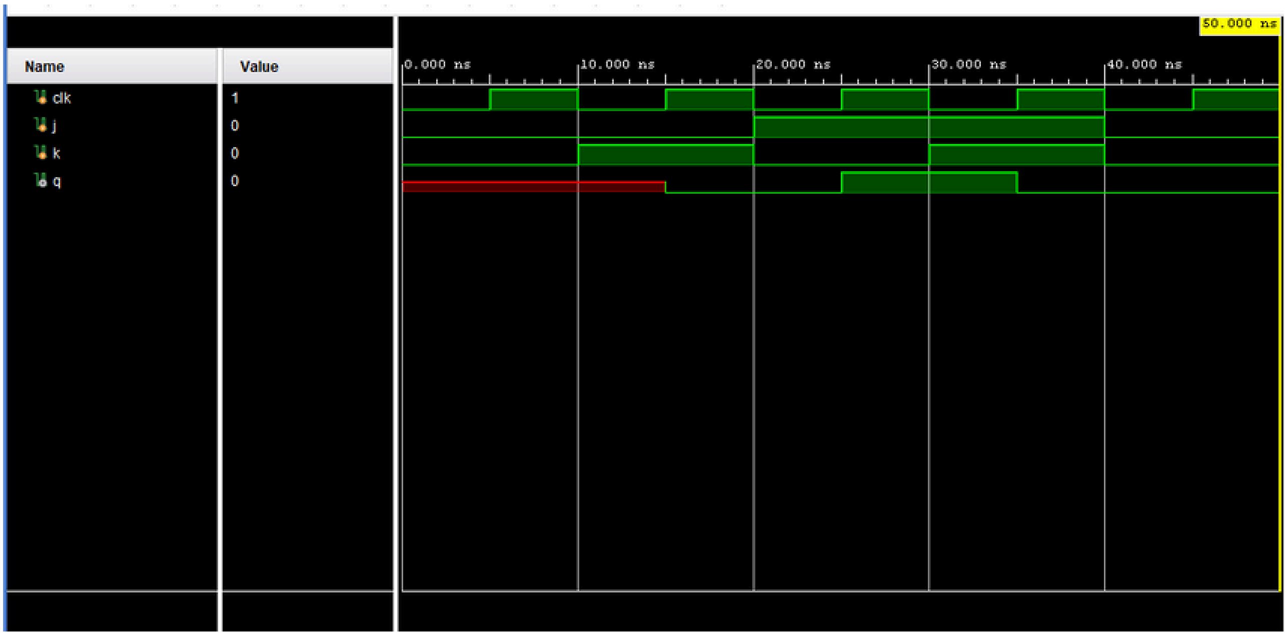
36.JK Flipflop

Verilog Code:

```
 module jk_flipflop(j,k,clk,q );
 input j,k,clk;
output reg q;
always @( posedge clk)
begin case({j,k})
 2'b00:q=q;
2'b01:q=1'b0;
2'b10:q=1'b1;
2'b11:q=~q;
endcase
 end endmodule
```

TestBench:

```
module tb8();
reg clk,j,k;
wire q;
jk_flipflop uut(j,k,clk,q);
initial
begin
clk=0;
forever #5 clk=~clk;
end
initial
begin
j=0; k=0; #10
j=0; k=1; #10
j=1; k=0; #10
j=1; k=1; #10
j=0; k=0; #10
$finish;
end
endmodule
```



Name	Value
clk	1
j	0
k	0
q	0

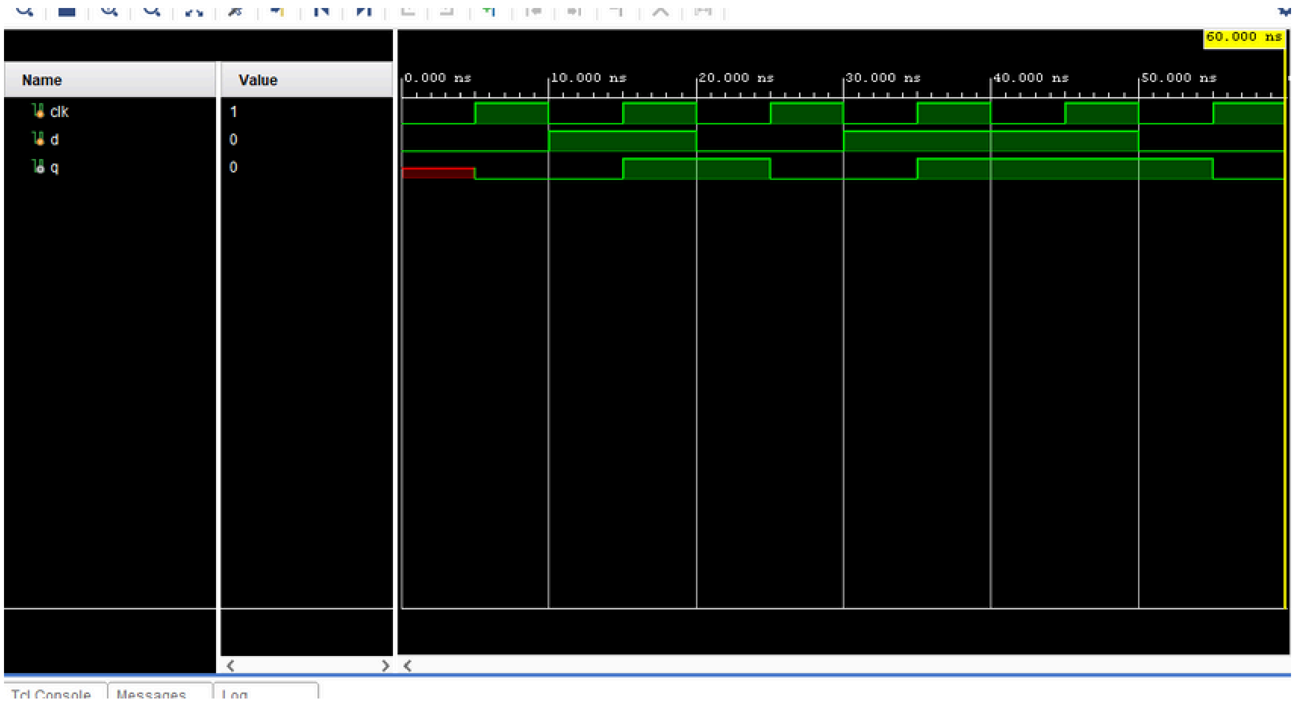
37.D Flipflop

Verilog Code:

```
module
d_flipflop(d,clk,q); input
d,clk; output reg q;
always @(posedge clk)
begin case(d) 1'b0: q=0;
1'b1: q=1; endcase end
endmodule
```

TestBench:

```
module tb9();
reg clk,d;
wire q;
d_flipflop uut(d,clk,q);
initial
begin
clk=0;
forever #5 clk=~clk;
end
initial
begin
d=0;#10
d=1;#10
d=0;#10
d=1;#10
d=1;#10
d=0;#10
$finish;
end
endmodule
```



38.T Flipflop

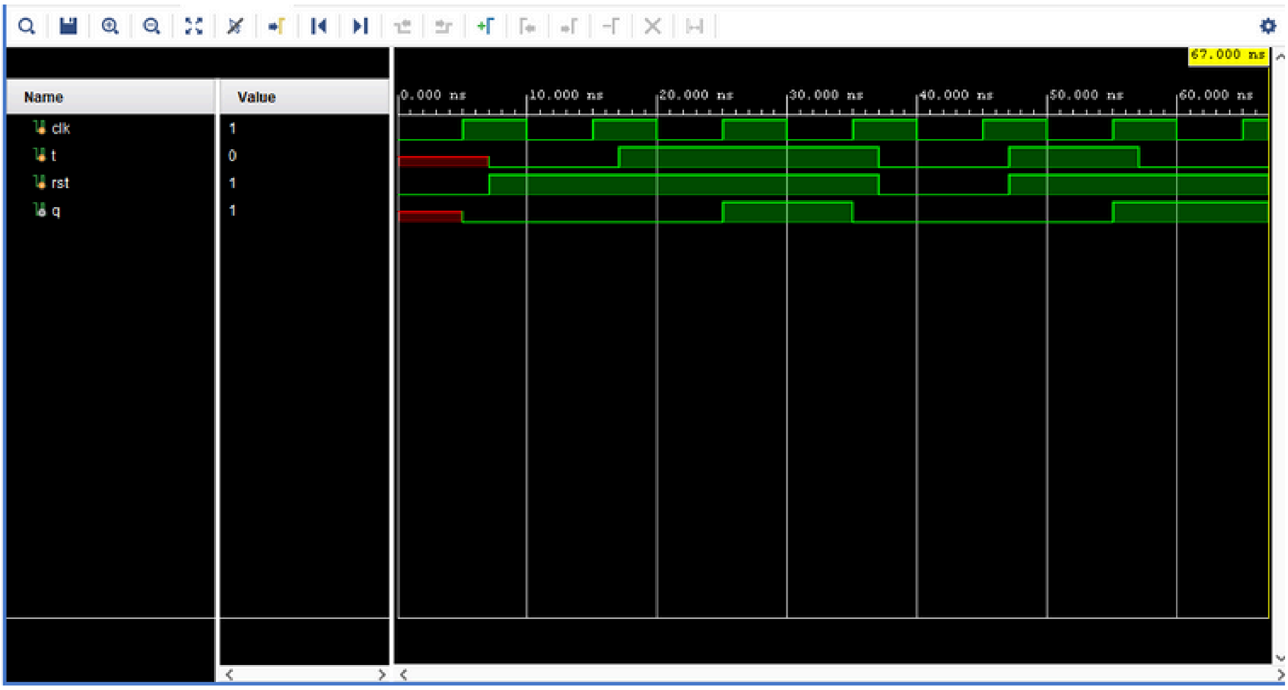
Verilog Code:

```
module t_ff(t,rst,clk,q );
input clk,t,rst;
output reg q;
always@(posedge clk)
begin
if(!rst)
q=1'b0;
else if(t)
q=~q;
end
endmodule
```

TestBench:

```
module tb10();
reg clk,t,rst;
wire q;
t_ff uut(t,rst,clk,q);
initial
begin
clk=0;
forever #5 clk=~clk;
end
initial
begin
```

```
rst=0;#7
rst=1;t=0;
#10     t=1;
#10     t=1;
#10
rst=0;t=0;
#10
rst=1;t=1;
#10     t=0;
#10
$finish;
end
endmodul
```



Name	Value
clk	1
t	0
rst	1
q	1

39.D Latch

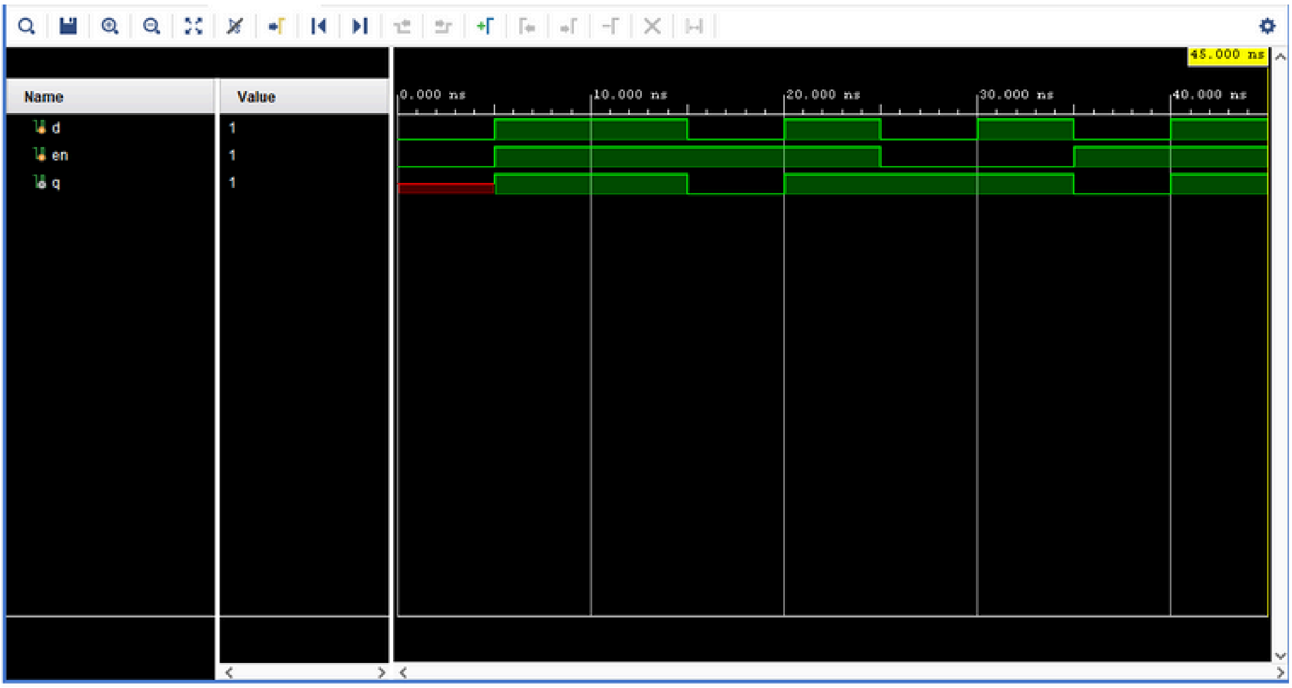
Verilog Code:

```
module d_latch(d,en,q);
input d,en;
output reg q;
always @(d,en)
begin
if(en)
q=d;
end
endmodule
```

TestBench:

```
module tb11( );
reg d,en;
```

```
wire   q;   d_latch
uut(d,en,q);  initial
begin  d=0;  en=0;
#5 d=1; en=1; #10
d=0;   #5   d=1;#5
en=0;         d=0;#5
d=1;#5         en=1;
d=0;   #5   d=1;#5
$finish;
end
endmodule
```



40.Asynchronous counter using T Flipflops

Verilog Code:

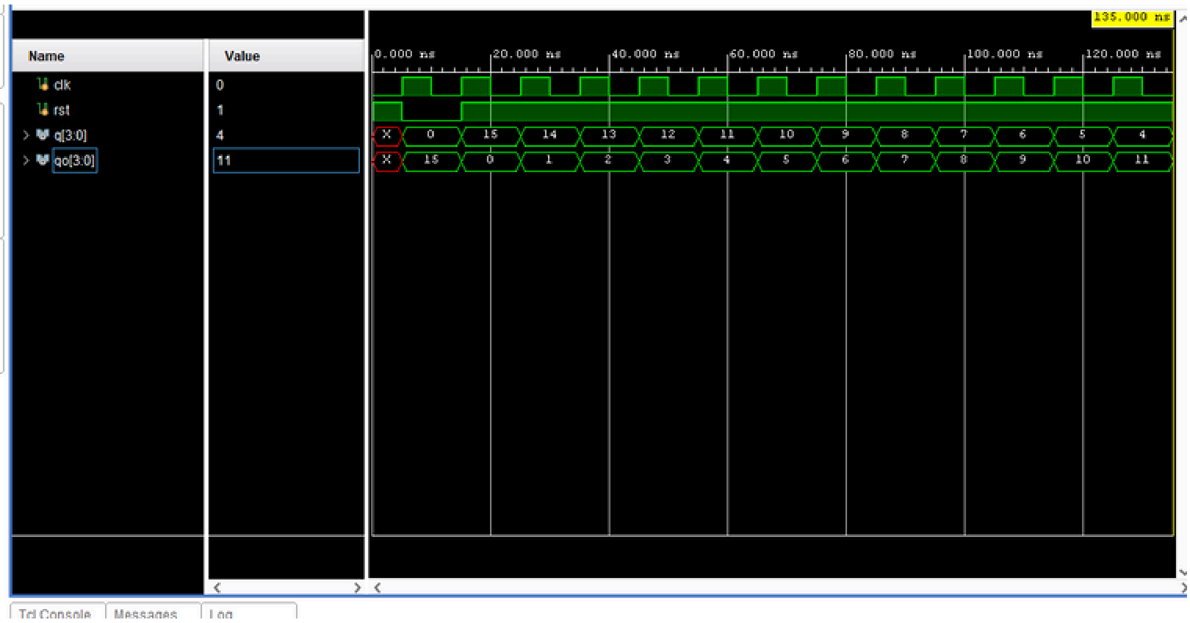
```
module t_ff(t,clk,rst,q);
input t,clk,rst;
output reg q;
always @(posedge clk,negedge rst)
begin
if(!rst)
q<=0;
else if(t)
q<=~q;
```

```
else
q<=q;
end
endmodule

module asyn_coun_t_ff(clk,rst,q,qo);
output [3:0]q;
output [3:0]qo;
input clk,rst;
assign t=1;
t_ff t1(t,clk,rst,q0);
t_ff t2(t,q0,rst,q1);
t_ff t3(t,q1,rst,q2);
t_ff t4(t,q2,rst,q3);
assign q={q3,q2,q1,q0};
assign qo=~q;
endmodule
```

TestBench:

```
module tb2( );
reg clk,rst;
wire[3:0]q,qo;
asyn_coun_t_ff uut(clk,rst,q,qo);
initial
begin
clk=0;
forever #5 clk=~clk;
end
initial
begin
rst=1; #5
rst=0;#10
rst=1;#120
$finish;
end
endmodule
```


135.000 ns
Name
Value
clk 0
rst 1
q[3:0] 4
qo[3:0] 11
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
X 0 15 14 13 12 11 10 9 8 7 6 5 4
X 15 0 1 2 3 4 5 6 7 8 9 10 11
Tcl Console
Messages
Log