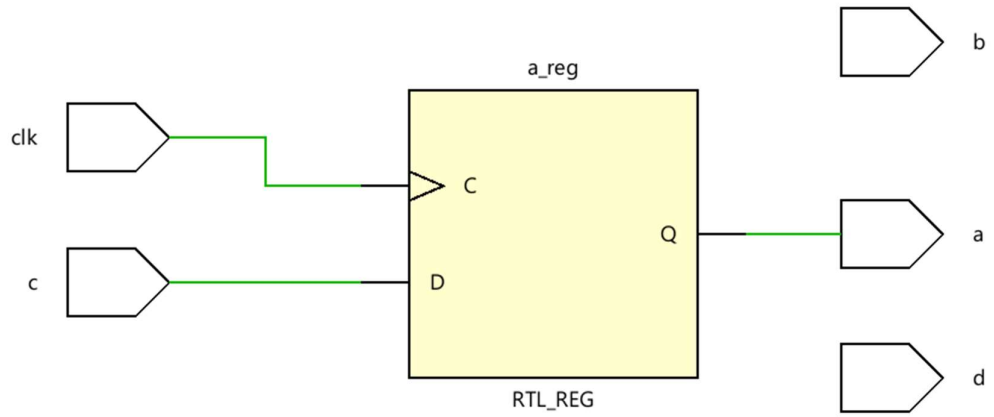
Blocking Vs Non Blocking

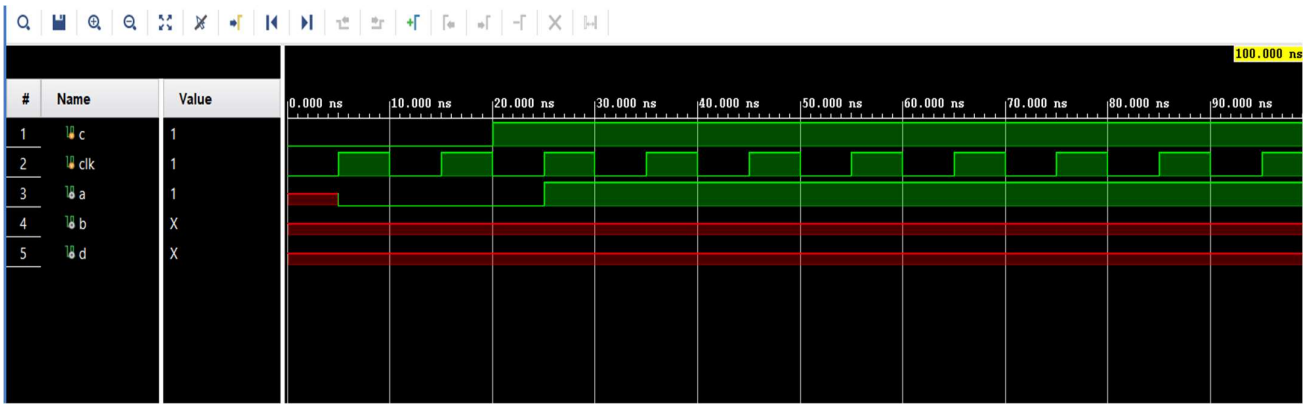
Code

```
module check( input  clk,c,output reg a,b,d   );



always@(posedge clk)
begin

a=c;



end
endmodule
```

Circuit



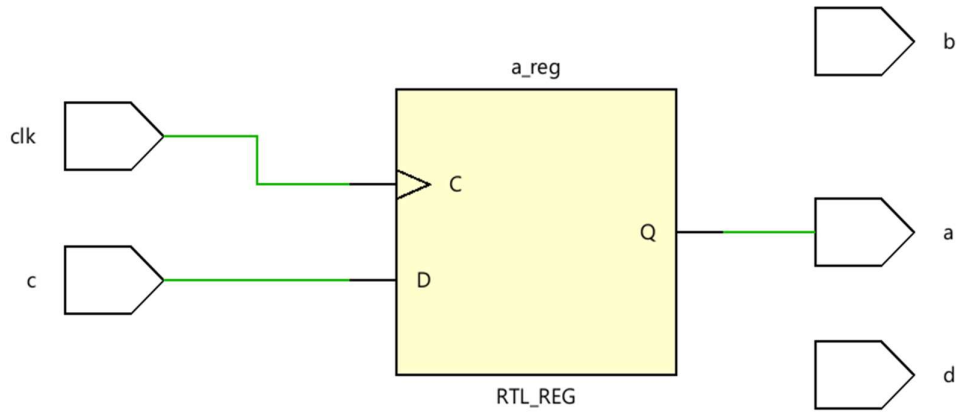
Simulation


100.000 ns
Name Value
1 c 1
2 clk 1
3 a 1
4 b X
5 d X
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns
80.000 ns
90.000 ns

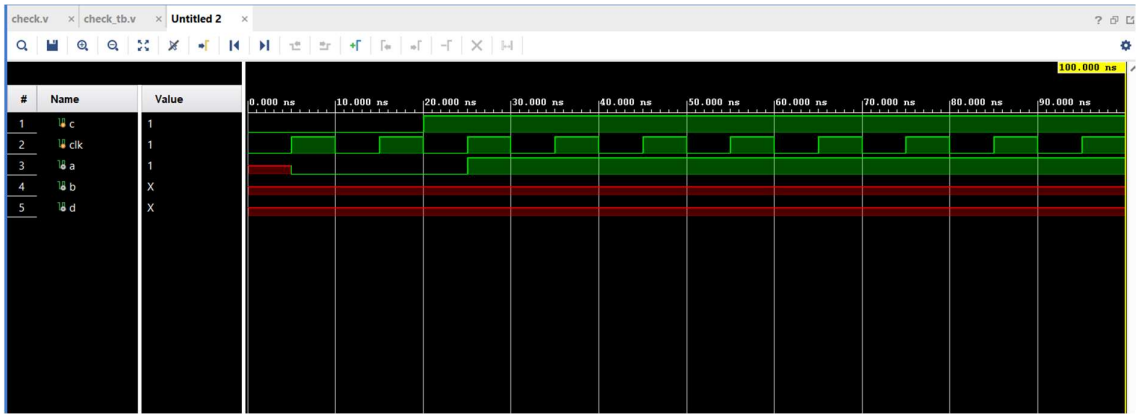
Code

```
module check( input  clk,c,output reg a,b,d   );


always@(posedge clk)
begin

a<=c;


end
endmodule
```

Circuit



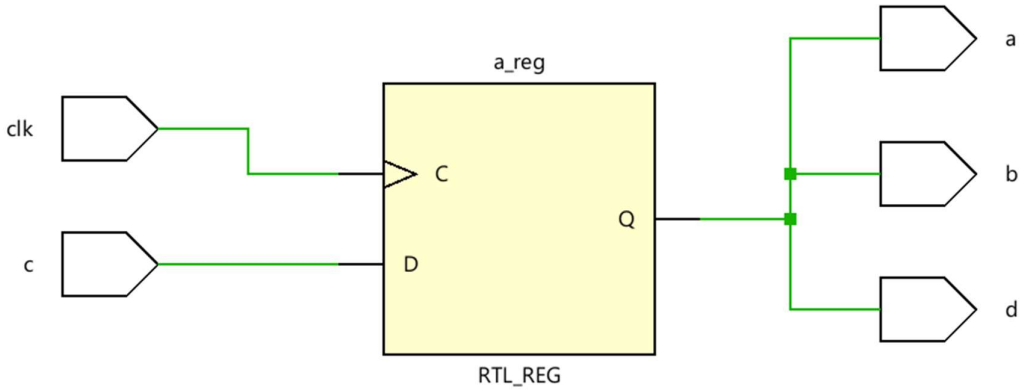
Simulation


check.v
check_tb.v
Untitled 2
100.000 ns
Name Value
1 c 1
2 clk 1
3 a 1
4 b X
5 d X
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns
80.000 ns
90.000 ns

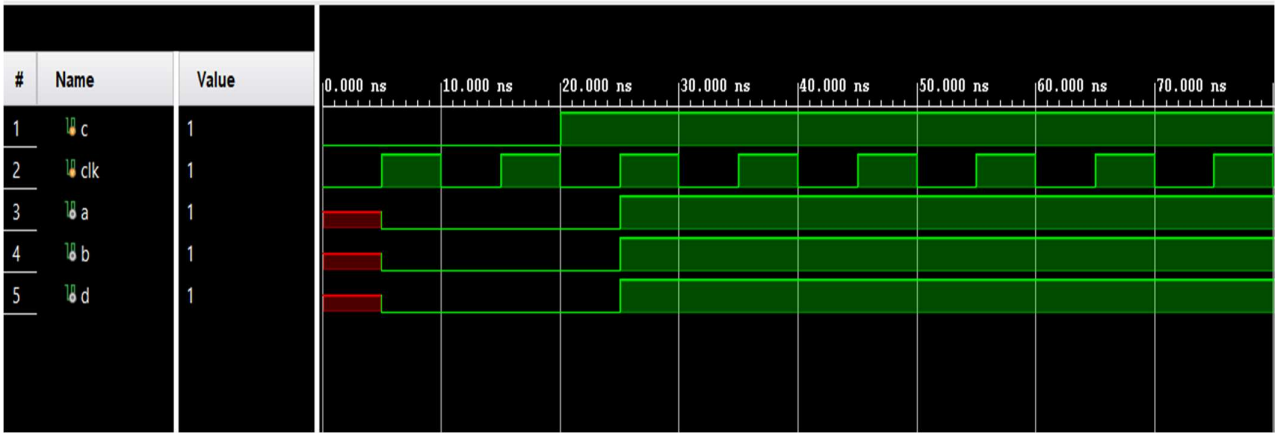
Code

```
module check( input  clk,c,output reg a,b,d   );



always@(posedge clk)
begin

a=c;
b=a;
d=b;


end
endmodule
```

Circuit



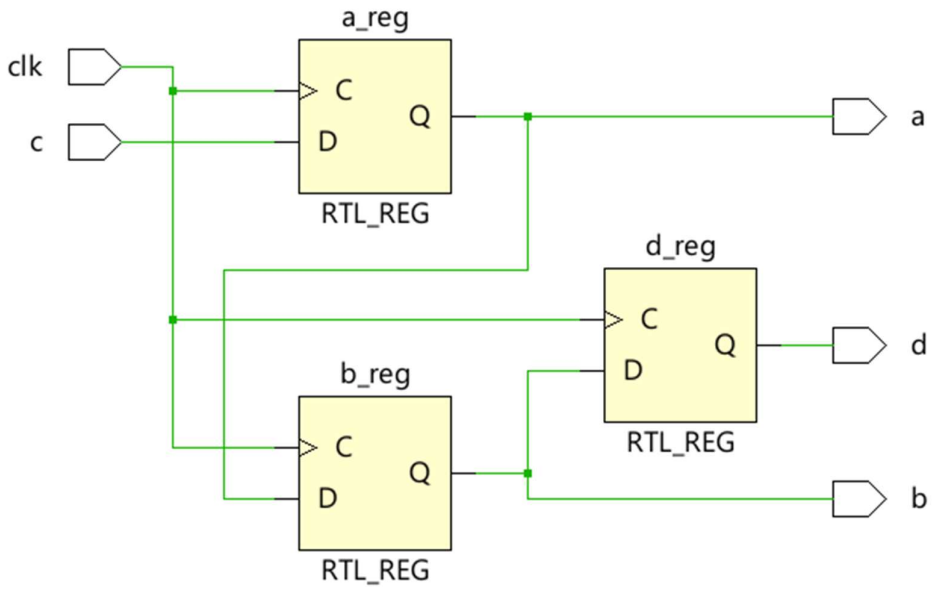
Simulation


#
Name
Value
1
c
1
2
clk
1
3
a
1
4
b
1
5
d
1
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns

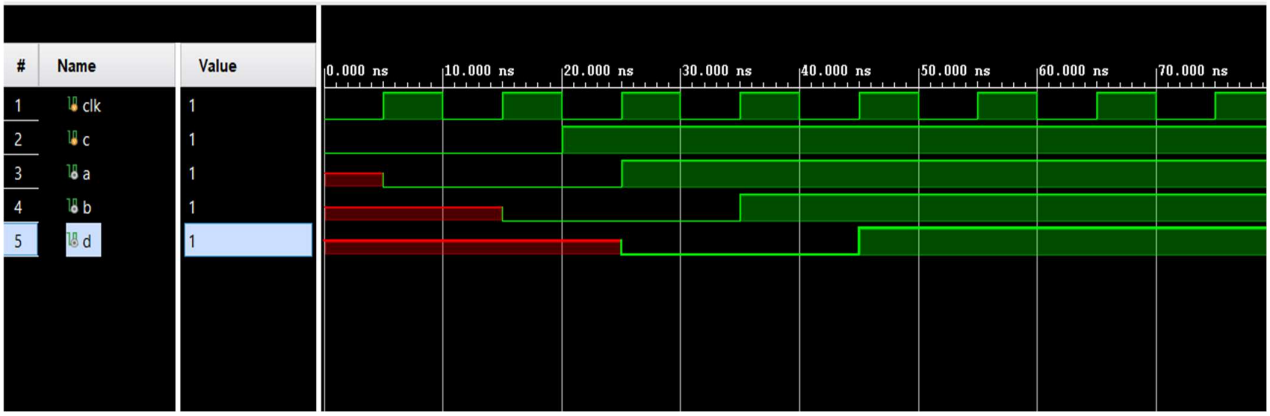
Code

```
module check( input  clk,c,output reg a,b,d   );


always@(posedge clk)
begin

 a<=c;
 b<=a;
 d<=b;


end
endmodule
```

Circuit



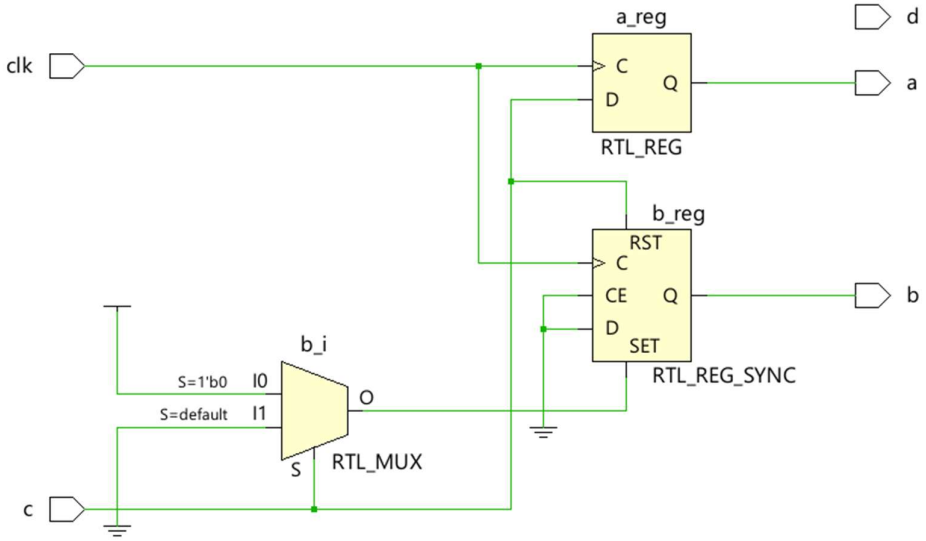
Simulation


#
Name
Value
1
clk
1
2
c
1
3
a
1
4
b
1
5
d
1
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns

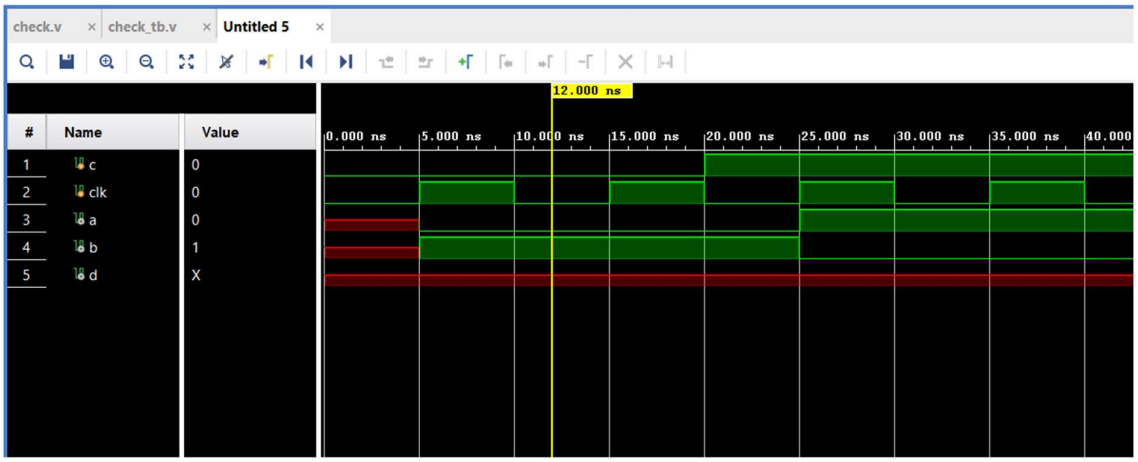
Code

```
module check( input  clk,c,output reg a,b,d   );


always@(posedge clk)
begin
a=c;
if(a==0)
b<=1;
else if(a==1)
b<=0;
//d=b;


end
endmodule
```

Circuit



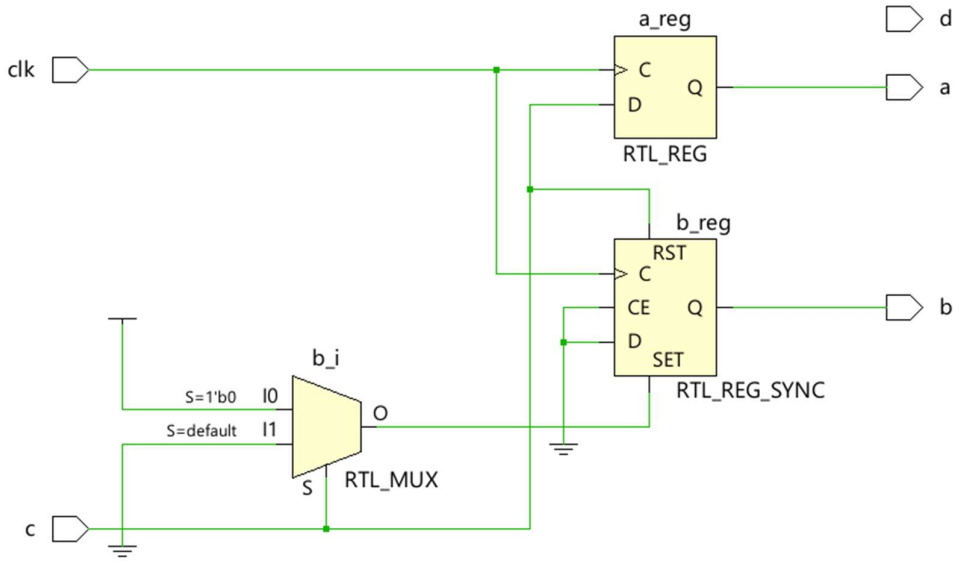
Simulation


check.v
check_tb.v
Untitled 5
12.000 ns
Name Value
1 c 0
2 clk 0
3 a 0
4 b 1
5 d X
0.000 ns
5.000 ns
10.000 ns
15.000 ns
20.000 ns
25.000 ns
30.000 ns
35.000 ns
40.000

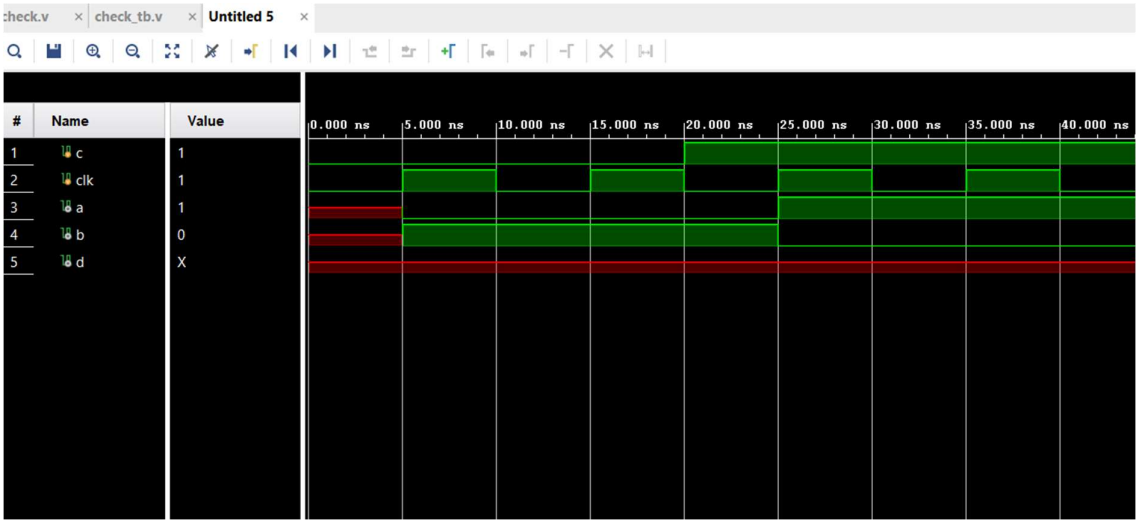
Code

```
module check( input  clk,c,output reg a,b,d   );


always@(posedge clk)
begin
a=c;
if(a==0)
b=1;
else if(a==1)
b=0;
//d=b;


end
endmodule
```

Circuit (no change if compared to previous one)



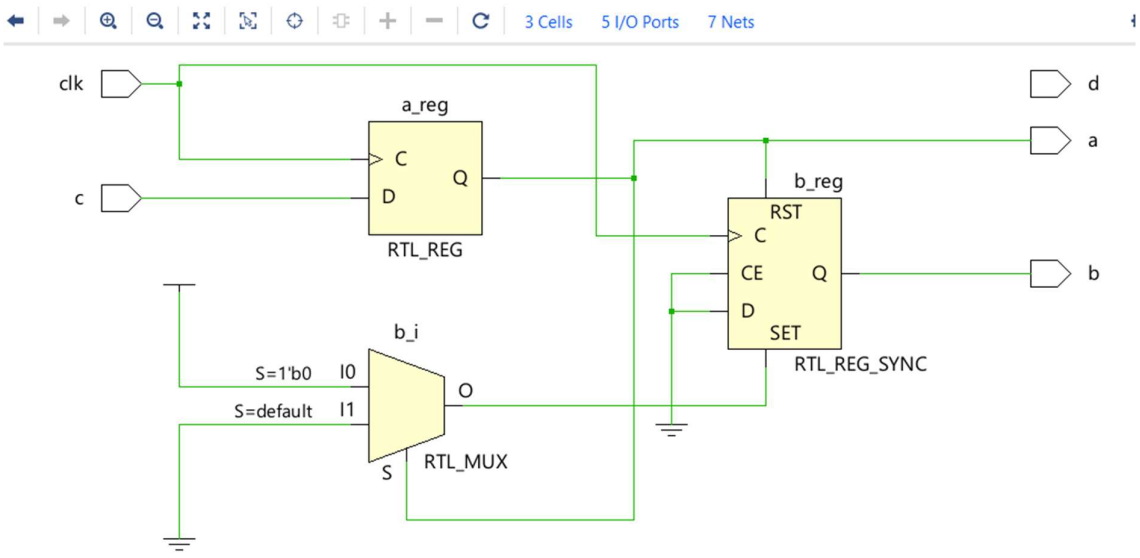
Simulation(no change if compared to previous one)



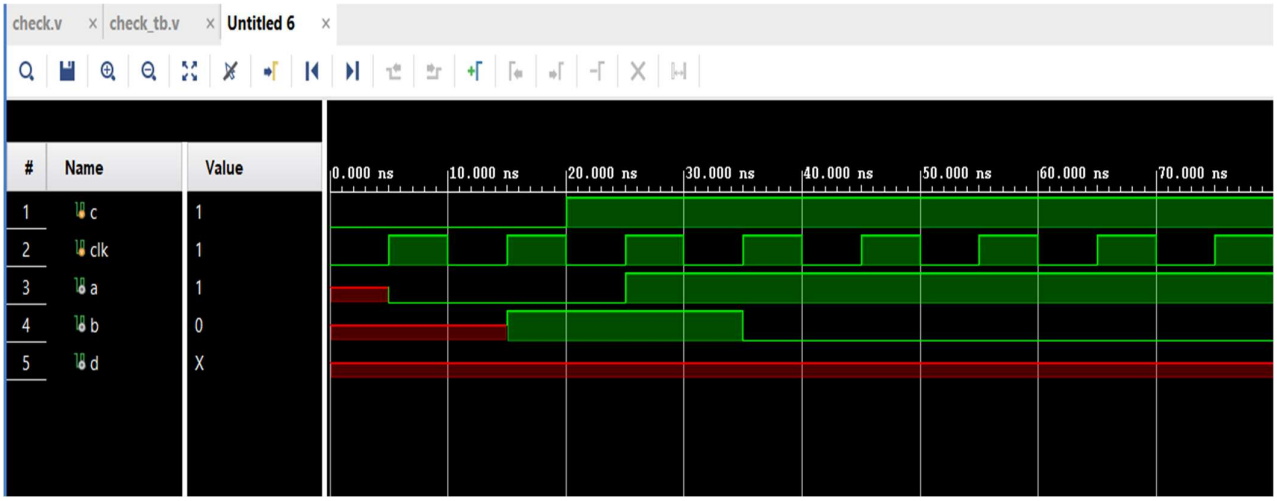
Code

```
module check( input  clk,c,output reg a,b,d   );



always@(posedge clk)
begin
a<=c;
if(a==0)
b<=1;
else if(a==1)
b<=0;
//d=b;


end
endmodule
```

Circuit



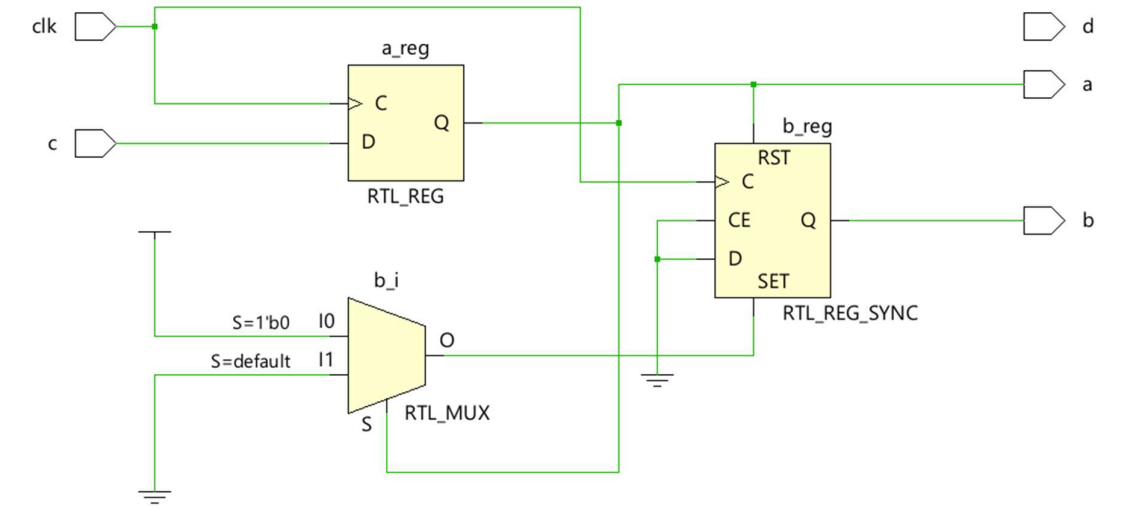
Simulation


check.v
check_tb.v
Untitled 6
Name Value
1 c 1
2 clk 1
3 a 1
4 b 0
5 d X
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns

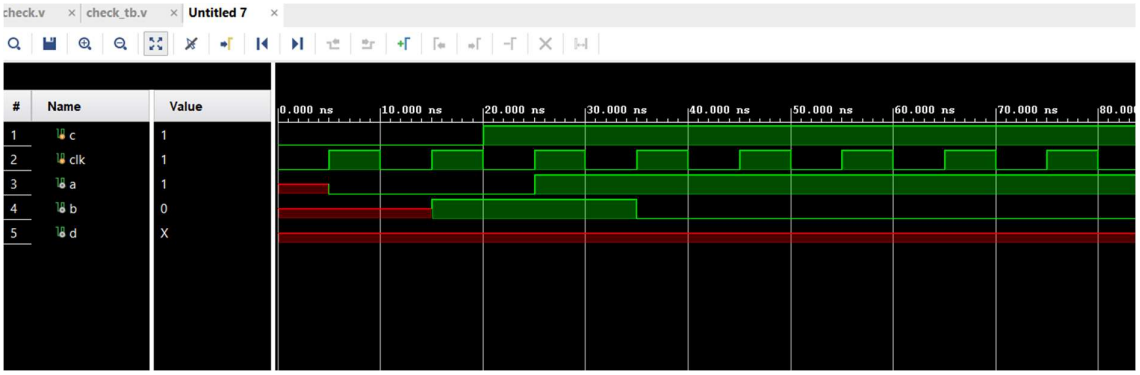
Code

```
module check( input  clk,c,output reg a,b,d   );



always@(posedge clk)
begin
a<=c;
if(a==0)
b=1;
else if(a==1)
b=0;
//d=b;


end
endmodule
```

Circuit (no change if compared to previous one)



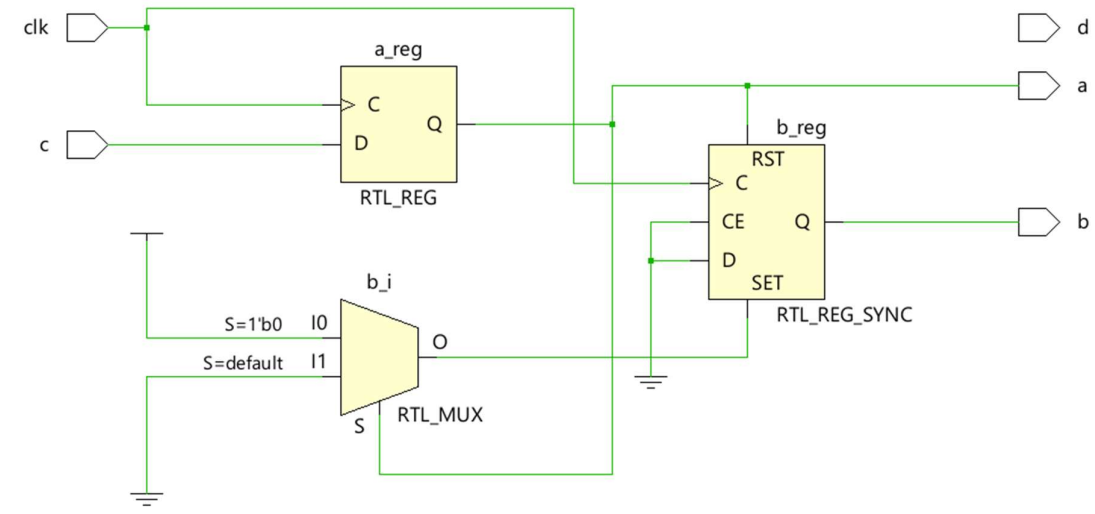
Simulation (no change if compared to previous one)



Code

```
2
3  module check( input  clk,c,output reg a,b,d   );
4
5
6
7  always@(posedge clk)
8  begin
9  a<=c;
0  if(a==0)
1  b<=1;
2  else if(a==1)
3  b<=0;
4  //d=b;
5
6
7  end
8  endmodule
9
```

Circuit

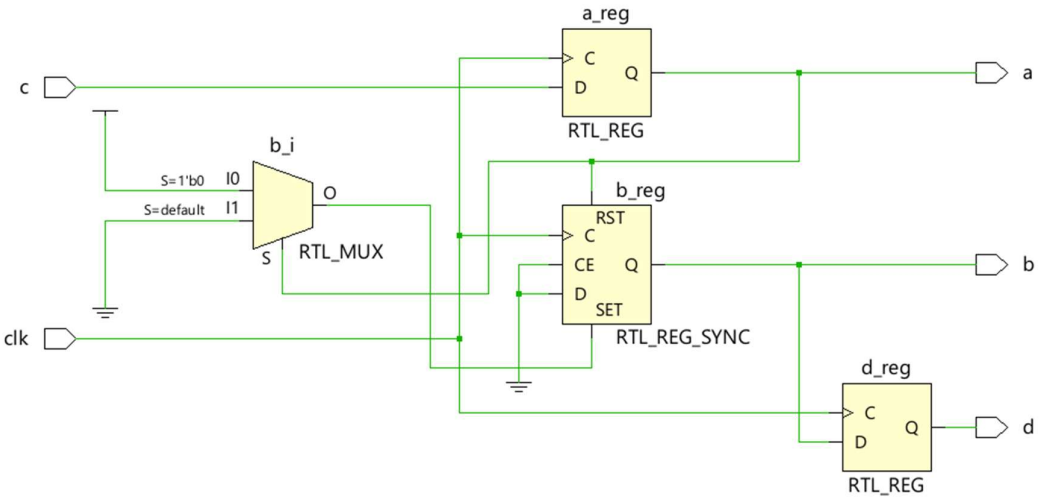


Simulation

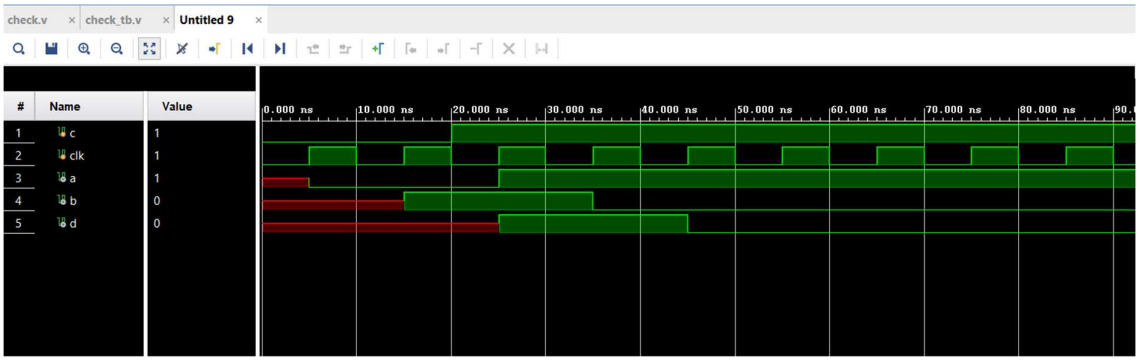
Code

```
module check( input  clk,c,output reg a,b,d   );



always@(posedge clk)
begin
a<=c;
if(a==0)
b<=1;
else if(a==1)
b<=0;
d<=b;


end
endmodule
```

Circuit



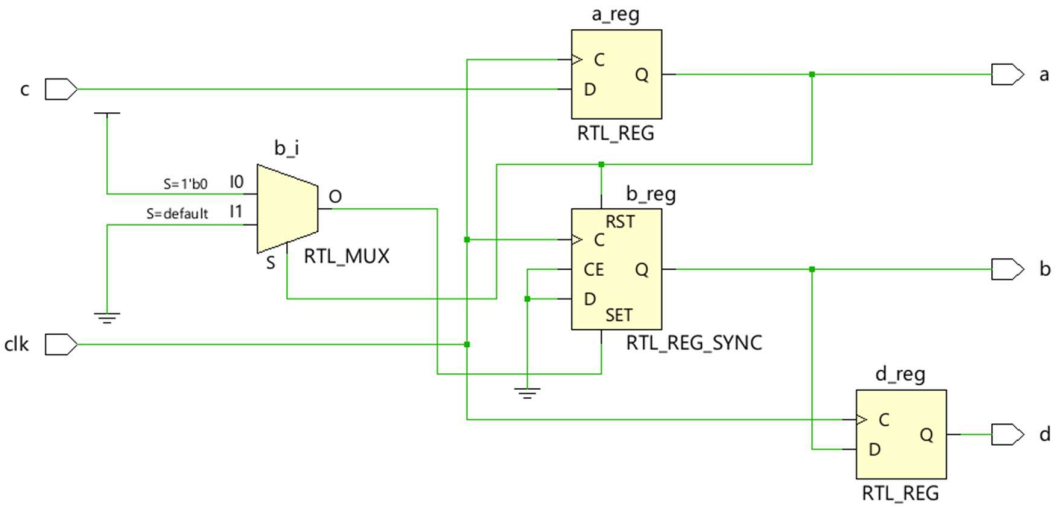
Simulation



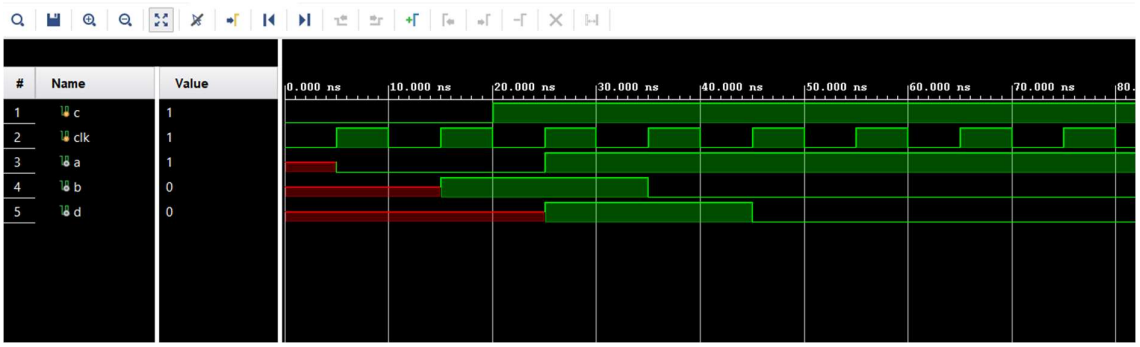
Code

```
module check( input  clk,c,output reg a,b,d   );



always@(posedge clk)
begin
a<=c;
if(a==0)
b<=1;
else if(a==1)
b<=0;
d=b;


end
endmodule
```

Circuit (no change if compared to previous one)



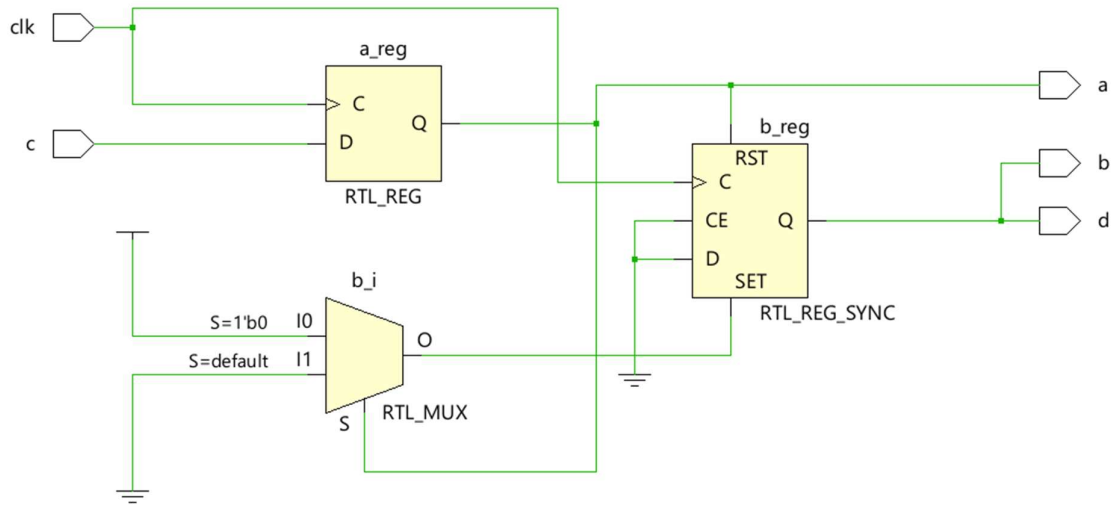
Simulation (no change if compared to previous one)



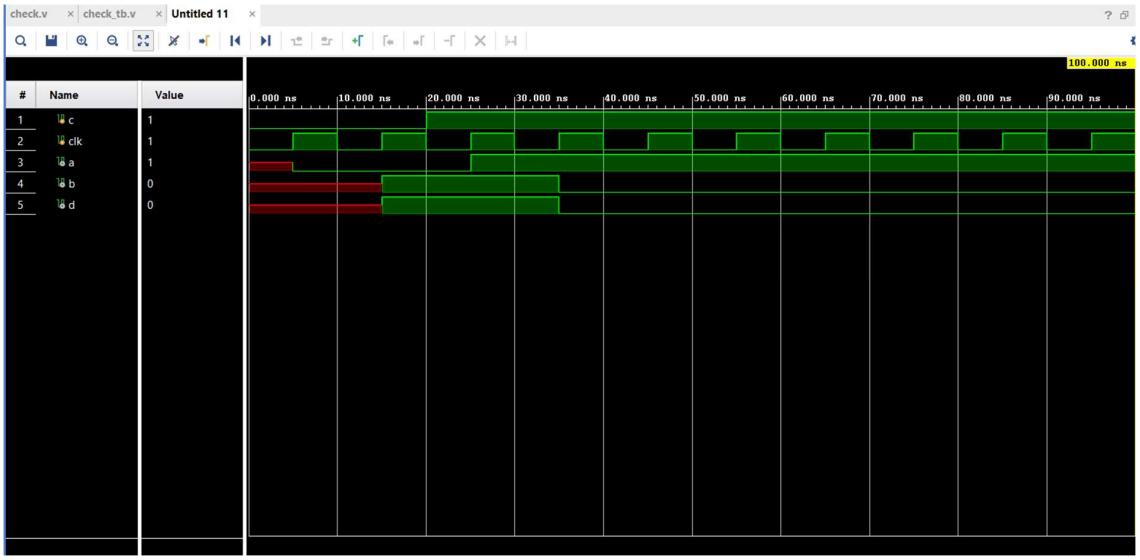
Code

```
module check( input  clk,c,output reg a,b,d   );



always@(posedge clk)
begin
a<=c;
if(a==0)
b=1;
else if(a==1)
b=0;
d=b;


end
endmodule
```

Circuit



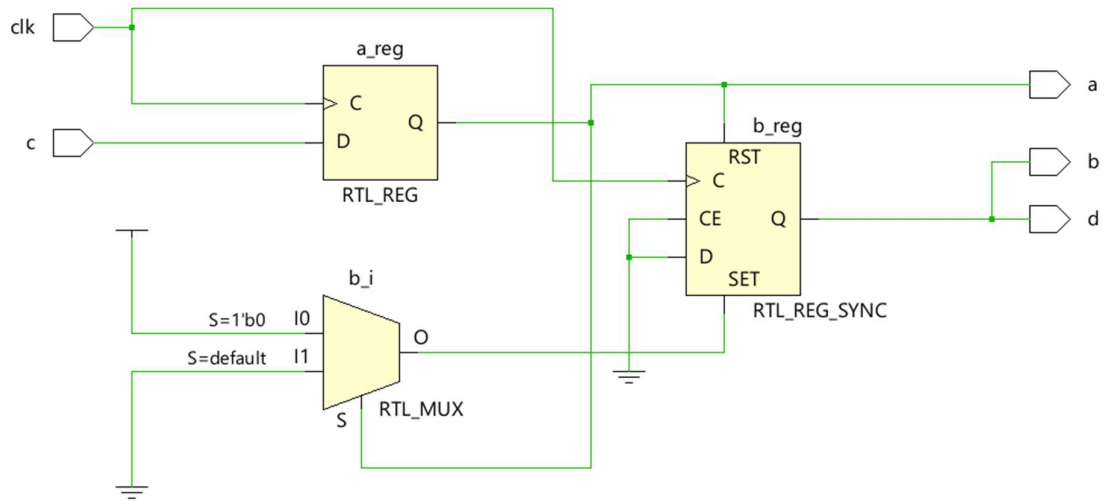
Simulation


check.v
check_tb.v
Untitled 11
100.000 ns
Name Value
1 c 1
2 clk 1
3 a 1
4 b 0
5 d 0
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns
70.000 ns
80.000 ns
90.000 ns

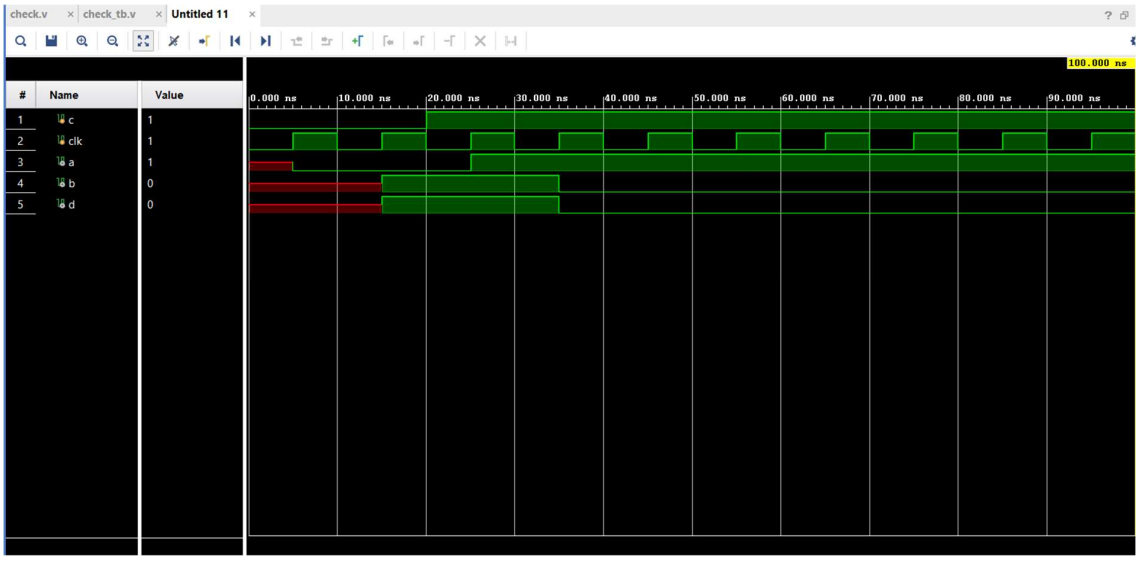
Code

```
module check( input  clk,c,output reg a,b,d   );


always@(posedge clk)
begin
a<=c;
if(a==0)
b=1;
else if(a==1)
b=0;
d<=b;


end
endmodule
```

Circuit (no change if compared to previous one)



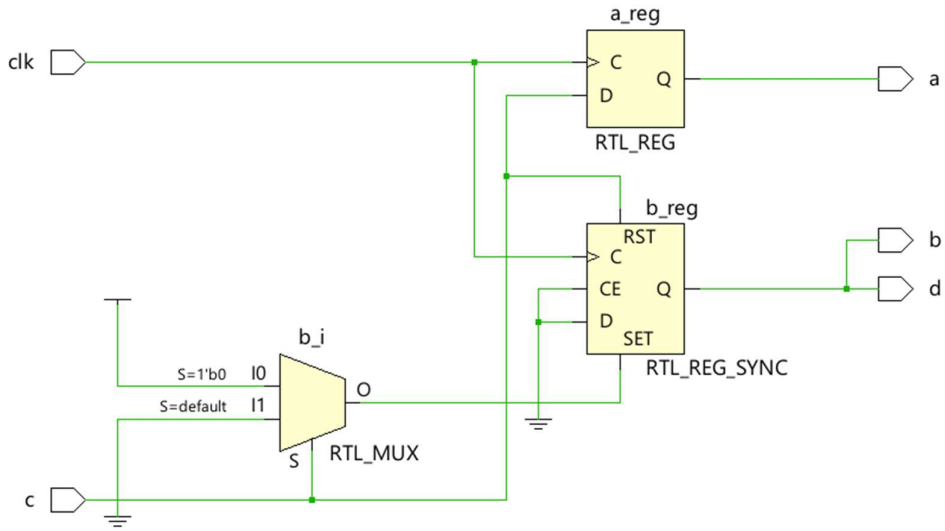
Simulation (no change if compared to previous one)



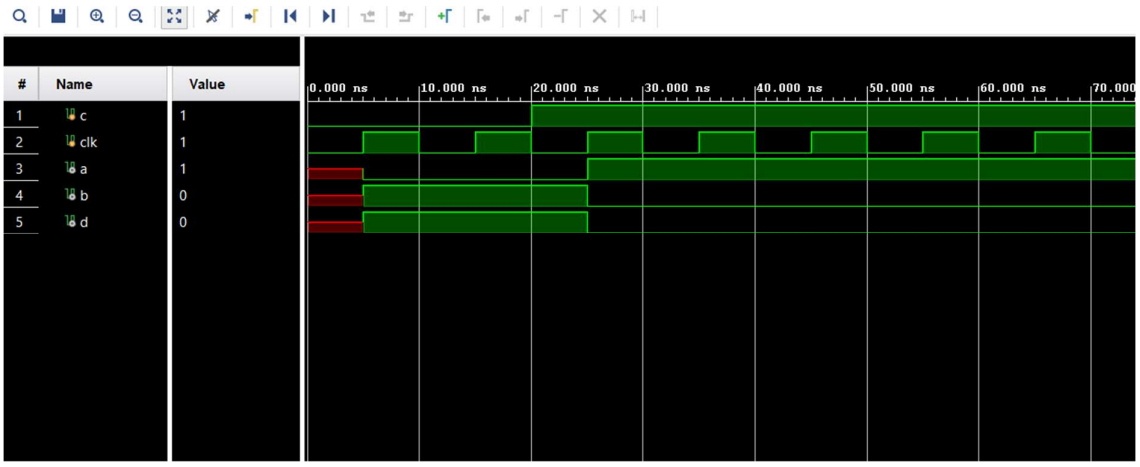
Code

```
module check( input  clk,c,output reg a,b,d   );



always@(posedge clk)
begin
a=c;
if(a==0)
b=1;
else if(a==1)
b=0;
d<=b;


end
endmodule
```

Circuit



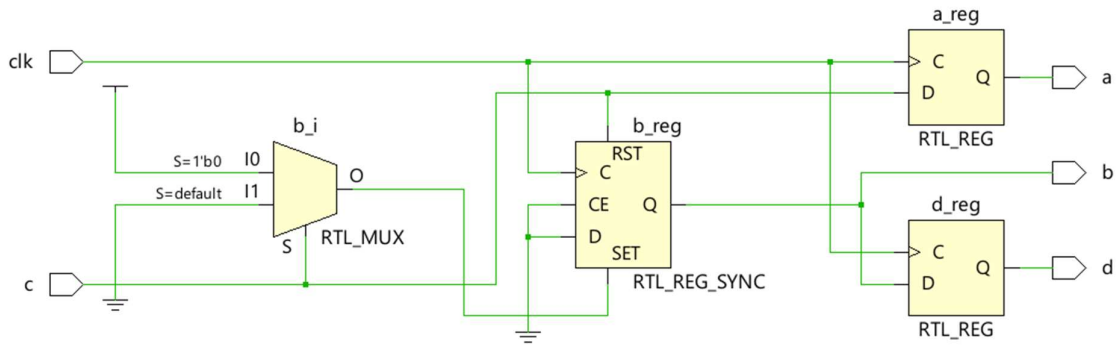
Simulation



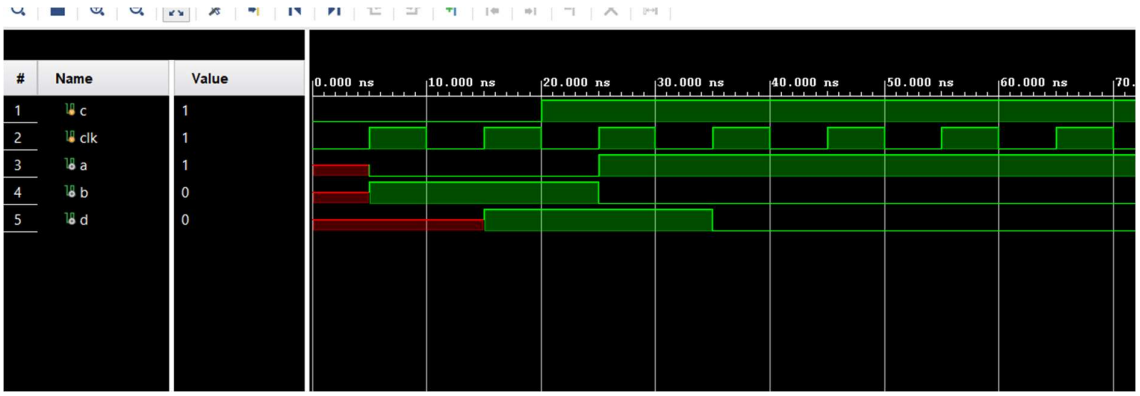
Code

```
module check( input  clk,c,output reg a,b,d   );


always@(posedge clk)
begin
  a=c;
if(a==0)
  b<=1;
else if(a==1)
b<=0;
  d<=b;


end
endmodule
```

Circuit



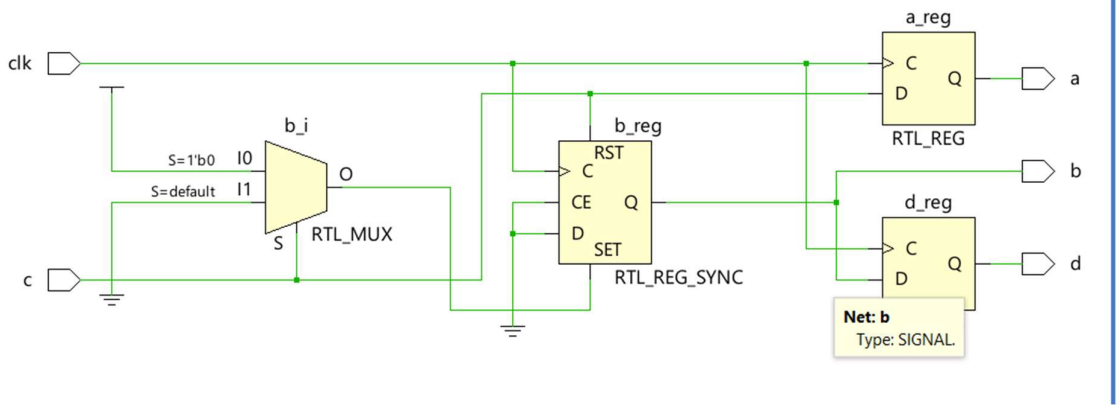
Simulation


#
Name
Value
1
c
1
2
clk
1
3
a
1
4
b
0
5
d
0
0.000 ns
10.000 ns
20.000 ns
30.000 ns
40.000 ns
50.000 ns
60.000 ns

Code

```
module check( input  clk,c,output reg a,b,d   );


always@(posedge clk)
begin
a=c;
if(a==0)
b<=1;
else if(a==1)
b<=0;
d=b;


end
endmodule
```

Circuit (no change if compared to previous one)



Simulation (no change if compared to previous one)