DIGITAL DESIGN

LAB12 REGISTER, COUNTER

2021 SUMMER TERM

LAB12

- Register
- Counter
- Practice

REGISTER

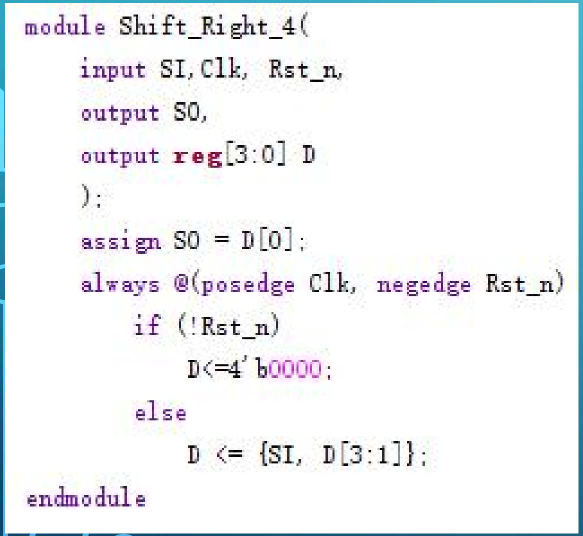
- In digital electronics, especially computing, **hardware registers** are circuits typically composed of flip flops, often with many characteristics similar to memory, such as: The ability to read or write multiple bits at a time, and using an address to select a particular register in a manner similar to a memory address.
- Hardware registers are used in the interface between software and peripherals. Software writes them to send information to the device, and reads them to get information from the device. Some hardware devices also include registers that are not visible to software, for their internal use.

REGISTER

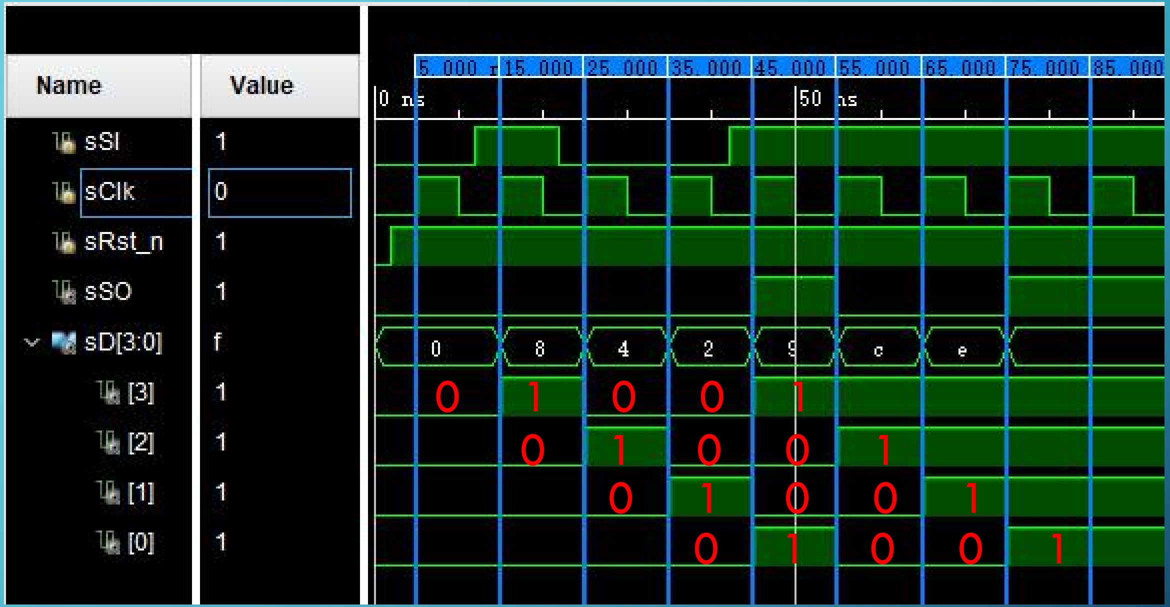
- In its broadest definition, a register consists of a group of flip-flops together with gates that affect their operation. The flip-flops hold the binary information, and the gates determine how the information is transferred into the register.
- A register capable of shifting the binary information held in each cell to its neighboring cell, in a selected direction, is called a *shift register*.

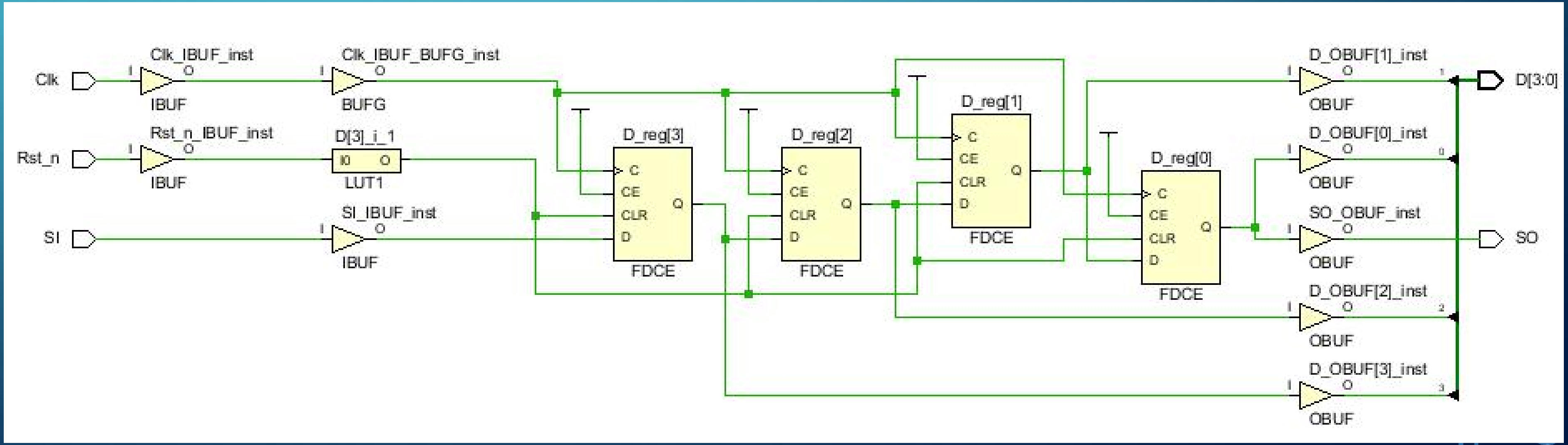
SHIFT REGISTER

- Shift right



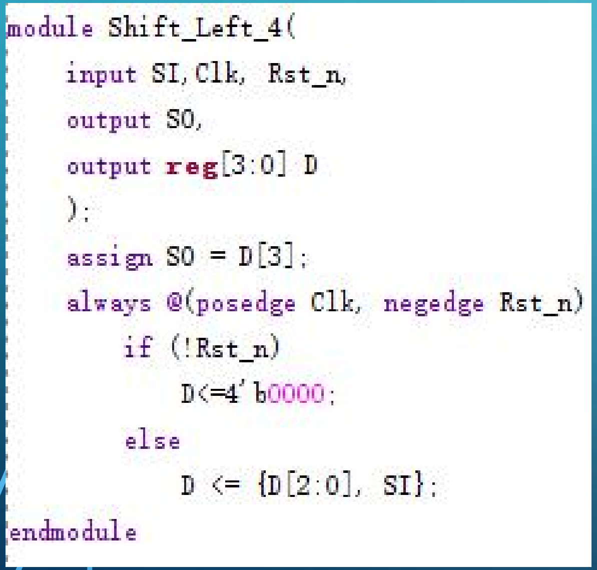
```
module Shift_Right_4(
    input SI,Clk, Rst_n,
    output SO,
    output reg[3:0] D
    );
    assign SO = D[0];
    always @(posedge Clk, negedge Rst_n)
        if (!Rst_n)
            D<=4'b0000;
        else
            D <= {SI, D[3:1]};
endmodule
```



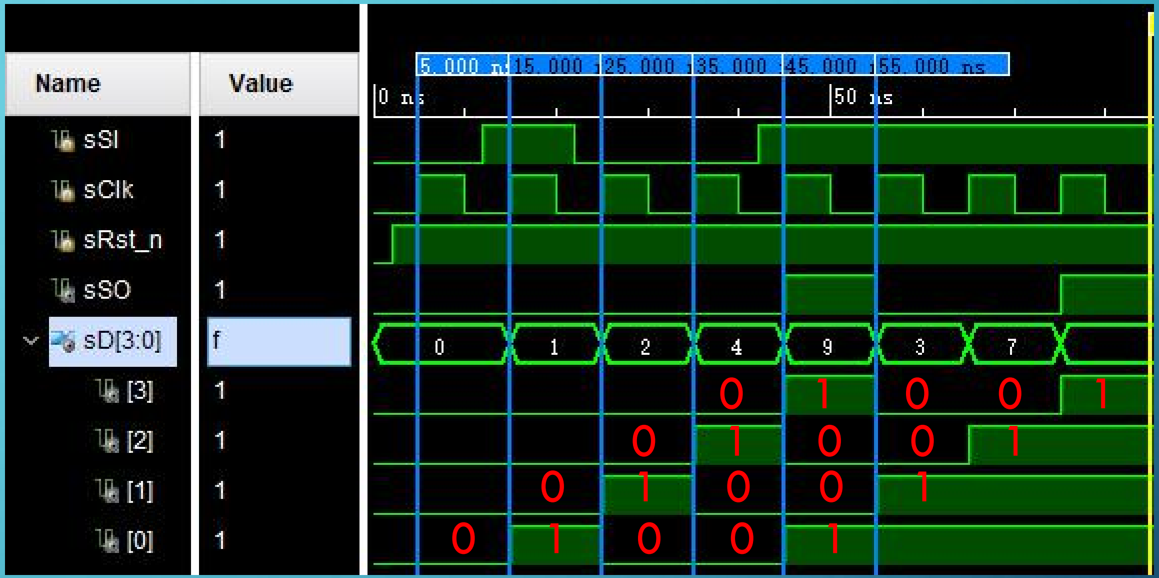


SHIFT REGISTER

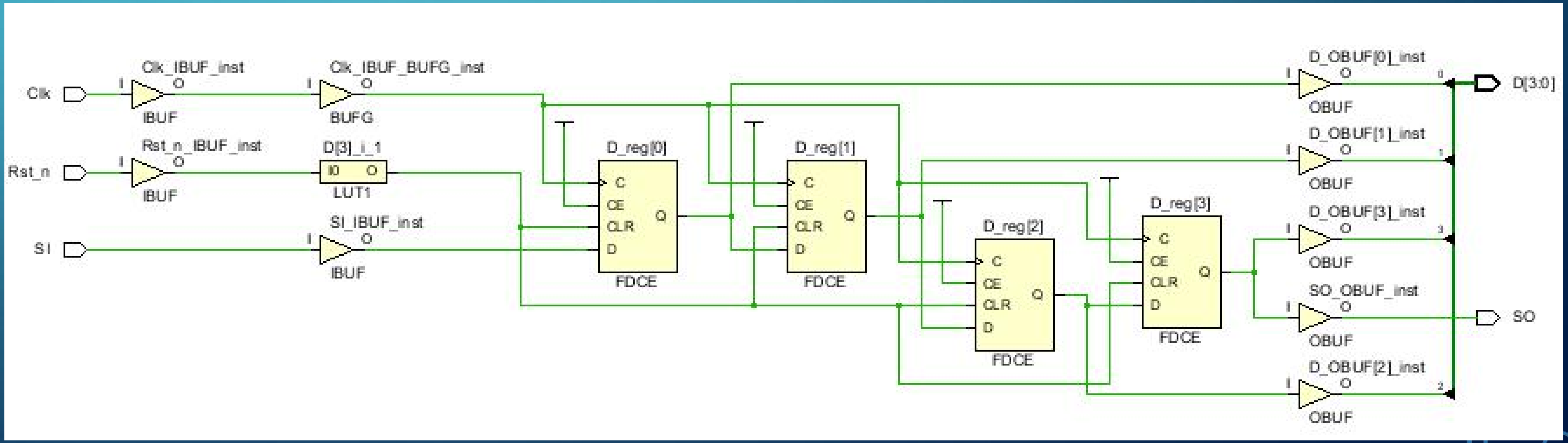
- Shift left



```
module Shift_Left_4(
    input SI,Clk, Rst_n,
    output SO,
    output reg[3:0] D
    );
    assign SO = D[3];
    always @(posedge Clk, negedge Rst_n)
        if (!Rst_n)
            D<=4'b0000;
        else
            D <= {D[2:0], SI};
endmodule
```

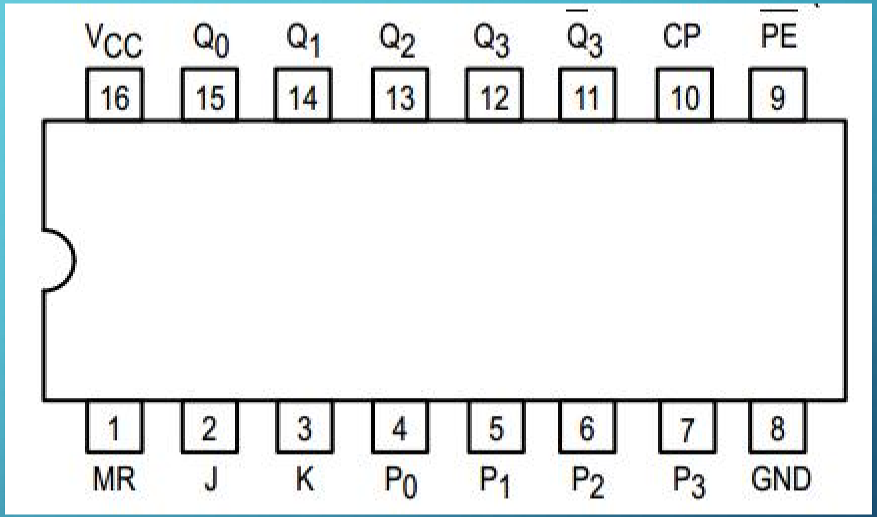


Name	Value
sSI	1
sClk	1
sRst_n	1
sSO	1
sD[3:0]	f
[3]	1
[2]	1
[1]	1
[0]	1



SHIFT REGISTER 74195

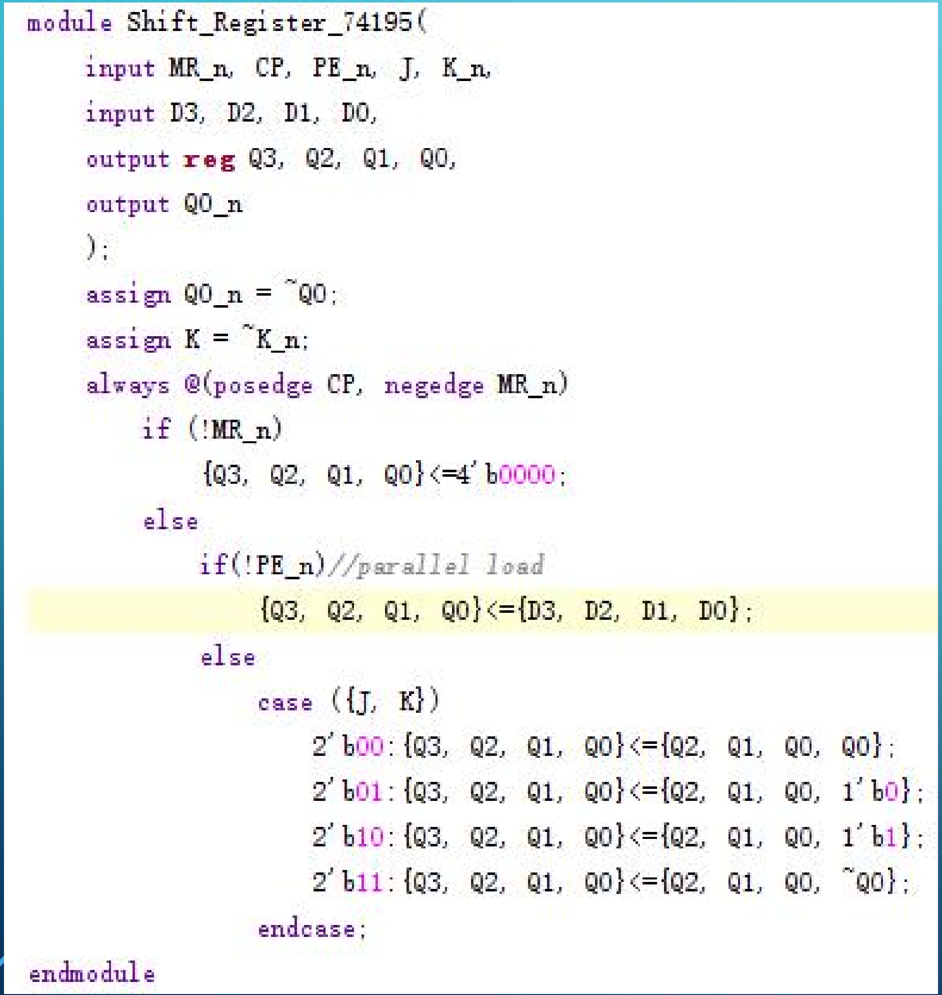
- **Pin names**
- $\overline{PE}$ Parallel Enable Input
- $P_0 \sim P_3$ Parallel Data Inputs
- J First Stage J Input
- $\overline{K}$ First Stage K Input
- CP Clock Input
- $\overline{MR}$ Master Reset Input
- $Q_0 \sim Q_3$ Parallel Outputs, Q0 is MSB
- $\overline{Q_3}$ Complementary Last Stage Output



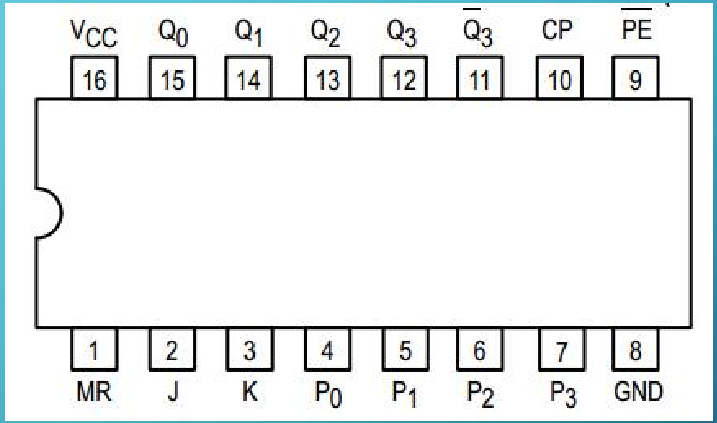
UNIVERSAL 4-BIT SHIFT REGISTER

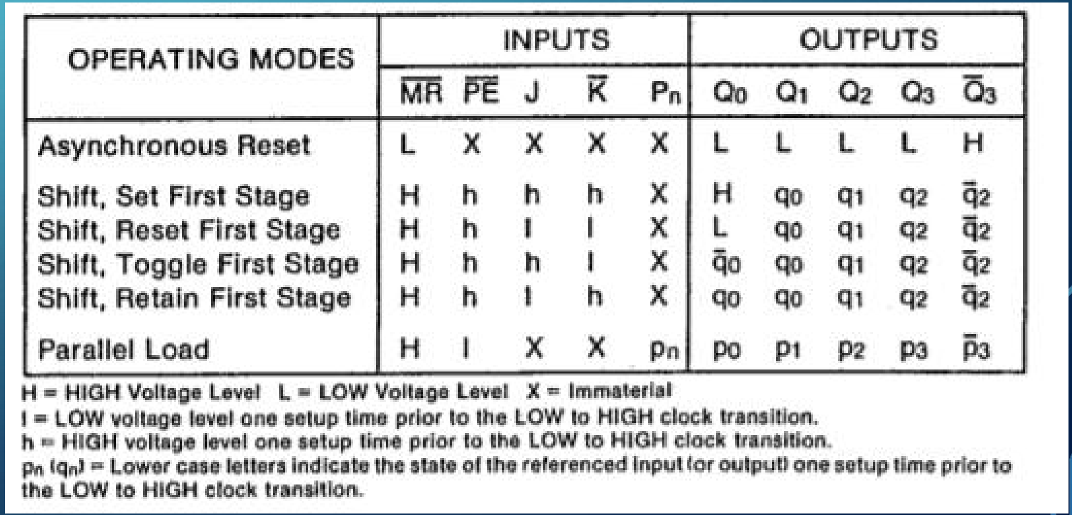
The SN54/74LS195A is a high speed 4-Bit Shift Register offering typical shift frequencies of 39 MHz. It is useful for a wide variety of register and counting applications.

SHIFT REGISTER 74195



```
module Shift_Register_74195(
    input MR_n, CP, PE_n, J, K_n,
    input D3, D2, D1, D0,
    output reg Q3, Q2, Q1, Q0,
    output Q0_n
    );
    assign Q0_n = ~Q0;
    assign K = ~K_n;
    always @(posedge CP, negedge MR_n)
        if (!MR_n)
            {Q3, Q2, Q1, Q0}<=4'b0000;
        else
            if(!PE_n)//parallel load
                {Q3, Q2, Q1, Q0}<={D3, D2, D1, D0};
            else
                case ({J, K})
                    2'b00:{Q3, Q2, Q1, Q0}<={Q2, Q1, Q0, Q0};
                    2'b01:{Q3, Q2, Q1, Q0}<={Q2, Q1, Q0, 1'b0};
                    2'b10:{Q3, Q2, Q1, Q0}<={Q2, Q1, Q0, 1'b1};
                    2'b11:{Q3, Q2, Q1, Q0}<={Q2, Q1, Q0, ~Q0};
                endcase;
endmodule
```

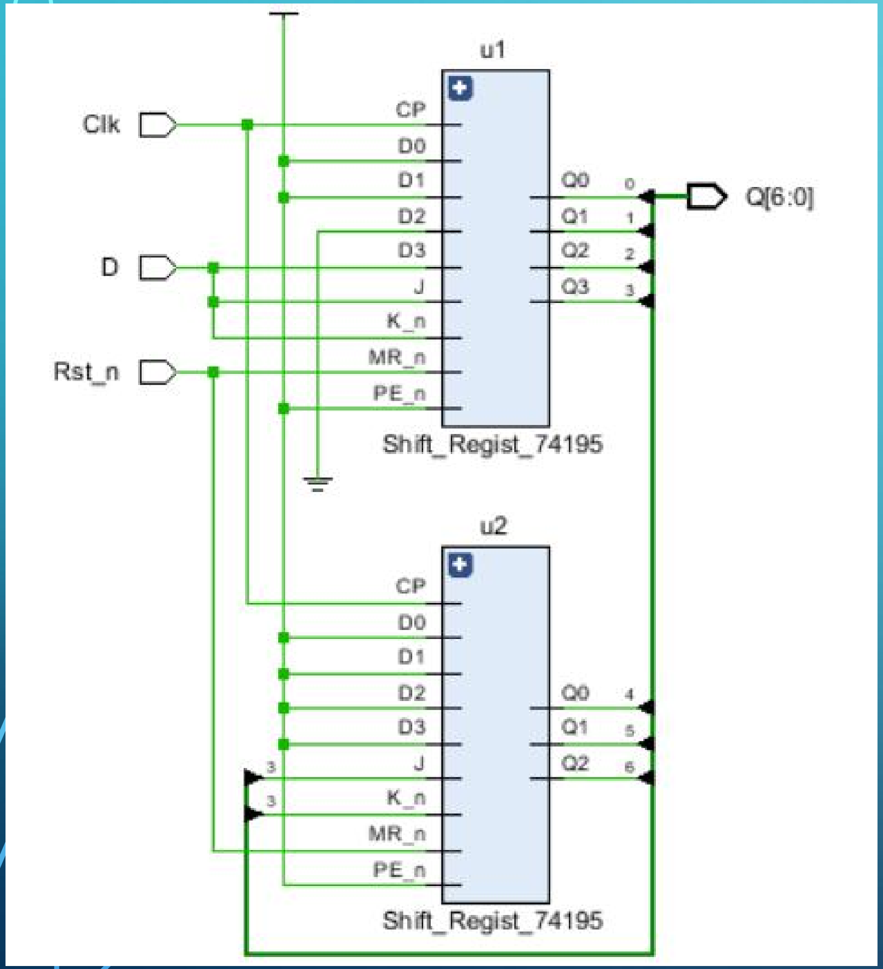


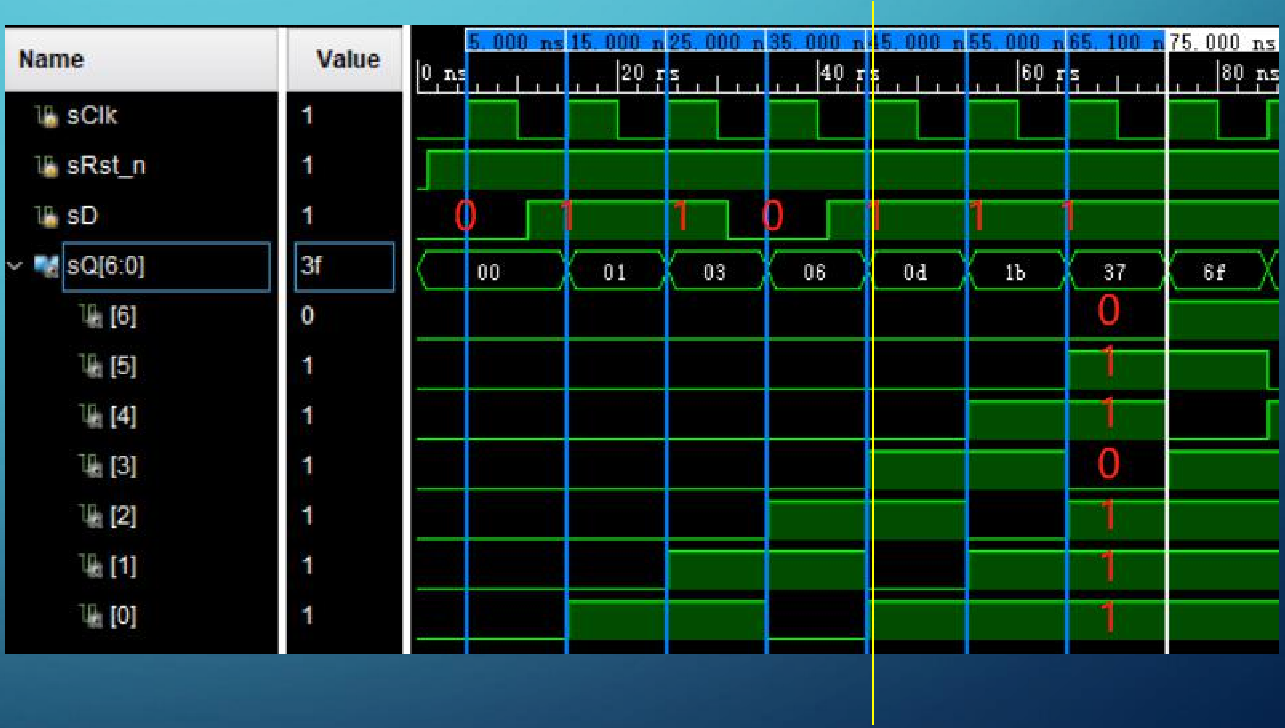


OPERATING MODES	INPUTS					OUTPUTS				
	$\overline{MR}$	$\overline{PE}$	J	$\overline{K}$	P_n	Q_0	Q_1	Q_2	Q_3	$\overline{Q_3}$
Asynchronous Reset	L	X	X	X	X	L	L	L	L	H
Shift, Set First Stage	H	h	h	h	X	H	q_0	q_1	q_2	$\overline{q_2}$
Shift, Reset First Stage	H	h	l	l	X	L	q_0	q_1	q_2	$\overline{q_2}$
Shift, Toggle First Stage	H	h	h	l	X	$\overline{q_0}$	q_0	q_1	q_2	$\overline{q_2}$
Shift, Retain First Stage	H	h	l	h	X	q_0	q_0	q_1	q_2	$\overline{q_2}$
Parallel Load	H	l	X	X	p_n	p_0	p_1	p_2	p_3	$\overline{p_3}$

H = HIGH Voltage Level L = LOW Voltage Level X = Immaterial
l = LOW voltage level one setup time prior to the LOW to HIGH clock transition.
h = HIGH voltage level one setup time prior to the LOW to HIGH clock transition.
p_n (q_n) = Lower case letters indicate the state of the referenced input (or output) one setup time prior to the LOW to HIGH clock transition.

SERIAL-PARALLEL CONVERTER WITH TWO 74195 CHIPS

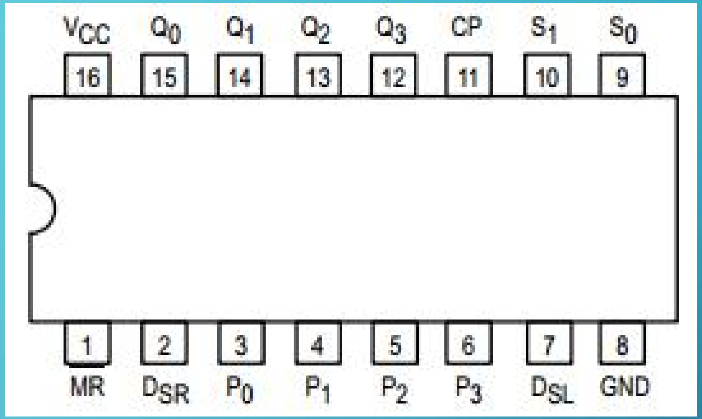




Name	Value
sClk	1
sRst_n	1
sD	1
sQ[6:0]	3f
[6]	0
[5]	1
[4]	1
[3]	1
[2]	1
[1]	1
[0]	1

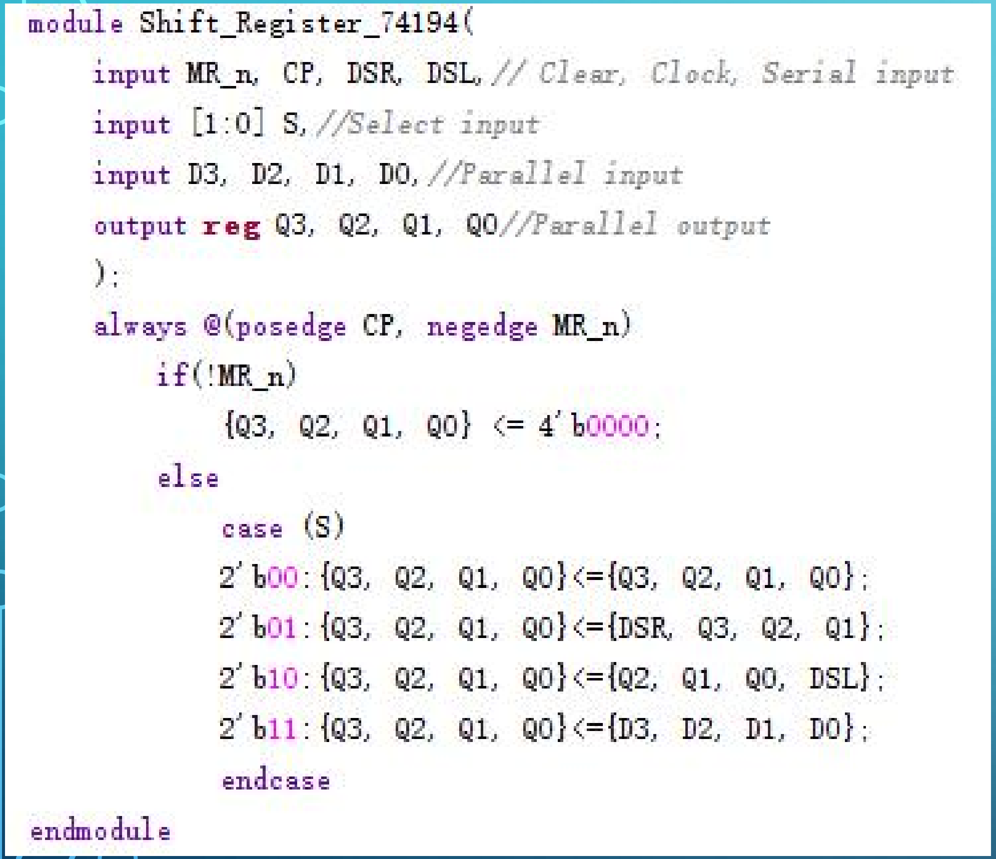
SHIFT REGISTER 74194

• S_0, S_1	Mode Control inputs
• $P_0 \sim P_3$	Parallel Data Inputs
• D_{SR}	Serial(Shift Right) Data Input
• D_{SL}	Serial(Shift Left) Data Input
• CP	Clock Input
• $\overline{MR}$	Master Reset Input
• $Q_0 \sim Q_3$	Parallel Outputs, Q0 is MSB

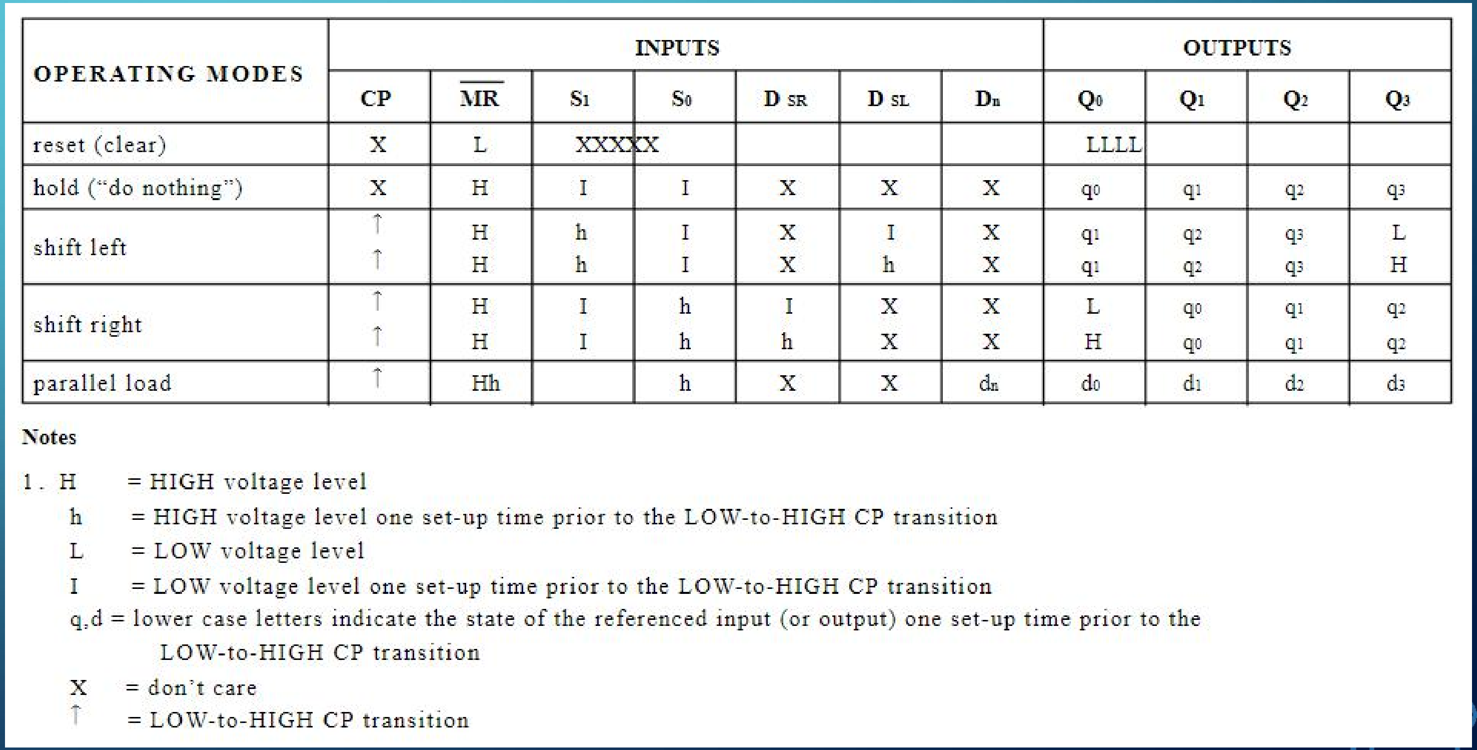


4-BIT BIDIRECTIONAL UNIVERSAL SHIFT REGISTER
The SN54/74LS194A is a High Speed 4-Bit Bidirectional Universal Shift Register. As a high speed multifunctional sequential building block, it is useful in a wide variety of applications. It may be used in serial-serial, shift left, shift right, serial-parallel, parallel-serial, and parallel-parallel data register transfers.

SHIFT REGISTER 74194



```
module Shift_Register_74194(
    input MR_n, CP, DSR, DSL, // Clear, Clock, Serial input
    input [1:0] S, //Select input
    input D3, D2, D1, D0, //Parallel input
    output reg Q3, Q2, Q1, Q0//Parallel output
    );
    always @(posedge CP, negedge MR_n)
        if(!MR_n)
            {Q3, Q2, Q1, Q0} <= 4'b0000;
        else
            case (S)
            2'b00:{Q3, Q2, Q1, Q0}<={Q3, Q2, Q1, Q0};
            2'b01:{Q3, Q2, Q1, Q0}<={DSR, Q3, Q2, Q1};
            2'b10:{Q3, Q2, Q1, Q0}<={Q2, Q1, Q0, DSL};
            2'b11:{Q3, Q2, Q1, Q0}<={D3, D2, D1, D0};
            endcase
endmodule
```



OPERATING MODES	INPUTS							OUTPUTS			
	CP	$\overline{MR}$	S_1	S_0	D_{SR}	D_{SL}	D_n	Q_0	Q_1	Q_2	Q_3
reset (clear)	X	L	XXXXX					LLLL			
hold ("do nothing")	X	H	I	I	X	X	X	q_0	q_1	q_2	q_3
shift left	↑ ↑	H H	h h	I I	X X	I h	X X	q_1 q_1	q_2 q_2	q_3 q_3	L H
shift right	↑ ↑	H H	I I	h h	I h	X X	X X	L H	q_0 q_0	q_1 q_1	q_2 q_2
parallel load	↑	Hh		h	X	X	d_n	d_0	d_1	d_2	d_3

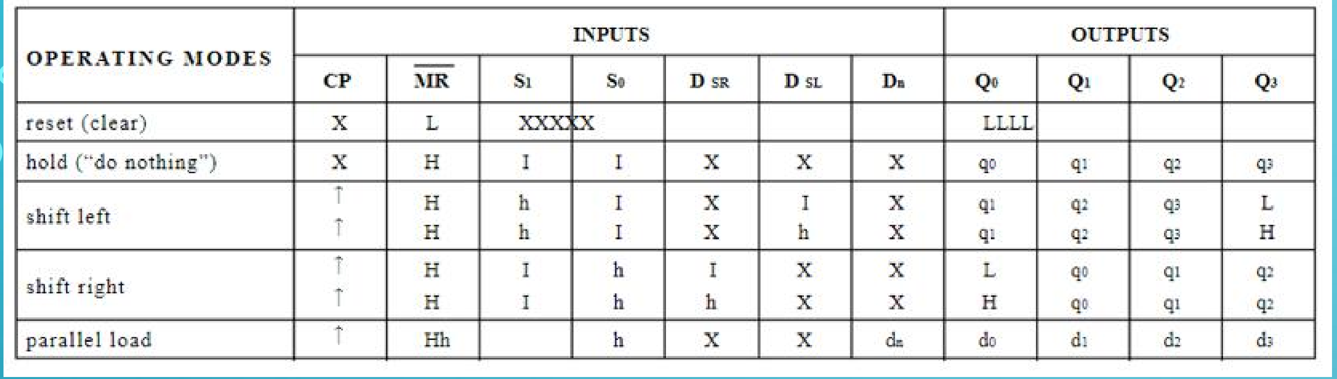
Notes

1. H = HIGH voltage level
   h = HIGH voltage level one set-up time prior to the LOW-to-HIGH CP transition
   L = LOW voltage level
   I = LOW voltage level one set-up time prior to the LOW-to-HIGH CP transition
   q,d = lower case letters indicate the state of the referenced input (or output) one set-up time prior to the LOW-to-HIGH CP transition
   X = don't care
   ↑ = LOW-to-HIGH CP transition

COUNTER

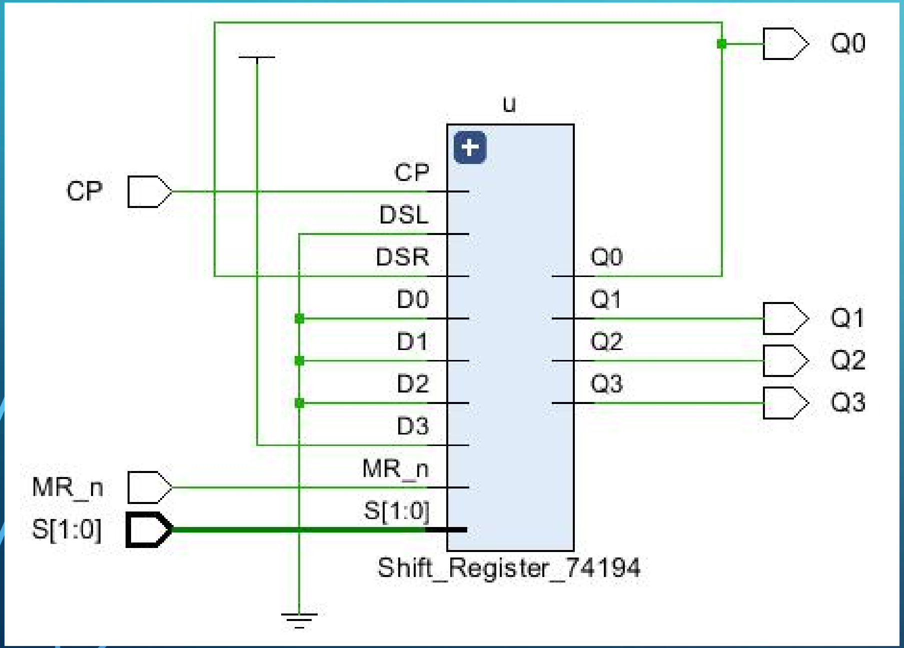
- In digital logic and computing, a **counter** is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. The most common type is a sequential digital logic circuit with an input line called the *clock* and multiple output lines. The values on the output lines represent a number in the binary or BCD number system. Each pulse applied to the clock input increments or decrements the number in the counter.
- A counter circuit is usually constructed of a number of flip-flops connected in cascade. Counters are a very widely used component in digital circuits, and are manufactured as separate integrated circuits and also incorporated as parts of larger integrated circuits

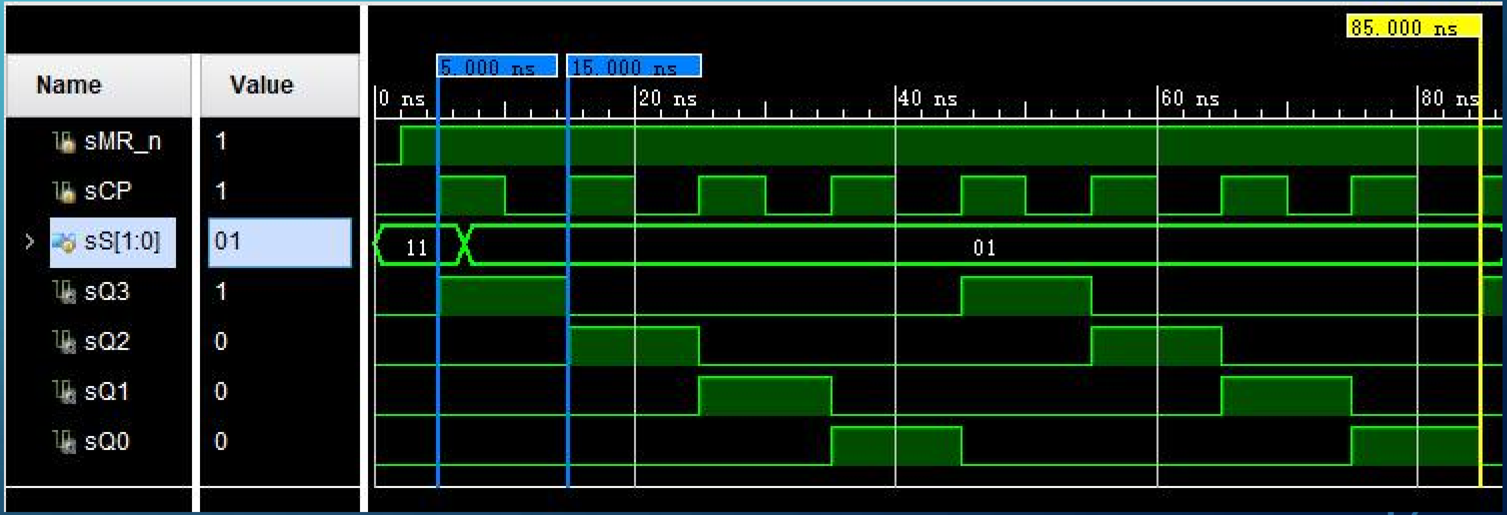
RING COUNTER—USING 74194



OPERATING MODES	INPUTS							OUTPUTS			
	CP	$\overline{MR}$	S_1	S_0	D_{SR}	D_{SL}	D_n	Q_0	Q_1	Q_2	Q_3
reset (clear)	X	L	XXXXX					LLLL			
hold ("do nothing")	X	H	I	I	X	X	X	q_0	q_1	q_2	q_3
shift left	↑	H	h	I	X	I	X	q_1	q_2	q_3	L
	↑	H	h	I	X	h	X	q_1	q_2	q_3	H
shift right	↑	H	I	h	I	X	X	L	q_0	q_1	q_2
	↑	H	I	h	h	X	X	H	q_0	q_1	q_2
parallel load	↑	Hh		h	X	X	d_n	d_0	d_1	d_2	d_3

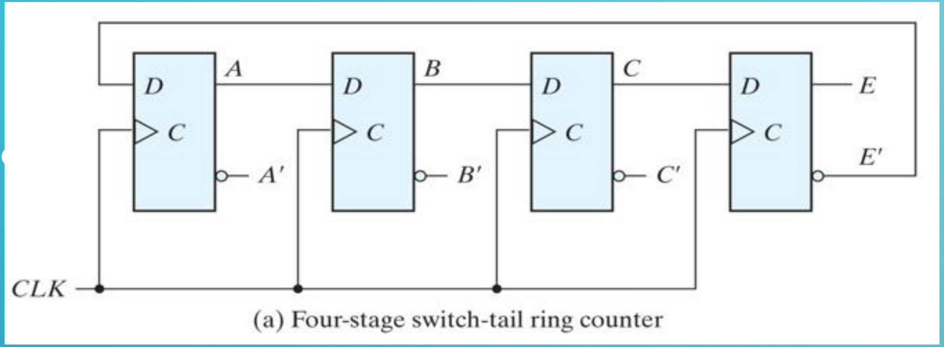
sequence number	Q3	Q2	Q1	Q0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1



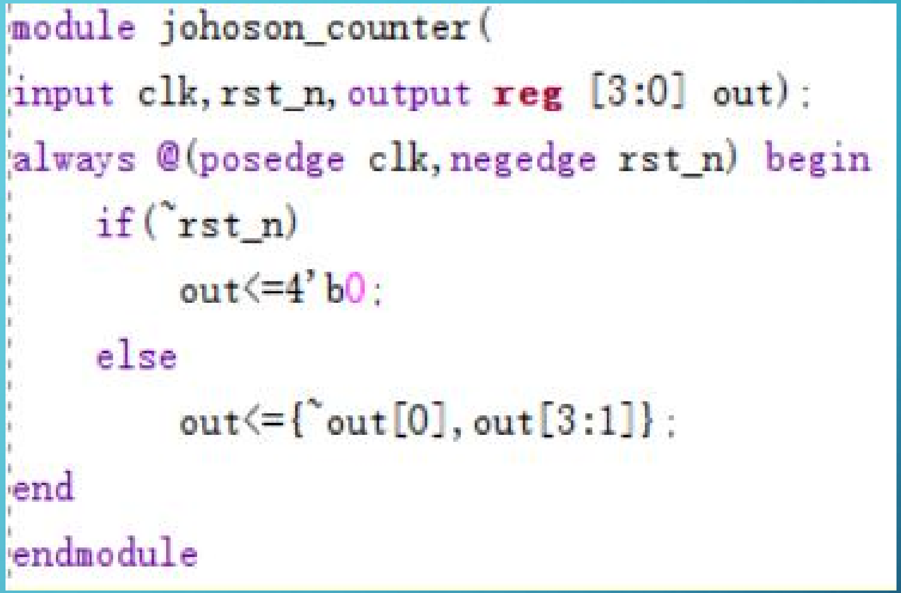


Name	Value
sMR_n	1
sCP	1
sS[1:0]	01
sQ3	1
sQ2	0
sQ1	0
sQ0	0

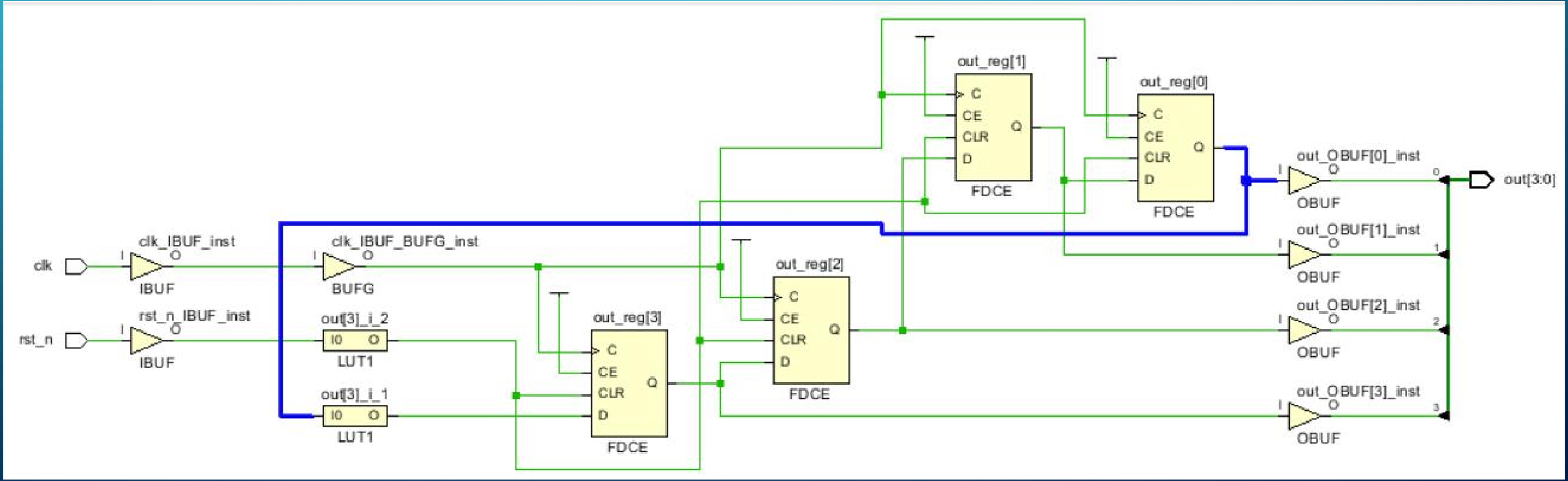
JOHNSON-COUNTER(1)



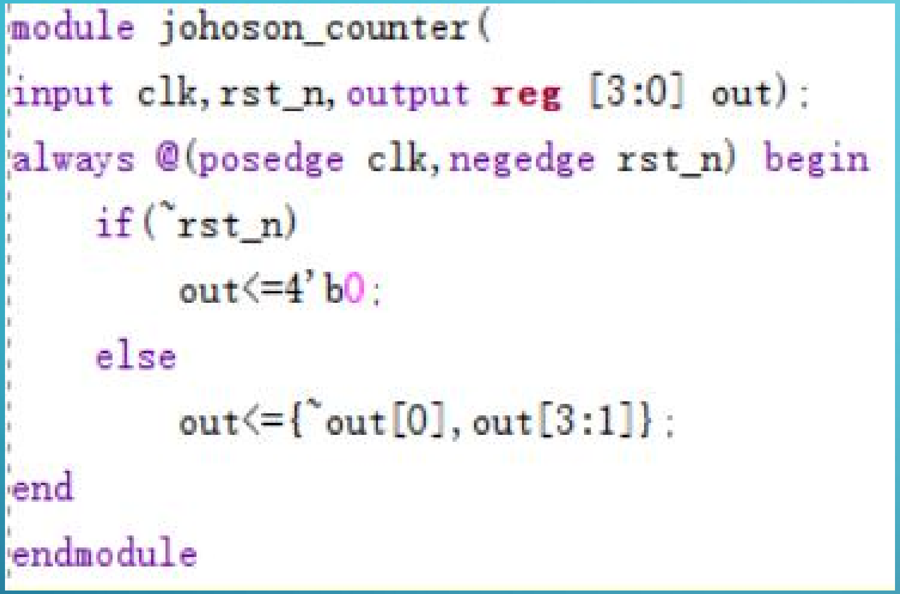
(a) Four-stage switch-tail ring counter



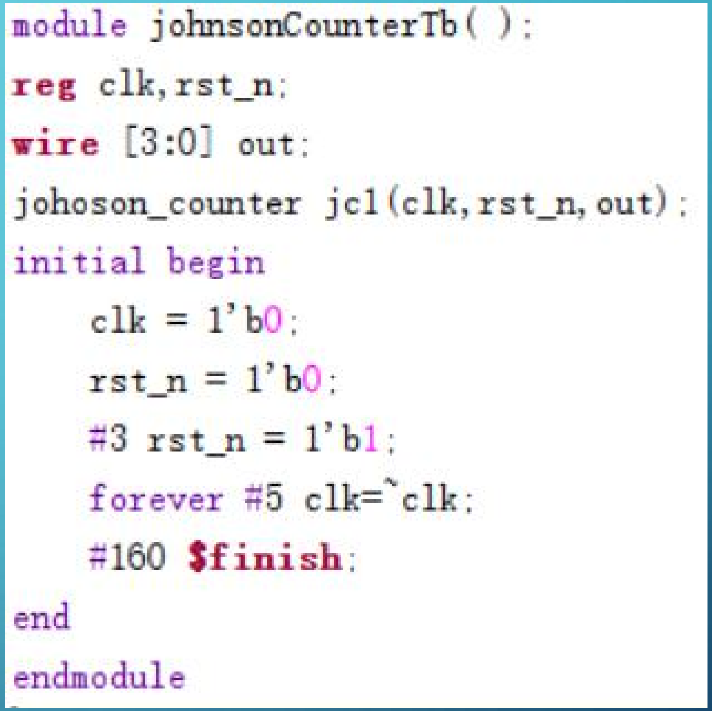
```
module johoson_counter(
input clk,rst_n,output reg [3:0] out);
always @(posedge clk,negedge rst_n) begin
    if(~rst_n)
        out<=4'b0;
    else
        out<={~out[0],out[3:1]};
end
endmodule
```



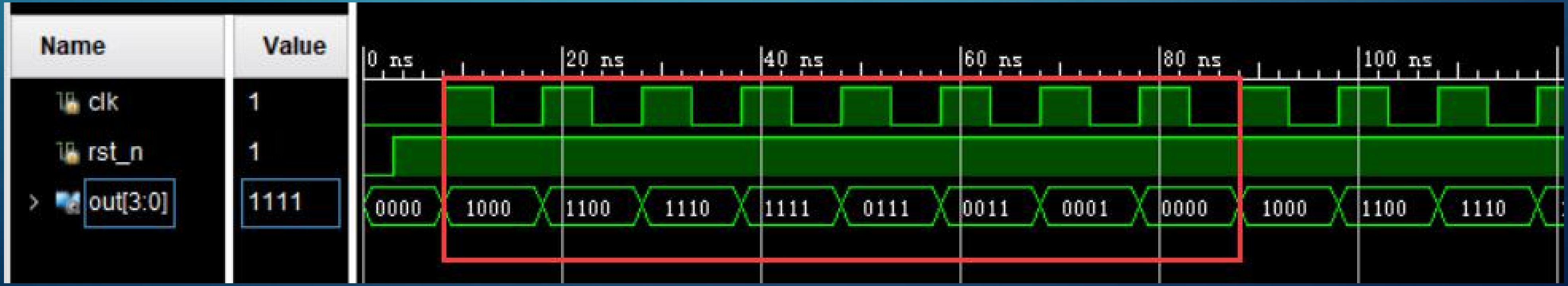
JOHNSON-COUNTER(2)



```
module johoson_counter(
input clk,rst_n,output reg [3:0] out);
always @(posedge clk,negedge rst_n) begin
    if(~rst_n)
        out<=4'b0;
    else
        out<={~out[0],out[3:1]};
end
endmodule
```



```
module johnsonCounterTb( );
reg clk,rst_n;
wire [3:0] out;
johoson_counter jc1(clk,rst_n,out);
initial begin
    clk = 1'b0;
    rst_n = 1'b0;
    #3 rst_n = 1'b1;
    forever #5 clk=~clk;
    #160 $finish;
end
endmodule
```



PRACTICE

Use Two 74194 to implement a 8-bits serial-parallel Converter.

- Do the design by using behavior modeling in verilog
- Verify its function using test-bench.
- Try to implement the circuit on EGO1 board