

Proble1:

```

1  library ieee;
2  Use ieee.std_logic_1164.all;
3  Entity mux2to1 Is
4      port (w0,w1,s:IN STD_LOGIC;
5            f : OUT STD_LOGIC );
6  end mux2to1;
7  ARCHITECTURE Behavior OF mux2to1 IS
8      BEGIN
9          WITH s select
10             f<=w0 when '0',
11             w1 when others;
12      END Behavior;
13

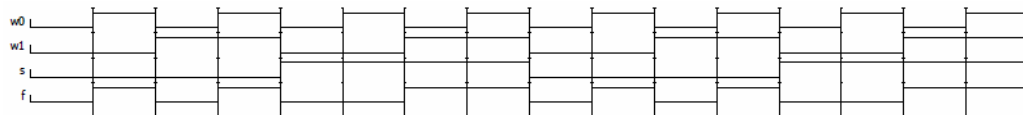
```

Input and output waveforms are generation:

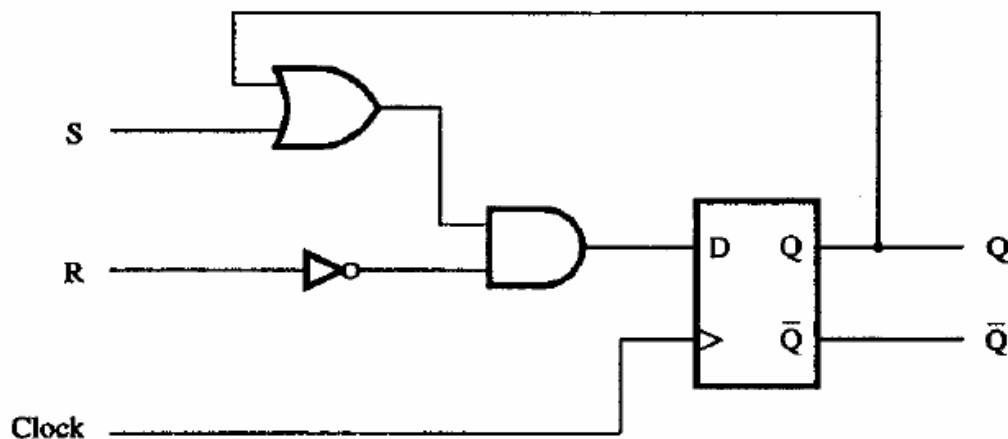
```

vsim mux2to1.vhd
force w0 0 0 ns,1 5 ns -r 10 ns
force w1 0 0 ns,1 10 ns -r 20 ns
force s 0 0 ns,1 20 ns -r 40 ns
run 80ns

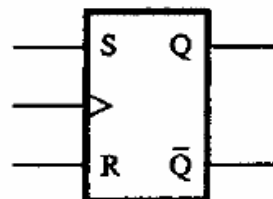
```



7.6

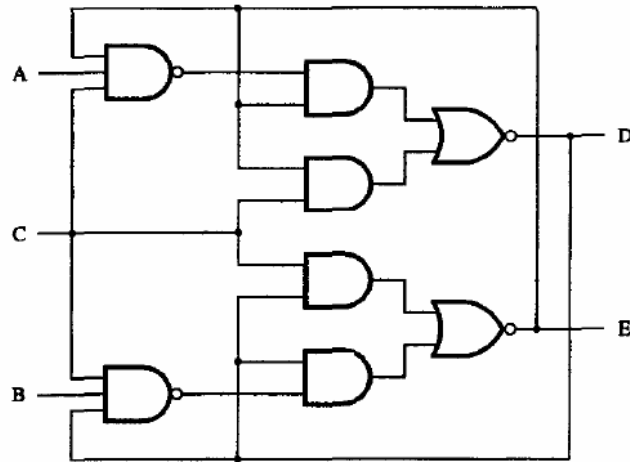


| S | R | $Q(t+1)$ |
|---|---|----------|
| 0 | 0 | $Q(t)$ |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



7.9

As the circuit in Figure P7.2 is drawn, it is not a useful flip-flop circuit, because setting $C = 0$ results in both of the circuit outputs being set to 0. Consider the slightly modified circuit shown below:



This modified circuit acts as a negative-edge-triggered JK flip-flop, in which $J = A$, $K = B$, $Clock = C$, $Q = D$, and $\bar{Q} = E$. This circuit is found in the standard chip called 74LS107A (plus a *Clear* input, which is not shown).

7.18 The counting sequence will be 000,001,010,111

7.28

```

1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3  USE ieee.std_logic_unsigned.all;
4  ENTITY prob7_28 IS
5      PORT (Clock, Reset: IN STD_LOGIC;
6            DATA: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
7            Q: BUFFER STD_LOGIC_VECTOR(3 DOWNTO 0));
8  END prob7_28;
9  ARCHITECTURE Behavior OF prob7_28 IS
10 BEGIN
11     PROCESS (Clock, Reset)
12     BEGIN
13         IF Reset= '1' THEN
14             Q<= "0000";
15         ELSIF Clock' EVENT AND Clock='1' THEN
16             Q<=Q+Data;
17         END IF ;
18     END PROCESS;
19 END Behavior;

```

First compile the code in MODELsim

Then type in command window:

vsim prob7_28

force Clock 0 0 ns, 1 5 ns -r 10 ns

force Reset 0 0 ns, 1 160 ns -r 320 ns

force Data(0) 0 0 ns, 1 10 ns -r 20 ns

force Data(1) 0 0 ns, 1 20 ns -r 40 ns

force Data(2) 0 0 ns, 1 40 ns -r 80 ns

force Data(3) 0 0 ns, 1 80 ns -r 160 ns

run 320 ns

