF_ARS IS
  PORT ( D, R, S, Clk : IN std_logic;
         Q : OUT std_logic );
END vPDFF_ARS;

ARCHITECTURE vPDFF_ARS_arch OF vPDFF_ARS IS
BEGIN
  PROCESS (R, S, Clk)
```

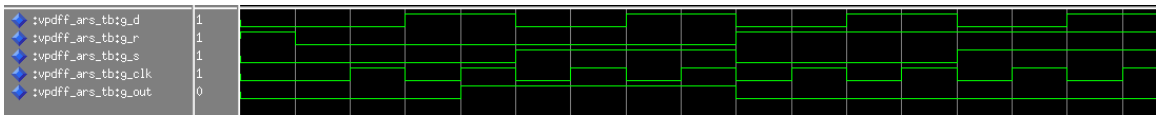
```
 BEGIN
  IF (R = '1') THEN
   Q <= '0';
  ELSIF (S = '1') THEN
   Q <= '1';
  ELSIF rising_edge(Clk) THEN
   Q <= D;
  END IF;
 END PROCESS;
END vPDFF_ARS_arch;
```

*** Source code (testbench)**

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

ENTITY vPDFF_ARS_tb IS
END vPDFF_ARS_tb;

ARCHITECTURE vPDFF_ARS_tb_arch OF vPDFF_ARS_tb IS
 COMPONENT vPDFF_ARS PORT ( D, R, S, Clk : IN std_logic; Q : OUT std_logic ); END
COMPONENT;
 SIGNAL g_D, g_R, g_S, g_Clk : std_logic;
 SIGNAL g_out : std_logic;
BEGIN
 u1 : vPDFF_ARS PORT MAP (g_D, g_R, g_S, g_Clk, g_out);

 PROCESS
 BEGIN
  g_D <= '0';
  g_R <= '1';
  g_S <= '0';
  g_Clk <= '0';
  WAIT FOR 0.1 ns;

  FOR i1 IN 0 TO 1 LOOP
   IF (i1 = 0) THEN
```

```
      g_R <= '0';
    ELSE
      g_R <= '1';
    END IF;

    FOR i2 IN 0 TO 1 LOOP
      IF (i2 = 0) THEN
        g_S <= '0';
      ELSE
        g_S <= '1';
      END IF;

      FOR i3 IN 0 TO 1 LOOP
        IF (i3 = 0) THEN
          g_D <= '0';
        ELSE
          g_D <= '1';
        END IF;

        FOR i4 IN 0 TO 1 LOOP
          IF (i4 = 0) THEN
            g_Clk <= '0';
          ELSE
            g_Clk <= '1';
          END IF;

          WAIT FOR 0.1 ns;
        END LOOP;
      END LOOP;
    END LOOP;
  END LOOP;

  WAIT FOR 100 ns;
 END PROCESS;
END vPDFF_ARS_tb_arch;
```

*** Waveform**


:vpdff_ars_tb:g_d
:vpdff_ars_tb:g_r
:vpdff_ars_tb:g_s
:vpdff_ars_tb:g_clk
:vpdff_ars_tb:g_out
1
1
1
1
0