

# Flip-flop

4 modula:

- sr\_ff
- d\_ff
- jk\_ff
- t\_ff

4 testbench fajla:

- sr\_ff\_tb
- d\_ff\_tb
- jk\_ff\_tb
- t\_ff\_tb

# Flip-flop

- Kreiramo projekat: flip\_flopovi\_vlsi
- Dodamo prvi modul: sr\_ff.v

# sr\_ff

```
module sr_ff (  
    input wire clk,  
    input wire rst,  
    input wire S,  
    input wire R,  
    output reg Q,  
    output reg Qn  
);  
  
always @(posedge clk or posedge rst) begin  
    if (rst) begin  
        Q <= 0;  
        Qn <= 1;  
    end else begin  
        case ({S, R})  
            2'b00: begin  
                Q <= Q; // zadržava stanje  
                Qn <= Qn;  
            end  
            2'b01: begin  
                Q <= 0; // reset  
                Qn <= 1;  
            end  
            2'b10: begin  
                Q <= 1; // set  
                Qn <= 0;  
            end  
            2'b11: begin  
                Q <= 0; // nedozvoljeno stanje (ovde definisemo)  
                Qn <= 0;  
            end  
        endcase  
    end  
end  
  
endmodule
```

stanje 11 je nedefinisano u teoriji

→ u HDL-u mora ga se definise: (ovde: 0,0)

# sr\_ff\_tb

lip-flop ne reaguje odmah kada se promeni S i R, nego tek na:

pozitivnu ivicu clock-a.

To je glavna razlika u odnosu na latch.

```
module sr_ff_tb;
```

```
reg clk;  
reg rst;  
reg S;  
reg R;  
wire Q;  
wire Qn;
```

```
sr_ff uut (  
    .clk(clk),  
    .rst(rst),  
    .S(S),  
    .R(R),  
    .Q(Q),  
    .Qn(Qn)  
);
```

```
always #5 clk = ~clk;
```

```
initial begin  
    clk = 0;  
    rst = 1;  
    S = 0;  
    R = 0;
```

```
    #10;  
    rst = 0;
```

```
    // 00 - zadržava stanje  
    #10;  
    S = 0; R = 0;
```

```
    // 10 - set  
    #10;  
    S = 1; R = 0;
```

```
    // 00 - zadržava stanje  
    #10;  
    S = 0; R = 0;
```

```
    // 01 - reset  
    #10;  
    S = 0; R = 1;
```

```
    // 00 - zadržava stanje  
    #10;  
    S = 0; R = 0;
```

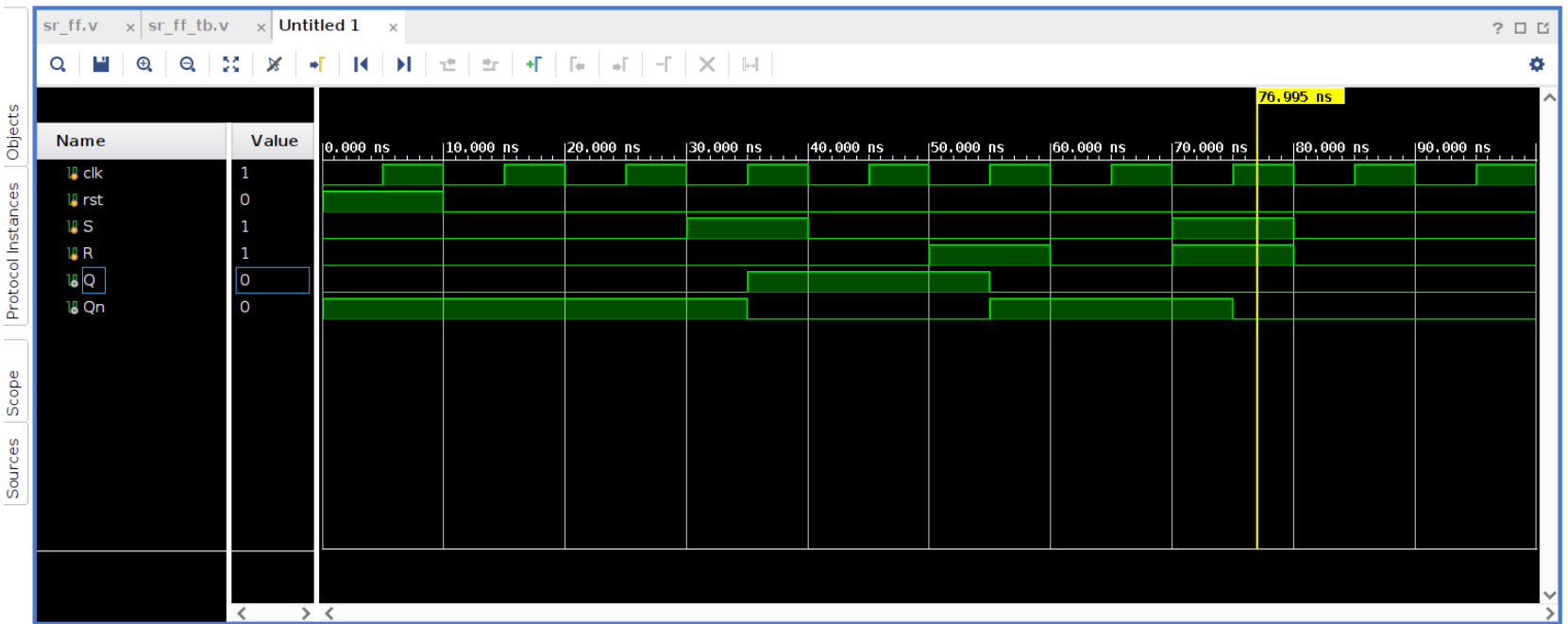
```
    // 11 - nedozvoljeno stanje  
    #10;  
    S = 1; R = 1;
```

```
    // povratak  
    #10;  
    S = 0; R = 0;
```

```
    #20;  
    $finish;  
end
```

```
endmodule
```

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION**
  - Run Simulation
- RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager



Tcl Console Messages Log

```

# run 1000ns
$finish called at time : 100 ns : File "/home/apeulic/fakultet/predmeti/Osnovne/VLSI/vezbe/V_5/flip_flopovi_vlsi/flip_flopovi_vlsi.srcs/sim_1/new/sr_ff_tb.v"
INFO: [USF-XSim-96] XSim completed. Design snapshot 'sr_ff_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:09 ; elapsed = 00:00:09 . Memory (MB): peak = 10291.379 ; gain = 76.457 ; free physical = 7239 ; free virtual = 1413
    
```

Type a Tcl command here

# D flip-flop

- Kreiramo modul: d\_ff.v

šta god je na D

biće zapamćeno na sledećoj ivici clock-a

```
module d_ff (  
    input wire clk,  
    input wire rst,  
    input wire D,  
    output reg Q,  
    output reg Qn  
);
```

```
always @(posedge clk or posedge rst) begin  
    if (rst) begin  
        Q <= 0;  
        Qn <= 1;  
    end else begin  
        Q <= D;  
        Qn <= ~D;  
    end  
end  
  
endmodule
```

# TB za D-FF

- Kreiramo simulacioni modul: `d_ff_tb.v`

```
module d_ff_tb;

reg clk;
reg rst;
reg D;
wire Q;
wire Qn;

d_ff uut (
    .clk(clk),
    .rst(rst),
    .D(D),
    .Q(Q),
    .Qn(Qn)
);

always #5 clk = ~clk;

initial begin
    clk = 0;
    rst = 1;
    D = 0;

    #10;
    rst = 0;

    // D = 0
    #10;
    D = 0;

    // D = 1
    #10;
    D = 1;

    // promena izmedju clock ivica
    #3;
    D = 0;

    #7;
    D = 1;

    // jos promena
    #10;
    D = 0;

    #10;
    D = 1;

    #20;
    $finish;
end

endmodule
```

- ▼ PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- ▼ IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- ▼ SIMULATION
  - Run Simulation
- ▼ RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- ▼ SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- ▼ IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- ▼ PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

**Sources** ? \_ □ ×

Updating

- > Constraints
- ▼ Simulation Sources (2)
  - ▼ sim\_1 (2)
    - > d\_ff\_tb (d\_ff\_tb.v) (1)
    - > sr\_ff\_tb (sr\_ff\_tb.v) (1)
- > Utility Sources

Hierarchy Libraries Compile Order

**Source File Properties** ? \_ □ ×

d\_ff\_tb.v

Enabled

Location: /home/apeulic/fakultet/predmeti/Osnovne/VLSI/vezbe/V\_5/flip\_fi

Type: Verilog

General Properties

Project Summary x sr\_ff.v x sr\_ff\_tb.v x d\_ff.v x **d\_ff\_tb.v** x ? □ ×

/home/apeulic/fakultet/predmeti/Osnovne/VLSI/vezbe/V\_5/flip\_flopovi\_vlsi/flip\_flopovi\_vlsi.sracs/sim\_1/new/d\_ff\_tb.v x

```

32     #10;
33     D = 1;
34
35     // promena izmedju clock ivica
36     #3;
37     D = 0;
38
39     #7;
40     D = 1;
41
42     // jos promena
43     #10;
44     D = 0;
45
46     #10;
47     D = 1;
48
49     #20;
50     $finish;
51 end
52
53 endmodule
    
```

Tcl Console Messages Log Reports **Design Runs** x ? \_ □ ×

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM
▼ synth_1	constrs_1	Not started															
▶ impl_1	constrs_1	Not started															

Set As Top

Flow Navigator

PROJECT MANAGER

- Settings
- Add Sources
- Language Templates
- IP Catalog

IP INTEGRATOR

- Create Block Design
- Open Block Design
- Generate Block Design

SIMULATION

- Run Simulation

RTL ANALYSIS

- Run Linter
- Open Elaborated Design

SYNTHESIS

- Run Synthesis
- Open Synthesized Design

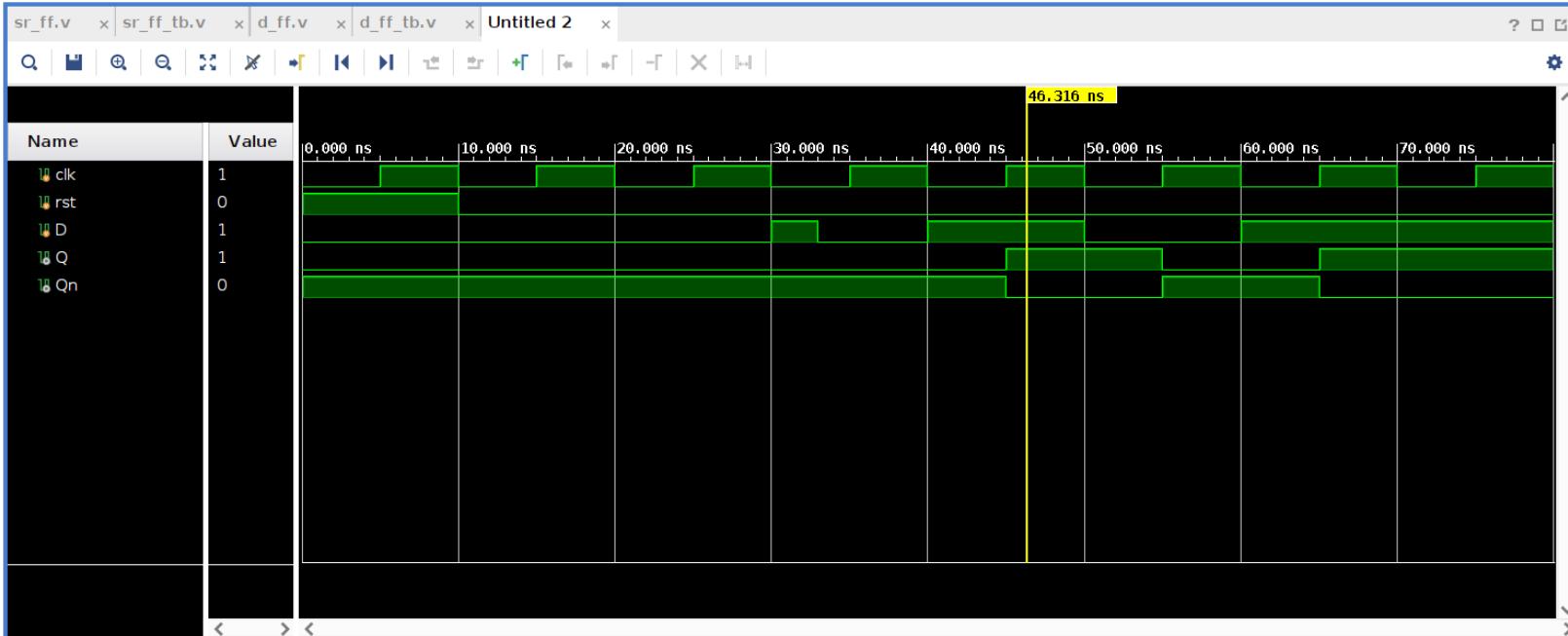
IMPLEMENTATION

- Run Implementation
- Open Implemented Design

PROGRAM AND DEBUG

- Generate Bitstream
- Open Hardware Manager

SIMULATION - Behavioral Simulation - Functional - sim\_1 - d\_ff\_tb



Tcl Console

```

# run 1000ns
$finish called at time : 80 ns : File "/home/apeulic/fakultet/predmeti/Osnovne/VLSI/vezbe/V_5/flip_flopovi_vlsi/flip_flopovi_vlsi.srcs/sim_1/new/d_ff_tb.v" L
INFO: [USF-XSim-96] XSim completed. Design snapshot 'd_ff_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:08 ; elapsed = 00:00:08 . Memory (MB): peak = 10484.887 ; gain = 64.789 ; free physical = 6990 ; free virtual = 1392
    
```

# JK flip-flop

- Kreiramo modul:jk\_ff.v

```
module jk_ff (  
    input wire clk,  
    input wire rst,  
    input wire J,  
    input wire K,  
    output reg Q,  
    output reg Qn  
);  
  
always @(posedge clk or posedge rst) begin  
    if (rst) begin  
        Q <= 0;  
        Qn <= 1;  
    end else begin  
        case ({J, K})  
            2'b00: begin  
                Q <= Q; // hold  
                Qn <= Qn;  
            end  
            2'b01: begin  
                Q <= 0; // reset  
                Qn <= 1;  
            end  
            2'b10: begin  
                Q <= 1; // set  
                Qn <= 0;  
            end  
            2'b11: begin  
                Q <= ~Q; // TOGGLE  
                Qn <= ~Qn;  
            end  
        endcase  
    end  
end  
  
endmodule
```

# Testbench za JK flip-flop

- Kreiramo modul tb, Add or Create Simulation Sources: jk\_ff\_tb.v

```
module jk_ff_tb;

reg clk;
reg rst;
reg J;
reg K;
wire Q;
wire Qn;

jk_ff uut (
    .clk(clk),
    .rst(rst),
    .J(J),
    .K(K),
    .Q(Q),
    .Qn(Qn)
);

always #5 clk = ~clk;

initial begin
    clk = 0;
    rst = 1;
    J = 0;
    K = 0;

    #10;
    rst = 0;

    // 00 - hold
    #10;
    J = 0; K = 0;

    // 10 - set
    #10;
    J = 1; K = 0;

    // 00 - hold
    #10;
    J = 0; K = 0;

    // 01 - reset
    #10;
    J = 0; K = 1;

    // 00 - hold
    #10;
    J = 0; K = 0;

    // 11 - toggle
    #10;
    J = 1; K = 1;

    // ostavi toggle jos malo da se vidi vise promena
    #20;
    J = 1; K = 1;

    // povratak na hold
    #10;
    J = 0; K = 0;

    #20;
    $finish;
end

endmodule
```

File Edit Flow Tools Reports Window Layout View Help Q: Quick Access Ready

Flow Navigator PROJECT MANAGER - flip\_flopovi\_vlsi

- PROJECT MANAGER
  - Settings
  - Add Sources
  - Language Templates
  - IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

Sources

Updating

- Simulation Sources (3)
  - sim\_1 (3)
    - jk\_ff\_tb (jk\_ff\_tb.v) (1)
    - d\_ff\_tb (d\_ff\_tb.v) (1)
    - sr\_ff\_tb (sr\_ff\_tb.v) (1)
- Utility Sources

Hierarchy Libraries Compile Order

Source File Properties

jk\_ff\_tb.v

Enabled

Location: /home/apeulic/fakultet/predmeti/Osnovne/VLSI/vezbe/V\_5/flip\_flopovi\_vlsi

Type: Verilog

General Properties

```

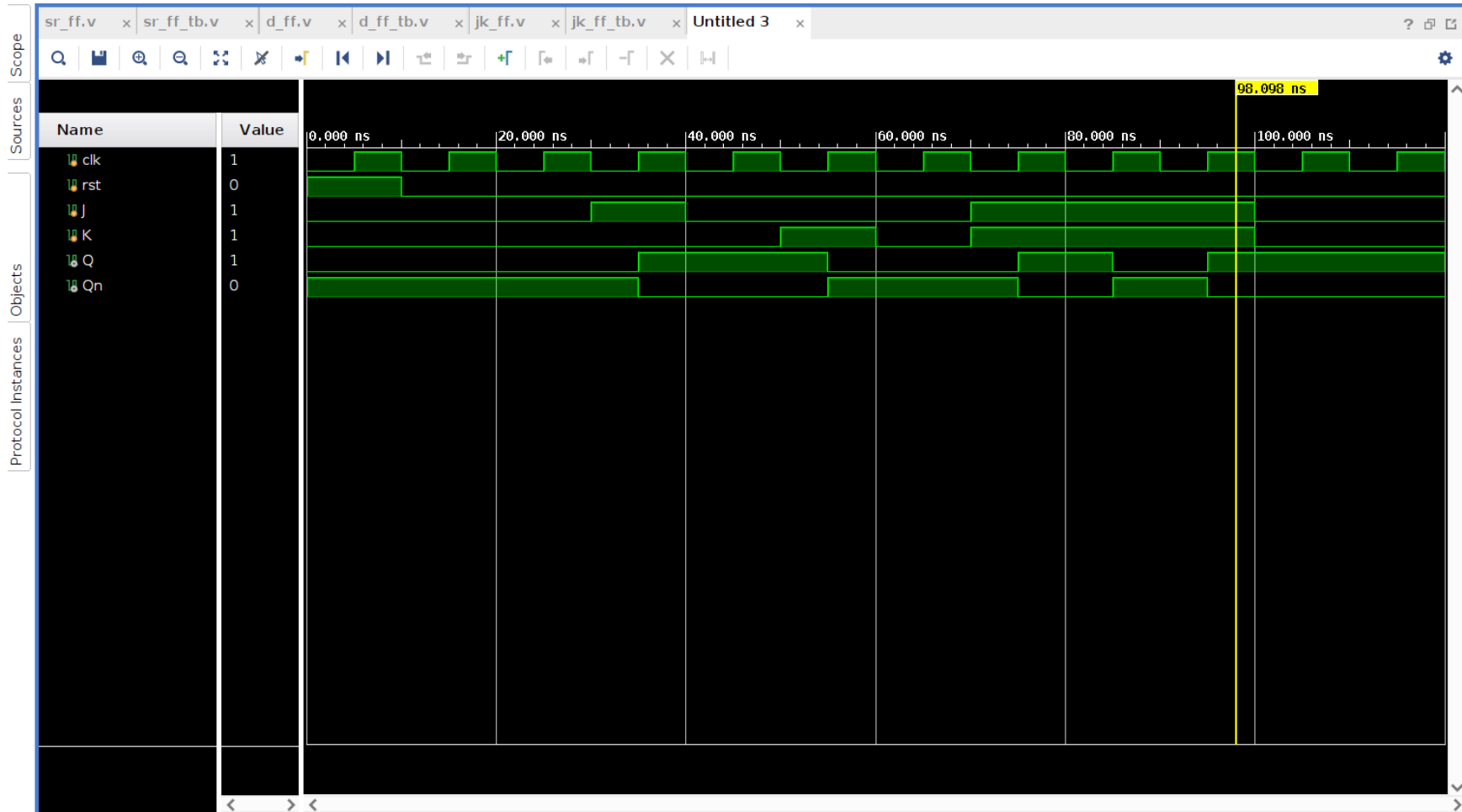
45
46 // 00 - hold
47 #10;
48 J = 0; K = 0;
49
50 // 11 - toggle
51 #10;
52 J = 1; K = 1;
53
54 // ostavi toggle jos malo da se vidi vise promena
55 #20;
56 J = 1; K = 1;
57
58 // povratak na hold
59 #10;
60 J = 0; K = 0;
61
62 #20;
63 $finish;
64 end
65
66 endmodule
    
```

Tcl Console Messages Log Reports Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM
synth_1	constrs_1	Not started															
impl_1	constrs_1	Not started															

Set As Top

- OBJECT MANAGER**
- Settings
- Add Sources
- Language Templates
- IP Catalog
- INTEGRATOR**
- Create Block Design
- Open Block Design
- Generate Block Design
- SIMULATION**
- Run Simulation
- ANALYSIS**
- Run Linter
- Open Elaborated Design
- SYNTHESIS**
- Run Synthesis
- Open Synthesized Design
- IMPLEMENTATION**
- Run Implementation
- Open Implemented Design
- PROGRAM AND DEBUG**
- Generate Bitstream
- Open Hardware Manager



# T je specijalan slučaj JK

- $J = T$
- $K = T$
- Kreiramo  $t_{ff.v}$

```
module t_ff (  
    input wire clk,  
    input wire rst,  
    input wire T,  
    output reg Q,  
    output reg Qn  
);  
  
always @(posedge clk or posedge rst) begin  
    if (rst) begin  
        Q <= 0;  
        Qn <= 1;  
    end else begin  
        if (T == 0) begin  
            Q <= Q; // hold  
            Qn <= Qn;  
        end else begin  
            Q <= ~Q; // toggle  
            Qn <= ~Qn;  
        end  
    end  
end  
  
endmodule
```

# Testbench za T flip-flop

- Add or Create Simulation Sources: t\_ff\_tb.v

```
module t_ff_tb;

reg clk;
reg rst;
reg T;
wire Q;
wire Qn;

t_ff uut (
    .clk(clk),
    .rst(rst),
    .T(T),
    .Q(Q),
    .Qn(Qn)
);

always #5 clk = ~clk;

initial begin
    clk = 0;
    rst = 1;
    T = 0;

    #10;
    rst = 0;

    // T = 0 - hold
    #10;
    T = 0;

    // T = 1 - toggle
    #10;
    T = 1;

    // ostavi toggle duze (da se vidi lepo)
    #40;
    T = 1;

    // vrati na hold
    #10;
    T = 0;

    // opet toggle
    #10;
    T = 1;

    #40;

    $finish;
end

endmodule
```



## Flow Navigator

## PROJECT MANAGER - flip\_flopovi\_vlsi

## PROJECT MANAGER

- Settings
- Add Sources
- Language Templates
- IP Catalog
- IP INTEGRATOR
  - Create Block Design
  - Open Block Design
  - Generate Block Design
- SIMULATION
  - Run Simulation
- RTL ANALYSIS
  - Run Linter
  - Open Elaborated Design
- SYNTHESIS
  - Run Synthesis
  - Open Synthesized Design
- IMPLEMENTATION
  - Run Implementation
  - Open Implemented Design
- PROGRAM AND DEBUG
  - Generate Bitstream
  - Open Hardware Manager

Sources

Updating

- Constraints
- Simulation Sources (4)
  - sim\_1 (4)
    - t\_ff\_tb (t\_ff\_tb.v) (1)**
    - d\_ff\_tb (d\_ff\_tb.v) (1)
    - jk\_ff\_tb (jk\_ff\_tb.v) (1)
    - sr\_ff\_tb (sr\_ff\_tb.v) (1)

Hierarchy Libraries Compile Order

Source File Properties

t\_ff\_tb.v

Enabled

Location: /home/apeulic/fakultet/predmeti/Osnovne/VLSI/vezbe/V\_5/flip\_f

Type: Verilog

General Properties

```

b.v x d_ff.v x d_ff_tb.v x jk_ff.v x jk_ff_tb.v x t_ff.v x t_ff_tb.v x
/home/apeulic/fakultet/predmeti/Osnovne/VLSI/vezbe/V_5/flip_flopovi_vlsi/flip_flopovi_vlsi.sracs/sim_1/new/t_ff_tb.v x
31 // T = 1 → toggle
32 #10;
33 T = 1;
34
35 // ostavi toggle duze (da se vidi lepo)
36 #40;
37 T = 1;
38
39 // vrati na hold
40 #10;
41 T = 0;
42
43 // opet toggle
44 #10;
45 T = 1;
46
47 #40;
48
49 $finish;
50 end
51
52 endmodule

```

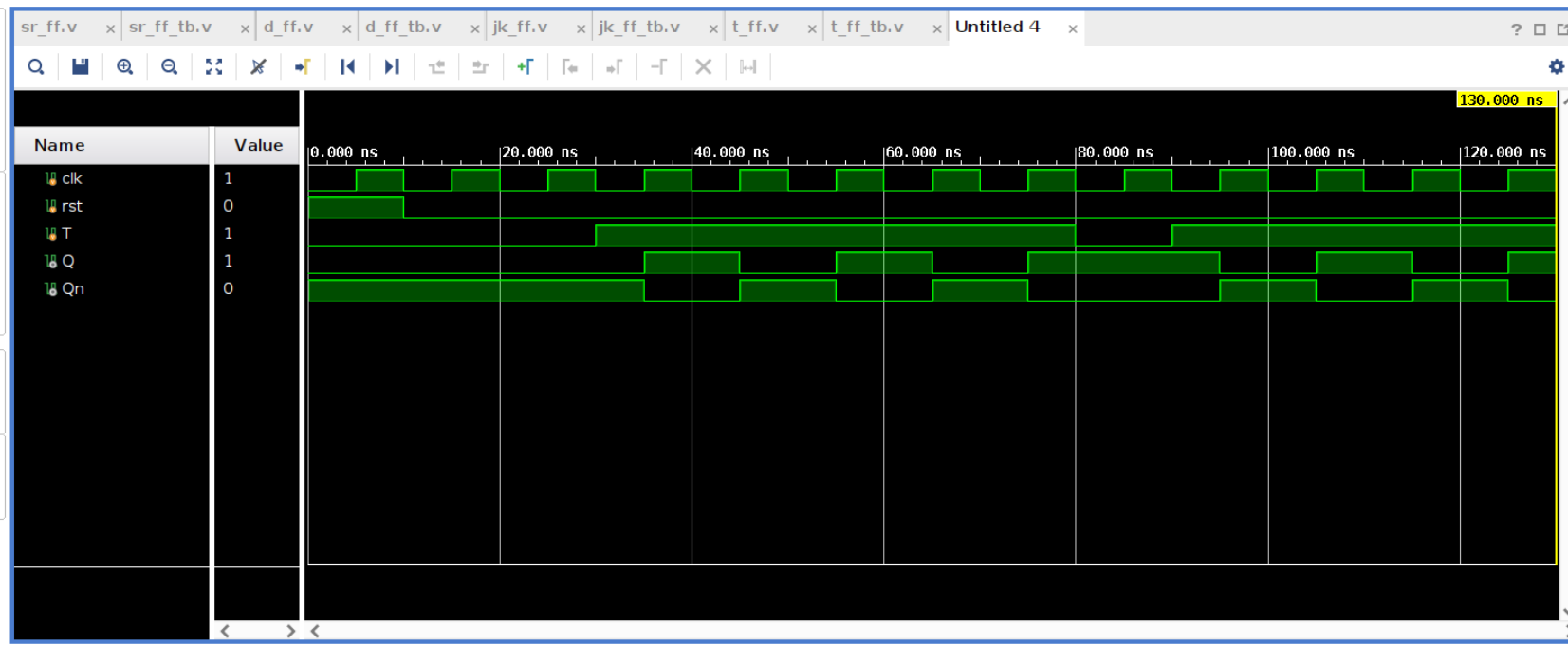
Tcl Console Messages Log Reports Design Runs

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	QoR Suggestions	LUT	FF	BRAM	URAM
synth_1	constrs_1	Not started															
impl_1	constrs_1	Not started															

Set As Top

- Flow Navigator
- PROJECT MANAGER
    - Settings
    - Add Sources
    - Language Templates
    - IP Catalog
  - IP INTEGRATOR
    - Create Block Design
    - Open Block Design
    - Generate Block Design
  - SIMULATION**
    - Run Simulation
  - RTL ANALYSIS
    - Run Linter
    - Open Elaborated Design
  - SYNTHESIS
    - Run Synthesis
    - Open Synthesized Design
  - IMPLEMENTATION
    - Run Implementation
    - Open Implemented Design
  - PROGRAM AND DEBUG
    - Generate Bitstream
    - Open Hardware Manager

SIMULATION - Behavioral Simulation - Functional - sim\_1 - t\_ff\_tb



Tcl Console

```

# run 1000ns
$finish called at time : 130 ns : File "/home/apeulic/fakultet/predmeti/Osnovne/VLSI/vezbe/V_5/flip_flopovi_vlsi/flip_flopovi_vlsi.srcs/sim_1/new/t_ff_tb.v"
INFO: [USF-XSim-96] XSim completed. Design snapshot 't_ff_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:07 ; elapsed = 00:00:08 . Memory (MB): peak = 10860.035 ; gain = 55.785 ; free physical = 6809 ; free virtual = 1378
    
```

Type a Tcl command here