Lab 2

Milen Dimitrov

(Lecture 5/6)

FPGA Verilog

1. Install ModelSim

2. Design a 2 bit adder

3. Simulate a 2-bit adder in ModelSim.

4. Submit a single PDF file that contains the Verilog code and a screenshot of the output.

Designing and implementing 4bit adder by different modeling strategies:

1. Gate Level Modeling

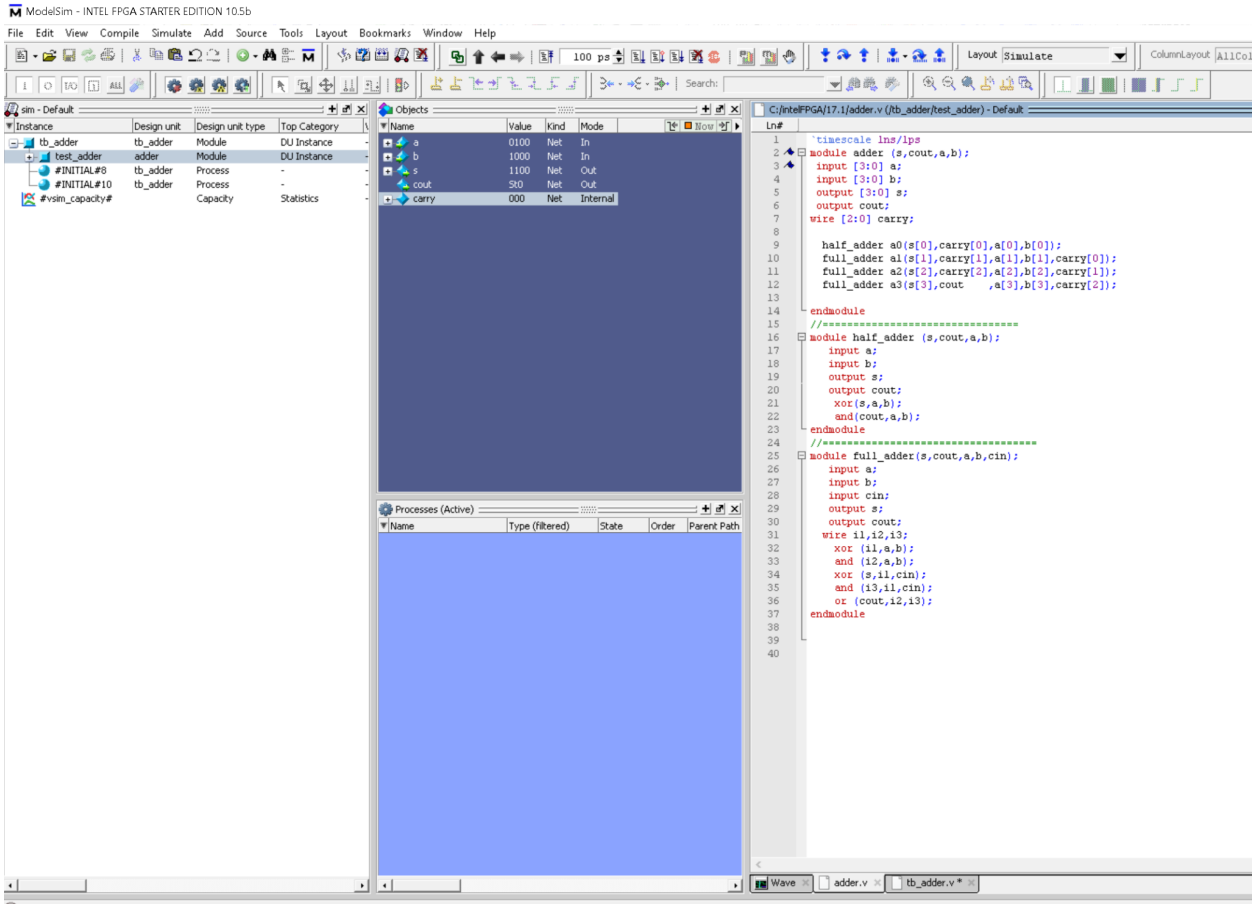
adder.v

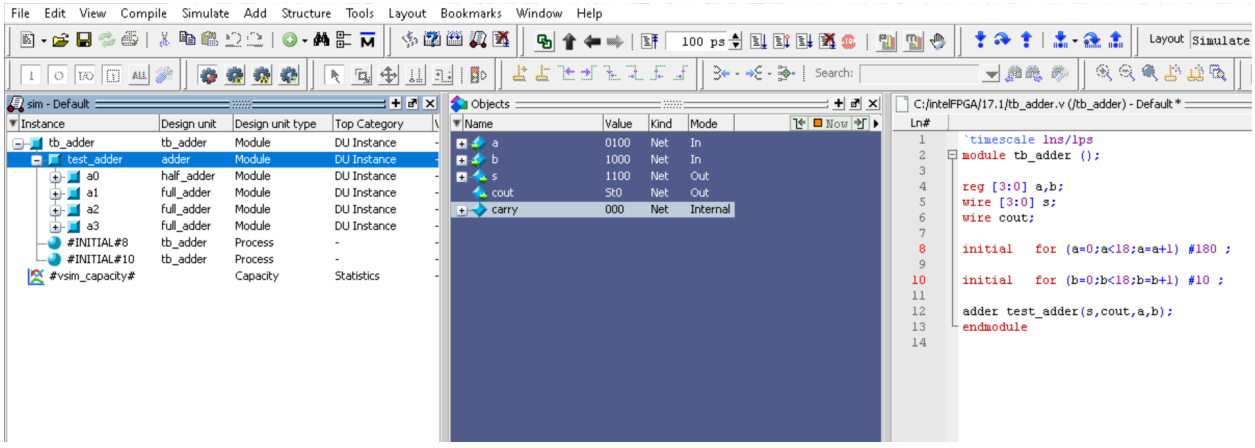
```
`timescale 1ns/1ps
module adder (s,cout,a,b);
 input [3:0] a;
 input [3:0] b;
 output [3:0] s;
 output cout;
wire [2:0] carry;

  half_adder a0(s[0],carry[0],a[0],b[0]);
  full_adder a1(s[1],carry[1],a[1],b[1],carry[0]);
  full_adder a2(s[2],carry[2],a[2],b[2],carry[1]);
  full_adder a3(s[3],cout    ,a[3],b[3],carry[2]);

endmodule
//================================
module half_adder (s,cout,a,b);
   input a;
   input b;
   output s;
   output cout;
    xor(s,a,b);
    and(cout,a,b);
endmodule
//================================
module full_adder(s,cout,a,b,cin);
```

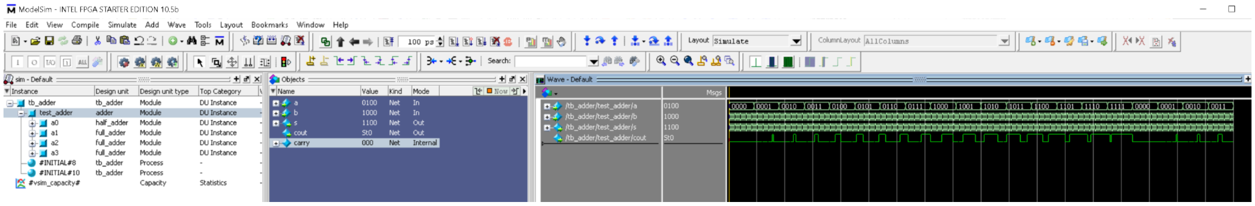
```
   input a;
   input b;
   input cin;
   output s;
   output cout;
  wire i1,i2,i3;
    xor (i1,a,b);
    and (i2,a,b);
    xor (s,i1,cin);
    and (i3,i1,cin);
    or (cout,i2,i3);
endmodule
```

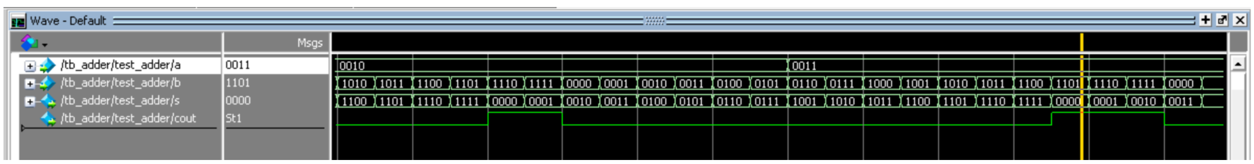
tb_adder.v

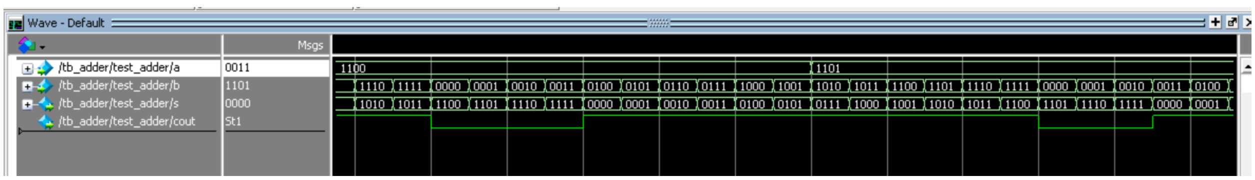
```
`timescale 1ns/1ps
module tb_adder ();

reg [3:0] a,b;
wire [3:0] s;
wire cout;

initial   for (a=0;a<18;a=a+1) #180 ;

initial   for (b=0;b<18;b=b+1) #10 ;

adder test_adder(s,cout,a,b);
endmodule
```

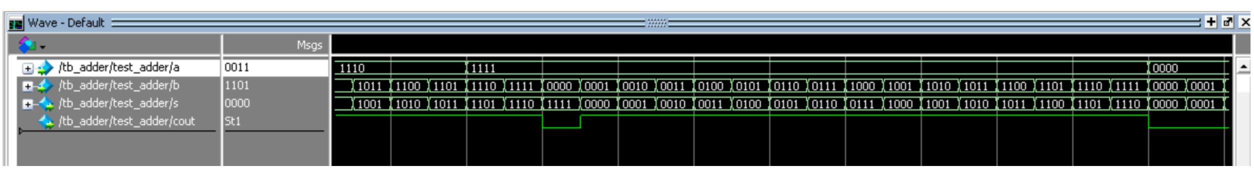

ModelSim - INTEL FPGA STARTER EDITION 10.5b
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type Top Category
tb_adder tb_adder Module DU Instance
test_adder adder Module DU Instance
#INITIAL#8 tb_adder Process -
#INITIAL#10 tb_adder Process -
#vsim_capacity# Capacity Statistics
Objects
Name Value Kind Mode
a 0100 Net In
b 1000 Net In
s 1100 Net Out
cout St0 Net Out
carry 000 Net Internal
Processes (Active)
Name Type (filtered) State Order Parent Path
C:/intelFPGA/17.1/adder.v (/tb_adder/test_adder) - Default
`timescale lns/lps
module adder (s,cout,a,b);
input [3:0] a;
input [3:0] b;
output [3:0] s;
output cout;
wire [2:0] carry;
half_adder a0(s[0],carry[0],a[0],b[0]);
full_adder a1(s[1],carry[1],a[1],b[1],carry[0]);
full_adder a2(s[2],carry[2],a[2],b[2],carry[1]);
full_adder a3(s[3],cout ,a[3],b[3],carry[2]);
endmodule
//==================================
module half_adder (s,cout,a,b);
input a;
input b;
output s;
output cout;
xor(s,a,b);
and(cout,a,b);
endmodule
//==================================
module full_adder(s,cout,a,b,cin);
input a;
input b;
input cin;
output s;
output cout;
wire i1,i2,i3;
xor (i1,a,b);
and (i2,a,b);
xor (s,i1,cin);
and (i3,i1,cin);
or (cout,i2,i3);
endmodule
Wave
adder.v
tb_adder.v *
Transcript
End time: 23:35:28 on May 10,2022, Elapsed time: 0:02:20
Errors: 0, Warnings: 0
vsim work.tb_adder
Start time: 23:35:28 on May 10,2022
Loading work.tb_adder
Loading work.adder
Loading work.half_adder
Loading work.full_adder
add wave -position insertpoint sim:/tb_adder/test_adder/*
VSIM 93> run 3600000
VSIM 94>


File Edit View Compile Simulate Add Structure Tools Layout Bookmarks Window Help
sim - Default
Instance Design unit Design unit type Top Category
tb_adder tb_adder Module DU Instance
test_adder adder Module DU Instance
a0 half_adder Module DU Instance
a1 full_adder Module DU Instance
a2 full_adder Module DU Instance
a3 full_adder Module DU Instance
#INITIAL#8 tb_adder Process -
#INITIAL#10 tb_adder Process -
#vsim_capacity# Capacity Statistics
Objects
Name Value Kind Mode
a 0100 Net In
b 1000 Net In
s 1100 Net Out
cout St0 Net Out
carry 000 Net Internal
C:/intelFPGA/17.1/tb_adder.v (/tb_adder) - Default *
`timescale lns/lps
module tb_adder ();
reg [3:0] a,b;
wire [3:0] s;
wire cout;
initial for (a=0;a<18;a=a+1) #180 ;
initial for (b=0;b<18;b=b+1) #10 ;
adder test_adder(s,cout,a,b);
endmodule







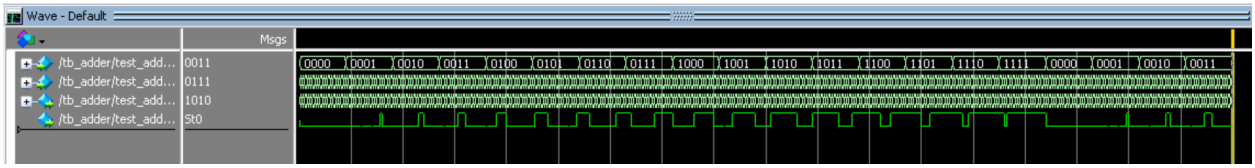


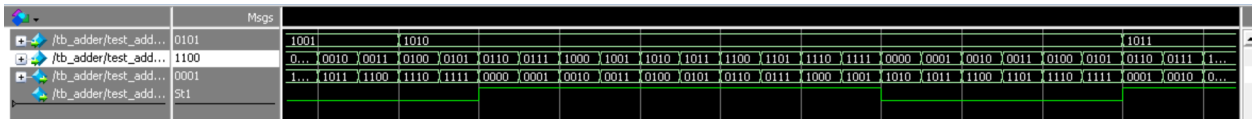
2. Data Flow Modeling

adder_data.v

```
`timescale 1ns/1ps
module adder_data (s,cout,a,b);
 input [3:0] a;
 input [3:0] b;
 output [3:0] s;
 output cout;
wire [2:0] carry;

  half_adder_data a0(s[0],carry[0],a[0],b[0]);
  full_adder_data a1(s[1],carry[1],a[1],b[1],carry[0]);
  full_adder_data a2(s[2],carry[2],a[2],b[2],carry[1]);
  full_adder_data a3(s[3],cout    ,a[3],b[3],carry[2]);

endmodule
```

```
//================================
module half_adder_data (s,cout,a,b);
   input a;
   input b;
   output s;
   output cout;
    assign s=a^b;
    assign cout=a&b;
endmodule
//================================
module full_adder_data(s,cout,a,b,cin);
   input a;
   input b;
   input cin;
   output s;
   output cout;
   wire i1,i2,i3;
     assign s=i1^cin;
     assign i2=a&b;
     assign i1=a^b;
     assign i3=cin&i1;
     assign cout=i2|i3;
endmodule
```



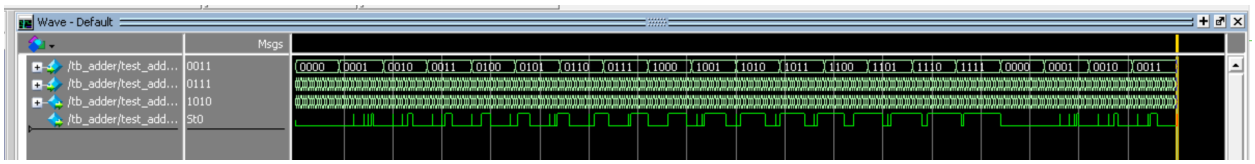


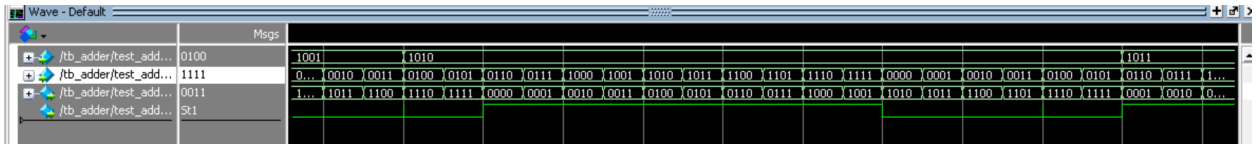
3. Behavioral Modeling

adder_beh.v

```
`timescale 1ns/1ps
module adder_beh (s,cout,a,b);
input [3:0] a;
 input [3:0] b;
 output [3:0] s;
 output cout;
wire [2:0] carry;

  half_adder_beh a0(s[0],carry[0],a[0],b[0]);
  full_adder_beh a1(s[1],carry[1],a[1],b[1],carry[0]);
  full_adder_beh a2(s[2],carry[2],a[2],b[2],carry[1]);
  full_adder_beh a3(s[3],cout    ,a[3],b[3],carry[2]);
```

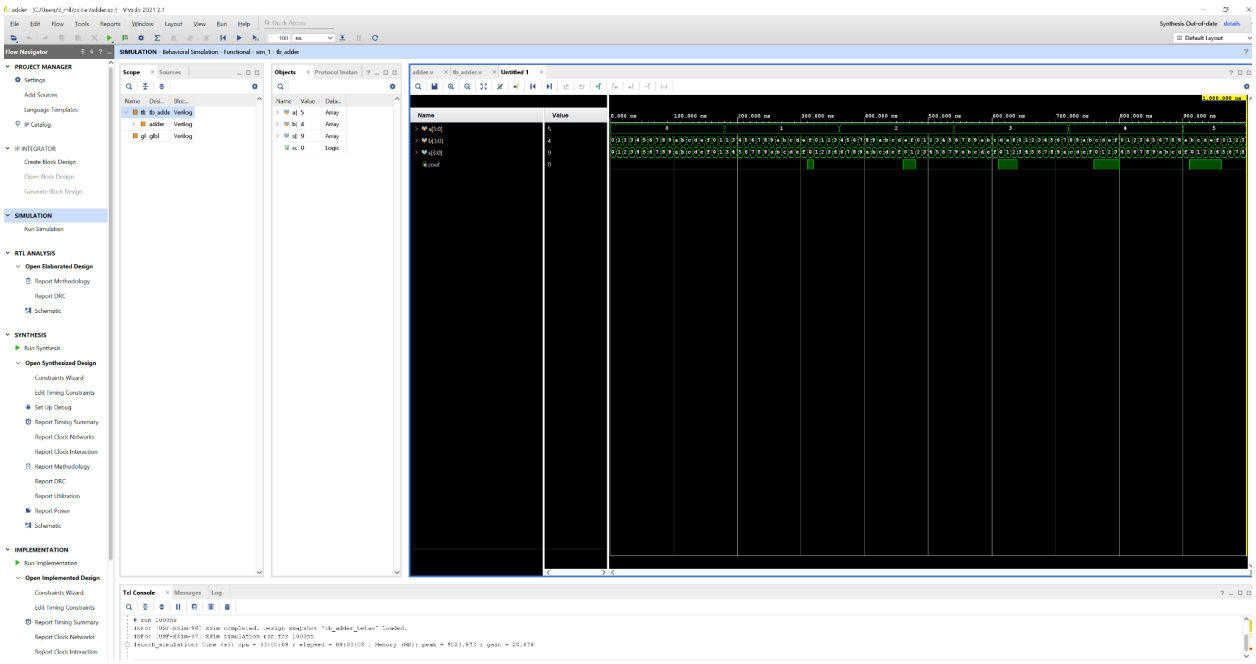
```
endmodule
//================================
module half_adder_beh (s,cout,a,b);
   input a;
   input b;
   output s;
   output cout;
reg s,cout;
always@(a,b)
   begin
      if (a==b)
         begin
         s=0;
         cout=b;
         end
      else
         begin
         s=1;
         cout=0;
         end
    end
endmodule
//==================================
module full_adder_beh (s,cout,a,b,cin);
   input a;
   input b;
   input cin;
   output s;
   output cout;
 reg s,cout;
 always@(a,b,cin)
    begin
      if (a==b)
         begin
         s=cin;
         cout=b;
         end
       else
         begin
         s=~cin;
         cout=cin;
         end
     end
endmodule
```

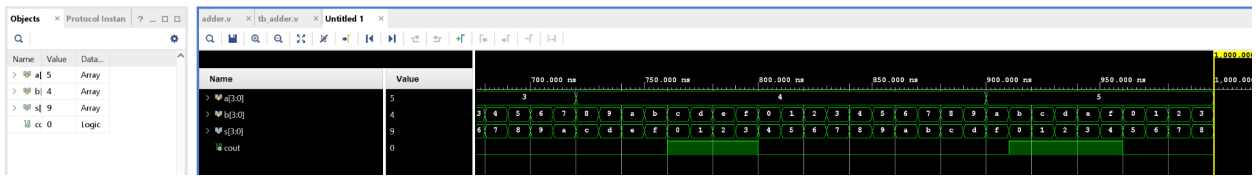




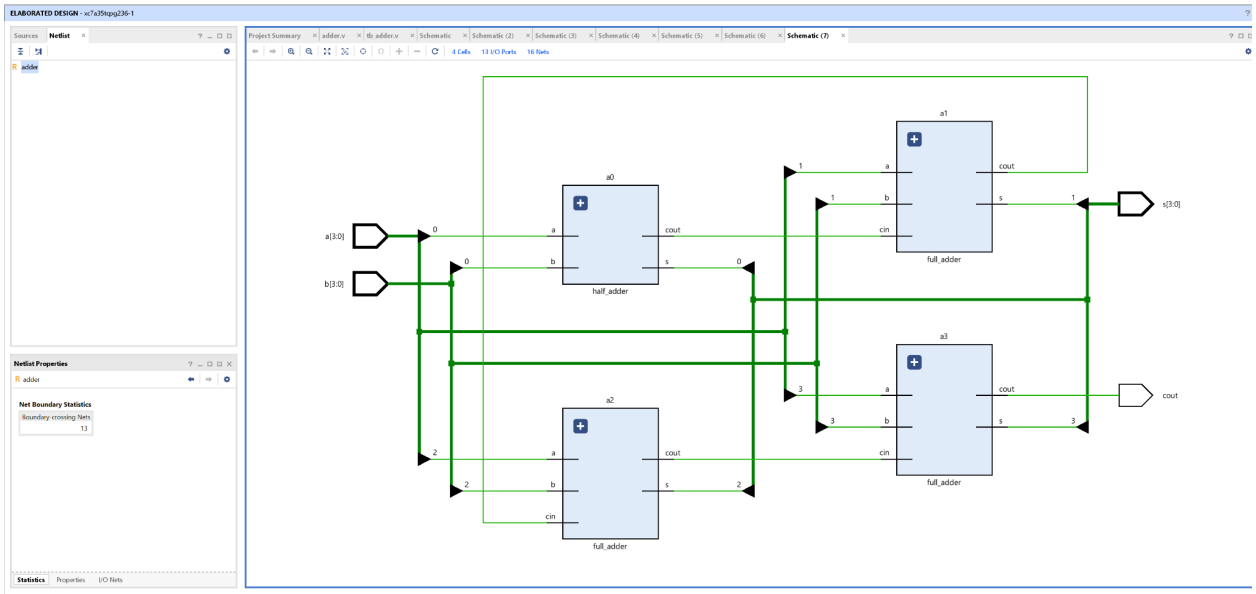
4. Running it in Vivado:

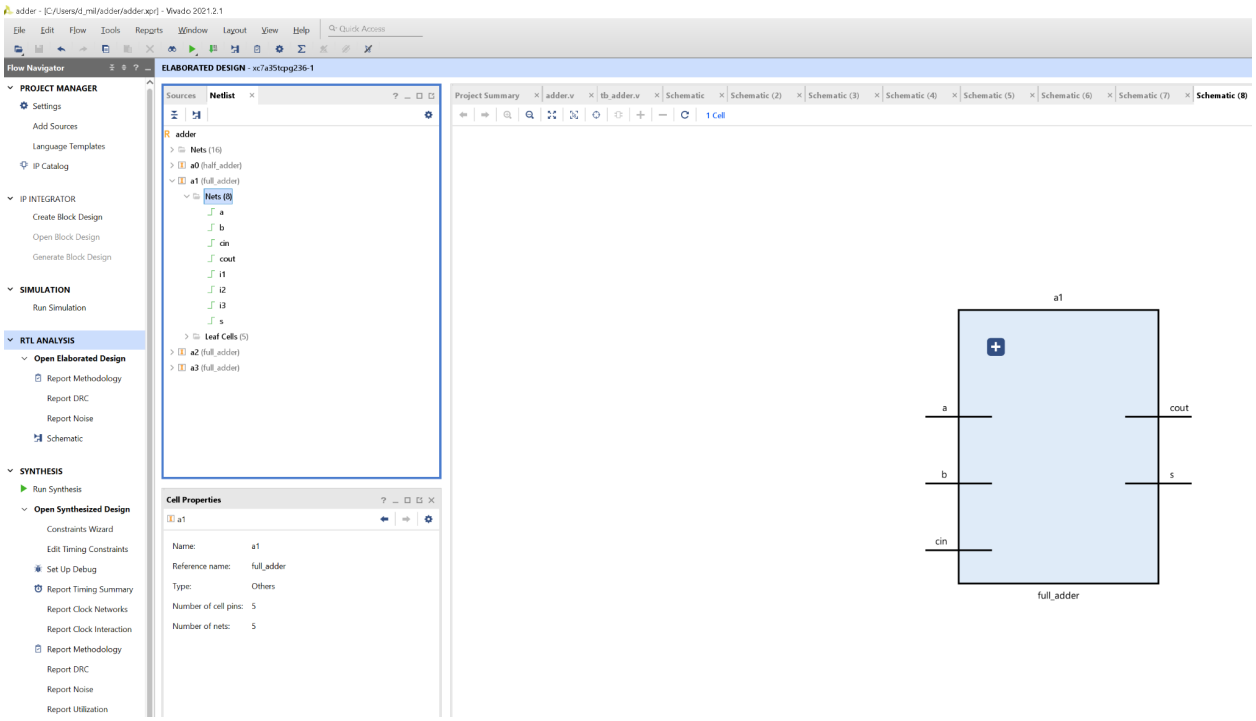
Same waveform in Vivado simulator



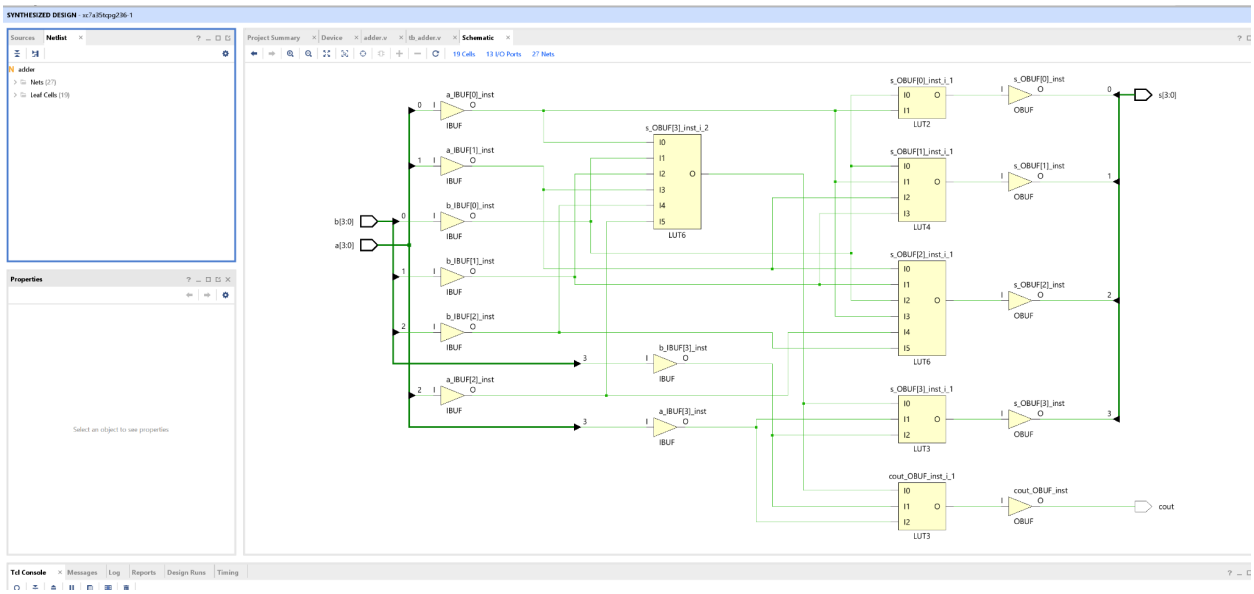


5. Schematics in Vivado :





6. Synthesized design in Vivado:



7. Generated bitstream and programed the FPGA

Constrains File:

```
set_property PACKAGE_PIN V17 [get_ports {a[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {a[0]}]
set_property PACKAGE_PIN V16 [get_ports {a[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {a[1]}]
set_property PACKAGE_PIN W16 [get_ports {a[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {a[2]}]
set_property PACKAGE_PIN W17 [get_ports {a[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {a[3]}]
set_property PACKAGE_PIN W15 [get_ports {b[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {b[0]}]
set_property PACKAGE_PIN V15 [get_ports {b[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {b[1]}]
set_property PACKAGE_PIN W14 [get_ports {b[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {b[2]}]
set_property PACKAGE_PIN W13 [get_ports {b[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {b[3]}]
set_property PACKAGE_PIN V2 [get_ports {sw[8]}]


## LEDs
set_property PACKAGE_PIN U16 [get_ports {s[0]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {s[0]}]
set_property PACKAGE_PIN E19 [get_ports {s[1]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {s[1]}]
```

```
set_property PACKAGE_PIN U19 [get_ports {s[2]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {s[2]}]
set_property PACKAGE_PIN V19 [get_ports {s[3]}]
        set_property IOSTANDARD LVCMOS33 [get_ports {s[3]}]
set_property PACKAGE_PIN W18 [get_ports {cout}]
        set_property IOSTANDARD LVCMOS33 [get_ports {cout}]
```