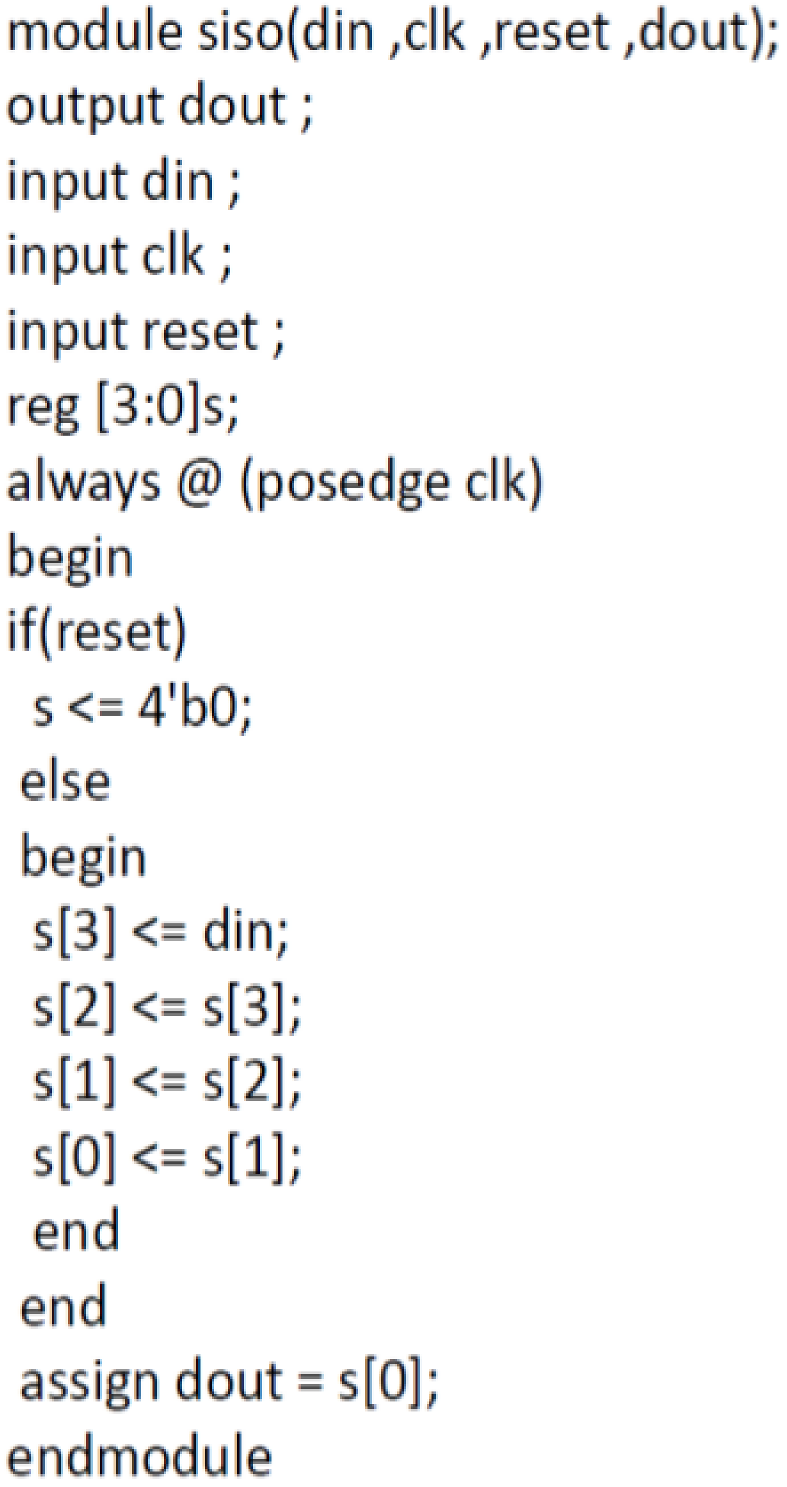
Shift Registers

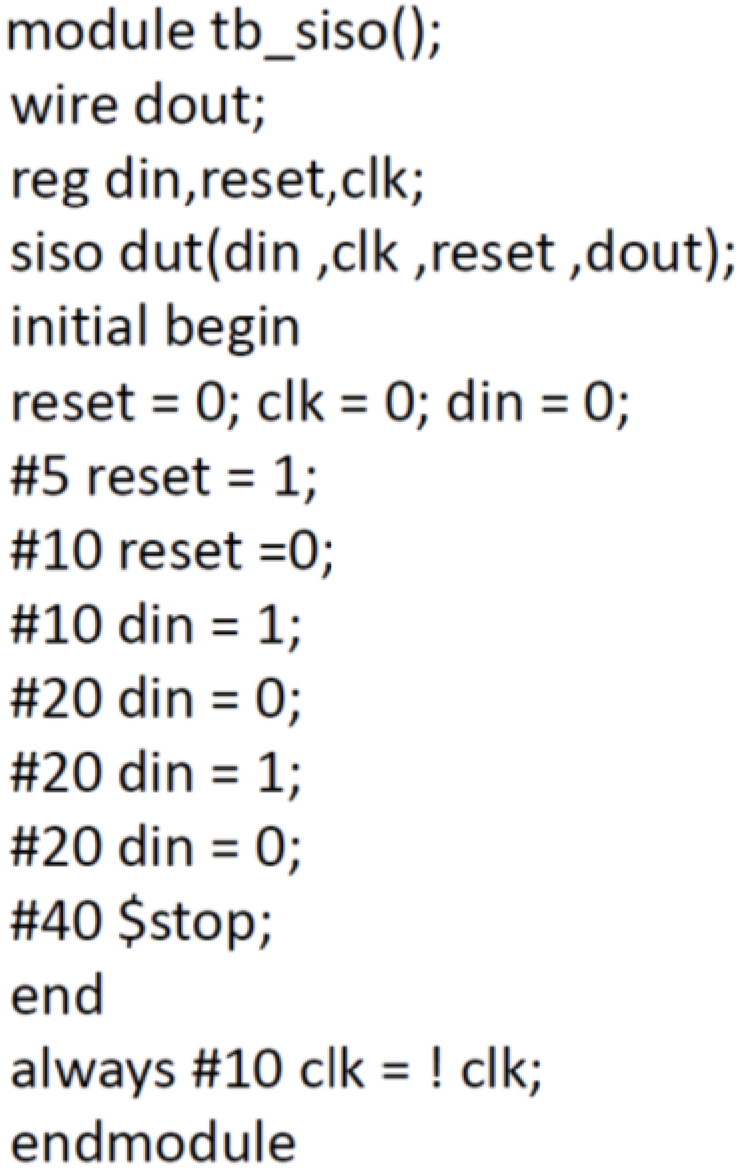
To design (a) SISO (b) SIPO (c) PIPO (d) PISO   using Verilog

(i) SISO



```
module siso(din ,clk ,reset ,dout);
output dout ;
input din ;
input clk ;
input reset ;
reg [3:0]s;
always @ (posedge clk)
begin
if(reset)
  s <= 4'b0;
 else
 begin
  s[3] <= din;
  s[2] <= s[3];
  s[1] <= s[2];
  s[0] <= s[1];
  end
 end
 assign dout = s[0];
endmodule
```

Testbench

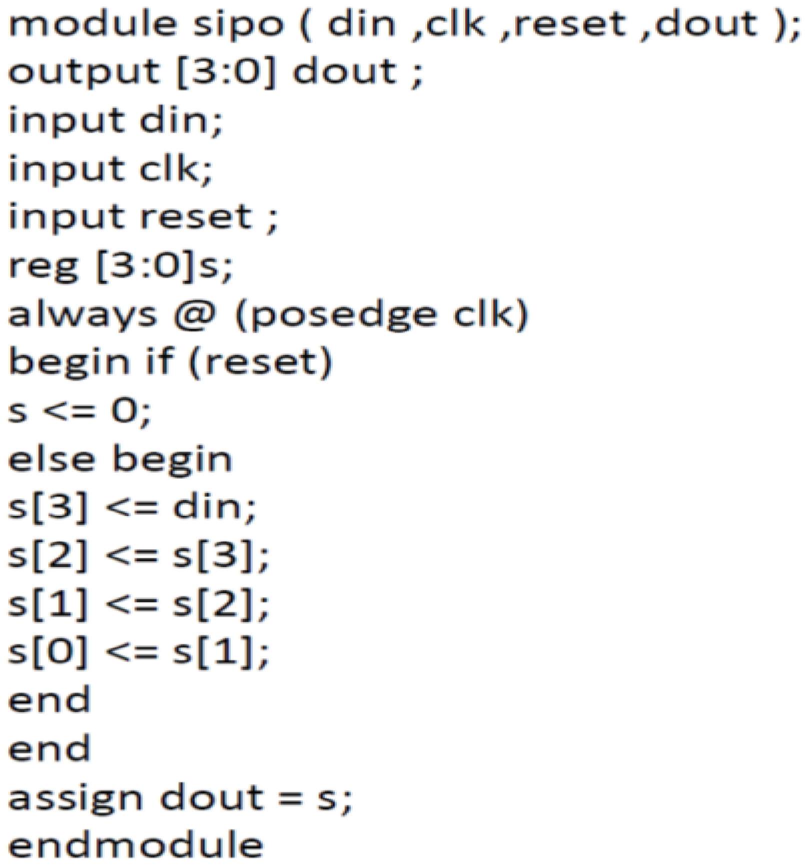


```
module tb_siso();
wire dout;
reg din,reset,clk;
siso dut(din ,clk ,reset ,dout);
initial begin
reset = 0; clk = 0; din = 0;
#5 reset = 1;
#10 reset =0;
#10 din = 1;
#20 din = 0;
#20 din = 1;
#20 din = 0;
#40 $stop;
end
always #10 clk = ! clk;
endmodule
```

Simulation Waveform

Report Utilization Summary

(ii)SIPO



```
module sipo ( din ,clk ,reset ,dout );
output [3:0] dout ;
input din;
input clk;
input reset ;
reg [3:0]s;
always @ (posedge clk)
begin if (reset)
s <= 0;
else begin
s[3] <= din;
s[2] <= s[3];
s[1] <= s[2];
s[0] <= s[1];
end
end
assign dout = s;
endmodule
```

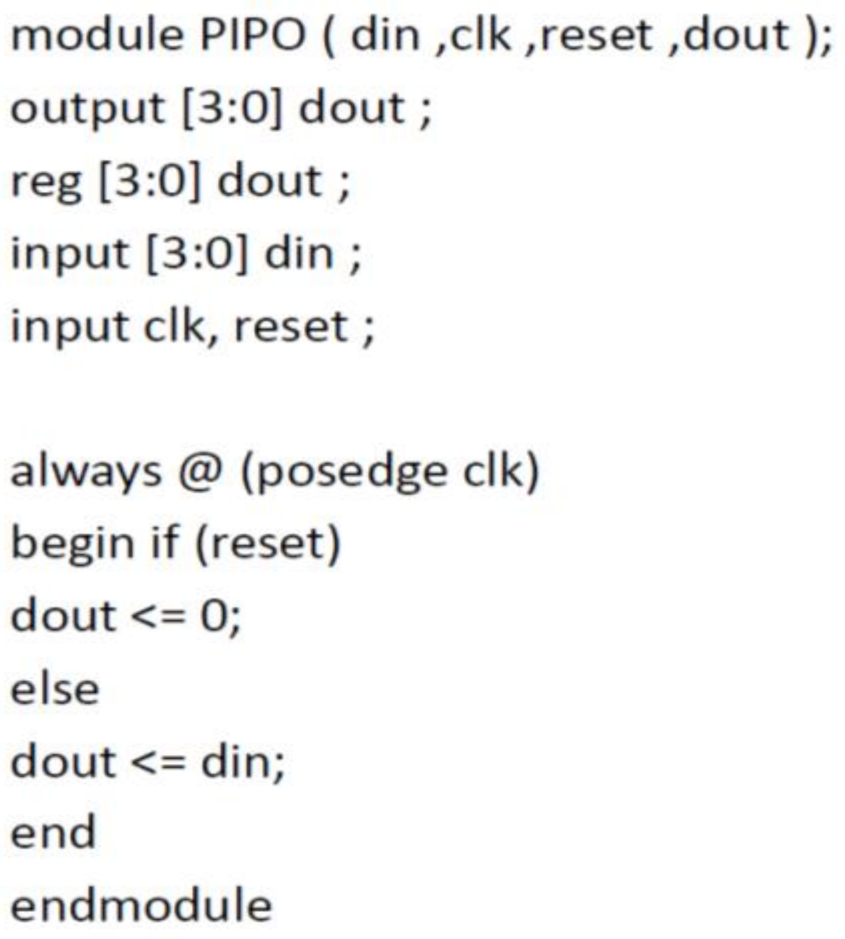
Testbench

```
module tb_sipo;
 reg din,clk,reset;
wire [3:0] dout;
sipo uut(din, clk, reset,dout);

initial
begin
reset=0;
clk=0;
din=0;
#20 reset =1;
#20 reset=0;
#20 din=1;
#20 din=0;
#20 din=1;
#20 din=0;
#20 din=1;
#20 $finish;
end
always #10 clk = ~ clk;
endmodule
```

Simulation Waveform

Report Utilization Summary

(iii)PIPO



```
module PIPO ( din ,clk ,reset ,dout );
output [3:0] dout ;
reg [3:0] dout ;
input [3:0] din ;
input clk, reset ;

always @ (posedge clk)
begin if (reset)
dout <= 0;
else
dout <= din;
end
endmodule
```

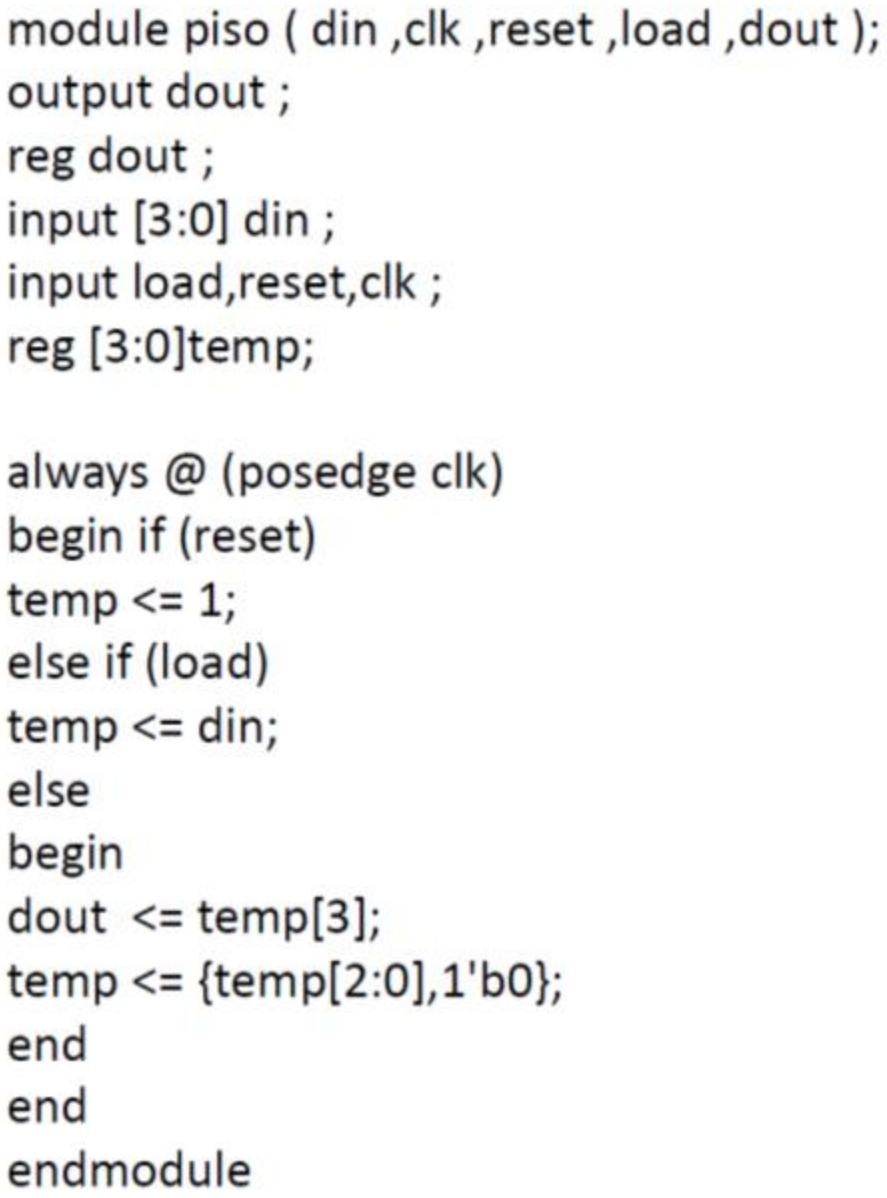
Testbench

```
module tb_pipo;
 reg [3:0] din;

reg clk,reset;
wire [3:0] dout;
pipo uut(din, clk, reset,dout);

initial
begin
reset=0;
clk=0;
din=4'b0000;
#20 reset =1;
#20 reset=0;
#60 din=4'b1001;
#60 din=4'b1010;
#60 din=4'b1111;

#60 $finish;
end
always #10 clk = ~ clk;
endmodule
```

Simulation Waveform

Report Utilization Summary

(iv) PISO



```
module piso ( din ,clk ,reset ,load ,dout );
output dout ;
reg dout ;
input [3:0] din ;
input load,reset,clk ;
reg [3:0]temp;

always @ (posedge clk)
begin if (reset)
temp <= 1;
else if (load)
temp <= din;
else
begin
dout  <= temp[3];
temp <= {temp[2:0],1'b0};
end
end
endmodule
```

Testbench

```
module tb_piso;
 reg [3:0] din;

reg clk,reset,load;
wire dout;
piso uut(din, clk, reset,loaddout);

initial
begin
reset=0;
load=0;
clk=0;
din=4'b0000;
#20 reset =1;
```

```
#20 reset=0;
#20 load =1;
#20 load =0;

#60 din=4'b1001;
#60 din=4'b1010;
#60 din=4'b1111;

#60 $finish;
end
always #10 clk = ~ clk;
endmodule
```

Simulation Waveform

Report Utilization Summary