ASIC DESIGN

HomeWork1

By- Usman Khan

NetId: uxk220003

Professor: Alice Wang

Semester: Spring 2024

<u>Introduction:</u>

For this HW, I have made use of Vivado 2023 software for Design and Simulation.

The HomeWork1 folder has two folders in it.

- Part 1: This folder contains both Behavioral and structural code along with the vivado file for you to run and check.
- Part 2: This folder contains the FSM and the add_sub codes. For this part, I have divided the code into two different modules in two different files.

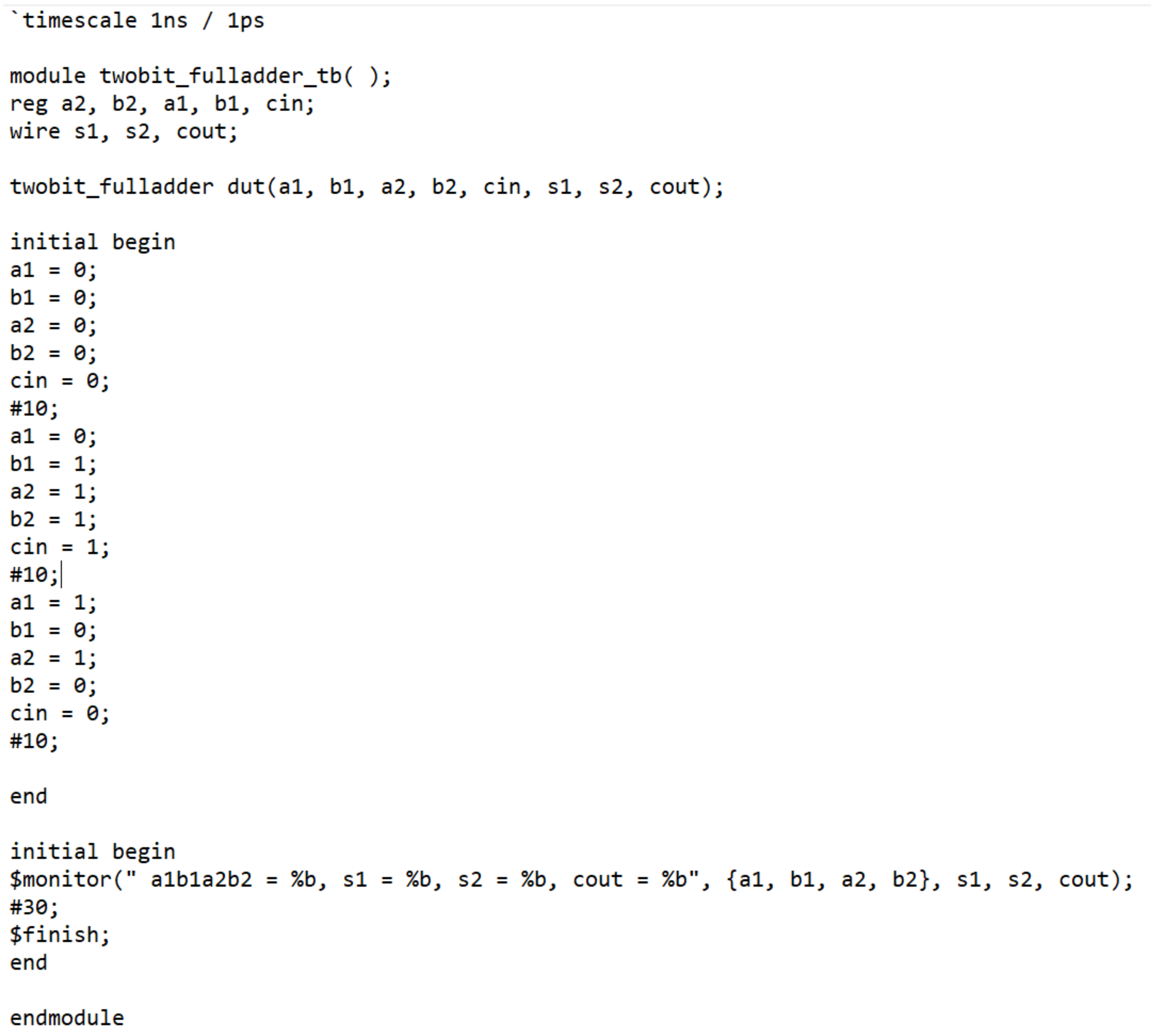
  The FSM module, on the posedge of the clock, goes from state0 to state1 (where it makes select = 0) and from state1 to state2 (where it makes select = 1)

2-Bit FULL ADDER:

Behavioral Code:

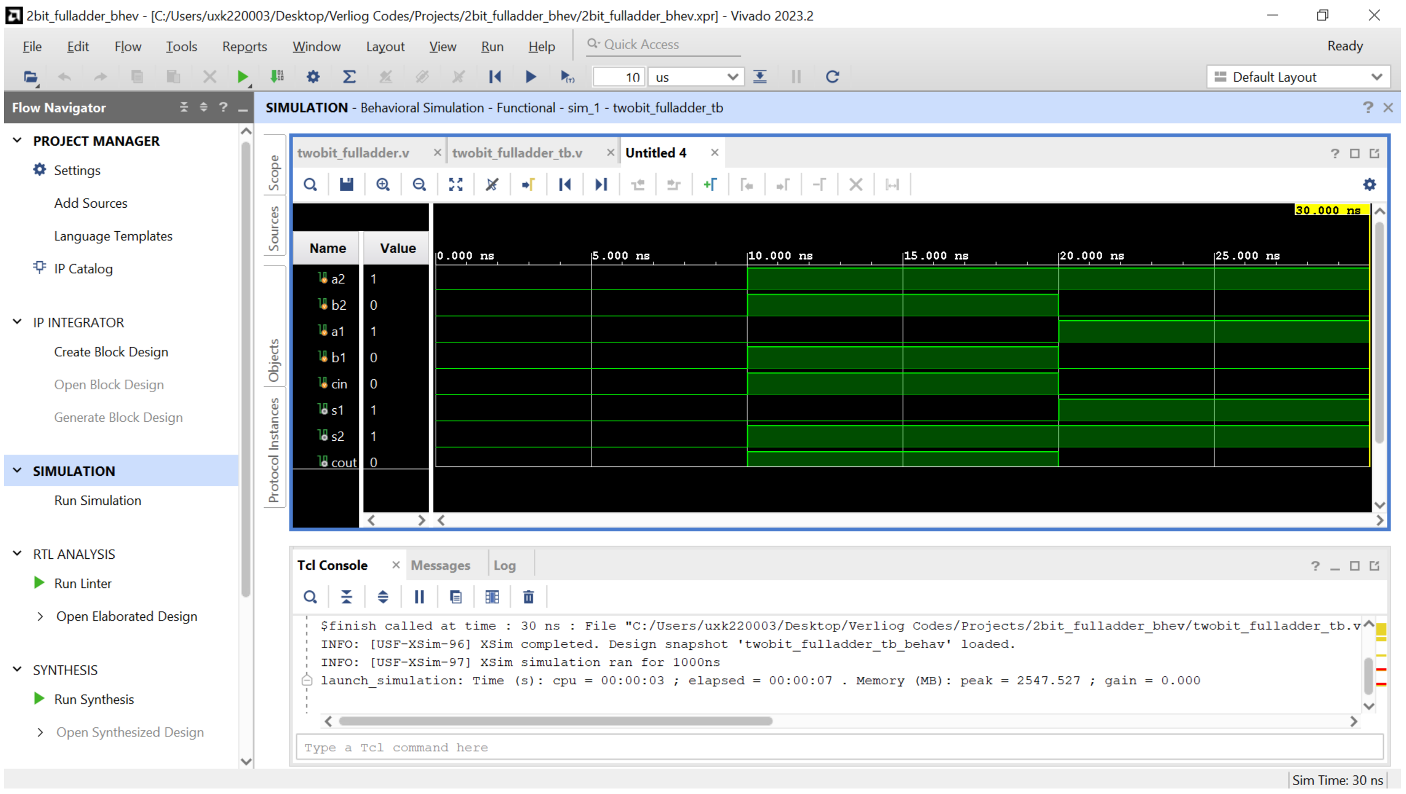
```
module full_adder(a,b,cin,sum,carry);
input a,b,cin;
output sum,carry;
wire w1,w2,w3;

assign w1 = a ^ b;
assign sum = w1 ^ cin;
assign w2 = cin & w1;
assign w3 = a & b;
assign carry = w2 | w3;

endmodule

module twobit_fulladder(a1, b1, a2, b2, cin, s1, s2, cout);
input a2, b2, a1, b1, cin;
output s1, s2, cout;

wire c1;

full_adder fa1(a1 ,b1, cin, s1, c1);
full_adder fa2(a2 ,b2, c1, s2, cout);

endmodule
```

TestBench for Behavioral Code:



```
`timescale 1ns / 1ps

module twobit_fulladder_tb( );
reg a2, b2, a1, b1, cin;
wire s1, s2, cout;

twobit_fulladder dut(a1, b1, a2, b2, cin, s1, s2, cout);

initial begin
a1 = 0;
b1 = 0;
a2 = 0;
b2 = 0;
cin = 0;
#10;
a1 = 0;
b1 = 1;
a2 = 1;
b2 = 1;
cin = 1;
#10;
a1 = 1;
b1 = 0;
a2 = 1;
b2 = 0;
cin = 0;
#10;

end

initial begin
$monitor(" a1b1a2b2 = %b, s1 = %b, s2 = %b, cout = %b", {a1, b1, a2, b2}, s1, s2, cout);
#30;
$finish;
end

endmodule
```

Simulation Results:



Name	Value
a2	1
b2	0
a1	1
b1	0
cin	0
s1	1
s2	1
cout	0

```
$finish called at time : 30 ns : File "C:/Users/uxk220003/Desktop/Verliog Codes/Projects/2bit_fulladder_bhev/twobit_fulladder_tb.v
INFO: [USF-XSim-96] XSim completed. Design snapshot 'twobit_fulladder_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:03 ; elapsed = 00:00:07 . Memory (MB): peak = 2547.527 ; gain = 0.000
```

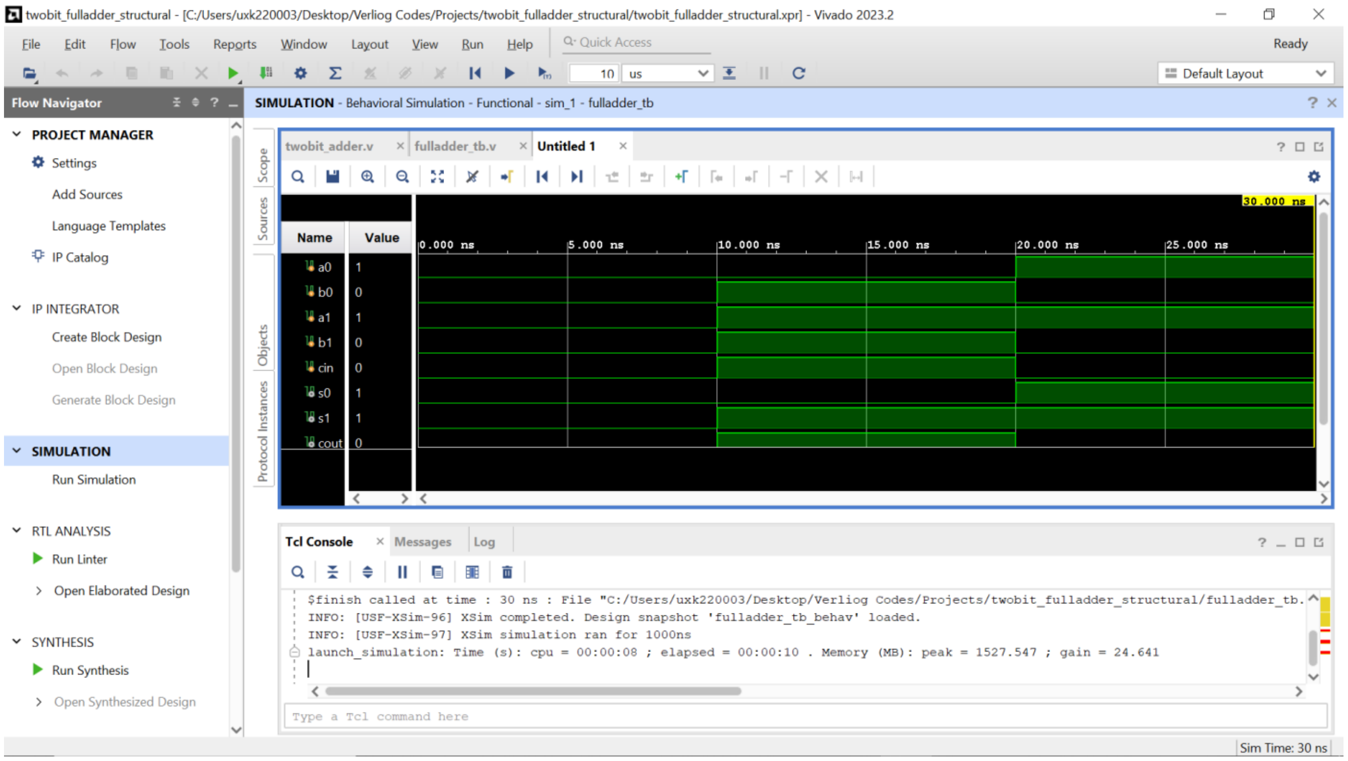
Structural Code:

```
module twobit_fulladder(a, b, cin, s, cout);
input a,b,cin;
output s, cout;
wire w0,w1,w2;

xor z1(w0, a, b);
and z2(w1, a, b);
and z3(w2, cin, w0);
xor z4(s, w0, cin);
or z5(cout, w1, w2);

endmodule

module twobit_adder(a0, b0, a1, b1, cin, s0, s1, cout);
input a0,b0,a1,b1, cin;
output s0,s1,cout;

wire c1;

twobit_fulladder fa1(a0, b0, cin, s0, c1);
twobit_fulladder fa2(a1, b1, c1, s1, cout);

endmodule
```

TestBench for Structural Code:

```
module fulladder_tb( );
reg a0,b0,a1,b1, cin;
wire s0,s1,cout;


twobit_adder dur(a0, b0, a1, b1, cin, s0, s1, cout);
initial begin
a0 = 0;
b0 = 0;
a1 = 0;
b1 = 0;
cin = 0;
#10;
a0 = 0;
b0 = 1;
a1 = 1;
b1 = 1;
cin = 1;
#10;
a0 = 1;
b0 = 0;
a1 = 1;
b1 = 0;
cin = 0;
#10;
end

initial begin
$monitor(" a0b0a1b1 = %b, s0 = %b, s1 = %b, cout = %b", {a0, b0, a1, b1}, s0, s1, cout);
#30;
$finish;
end

endmodule
```

Simulation Results:



twobit_fulladder_structural - [C:/Users/uxk220003/Desktop/Verliog Codes/Projects/twobit_fulladder_structural/twobit_fulladder_structural.xpr] - Vivado 2023.2

SIMULATION - Behavioral Simulation - Functional - sim_1 - fulladder_tb

Name	Value
a0	1
b0	0
a1	1
b1	0
cin	0
s0	1
s1	1
cout	0

```
$finish called at time : 30 ns : File "C:/Users/uxk220003/Desktop/Verliog Codes/Projects/twobit_fulladder_structural/fulladder_tb.
INFO: [USF-XSim-96] XSim completed. Design snapshot 'fulladder_tb_behav' loaded.
INFO: [USF-XSim-97] XSim simulation ran for 1000ns
launch_simulation: Time (s): cpu = 00:00:08 ; elapsed = 00:00:10 . Memory (MB): peak = 1527.547 ; gain = 24.641
```

Sim Time: 30 ns

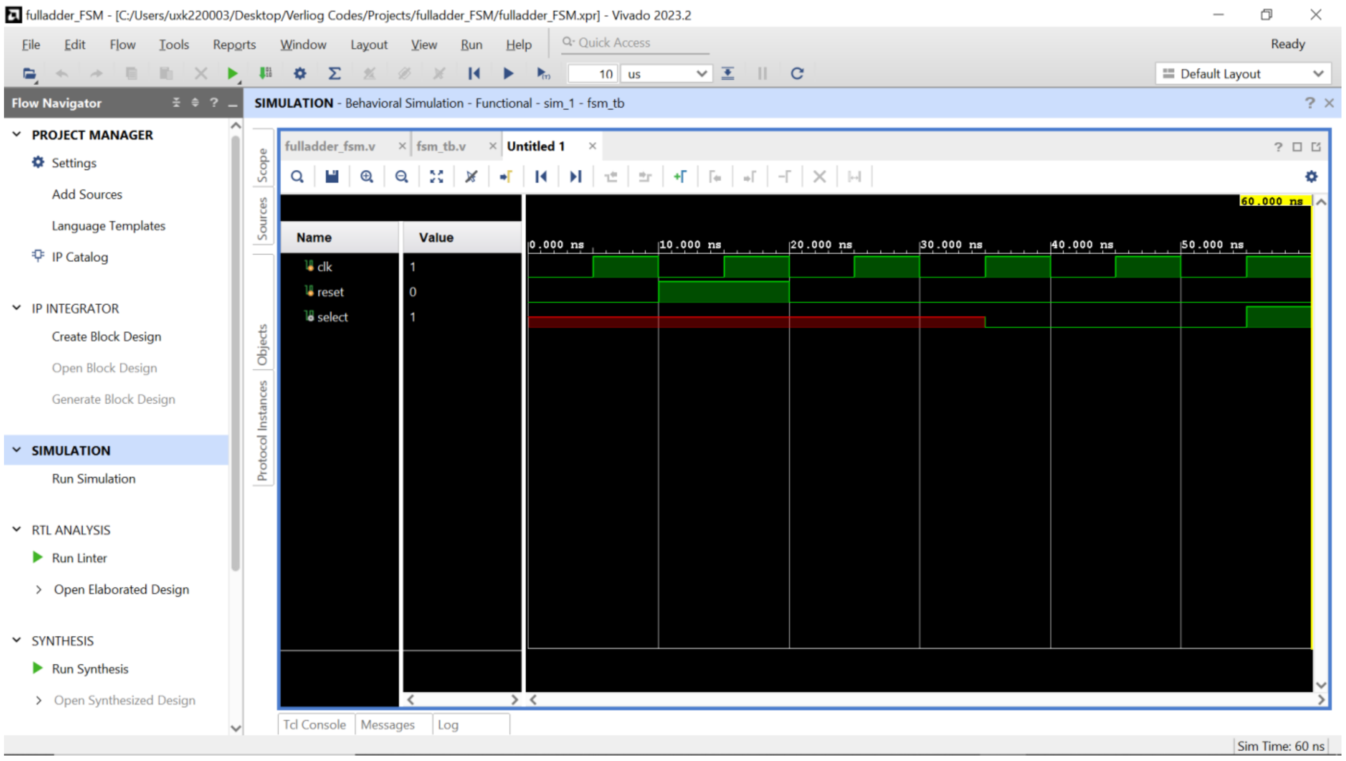
FSM Code:

```
module fulladder_fsm(clk, reset, select);
input clk, reset;
output reg select;
parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10;

reg [1:0] state;
reg [1:0] nextstate;

always @(posedge clk)
begin
case(state)
s0: begin
     nextstate <=  s1;
    end
s1: begin
     nextstate <= s2;
     select <= 0;
    end

s2: begin
     nextstate <= s0;
     select <= 1;
    end
default: begin
          nextstate <= s0;
         end
endcase
end

always @(posedge clk or posedge reset)
begin
  if(reset)
   state <= s0;
  else
   state <= nextstate;
end
endmodule
```

TestBench for FSM:

```
module fsm_tb( );
reg clk, reset;
wire select;

fulladder_fsm dut(clk, reset, select);

initial begin
clk = 0;
forever #5 clk = ~clk;
end

initial begin
reset = 1'b0;
#10;
reset = 1'b1;
#10;
reset = 1'b0;
end

initial begin
$monitor("clk = %b, reset = %b, select = %b", clk, reset, select);
#60;
$finish;
end

endmodule
```

Simulation Results:



fulladder_FSM - [C:/Users/uxk220003/Desktop/Verliog Codes/Projects/fulladder_FSM/fulladder_FSM.xpr] - Vivado 2023.2

SIMULATION - Behavioral Simulation - Functional - sim_1 - fsm_tb

Name	Value
clk	1
reset	0
select	1

Sim Time: 60 ns

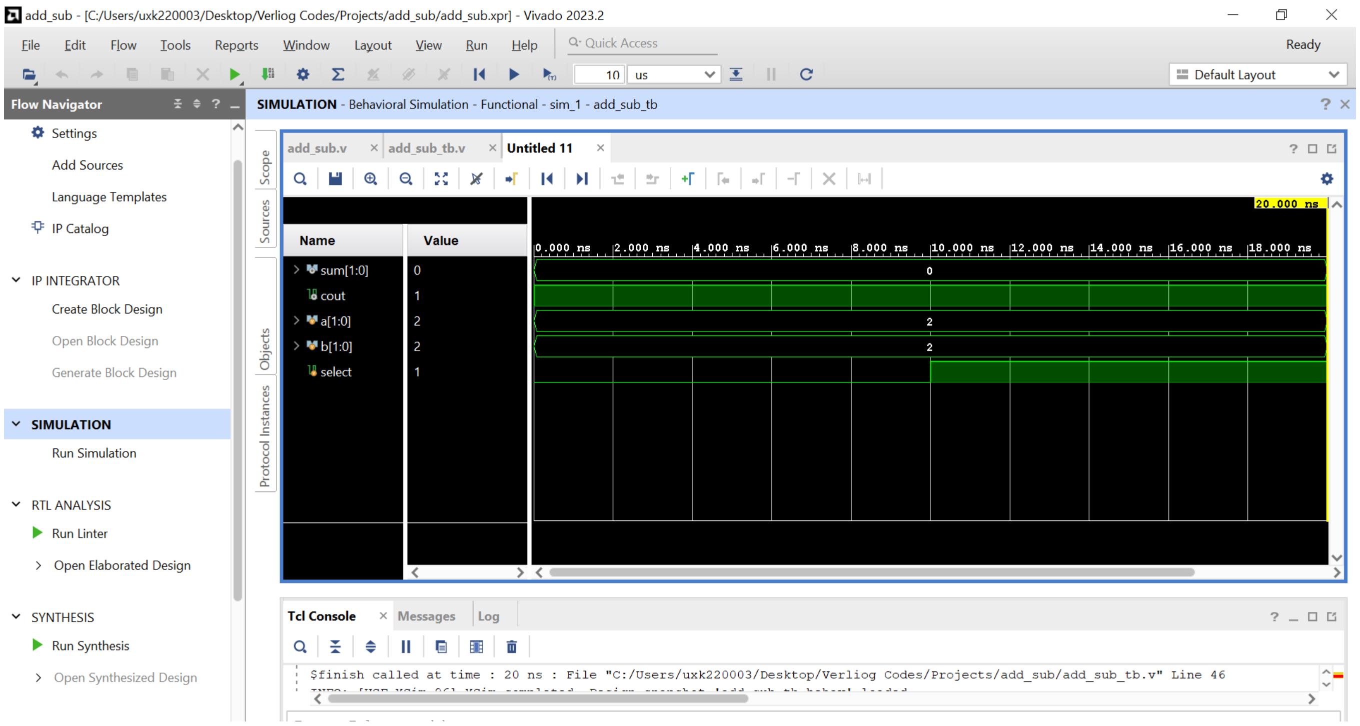
Add-Sub module the select input from FSM:

```
module fulladder(s, cout, a, b, cin);
input a, b, cin;
output s, cout;
wire w0, w1, w2;

assign w0 = a ^ b;
assign s = cin ^ w0;
assign w1 = a & b;
assign w2 = cin & w0;
assign cout = w2 | w1;

endmodule


module add_sub(sum,cout,a,b,select);
output [1:0] sum;
output cout;
input [1:0]a,b;
input select; //basically a cin
wire c0;
wire w1, w2;

xor x1(w1, b[0],select);
xor x2(w2, b[1], select);

fulladder FA1(sum[0], c0, a[0],w1,select);
fulladder FA2(sum[1], cout, a[1],w2,c0);

endmodule
```

TestBench for Add-Sub Module:

```
module add_sub_tb( );
wire [1:0] sum;
wire cout;
reg [1:0]a,b;
reg select; //basically a cin

add_sub dut(sum,cout,a,b,select);

initial begin
a = 2'b10;
b = 2'b10;
select = 0;
#10;

a = 2'b10;
b = 2'b10;
select = 1;
#10;
end

initial begin
$monitor("a = %b, b = %b, select = %b, sum = %b, cout = %b", a,b,select, sum, cout);
#20;
$finish;
end

endmodule
```

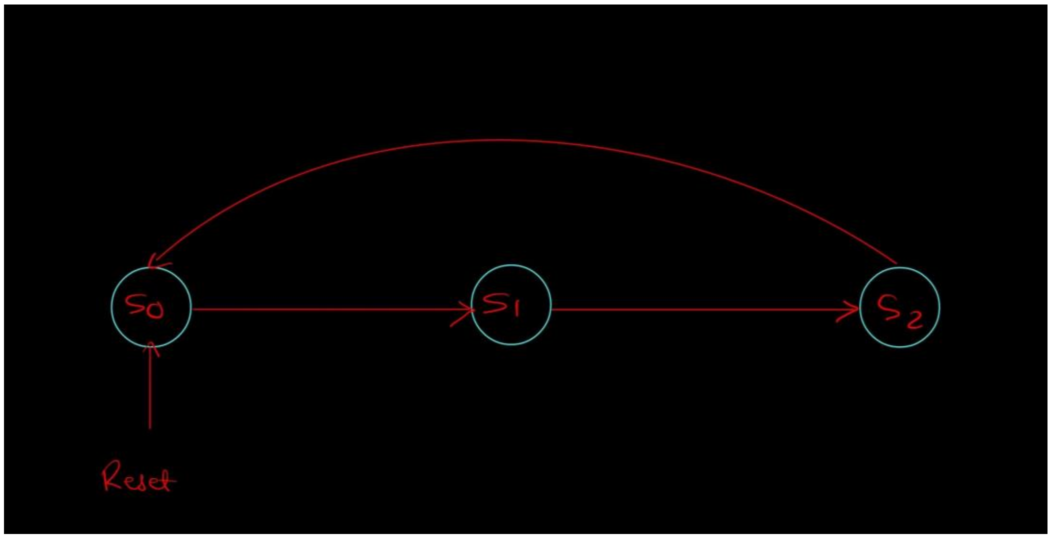
Simulation Results:



Name	Value
sum[1:0]	0
cout	1
a[1:0]	2
b[1:0]	2
select	1

```
$finish called at time : 20 ns : File "C:/Users/uxk220003/Desktop/Verliog Codes/Projects/add_sub/add_sub_tb.v" Line 46
```

State Transition Diagram:


S0
S1
S2
Reset