DLD

EXPERIMENT – 9

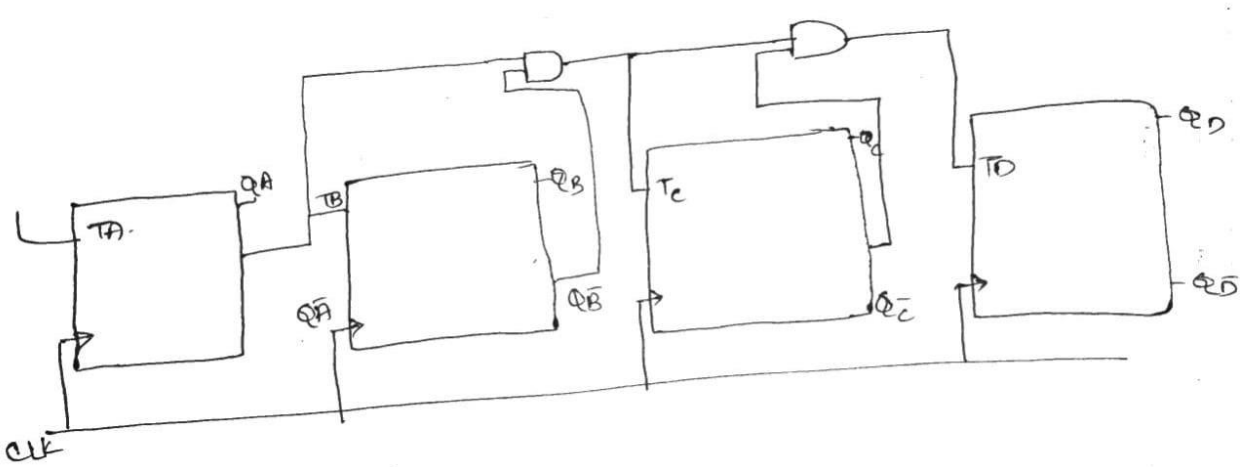
Aim: To simulate the logic gates of 4-bit Synchronous Down counter

Tools used: vivado software

Truth table:

TRUTH TABLE											
P.S				N.S				OUTPUTS			
QD	QC	QB	QA	QD+	QC+	QB+	QA+	TD	TC	TB	TA
1	1	1	1	1	1	1	0	0	0	0	1
1	1	1	0	1	1	0	1	0	0	1	1
1	1	0	1	1	1	0	0	0	0	0	1
1	1	0	0	1	0	1	1	0	1	1	1
1	0	1	1	1	0	1	0	0	0	0	1
1	0	1	0	1	0	0	1	0	0	1	1
1	0	0	1	1	0	0	0	0	0	0	1
1	0	0	0	0	1	1	1	1	1	1	1
0	1	1	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	0	0	1	1
0	1	0	1	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	1	0	1	1	1
0	0	1	1	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	0	0	1	1
0	0	0	1	0	0	0	0	0	0	0	11
0	0	0	0	1	1	1	1	1	1	1	1

Diagram:



Expression:

TD = ~QC.~QB.~QA

TC = ~QB.~QA

TB = ~QA

TA = 1

Verilog codes:

```
module fourbit_syncdown(clk,in,count);
input clk,in;
output [3:0] count;
wire QA,QB,QC,QD,QA_bar,QB_bar,QC_bar,z1,z2;
tff t1(clk,in,QA,QA_bar);
tff t2(clk,QA_bar,QB,QB_bar);
```

```
andgate a1(QA_bar,QB_bar,z);
tff t3(clk,z,QC,QC_bar);
and_gate a2(z,QC_bar,z1);
tff t4(clk,z1,QD,QD_bar);
send s1(QA_bar,QB_bar,QC_bar,QD_bar,count);
endmodule
module send(a,b,c,d,out);
input a,b,c,d;
output [3:0] out;
assign out[0] = a;
assign out[1] = b;
assign out[2] = c;
assign out[3] = d;
endmodule
module tff(clk,in,out,out_bar);
input clk,in;
output out,out_bar;
reg out, out_bar;
initial
 out = 0;
always@(posedge clk)
begin
```

```
 if(in == 1) begin
 out = ~out;
 out_bar = ~out;
 end
 else
 begin
 out = out;
 out_bar = ~out;
 end
end
endmodule
module andgate(a,b,z);
input a,b;
output z;
assign z = a&b;
endmodule
```

Test bench code:

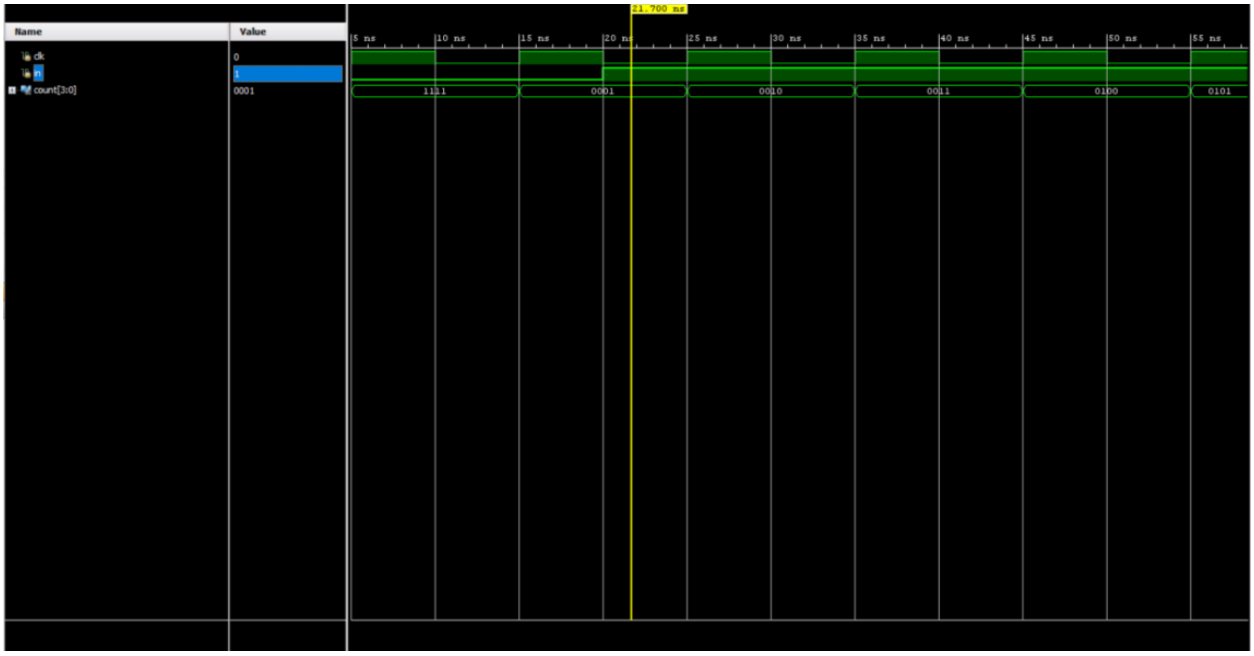
```
module tb_foutbit_syncdowncounter;
reg clk,in;
wire [3:0] count;
fourbit_syncdownc1(clk,in,count);
```

```
initial
begin
clk = 1'b0; in = 1'b0;
#20 in = 1'b1;
end
always
begin
#5 clk = ~clk;
end
endmodule
```

output:


11.900 ns
Name
Value
clk
0
in
0
count[3:0]
1111
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
1111
0001
0010
0011
0100
0101

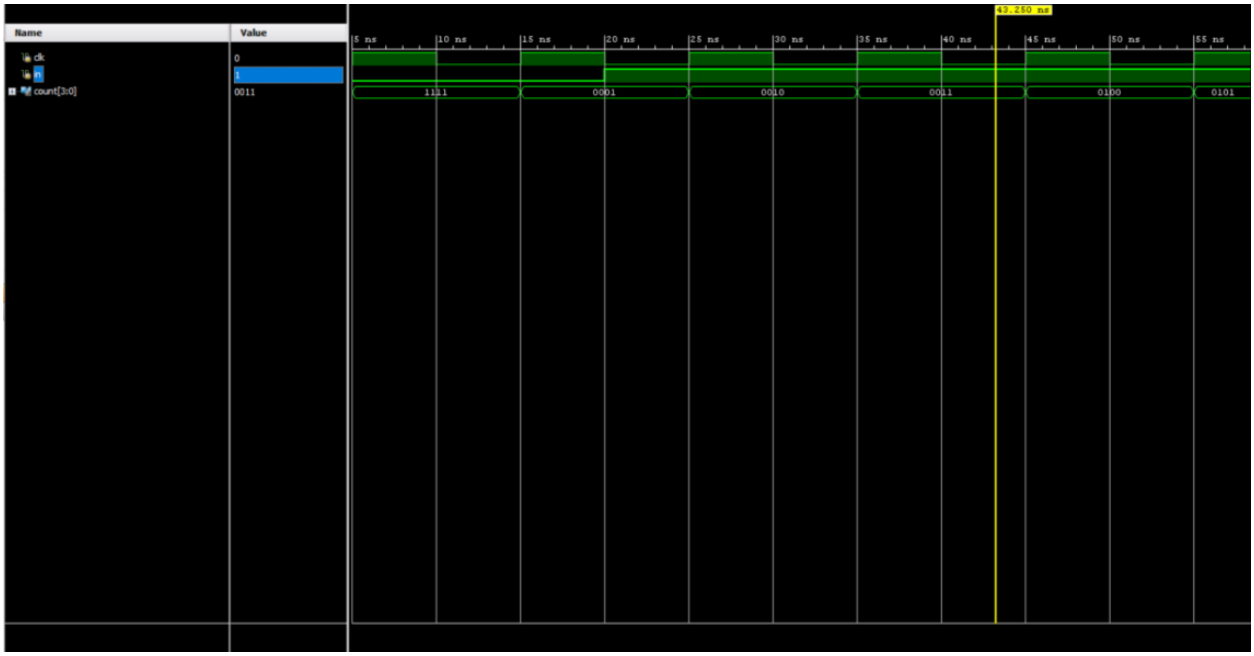

26.800 ns
Name
Value
clk
1
in
1
count[3:0]
0010
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
1111
0001
0010
0011
0100
0101


Name
Value
clk
in
count[3:0]
0
1
0001
11.700 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
1111
0001
0010
0011
0100
0101

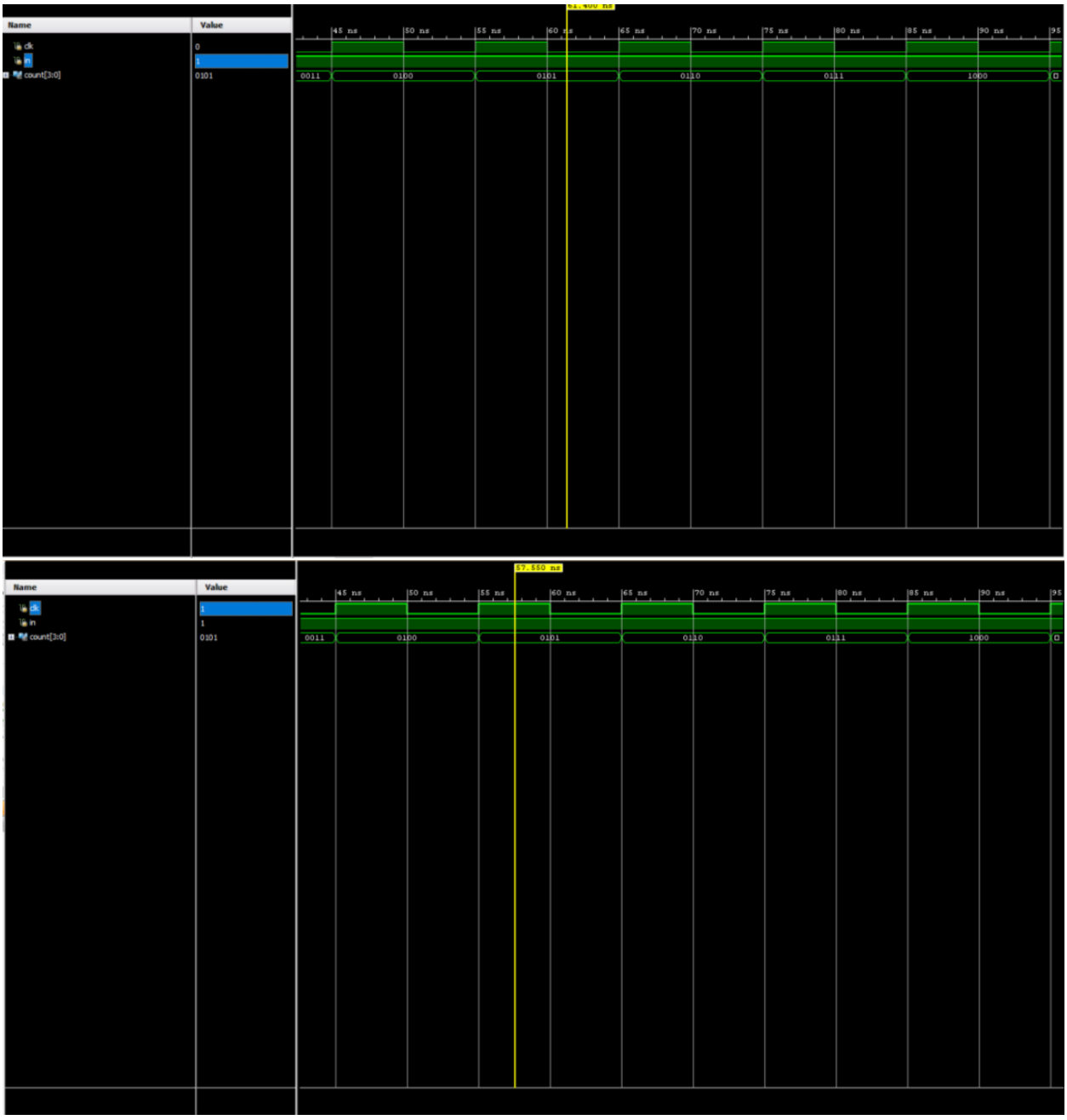

Name
Value
clk
in
count[3:0]
1
0
1111
7.050 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
1111
0001
0010
0011
0100
0101

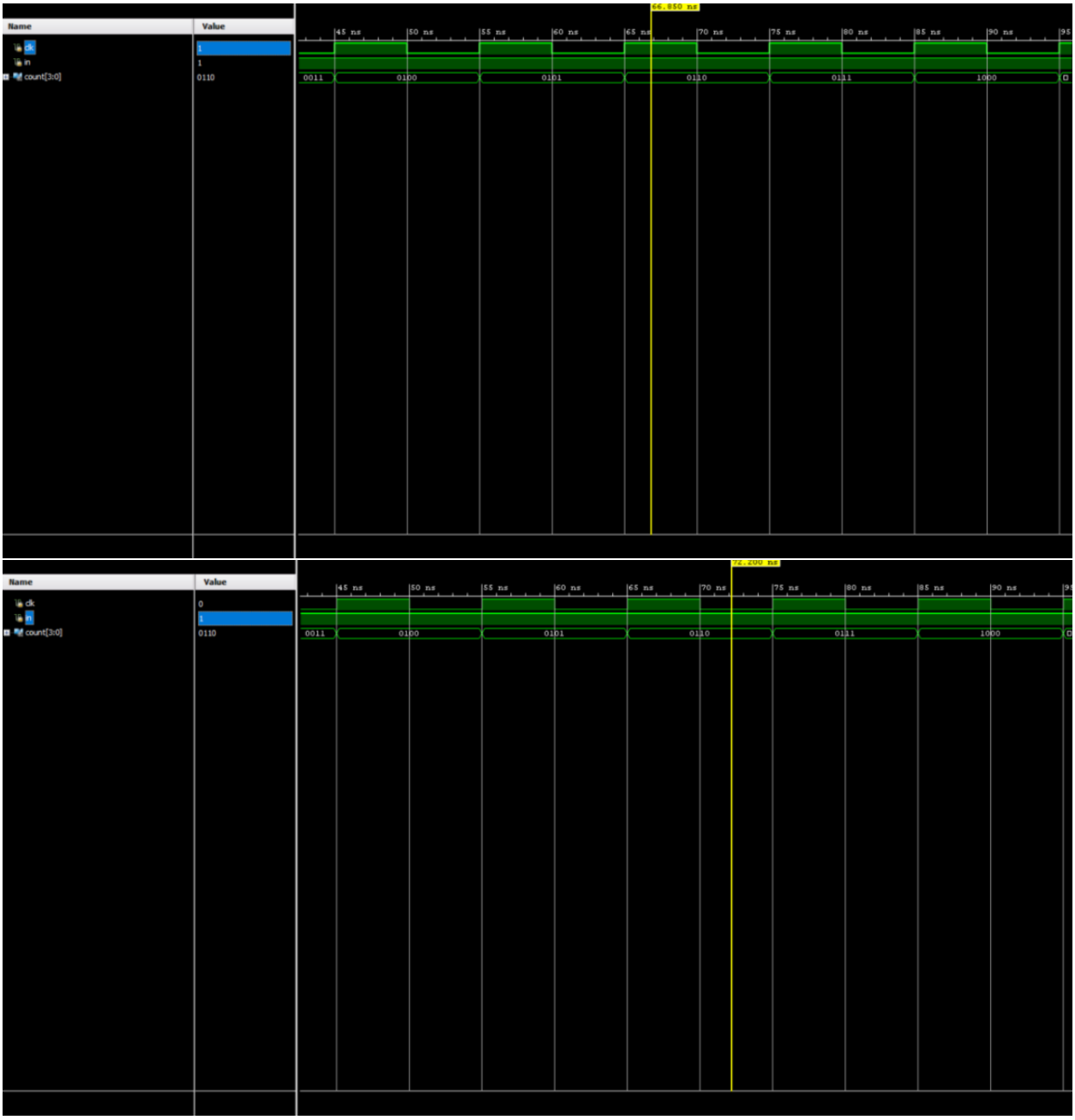

Name
Value
clk
in
count[3:0]
1
1
0100
47.150 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
1111
0001
0010
0011
0100
0101


Name
Value
clk
in
count[3:0]
0
1
0010
32.150 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
1111
0001
0010
0011
0100
0101


Name
Value
clk
rst
count[3:0]
0
1
0011
43.250 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
1111
0001
0010
0011
0100
0101


Name
Value
clk
rst
count[3:0]
1
0
0001
17.400 ns
5 ns
10 ns
15 ns
20 ns
25 ns
30 ns
35 ns
40 ns
45 ns
50 ns
55 ns
1111
0001
0010
0011
0100
0101


Name
Value
clk
in
count[3:0]
0
1
0101
45 ns
50 ns
55 ns
60 ns
65 ns
70 ns
75 ns
80 ns
85 ns
90 ns
0011
0100
0101
0110
0111
1000
Name
Value
clk
in
count[3:0]
1
1
0101
57.550 ns
45 ns
50 ns
55 ns
60 ns
65 ns
70 ns
75 ns
80 ns
85 ns
90 ns
0011
0100
0101
0110
0111
1000



Conclusion:

Such that we can find the results of the codes with this software is very easy and fast and I came to know that sync 4 bit down counter Verilog code