**Experiment no: 11 Date:**

**HDL PROGRAM FOR COMBINATIONAL CIRCUITS**

**AIM:**

To develop the source code for adders and subtractors by using VERILOG and obtain the simulation & synthesis.

**ALGORITM:**

Step1: Define the specifications and initialize the design.

Step2: Declare the name of the entity and architecture by using VHDL source code.

Step3: Write the source code in VERILOG.

Step4: Check the syntax and debug the errors if found, obtain the synthesis report.

Step5: Verify the output by simulating the source code.

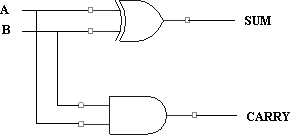
Step6: Write all possible combinations of input using the test bench.

Step7: Obtain the place and route report.

**BASIC ADDERS & SUBTRACTORS:**

**HALF ADDER:**

LOGIC DIAGRAM:



TRUTH TABLE:

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **SUM** | **CARRY** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

**VHDL SOURCE CODE:**

**Dataflow Modeling:**

module HF(sum,carry,a,b);

output sum, carry;

input a, b;

assign sum = a ^ b; // assigning sum

assign carry = a & b; // assigning carry

endmodule

**Test bench**

module HF\_TB();

reg a, b;

wire sum, carry;

HF uut(sum,carry,a,b);

initial

begin

a = 0; b =0;

#5; a = 0; b =1;

#5; a = 1; b =0;

#5; a = 1; b =1;

#5;

end

//Dump waves (only required here)

initial

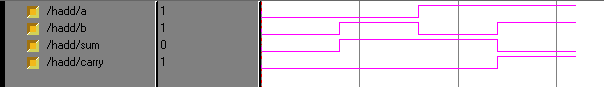
begin

$dumpfile("dump.vcd");

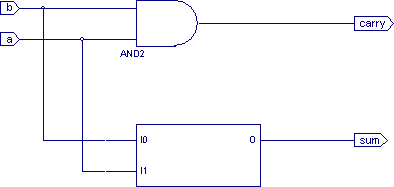
$dumpvars(1);

end

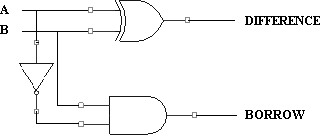
endmodule

**Simulation output:**

**Synthesis RTL Schematic:**



**HALF SUBTRACTOR:**

LOGIC DIAGRAM:

TRUTH TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **DIFFERENCE** | **BORROW** |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

**VERILOG SOURCE CODE:**

**Dataflow Modeling:**

module Half\_Subtractor\_2(output D, B, input X, Y);

assign D = X ^ Y;

assign B = ~X & Y;

endmodule

**Test Bench**

module Half\_Subtractor\_2\_tb;

wire D, B;

reg X, Y;

Half\_Subtractor\_2 Instance0 (D, B, X, Y);

initial begin

X = 0; Y = 0;

#1 X = 0; Y = 1;

#1 X = 1; Y = 0;

#1 X = 1; Y = 1;

end

initial begin

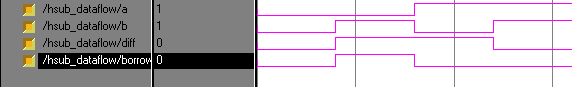
$monitor ("%t, X = %d| Y = %d| B = %d| D = %d", $time, X, Y, B, D);

$dumpfile("dump.vcd");

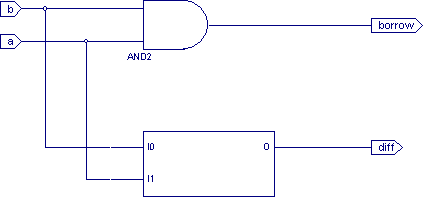
$dumpvars();

end

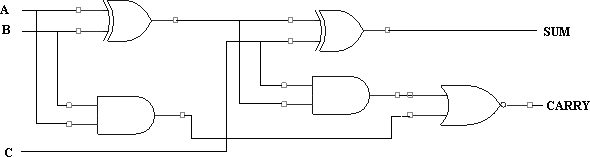
endmodule

**Simulation output:**

**Synthesis RTL Schematic:**



**FULL ADDER:**

LOGIC DIAGRAM:

TRUTH TABLE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **SUM** | **CARRY** |
| 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 |

**VERILOG SOURCE CODE:**

**Dataflow Modeling:**

module Full\_Adder (a, b, c, sum, cout);

input a;

input b;

input c;

output sum;

output cout;

assign sum=a^b^c;

assign cout=(a & b) | (b & c) | (c & a);

endmodule

**Test Bench**

module Test\_Full\_Adder;

reg a, b, c;

wire sum, cout;

Full\_Adder FA (a, b, c, cout, sum);

initial begin

$dumpfile("Test\_Full\_Adder.vcd");

$dumpvars(1, FA);

a=0; b=0; c=0;

#20 a=1; b=1;

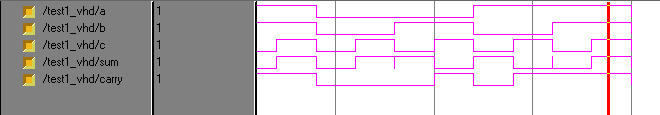
#20 a=0; b=0; c=1;

#20 a=1; c=0;

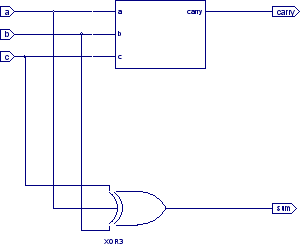
#20 $finish;

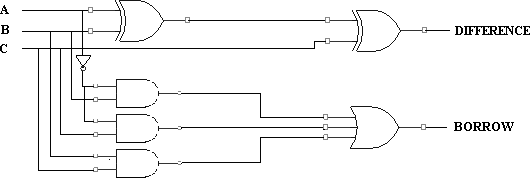
end

endmodule

**Simulation output:**

**Synthesis RTL Schematic:**



**FULL SUBTRACTOR:**

LOGIC DIAGRAM:

TRUTH TABLE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **DIFFERENCE** | **BORROW** |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**VERILOG SOURCE CODE:**

**Dataflow Modeling:**

module Full\_Subtractor\_3(output D, B, input X, Y, Z);

assign D = X ^ Y ^ Z;

assign B = ~X & (Y^Z) | Y & Z;

endmodule

**Test bench**

module Full\_Subtractor\_3\_tb;

wire D, B;

reg X, Y, Z;

Full\_Subtractor\_3 Instance0 (D, B, X, Y, Z);

initial begin

X = 0; Y = 0; Z = 0;

#1 X = 0; Y = 0; Z = 1;

#1 X = 0; Y = 1; Z = 0;

#1 X = 0; Y = 1; Z = 1;

#1 X = 1; Y = 0; Z = 0;

#1 X = 1; Y = 0; Z = 1;

#1 X = 1; Y = 1; Z = 0;

#1 X = 1; Y = 1; Z = 1;

end

initial begin

$monitor ("%t, X = %d| Y = %d| Z = %d| B = %d| D = %d", $time, X, Y, Z, B, D);

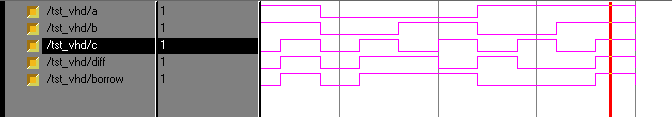
$dumpfile("dump.vcd");

$dumpvars();

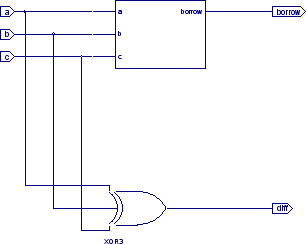
end

endmodule

**Simulation output:**



**Synthesis RTL Schematic:**



**RESULT:**

Thus the OUTPUT of HDL program for Combinational circuits is done and verified.