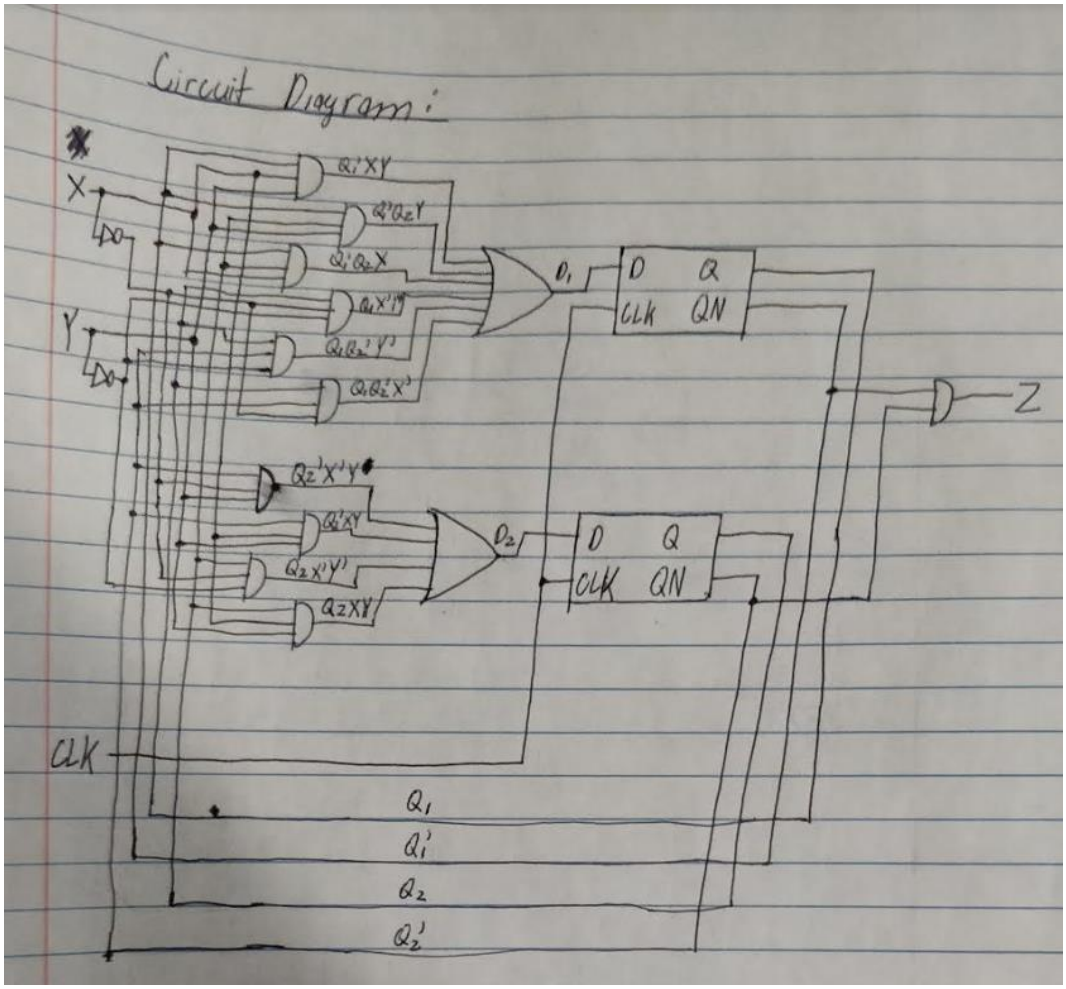
Task 1: Counting 1s from two inputs (Structural)

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.

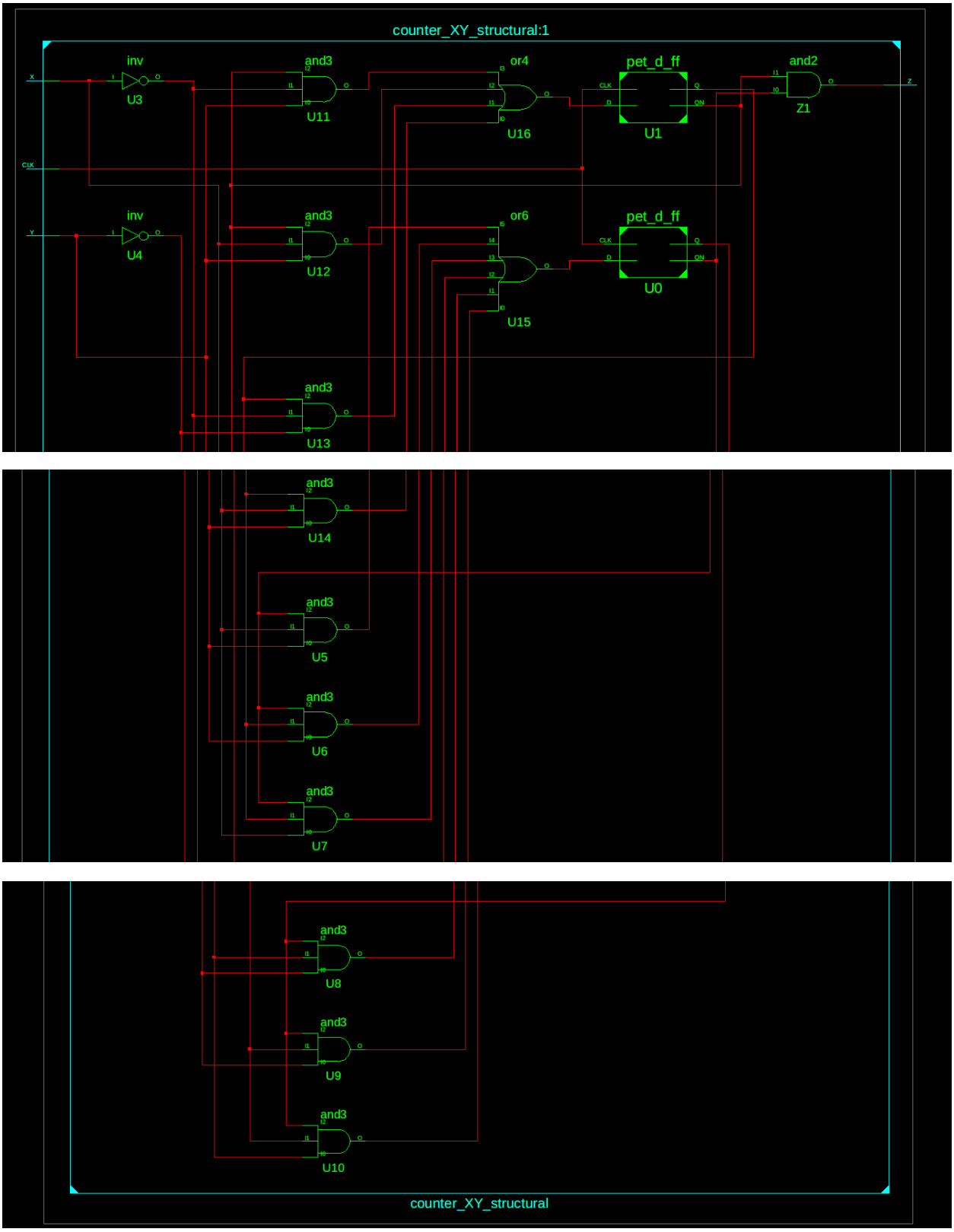
	XY				
S	**00**	**01**	**10**	**11**	**Z**
S0	S0	S1	S1	S2	1
S1	S1	S2	S2	S3	0
S2	S2	S3	S3	S0	0
S3	S3	S0	S0	S1	0
	S*				

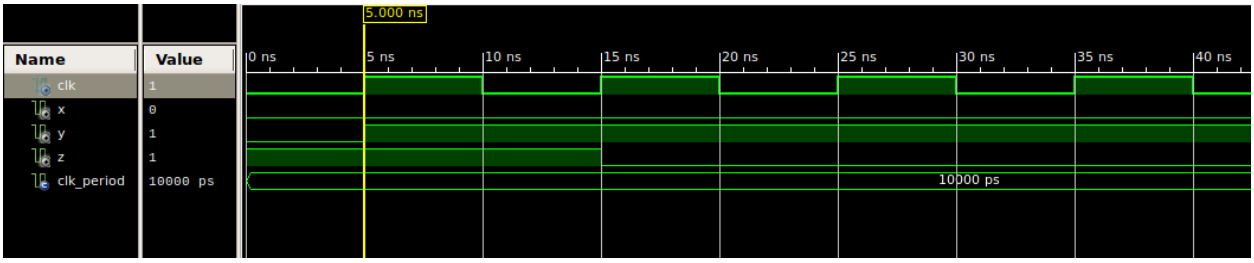


```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity counter_XY_structural is
    Port ( CLK : in  STD_LOGIC;
           X : in  STD_LOGIC;
           Y : in  STD_LOGIC;
           Z : out  STD_LOGIC);
end counter_XY_structural;

architecture Behavioral of counter_XY_structural is
   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;
   component INV port(I: in std_logic;
                      O: out std_logic);
   end component;
   component AND3 port(I2,I1,I0: in std_logic;
                       O: out std_logic);
   end component;
   component OR4 port(I3,I2,I1,I0: in std_logic;
                      O: out std_logic);
   end component;
   component OR6 port(I5,I4,I3,I2,I1,I0: in std_logic;
                      O: out std_logic);
   end component;
   component pet_d_ff
      port(CLK,D: in std_logic;
           Q,QN: out std_logic);
   end component;
begin
     U0: pet_d_ff port map(CLK,D1,Q1,Q1_L);
     U1: pet_d_ff port map(CLK,D2,Q2,Q2_L);
     U2: Z <= Q1_L and Q2_L;
     U3: INV port map(X,X_L);
     U4: INV port map(Y,Y_L);
     U5: AND3 port map(Q1_L,X,Y,t0);
     U6: AND3 port map(Q1_L,Q2,Y,t1);
     U7: AND3 port map(Q1_L,Q2,X,t2);
     U8: AND3 port map(Q1,X_L,Y_L,t3);
     U9: AND3 port map(Q1,Q2_L,Y_L,t4);
    U10: AND3 port map(Q1,Q2_L,X_L,t5);
    U11: AND3 port map(Q2_L,X_L,Y,t6);
    U12: AND3 port map(Q2_L,X,Y_L,t7);
    U13: AND3 port map(Q2,X_L,Y_L,t8);
    U14: AND3 port map(Q2,X,Y,t9);
    U15: OR6 port map(t0,t1,t2,t3,t4,t5,D1);
    U16: OR4 port map(t6,t7,t8,t9,D2);
end Behavioral;

```

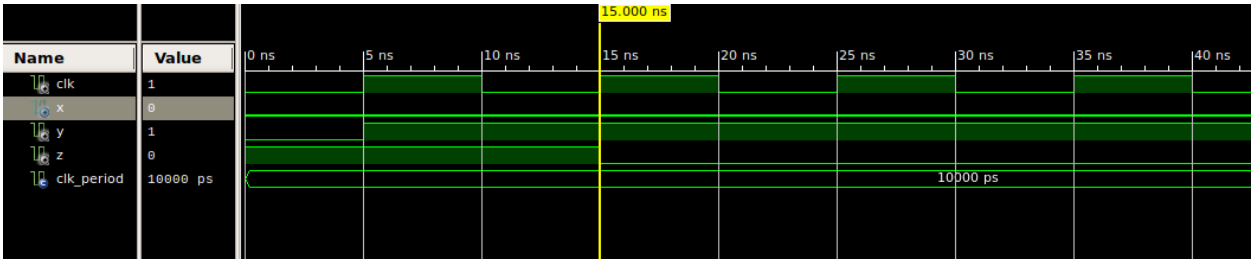
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY TB_structural IS
END TB_structural;

ARCHITECTURE behavior OF TB_structural IS

    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT counter_XY_structural
    PORT(
         CLK : IN  std_logic;
         X : IN  std_logic;
         Y : IN  std_logic;
         Z : OUT  std_logic
        );
    END COMPONENT;

   --Inputs
   signal CLK : std_logic := '0';
   signal X : std_logic := '0';
   signal Y : std_logic := '0';
   --Outputs
   signal Z : std_logic;
   -- Clock period definitions
   constant CLK_period : time := 10 ns;

```

```
BEGIN
   uut: counter_XY_structural PORT MAP (
          CLK => CLK,
          X => X,
          Y => Y,
          Z => Z
        );

   -- Clock process definitions
   CLK_process :process
   begin
      CLK <= '0';
      wait for CLK_period/2;
      CLK <= '1';
      wait for CLK_period/2;
   end process;

   -- Stimulus process
   stim_proc: process
   begin
      wait for 5 ns;
      -- Test case #1
      X<='0'; Y<='1'; wait for CLK_period*4; X<='0'; Y<='0';

      -- Test case #2
      X<='1'; Y<='0'; wait for CLK_period*4; X<='0'; Y<='0'; wait for CLK_period;

      -- Test case #3
      X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
      X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
      X<='0'; Y<='0'; wait for CLK_period*2;

      -- Test case #4
      X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
      X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
      X<='0'; Y<='0'; wait for CLK_period*2;

      -- Test case #5
      X<='0'; Y<='1'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
      X<='0'; Y<='0'; wait for CLK_period*2;
      X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
      X<='0'; Y<='0'; wait for CLK_period*2;

      wait;
   end process;
END;
```

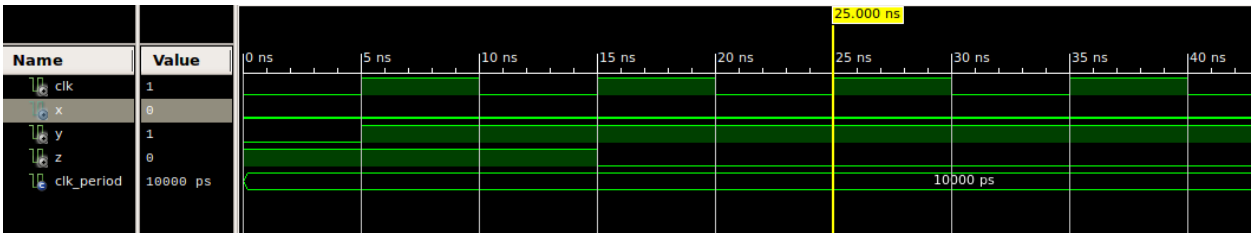

counter_XY_structural:1
inv
U3
and3
U11
or4
U16
pet_d_ff
U1
and2
Z1
X
CLK
Y
Z
inv
U4
and3
U12
or6
U15
pet_d_ff
U0
and3
U13
and3
U14
and3
U5
and3
U6
and3
U7
and3
U8
and3
U9
and3
U10
counter_XY_structural



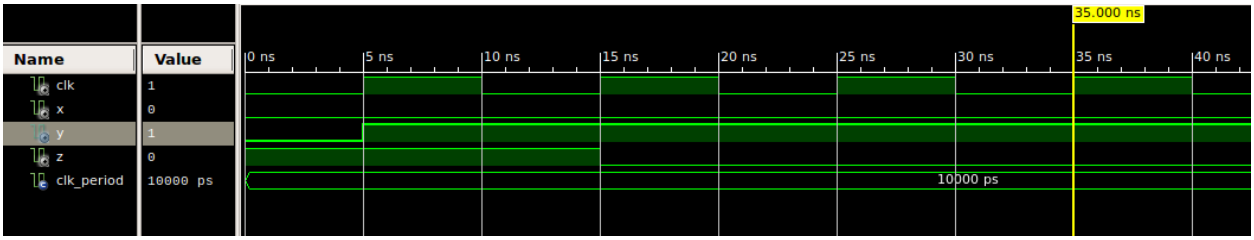
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



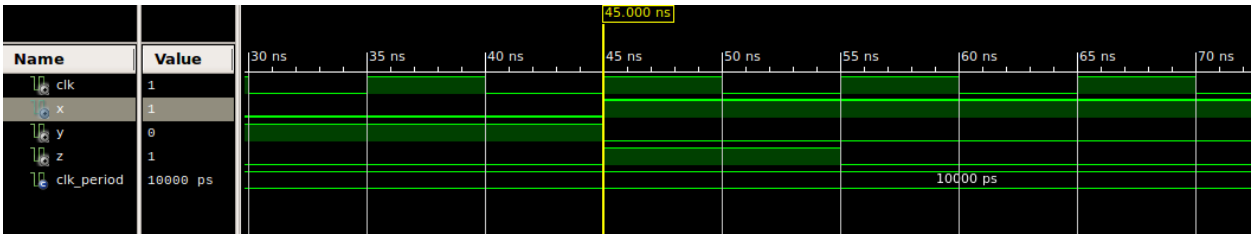
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



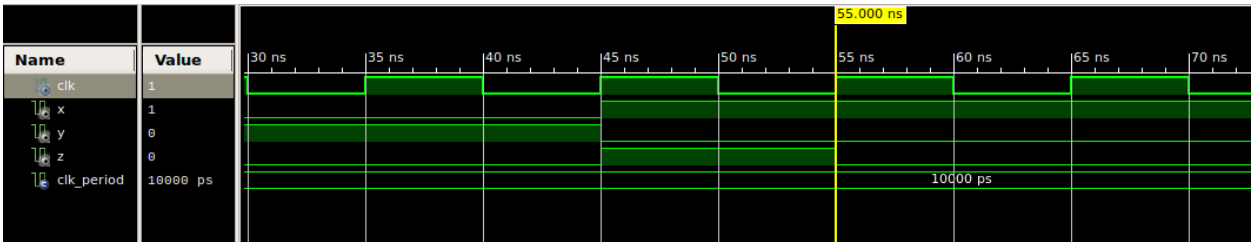
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



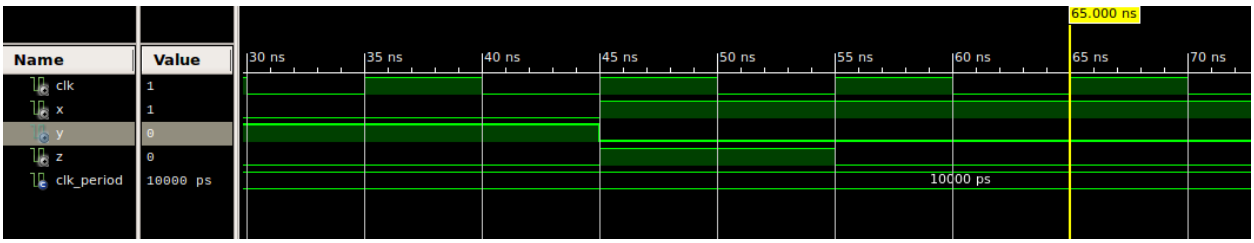
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



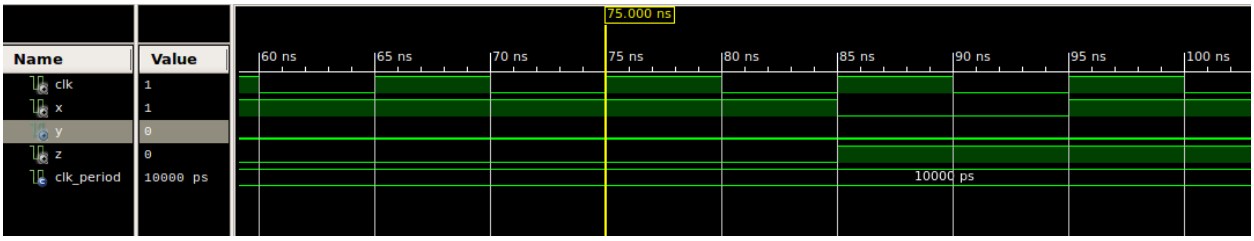
This outcome is correct because the current state is s3 and right before the clock tick X is 0 and Y is 1, therefore the next state is s0 and Z is 1.



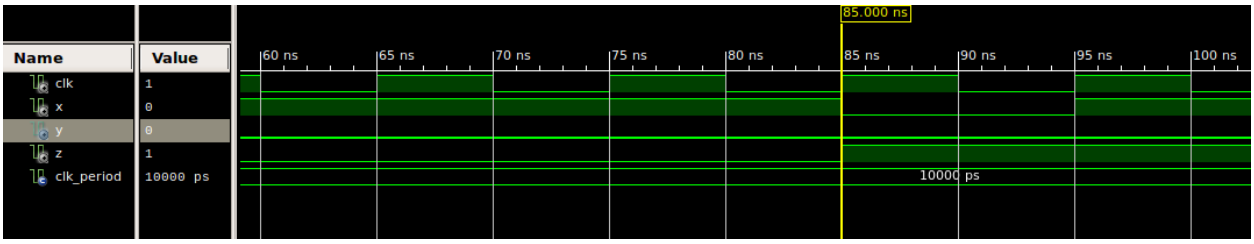
This outcome is correct because the current state is s0 and right before the clock tick X is 1 and Y is 0, therefore the next state is s1 and Z is 0.



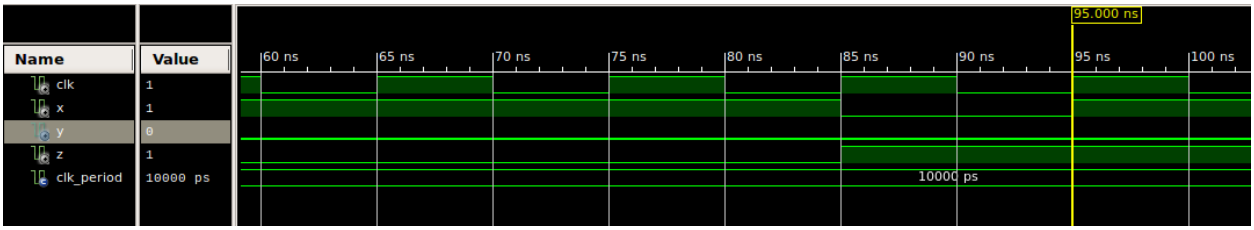
This outcome is correct because the current state is s1 and right before the clock tick X is 1 and Y is 0, therefore the next state is s2 and Z is 0.



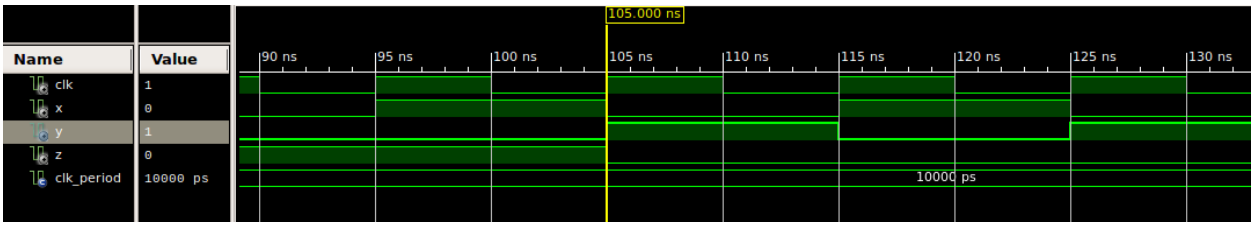
This outcome is correct because the current state is s2 and right before the clock tick X is 1 and Y is 0, therefore the next state is s3 and Z is 0.



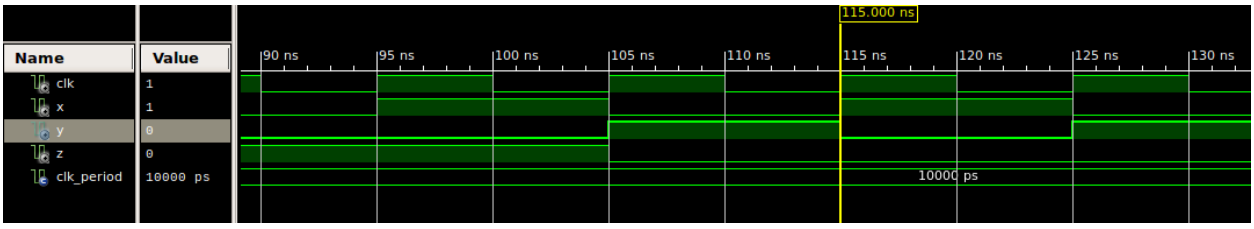
This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.



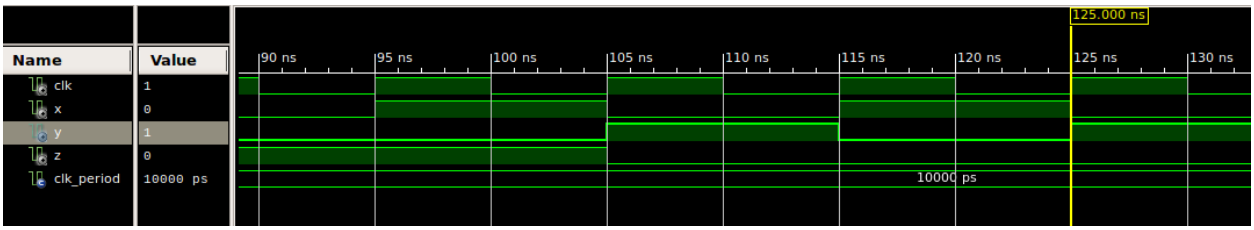
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



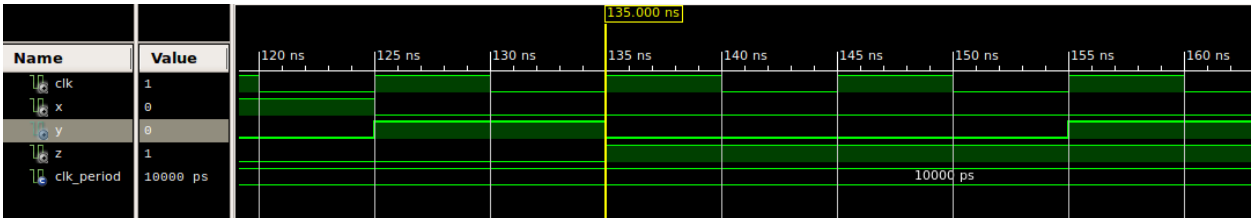
This outcome is correct because the current state is s0 and right before the clock tick X is 1 and Y is 0, therefore the next state is s1 and Z is 0.



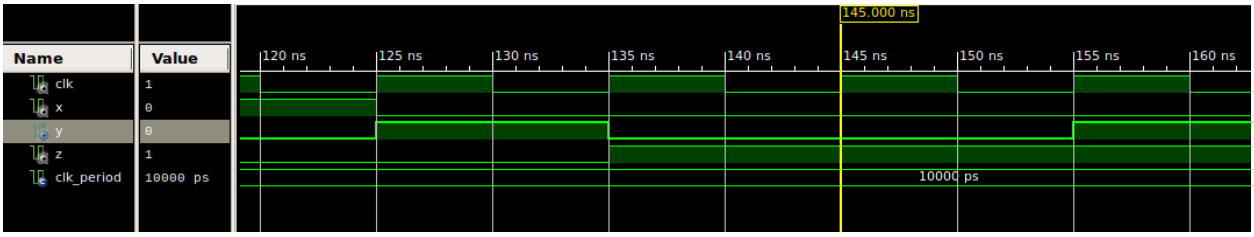
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



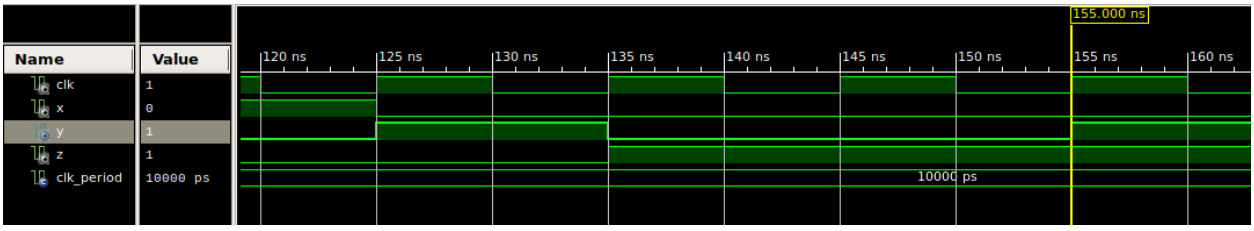
This outcome is correct because the current state is s2 and right before the clock tick X is 1 and Y is 0, therefore the next state is s3 and Z is 0.



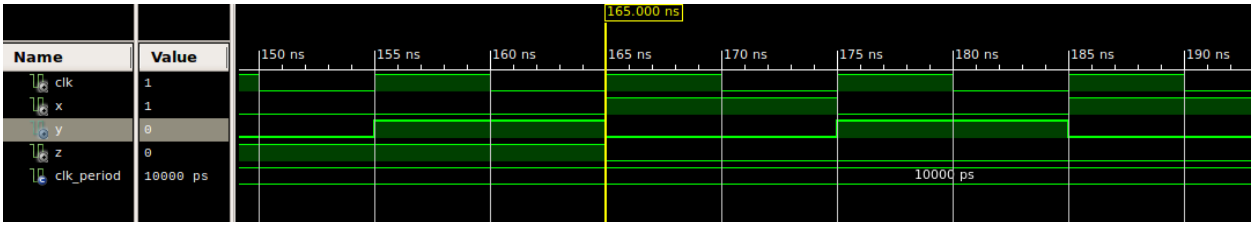
This outcome is correct because the current state is s3 and right before the clock tick X is 0 and Y is 1, therefore the next state is s0 and Z is 1.



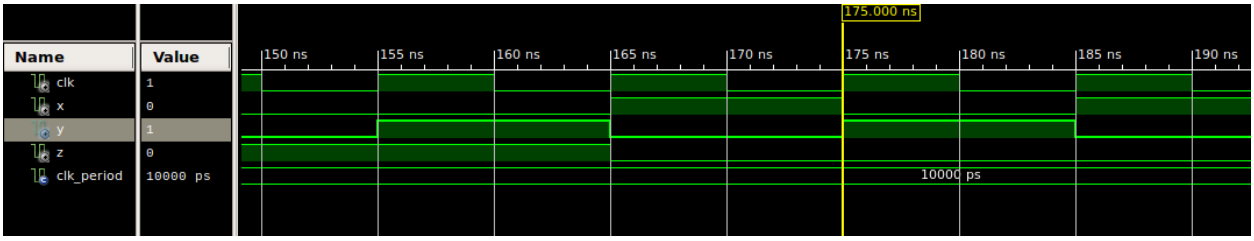
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



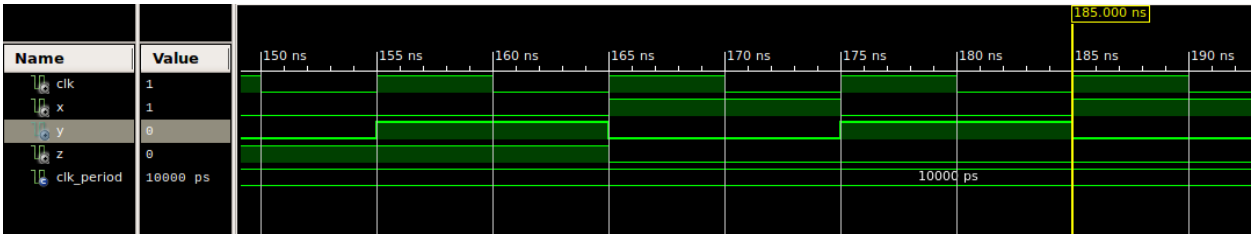
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



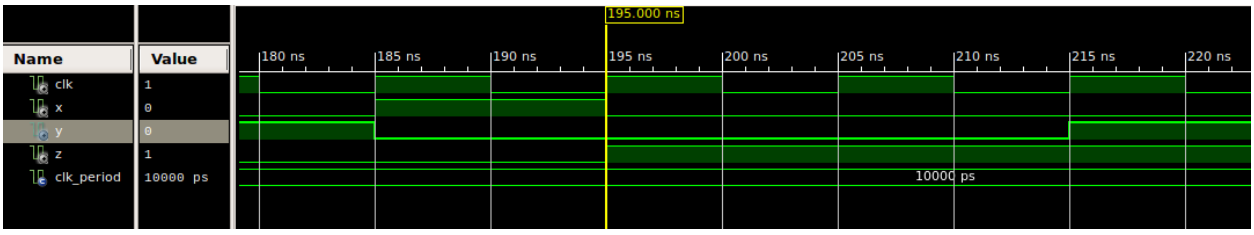
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



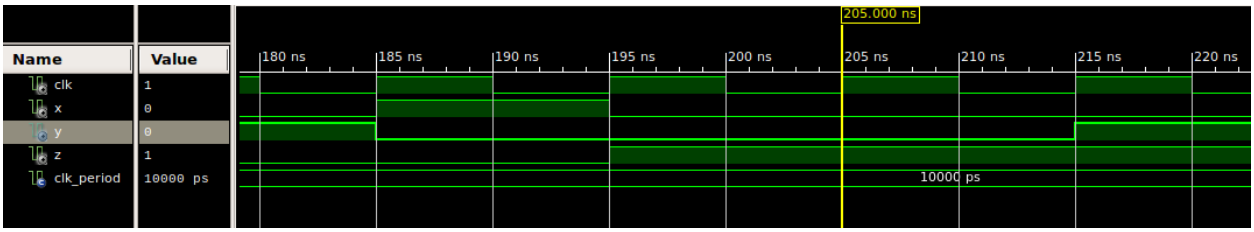
This outcome is correct because the current state is s1 and right before the clock tick X is 1 and Y is 0, therefore the next state is s2 and Z is 0.



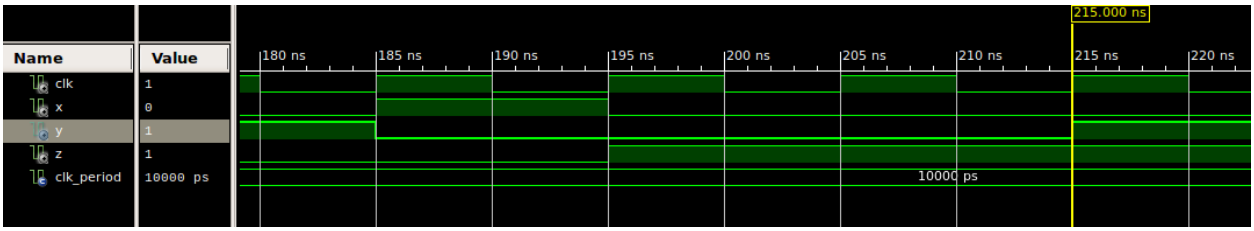
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



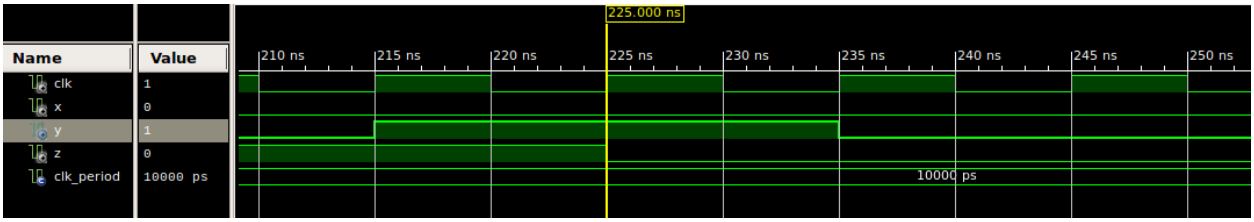
This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.



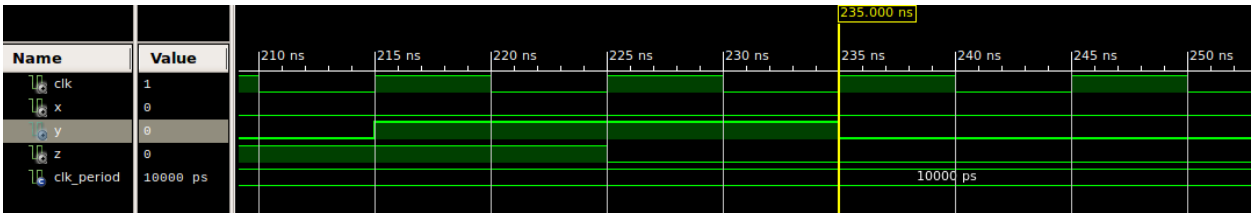
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



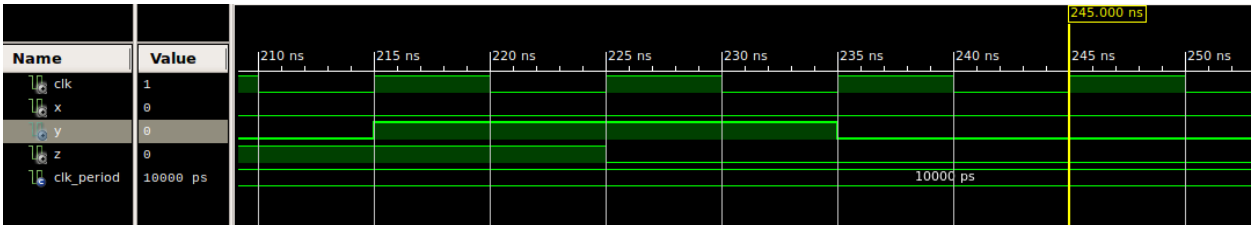
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



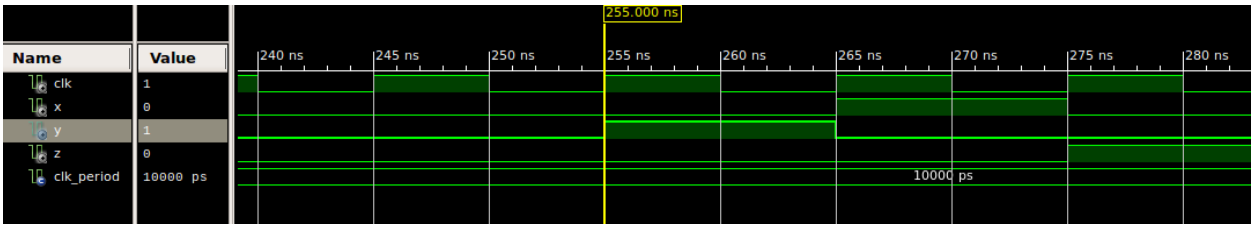
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



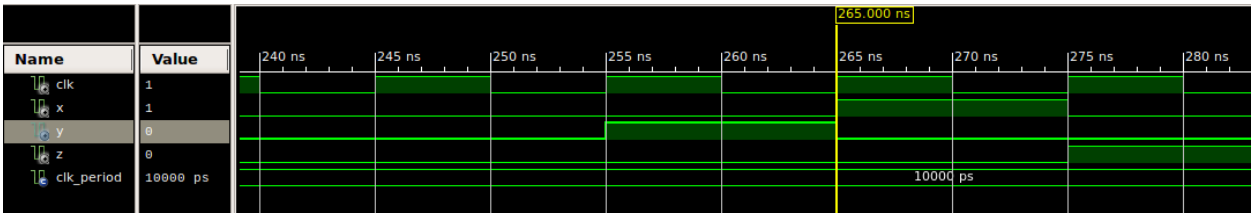
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



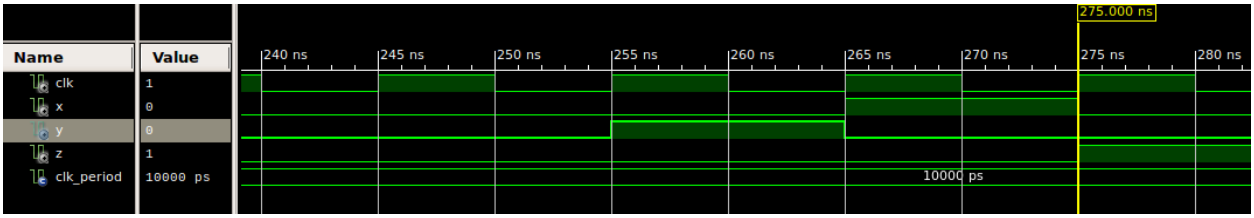
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 0, therefore the next state is s2 and Z is 0.



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.

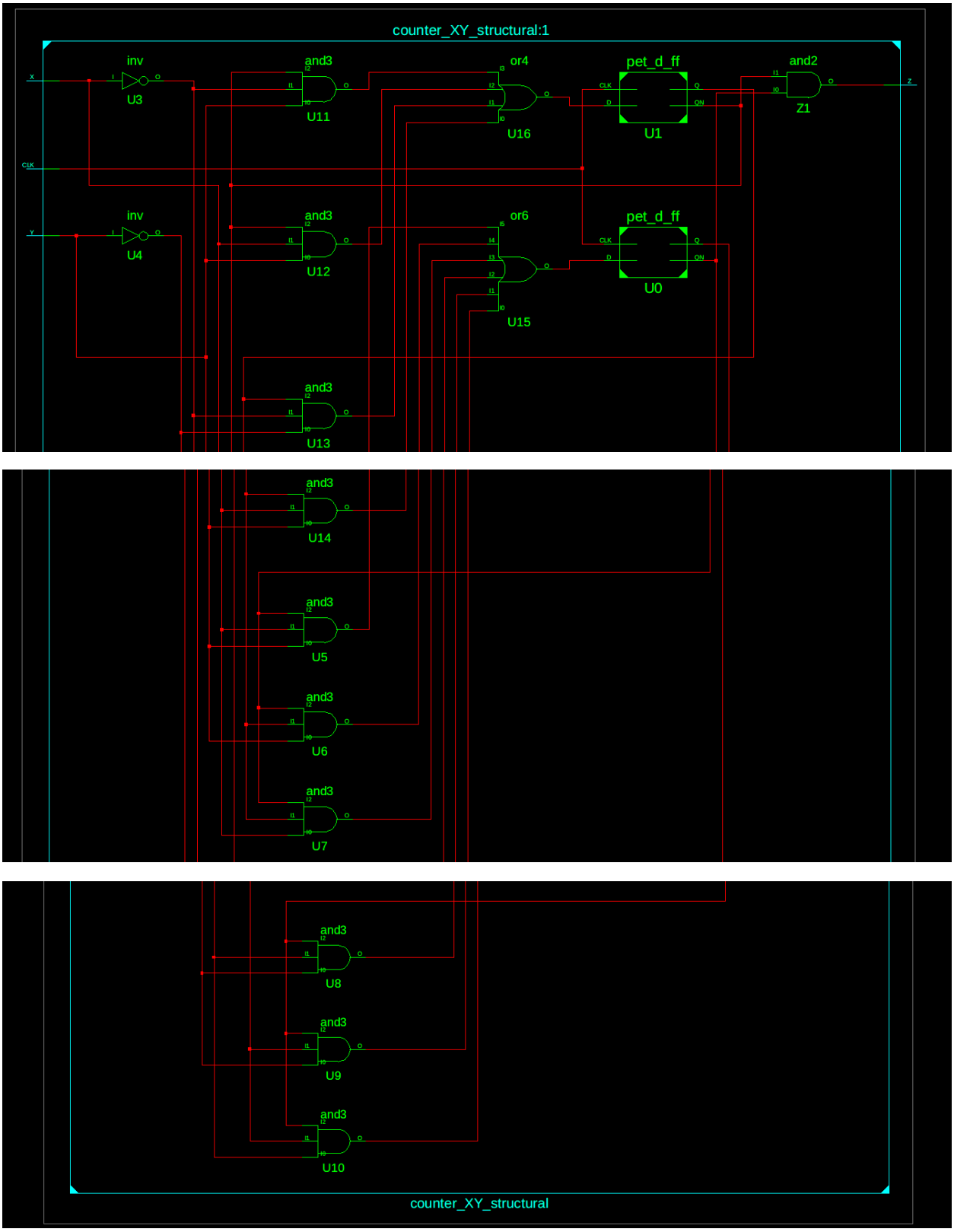
	XY				
S	**00**	**01**	**10**	**11**	**Z**
S0	S0	S1	S1	S2	1
S1	S1	S2	S2	S3	0
S2	S2	S3	S3	S0	0
S3	S3	S0	S0	S1	0
	S*				

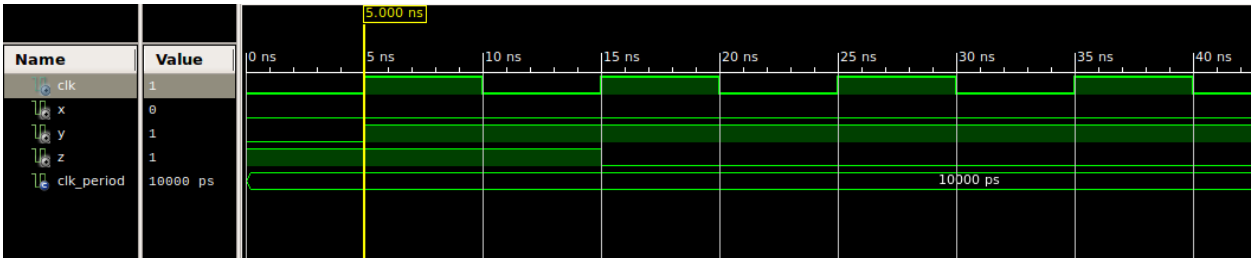
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity counter_XY_behavioral is
    Port ( CLK : in  STD_LOGIC;
           X : in  STD_LOGIC;
           Y : in  STD_LOGIC;
           Z : out  STD_LOGIC);
end counter_XY_behavioral;

architecture Behavioral of counter_XY_behavioral is
   type state_type is (s0,s1,s2,s3);
   signal state: state_type := s0;
   signal X0,Y0: std_logic := '0';
begin
   process(CLK)
   begin
      if rising_edge(CLK) then
         case state is
            when s0 =>
               if X0='0' and Y0='0' then
                  state <= s0;
               elsif X0='0' and Y0='1' then
                  state <= s1;
               elsif X0='1' and Y0='0' then
                  state <= s1;
               elsif X0='1' and Y0='1' then
                  state <= s2;
               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
```

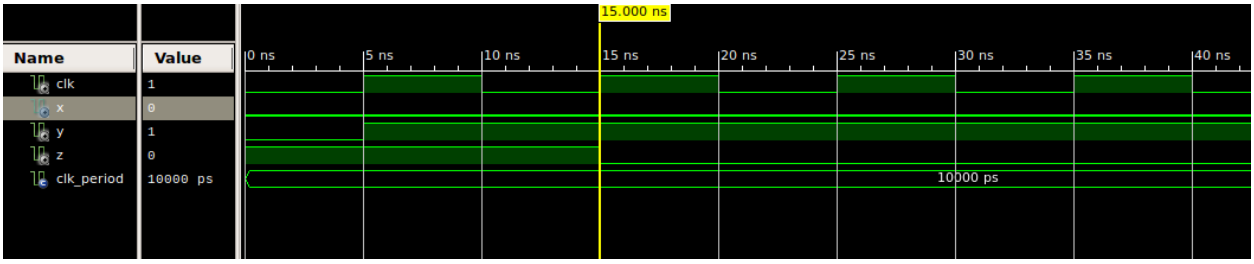
```vhdl
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY TB_behavioral IS
END TB_behavioral;

ARCHITECTURE behavior OF TB_behavioral IS
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT counter_XY_behavioral
    PORT(
         CLK : IN  std_logic;
         X : IN  std_logic;
         Y : IN  std_logic;
         Z : OUT  std_logic
        );
    END COMPONENT;

   --Inputs
   signal CLK : std_logic := '0';
   signal X : std_logic := '0';
   signal Y : std_logic := '0';
   --Outputs
   signal Z : std_logic;
   -- Clock period definitions
   constant CLK_period : time := 10 ns;

```

```
BEGIN
   uut: counter_XY_behavioral PORT MAP (
          CLK => CLK,
          X => X,
          Y => Y,
          Z => Z
        );

   -- Clock process definitions
   CLK_process :process
   begin
      CLK <= '0';
      wait for CLK_period/2;
      CLK <= '1';
      wait for CLK_period/2;
   end process;


   -- Stimulus process
   stim_proc: process
   begin
      wait for 5 ns;
      -- Test case #1
      X<='0'; Y<='1'; wait for CLK_period*4; X<='0'; Y<='0';

      -- Test case #2
      X<='1'; Y<='0'; wait for CLK_period*4; X<='0'; Y<='0'; wait for CLK_period;


       -- Test case #3
       X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
       X<='1'; Y<='0'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
       X<='0'; Y<='0'; wait for CLK_period*2;

       -- Test case #4
       X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
       X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
       X<='0'; Y<='0'; wait for CLK_period*2;

       -- Test case #5
       X<='0'; Y<='1'; wait for CLK_period; X<='0'; Y<='1'; wait for CLK_period;
       X<='0'; Y<='0'; wait for CLK_period*2;
       X<='0'; Y<='1'; wait for CLK_period; X<='1'; Y<='0'; wait for CLK_period;
       X<='0'; Y<='0'; wait for CLK_period*2;

       wait;
    end process;

END;
```

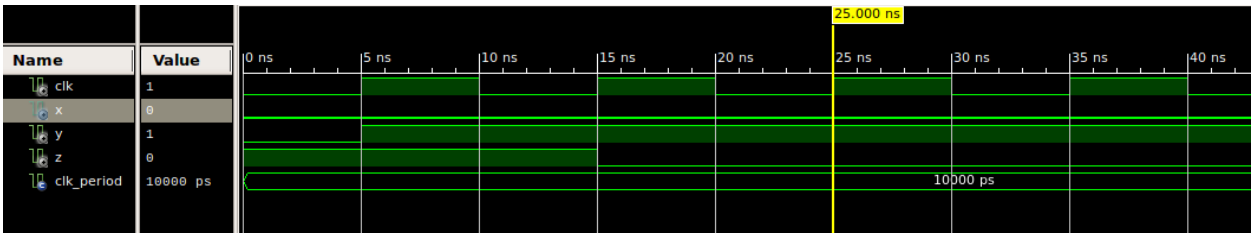

counter_XY_structural:1
inv
U3
and3
U11
or4
U16
pet_d_ff
U1
and2
Z1
X
CLK
Y
Z
inv
U4
and3
U12
or6
U15
pet_d_ff
U0
and3
U13
and3
U14
and3
U5
and3
U6
and3
U7
and3
U8
and3
U9
and3
U10
counter_XY_structural



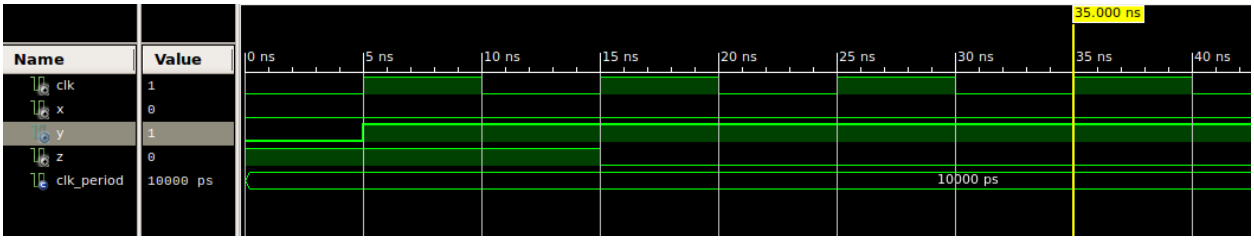
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



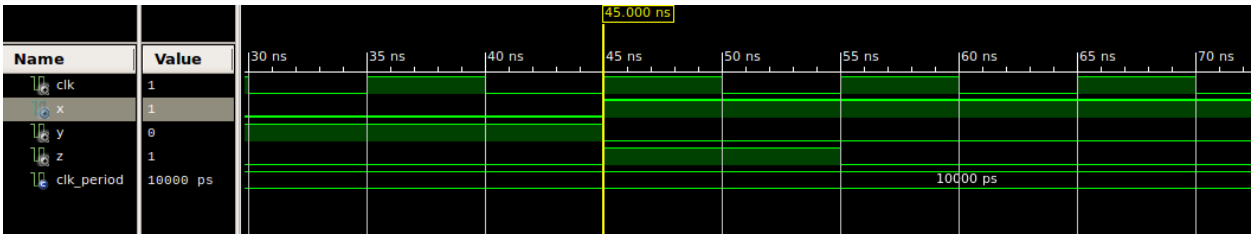
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



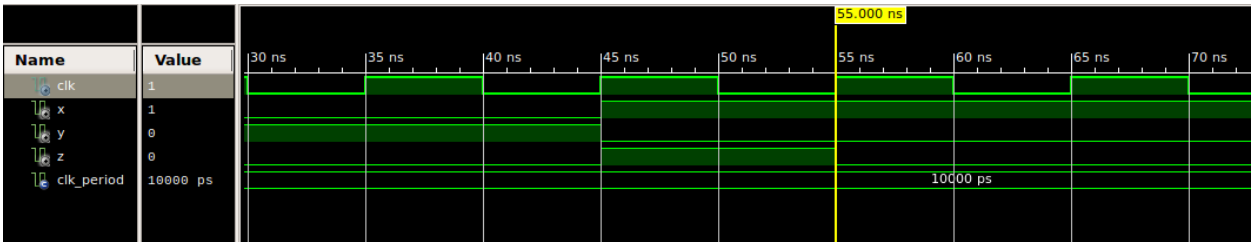
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



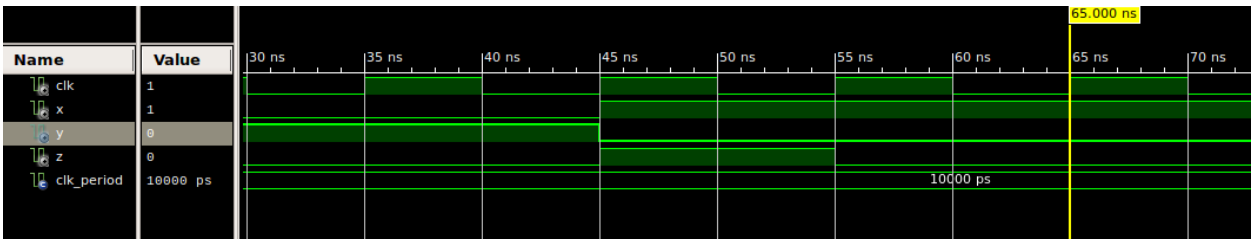
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



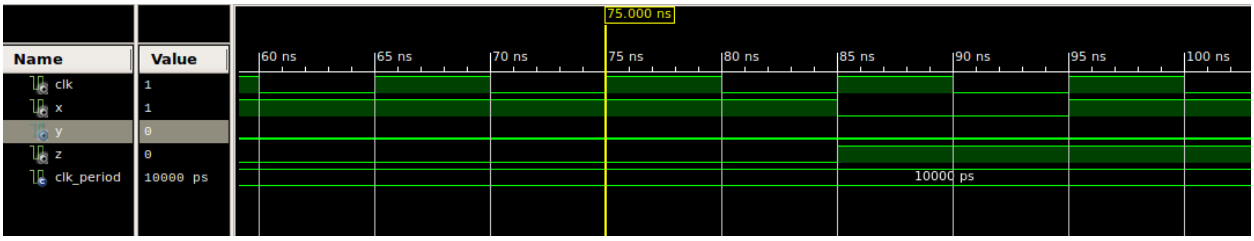
This outcome is correct because the current state is s3 and right before the clock tick X is 0 and Y is 1, therefore the next state is s0 and Z is 1.



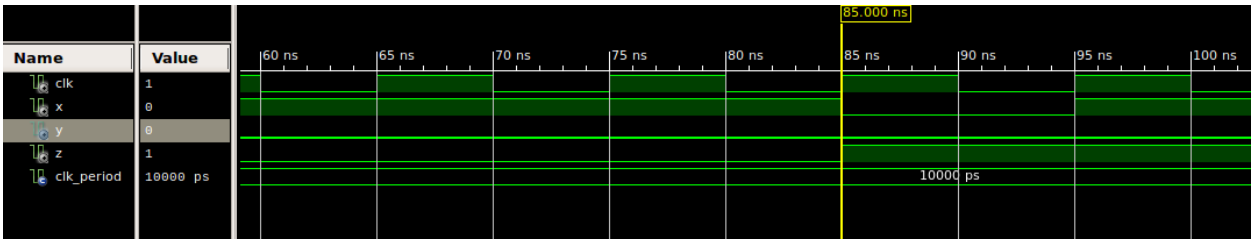
This outcome is correct because the current state is s0 and right before the clock tick X is 1 and Y is 0, therefore the next state is s1 and Z is 0.



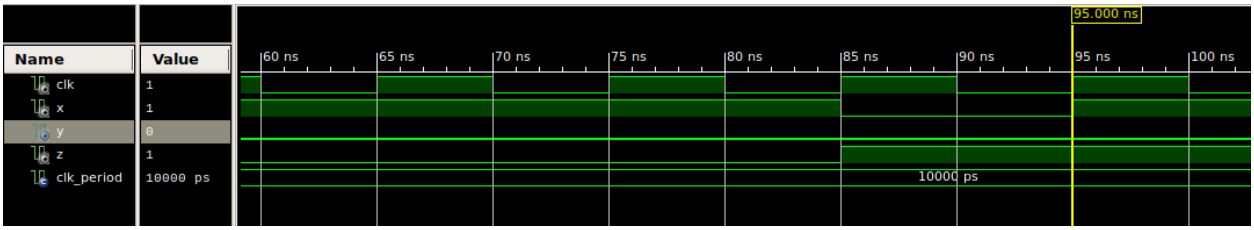
This outcome is correct because the current state is s1 and right before the clock tick X is 1 and Y is 0, therefore the next state is s2 and Z is 0.



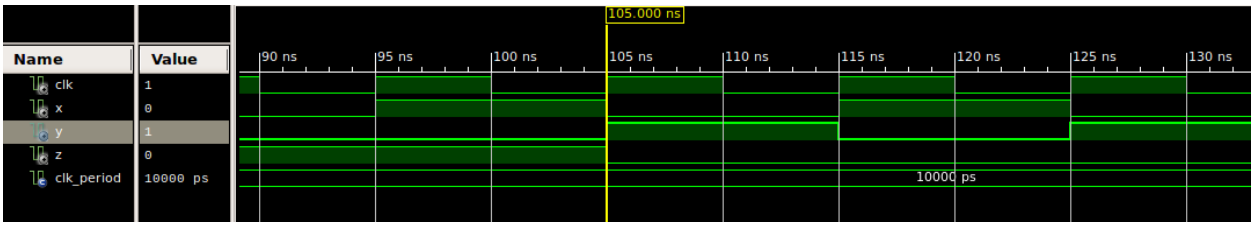
This outcome is correct because the current state is s2 and right before the clock tick X is 1 and Y is 0, therefore the next state is s3 and Z is 0.



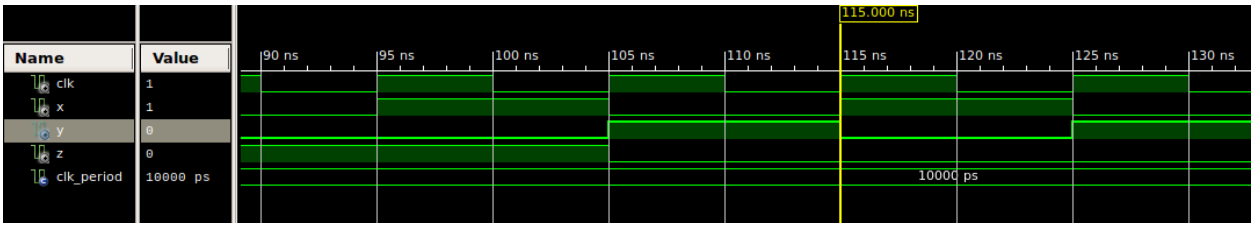
This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.



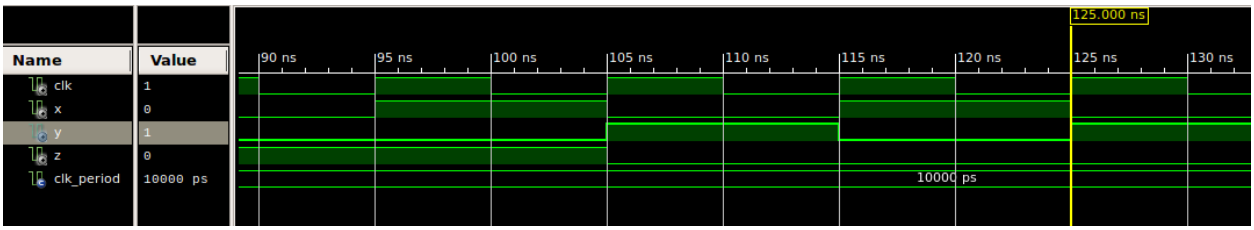
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



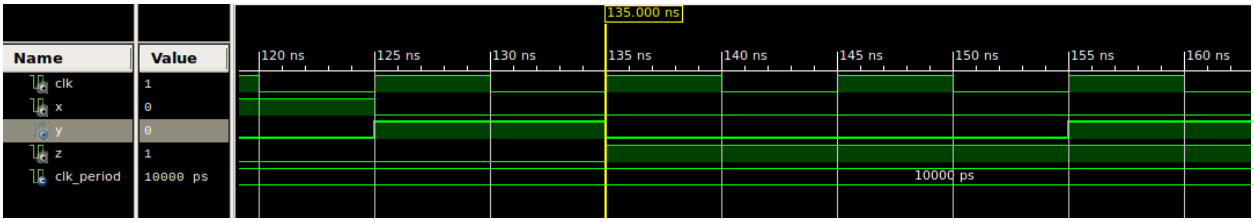
This outcome is correct because the current state is s0 and right before the clock tick X is 1 and Y is 0, therefore the next state is s1 and Z is 0.



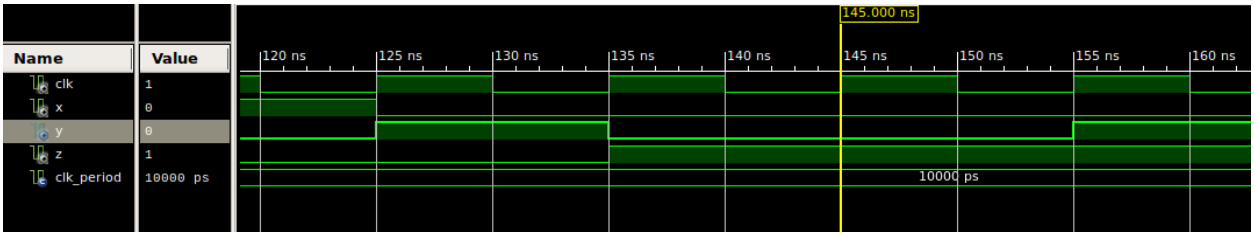
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



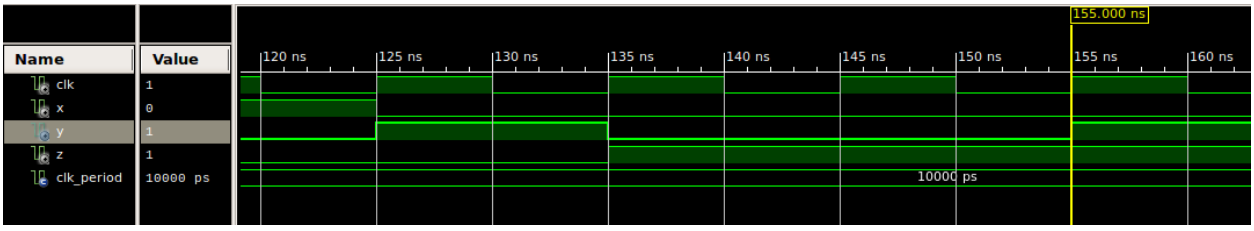
This outcome is correct because the current state is s2 and right before the clock tick X is 1 and Y is 0, therefore the next state is s3 and Z is 0.



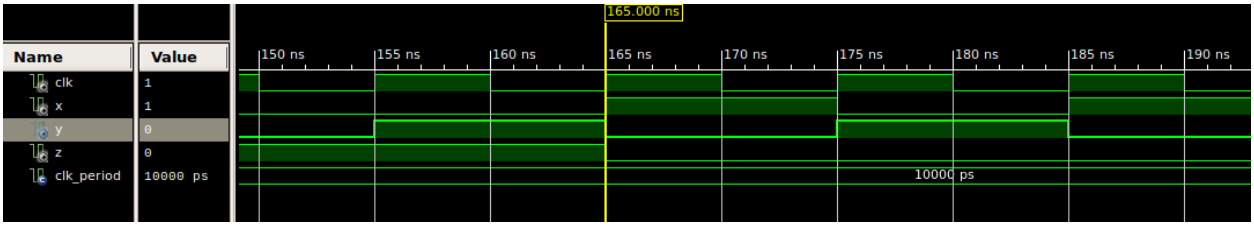
This outcome is correct because the current state is s3 and right before the clock tick X is 0 and Y is 1, therefore the next state is s0 and Z is 1.



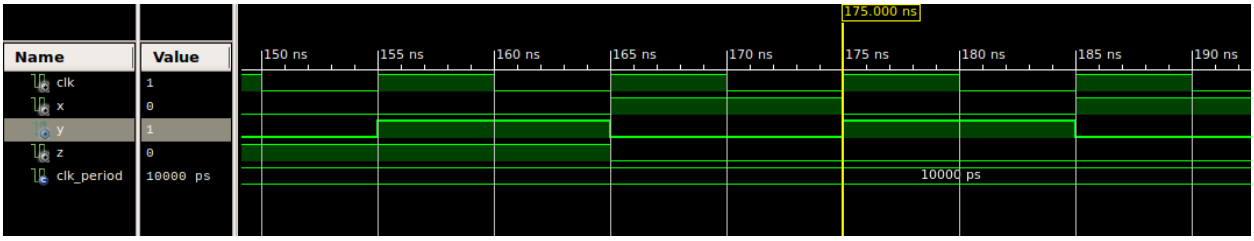
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



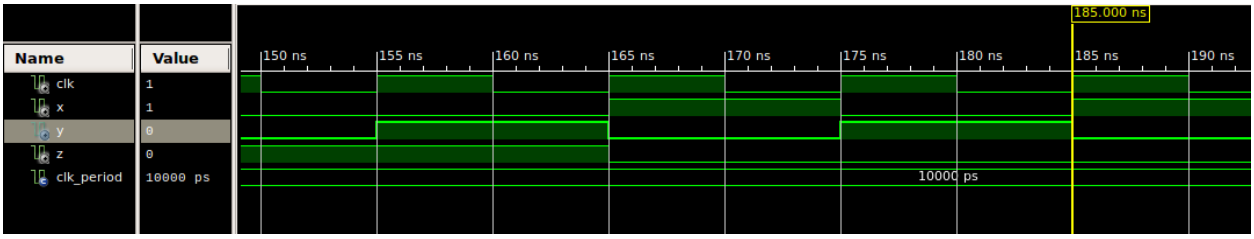
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



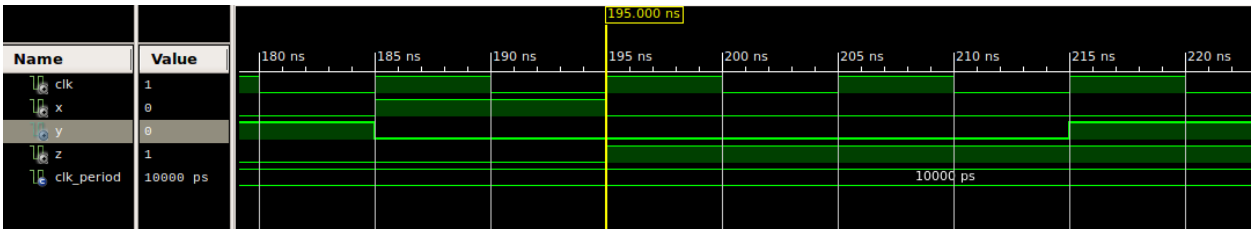
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



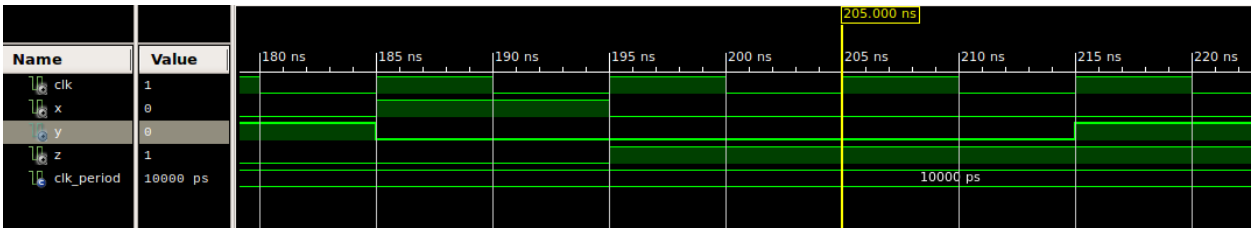
This outcome is correct because the current state is s1 and right before the clock tick X is 1 and Y is 0, therefore the next state is s2 and Z is 0.



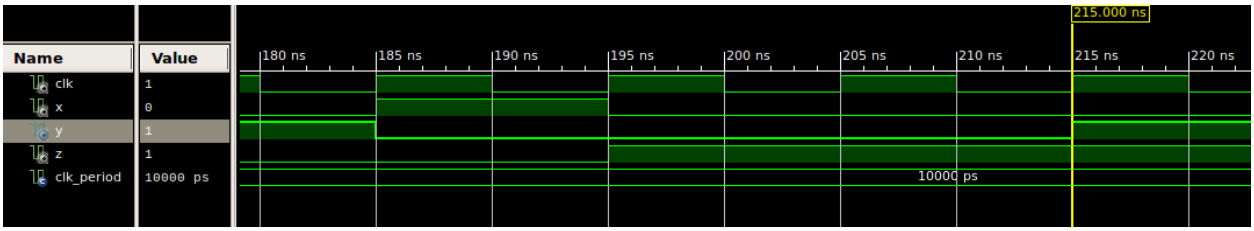
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 1, therefore the next state is s3 and Z is 0.



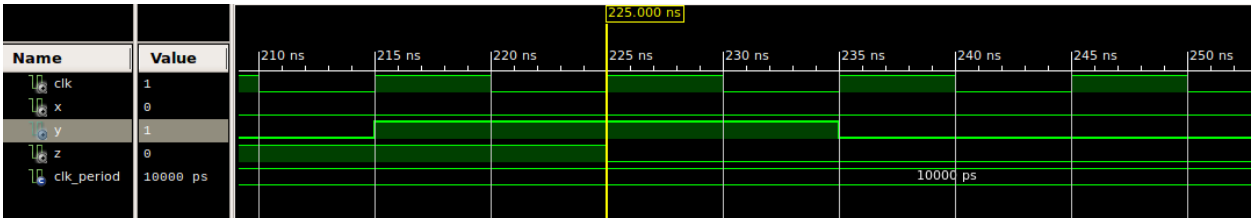
This outcome is correct because the current state is s3 and right before the clock tick X is 1 and Y is 0, therefore the next state is s0 and Z is 1.



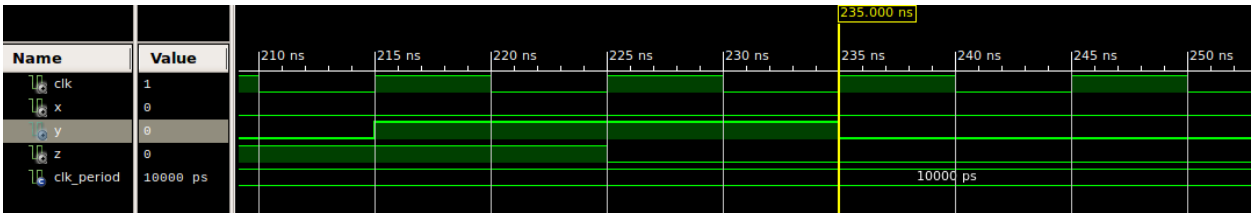
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



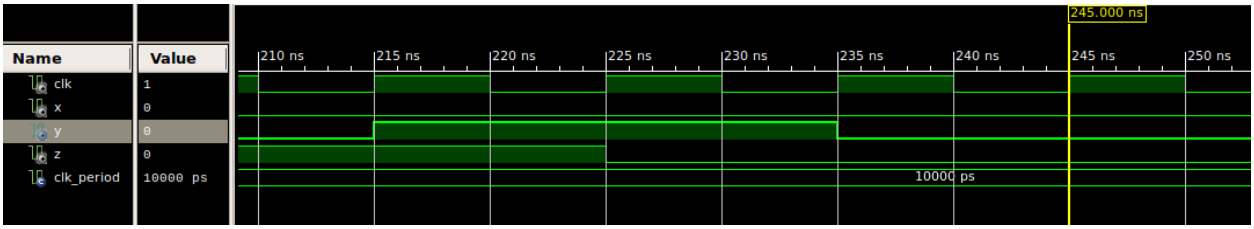
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 0, therefore the next state is s0 and Z is 1.



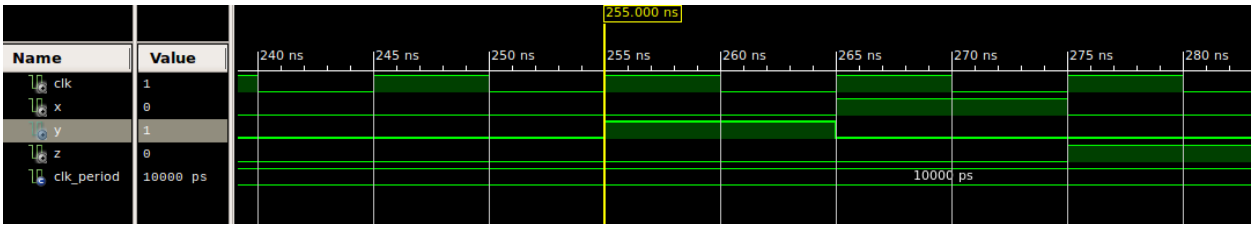
This outcome is correct because the current state is s0 and right before the clock tick X is 0 and Y is 1, therefore the next state is s1 and Z is 0.



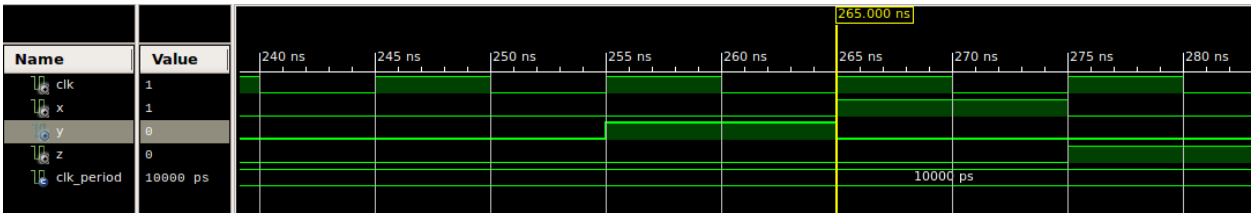
This outcome is correct because the current state is s1 and right before the clock tick X is 0 and Y is 1, therefore the next state is s2 and Z is 0.



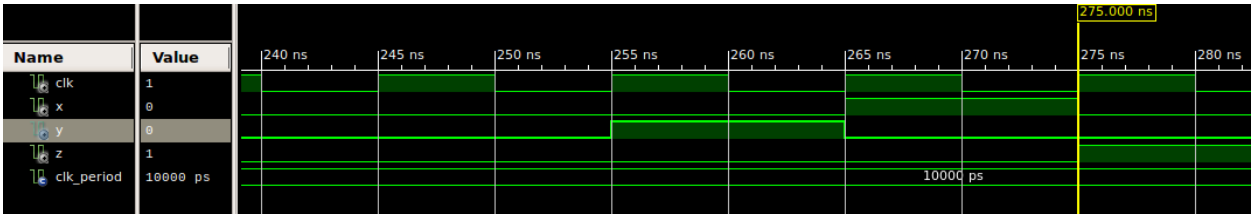
This outcome is correct because the current state is s2 and right before the clock tick X is 0 and Y is 0, therefore the next state is s2 and Z is 0.



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.