Advanced Digital Circuit Design - Synchronous Sequential Logic

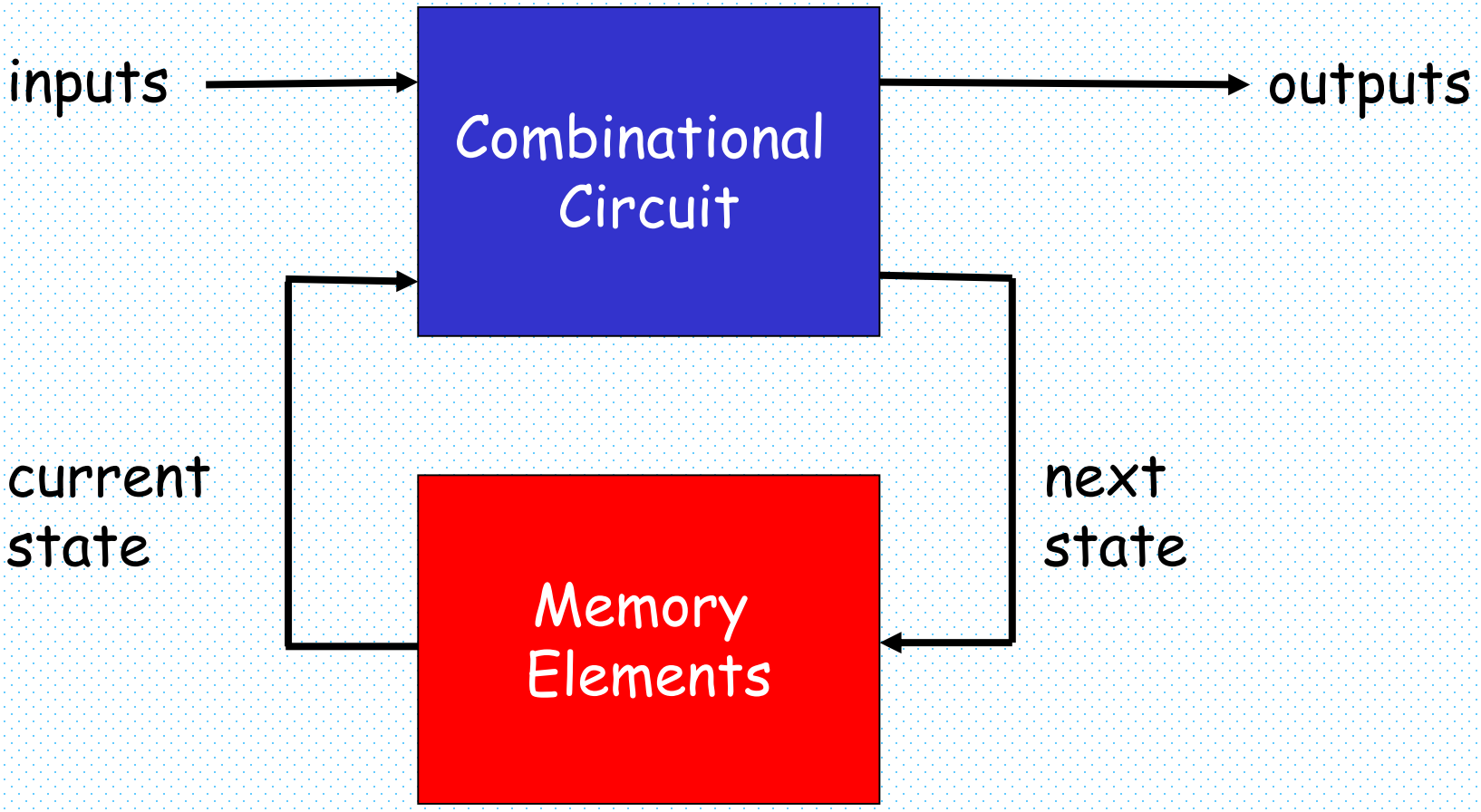
Prof. Dr. Berna Örs Yalçın

Istanbul Technical University
Faculty of Electrical and Electronics Engineering
Department of Electronics and Communication Engineering
siddika.ors@itu.edu.tr

Sequential Logic

- Digital circuits we have learned, so far, have been combinational
  - no memory,
  - outputs are entirely defined by the "current" inputs
- However, many digital systems encountered everyday life are sequential (i.e. they have memory)
  - the memory elements remember past inputs
  - outputs of sequential circuits are not only dependent on the current input but also the state of the memory elements.

Sequential Circuits Model

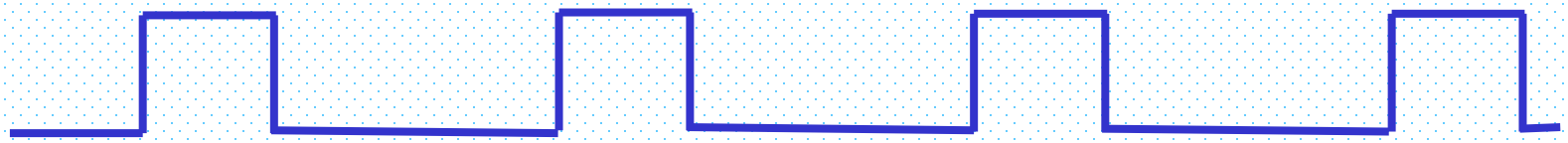


current state is a function of past inputs and initial state

Classification 1/2

- Two types of sequential circuits

1. Synchronous
   - Signals affect the memory elements at discrete instants of time.
   - Discrete instants of time requires synchronization.
   - Synchronization is usually achieved through the use of a common clock.
   - A "clock generator" is a device that generates a periodic train of pulses.

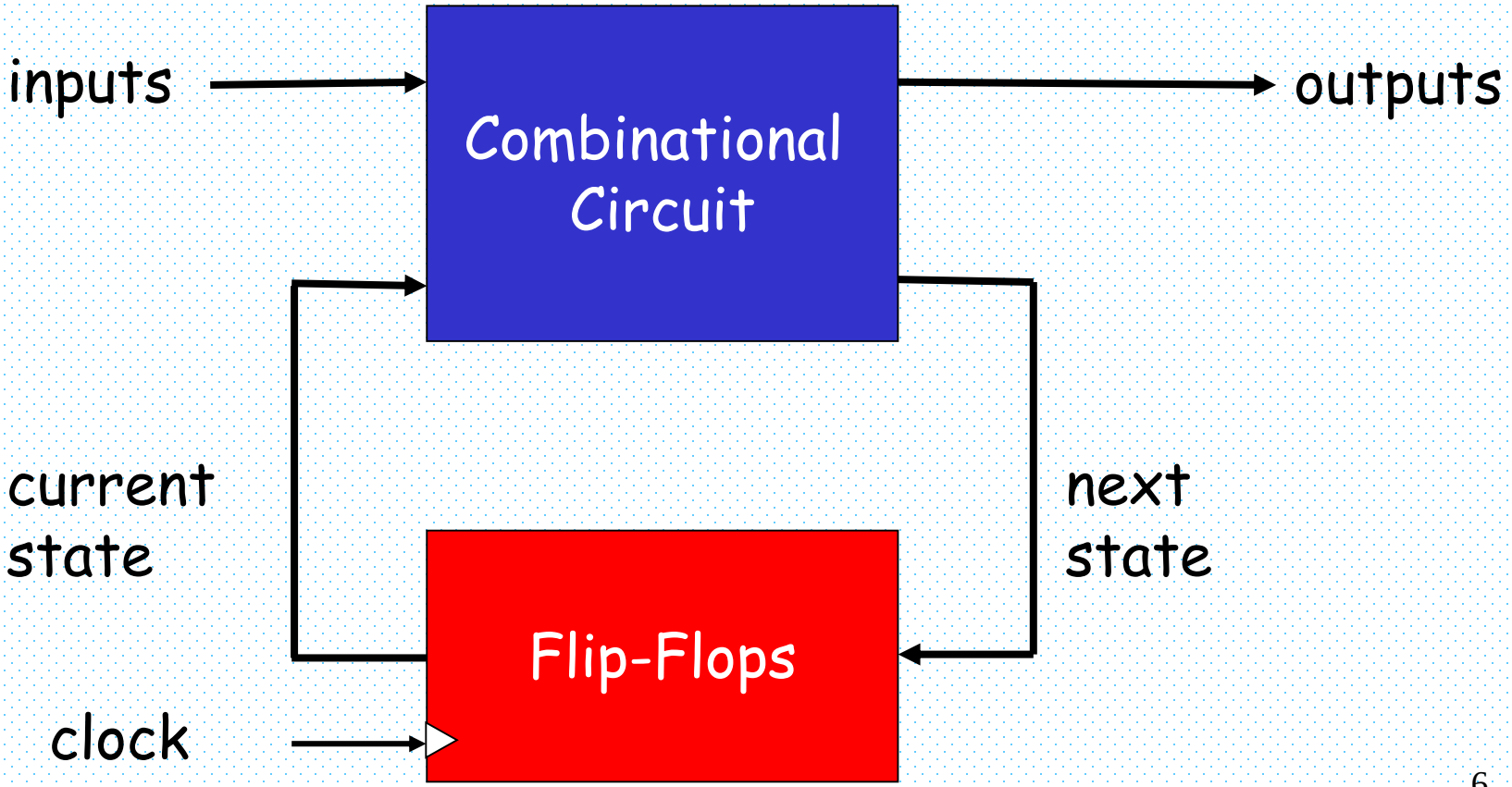


Classification 2/2

1. Synchronous
   - The state of the memory elements are updated with the arrival of each pulse
   - This type of logical circuit is also known as clocked sequential circuits.
2. Asynchronous
   - No clock
   - behavior of an asynchronous sequential circuits depends upon the input signals at any instant of time and the order in which the inputs change.
   - Memory elements in asynchronous circuits are regarded as time-delay elements

Clocked Sequential Circuits

- Memory elements are flip-flops which are logic devices capable of storing one bit of information each.



Clocked Sequential Circuits

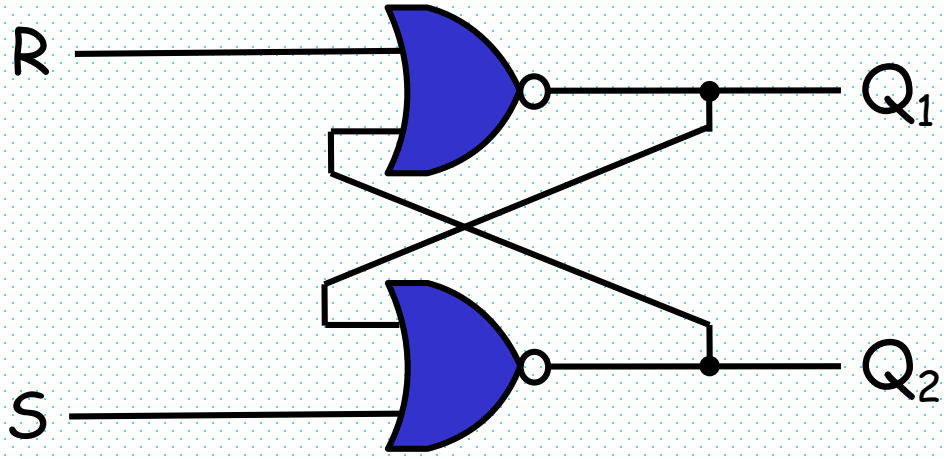
- The outputs of a clocked sequential circuit can come from the combinational circuit, from the outputs of the flip-flops or both.
- The state of the flip-flops can change only during a clock pulse transition
  - i.e. low-to-high and high-to-low
  - clock edge
- When the clock maintains its value, the flip-flop output does not change
- The transition from one state to the next occurs at the clock edge.

Latches

- The most basic types of memory elements are not flip-flops, but latches.
- A latch is a memory device that can maintain a binary state indefinitely.
- Latches are, in fact, asynchronous devices and they usually do not require a clock to operate.
- Therefore, they are not directly used in clocked synchronous sequential circuits.
- They are rather used to construct flip-flops.

SR-Latch

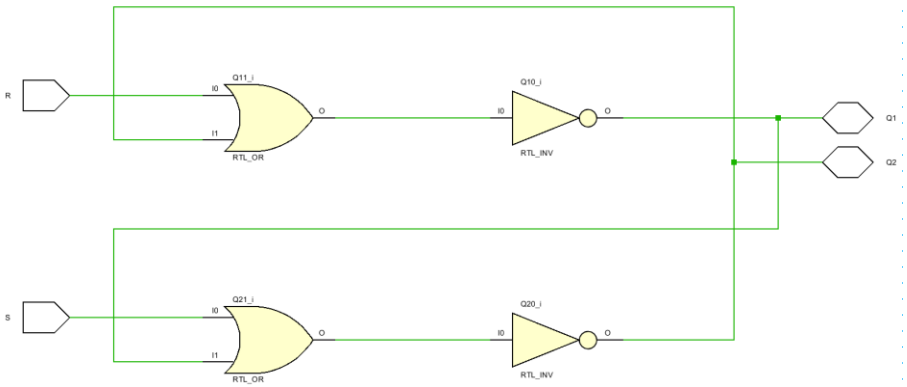
- made of cross-coupled NOR (or NAND) gates



S	R	Q_1	Q_2	
1	0	1	0	
0	0	1	0	$Q_2 = Q_1'$
0	1	0	1	
0	0	0	1	
1	1	0	0	Undefined

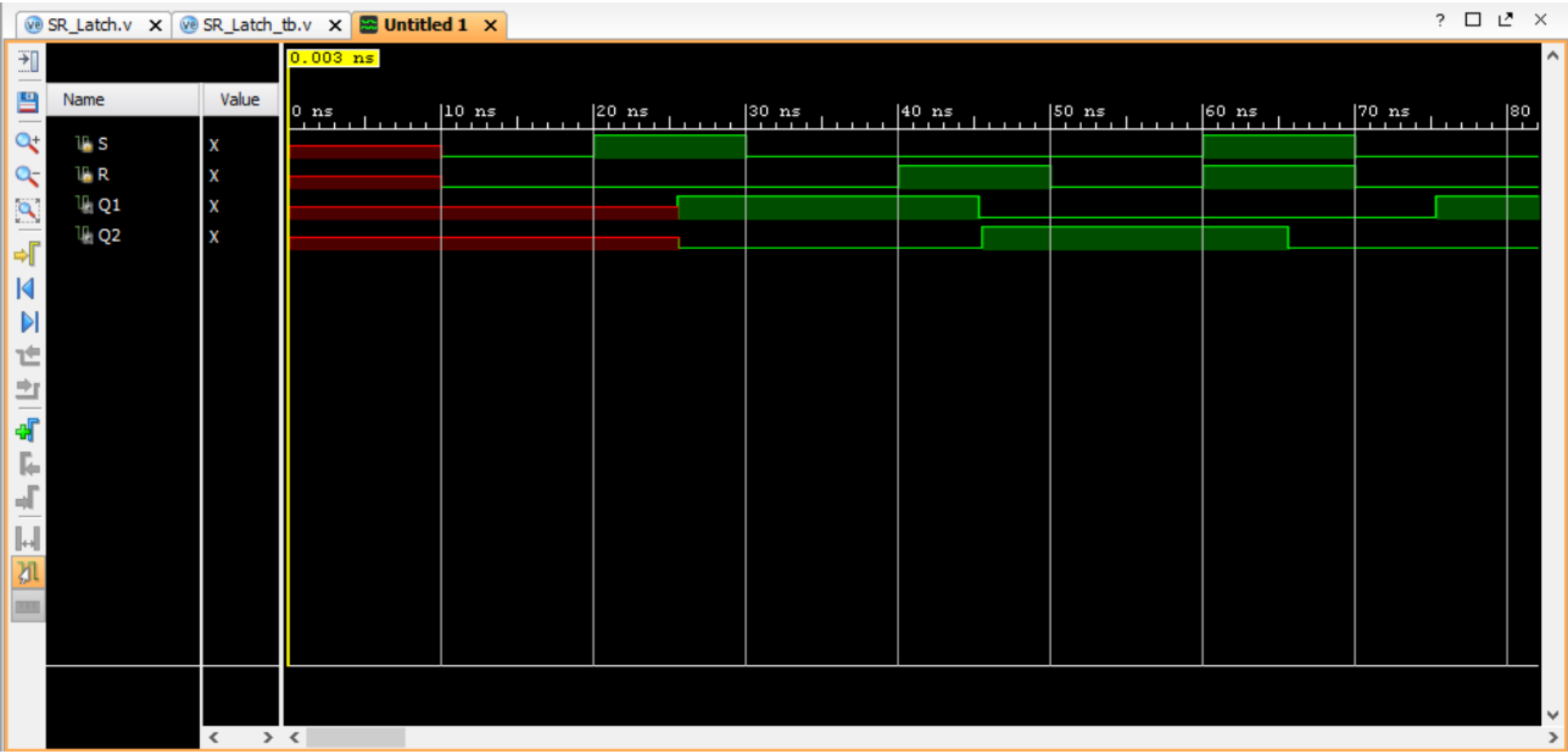
VHDL Entity of SR Latch and Testbench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity SR_Latch_NOR is
    Port ( S : in STD_LOGIC;
           R : in STD_LOGIC;
           Q1 : inout STD_LOGIC;
           Q2 : inout STD_LOGIC);
end SR_Latch_NOR;
architecture Behavioral of SR_Latch_NOR is
begin
    Q1 <= not (R or Q2);
    Q2 <= not (S or Q1);
end Behavioral;
```



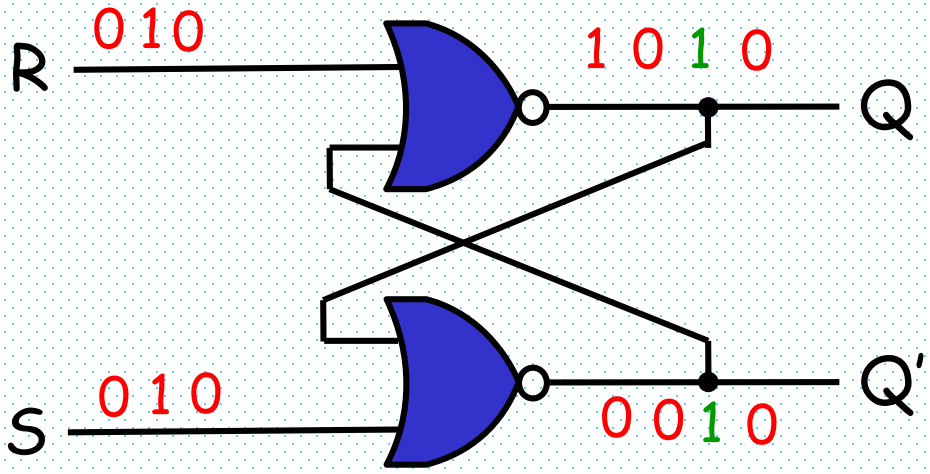

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity SR_Latch_NOR_tb is
end SR_Latch_NOR_tb;
architecture Behavioral of SR_Latch_NOR_tb is
    component SR_Latch_NOR is
        Port ( S : in STD_LOGIC;
               R : in STD_LOGIC;
               Q1 : inout STD_LOGIC;
               Q2 : inout STD_LOGIC);
    end component;
    signal S,R,Q1,Q2 : STD_LOGIC;
begin
    DUT: SR_Latch_NOR Port map(S,R,Q1,Q2);
    process    begin
        R<='0'; S<='0';       wait for 5 ns;
        R<='0'; S<='1';       wait for 5 ns;
        R<='0'; S<='0';       wait for 5 ns;
        R<='1'; S<='0';       wait for 5 ns;
        R<='0'; S<='0';       wait for 5 ns;
        R<='1'; S<='1';       wait;
    end process;
end Behavioral;
```

Simulation Waveform



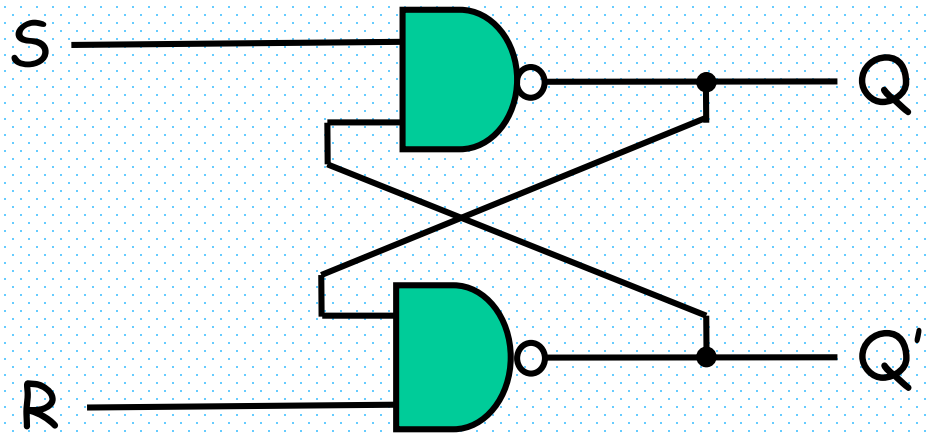
Undefined State of SR-Latch

- S = R = 1 may result in an undefined state
  - the next state is unpredictable when both S and R goes to 0 at the same time.
  - It may oscillate
  - Or the outcome state depend on which of S and R goes to 0 first.



It oscillates

SR-Latches with NAND Gates

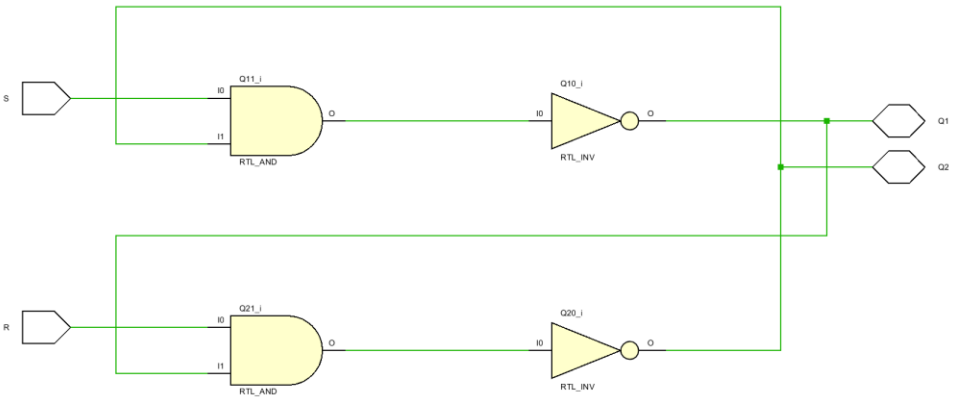


Also known as S'R'-latch

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	After S = 1, R = 0
0	1	1	0	
1	1	1	0	After S = 0, R = 1
0	0	1	1	Undefined

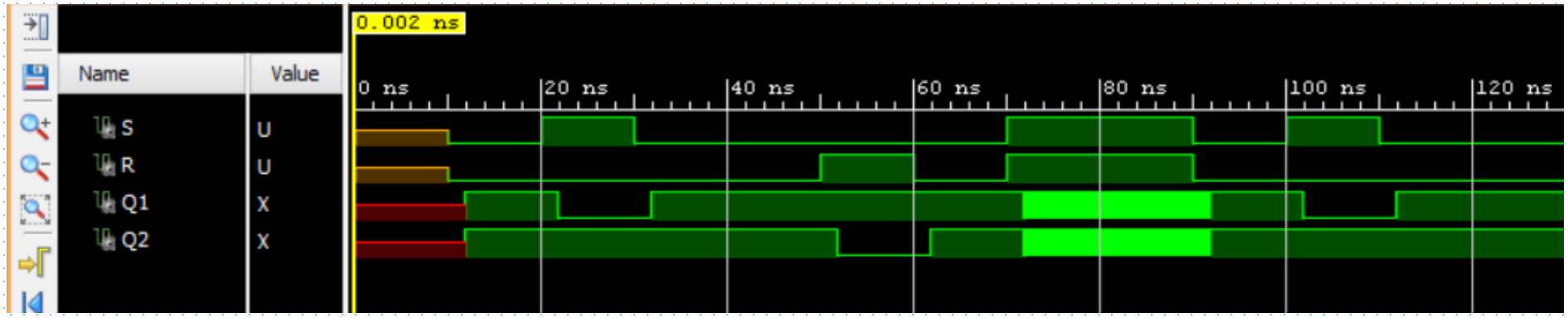
VHDL Entity of SR Latch with NAND and Testbench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity SR_Latch_NAND is
    Port ( S : in STD_LOGIC;
           R : in STD_LOGIC;
           Q1 : inout STD_LOGIC;
           Q2 : inout STD_LOGIC);
end SR_Latch_NAND;
architecture Behavioral of SR_Latch_NAND is
begin
  Q1 <= not (S and Q2);
  Q2 <= not (R and Q1);
end Behavioral;
```



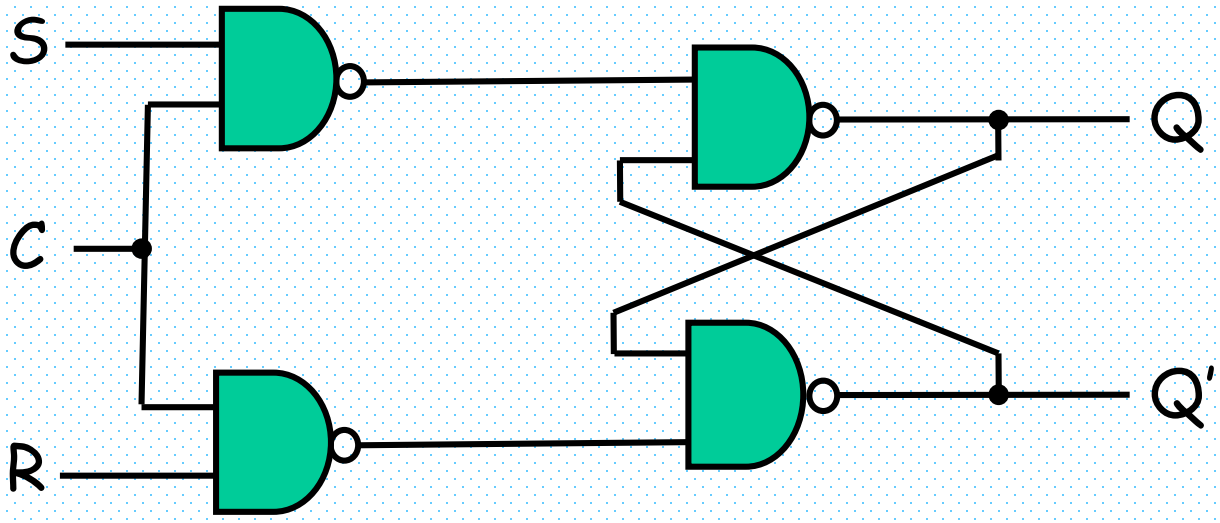

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity SR_Latch_NAND_tb is
end SR_Latch_NAND_tb;
architecture Behavioral of SR_Latch_NAND_tb is
component SR_Latch_NAND
    Port ( S : in STD_LOGIC;
           R : in STD_LOGIC;
           Q1 : inout STD_LOGIC;
           Q2 : inout STD_LOGIC);
end component;
signal S, R, Q1, Q2 : STD_LOGIC;
begin
DUT : SR_Latch_NAND Port map(S,R,Q1,Q2);
  tb: process begin
    wait for 10 ns;    R <= '0'; S <= '0';
    wait for 10 ns;    R <= '0'; S <= '1';
    wait for 10 ns;    R <= '0'; S <= '0';
    wait for 10 ns;    R <= '0'; S <= '0';
    wait for 10 ns;    R <= '1'; S <= '0';
    wait for 10 ns;    R <= '0'; S <= '0';
    wait for 10 ns;    R <= '1'; S <= '1';
  end process;
end Behavioral;
```

Simulation Waveform



SR-Latch with Control Input

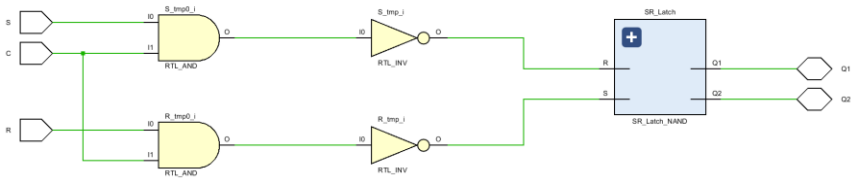
- Control inputs allow the changes at S and R to change the state of the latch.



C	S	R	Q Q'
0	X	X	No change
1	0	0	No change
1	0	1	Q = 0 Reset state
1	1	0	Q = 1 Set state
1	1	1	Indeterminate

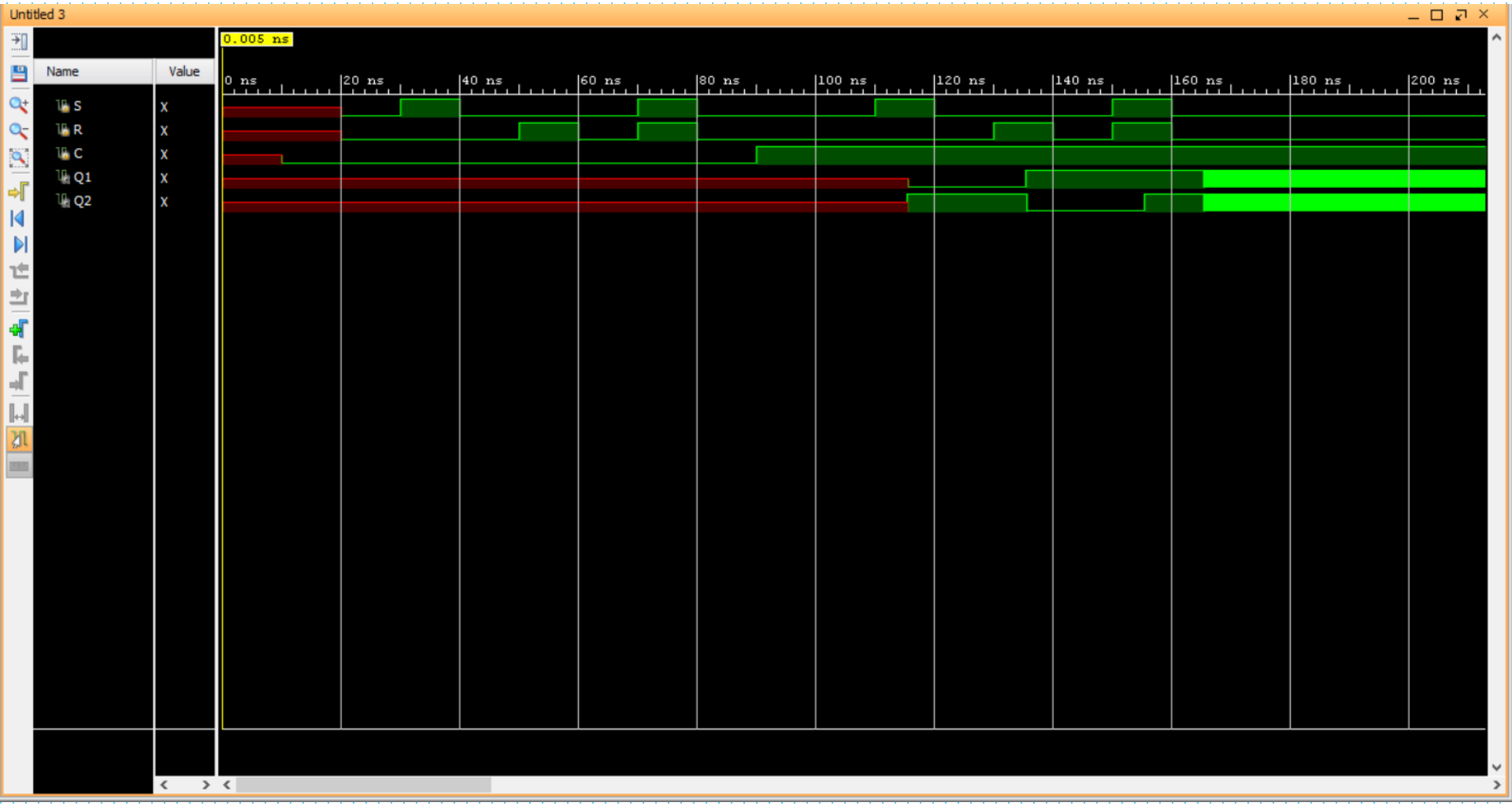
VHDL Entity of SR Latch and Testbench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity SR_Latch_NAND is
    Port(S : in STD_LOGIC;
         R : in STD_LOGIC;
         Q1 : inout STD_LOGIC;
         Q2 : inout STD_LOGIC);
end SR_Latch_NAND;
architecture Behavioral of SR_Latch_NAND is
begin
    Q1 <= not (S and Q2);
    Q2 <= not (R and Q1);
end Behavioral;
```



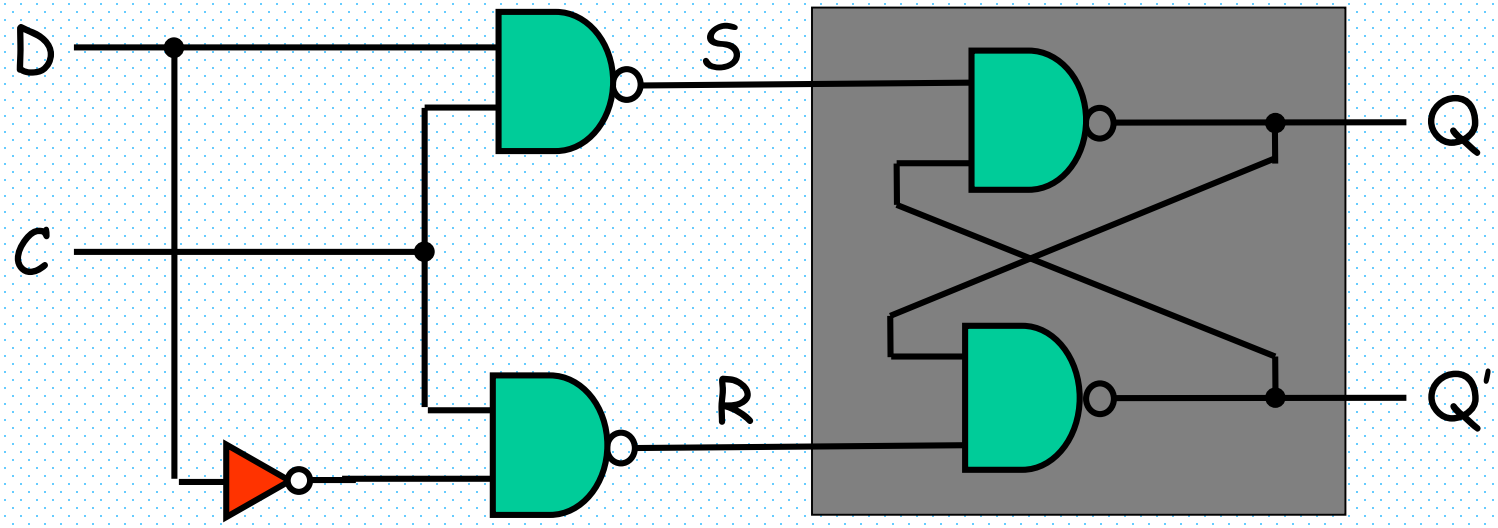

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity SR_Latch_with_Control is
    Port(S : in STD_LOGIC;
         R : in STD_LOGIC;
         C : in STD_LOGIC;
         Q1 : inout STD_LOGIC;
         Q2 : inout STD_LOGIC);
end SR_Latch_with_Control;
architecture Behavioral of SR_Latch_with_Control is
    component SR_Latch_NAND is
        Port(S : in STD_LOGIC;
             R : in STD_LOGIC;
             Q1 : inout STD_LOGIC;
             Q2 : inout STD_LOGIC);
    end component;
    signal S_tmp,R_tmp : STD_LOGIC;
begin
    S_tmp <= not (S and C);
    R_tmp <= not (R and C);
    SR_Latch: SR_Latch_NAND port
map(R_tmp,S_tmp,Q1,Q2);
end Behavioral;
```

Simulation Waveform



D-Latch

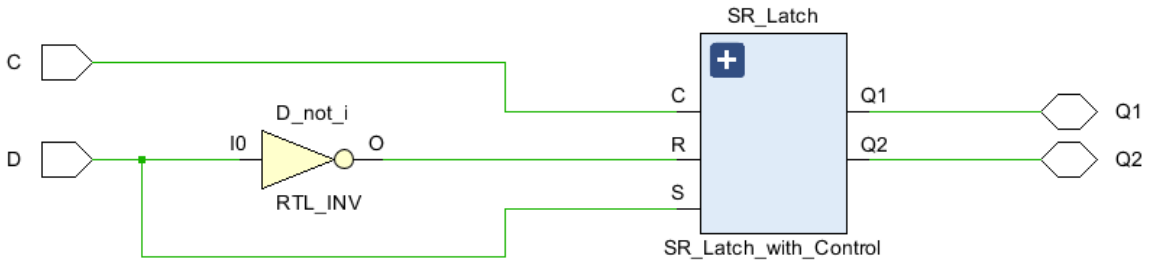
- SR latches are seldom used in practice because the indeterminate state may cause instability
- Remedy: D-latches



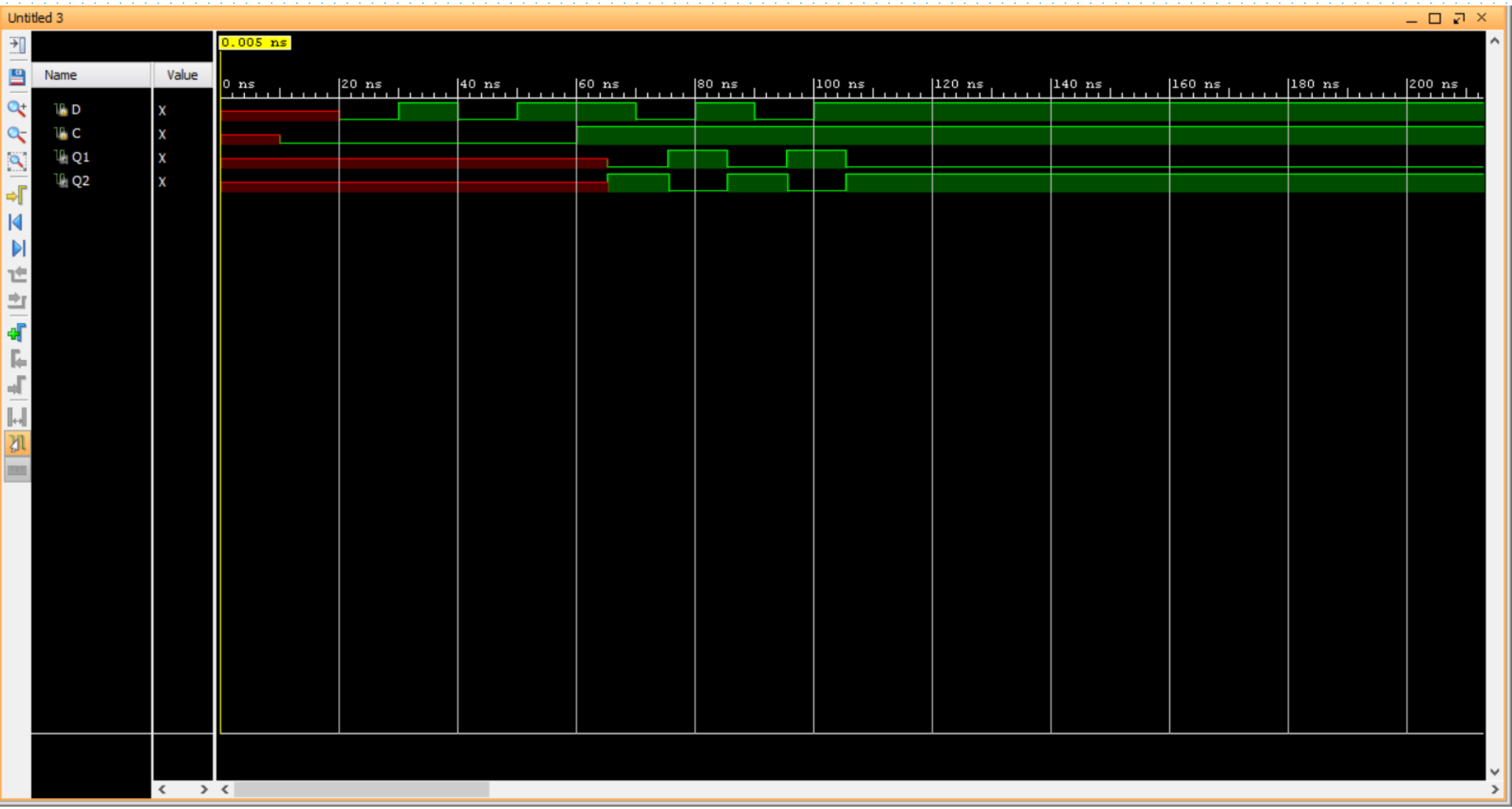
This circuit guarantees that the inputs to the SR-latch is always complement of each other when $C = 1$.

VHDL Entity of D Latch

```
entity D_Latch is
    Port(D : in STD_LOGIC;
         C : in STD_LOGIC;
         Q1 : inout STD_LOGIC;
         Q2 : inout STD_LOGIC);
end D_Latch;
architecture Behavioral of D_Latch is
    component SR_Latch_with_Control is
        Port(S : in STD_LOGIC;
             R : in STD_LOGIC;
             C : in STD_LOGIC;
             Q1 : inout STD_LOGIC;
             Q2 : inout STD_LOGIC);
    end component;
    signal D_not : STD_LOGIC;
begin
    D_not <= not D;
    SR_Latch: SR_Latch_with_Control Port
map(D,D_not,C,Q1,Q2);
end Behavioral;
```



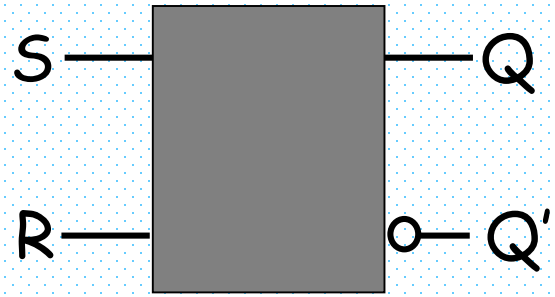
Simulation Waveform



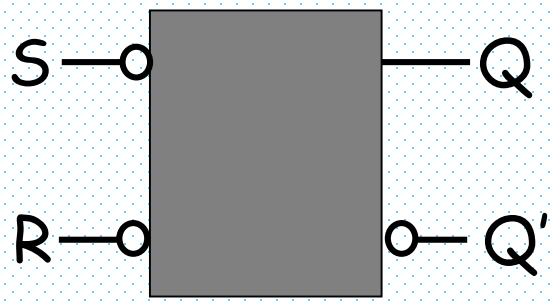
D-Latch

C	D	Next state of Q
0	X	No change
1	0	Q = 0; reset state
1	1	Q = 1; set state

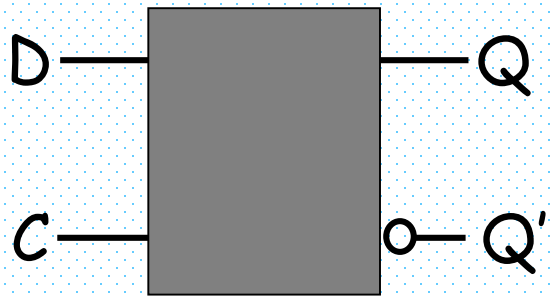
- We say that the D input is sampled when C = 1



SR-latch



S'R'-latch



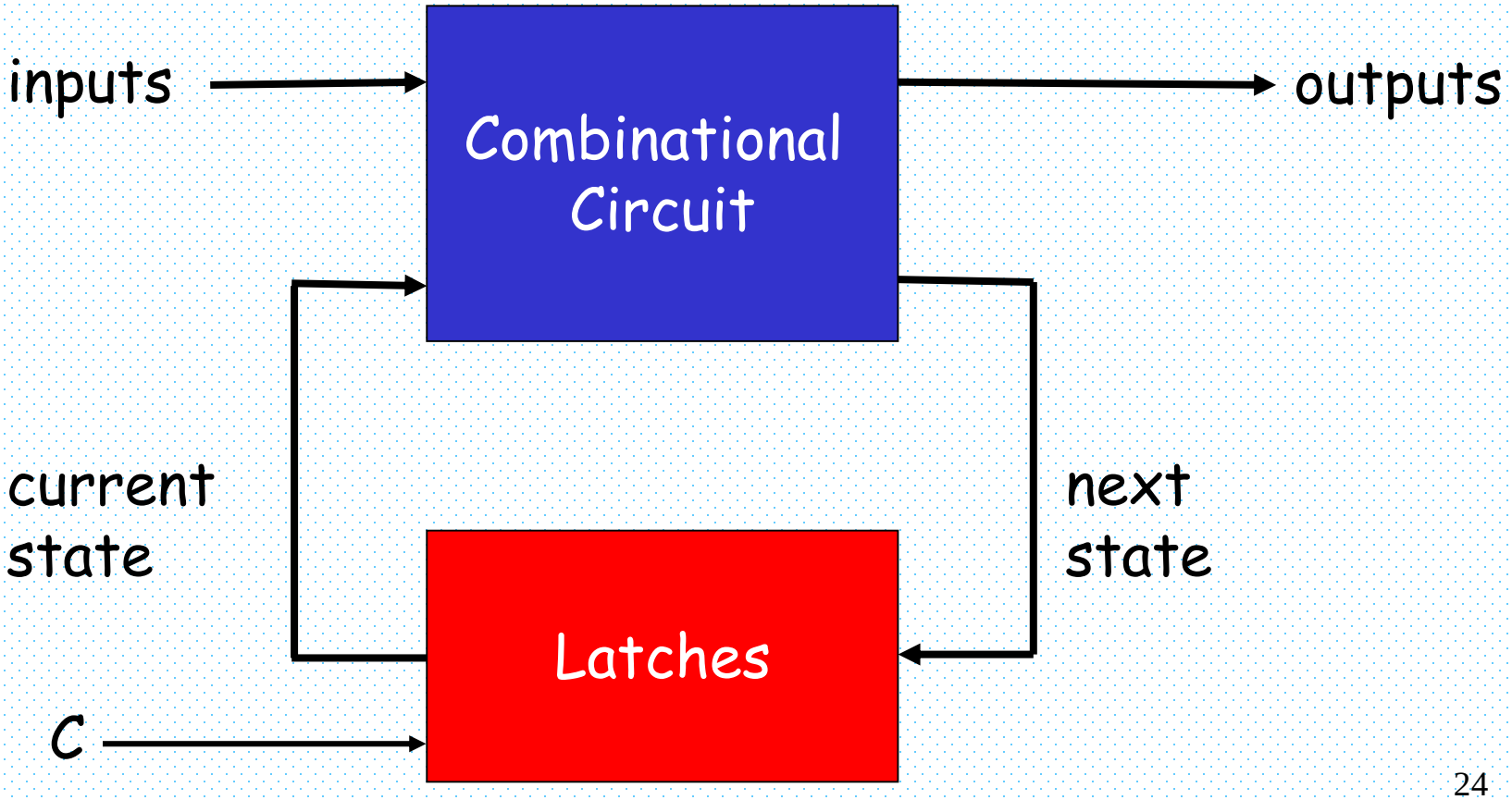
D-latch

D-Latch as a Storage Unit

- D-latches can be used as temporary storage
- The input of D-latch is transferred to the Q output when *C* = 1
- When *C* = 0 the binary information is retained.
- We call latches level-sensitive devices.
  - So long as *C* remains at logic-1 level, any change in data input will change the state and the output of the latch.
  - Level sensitive latches may suffer from a serious problem.
- Memory devices that are sensitive to the rising or falling edge of control input is called flip-flops.

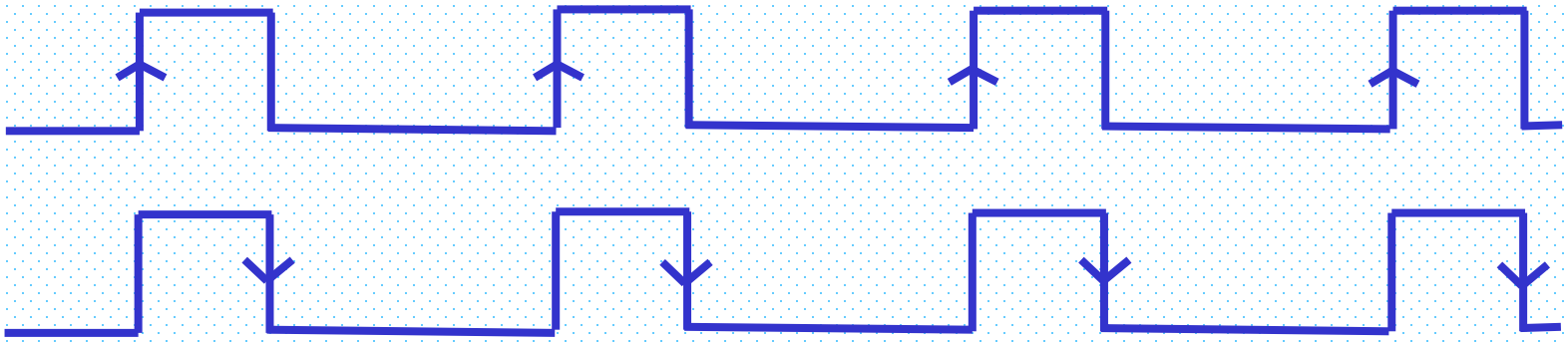
Need for Flip-Flops 1/2

- Outputs may keep changing so long as *C* = 1



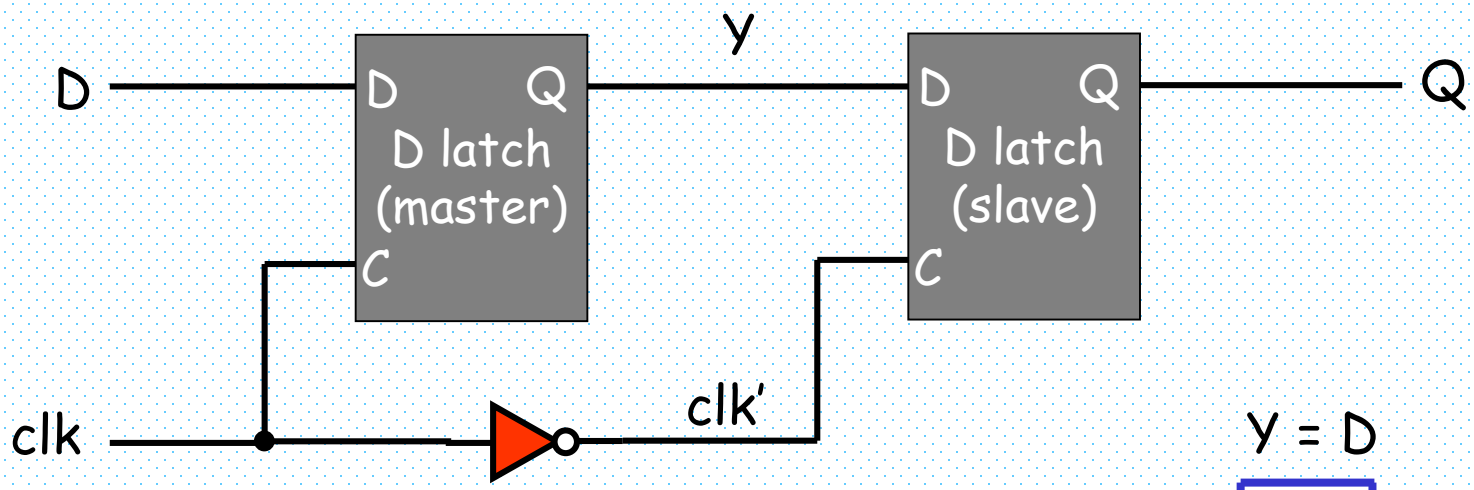
Need for Flip-Flops 2/2

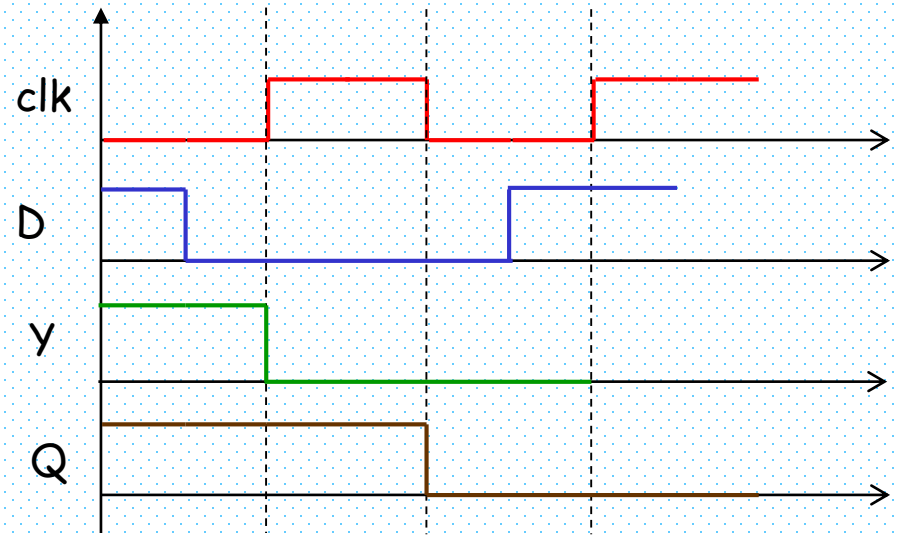
- Another issue (related to the first one)
  - The states of the memory elements to change synchronously
  - memory elements should respond to the changes in input at certain points in time.
  - This is the very characteristics of synchronous circuits.
  - To this end, we use flip-flops that change states during a signal transition of control input (clock)

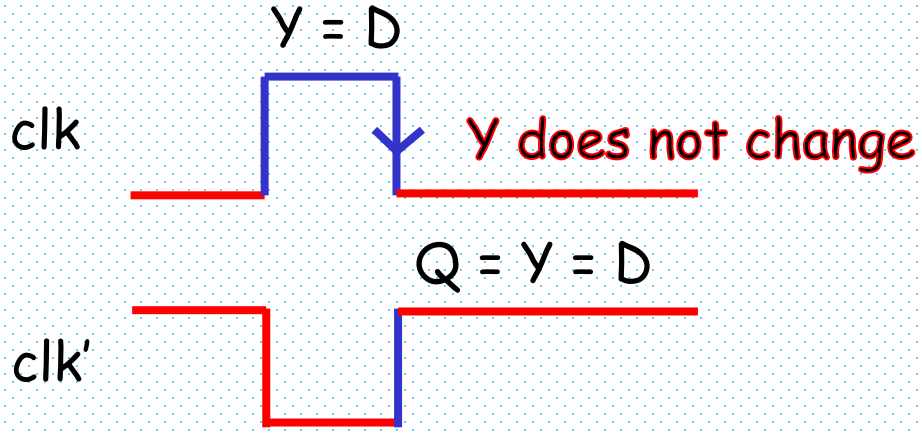


Edge-Triggered D Flip-Flop

- An edge-triggered D flip-flop can be constructed using two D latches



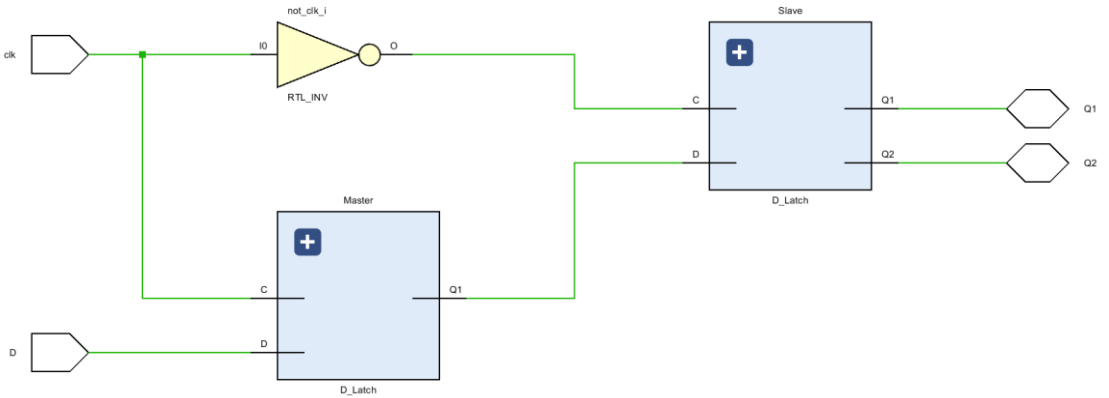




Negative edge-triggered D flip-flop

VHDL Module of Negative Edge Triggered D FF

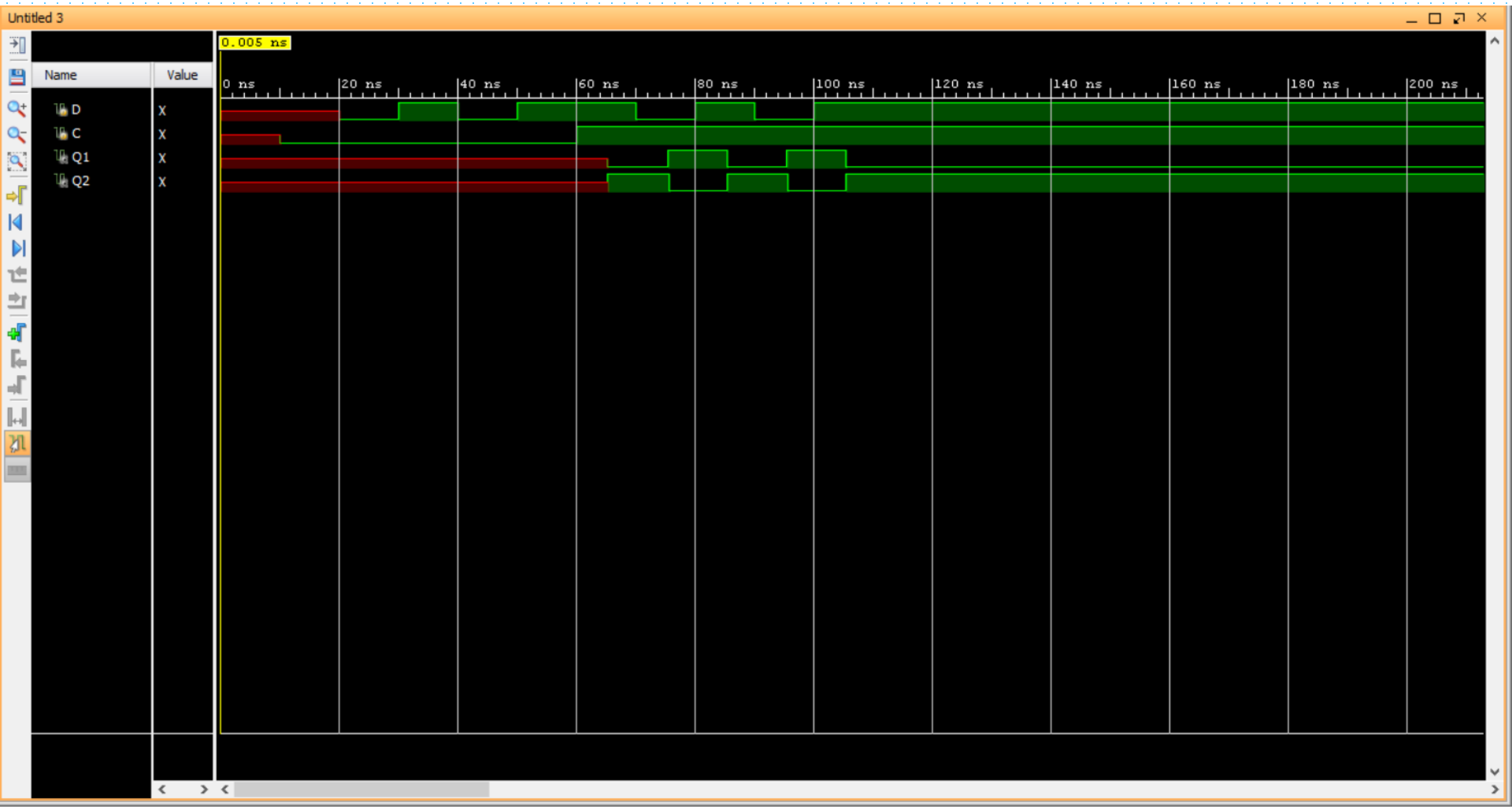
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity D_FF_Neg_Edge is
    Port(D : in STD_LOGIC;
         clk : in STD_LOGIC;
         Q1 : inout STD_LOGIC;
         Q2 : inout STD_LOGIC);
end D_FF_Neg_Edge;
architecture Behavioral of D_FF_Neg_Edge is
    component D_Latch is
    Port(D : in STD_LOGIC;
         C : in STD_LOGIC;
         Q1 : inout STD_LOGIC;
         Q2 : inout STD_LOGIC);
    end component;
    signal Y, not_Y,not_clk : STD_LOGIC;
begin
    not_clk <= not clk;
    Master: D_Latch Port map(D,clk,Y,not_Y);
    Slave: D_Latch Port map(Y,not_clk,Q1,Q2);
end Behavioral;
```



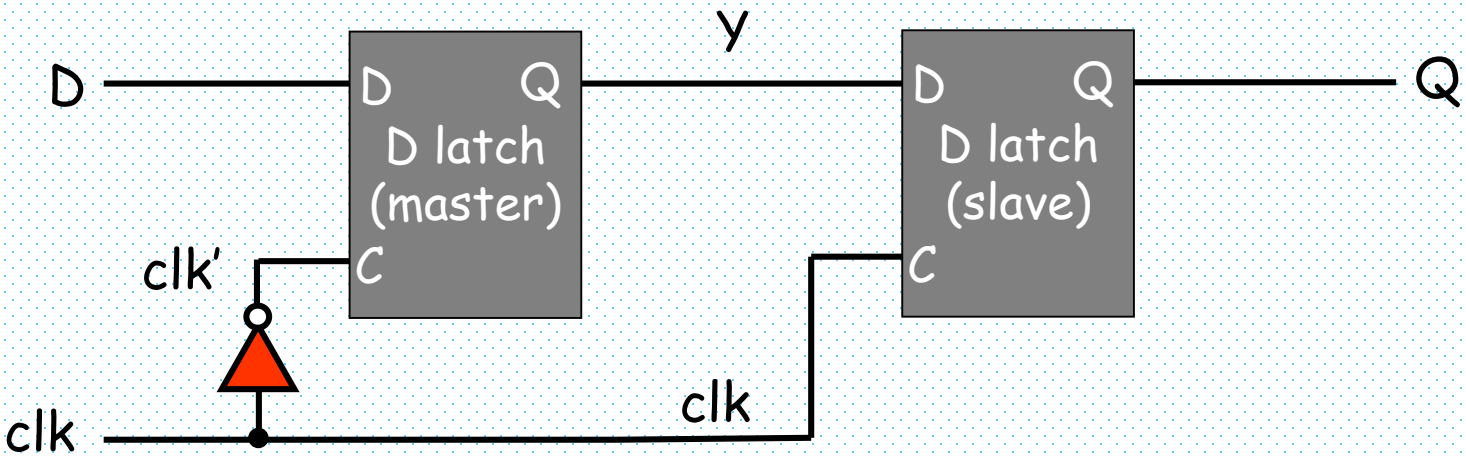
Testbench for Negative Edge Triggered D FF

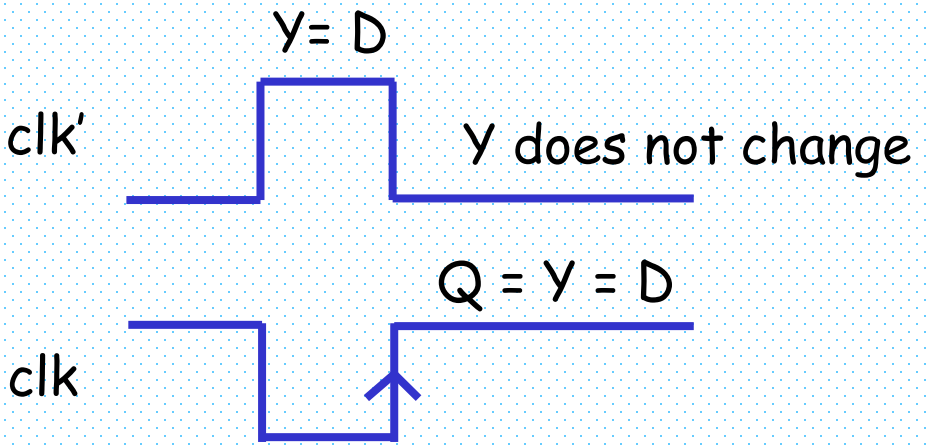
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity D_FF_Neg_Edge_tb is
end D_FF_Neg_Edge_tb;
architecture Behavioral of D_FF_Neg_Edge_tb is
    component D_FF_Neg_Edge is
    Port(D : in STD_LOGIC;
         clk : in STD_LOGIC;
         Q1 : inout STD_LOGIC;
         Q2 : inout STD_LOGIC);
    end component;
    signal D,clk,Q1,Q2 : STD_LOGIC := '0';
begin
    DUT: D_FF_Neg_Edge Port map (D,clk,Q1,Q2);
    process     begin
        wait for 5 ns;          clk <= not clk;
    end process;
    process     begin
        wait for 10 ns; D<='0';         wait for 10 ns; D<='1';
        wait for 10 ns; D<='1';         wait for 10 ns; D<='1';
        wait for 10 ns; D<='0';         wait for 10 ns; D<='1';
        wait for 10 ns; D<='0';         wait for 10 ns; D<='1';
    end process;
end Behavioral;
```

Simulation Waveform



Positive Edge-Triggered D Flip-Flop



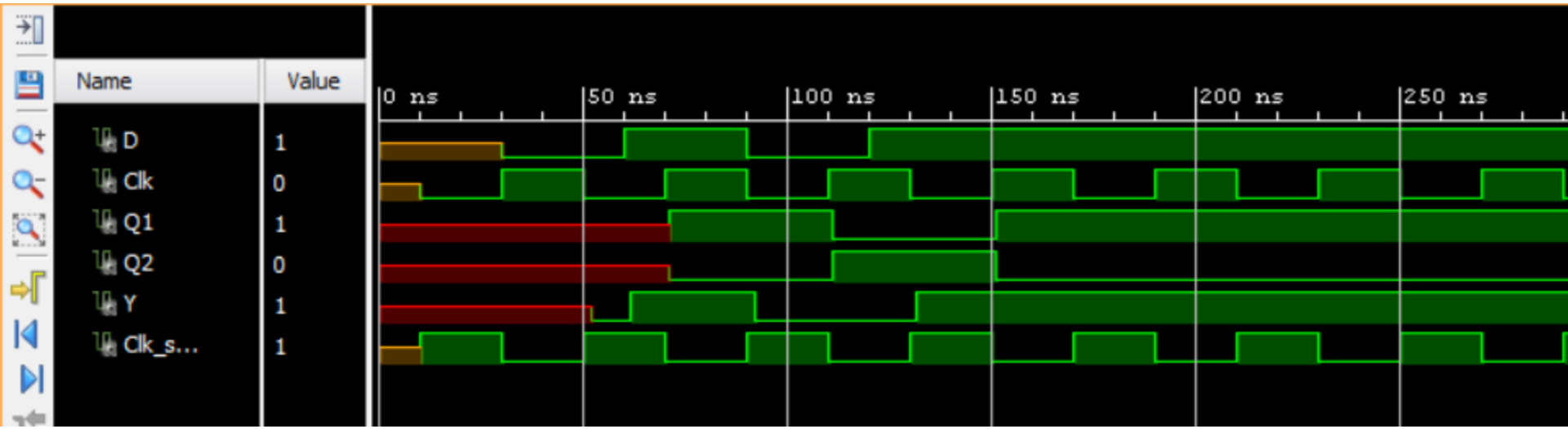


VHDL Entity of Positive Edge Triggered D FF and Testbench

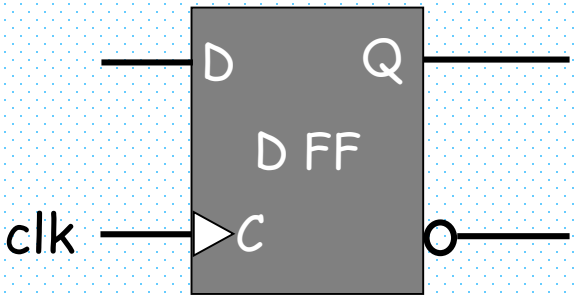
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity D_FF_Pos_Edge is
   Port ( D : in STD_LOGIC;
          Clk : in STD_LOGIC;
          Q1 : inout STD_LOGIC;
          Q2 : inout STD_LOGIC);
end D_FF_Pos_Edge;

architecture Behavioral of D_FF_Pos_Edge is
  component D_latch
   Port ( D : in STD_LOGIC;
          C : in STD_LOGIC;
          Q1 : inout STD_LOGIC;
          Q2 : inout STD_LOGIC);
  end component;
  signal Y,Clk_signal,Q2_signal : STD_LOGIC;
begin
  Master : D_latch Port map(D,Clk_signal,Y,Q2_signal);
  Slave : D_latch Port map(Y,Clk,Q1,Q2);
  Clk_signal <= not Clk;
end Behavioral;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity D_FF_Pos_Edge_tb is
end D_FF_Pos_Edge_tb;
architecture Behavioral of D_FF_Pos_Edge_tb is
  component D_FF_Pos_Edge
    Port ( D : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q1 : inout STD_LOGIC;
           Q2 : inout STD_LOGIC);
  end component;
  signal D,Clk,Q1,Q2 : STD_LOGIC;
begin
  DUT : D_FF_Pos_Edge Port map(D,Clk,Q1,Q2);
  process   begin
    wait for 10 ns;    Clk <= '0';
    for i in 1 to 1000 loop
      wait for 20 ns;      Clk <= not Clk;
    end loop;
    wait;
  end process;
  process  begin
    wait for 30 ns;    D <= '0';
    wait for 30 ns;    D <= '1';
    wait for 30 ns;    D <= '0';
    wait for 30 ns;    D <= '1';
    wait;
  end process;
end Behavioral;
```

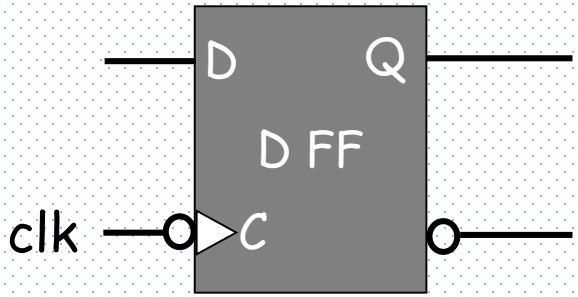
Simulation Waveform



Symbols for D Flip-Flops



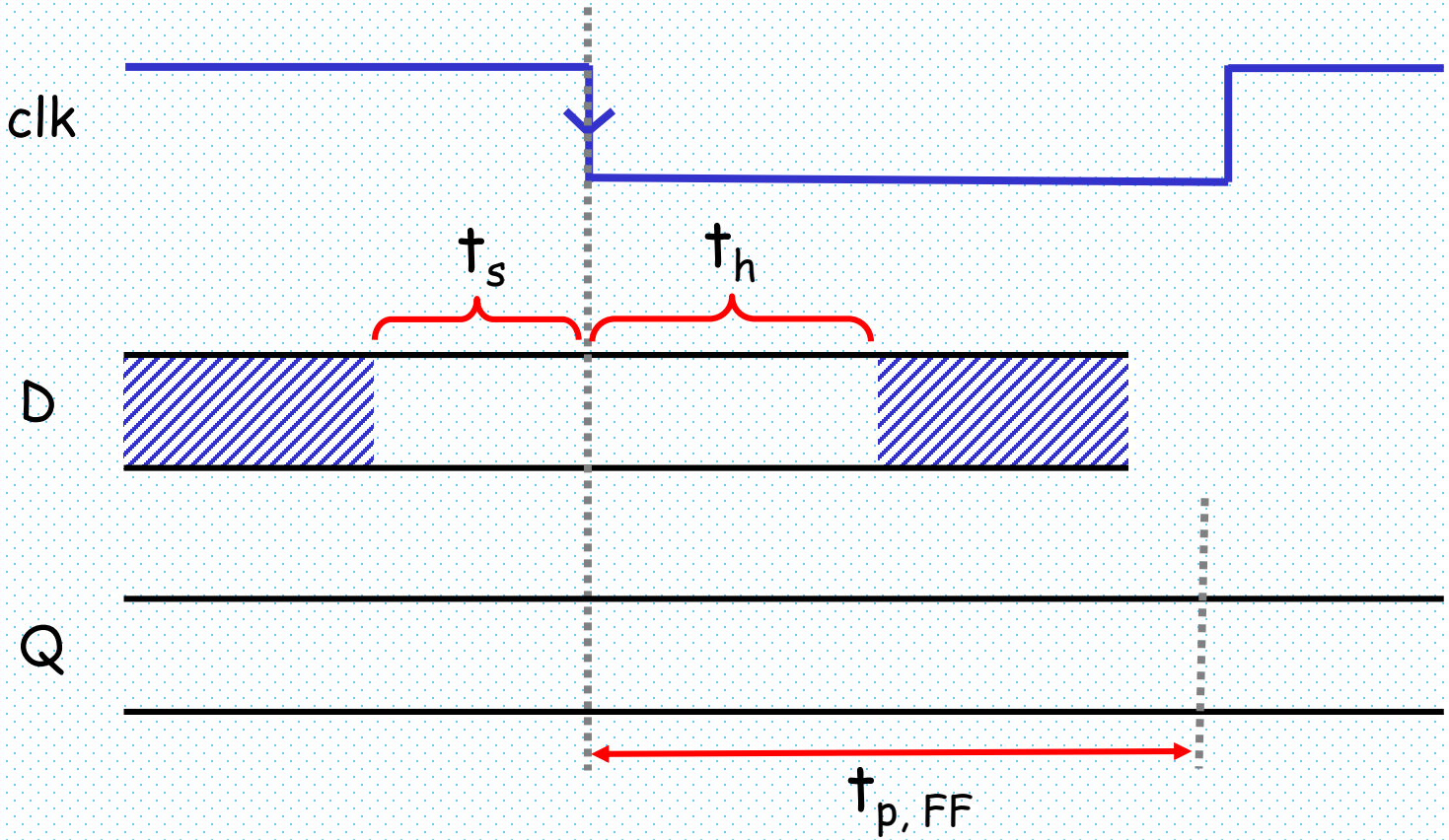
Positive edge-triggered
D Flip-Flop



Negative edge-triggered
D Flip-Flop

Setup & Hold Times 1/2

- Timing parameters are associated with the operation of flip-flops
- Recall Q gets the value of D in clock transition



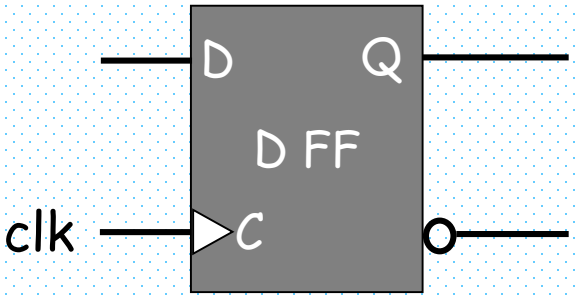
Setup & Hold Times 2/2

- Setup time, t_s
  - The change in the input D must be made before the clock transition.
  - Input D must maintain this new value for a certain minimum amount time.
  - If a change occurs at D less than t_s second before the clock transition, then the output may not acquire this new value.
  - It may even demonstrate unstable behavior.
- Hold time, t_h,
  - Similarly the value at D must be maintained for a minimum amount of time (i.e. t_h) after the clock transition.

Propagation Time

- Even if setup and hold times are achieved, it takes some time the circuit to propagate the input value to the output.
- This is because of the fact that flip-flops are made of logic gates that have certain propagation times.

D Flip-Flop



Positive edge-triggered
D Flip-Flop

- Characteristic equation
  - Q(t+1) = D

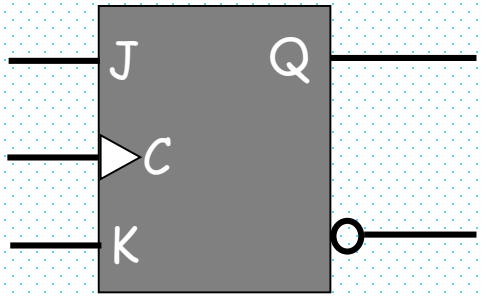
D	Q(t+1)
0	0
1	1

Characteristic Table

Other Flip-Flops

- D flip-flop is the most common
  - since it requires the fewest number of gates to construct.
- Two other widely used flip-flops
  - JK flip-flops
  - T flip-flops

JK Flip-Flop



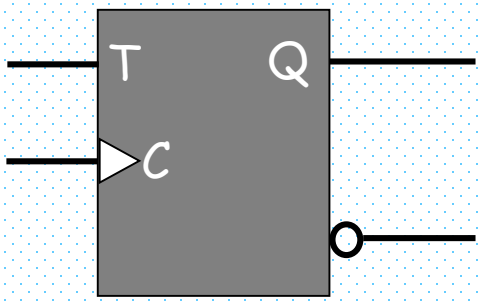
J	K	Q(t+1)	Next State
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement

- Characteristic Equation
  - Q(t+1) = JQ'(t) + K'Q(t)
  - Y = Jy' + K'y

J	K	y	Next State
0	0	y	No change
0	1	0	Reset
1	0	1	Set
1	1	y'	Complement

Characteristic Table

T (Toggle) Flip-Flop



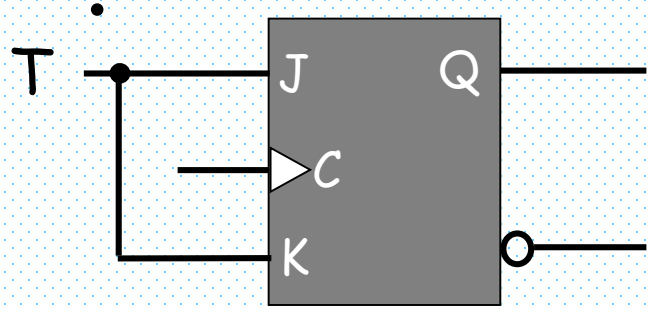
T	Q(t+1)	next state
0	Q(t)	no change
1	Q'(t)	Complement

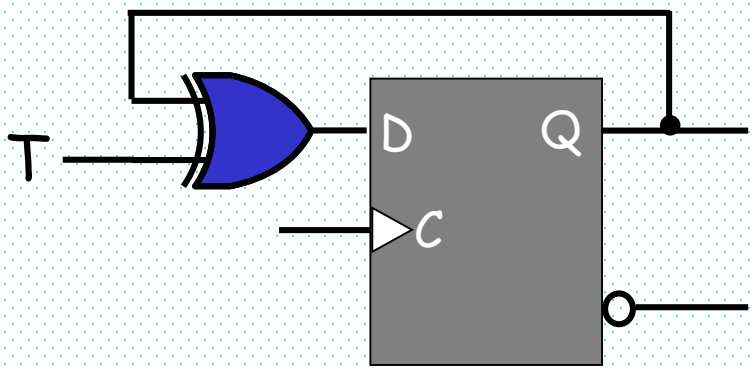
T	y	next state
0	y	no change
1	y'	Complement

Characteristic Equation

Characteristic Table

- $Q(t+1) = T \oplus Q(t) = TQ'(t) + T'Q(t)$
- $Y = T \oplus y = Ty' + T'y$
-

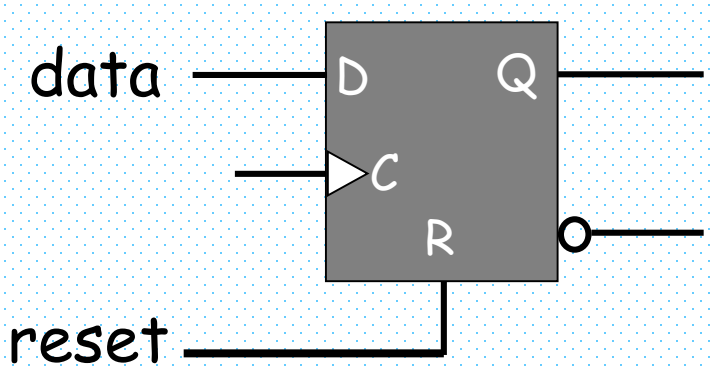


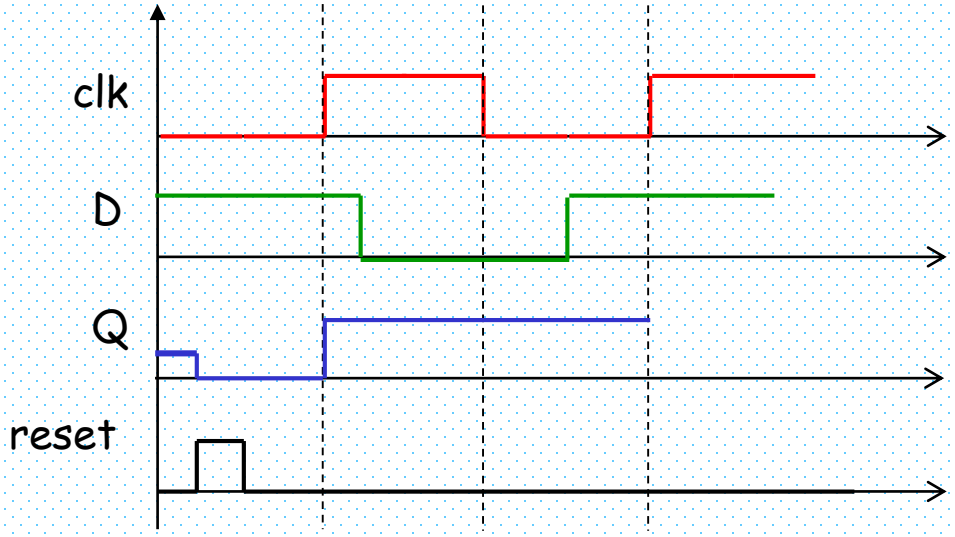


Asynchronous Inputs of Flip-Flops

- They are used to force the flip-flop to a particular state independent of clock
  - "Preset" (direct set) set FF state to 1
  - "Clear" (direct reset) set FF state to 0
- They are especially useful at startup.
  - In digital circuits when the power is turned on, the state of flip-flops are unknown.
  - Asynchronous inputs are used to bring all flip-flops to a known "starting" state prior to clock operation.

Asynchronous Inputs





reset	C	D	Q	Q′	
1	X	X	0	1	Starting State
0	↑	0	0	1	
0	↑	1	1	0	

Design Process

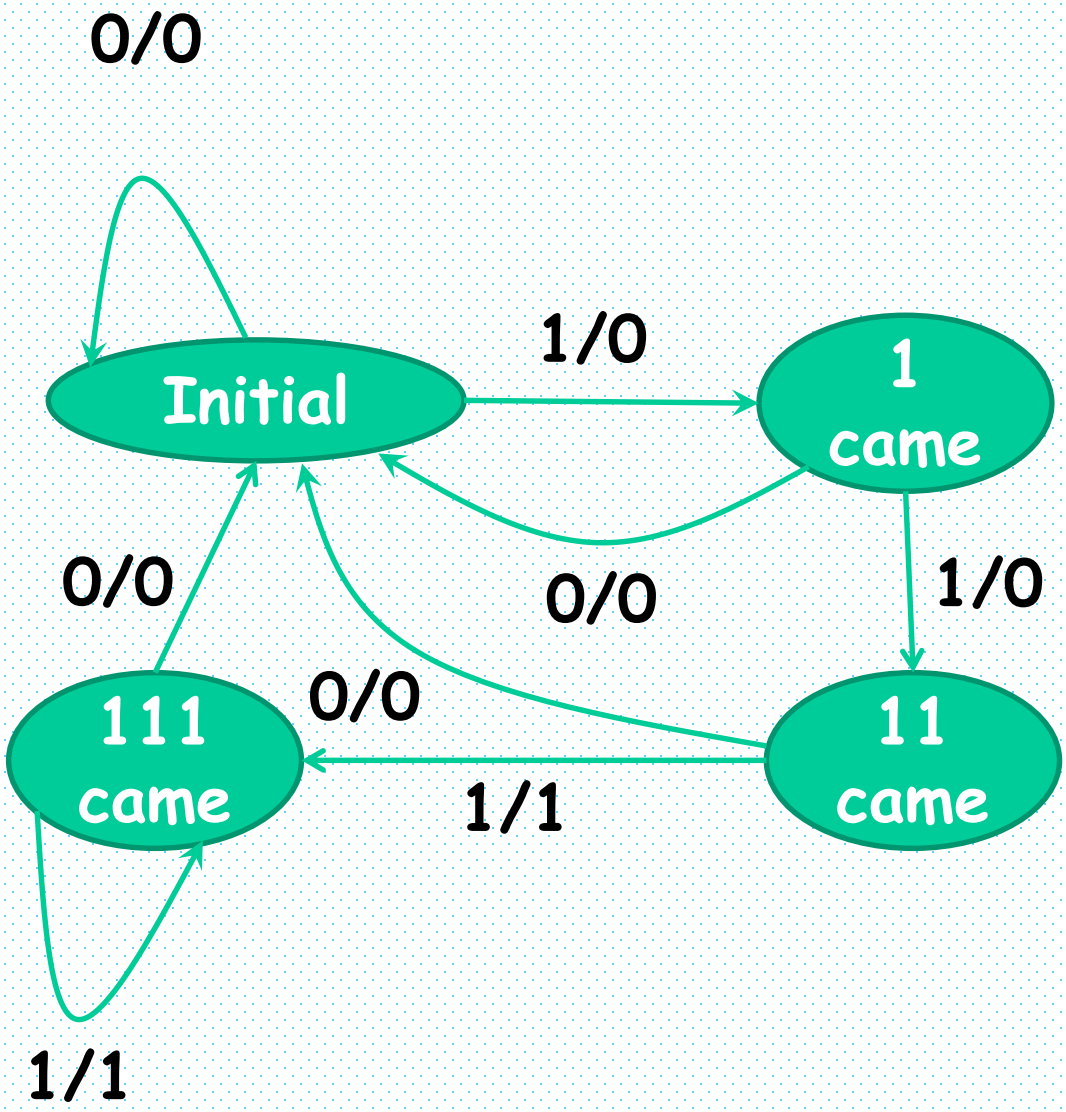
1. Verbal description of desired operation
2. Draw the state diagram
3. Reduce the number of states if necessary and possible: s = number of states
4. Determine the number of flip-flops: $n = \lceil \log_2 s \rceil$
5. State assignment: $\underbrace{00\ldots0}_{n-bits}, \underbrace{00\ldots1}_{n-bits}, \underbrace{00\ldots10}_{n-bits}, \ldots$
6. Obtaine the encoded state table
7. Choose the type of the flip-flops
8. Derive the simplified flip-flop input equations
9. Derive the simplified output equations
10. Draw the logic diagram

Example: Design of a Synchronous Sequential Circuit

- Verbal description
  - 1st Step: we want a circuit that detects three or more consecutive 1's in a string of bits.
    - Input: string of bits of any length
    - Output:
      - "1" if the circuit detects such a pattern in the string
      - "0" otherwise

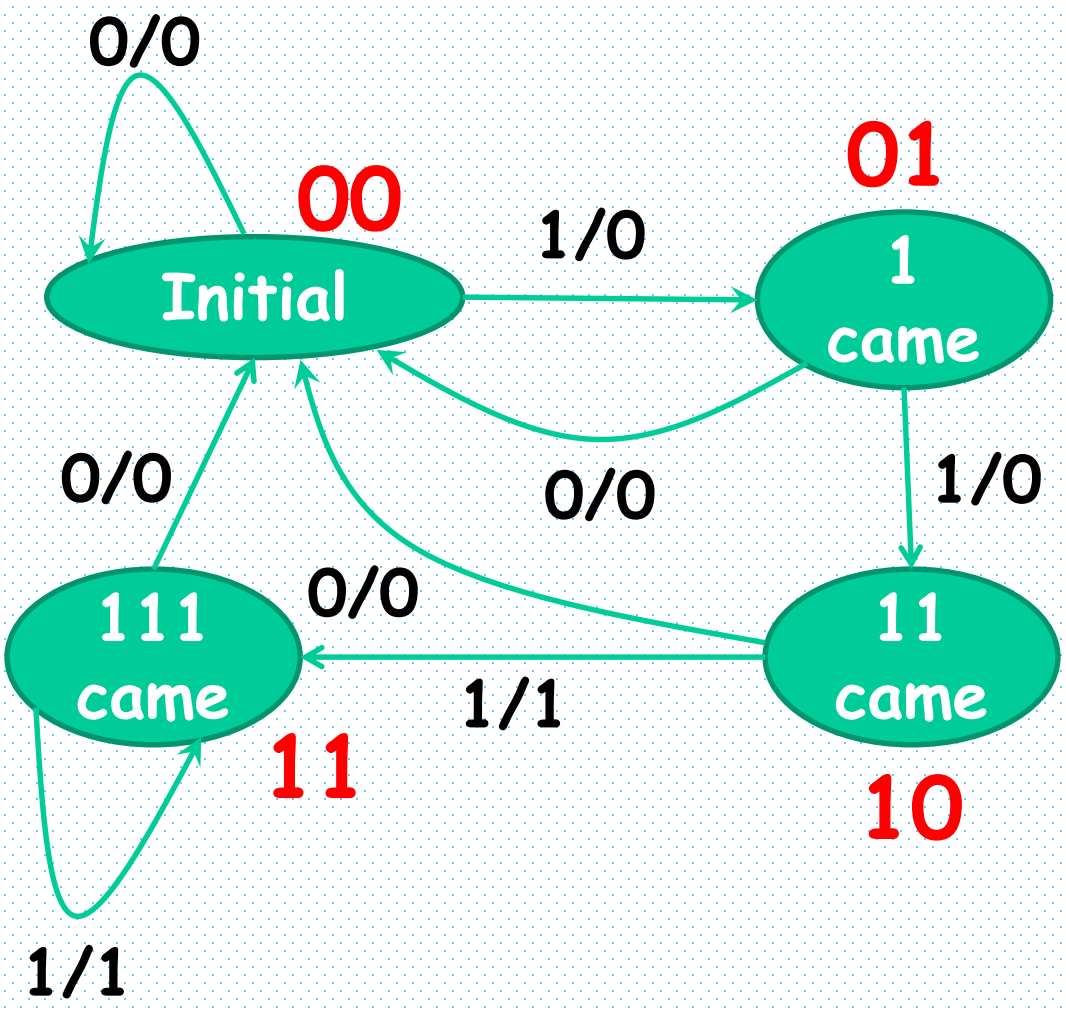
Example: State Diagram

2nd Step: **Draw the state diagram**



Synthesis with D Flip-Flops 1/5

- 3rd Step: State reduction
- 4th Step: Number of flip-flops
  - 4 states
  - ? flip-flop
- 5th Step: State assignment



Synthesis with D Flip-Flops 2/5

- 6th Step: Obtain the state table

Present State		Input	Next State		Output
y_1	y_2	x	Y_1	Y_2	z
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Synthesis with D Flip-Flops 3/5

- 7th Step: Choose the type of the flip-flops
  - D type flip-flops
- 8th Step: : Derive the simplified flip-flop input equations
  - Boolean expressions for D_1 and D_2

y_1 \ y_2x	00	01	11	10
0	0	0	1	0
1	0	1	1	0

$D_1 = y_1x + y_2x$

y_1 \ y_2x	00	01	11	10
0	0	1	0	0
1	0	1	1	0

$D_2 = y_1x + y_2'x$

Synthesis with D Flip-Flops 4/5

- 9th Step: : Derive the simplified output equations
  - Boolean expressions for z

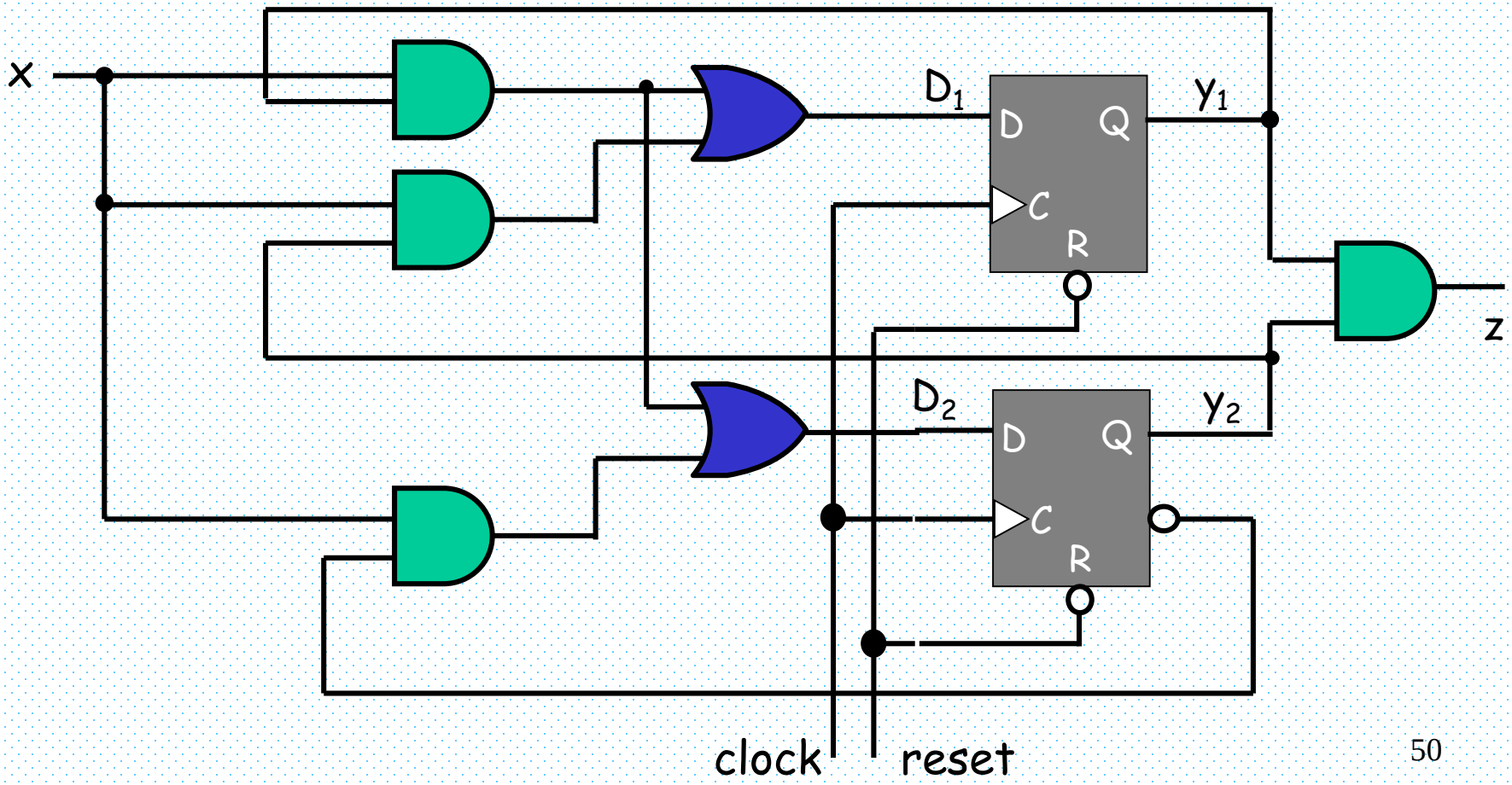
y_1 \ y_2x	00	01	11	10
0	0	0	0	0
1	0	0	1	1

$z = y_1y_2$

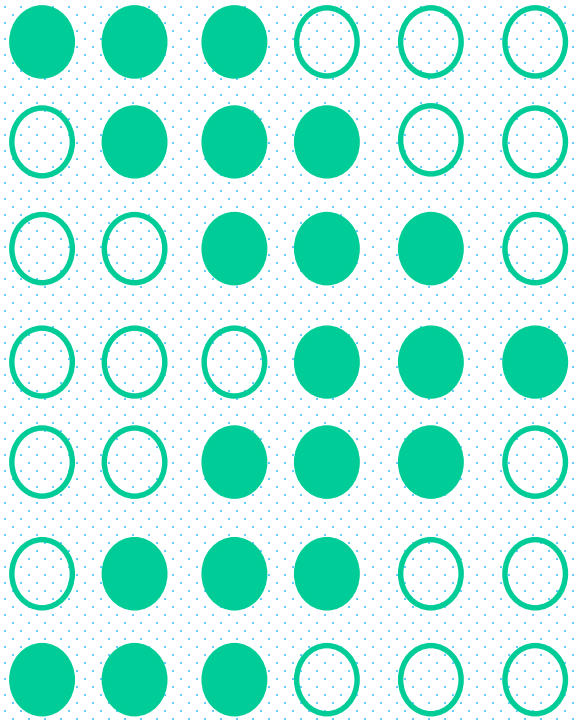
Synthesis with D Flip-Flops 5/5

- 10th Step: Draw the logic diagram

$D_1 = y_1x + y_2x$ $D_2 = y_1x + y_2'x$ $z = y_1y_2$



Synthesis with JK Flip-Flops and MUXs



Number of states= 6

Number of state variables= 3

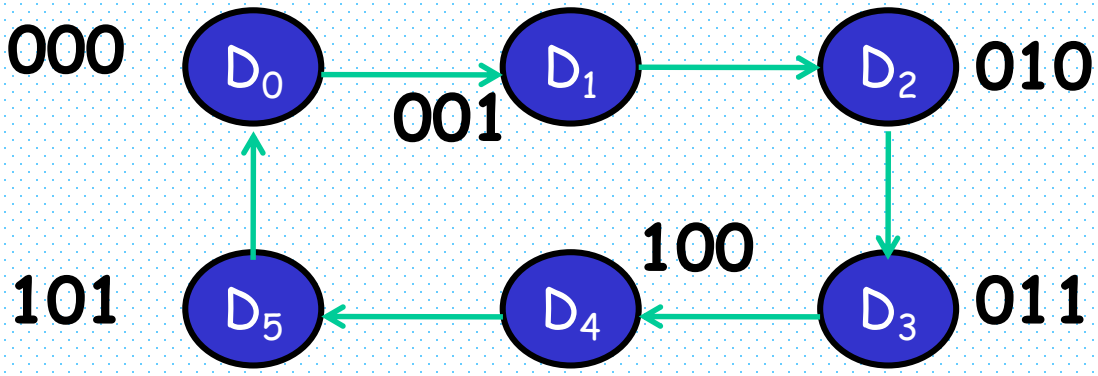
Number of flip-flops= 3

Number of Inputs= 0

Number of Outputs= 6

- 6 shifting lights
- ● = lojik-1
- O = lojik-0

State Diagram & Table



$Y = Jy' + K'y$

J	K	Y
0	0	y
0	1	0
1	0	1
1	1	Q'

Present State	Next State	Flip-flop inputs						Outputs					
y_2 y_1 y_0	Y_2 Y_1 Y_0	J_2	K_2	J_1	K_1	J_0	K_0	z_5	z_4	z_3	z_2	z_1	z_0
0 0 0	0 0 1	0	k	0	k	1	k	1	1	1	0	0	0
0 0 1	0 1 0	0	k	1	k	k	1	0	1	1	1	0	0
0 1 0	0 1 1	0	k	k	0	1	k	0	0	1	1	1	0
0 1 1	1 0 0	1	k	k	1	k	1	0	0	0	1	1	1
1 0 0	1 0 1	k	0	0	k	1	k	0	0	1	1	1	0
1 0 1	0 0 0	k	1	0	k	k	1	0	1	1	1	0	0

Inplementation of Flip-Flop Input Equations

y_2 \ y_1y_0	00	01	11	10
0	0	0	0	1
1	k	k	k	k

$J_2 = y_1y_0'$

y_2 \ y_1y_0	00	01	11	10
0	k	k	k	k
1	0	1	k	k

$K_2 = y_0$

y_2 \ y_1y_0	00	01	11	10
0	0	1	k	k
1	0	0	k	k

$J_1 = y_2'y_0$

y_2 \ y_1y_0	00	01	11	10
0	k	k	1	0
1	k	k	k	k

$K_1 = y_0$

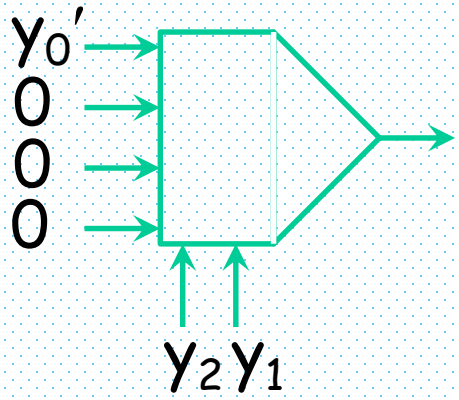
y_2 \ y_1y_0	00	01	11	10
0	1	k	k	1
1	1	k	k	k

$J_0 = 1$

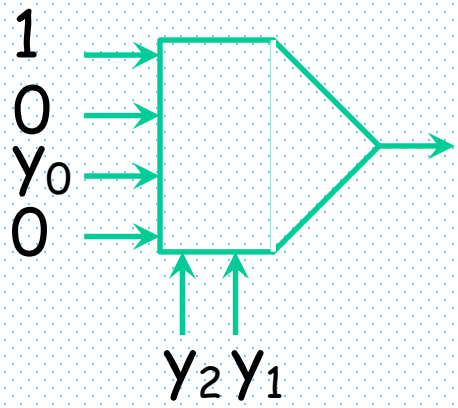
y_2 \ y_1y_0	00	01	11	10
0	k	1	1	k
1	k	1	k	k

$K_1 = 1$

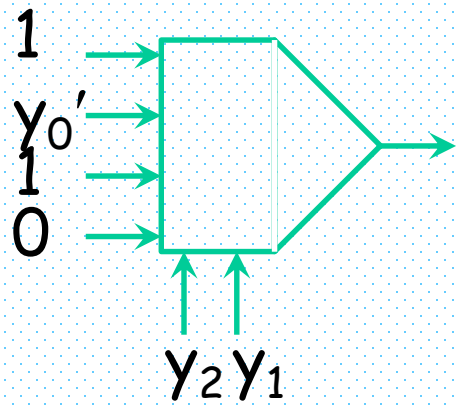
Inplementation of Output Equations



$z_5 = y_2'y_1'y_0' + k(y_2y_1y_0' + y_2y_1y_0)$

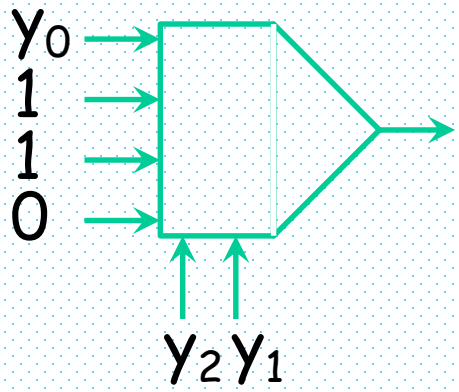


$z_4 = y_2'y_1'y_0' + y_2'y_1'y_0 + y_2y_1'y_0 + k(y_2y_1y_0' + y_2y_1y_0)$

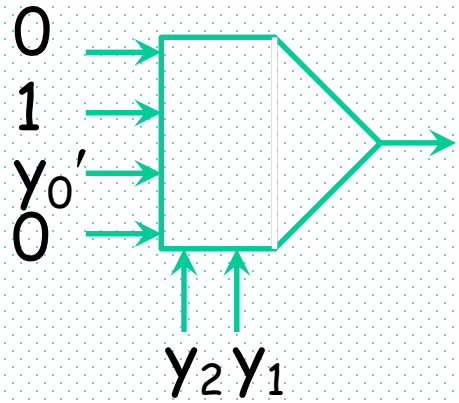


$z_3 = y_2'y_1'y_0' + y_2'y_1'y_0 + y_2'y_1y_0' + y_2y_1'y_0' + y_2y_1'y_0 + k(y_2y_1y_0' + y_2y_1y_0)$

Inplementation of Output Equations

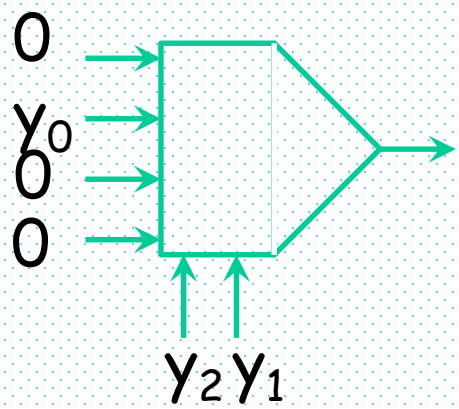


$$z_2 = y_2'y_1'y_0 + y_2'y_1y_0' + y_2'y_1y_0 + y_2y_1'y_0' + y_2y_1'y_0 + k(y_2y_1y_0' + y_2y_1y_0)$$



$$z_1 = y_2'y_1y_0' + y_2'y_1y_0 + y_2y_1'y_0' + k(y_2y_1y_0' + y_2y_1y_0)$$

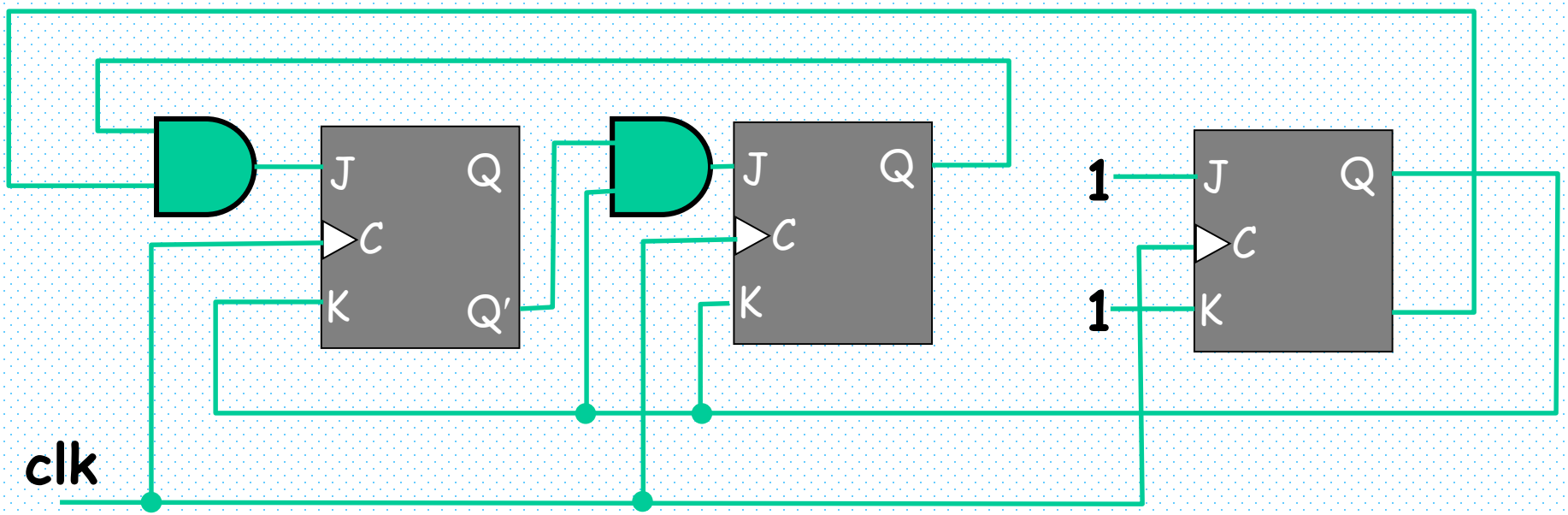
Inplementation of Output Equations



$z_0 = y_2' y_1 y_0 + k(y_2 y_1 y_0' + y_2 y_1 y_0)$

Logic Diagram

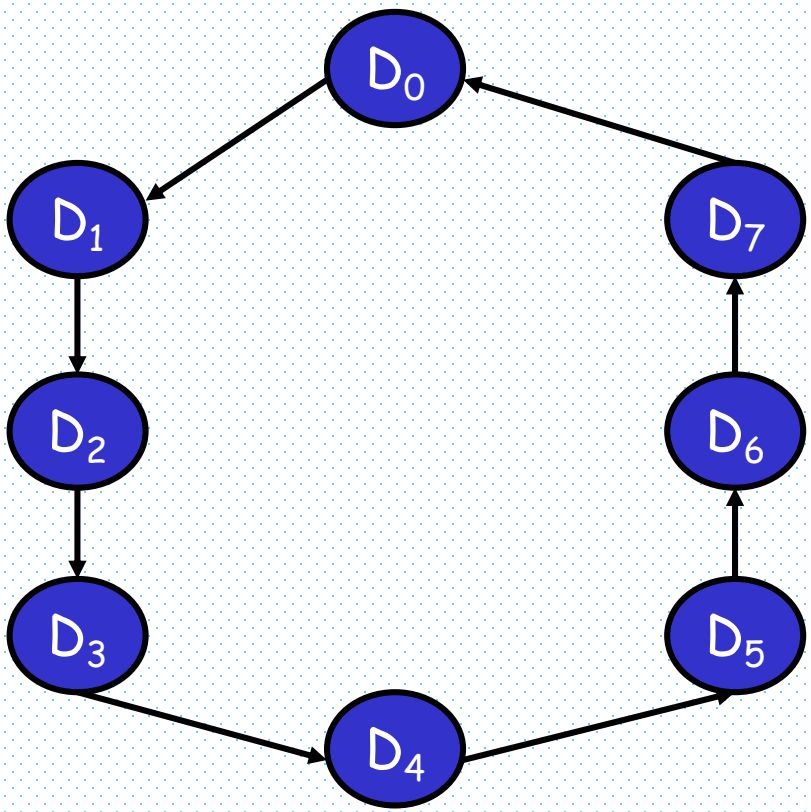
$J_2 = y_1y_0'$ $K_2 = y_0$ $J_1 = y_2'y_0$ $K_1 = y_0$ $J_0 = 1$ $K_1 = 1$



Synthesis with T Flip-Flops 1/4

- Example: 3-bit binary counter

0→1→2→ … → 7 → 0 → 1 → 2



State Diagram

How many flip-flops?

State assignments

- D_0 → 000
- D_1 → 001
- D_2 → 010
- . . .
- D_7 → 111

Synthesis with T Flip-Flops 2/4

- State Table

present state			next state			FF inputs		
y_2	y_1	y_0	Y_2	Y_1	Y_0	T_2	T_1	T_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Synthesis with T Flip-Flops 3/4

- Flip-Flop input equations

y_2 \ y_1y_0	00	01	11	10
0	0	0	1	0
1	0	0	1	0

$T_2 = y_1y_0$

y_2 \ y_1y_0	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$T_1 = y_0$

$T_0 = 1$

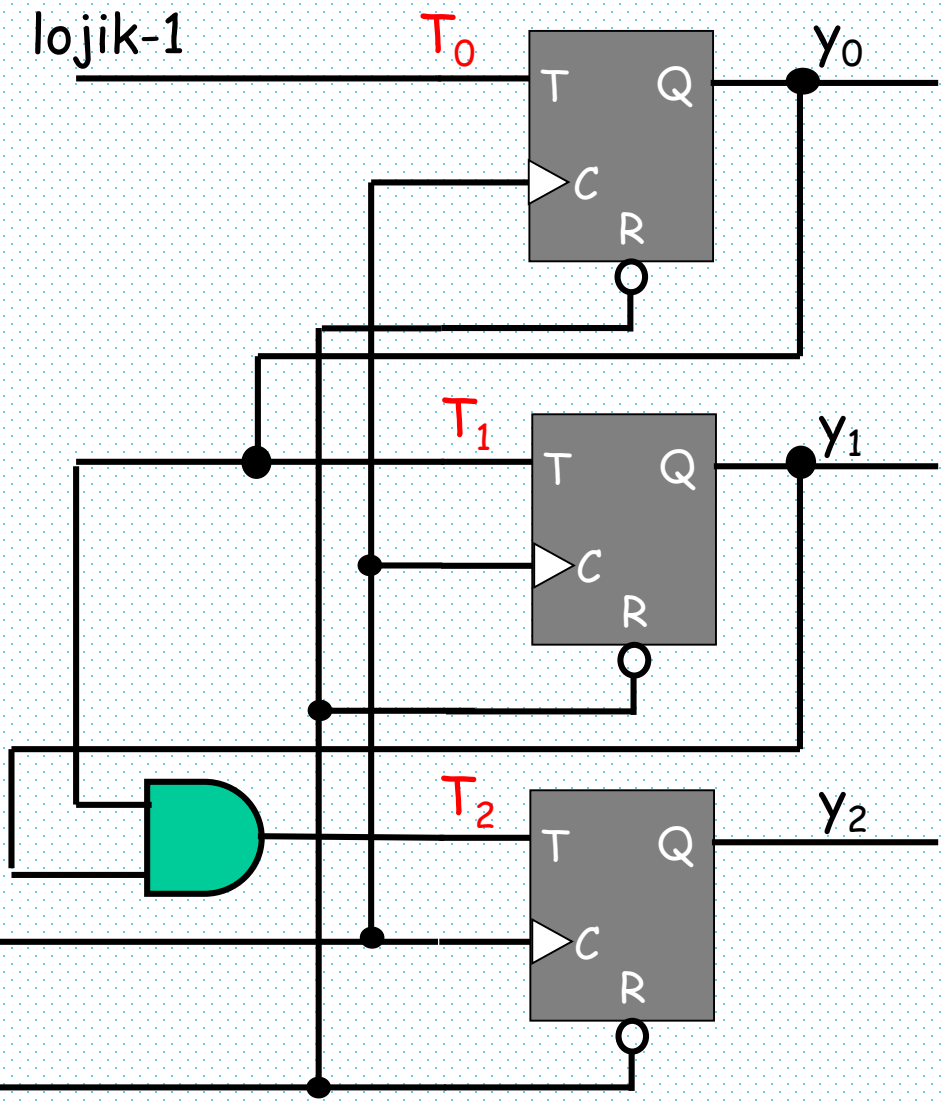
Synthesis with T Flip-Flops 4/4

- Circuit

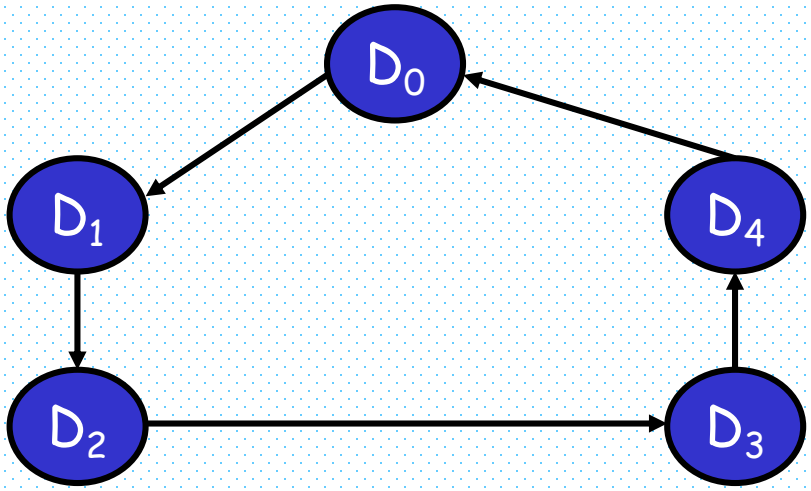
$T_2 = y_1 y_0$

$T_1 = y_0$

$T_0 = 1$



Unused States



Modulo-5 counter

Present State			Next State		
Y_2	Y_1	Y_0	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0

Example: Unused States 1/4

Present State			Next State		
y_2	y_1	y_0	y_2	y_1	y_0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0

y_2 \ y_1y_0	00	01	11	10
0	0	0	1	0
1	0	X	X	X

$Y_2 = y_1y_0$

y_2 \ y_1y_0	00	01	11	10
0	0	1	0	1
1	0	X	X	X

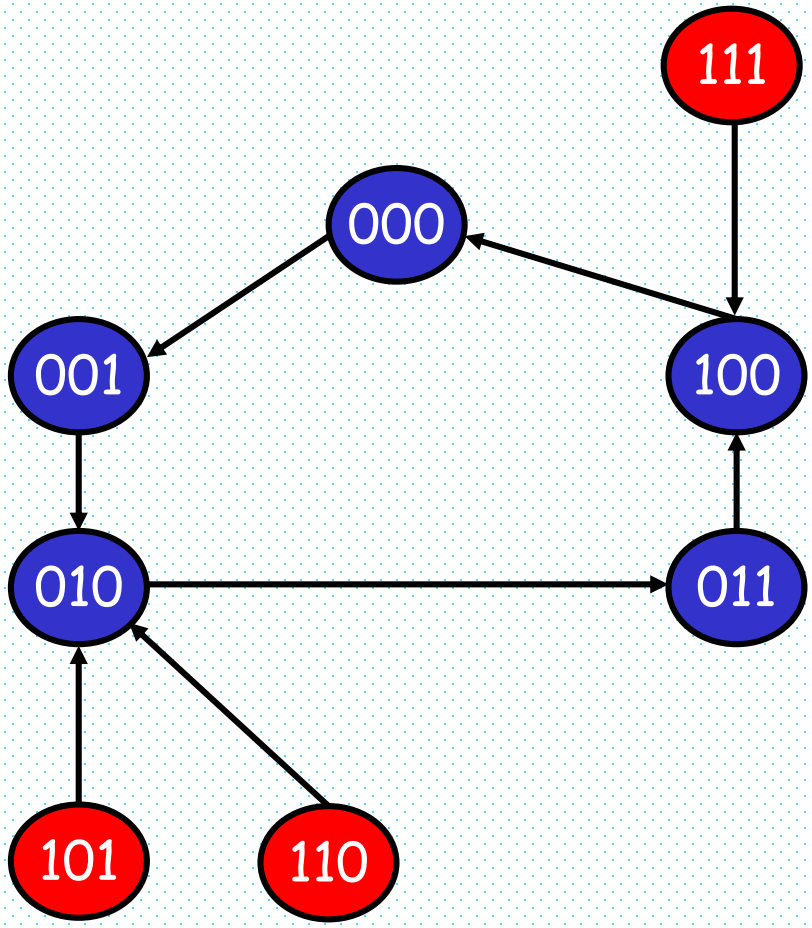
$Y_1 = y_1' y_0 + y_1y_0'$
$= y_1 \oplus y_0$

y_2 \ y_1y_0	00	01	11	10
0	1	0	0	1
1	0	X	X	X

$Y_0 = y_2' y_0'$

Example: Unused States 2/4

Present State			Next State		
y_2	y_1	y_0	y_2	y_1	y_0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	0	0



$Y_2 = y_1 y_0$

$Y_1 = y_1 \oplus y_0$

$Y_0 = y_2' y_0'$

The circuit is not locked type.

Example: Unused States 3/4

- Not using don't care conditions

Present State			Next State		
A	B	C	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0

A \ BC	00	01	11	10
0	0	0	1	0
1	0	0	0	0

$A(t+1) = A'BC$

A \ BC	00	01	11	10
0	0	1	0	1
1	0	0	0	0

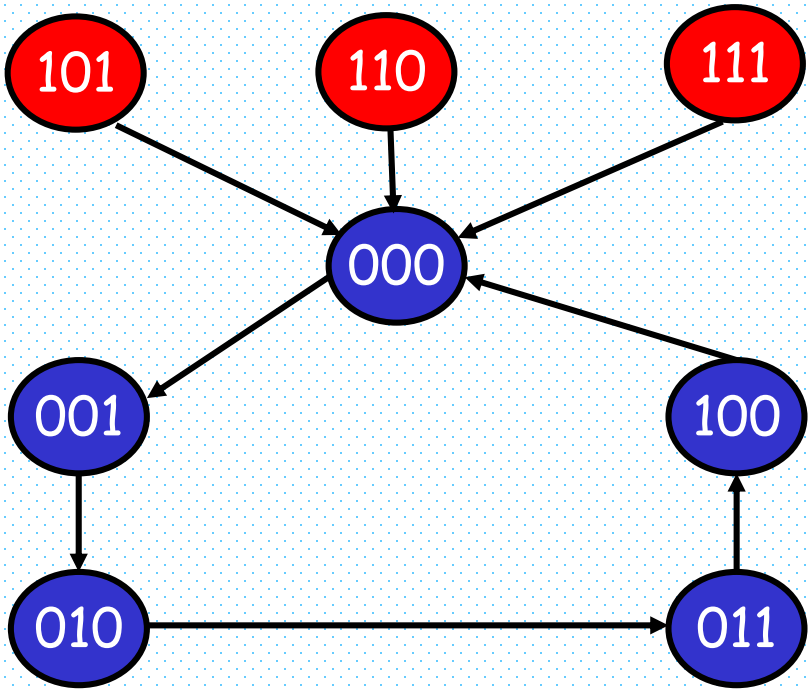
$B(t+1) = A'B'C + A'BC'$
$= A'(B \oplus C)$

A \ BC	00	01	11	10
0	1	0	0	1
1	0	0	0	0

$C(t+1) = A'C'$

Example: Unused States 4/4

Present State			Next State		
A	B	C	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0



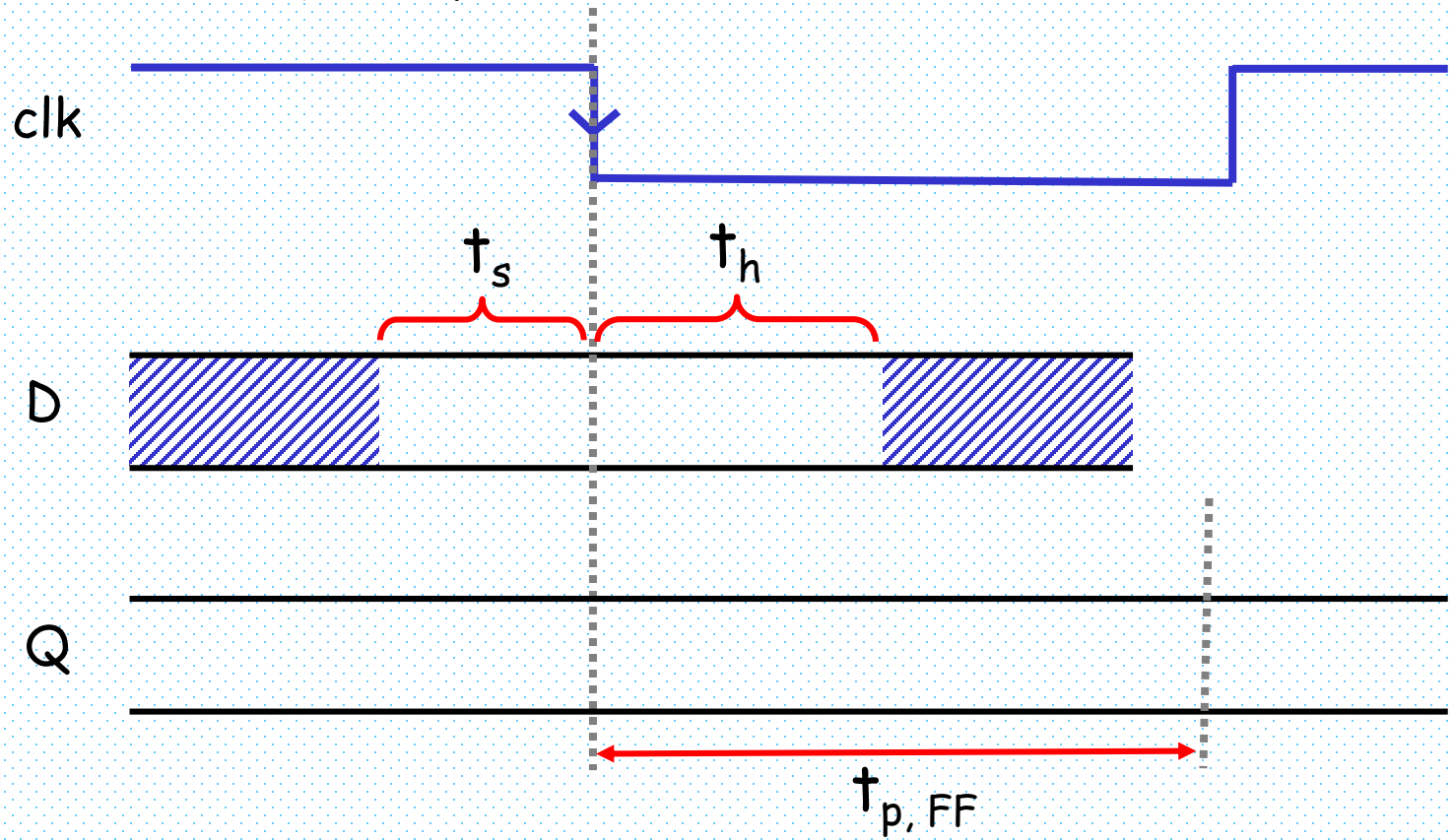
$A(t+1) = A'BC$

$B(t+1) = A'(B \oplus C)$

$C(t+1) = A'C'$

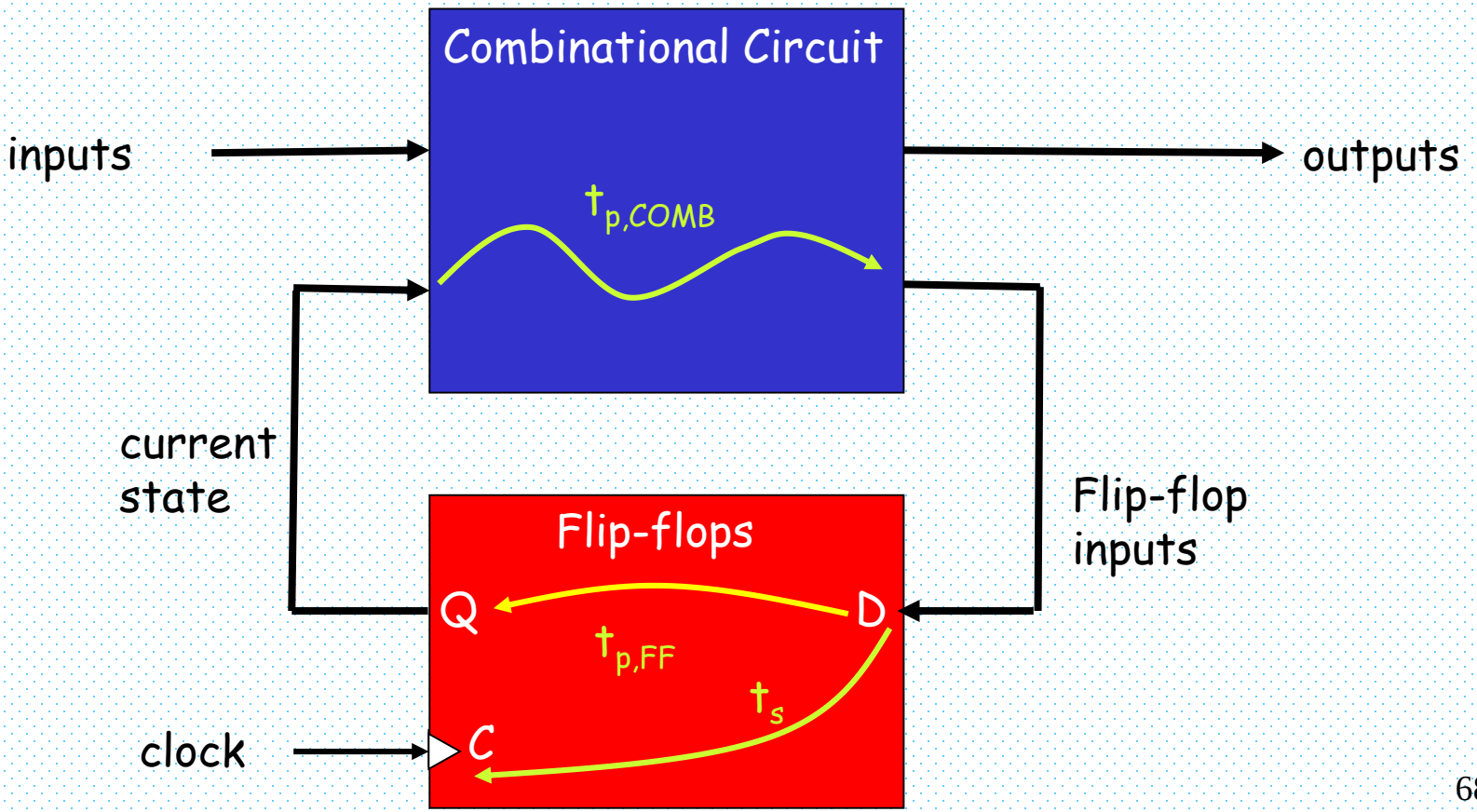
Sequential Circuit Timing 1/3

- It is important to analyze the timing behavior of a sequential circuit
  - Ultimate goal is to determine the maximum clock frequency



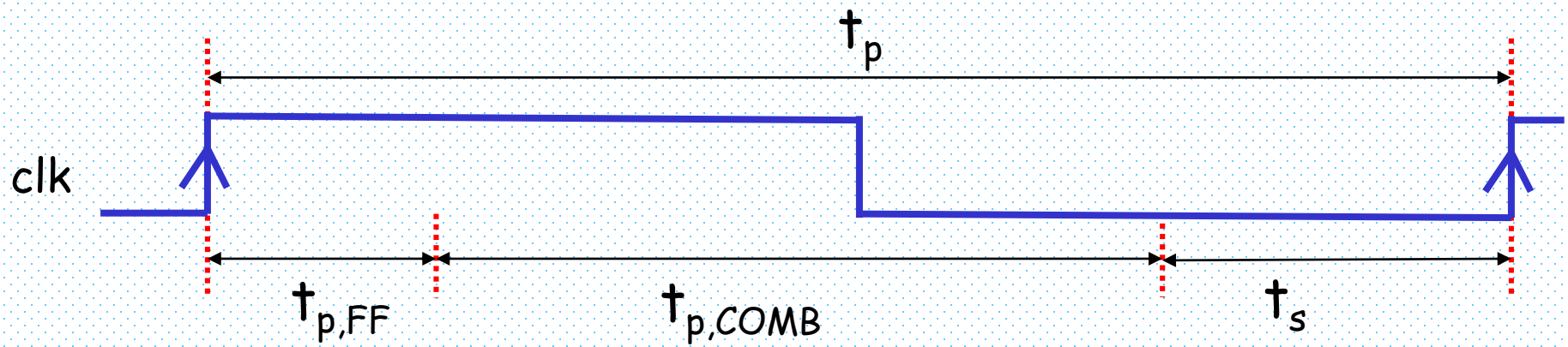
Sequential Circuit Timing 2/3

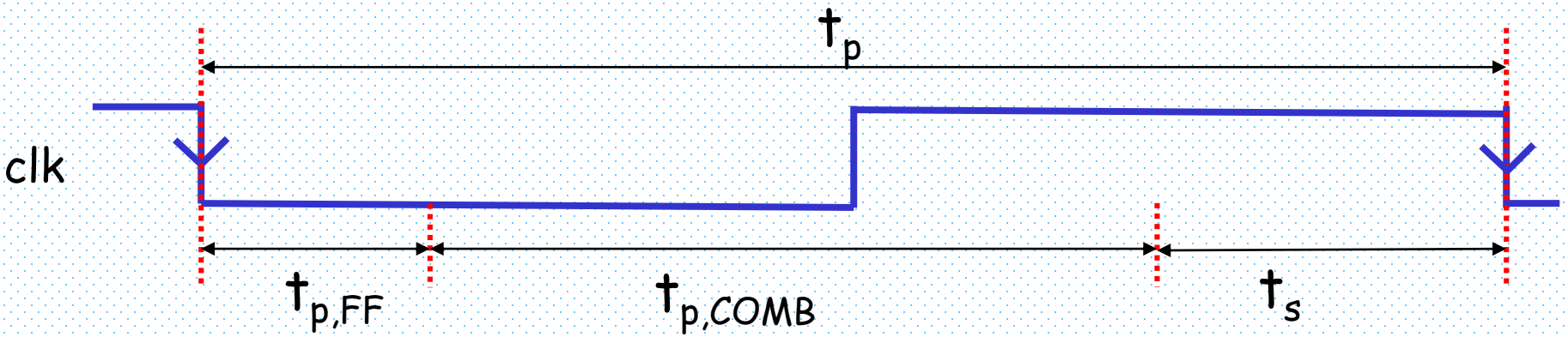
$t_{p,FF} + t_{p,COMB} >> t_h$



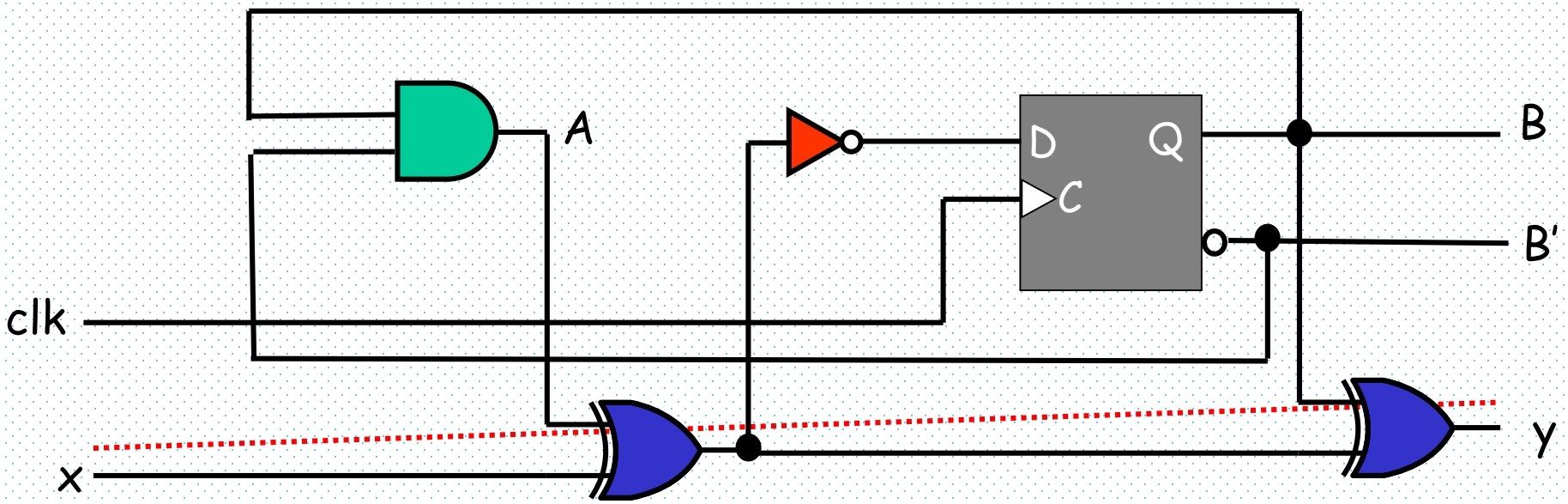
Sequential Circuit Timing 3/3

- Minimum clock period (or maximum clock frequency)





Example: Sequential Circuit Timing



$t_{p,NOT}$ = 0.5 ns

$t_{p,XOR}$ = 2.0 ns

$t_{p,FF}$ = 2.0 ns

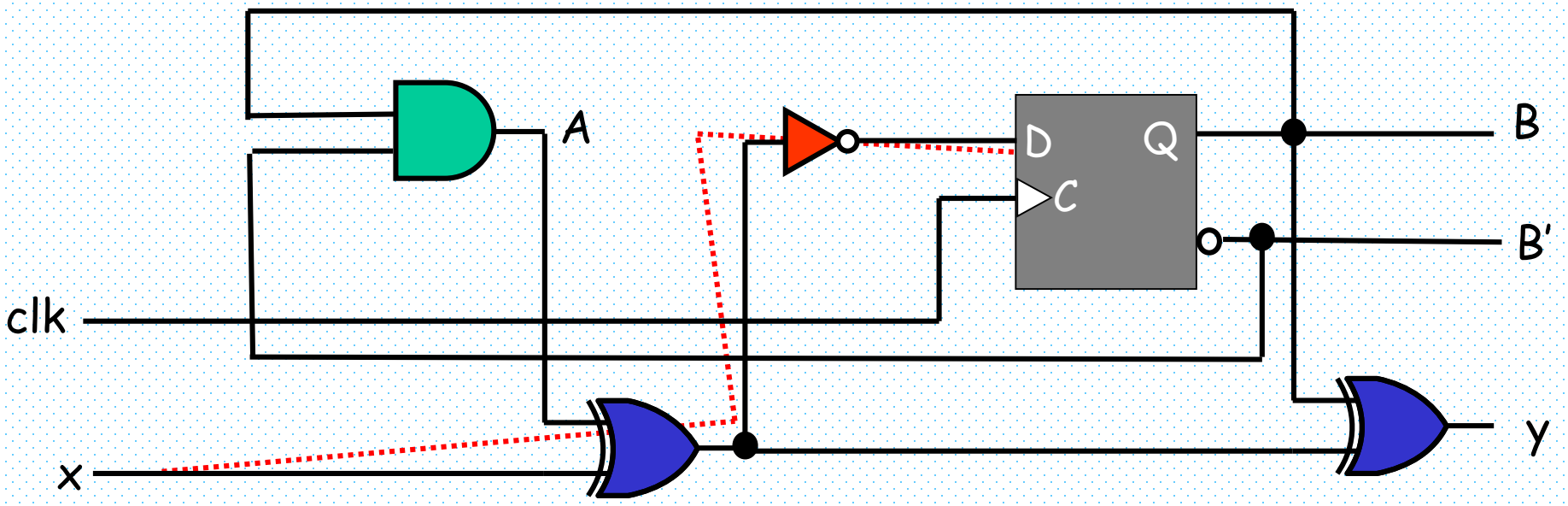
$t_{p,AND} = t_s$ = 1.0 ns

t_h = 0.25 ns

Find the longest path delay from external input to the output

$t_{p,XOR} + t_{p,XOR}$ = 2.0 + 2.0 = 4.0 ns

Example: Sequential Circuit Timing



$t_{p,NOT}$ = 0.5 ns

$t_{p,XOR}$ = 2.0 ns

$t_{p,FF}$ = 2.0 ns

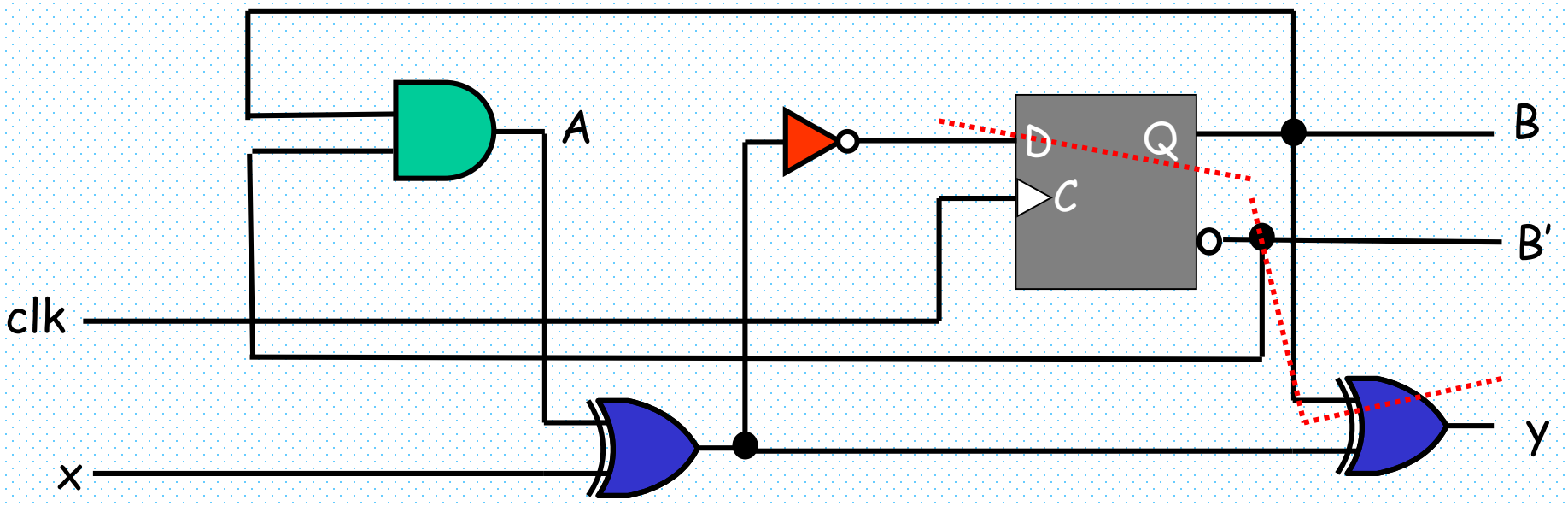
$t_{p,AND} = t_s$ = 1.0 ns

t_h = 0.25 ns

Find the longest path delay in the circuit from external input to positive clock edge

$t_{p,XOR} + t_{p,NOT}$ = 2.0 + 0.5 = 2.5 ns

Example: Sequential Circuit Timing



$t_{p,NOT}$ = 0.5 ns

$t_{p,XOR}$ = 2.0 ns

$t_{p,FF}$ = 2.0 ns

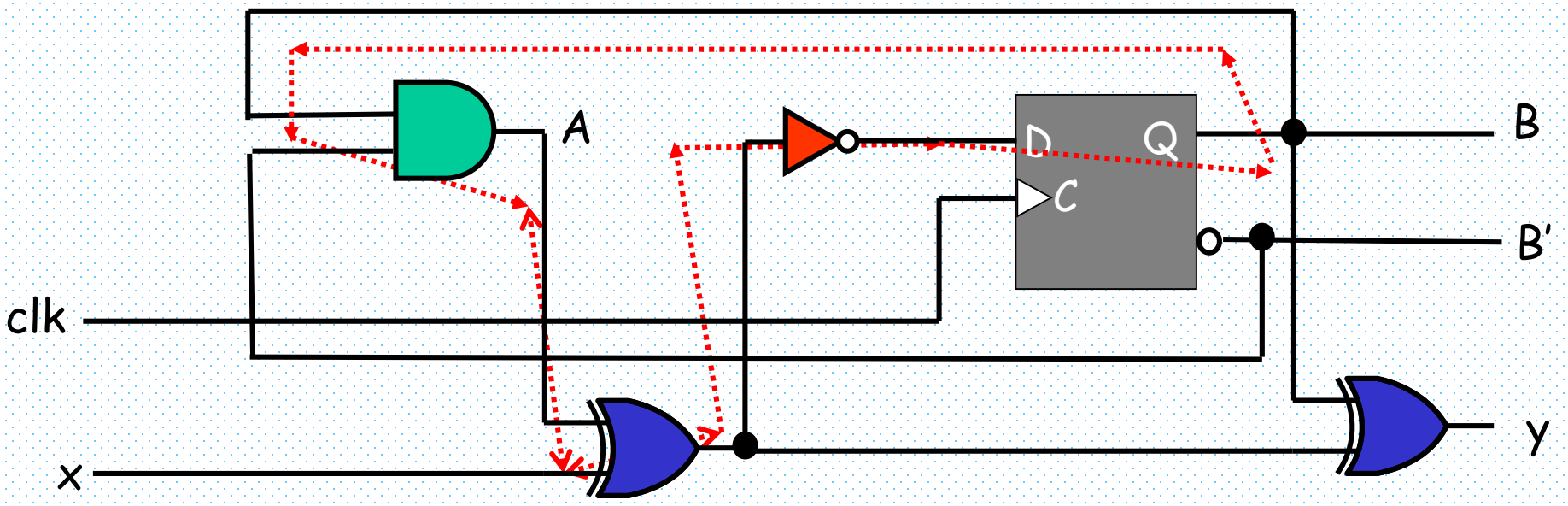
$t_{p,AND} = t_s$ = 1.0 ns

t_h = 0.25 ns

Find the longest path delay from positive clock edge to output

$t_{p,FF} + t_{p,XOR}$ = 2.0 + 2.0 = 4.0 ns

Example: Sequential Circuit Timing



$t_{p,NOT} = 0.5$ ns

$t_{p,XOR} = 2.0$ ns

$t_{p,FF} = 2.0$ ns

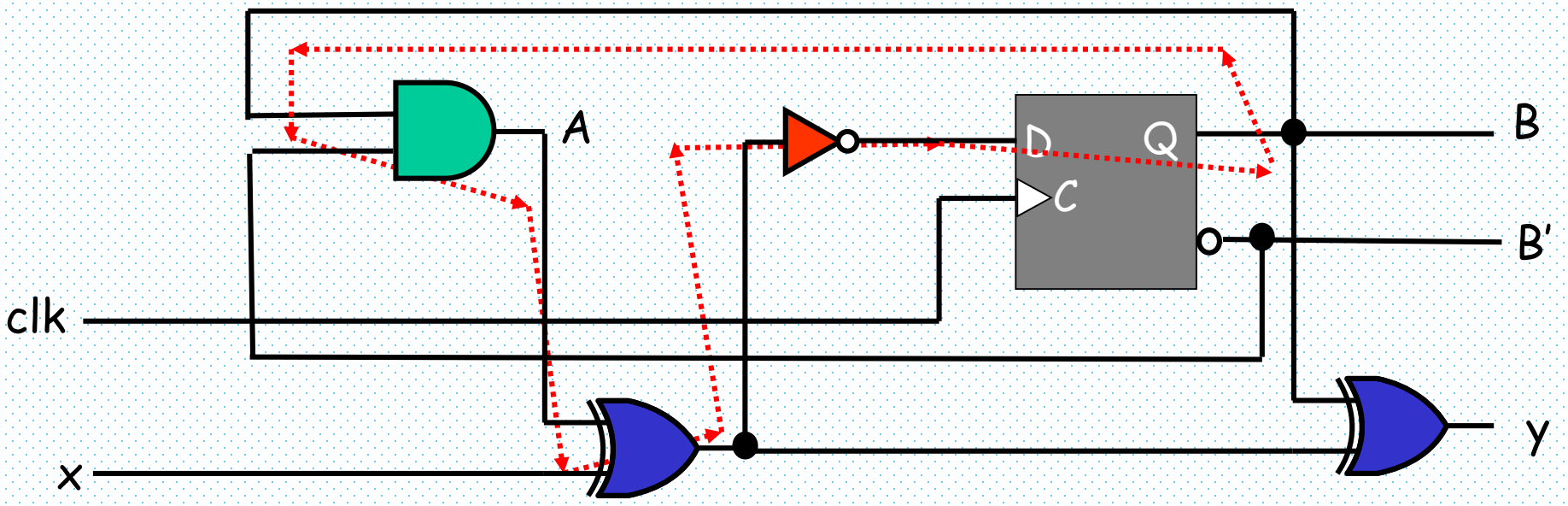
$t_{p,AND} = t_s = 1.0$ ns

$t_h = 0.25$ ns

Find the longest path delay from positive clock edge to positive clock edge

$$t_{p,FF} + t_{p,AND} + t_{p,XOR} + t_{p,NOT}$$
$$= 2.0 + 1.0 + 2.0 + 0.5 = 5.5 \text{ ns}$$

Example: Sequential Circuit Timing



$t_{p,NOT} = 0.5$ ns

$t_{p,XOR} = 2.0$ ns

$t_{p,FF} = 2.0$ ns

$t_{p,AND} = t_s = 1.0$ ns

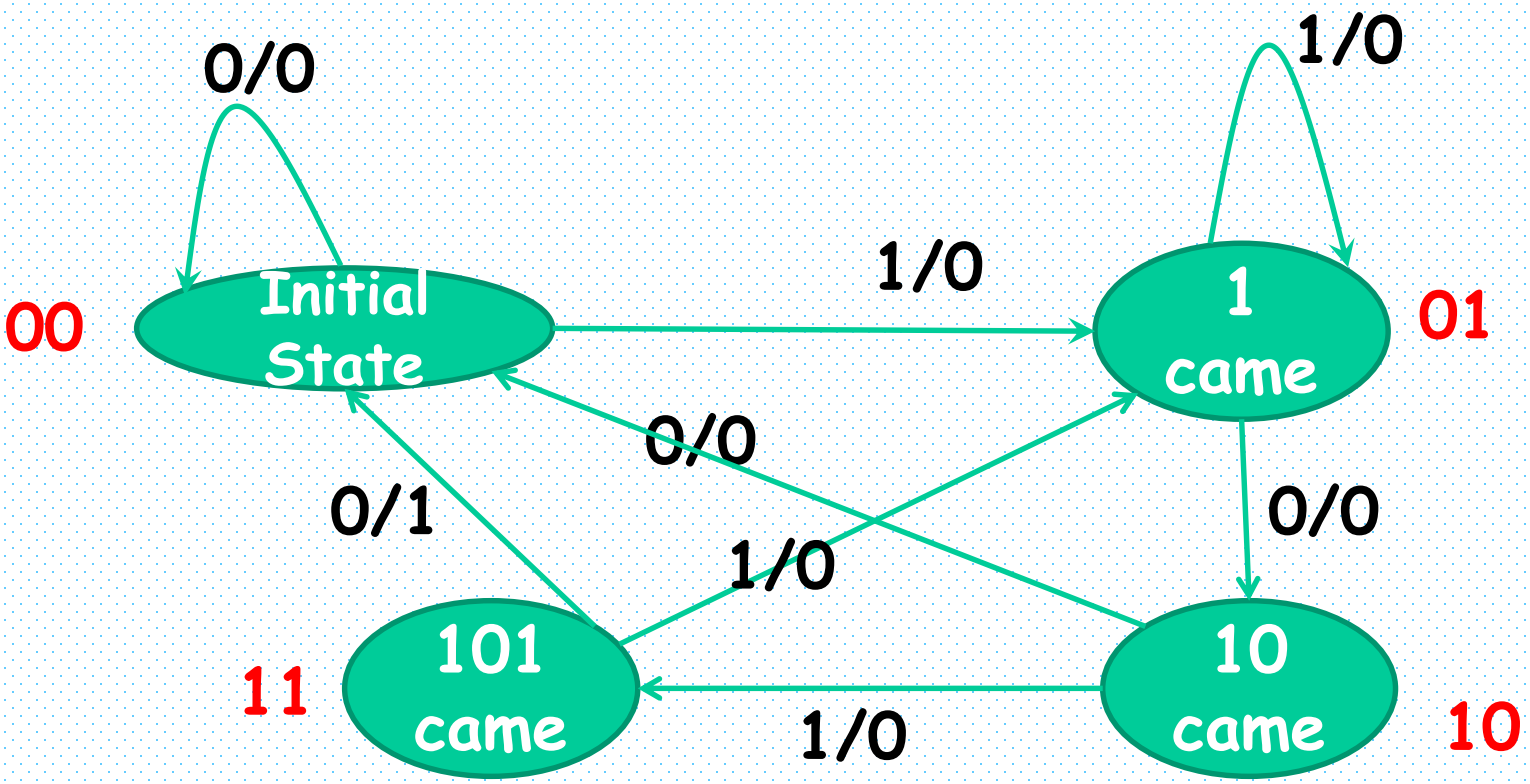
$t_h = 0.25$ ns

Determine the maximum frequency of operation of the circuit in megahertz

$$t_p = t_{p,FF} + t_{p,AND} + t_{p,XOR} + t_{p,NOT} + t_s$$
$$= 2.0 + 1.0 + 2.0 + 0.5 + 1.0 = 6.5 \text{ ns}$$
$$f_{max} = 1/t_p = 1/(6.5 \times 10^{-9}) \approx 154 \text{ MHz}$$

Design Example

- Design the synchronous sequential circuit which gives "1" as output when the last 4 values from the 1-bit input are 1010.
- Example: x= 1010 1011 ise z= 0001 0000

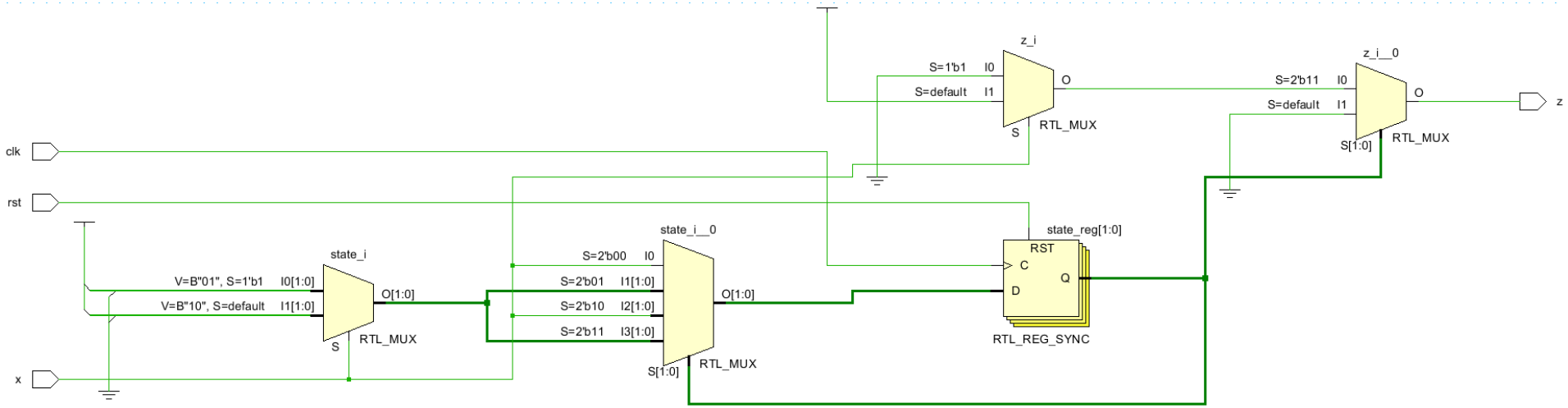


Mealy Machine

VHDL Code

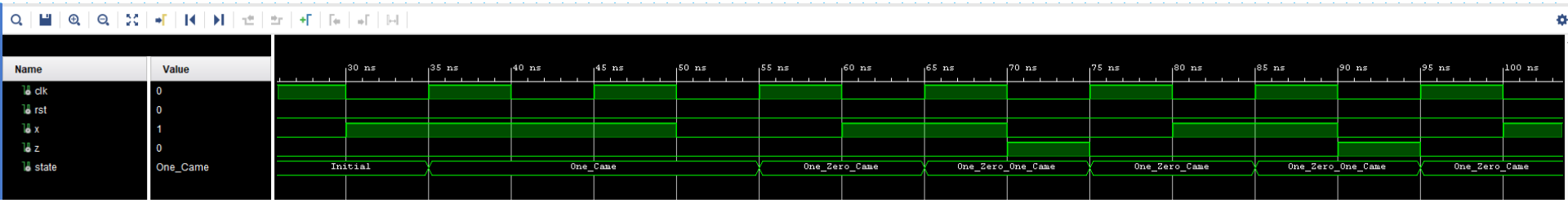
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity FSM_Mealy_1010 is
    Port ( clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           x : in STD_LOGIC;
           z : out STD_LOGIC);
end FSM_Mealy_1010;
architecture Behavioral of FSM_Mealy_1010 is
  type state_type is
   (Initial,One_Came,One_Zero_Came,One_Z
   ero_One_Came);
  signal state : state_type;
begin
  state_transition: process(clk)    begin
    if(clk'event and clk='1') then
      if(rst='1') then
        state <= Initial;
      else
        case state is
          when Initial =>
              if(x='1') then
                  state <= One_Came;
              else
                  state <= Initial;   end if;
          when One_Came =>
              if(x='1') then      state <= One_Came;
              else                state <= One_Zero_Came;
              end if;
          when One_Zero_Came =>
              if(x='1') then      state <= One_Zero_One_Came;
              else                state <= Initial;
              end if;
          when One_Zero_One_Came =>
              if(x='1') then      state <= One_Came;
              else                state <= Initial;
              end if;
        end case;
    end if;
    end if;
    end process;
    output: process(state,x)
      begin
        case state is
          when One_Zero_One_Came =>
              if(x='1') then     z  <=  '0';
              else               z  <=  '1';
              end if;
          when others =>         z  <=  '0';
        end case;
      end process;
end Behavioral;
```

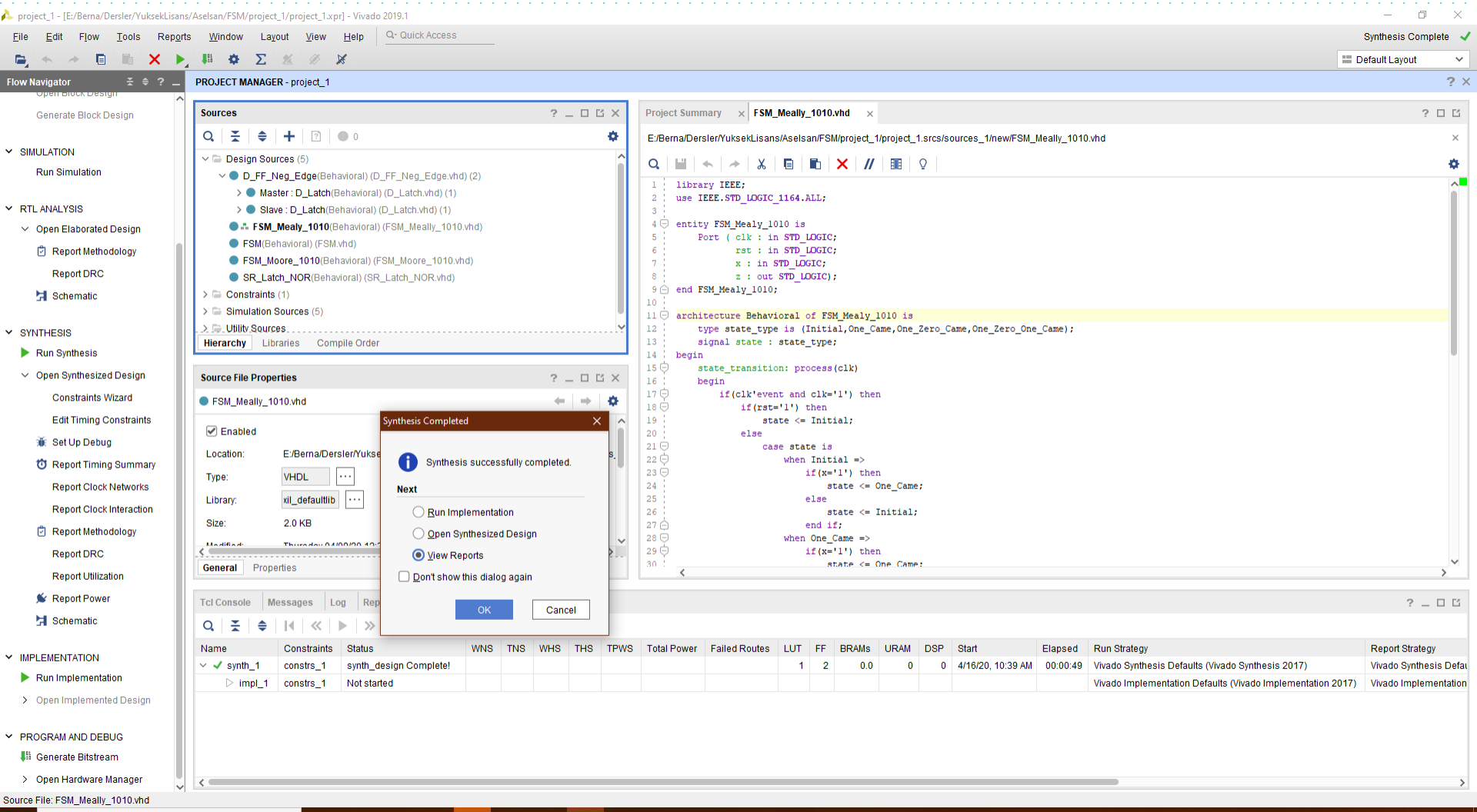
Mealy Machine RTL Schematic

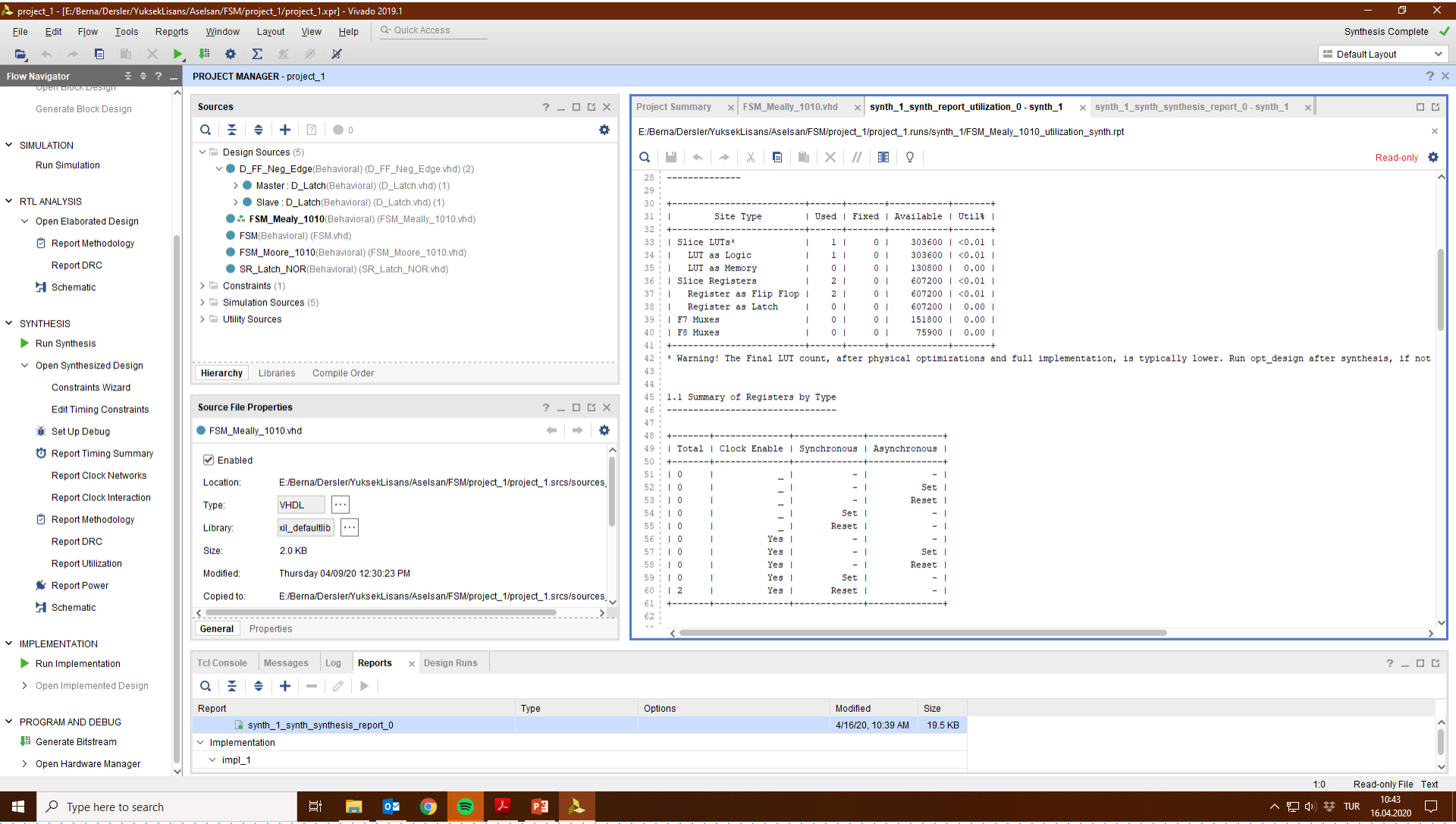


Testbench and Benaviour Simulation

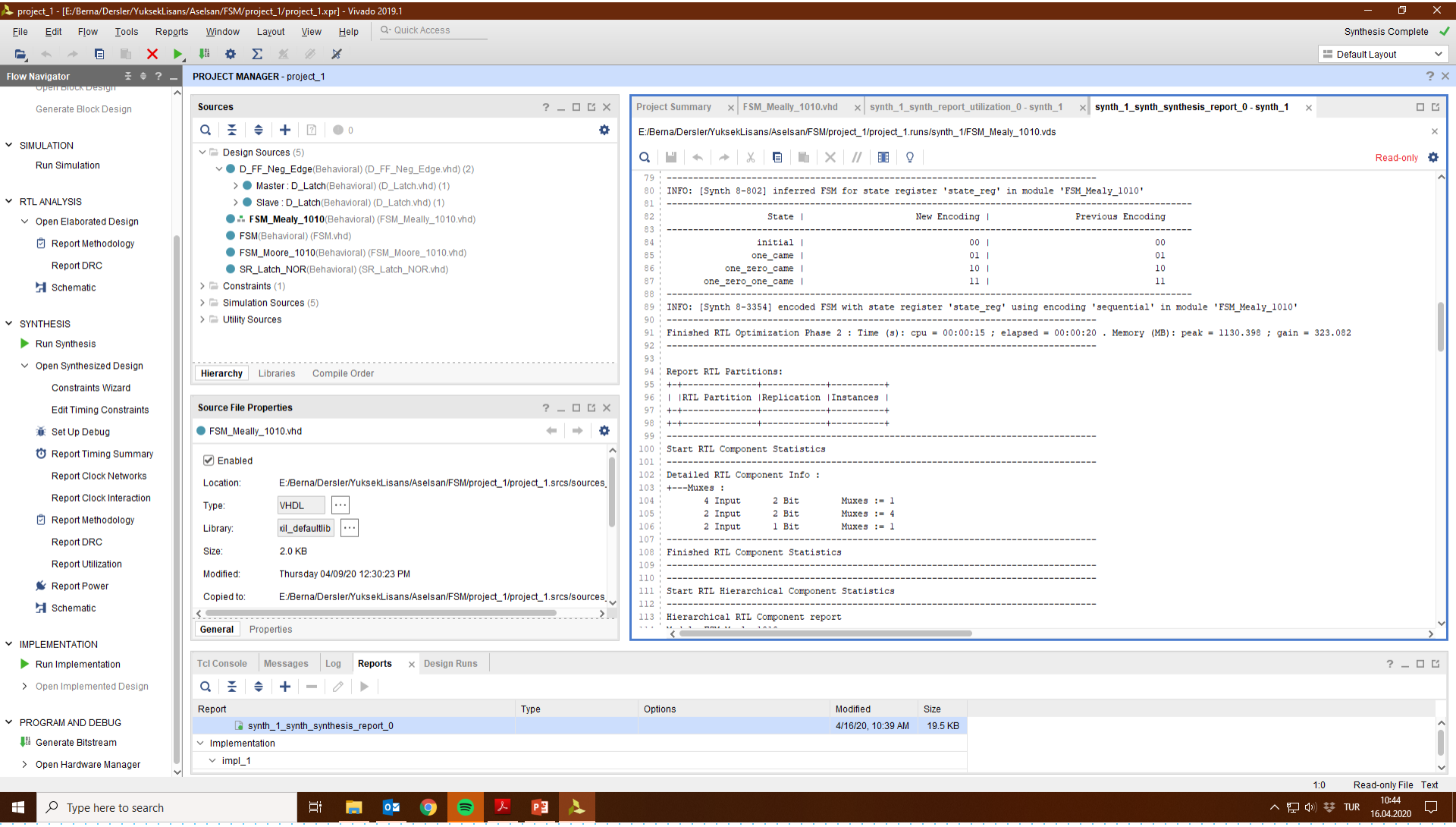
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FSM_Mealy_1010_tb is
end FSM_Mealy_1010_tb;

architecture Behavioral of
      FSM_Mealy_1010_tb is
    component FSM_Mealy_1010 is
    Port ( clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           x : in STD_LOGIC;
           z : out STD_LOGIC);
    end component;
    signal clk : STD_LOGIC :='0';
    signal rst,x,z : STD_LOGIC;
begin
DUT: FSM_Mealy_1010 Port map(clk,rst,x,z);
process
    begin
        wait for 5 ns;
        clk <= not clk;
    end process;
 process        begin
        rst <= '1';
        wait for 10 ns; rst<='0';
        wait for 10 ns; x<='0';
        wait for 10 ns; x<='1';
        wait for 10 ns; x<='1';
        wait for 10 ns; x<='0';
        wait for 10 ns; x<='1';
        wait for 10 ns; x<='0';
        wait for 10 ns; x<='1';
        wait for 10 ns; x<='0';
        wait for 10 ns; x<='1';
        wait for 10 ns; x<='1';
        wait for 10 ns; x<='1';
end process;
end Behavioral;
```

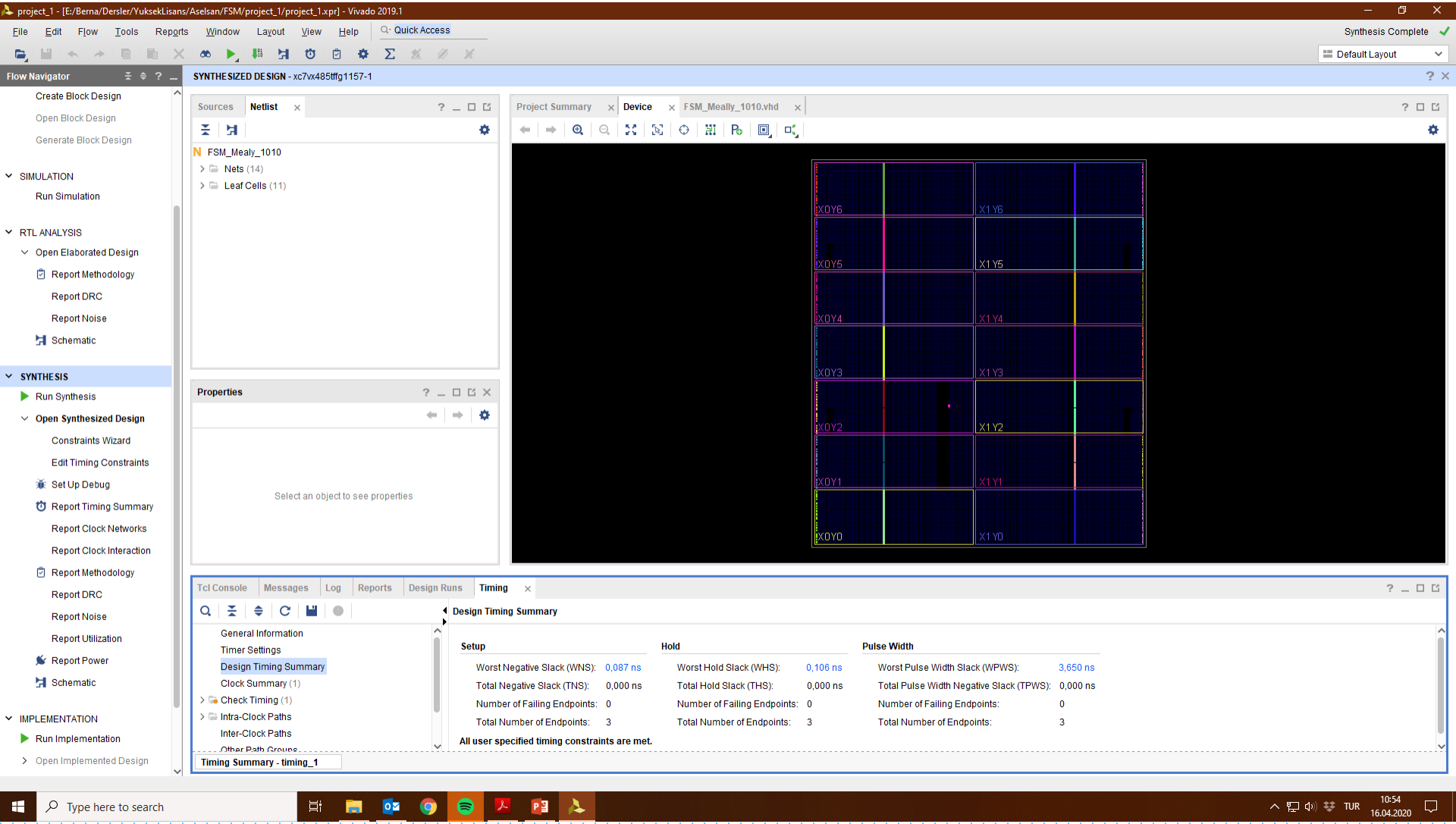


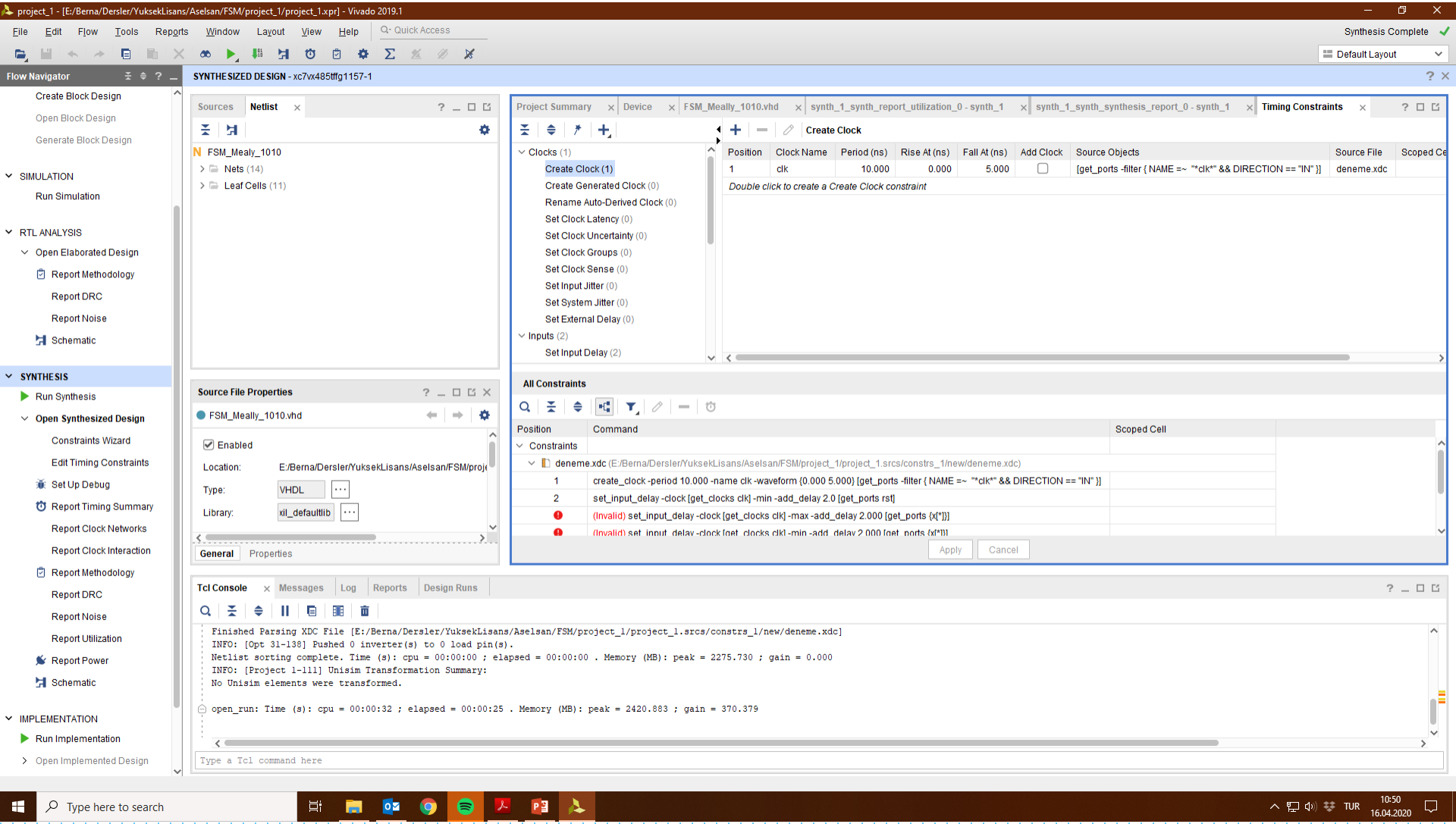

project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Synthesis Complete
Default Layout
Flow Navigator
PROJECT MANAGER - project_1
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
Sources
Design Sources (5)
D_FF_Neg_Edge(Behavioral) (D_FF_Neg_Edge.vhd) (2)
Master : D_Latch(Behavioral) (D_Latch.vhd) (1)
Slave : D_Latch(Behavioral) (D_Latch.vhd) (1)
FSM_Mealy_1010(Behavioral) (FSM_Mealy_1010.vhd)
FSM(Behavioral) (FSM.vhd)
FSM_Moore_1010(Behavioral) (FSM_Moore_1010.vhd)
SR_Latch_NOR(Behavioral) (SR_Latch_NOR.vhd)
Constraints (1)
Simulation Sources (5)
Utility Sources
Hierarchy Libraries Compile Order
Source File Properties
FSM_Mealy_1010.vhd
Enabled
Location: E:/Berna/Dersler/Yukse
Type: VHDL
Library: xil_defaultlib
Size: 2.0 KB
General Properties
Synthesis Completed
Synthesis successfully completed.
Next
Run Implementation
Open Synthesized Design
View Reports
Don't show this dialog again
OK
Cancel
Project Summary
FSM_Mealy_1010.vhd
E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.srcs/sources_1/new/FSM_Mealy_1010.vhd
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity FSM_Mealy_1010 is
5 Port (clk : in STD_LOGIC;
6 rst : in STD_LOGIC;
7 x : in STD_LOGIC;
8 z : out STD_LOGIC);
9 end FSM_Mealy_1010;
10
11 architecture Behavioral of FSM_Mealy_1010 is
12 type state_type is (Initial,One_Came,One_Zero_Came,One_Zero_One_Came);
13 signal state : state_type;
14 begin
15 state_transition: process(clk)
16 begin
17 if(clk'event and clk='1') then
18 if(rst='1') then
19 state <= Initial;
20 else
21 case state is
22 when Initial =>
23 if(x='1') then
24 state <= One_Came;
25 else
26 state <= Initial;
27 end if;
28 when One_Came =>
29 if(x='1') then
Tcl Console Messages Log
Name Constraints Status WNS TNS WHS THS TPWS Total Power Failed Routes LUT FF BRAMs URAM DSP Start Elapsed Run Strategy Report Strategy
synth_1 constrs_1 synth_design Complete! 1 2 0.0 0 0 4/16/20, 10:39 AM 00:00:49 Vivado Synthesis Defaults (Vivado Synthesis 2017) Vivado Synthesis Defau
impl_1 constrs_1 Not started Vivado Implementation Defaults (Vivado Implementation 2017) Vivado Implementation
Source File: FSM_Mealy_1010.vhd
Type here to search
TUR
10:40
16.04.2020

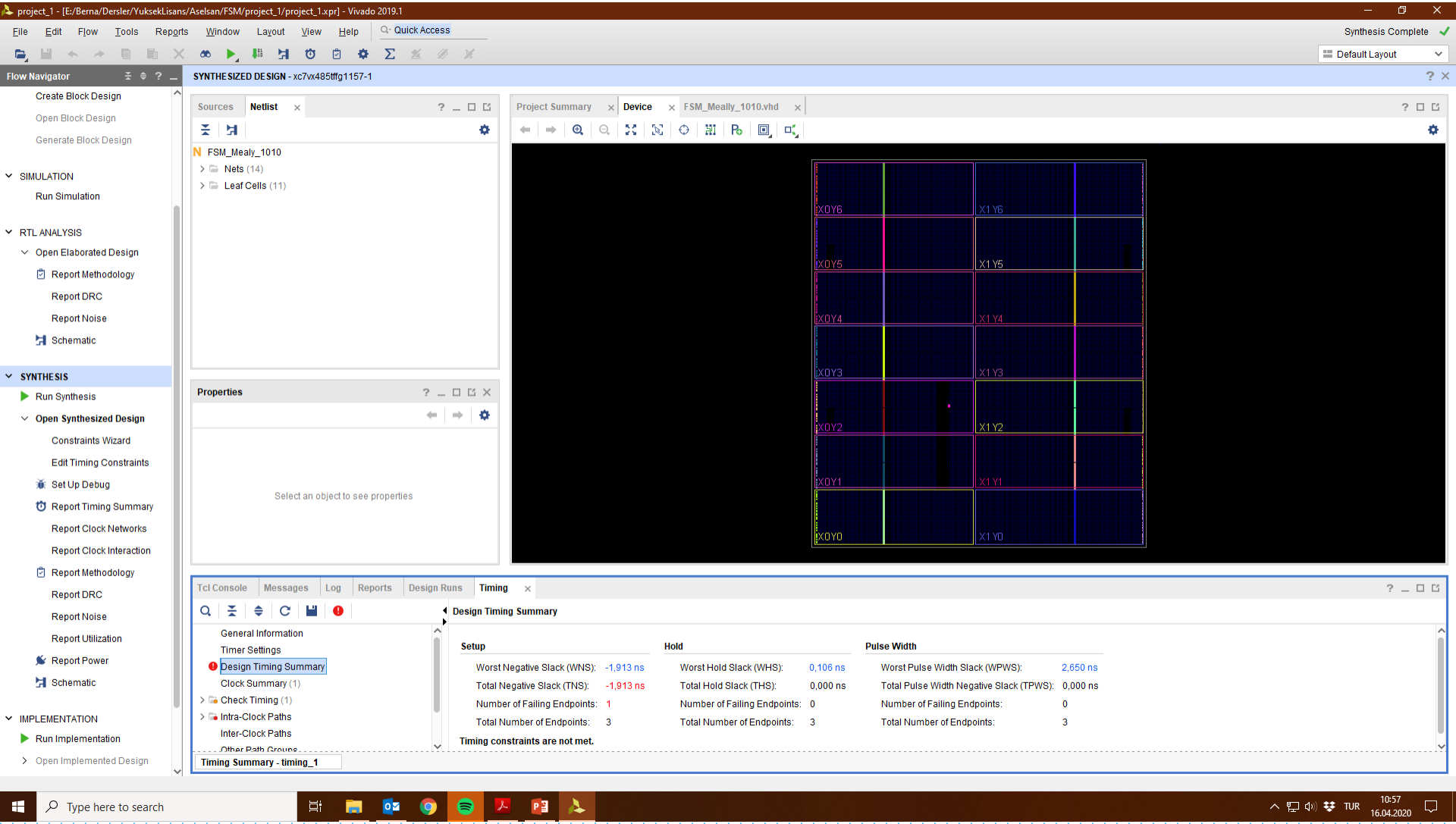


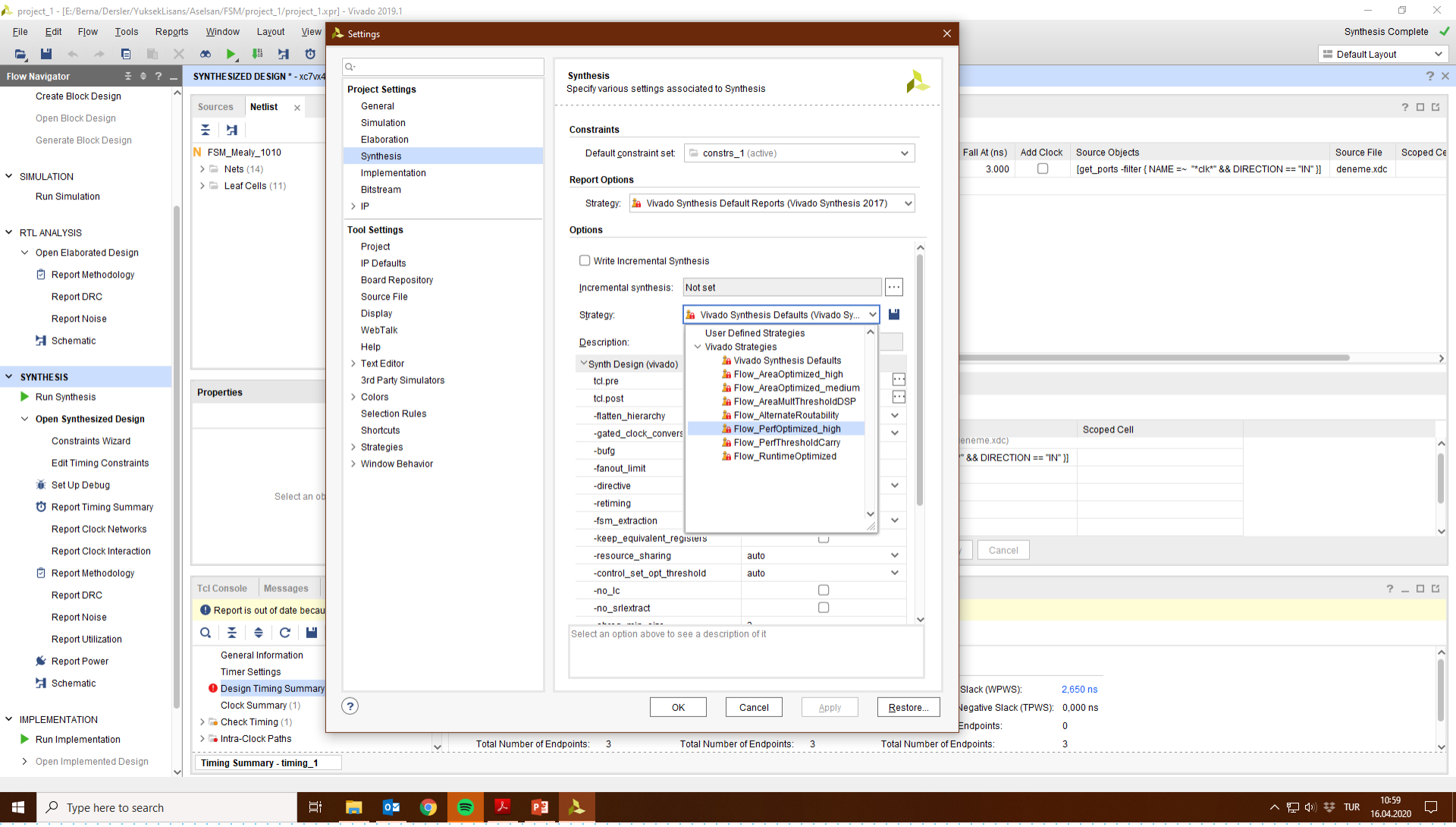

```
--------------

+-------------------------+------+-------+-----------+-------+
|        Site Type        | Used | Fixed | Available | Util% |
+-------------------------+------+-------+-----------+-------+
| Slice LUTs*             |    1 |     0 |    303600 | <0.01 |
|   LUT as Logic          |    1 |     0 |    303600 | <0.01 |
|   LUT as Memory         |    0 |     0 |    130800 |  0.00 |
| Slice Registers         |    2 |     0 |    607200 | <0.01 |
|   Register as Flip Flop |    2 |     0 |    607200 | <0.01 |
|   Register as Latch     |    0 |     0 |    607200 |  0.00 |
| F7 Muxes                |    0 |     0 |    151800 |  0.00 |
| F8 Muxes                |    0 |     0 |     75900 |  0.00 |
+-------------------------+------+-------+-----------+-------+
* Warning! The Final LUT count, after physical optimizations and full implementation, is typically lower. Run opt_design after synthesis, if not


1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |            _ |           - |            - |
| 0     |            _ |           - |          Set |
| 0     |            _ |           - |        Reset |
| 0     |            _ |         Set |            - |
| 0     |            _ |       Reset |            - |
| 0     |          Yes |           - |            - |
| 0     |          Yes |           - |          Set |
| 0     |          Yes |           - |        Reset |
| 0     |          Yes |         Set |            - |
| 2     |          Yes |       Reset |            - |
+-------+--------------+-------------+--------------+
```

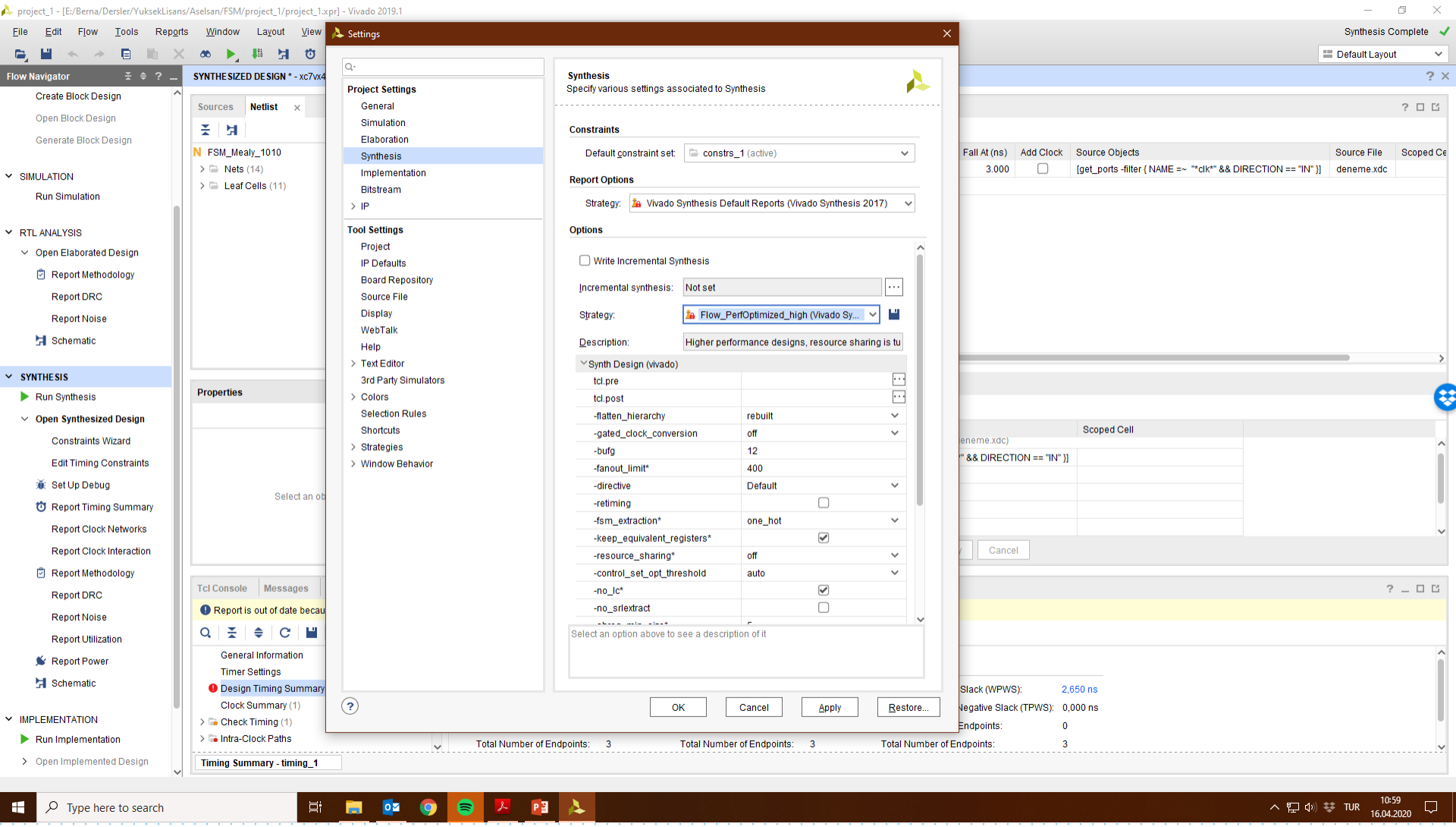

project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Synthesis Complete
Default Layout
Flow Navigator
PROJECT MANAGER - project_1
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
PROGRAM AND DEBUG
Generate Bitstream
Open Hardware Manager
Sources
Design Sources (5)
D_FF_Neg_Edge(Behavioral) (D_FF_Neg_Edge.vhd) (2)
Master : D_Latch(Behavioral) (D_Latch.vhd) (1)
Slave : D_Latch(Behavioral) (D_Latch.vhd) (1)
FSM_Mealy_1010(Behavioral) (FSM_Mealy_1010.vhd)
FSM(Behavioral) (FSM.vhd)
FSM_Moore_1010(Behavioral) (FSM_Moore_1010.vhd)
SR_Latch_NOR(Behavioral) (SR_Latch_NOR.vhd)
Constraints (1)
Simulation Sources (5)
Utility Sources
Hierarchy Libraries Compile Order
Source File Properties
FSM_Mealy_1010.vhd
Enabled
Location: E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.srcs/sources_
Type: VHDL
Library: xil_defaultlib
Size: 2.0 KB
Modified: Thursday 04/09/20 12:30:23 PM
Copied to: E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.srcs/sources_
General Properties
Project Summary
FSM_Mealy_1010.vhd
synth_1_synth_report_utilization_0 - synth_1
synth_1_synth_synthesis_report_0 - synth_1
E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.runs/synth_1/FSM_Mealy_1010.vds
Read-only
INFO: [Synth 8-802] inferred FSM for state register 'state_reg' in module 'FSM_Mealy_1010'
State | New Encoding | Previous Encoding
initial | 00 | 00
one_came | 01 | 01
one_zero_came | 10 | 10
one_zero_one_came | 11 | 11
INFO: [Synth 8-3354] encoded FSM with state register 'state_reg' using encoding 'sequential' in module 'FSM_Mealy_1010'
Finished RTL Optimization Phase 2 : Time (s): cpu = 00:00:15 ; elapsed = 00:00:20 . Memory (MB): peak = 1130.398 ; gain = 323.082
Report RTL Partitions:
|RTL Partition |Replication |Instances |
Start RTL Component Statistics
Detailed RTL Component Info :
+---Muxes :
4 Input 2 Bit Muxes := 1
2 Input 2 Bit Muxes := 4
2 Input 1 Bit Muxes := 1
Finished RTL Component Statistics
Start RTL Hierarchical Component Statistics
Hierarchical RTL Component report
Tcl Console Messages Log Reports Design Runs
Report Type Options Modified Size
synth_1_synth_synthesis_report_0 4/16/20, 10:39 AM 19.5 KB
Implementation
impl_1
1:0 Read-only File Text
Type here to search
TUR 10:44 16.04.2020

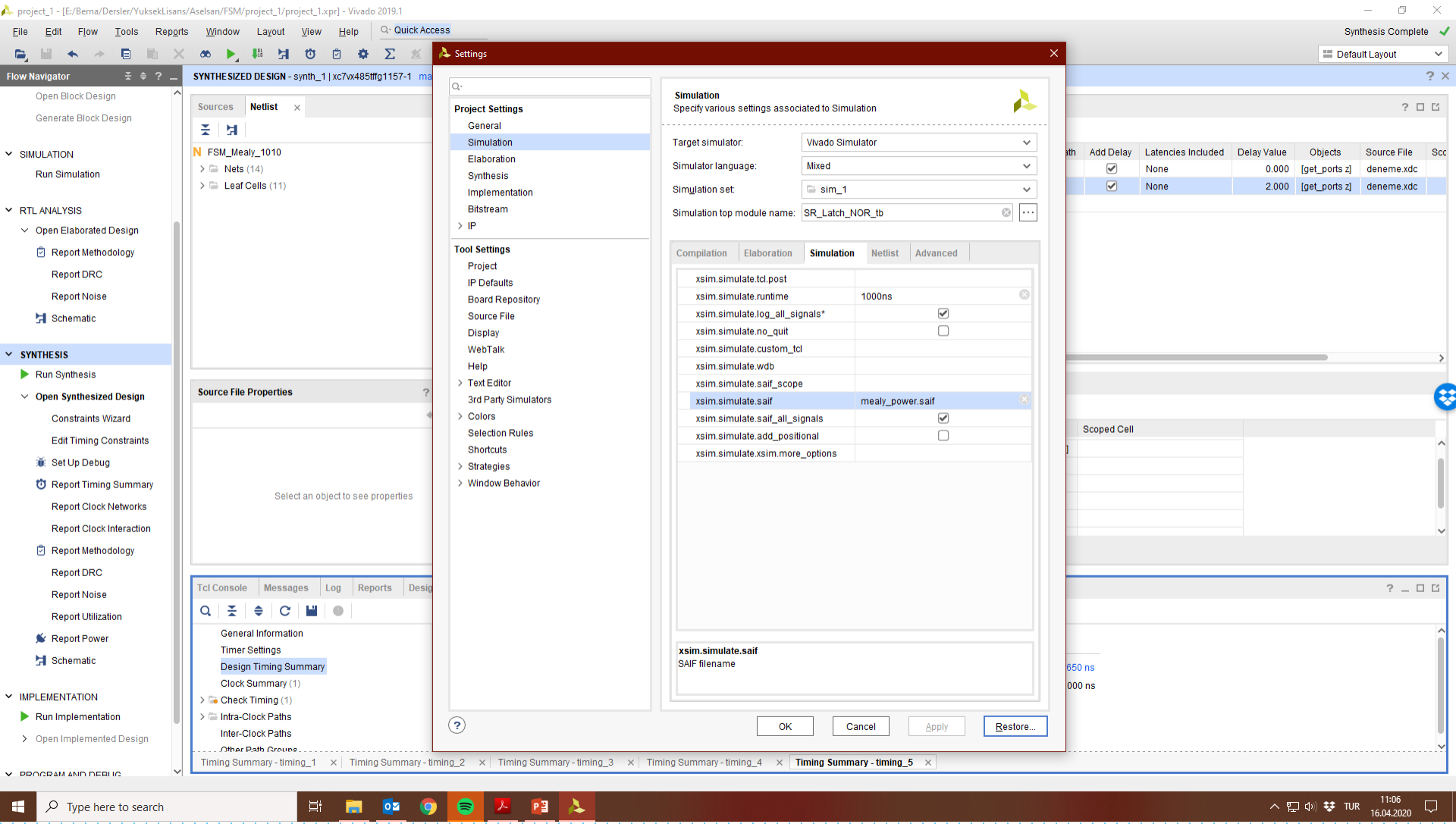

project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Synthesis Complete
Default Layout
Flow Navigator
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
SYNTHESIZED DESIGN - xc7vx485tffg1157-1
Sources
Netlist
FSM_Mealy_1010
Nets (14)
Leaf Cells (11)
Properties
Select an object to see properties
Project Summary
Device
FSM_Mealy_1010.vhd
X0Y6
X1 Y6
X0Y5
X1 Y5
X0Y4
X1 Y4
X0Y3
X1 Y3
X0Y2
X1 Y2
X0Y1
X1 Y1
X0Y0
X1 Y0
Tcl Console
Messages
Log
Reports
Design Runs
Timing
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (1)
Intra-Clock Paths
Inter-Clock Paths
Design Timing Summary
Setup
Worst Negative Slack (WNS): 0,087 ns
Total Negative Slack (TNS): 0,000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 3
Hold
Worst Hold Slack (WHS): 0,106 ns
Total Hold Slack (THS): 0,000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 3
Pulse Width
Worst Pulse Width Slack (WPWS): 3,650 ns
Total Pulse Width Negative Slack (TPWS): 0,000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 3
All user specified timing constraints are met.
Timing Summary - timing_1
Type here to search
TUR
10:54
16.04.2020

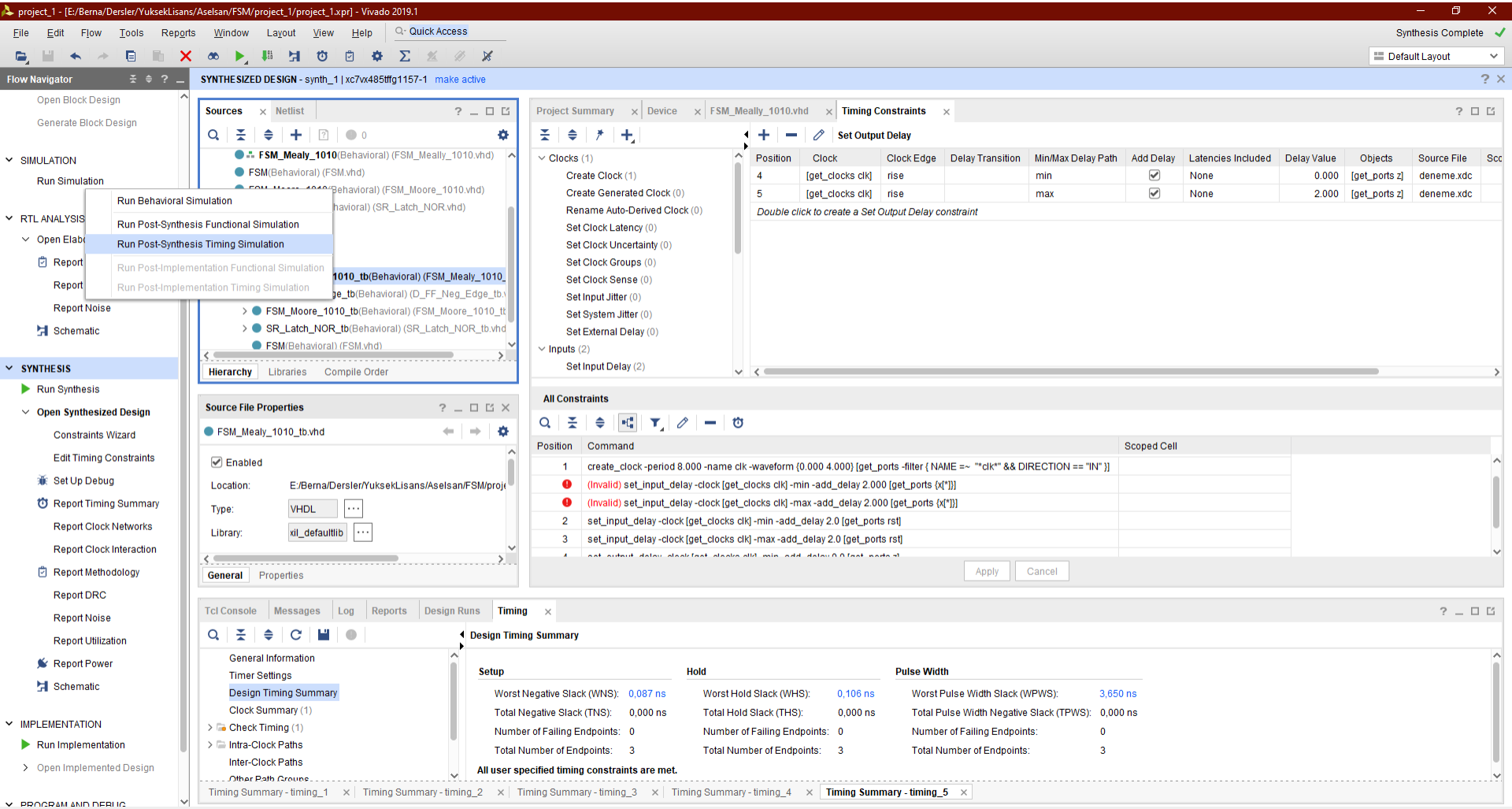

project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Synthesis Complete
Default Layout
Flow Navigator
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
SYNTHESIZED DESIGN - xc7vx485tffg1157-1
Sources
Netlist
FSM_Mealy_1010
Nets (14)
Leaf Cells (11)
Source File Properties
FSM_Mealy_1010.vhd
Enabled
Location: E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/proj
Type: VHDL
Library: xil_defaultlib
General
Properties
Project Summary
Device
FSM_Mealy_1010.vhd
synth_1_synth_report_utilization_0 - synth_1
synth_1_synth_synthesis_report_0 - synth_1
Timing Constraints
Create Clock
Clocks (1)
Create Clock (1)
Create Generated Clock (0)
Rename Auto-Derived Clock (0)
Set Clock Latency (0)
Set Clock Uncertainty (0)
Set Clock Groups (0)
Set Clock Sense (0)
Set Input Jitter (0)
Set System Jitter (0)
Set External Delay (0)
Inputs (2)
Set Input Delay (2)
Position Clock Name Period (ns) Rise At (ns) Fall At (ns) Add Clock Source Objects Source File Scoped Ce
1 clk 10.000 0.000 5.000 [get_ports -filter { NAME =~ "*clk*" && DIRECTION == "IN" }] deneme.xdc
Double click to create a Create Clock constraint
All Constraints
Position Command Scoped Cell
Constraints
deneme.xdc (E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.srcs/constrs_1/new/deneme.xdc)
1 create_clock -period 10.000 -name clk -waveform {0.000 5.000} [get_ports -filter { NAME =~ "*clk*" && DIRECTION == "IN" }]
2 set_input_delay -clock [get_clocks clk] -min -add_delay 2.0 [get_ports rst]
(Invalid) set_input_delay -clock [get_clocks clk] -max -add_delay 2.000 [get_ports {x[*]}]
Apply
Cancel
Tcl Console
Messages
Log
Reports
Design Runs
Finished Parsing XDC File [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.srcs/constrs_1/new/deneme.xdc]
INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elapsed = 00:00:00 . Memory (MB): peak = 2275.730 ; gain = 0.000
INFO: [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.
open_run: Time (s): cpu = 00:00:32 ; elapsed = 00:00:25 . Memory (MB): peak = 2420.883 ; gain = 370.379
Type a Tcl command here
Type here to search
TUR
10:50
16.04.2020

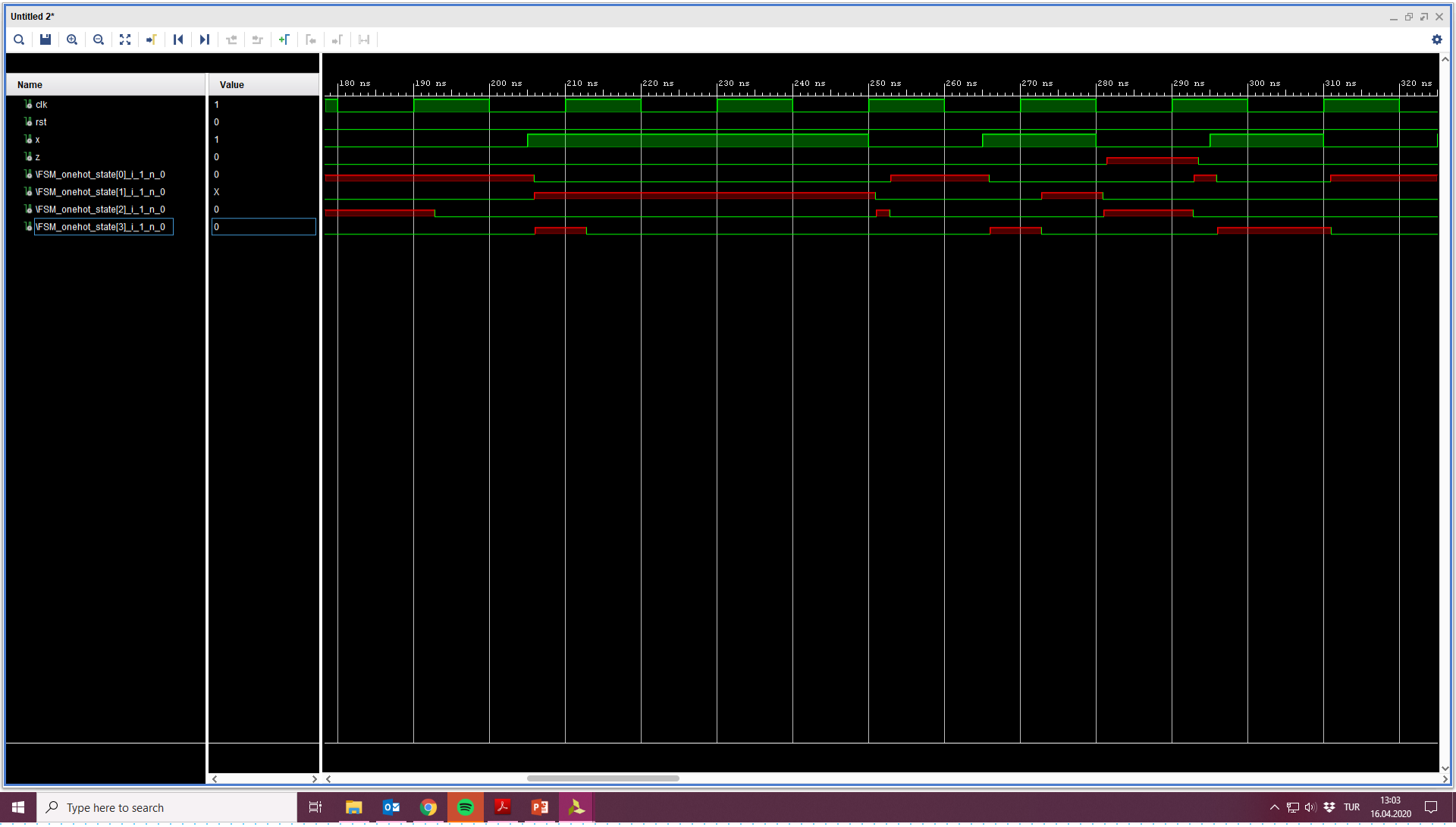

project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Synthesis Complete
Default Layout
Flow Navigator
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
SYNTHESIZED DESIGN - xc7vx485tffg1157-1
Sources
Netlist
FSM_Mealy_1010
Nets (14)
Leaf Cells (11)
Properties
Select an object to see properties
Project Summary
Device
FSM_Mealy_1010.vhd
X0Y6
X1 Y6
X0Y5
X1 Y5
X0Y4
X1 Y4
X0Y3
X1 Y3
X0Y2
X1 Y2
X0Y1
X1 Y1
X0Y0
X1 Y0
Tcl Console
Messages
Log
Reports
Design Runs
Timing
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (1)
Intra-Clock Paths
Inter-Clock Paths
Design Timing Summary
Setup
Worst Negative Slack (WNS): -1,913 ns
Total Negative Slack (TNS): -1,913 ns
Number of Failing Endpoints: 1
Total Number of Endpoints: 3
Hold
Worst Hold Slack (WHS): 0,106 ns
Total Hold Slack (THS): 0,000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 3
Pulse Width
Worst Pulse Width Slack (WPWS): 2,650 ns
Total Pulse Width Negative Slack (TPWS): 0,000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 3
Timing constraints are not met.
Timing Summary - timing_1
Type here to search
TUR
10:57
16.04.2020

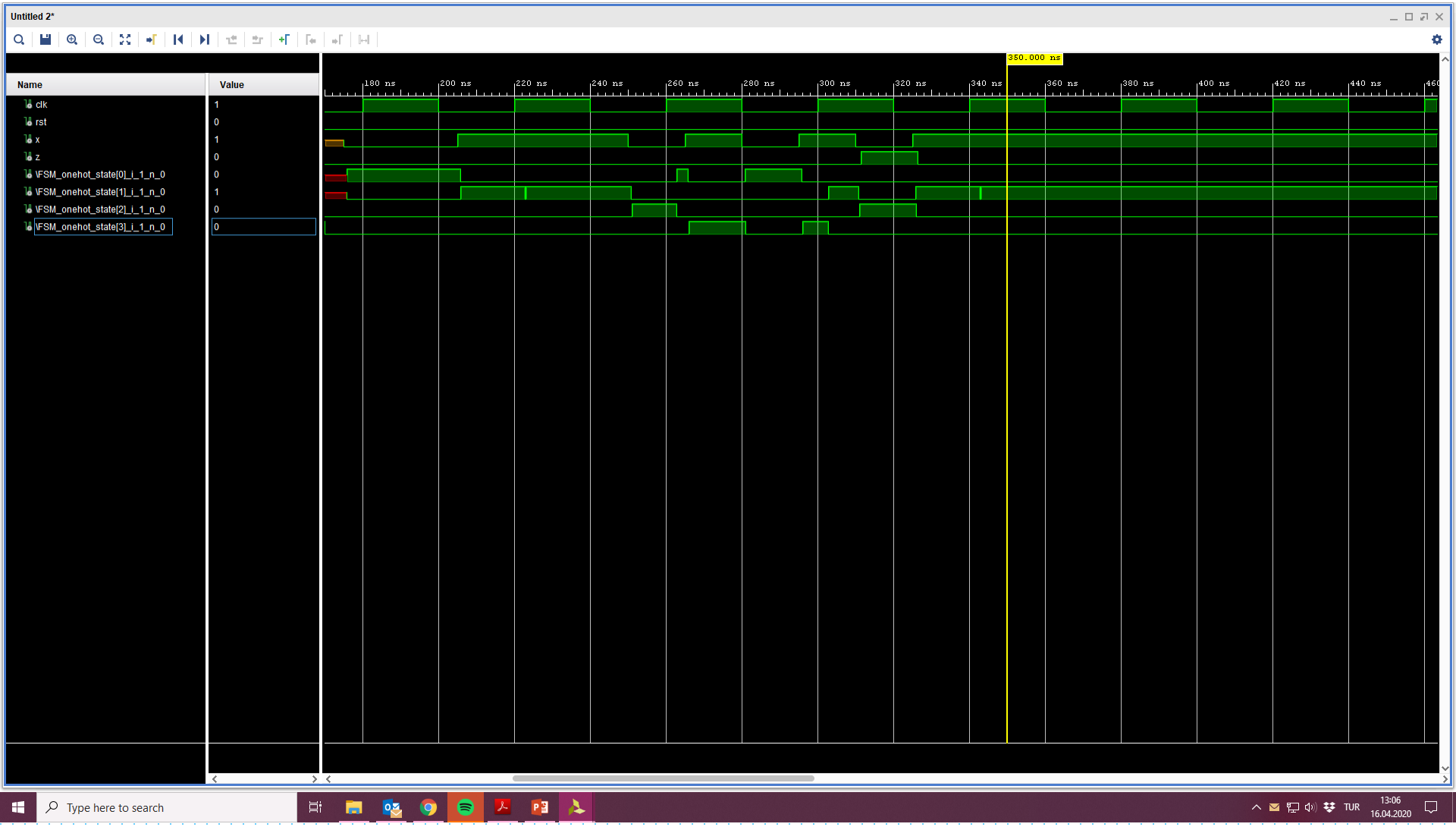

project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View
Synthesis Complete
Default Layout
Flow Navigator
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
SYNTHESIZED DESIGN * - xc7vx4
Sources
Netlist
FSM_Mealy_1010
Nets (14)
Leaf Cells (11)
Properties
Tcl Console
Messages
Report is out of date becau
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (1)
Intra-Clock Paths
Timing Summary - timing_1
Total Number of Endpoints: 3
Total Number of Endpoints: 3
Total Number of Endpoints: 3
Fall At (ns)
Add Clock
Source Objects
Source File
Scoped Ce
3.000
[get_ports -filter { NAME =~ "*clk*" && DIRECTION == "IN" }]
deneme.xdc
Scoped Cell
Cancel
Slack (WPWS): 2,650 ns
Negative Slack (TPWS): 0,000 ns
Endpoints: 0
Settings
Project Settings
General
Simulation
Elaboration
Synthesis
Implementation
Bitstream
IP
Tool Settings
Project
IP Defaults
Board Repository
Source File
Display
WebTalk
Help
Text Editor
3rd Party Simulators
Colors
Selection Rules
Shortcuts
Strategies
Window Behavior
Synthesis
Specify various settings associated to Synthesis
Constraints
Default constraint set: constrs_1 (active)
Report Options
Strategy: Vivado Synthesis Default Reports (Vivado Synthesis 2017)
Options
Write Incremental Synthesis
Incremental synthesis: Not set
Strategy: Vivado Synthesis Defaults (Vivado Sy...
Description:
User Defined Strategies
Vivado Strategies
Vivado Synthesis Defaults
Flow_AreaOptimized_high
Flow_AreaOptimized_medium
Flow_AreaMultThresholdDSP
Flow_AlternateRoutability
Flow_PerfOptimized_high
Flow_PerfThresholdCarry
Flow_RuntimeOptimized
Synth Design (vivado)
tcl.pre
tcl.post
-flatten_hierarchy
-gated_clock_convers
-bufg
-fanout_limit
-directive
-retiming
-fsm_extraction
-keep_equivalent_registers
-resource_sharing auto
-control_set_opt_threshold auto
-no_lc
-no_srlextract
Select an option above to see a description of it
OK
Cancel
Apply
Restore...
Type here to search
TUR
10:59
16.04.2020

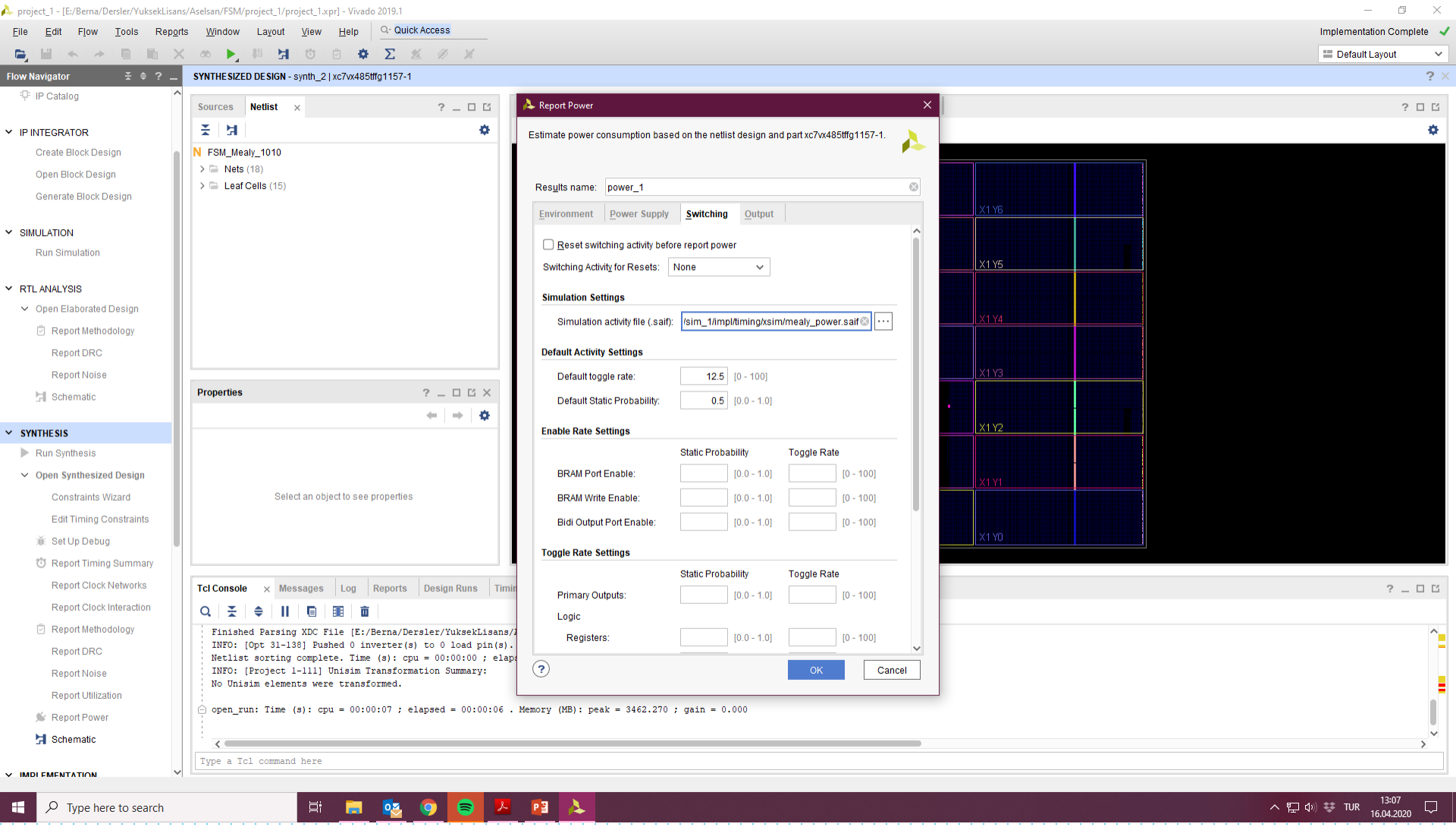

project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View
Synthesis Complete
Default Layout
Flow Navigator
SYNTHESIZED DESIGN * - xc7vx4
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
Sources
Netlist
FSM_Mealy_1010
Nets (14)
Leaf Cells (11)
Properties
Tcl Console
Messages
Report is out of date becau
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (1)
Intra-Clock Paths
Timing Summary - timing_1
Fall At (ns)
Add Clock
Source Objects
Source File
Scoped Ce
3.000
[get_ports -filter { NAME =~ "*clk*" && DIRECTION == "IN" }]
deneme.xdc
Scoped Cell
Cancel
Slack (WPWS): 2,650 ns
Negative Slack (TPWS): 0,000 ns
Endpoints: 0
Total Number of Endpoints: 3
Total Number of Endpoints: 3
Total Number of Endpoints: 3
Settings
Project Settings
General
Simulation
Elaboration
Synthesis
Implementation
Bitstream
IP
Tool Settings
Project
IP Defaults
Board Repository
Source File
Display
WebTalk
Help
Text Editor
3rd Party Simulators
Colors
Selection Rules
Shortcuts
Strategies
Window Behavior
Synthesis
Specify various settings associated to Synthesis
Constraints
Default constraint set: constrs_1 (active)
Report Options
Strategy: Vivado Synthesis Default Reports (Vivado Synthesis 2017)
Options
Write Incremental Synthesis
Incremental synthesis: Not set
Strategy: Flow_PerfOptimized_high (Vivado Sy...
Description: Higher performance designs, resource sharing is tu
Synth Design (vivado)
tcl.pre
tcl.post
-flatten_hierarchy rebuilt
-gated_clock_conversion off
-bufg 12
-fanout_limit* 400
-directive Default
-retiming
-fsm_extraction* one_hot
-keep_equivalent_registers*
-resource_sharing* off
-control_set_opt_threshold auto
-no_lc*
-no_srlextract
Select an option above to see a description of it
OK
Cancel
Apply
Restore...
Type here to search
TUR
10:59
16.04.2020

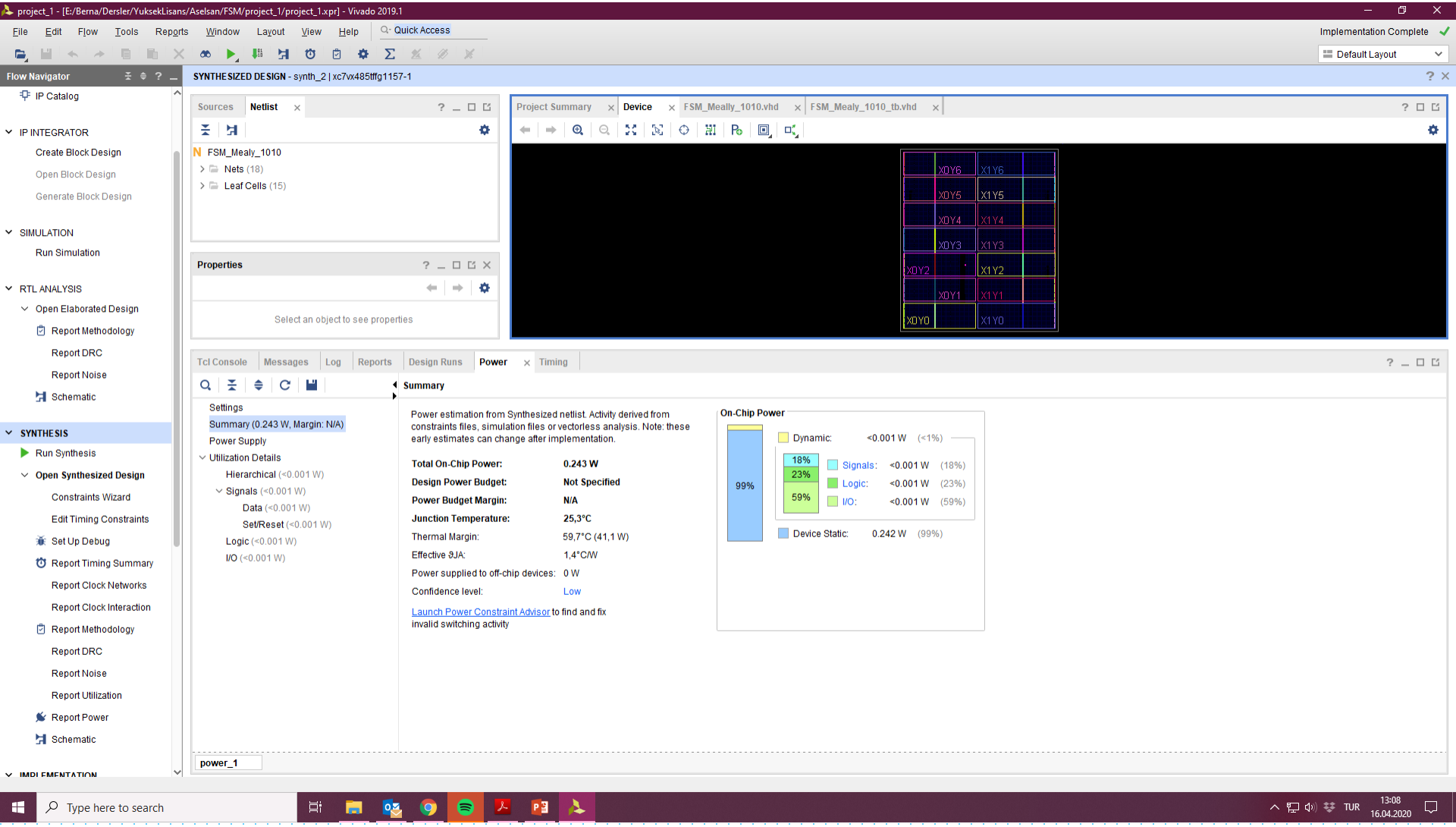

project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Synthesis Complete
Default Layout
Flow Navigator
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
SYNTHESIZED DESIGN - synth_1 | xc7vx485tffg1157-1
Sources
Netlist
FSM_Mealy_1010
Nets (14)
Leaf Cells (11)
Source File Properties
Select an object to see properties
Tcl Console
Messages
Log
Reports
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (1)
Intra-Clock Paths
Inter-Clock Paths
Add Delay
Latencies Included
Delay Value
Objects
Source File
None
0.000
[get_ports z]
deneme.xdc
None
2.000
[get_ports z]
deneme.xdc
Scoped Cell
Timing Summary - timing_1
Timing Summary - timing_2
Timing Summary - timing_3
Timing Summary - timing_4
Timing Summary - timing_5
Settings
Project Settings
General
Simulation
Elaboration
Synthesis
Implementation
Bitstream
IP
Tool Settings
Project
IP Defaults
Board Repository
Source File
Display
WebTalk
Help
Text Editor
3rd Party Simulators
Colors
Selection Rules
Shortcuts
Strategies
Window Behavior
Simulation
Specify various settings associated to Simulation
Target simulator: Vivado Simulator
Simulator language: Mixed
Simulation set: sim_1
Simulation top module name: SR_Latch_NOR_tb
Compilation
Elaboration
Simulation
Netlist
Advanced
xsim.simulate.tcl.post
xsim.simulate.runtime 1000ns
xsim.simulate.log_all_signals*
xsim.simulate.no_quit
xsim.simulate.custom_tcl
xsim.simulate.wdb
xsim.simulate.saif_scope
xsim.simulate.saif mealy_power.saif
xsim.simulate.saif_all_signals
xsim.simulate.add_positional
xsim.simulate.xsim.more_options
xsim.simulate.saif
SAIF filename
OK
Cancel
Apply
Restore...
650 ns
000 ns
Type here to search
TUR
11:06
16.04.2020


project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Synthesis Complete
Default Layout
Flow Navigator
SYNTHESIZED DESIGN - synth_1 | xc7vx485tffg1157-1 make active
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
Run Implementation
Open Implemented Design
Run Behavioral Simulation
Run Post-Synthesis Functional Simulation
Run Post-Synthesis Timing Simulation
Run Post-Implementation Functional Simulation
Run Post-Implementation Timing Simulation
Sources
Netlist
FSM_Mealy_1010(Behavioral) (FSM_Mealy_1010.vhd)
FSM(Behavioral) (FSM.vhd)
FSM_Moore_1010_tb(Behavioral) (FSM_Moore_1010_tb.vhd)
SR_Latch_NOR_tb(Behavioral) (SR_Latch_NOR_tb.vhd)
FSM(Behavioral) (FSM.vhd)
Hierarchy Libraries Compile Order
Source File Properties
FSM_Mealy_1010_tb.vhd
Enabled
Location: E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/proje
Type: VHDL
Library: xil_defaultlib
General Properties
Project Summary
Device
FSM_Mealy_1010.vhd
Timing Constraints
Set Output Delay
Clocks (1)
Create Clock (1)
Create Generated Clock (0)
Rename Auto-Derived Clock (0)
Set Clock Latency (0)
Set Clock Uncertainty (0)
Set Clock Groups (0)
Set Clock Sense (0)
Set Input Jitter (0)
Set System Jitter (0)
Set External Delay (0)
Inputs (2)
Set Input Delay (2)
Position Clock Clock Edge Delay Transition Min/Max Delay Path Add Delay Latencies Included Delay Value Objects Source File
4 [get_clocks clk] rise min None 0.000 [get_ports z] deneme.xdc
5 [get_clocks clk] rise max None 2.000 [get_ports z] deneme.xdc
Double click to create a Set Output Delay constraint
All Constraints
Position Command Scoped Cell
1 create_clock -period 8.000 -name clk -waveform {0.000 4.000} [get_ports -filter { NAME =~ "*clk*" && DIRECTION == "IN" }]
(Invalid) set_input_delay -clock [get_clocks clk] -min -add_delay 2.000 [get_ports {x[*]}]
(Invalid) set_input_delay -clock [get_clocks clk] -max -add_delay 2.000 [get_ports {x[*]}]
2 set_input_delay -clock [get_clocks clk] -min -add_delay 2.0 [get_ports rst]
3 set_input_delay -clock [get_clocks clk] -max -add_delay 2.0 [get_ports rst]
Apply Cancel
Tcl Console Messages Log Reports Design Runs Timing
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (1)
Intra-Clock Paths
Inter-Clock Paths
Design Timing Summary
Setup
Worst Negative Slack (WNS): 0,087 ns
Total Negative Slack (TNS): 0,000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 3
Hold
Worst Hold Slack (WHS): 0,106 ns
Total Hold Slack (THS): 0,000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 3
Pulse Width
Worst Pulse Width Slack (WPWS): 3,650 ns
Total Pulse Width Negative Slack (TPWS): 0,000 ns
Number of Failing Endpoints: 0
Total Number of Endpoints: 3
All user specified timing constraints are met.
Timing Summary - timing_1
Timing Summary - timing_2
Timing Summary - timing_3
Timing Summary - timing_4
Timing Summary - timing_5
Vivado Simulator
Type here to search
TUR
11:09
16.04.2020

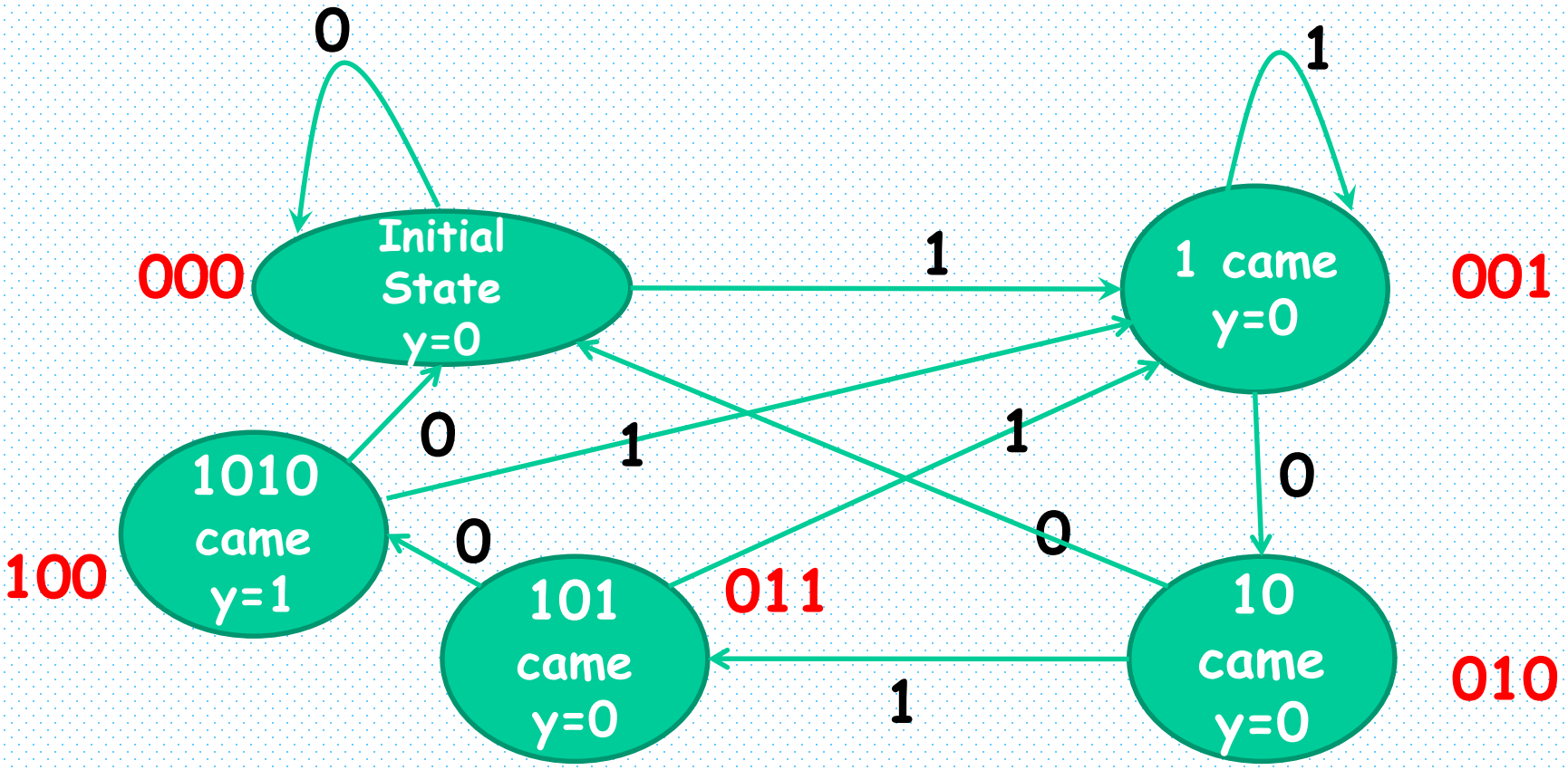

Untitled 2*
Name
Value
clk
1
rst
0
x
1
z
0
\FSM_onehot_state[0]_i_1_n_0
0
\FSM_onehot_state[1]_i_1_n_0
X
\FSM_onehot_state[2]_i_1_n_0
0
\FSM_onehot_state[3]_i_1_n_0
0
180 ns
190 ns
200 ns
210 ns
220 ns
230 ns
240 ns
250 ns
260 ns
270 ns
280 ns
290 ns
300 ns
310 ns
320 ns
Type here to search
TUR
13:03
16.04.2020


Untitled 2*
Name
Value
clk
1
rst
0
x
1
z
0
\FSM_onehot_state[0]_i_1_n_0
0
\FSM_onehot_state[1]_i_1_n_0
1
\FSM_onehot_state[2]_i_1_n_0
0
\FSM_onehot_state[3]_i_1_n_0
0
180 ns
200 ns
220 ns
240 ns
260 ns
280 ns
300 ns
320 ns
340 ns
350.000 ns
360 ns
380 ns
400 ns
420 ns
440 ns
Type here to search
TUR
13:06
16.04.2020


project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
SYNTHESIZED DESIGN - synth_2 | xc7vx485tffg1157-1
Sources
Netlist
FSM_Mealy_1010
Nets (18)
Leaf Cells (15)
Properties
Select an object to see properties
Report Power
Estimate power consumption based on the netlist design and part xc7vx485tffg1157-1.
Results name: power_1
Environment
Power Supply
Switching
Output
Reset switching activity before report power
Switching Activity for Resets: None
Simulation Settings
Simulation activity file (.saif): /sim_1/impl/timing/xsim/mealy_power.saif
Default Activity Settings
Default toggle rate: 12.5 [0 - 100]
Default Static Probability: 0.5 [0.0 - 1.0]
Enable Rate Settings
Static Probability
Toggle Rate
BRAM Port Enable: [0.0 - 1.0] [0 - 100]
BRAM Write Enable: [0.0 - 1.0] [0 - 100]
Bidi Output Port Enable: [0.0 - 1.0] [0 - 100]
Toggle Rate Settings
Static Probability
Toggle Rate
Primary Outputs: [0.0 - 1.0] [0 - 100]
Logic
Registers: [0.0 - 1.0] [0 - 100]
OK
Cancel
X1 Y6
X1 Y5
X1 Y4
X1 Y3
X1 Y2
X1 Y1
X1 Y0
Tcl Console
Messages
Log
Reports
Design Runs
Finished Parsing XDC File [E:/Berna/Dersler/YuksekLisans/
INFO: [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
Netlist sorting complete. Time (s): cpu = 00:00:00 ; elaps
INFO: [Project 1-111] Unisim Transformation Summary:
No Unisim elements were transformed.
open_run: Time (s): cpu = 00:00:07 ; elapsed = 00:00:06 . Memory (MB): peak = 3462.270 ; gain = 0.000
Type a Tcl command here
Type here to search
TUR
13:07
16.04.2020


project_1 - [E:/Berna/Dersler/YuksekLisans/Aselsan/FSM/project_1/project_1.xpr] - Vivado 2019.1
File Edit Flow Tools Reports Window Layout View Help
Quick Access
Implementation Complete
Default Layout
Flow Navigator
IP Catalog
IP INTEGRATOR
Create Block Design
Open Block Design
Generate Block Design
SIMULATION
Run Simulation
RTL ANALYSIS
Open Elaborated Design
Report Methodology
Report DRC
Report Noise
Schematic
SYNTHESIS
Run Synthesis
Open Synthesized Design
Constraints Wizard
Edit Timing Constraints
Set Up Debug
Report Timing Summary
Report Clock Networks
Report Clock Interaction
Report Methodology
Report DRC
Report Noise
Report Utilization
Report Power
Schematic
IMPLEMENTATION
SYNTHESIZED DESIGN - synth_2 | xc7vx485tffg1157-1
Sources
Netlist
FSM_Mealy_1010
Nets (18)
Leaf Cells (15)
Properties
Select an object to see properties
Project Summary
Device
FSM_Mealy_1010.vhd
FSM_Mealy_1010_tb.vhd
X0Y6
X1Y6
X0Y5
X1Y5
X0Y4
X1Y4
X0Y3
X1Y3
X0Y2
X1Y2
X0Y1
X1Y1
X0Y0
X1Y0
Tcl Console
Messages
Log
Reports
Design Runs
Power
Timing
Summary
Settings
Summary (0.243 W, Margin: N/A)
Power Supply
Utilization Details
Hierarchical (<0.001 W)
Signals (<0.001 W)
Data (<0.001 W)
Set/Reset (<0.001 W)
Logic (<0.001 W)
I/O (<0.001 W)
Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.
Total On-Chip Power: 0.243 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 25,3°C
Thermal Margin: 59,7°C (41,1 W)
Effective ϑJA: 1,4°C/W
Power supplied to off-chip devices: 0 W
Confidence level: Low
Launch Power Constraint Advisor to find and fix invalid switching activity
On-Chip Power
99%
Dynamic: <0.001 W (<1%)
18%
23%
59%
Signals: <0.001 W (18%)
Logic: <0.001 W (23%)
I/O: <0.001 W (59%)
Device Static: 0.242 W (99%)
power_1
Type here to search
TUR
13:08
16.04.2020

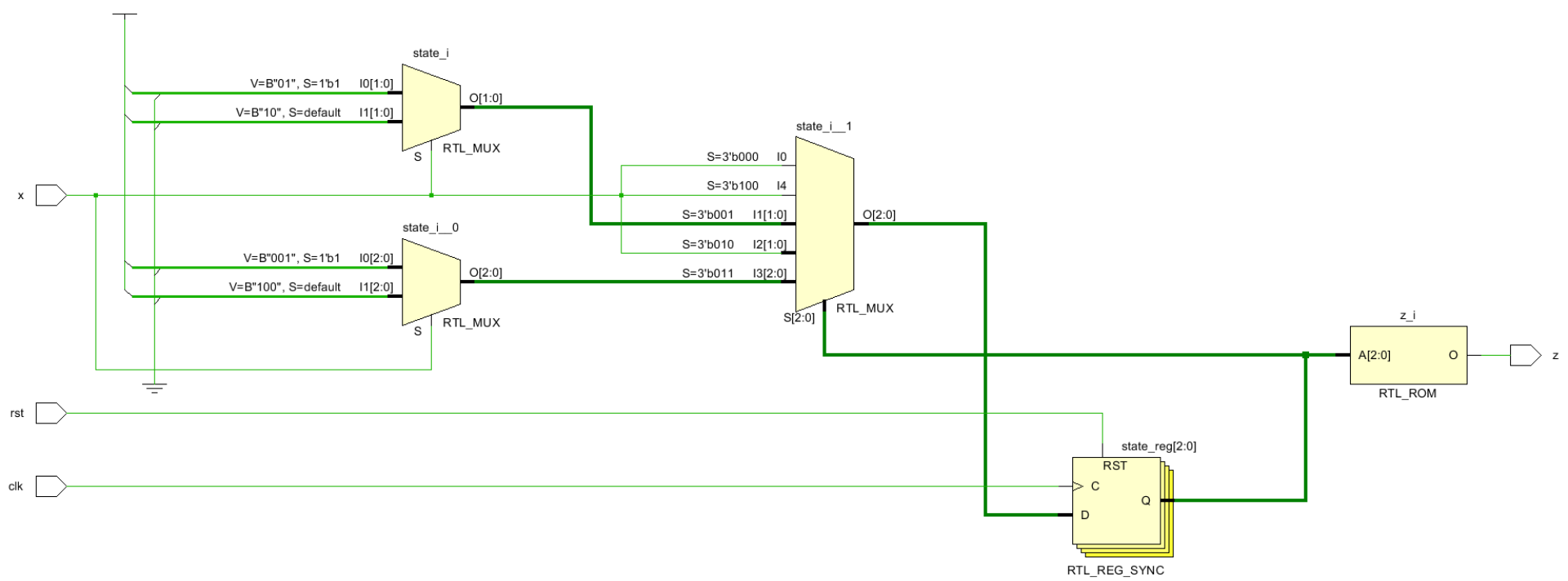
Moore Machine



VHDL Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity FSM_Mealy_1010 is
    Port ( clk : in STD_LOGIC;
           rst : in STD_LOGIC;
           x : in STD_LOGIC;
           z : out STD_LOGIC);
end FSM_Mealy_1010;
architecture Behavioral of FSM_Mealy_1010 is
  type state_type is
   (Initial,One_Came,One_Zero_Came,One_Ze
   ro_One_Came,One_Zero_One_Zero_Came);
  signal state : state_type;
begin
  state_transition: process(clk)    begin
    if(clk'event and clk='1') then
      if(rst='1') then
        state <= Initial;
      else
        case state is
          when Initial =>
              if(x='1') then
                  state <= One_Came;
              else
                  state <= Initial;   end if;
          when One_Came =>
              if(x='1') then      state <= One_Came;
              else                state <= One_Zero_Came;
              end if;
          when One_Zero_Came =>
              if(x='1') then      state <= One_Zero_One_Came;
              else                state <= Initial;
              end if;
          when One_Zero_One_Came =>
              if(x='1') then      state <= One_Came;
              else          state <= One_Zero_One_Zero_Came;
              end if;
          when One_Zero_One_Zero_Came =>
              if(x='1') then      state <= One_Came;
              else                state <= Initial;
              end if;
        end case;
    end if;   end if;
end process;
output: process(state)
  begin
    case state is
        when One_Zero_One_Zero_Came =>
          z  <=  '1';
        when others =>
          z  <=  '0';
        end case;
  end process;
end Behavioral;
```

Moore Machine RTL Schematic



Timing Diagram