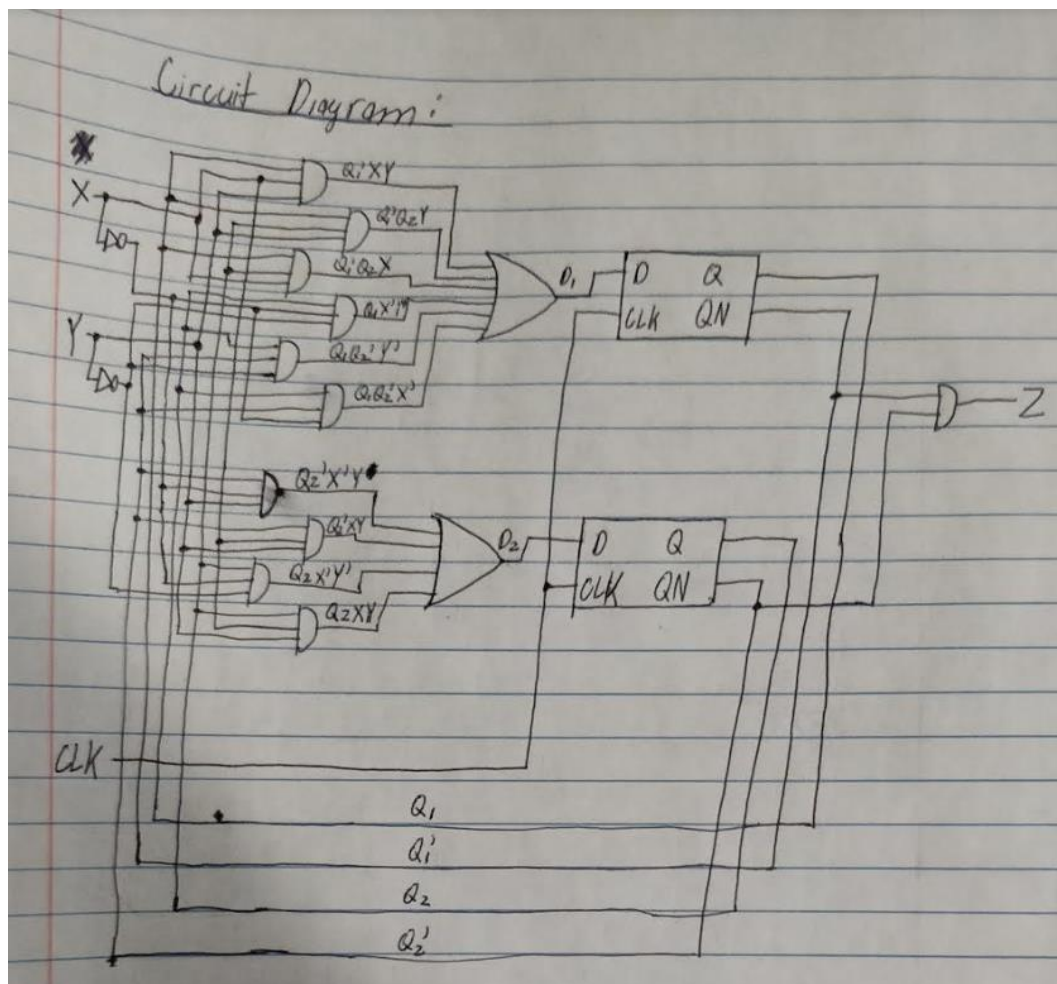


## Task 1: Counting 1s from two inputs (Structural)

This module is a clock synchronous state machine that takes two inputs X and Y alongside a clock input and counts how many 1s appear. The output Z will be 1 if the number of 1s so far is a multiple of 4 and 0 otherwise. The following will be the structural implementation of this module.

S	XY				Z
	00	01	10	11	
S0	S0	S1	S1	S2	1
S1	S1	S2	S2	S3	0
S2	S2	S3	S3	S0	0
S3	S3	S0	S0	S1	0

**S\***



```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  library UNISIM;
4  use UNISIM.VComponents.all;
5
6  entity counter_XY_structural is
7      Port ( CLK : in  STD_LOGIC;
8            X  : in  STD_LOGIC;
9            Y  : in  STD_LOGIC;
10           Z  : out STD_LOGIC);
11 end counter_XY_structural;
12
13 architecture Behavioral of counter_XY_structural is
14     signal Q1,Q1_L,Q2,Q2_L,X_L,Y_L,D1,D2,t0,t1,t2,t3,t4,t5,t6,t7,t8,t9: std_logic;
15     component INV port(I: in std_logic;
16                       O: out std_logic);
17     end component;
18     component AND3 port(I2,I1,I0: in std_logic;
19                       O: out std_logic);
20     end component;
21     component OR4 port(I3,I2,I1,I0: in std_logic;
22                      O: out std_logic);
23     end component;
24     component OR6 port(I5,I4,I3,I2,I1,I0: in std_logic;
25                      O: out std_logic);
26     end component;
27     component pet_d_ff
28     port (CLK,D: in std_logic;
29          Q,QN: out std_logic);
30     end component;
31 begin
32     U0: pet_d_ff port map(CLK,D1,Q1,Q1_L);
33     U1: pet_d_ff port map(CLK,D2,Q2,Q2_L);
34     U2: Z <= Q1_L and Q2_L;
35     U3: INV port map(X,X_L);
36     U4: INV port map(Y,Y_L);
37     U5: AND3 port map(Q1_L,X,Y,t0);
38     U6: AND3 port map(Q1_L,Q2,Y,t1);
39     U7: AND3 port map(Q1_L,Q2,X,t2);
40     U8: AND3 port map(Q1,X_L,Y_L,t3);
41     U9: AND3 port map(Q1,Q2_L,Y_L,t4);
42     U10: AND3 port map(Q1,Q2_L,X_L,t5);
43     U11: AND3 port map(Q2_L,X_L,Y,t6);
44     U12: AND3 port map(Q2_L,X,Y_L,t7);
45     U13: AND3 port map(Q2,X_L,Y_L,t8);
46     U14: AND3 port map(Q2,X,Y,t9);
47     U15: OR6 port map(t0,t1,t2,t3,t4,t5,D1);
48     U16: OR4 port map(t6,t7,t8,t9,D2);
49 end Behavioral;
50

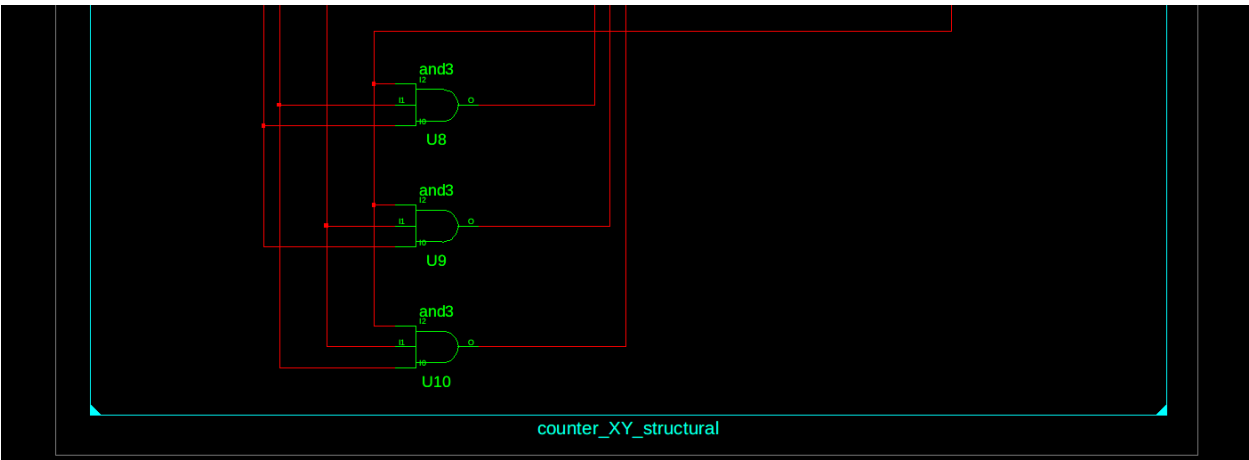
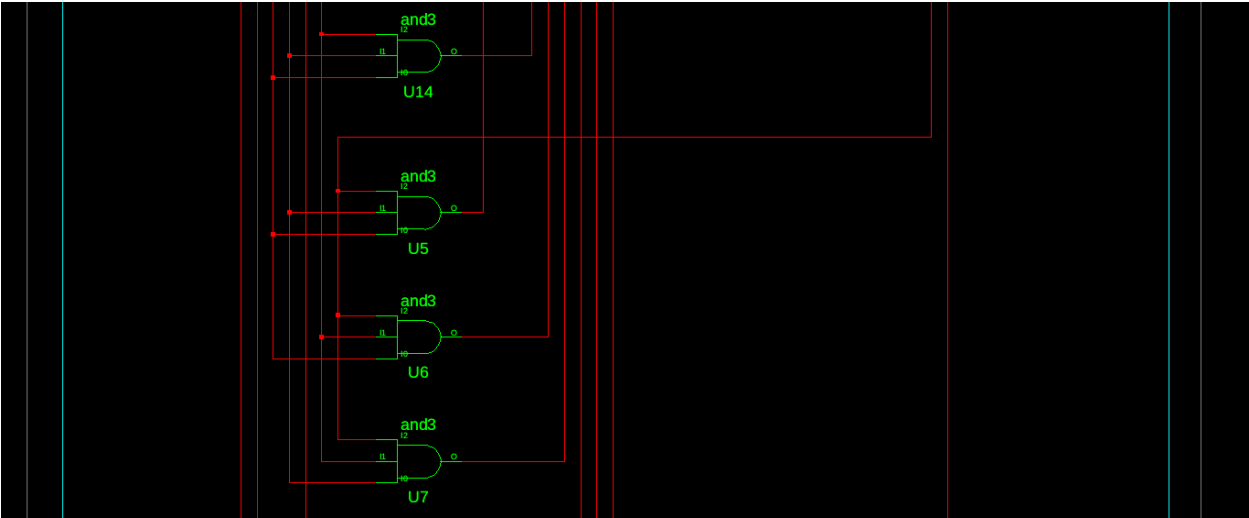
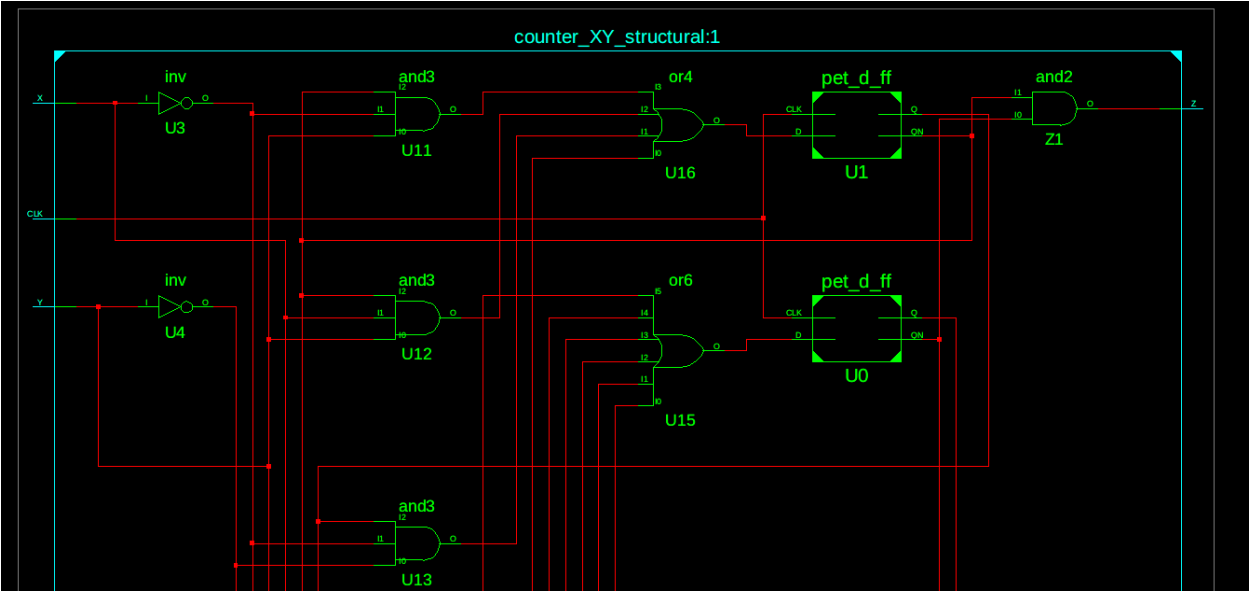
```

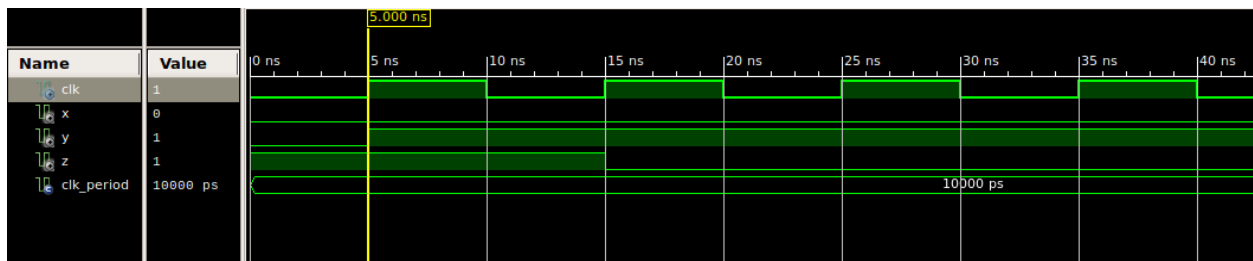
```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3
4  ENTITY TB_structural IS
5  END TB_structural;
6
7  ARCHITECTURE behavior OF TB_structural IS
8
9      -- Component Declaration for the Unit Under Test (UUT)
10     COMPONENT counter_XY_structural
11     PORT(
12         CLK : IN  std_logic;
13         X : IN  std_logic;
14         Y : IN  std_logic;
15         Z : OUT std_logic
16     );
17     END COMPONENT;
18
19     --Inputs
20     signal CLK : std_logic := '0';
21     signal X : std_logic := '0';
22     signal Y : std_logic := '0';
23     --Outputs
24     signal Z : std_logic;
25     -- Clock period definitions
26     constant CLK_period : time := 10 ns;
27
```

```

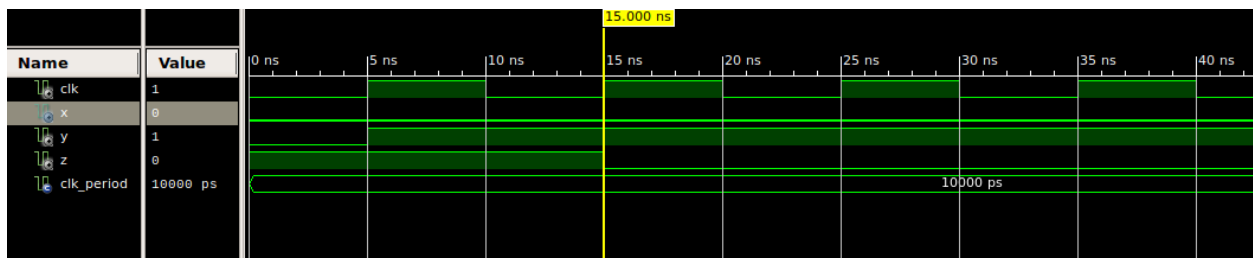
28 BEGIN
29     uut: counter_XY_structural PORT MAP (
30         CLK => CLK,
31         X => X,
32         Y => Y,
33         Z => Z
34     );
35
36     -- Clock process definitions
37     CLK_process :process
38     begin
39         CLK <= '0';
40         wait for CLK_period/2;
41         CLK <= '1';
42         wait for CLK_period/2;
43     end process;
44
45     -- Stimulus process
46     stim_proc: process
47     begin
48         wait for 5 ns;
49         -- Test case #1
50         X<='0'; Y<='1'; wait for CLK_period*4; X<='0'; Y<='0';
51
52         -- Test case #2
53         X<='1'; Y<='0'; wait for CLK_period*4; X<='0'; Y<='0'; wait for CLK_period;
54
55         -- Test case #3
56         X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
57         X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
58         X<='0'; Y<='0'; wait for CLK_period*2;
59
60         -- Test case #4
61         X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
62         X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
63         X<='0'; Y<='0'; wait for CLK_period*2;
64
65         -- Test case #5
66         X<='0'; Y<='1'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
67         X<='0'; Y<='0'; wait for CLK_period*2;
68         X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
69         X<='0'; Y<='0'; wait for CLK_period*2;
70
71         wait;
72     end process;
73 END;
74

```

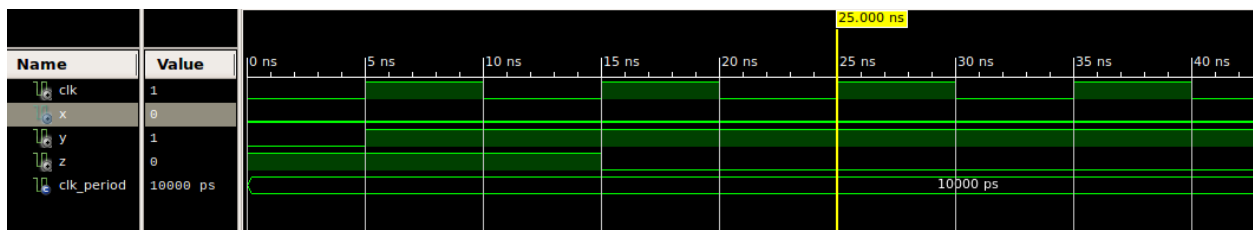




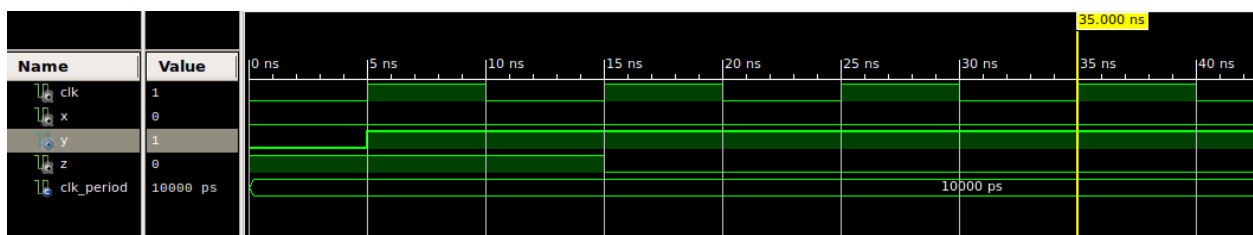
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



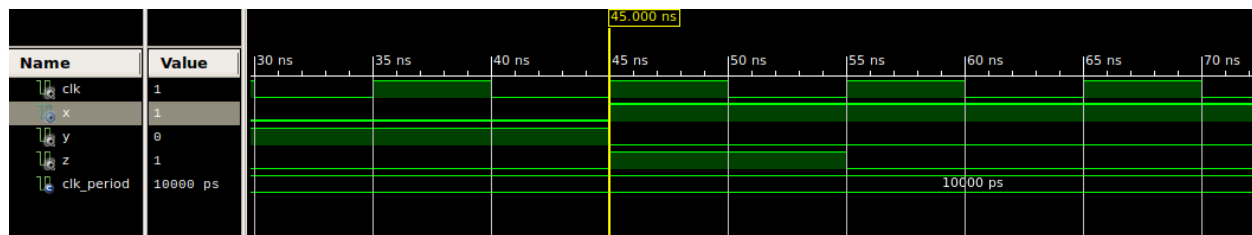
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



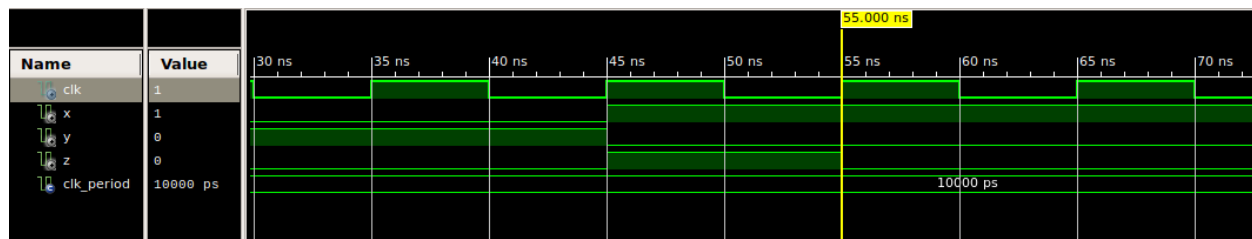
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



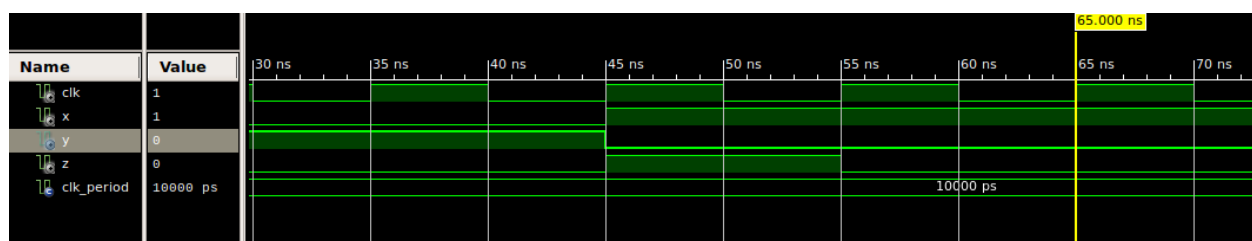
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



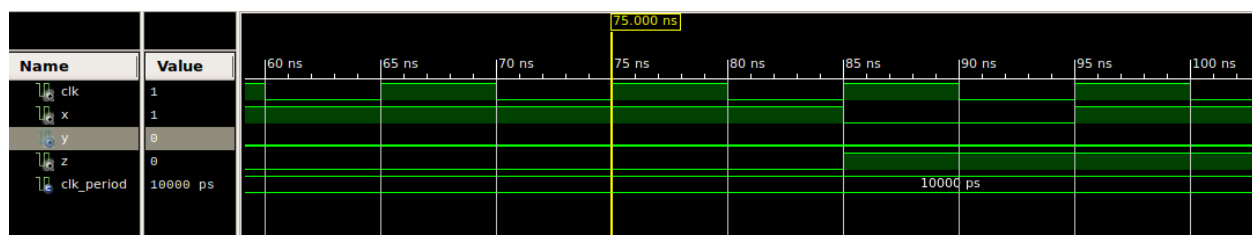
This outcome is correct because the current state is s3 and right before the clock tick X is 0 and Y is 1, therefore the next state is s0 and Z is 1.



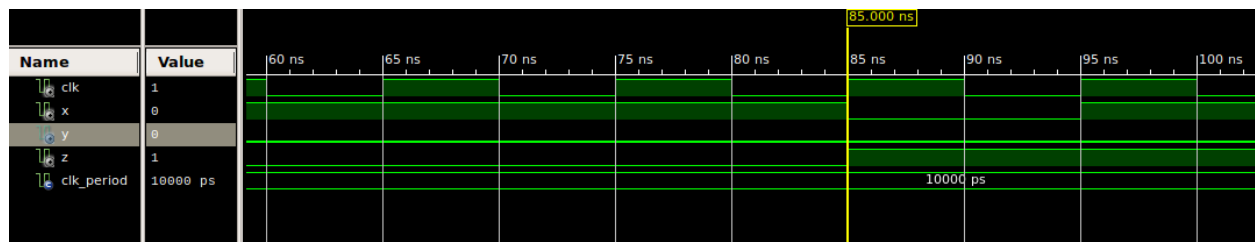
This outcome is correct because the current state is s0 and right before the clock tick X is 1 and Y is 0, therefore the next state is s1 and Z is 0.



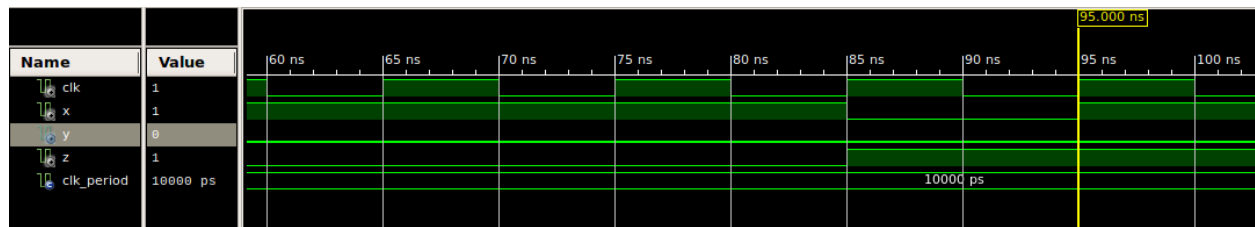
This outcome is correct because the current state is s1 and right before the clock tick X is 1 and Y is 0, therefore the next state is s2 and Z is 0.



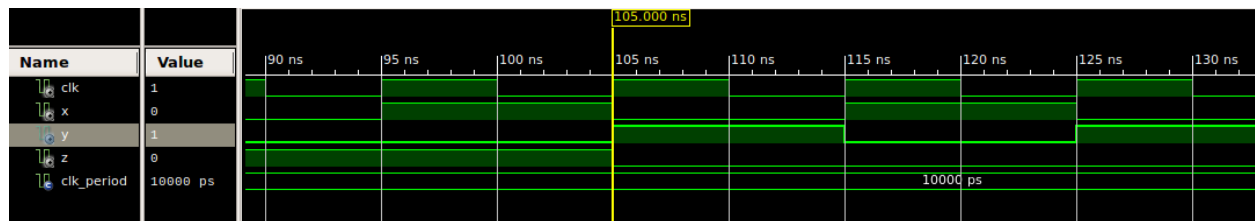
This outcome is correct because the current state is s2 and right before the clock tick X is 1 and Y is 0, therefore the next state is s3 and Z is 0.



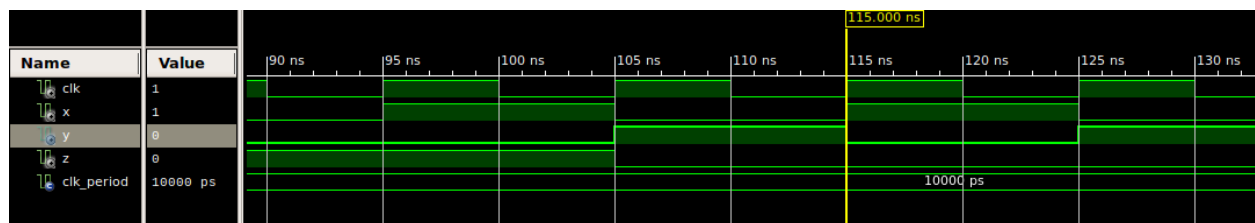
This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.



This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.

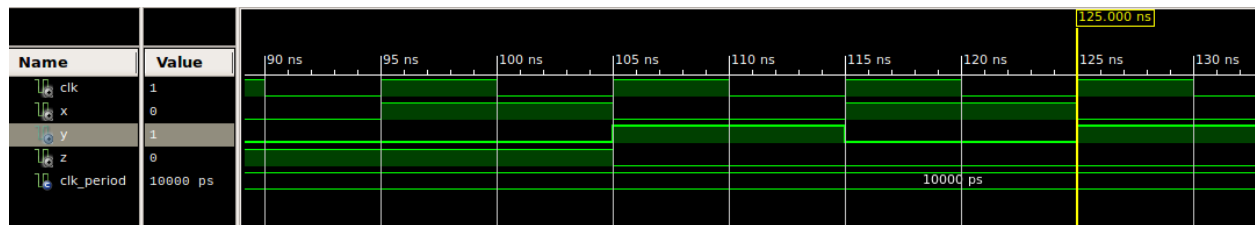


This outcome is correct because the current state is s0 and right before the clock tick X is 1 and Y is 0, therefore the next state is s1 and Z is 0.

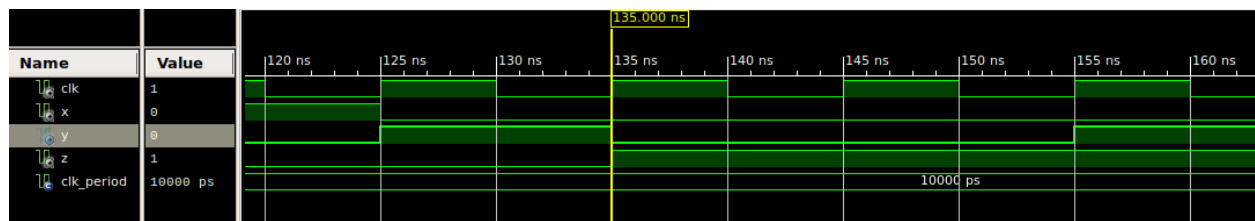


This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.

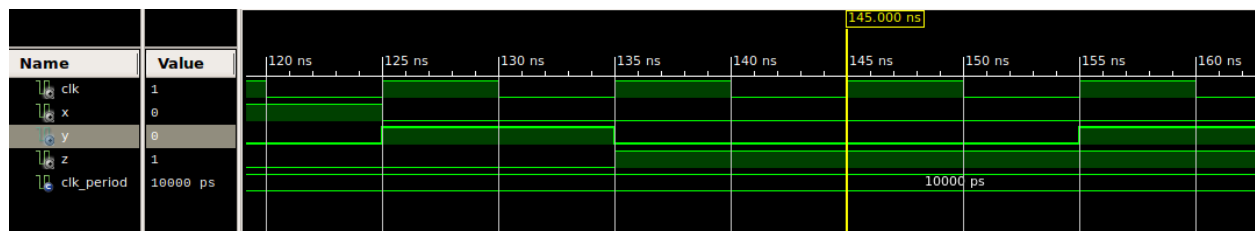




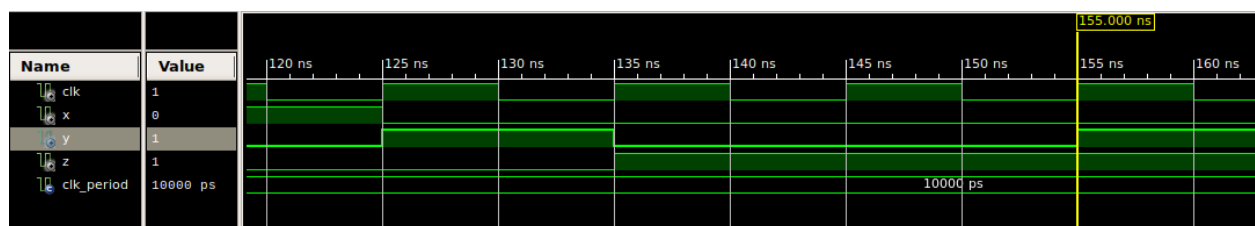
This outcome is correct because the current state is s2 and right before the clock tick X is 1 and Y is 0, therefore the next state is s3 and Z is 0.



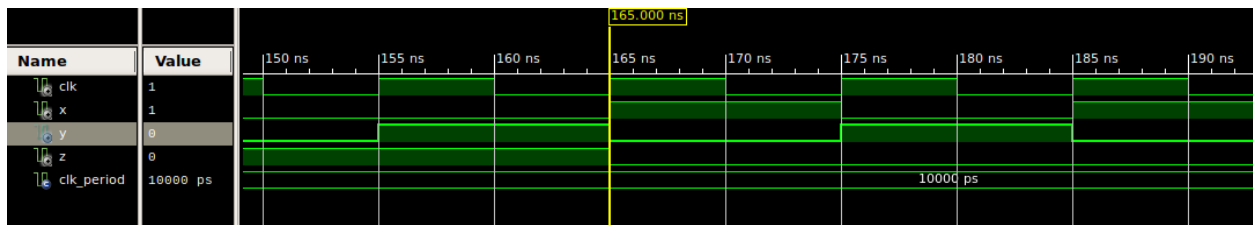
This outcome is correct because the current state is s3 and right before the clock tick X is 0 and Y is 1, therefore the next state is s0 and Z is 1.



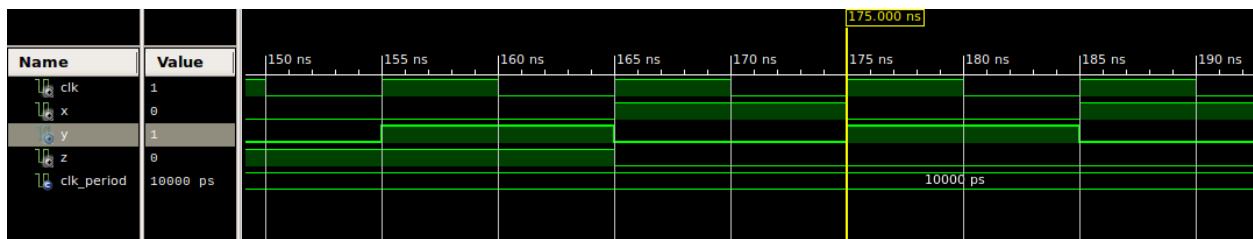
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



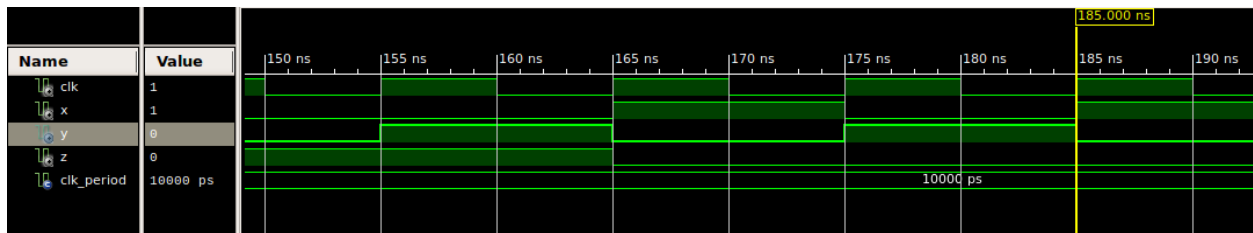
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



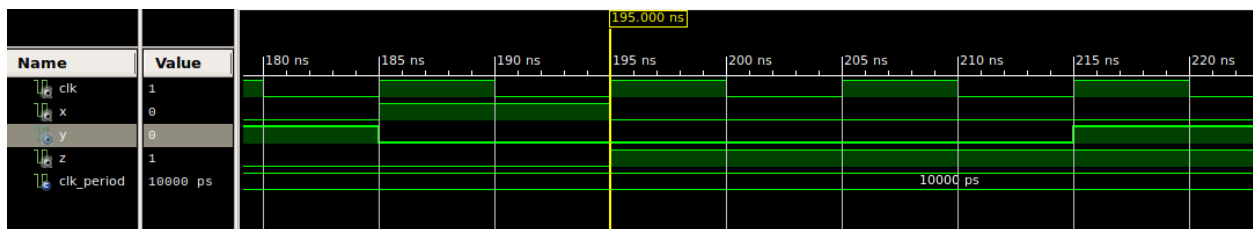
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



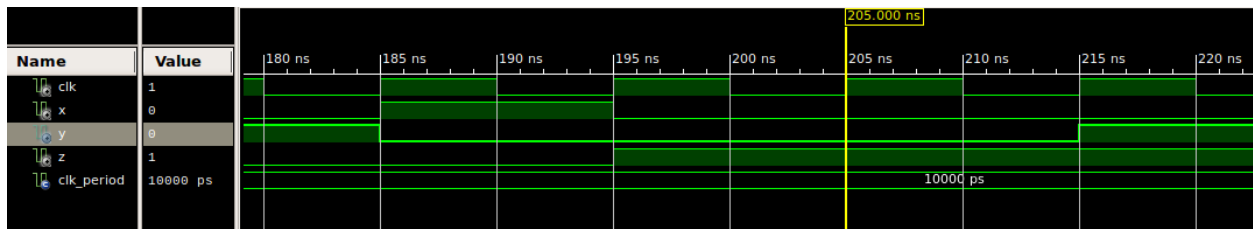
This outcome is correct because the current state is s1 and right before the clock tick X is 1 and Y is 0, therefore the next state is s2 and Z is 0.



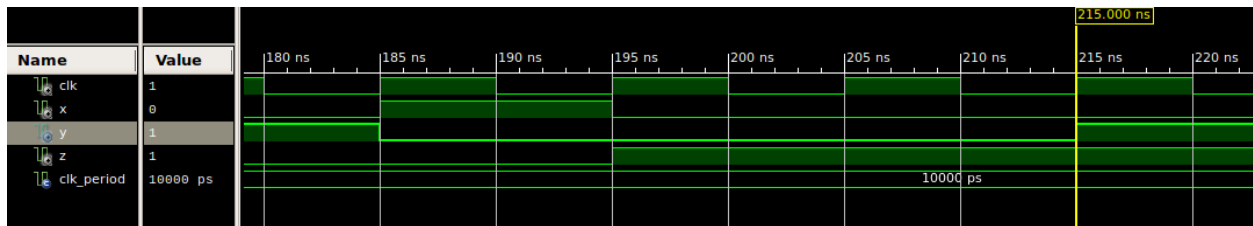
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



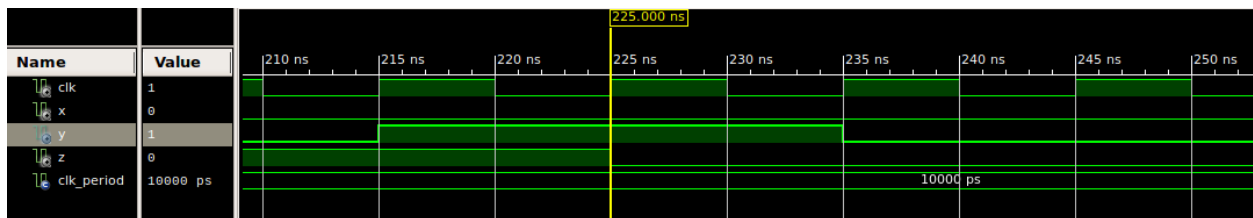
This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.



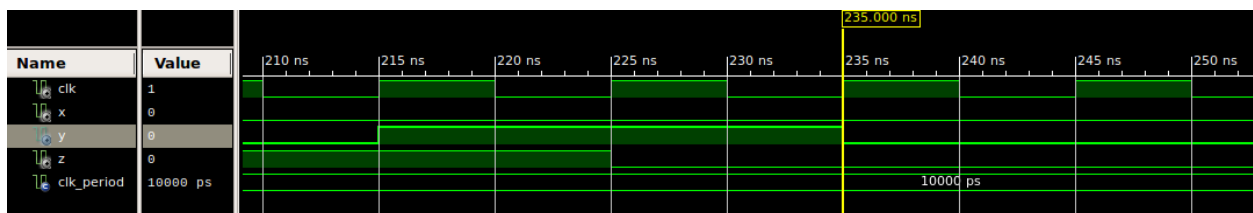
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



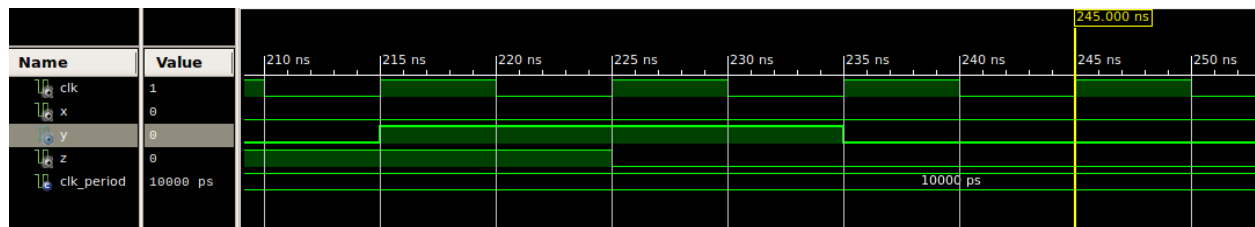
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



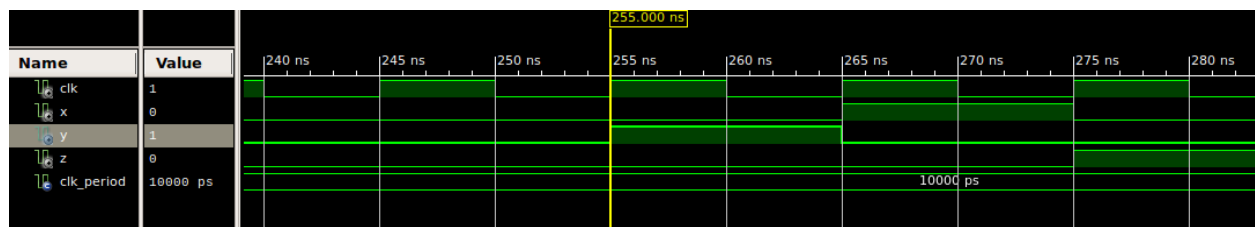
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



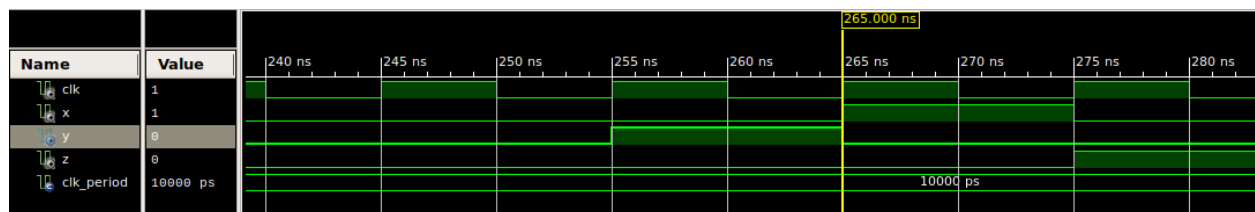
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



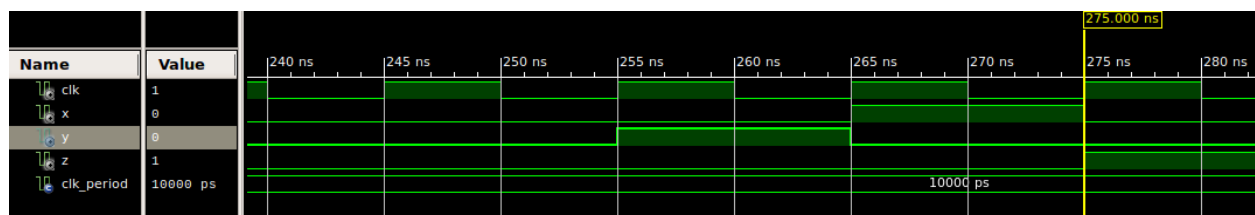
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 0, therefore the next state is s2 and Z is 0.



This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 0, therefore the next state is s2 and Z is 0.



This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.

## Task 2: Counting 1s from two inputs (Behavioral)

This module is a clock synchronous state machine that takes two inputs X and Y alongside a clock input and counts how many 1s appear. The output Z will be 1 if the number of 1s so far is a multiple of 4 and 0 otherwise. The following will be the behavioral implementation of this module.

S	XY				Z
	00	01	10	11	
S0	S0	S1	S1	S2	1
S1	S1	S2	S2	S3	0
S2	S2	S3	S3	S0	0
S3	S3	S0	S0	S1	0
S*					

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 library UNISIM;
4 use UNISIM.VComponents.all;
5
6 entity counter_XY_behavioral is
7     Port ( CLK : in  STD_LOGIC;
8           X : in  STD_LOGIC;
9           Y : in  STD_LOGIC;
10          Z : out  STD_LOGIC);
11 end counter_XY_behavioral;
12
13 architecture Behavioral of counter_XY_behavioral is
14     type state_type is (s0,s1,s2,s3);
15     signal state: state_type := s0;
16     signal X0,Y0: std_logic := '0';
17 begin
18     process(CLK)
19     begin
20         if rising_edge(CLK) then
21             case state is
22                 when s0 =>
23                     if X0='0' and Y0='0' then
24                         state <= s0;
25                     elsif X0='0' and Y0='1' then
26                         state <= s1;
27                     elsif X0='1' and Y0='0' then
28                         state <= s1;
29                     elsif X0='1' and Y0='1' then
30                         state <= s2;
31                     end if;
```

```

32         when s1 =>
33             if X0='0' and Y0='0' then
34                 state <= s1;
35             elsif X0='0' and Y0='1' then
36                 state <= s2;
37             elsif X0='1' and Y0='0' then
38                 state <= s2;
39             elsif X0='1' and Y0='1' then
40                 state <= s3;
41             end if;
42         when s2 =>
43             if X0='0' and Y0='0' then
44                 state <= s2;
45             elsif X0='0' and Y0='1' then
46                 state <= s3;
47             elsif X0='1' and Y0='0' then
48                 state <= s3;
49             elsif X0='1' and Y0='1' then
50                 state <= s0;
51             end if;
52         when s3 =>
53             if X0='0' and Y0='0' then
54                 state <= s3;
55             elsif X0='0' and Y0='1' then
56                 state <= s0;
57             elsif X0='1' and Y0='0' then
58                 state <= s0;
59             elsif X0='1' and Y0='1' then
60                 state <= s1;
61             end if;

```

```

62         end case;
63
64         X0 <= X;
65         Y0 <= Y;
66     end if;
67 end process;
68
69     Z <= '1' when state=s0 else '0';
70 end Behavioral;
71
72

```

```
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.ALL;
3
4  ENTITY TB_behavioral IS
5  END TB_behavioral;
6
7  ARCHITECTURE behavior OF TB_behavioral IS
8      -- Component Declaration for the Unit Under Test (UUT)
9      COMPONENT counter_XY_behavioral
10     PORT(
11         CLK : IN  std_logic;
12         X : IN  std_logic;
13         Y : IN  std_logic;
14         Z : OUT std_logic
15     );
16     END COMPONENT;
17
18     --Inputs
19     signal CLK : std_logic := '0';
20     signal X : std_logic := '0';
21     signal Y : std_logic := '0';
22     --Outputs
23     signal Z : std_logic;
24     -- Clock period definitions
25     constant CLK_period : time := 10 ns;
26
```



```

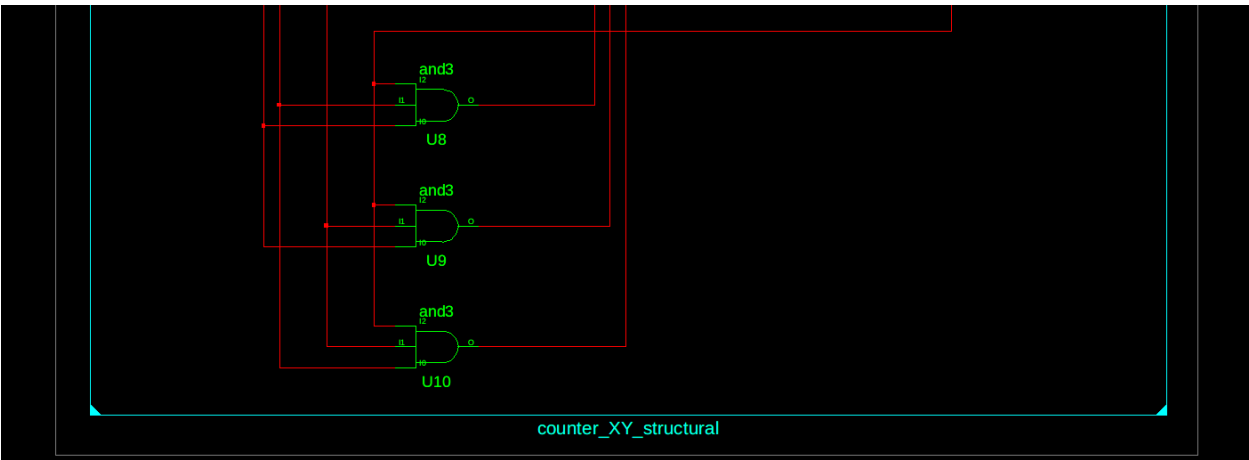
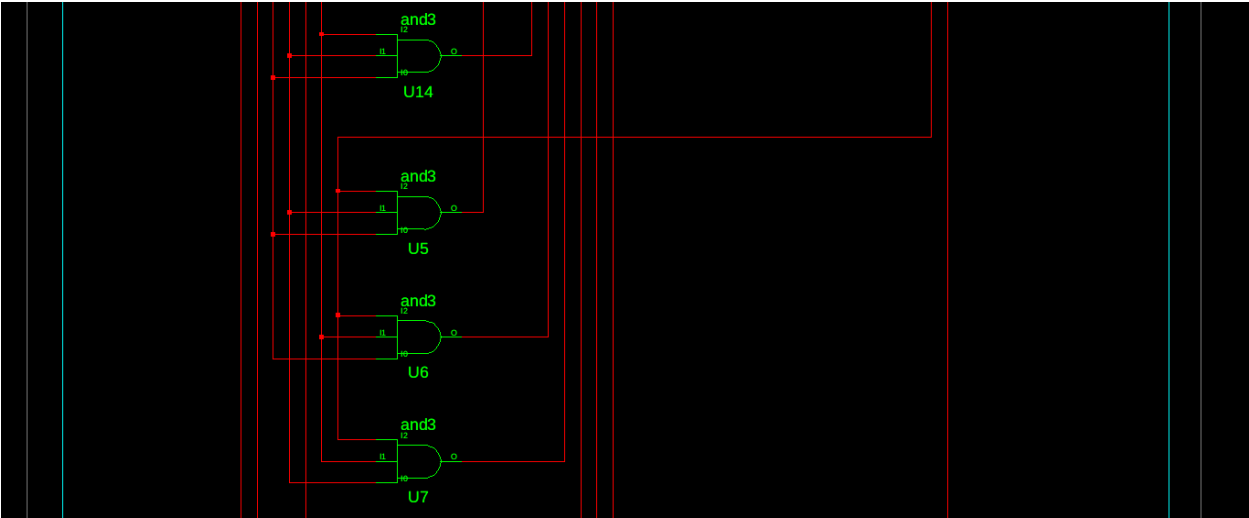
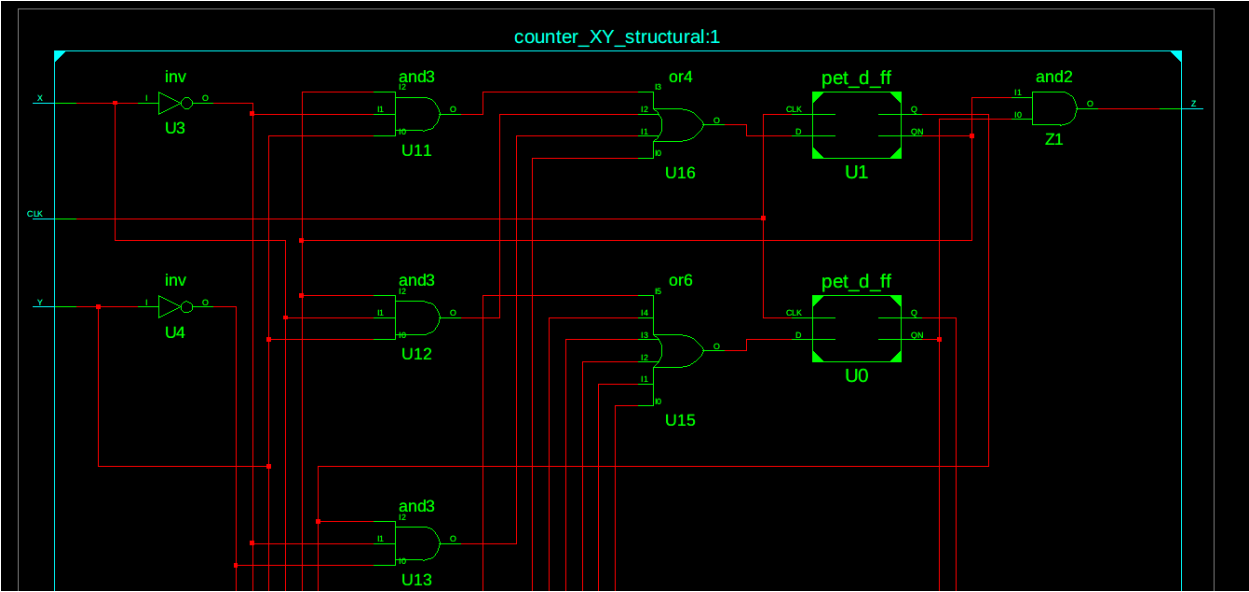
27 BEGIN
28     uut: counter_XY_behavioral PORT MAP (
29         CLK => CLK,
30         X => X,
31         Y => Y,
32         Z => Z
33     );
34
35     -- Clock process definitions
36     CLK_process : process
37     begin
38         CLK <= '0';
39         wait for CLK_period/2;
40         CLK <= '1';
41         wait for CLK_period/2;
42     end process;
43
44     -- Stimulus process
45     stim_proc: process
46     begin
47         wait for 5 ns;
48         -- Test case #1
49         X<='0'; Y<='1'; wait for CLK_period*4; X<='0'; Y<='0';
50
51         -- Test case #2
52         X<='1'; Y<='0'; wait for CLK_period*4; X<='0'; Y<='0'; wait for CLK_period;
53
54

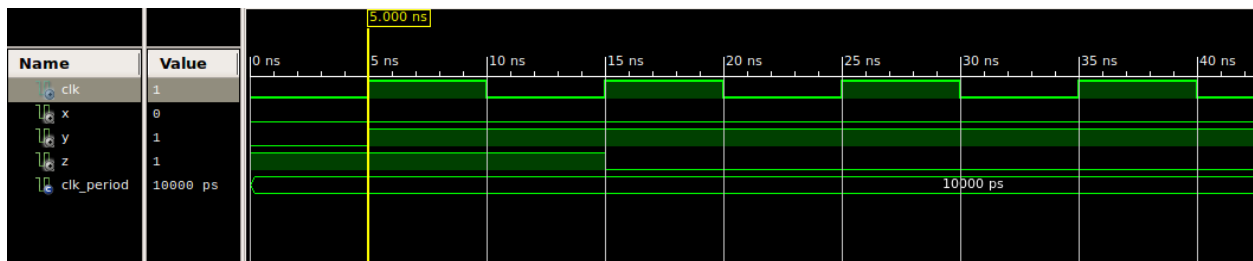
```

```

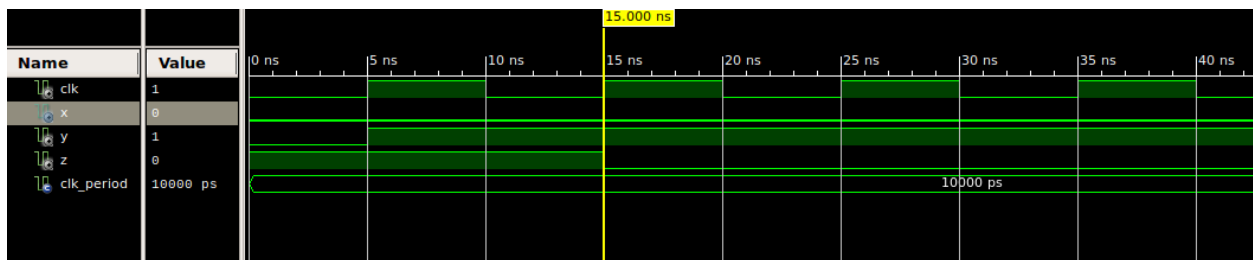
54
55         -- Test case #3
56         X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
57         X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
58         X<='0'; Y<='0'; wait for CLK_period*2;
59
60         -- Test case #4
61         X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
62         X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
63         X<='0'; Y<='0'; wait for CLK_period*2;
64
65         -- Test case #5
66         X<='0'; Y<='1'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
67         X<='0'; Y<='0'; wait for CLK_period*2;
68         X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
69         X<='0'; Y<='0'; wait for CLK_period*2;
70
71         wait;
72     end process;
73
74 END;

```

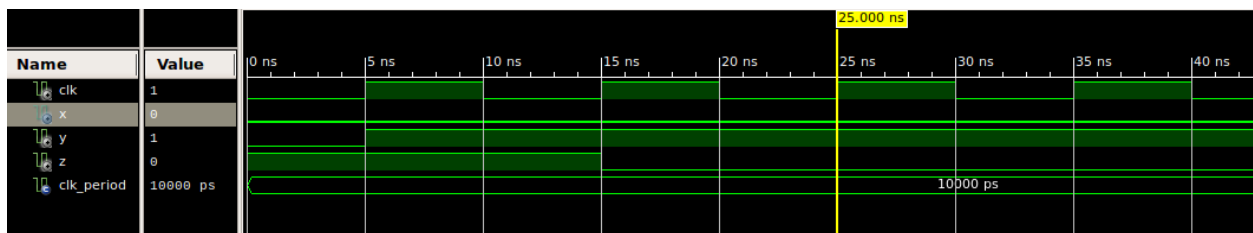




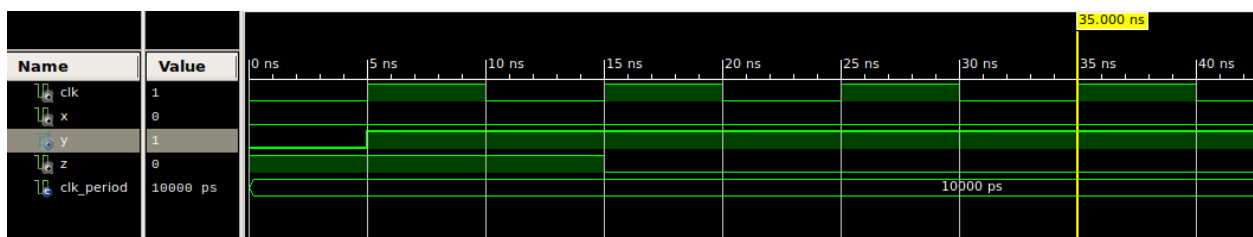
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



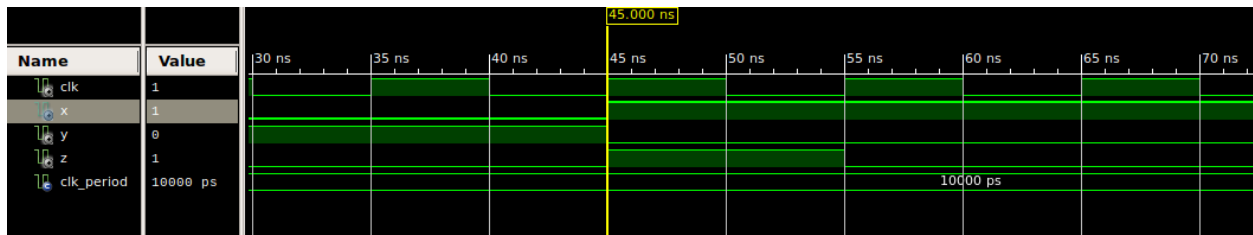
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



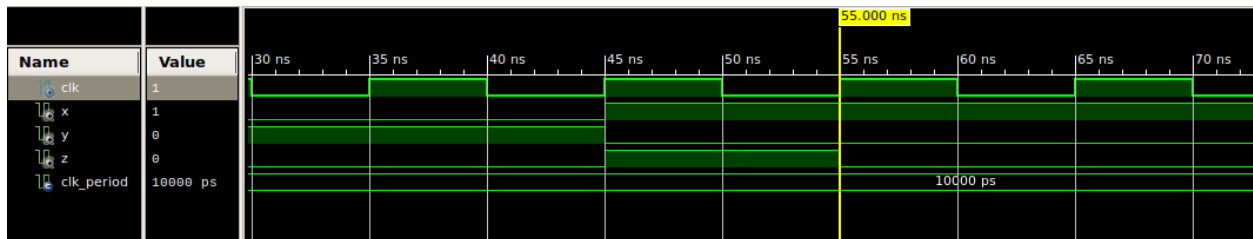
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



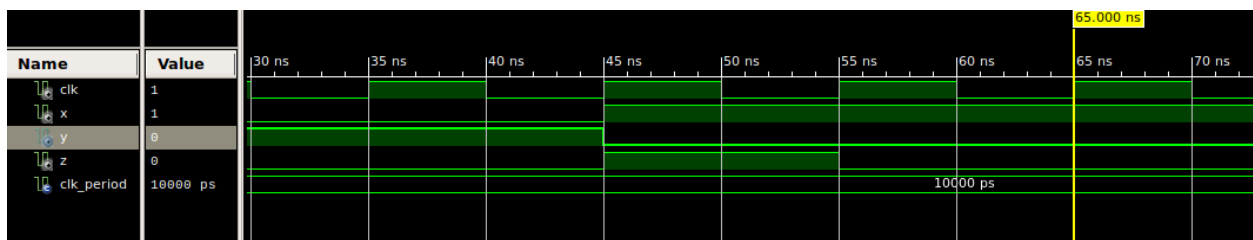
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



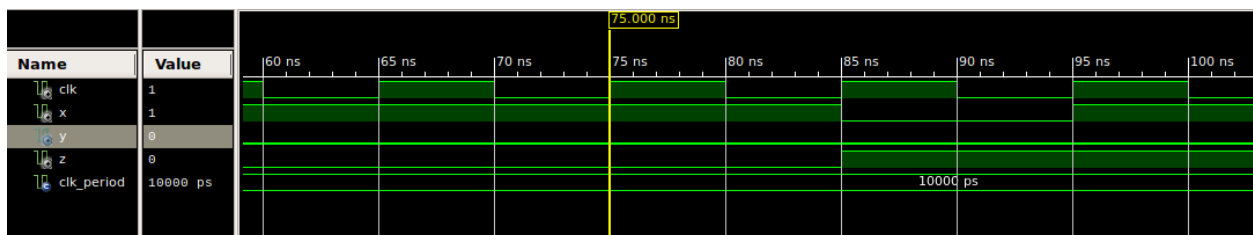
This outcome is correct because the current state is s3 and right before the clock tick X is 0 and Y is 1, therefore the next state is s0 and Z is 1.



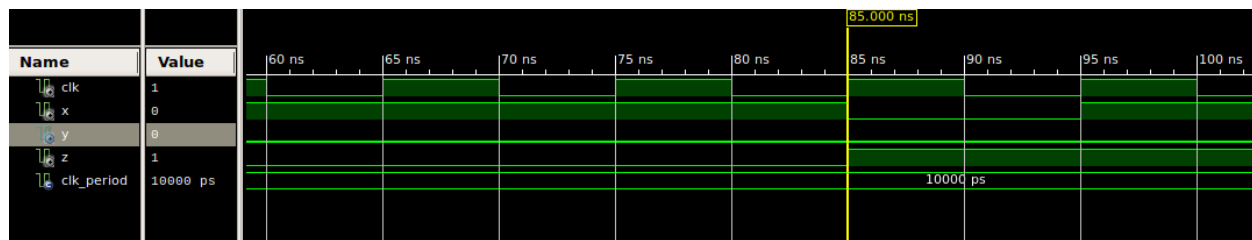
This outcome is correct because the current state is s0 and right before the clock tick X is 1 and Y is 0, therefore the next state is s1 and Z is 0.



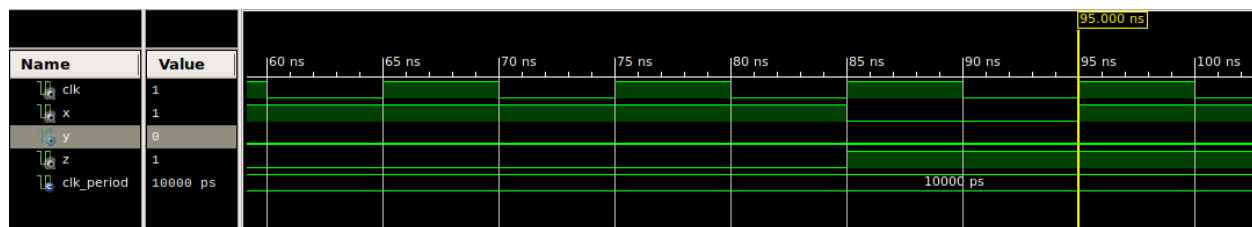
This outcome is correct because the current state is s1 and right before the clock tick X is 1 and Y is 0, therefore the next state is s2 and Z is 0.



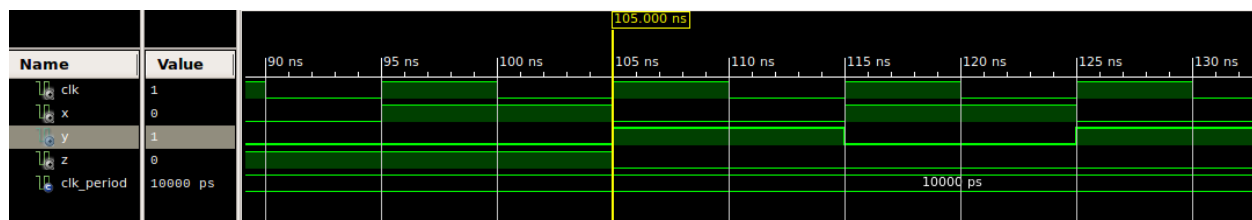
This outcome is correct because the current state is s2 and right before the clock tick X is 1 and Y is 0, therefore the next state is s3 and Z is 0.



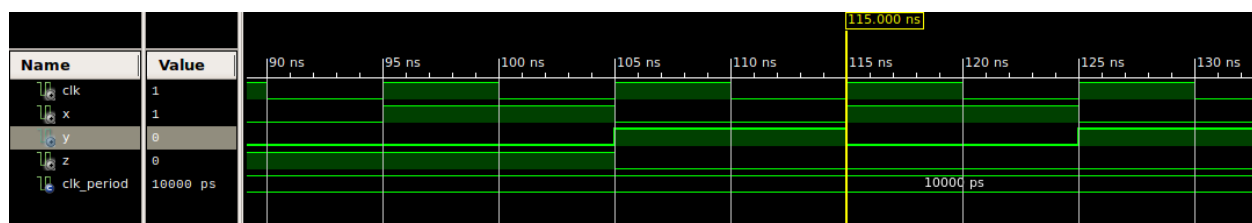
This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.



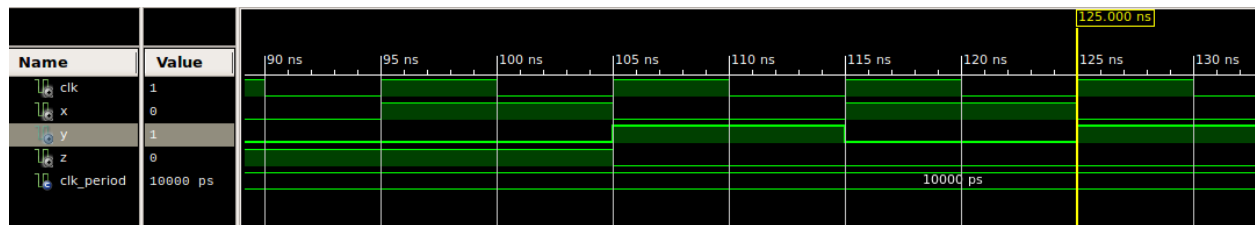
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



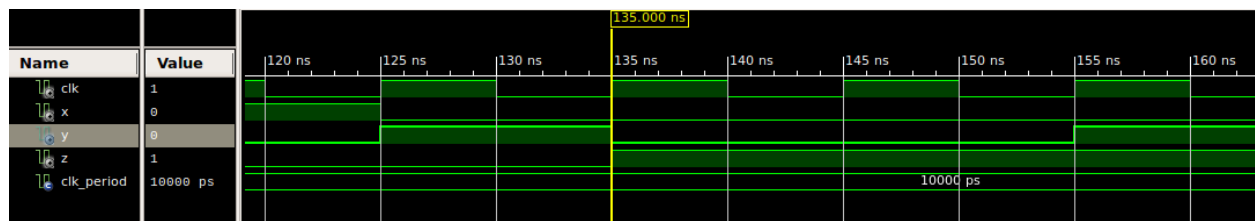
This outcome is correct because the current state is s0 and right before the clock tick X is 1 and Y is 0, therefore the next state is s1 and Z is 0.



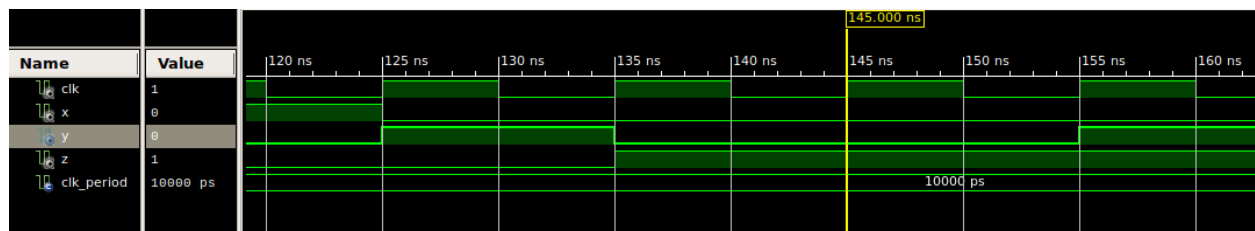
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



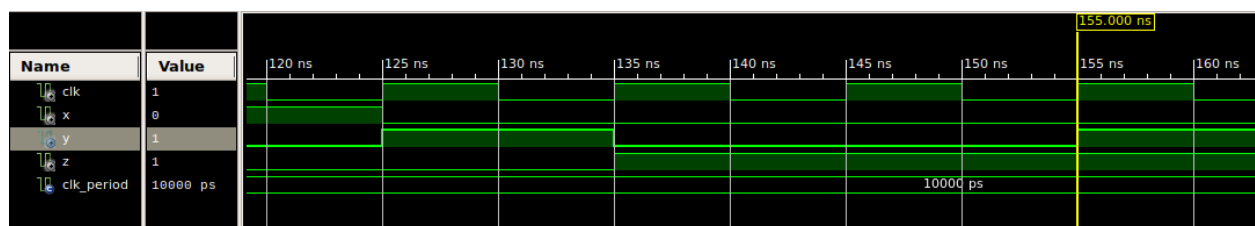
This outcome is correct because the current state is s2 and right before the clock tick X is 1 and Y is 0, therefore the next state is s3 and Z is 0.



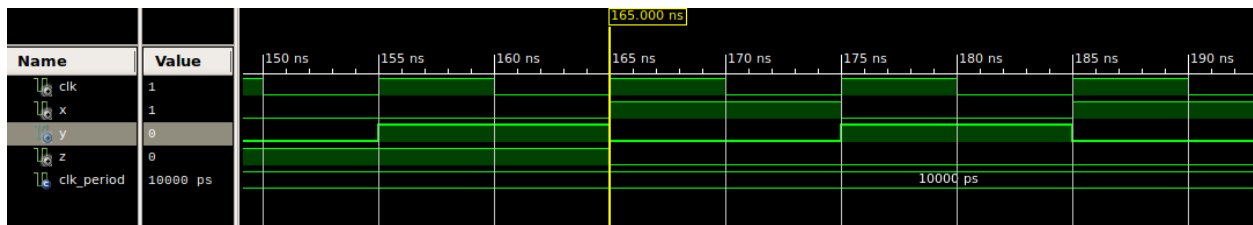
This outcome is correct because the current state is s3 and right before the clock tick X is 0 and Y is 1, therefore the next state is s0 and Z is 1.



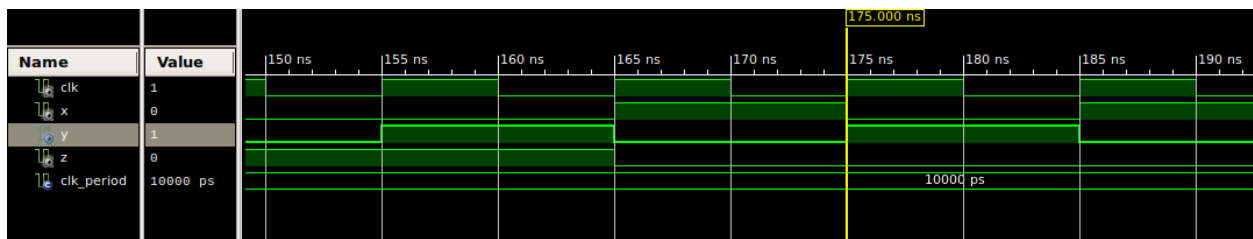
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



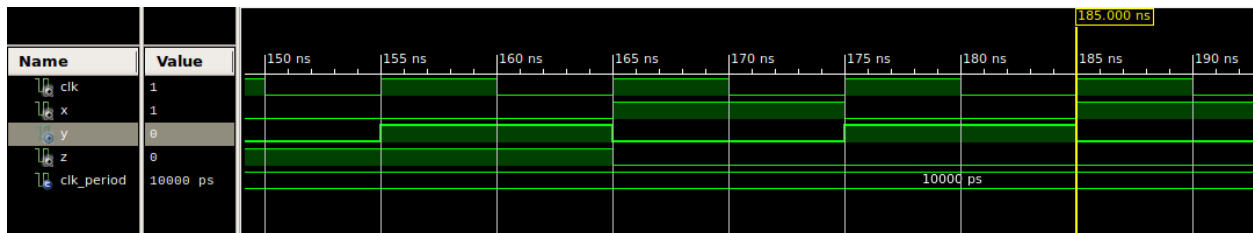
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



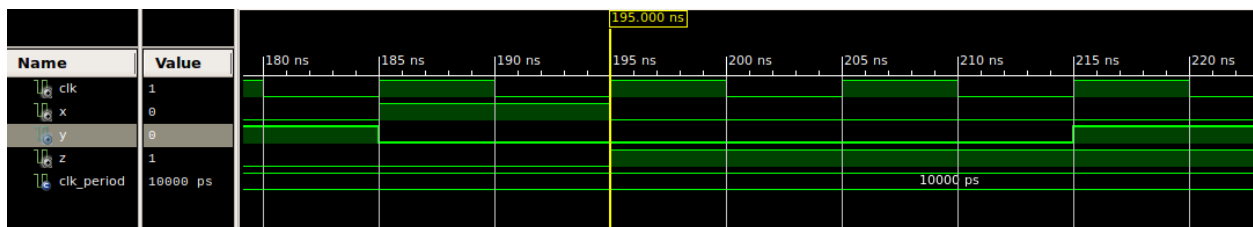
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



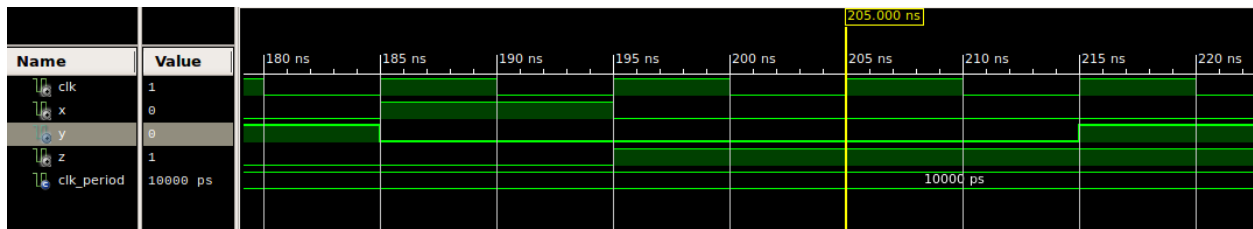
This outcome is correct because the current state is s1 and right before the clock tick X is 1 and Y is 0, therefore the next state is s2 and Z is 0.



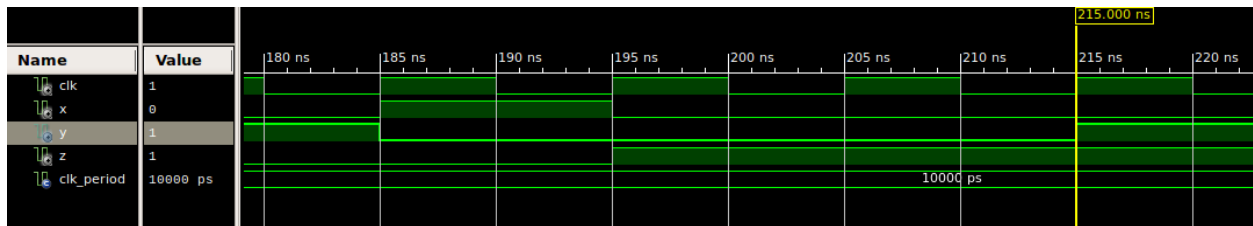
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



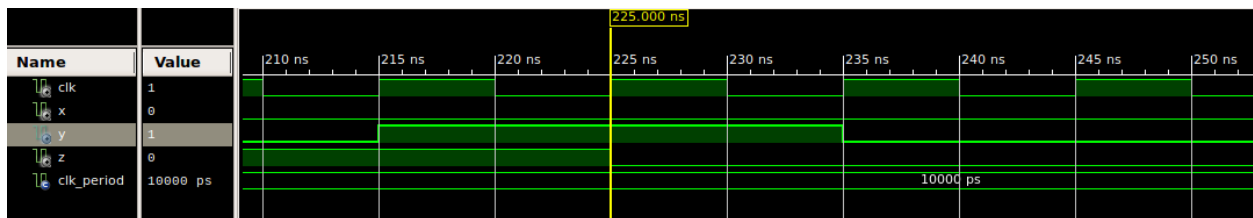
This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.



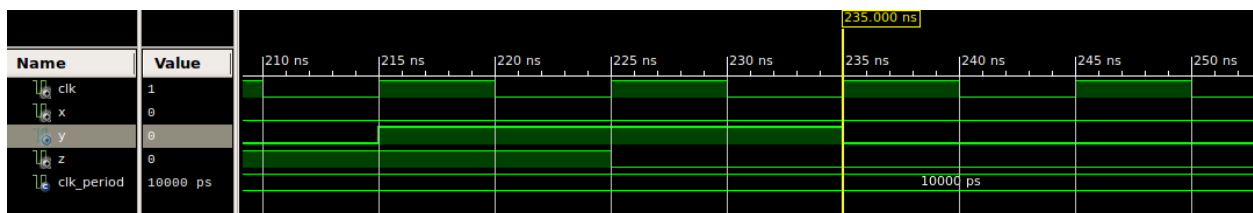
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.

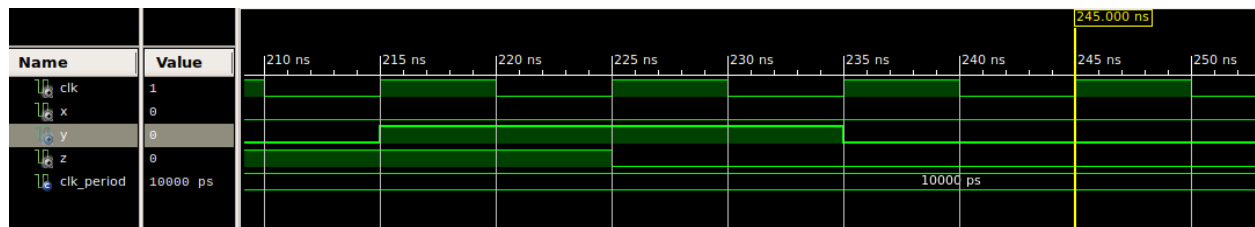


This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.

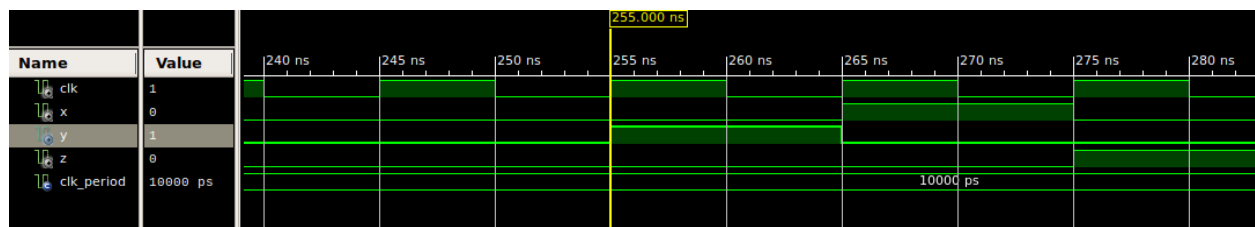


This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.

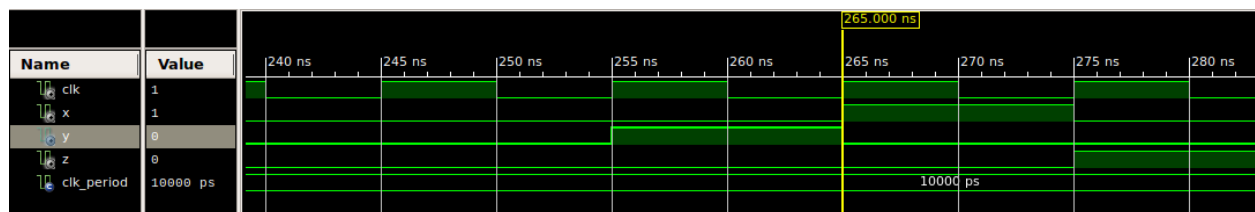




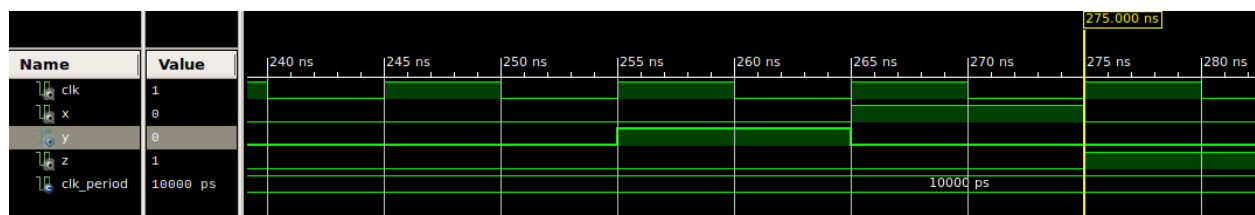
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 0, therefore the next state is s2 and Z is 0.



This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 0, therefore the next state is s2 and Z is 0.



This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.