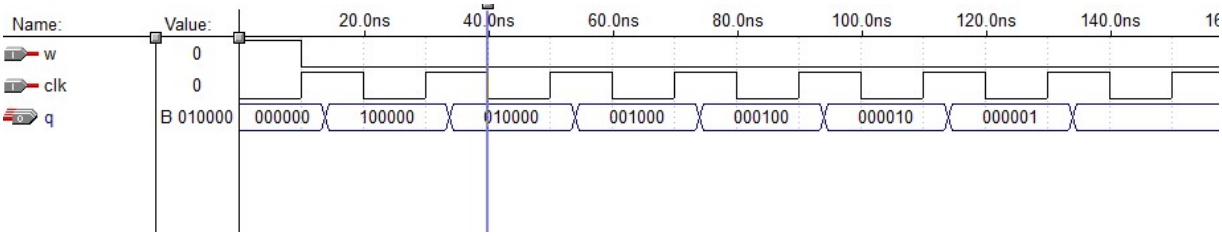
Q 4. 6-bit shift register

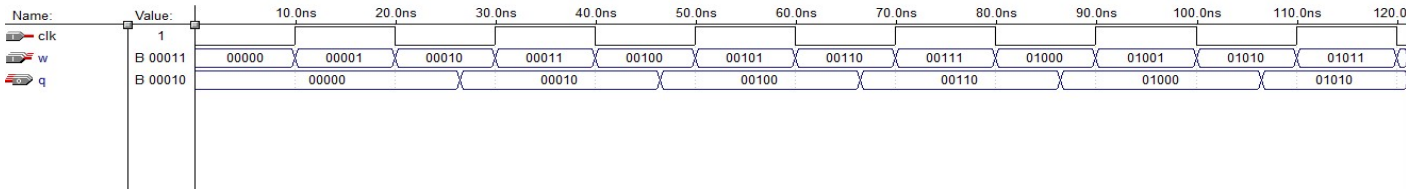
```
module shiftright(w,clk,q);
input w,clk;
output [5:0]q;
reg [5:0]q;
parameter n=6;
integer i;
always@(posedge clk)
begin
for(i=0;i<n-1;i=i+1)
q[i]<=q[i+1];
q[n-1]<=w;
end
endmodule
```



Q3. 5-bit register

```
module dff1(d,clk,q);
input d,clk;
output q;
reg q;
```

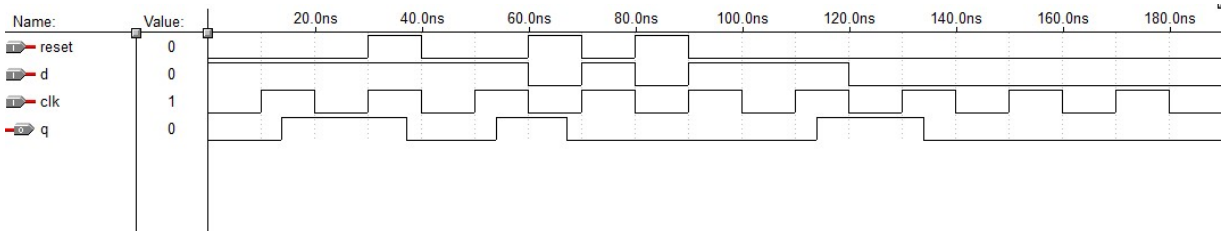
```
always@(negedge clk)
begin
q<=d;
end
endmodule
module reg5bit(w,clk,q);
input [4:0]w;
input clk;
output [4:0]q;
reg [4:0]q;
dff1 s0(w[0],clk,q[0]);
dff1 s1(w[1],clk,q[1]);
dff1 s2(w[2],clk,q[2]);
dff1 s3(w[3],clk,q[3]);
dff1 s4(w[4],clk,q[4]);
endmodule
```



Q1. D-flip-flop

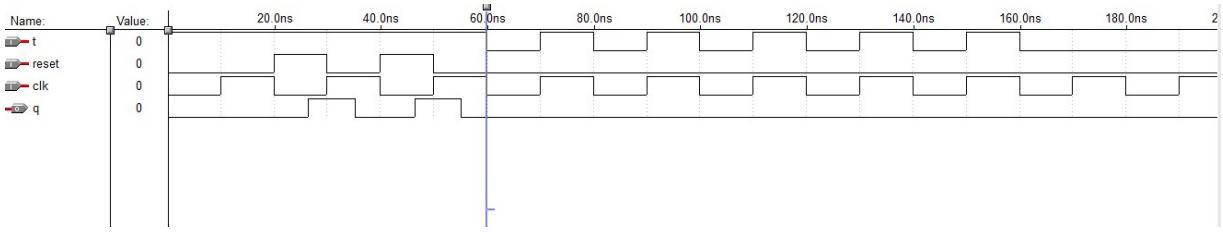
```
module dff(d,reset,clk,q);
input d,clk,reset;
output q;
reg q;
always@(posedge reset or posedge clk)
```

```
if (reset)
q=0;
else
q=d;
endmodule
```



Q2. T-flip-flop

```
module tff(t,clk,reset,q);
input t,clk,reset;
output q;
reg q;
always@(negedge reset or negedge clk)
begin
if(!reset)
q=0;
else
begin
if(t)
q=~q;
end
end
endmodule
```



Q3. JK-flipflop

```
module jkff(clk,q,j,k,reset);
input clk,j,k,reset;
output q;
reg q;
always@(posedge clk)
begin
casex({reset,j,k})
3'b1xx:q<=0;
3'b001:q<=0;
3'b010:q<=1;
3'b011:q<=~q;
3'b000:q<=q;

endcase


end
endmodule
```