Digital Design and Computer Organisation Laboratory

UE22CS251A

3rd Semester, Academic Year 2023-24

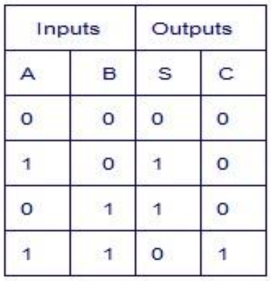
Date: 26/08/2024

Name: Keerthan P.V.	SRN: PES2UG23CS272	Section 3E

Week:3 Program Number:1

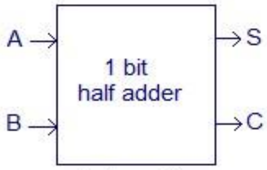
TITLE:

WRITE A VERILOG PROGRAM TO MODEL A HALF ADDER THAT CAN ADD TWO BITS. GENERATE THE VVP OUTPUT AND SIMULATION WAVEFORM USING GTKWAVE. VERIFY THE OUTPUT AND WAVEFORM WITH THE TRUTH TABLE

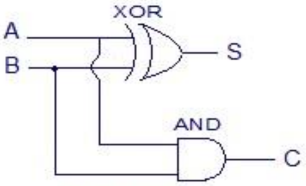


Inputs		Outputs	
A	B	S	C
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

Truth table



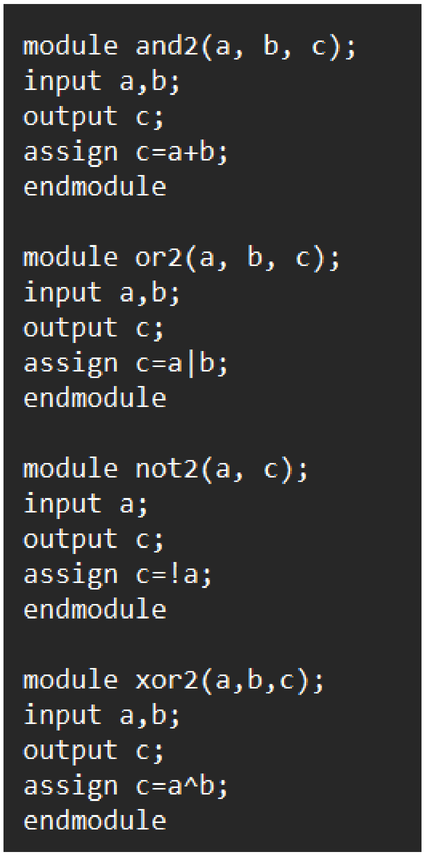
Schematic



Realization

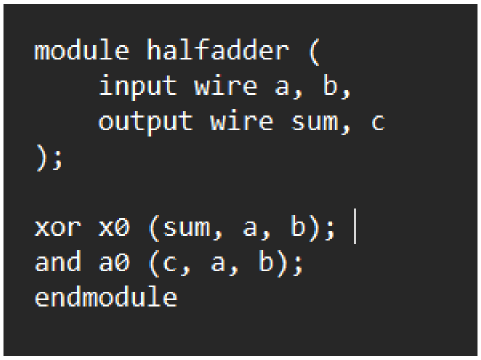
I. Verilog Code and Test Bench Code

1.libha.v



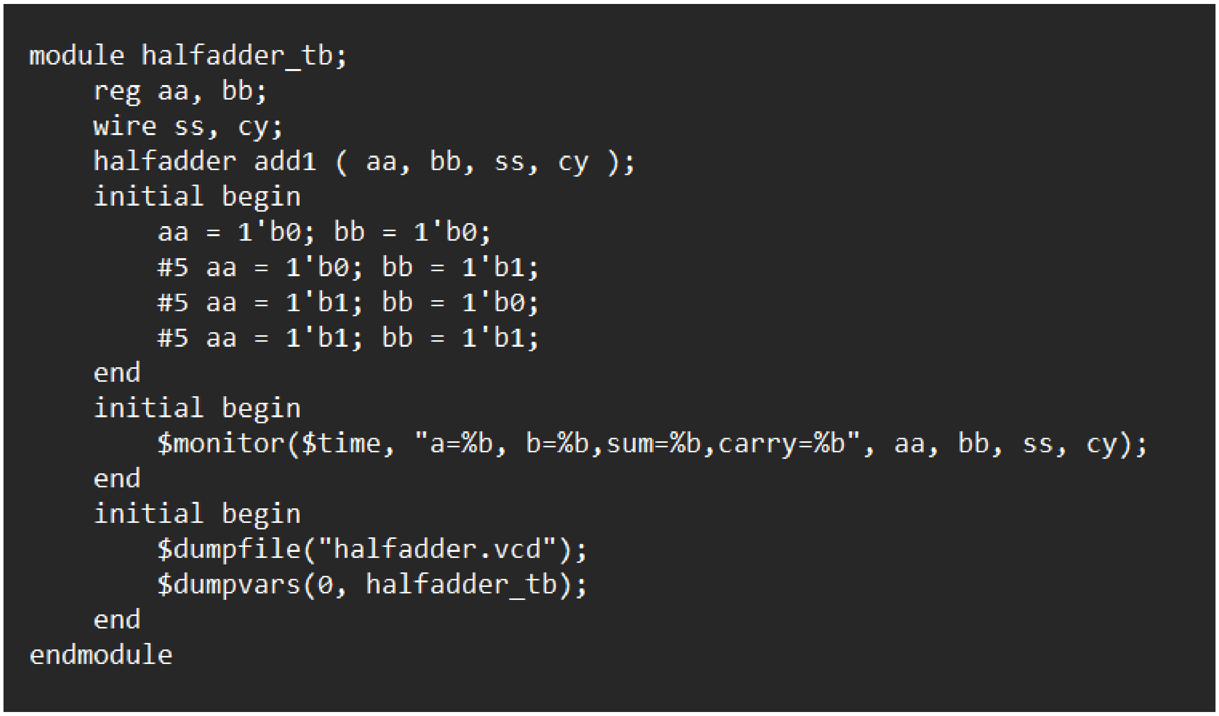
```
module and2(a, b, c);
input a,b;
output c;
assign c=a+b;
endmodule

module or2(a, b, c);
input a,b;
output c;
assign c=a|b;
endmodule

module not2(a, c);
input a;
output c;
assign c=!a;
endmodule

module xor2(a,b,c);
input a,b;
output c;
assign c=a^b;
endmodule
```

2.ha.v



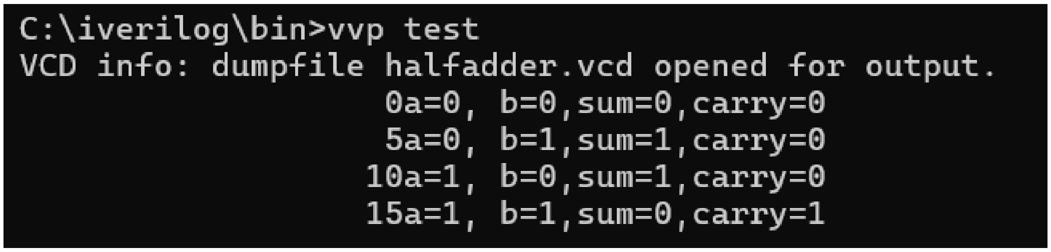
```
module halfadder (
    input wire a, b,
    output wire sum, c
);

xor x0 (sum, a, b);
and a0 (c, a, b);
endmodule
```

3.tb_ha.v



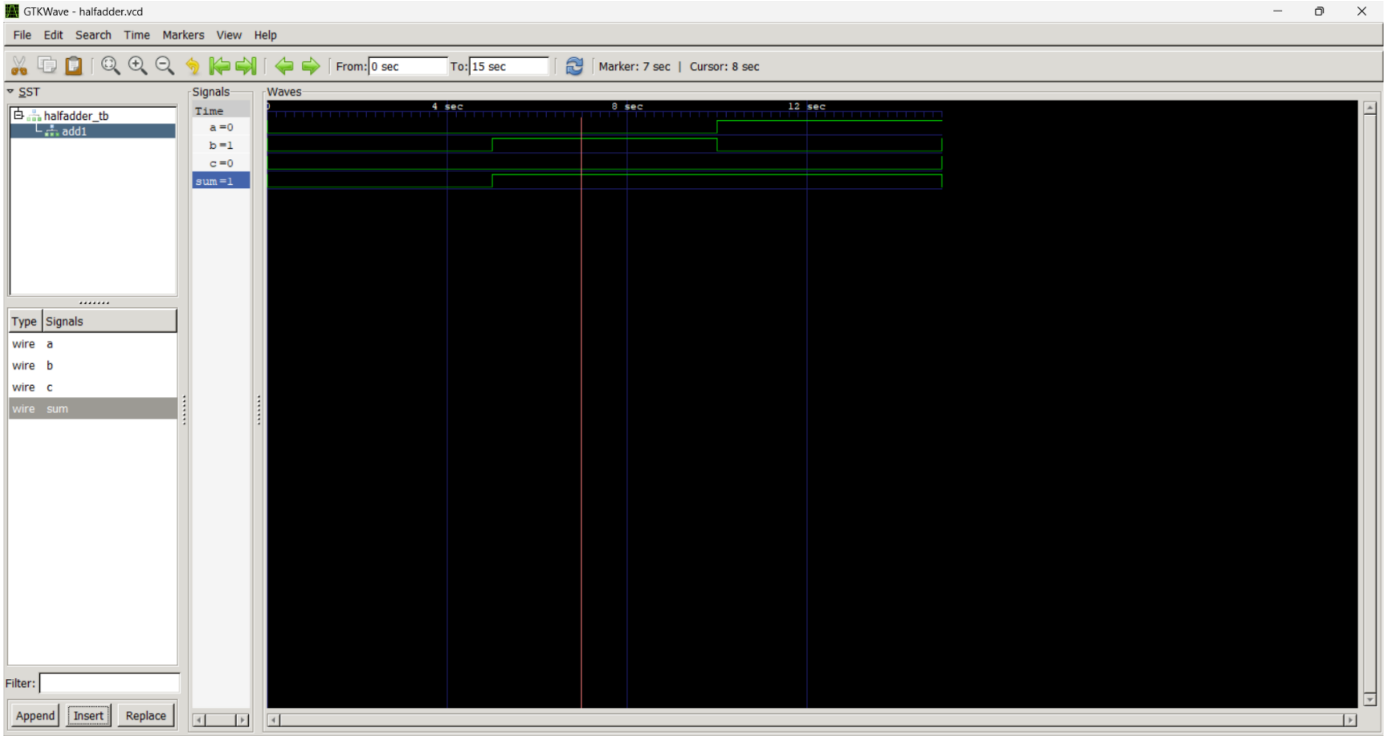
```
module halfadder_tb;
    reg aa, bb;
    wire ss, cy;
    halfadder add1 ( aa, bb, ss, cy );
    initial begin
        aa = 1'b0; bb = 1'b0;
        #5 aa = 1'b0; bb = 1'b1;
        #5 aa = 1'b1; bb = 1'b0;
        #5 aa = 1'b1; bb = 1'b1;
    end
    initial begin
        $monitor($time, "a=%b, b=%b,sum=%b,carry=%b", aa, bb, ss, cy);
    end
    initial begin
        $dumpfile("halfadder.vcd");
        $dumpvars(0, halfadder_tb);
    end
endmodule
```

II. Verilog VVP Output Screen Shot



```
C:\iverilog\bin>vvp test
VCD info: dumpfile halfadder.vcd opened for output.
                   0a=0, b=0,sum=0,carry=0
                   5a=0, b=1,sum=1,carry=0
                  10a=1, b=0,sum=1,carry=0
                  15a=1, b=1,sum=0,carry=1
```

III. GTKWAVE Screenshot



From: 0 sec To: 15 sec

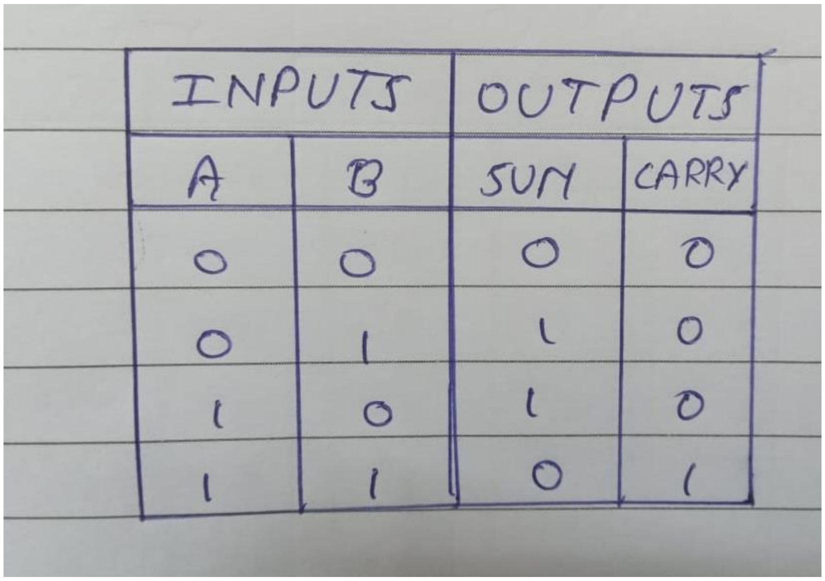
Marker: 7 sec | Cursor: 8 sec

halfadder_tb
add1

Type	Signals
wire	a
wire	b
wire	c
wire	sum

a =0
b =1
c =0
sum =1

Truth table:



INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

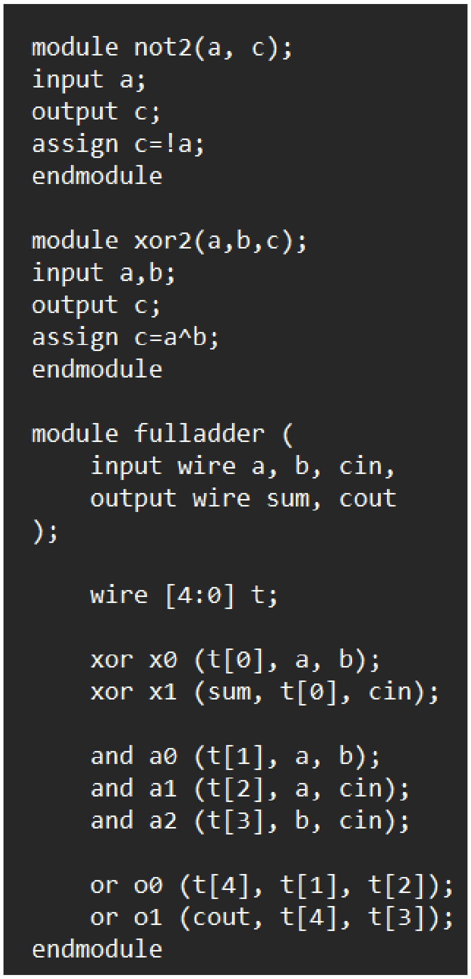
Week:3 Program Number: 2

TITLE :

WRITE A VERILOG PROGRAM TO MODEL A FULL ADDER THAT CAN ADD TWO BITS ALONG WITH AN INPUT CARRY. GENERATE THE VVP OUTPUT AND SIMULATION WAVEFORM USING GTKWAVE. VERIFY THE OUTPUT AND WAVEFORM WITH THE TRUTH TABLE

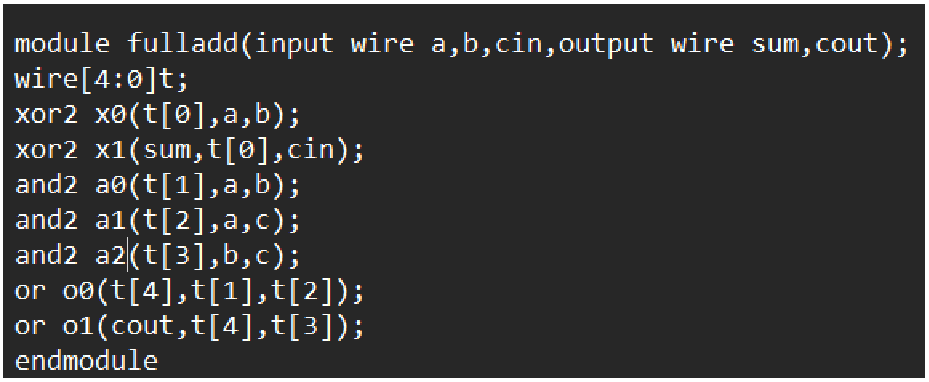
I. Verilog Code and Test Bench Code

1. fa.v



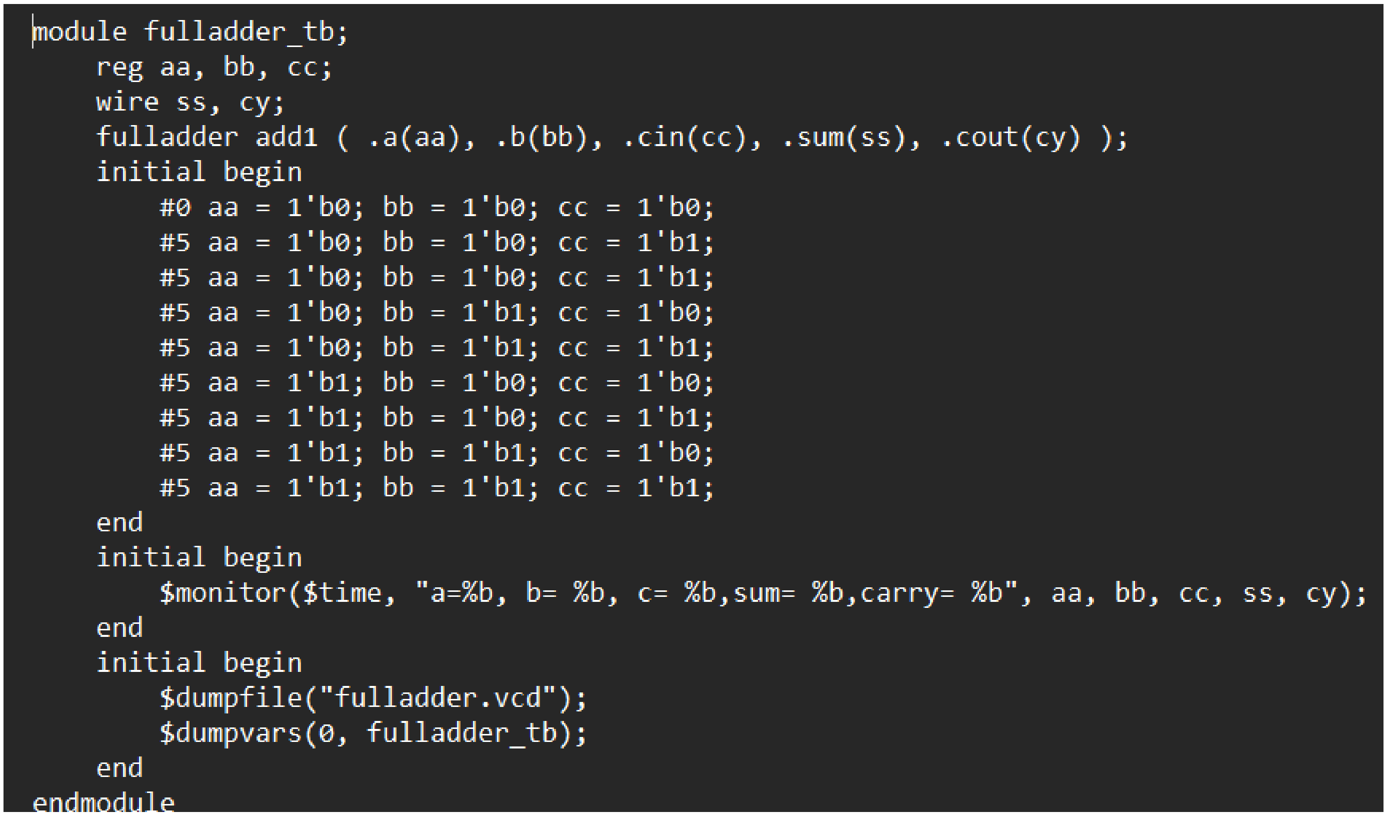
```
module not2(a, c);
input a;
output c;
assign c=!a;
endmodule

module xor2(a,b,c);
input a,b;
output c;
assign c=a^b;
endmodule

module fulladder (
    input wire a, b, cin,
    output wire sum, cout
);

    wire [4:0] t;

    xor x0 (t[0], a, b);
    xor x1 (sum, t[0], cin);

    and a0 (t[1], a, b);
    and a1 (t[2], a, cin);
    and a2 (t[3], b, cin);

    or o0 (t[4], t[1], t[2]);
    or o1 (cout, t[4], t[3]);
endmodule
```

2.fas.v



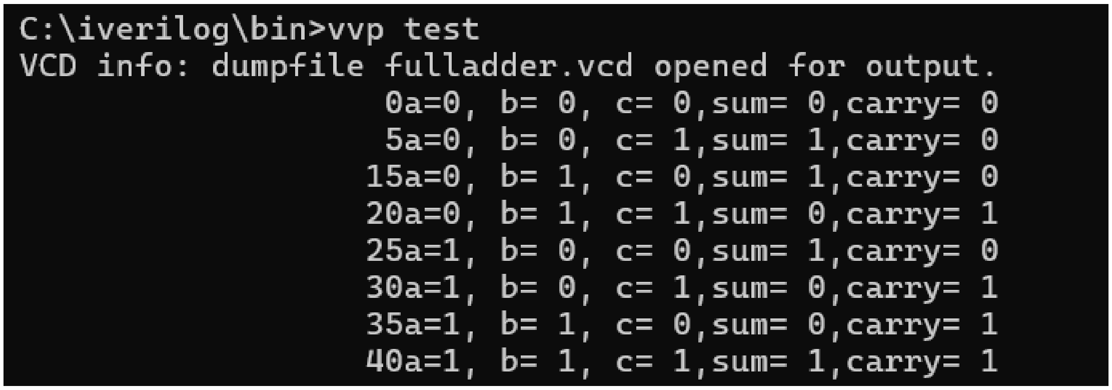
```
module fulladd(input wire a,b,cin,output wire sum,cout);
wire[4:0]t;
xor2 x0(t[0],a,b);
xor2 x1(sum,t[0],cin);
and2 a0(t[1],a,b);
and2 a1(t[2],a,c);
and2 a2(t[3],b,c);
or o0(t[4],t[1],t[2]);
or o1(cout,t[4],t[3]);
endmodule
```

3.fa_tb.v



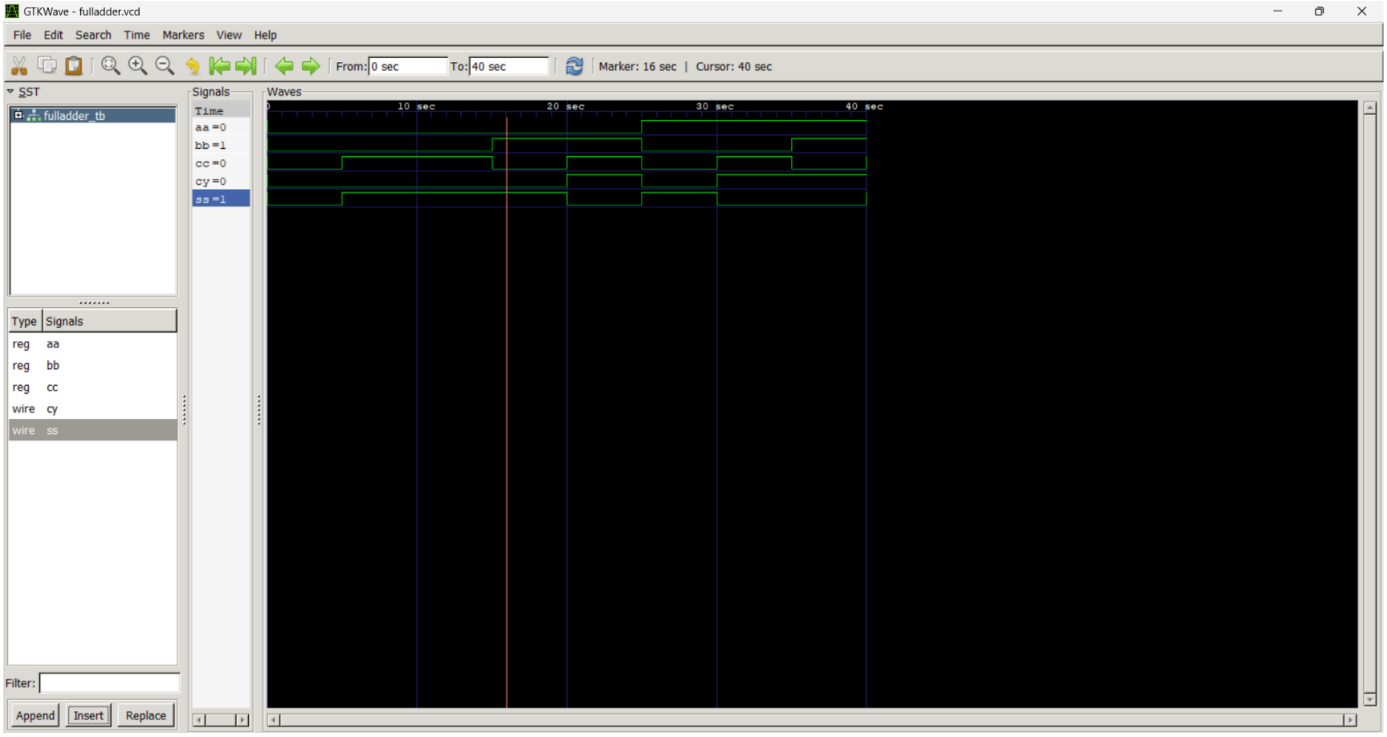
```
module fulladder_tb;
    reg aa, bb, cc;
    wire ss, cy;
    fulladder add1 ( .a(aa), .b(bb), .cin(cc), .sum(ss), .cout(cy) );
    initial begin
        #0 aa = 1'b0; bb = 1'b0; cc = 1'b0;
        #5 aa = 1'b0; bb = 1'b0; cc = 1'b1;
        #5 aa = 1'b0; bb = 1'b0; cc = 1'b1;
        #5 aa = 1'b0; bb = 1'b1; cc = 1'b0;
        #5 aa = 1'b0; bb = 1'b1; cc = 1'b1;
        #5 aa = 1'b1; bb = 1'b0; cc = 1'b0;
        #5 aa = 1'b1; bb = 1'b0; cc = 1'b1;
        #5 aa = 1'b1; bb = 1'b1; cc = 1'b0;
        #5 aa = 1'b1; bb = 1'b1; cc = 1'b1;
    end
    initial begin
        $monitor($time, "a=%b, b= %b, c= %b,sum= %b,carry= %b", aa, bb, cc, ss, cy);
    end
    initial begin
        $dumpfile("fulladder.vcd");
        $dumpvars(0, fulladder_tb);
    end
endmodule
```

II. Verilog VVP Output Screen Shot



```
C:\iverilog\bin>vvp test
VCD info: dumpfile fulladder.vcd opened for output.
                   0a=0, b= 0, c= 0,sum= 0,carry= 0
                   5a=0, b= 0, c= 1,sum= 1,carry= 0
                  15a=0, b= 1, c= 0,sum= 1,carry= 0
                  20a=0, b= 1, c= 1,sum= 0,carry= 1
                  25a=1, b= 0, c= 0,sum= 1,carry= 0
                  30a=1, b= 0, c= 1,sum= 0,carry= 1
                  35a=1, b= 1, c= 0,sum= 0,carry= 1
                  40a=1, b= 1, c= 1,sum= 1,carry= 1
```

III. GTKWAVE Screenshot



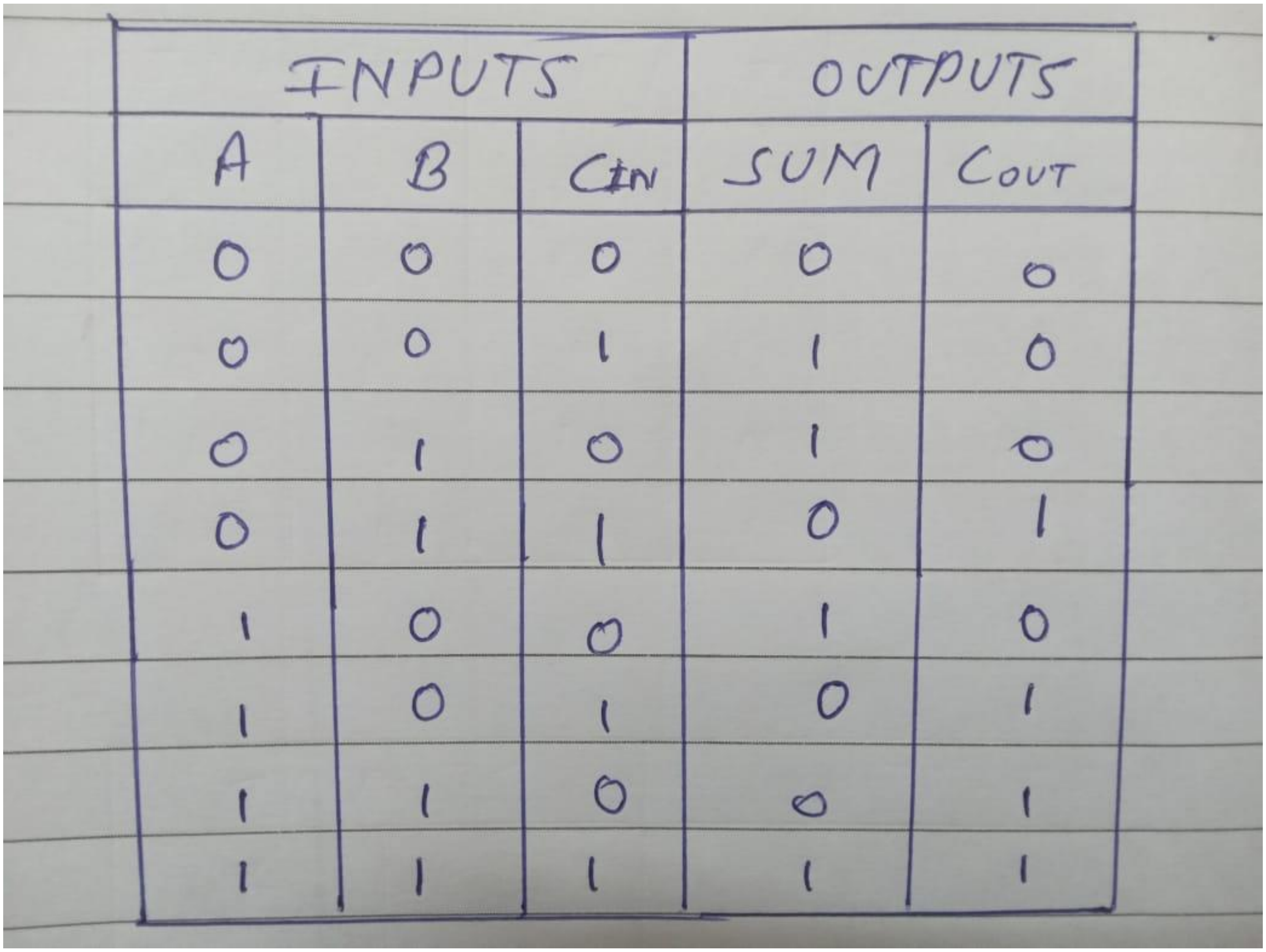
GTKWave - fulladder.vcd

From: 0 sec To: 40 sec

Marker: 16 sec | Cursor: 40 sec

Type	Signals
reg	aa
reg	bb
reg	cc
wire	cy
wire	ss

Truth table:



INPUTS			OUTPUTS	
A	B	CIN	SUM	COUT
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

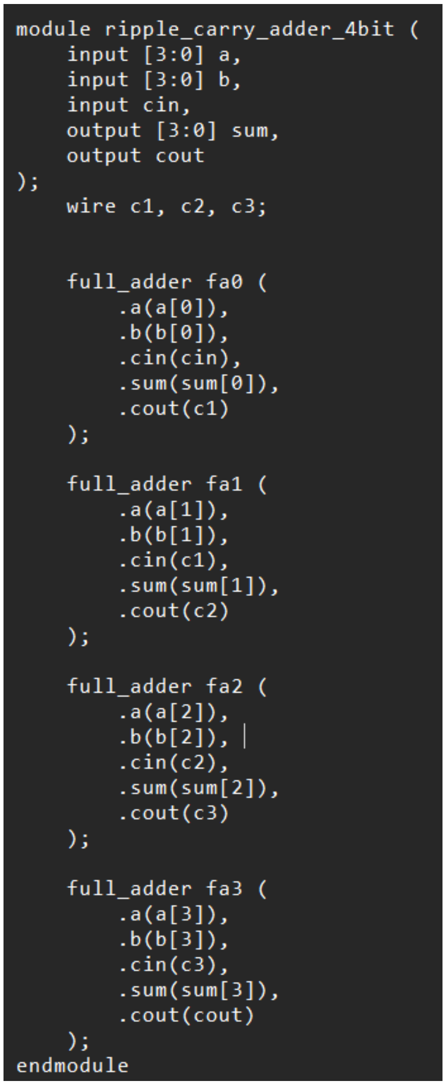
Week:3 Program Number: 3

TITLE:

WRITE A VERILOG PROGRAM TO MODEL A FOUR BIT FULL ADDER THAT GENERATES A FOUR BIT SUM AND A 1 BIT CARRY OUTPUT.GENERATE THE VVP OUTPUT AND SIMULATION WAVEFORM
USING GTKWAVE. VERIFY THE OUTPUT AND WAVEFORM WITH THE TRUTH TABLE

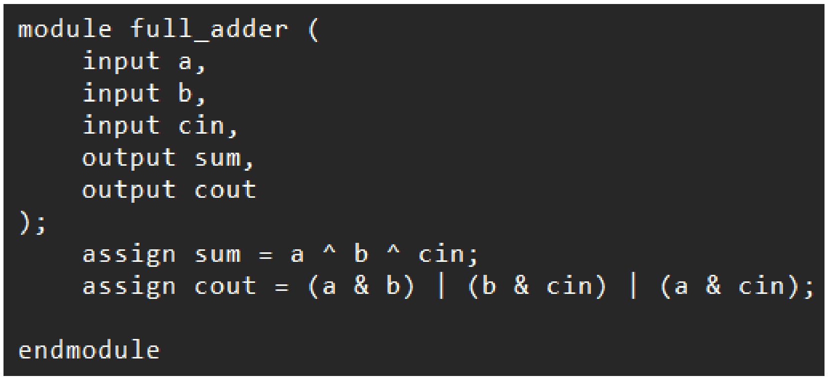
I. Verilog Code and Test Bench Code

1.rca.v



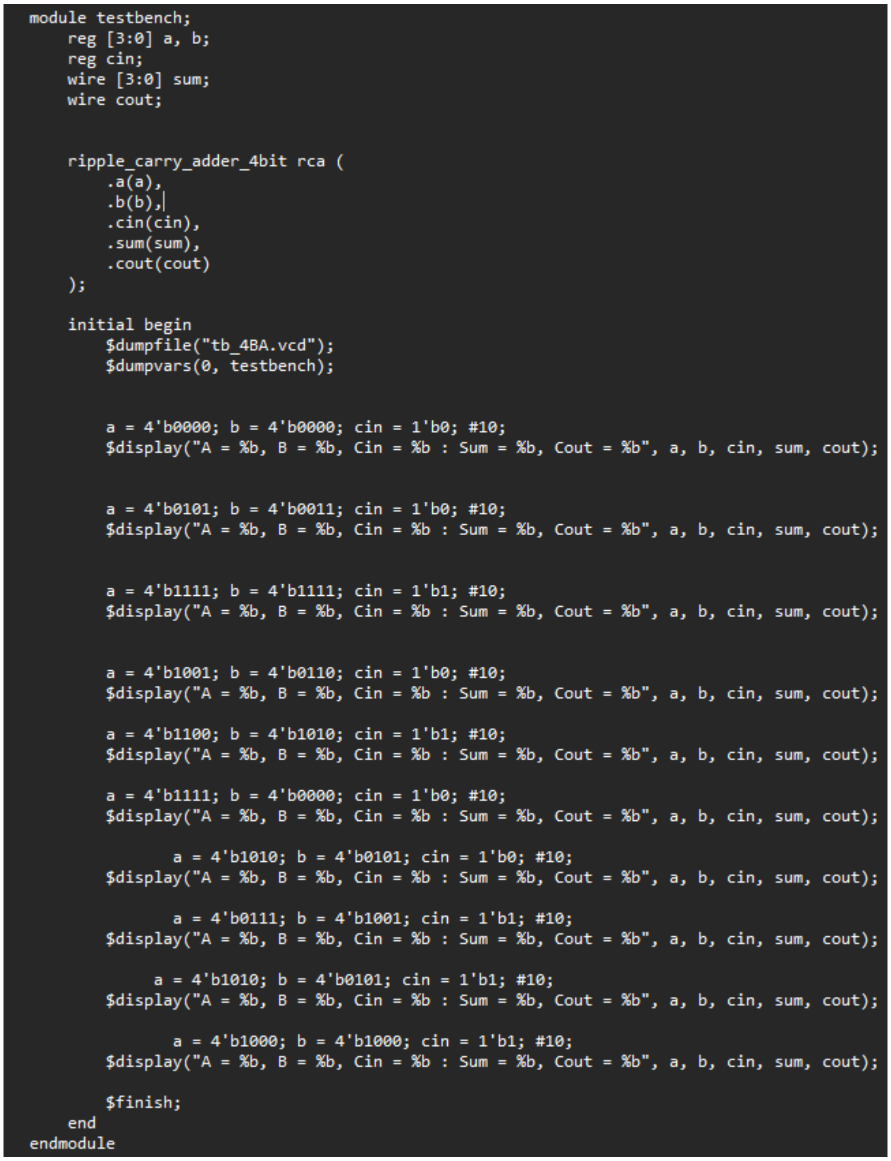
```
module ripple_carry_adder_4bit (
    input [3:0] a,
    input [3:0] b,
    input cin,
    output [3:0] sum,
    output cout
);
    wire c1, c2, c3;


    full_adder fa0 (
        .a(a[0]),
        .b(b[0]),
        .cin(cin),
        .sum(sum[0]),
        .cout(c1)
    );

    full_adder fa1 (
        .a(a[1]),
        .b(b[1]),
        .cin(c1),
        .sum(sum[1]),
        .cout(c2)
    );

    full_adder fa2 (
        .a(a[2]),
        .b(b[2]),
        .cin(c2),
        .sum(sum[2]),
        .cout(c3)
    );

    full_adder fa3 (
        .a(a[3]),
        .b(b[3]),
        .cin(c3),
        .sum(sum[3]),
        .cout(cout)
    );
endmodule
```

2.rcas.v



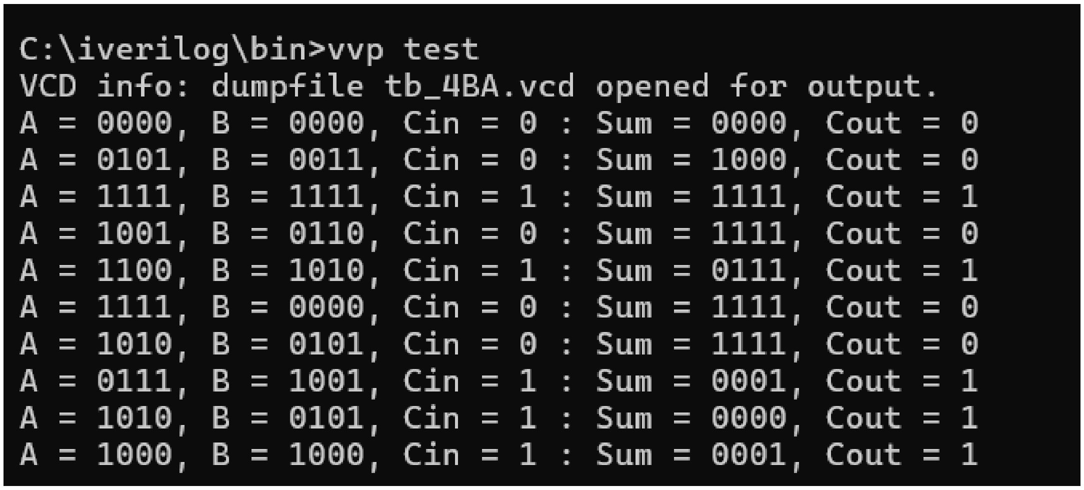
```
module full_adder (
    input a,
    input b,
    input cin,
    output sum,
    output cout
);
    assign sum = a ^ b ^ cin;
    assign cout = (a & b) | (b & cin) | (a & cin);

endmodule
```

3.rca_tb.v



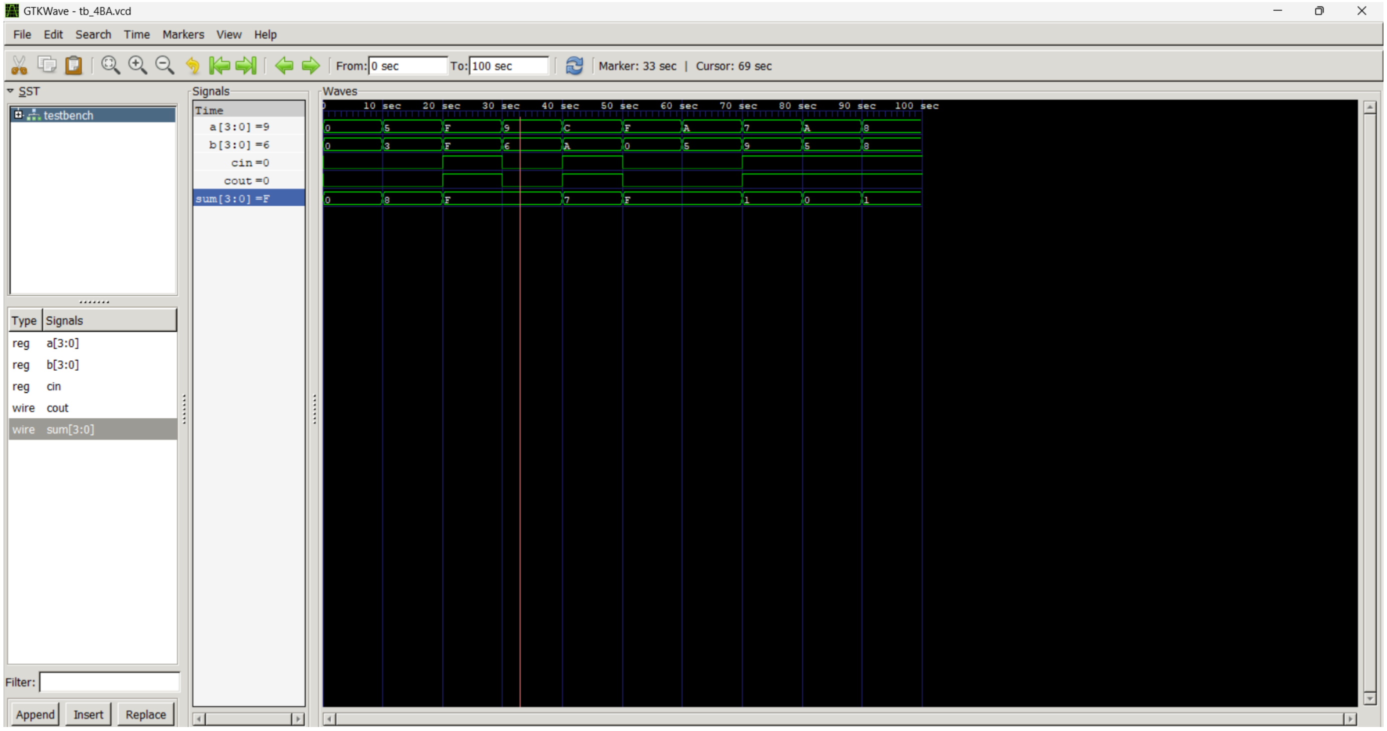
```
module testbench;
    reg [3:0] a, b;
    reg cin;
    wire [3:0] sum;
    wire cout;


    ripple_carry_adder_4bit rca (
        .a(a),
        .b(b),
        .cin(cin),
        .sum(sum),
        .cout(cout)
    );

    initial begin
        $dumpfile("tb_4BA.vcd");
        $dumpvars(0, testbench);


        a = 4'b0000; b = 4'b0000; cin = 1'b0; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);


        a = 4'b0101; b = 4'b0011; cin = 1'b0; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);


        a = 4'b1111; b = 4'b1111; cin = 1'b1; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);


        a = 4'b1001; b = 4'b0110; cin = 1'b0; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);

        a = 4'b1100; b = 4'b1010; cin = 1'b1; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);

        a = 4'b1111; b = 4'b0000; cin = 1'b0; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);

               a = 4'b1010; b = 4'b0101; cin = 1'b0; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);

               a = 4'b0111; b = 4'b1001; cin = 1'b1; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);

             a = 4'b1010; b = 4'b0101; cin = 1'b1; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);

               a = 4'b1000; b = 4'b1000; cin = 1'b1; #10;
        $display("A = %b, B = %b, Cin = %b : Sum = %b, Cout = %b", a, b, cin, sum, cout);

        $finish;
    end
endmodule
```

II. Verilog VVP Output Screen Shot



```
C:\iverilog\bin>vvp test
VCD info: dumpfile tb_4BA.vcd opened for output.
A = 0000, B = 0000, Cin = 0 : Sum = 0000, Cout = 0
A = 0101, B = 0011, Cin = 0 : Sum = 1000, Cout = 0
A = 1111, B = 1111, Cin = 1 : Sum = 1111, Cout = 1
A = 1001, B = 0110, Cin = 0 : Sum = 1111, Cout = 0
A = 1100, B = 1010, Cin = 1 : Sum = 0111, Cout = 1
A = 1111, B = 0000, Cin = 0 : Sum = 1111, Cout = 0
A = 1010, B = 0101, Cin = 0 : Sum = 1111, Cout = 0
A = 0111, B = 1001, Cin = 1 : Sum = 0001, Cout = 1
A = 1010, B = 0101, Cin = 1 : Sum = 0000, Cout = 1
A = 1000, B = 1000, Cin = 1 : Sum = 0001, Cout = 1
```

III. GTKWAVE Screenshot



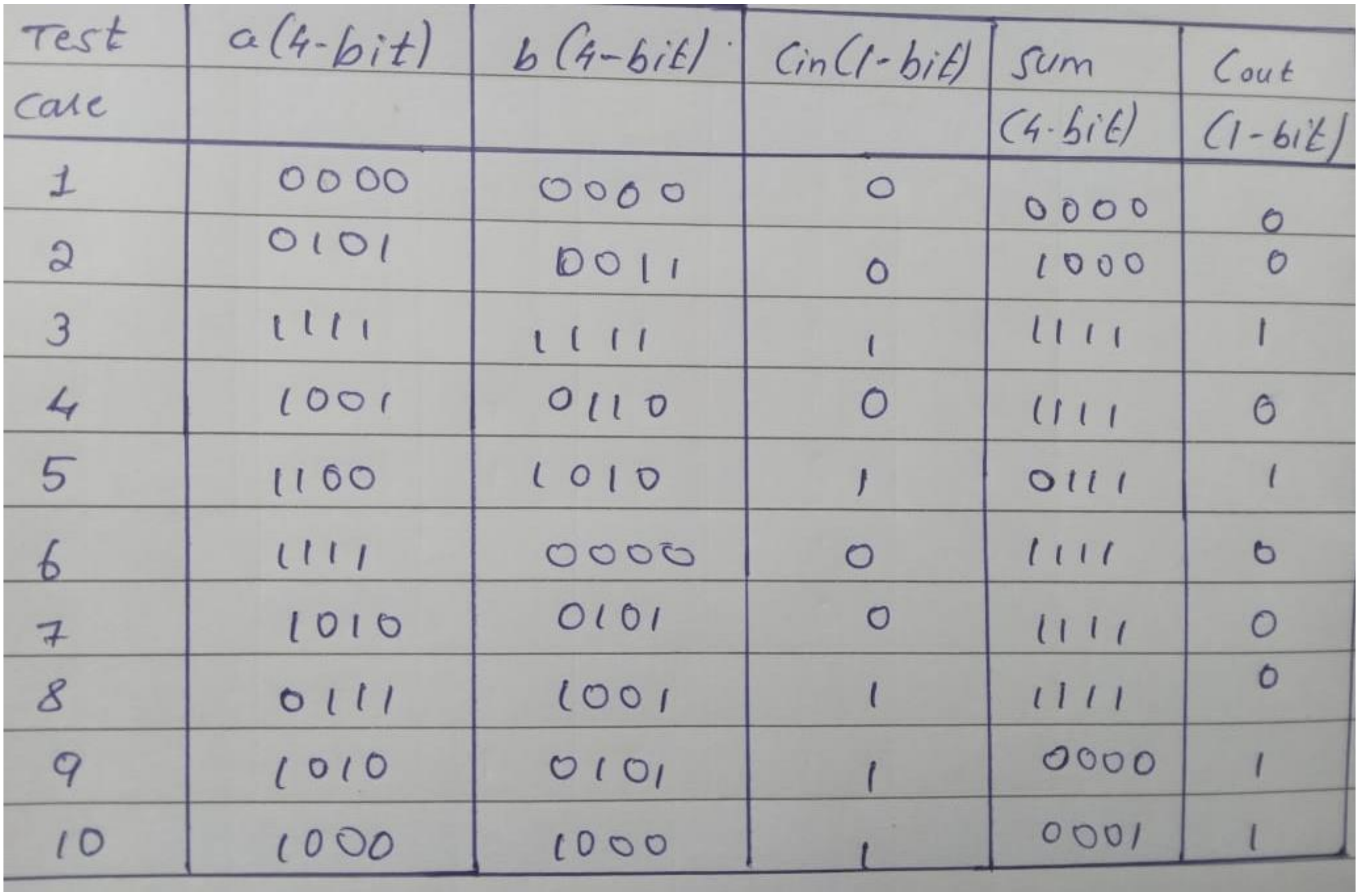
From: 0 sec To: 100 sec

Marker: 33 sec | Cursor: 69 sec

a[3:0] =9
b[3:0] =6
cin =0
cout =0
sum[3:0] =F

Type	Signals
reg	a[3:0]
reg	b[3:0]
reg	cin
wire	cout
wire	sum[3:0]

Truth table:



Test	a(4-bit)	b(4-bit)	Cin(1-bit)	Sum	Cout
Case				(4-bit)	(1-bit)
1	0000	0000	0	0000	0
2	0101	0011	0	1000	0
3	1111	1111	1	1111	1
4	1001	0110	0	1111	0
5	1100	1010	1	0111	1
6	1111	0000	0	1111	0
7	1010	0101	0	1111	0
8	0111	1001	1	1111	0
9	1010	0101	1	0000	1
10	1000	1000	1	0001	1

Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature: Keerthan P.V
Name: Keerthan P.V
SRN:PES2UG23CS272
Section: 3E
Date:26/08/2024