Mlitary College of Signals

National University of Sciences & Technology

DIGITAL LOGIC DESIGN - LAB

Submitted to: Lab Instructor Muhammad Hammad

Lab Report Number	Open_Ended_Lab
Submission Date:	January 26, 2022

Group Members Details

S. No	Names
1.	Hamna Younis
2.	Muhammad Awais

PROBLEM STATEMENT:

Design and implement 3-bit multiplier. Your multiplier should take binary input and display output on 7segment display. Use two output displays to display two decimal digits. Maximum multiplication output should be 49 in decimal. Implement the prototype first using proteus and then translate it to Hardware. Simulate the model in Modelsim and bring its Verilog implementation as well. Students are required to demonstrate the complete working of design and submit a detailed lab report.

DELIVERABLES:

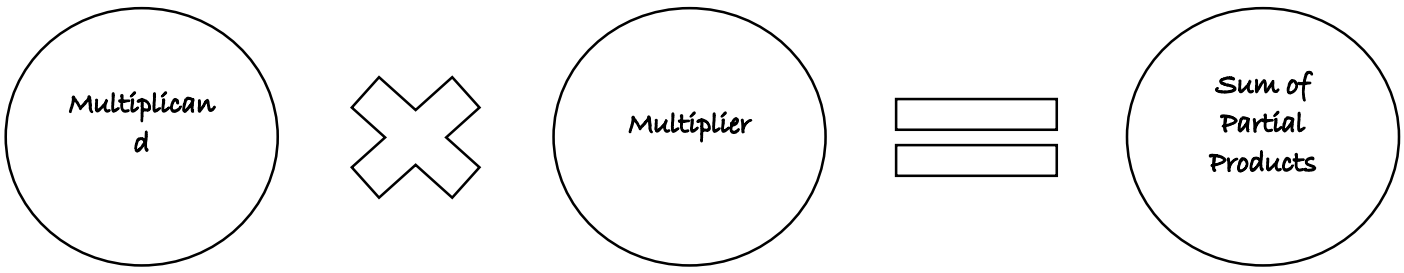
- Hardware Design.
- Modelism Simulation
- A well elaborated lab report having system block diagram, I/O resources, system cost, applications and references.
- Design pictures and a working video.

INTRODUCTION:

A binary multiplier is an electronic circuit that multiplies binary integers in digital electronics, such as computers. A binary multiplier is a digital or combinational logic circuit that multiplies two binary values. The two numbers are called multiplicand and multiplier, respectively, and the result is called a product. Both the multiplicand and the multiplier might be of different bit sizes. The bit size of the product is determined by the bit size of the multiplicand and multiplier. The product's bit size is equal to the sum of the multiplier's and multiplicand's bit sizes.

WORKING:

- Combinational Multipliers do **multiplication of two unsigned binary numbers**.
- Each bit of the multiplier is multiplied against the multiplicand, the product is aligned according to the position of the bit within the multiplier, and the resulting products are then summed to form the final result.



HARDWARE DESIGN:

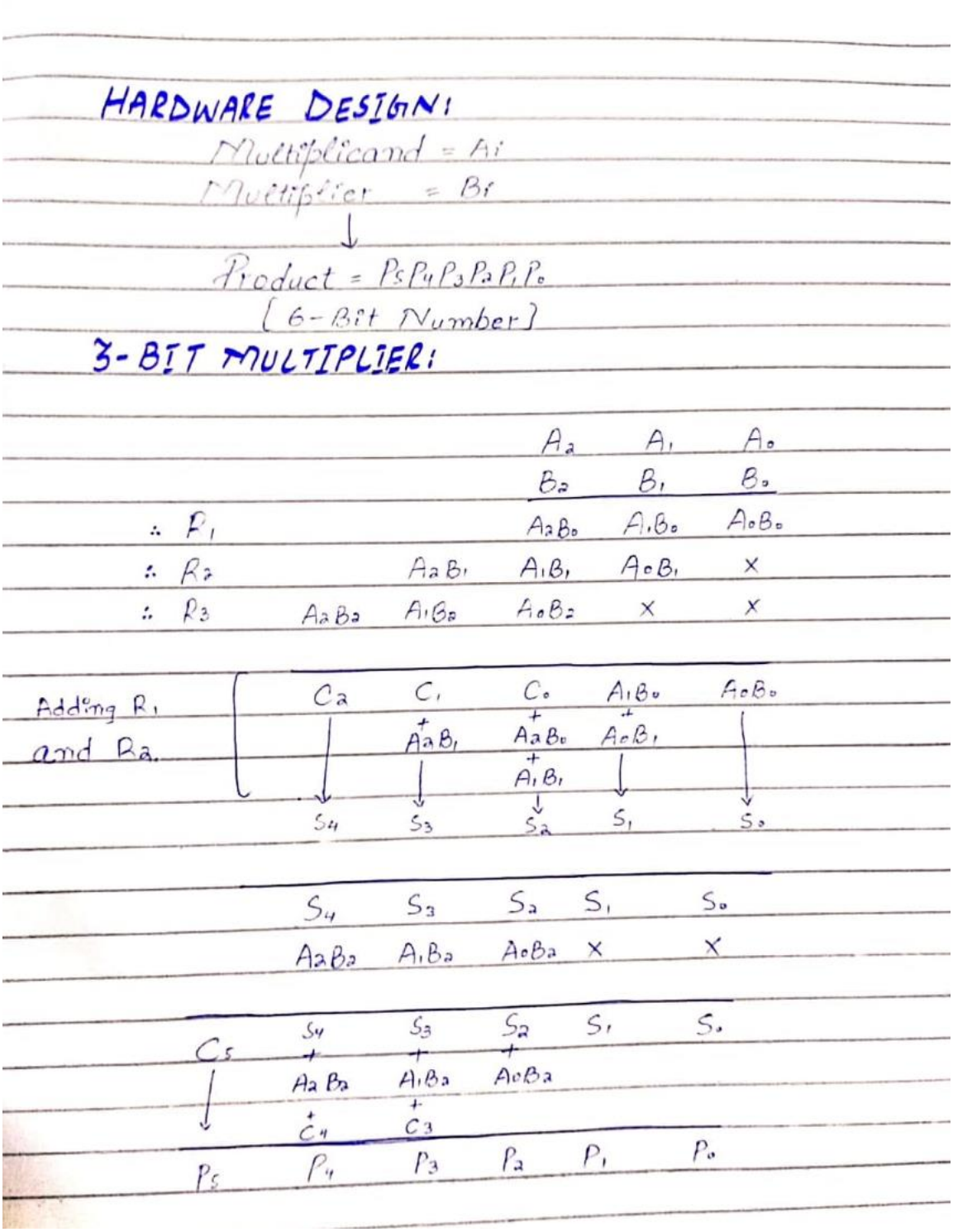
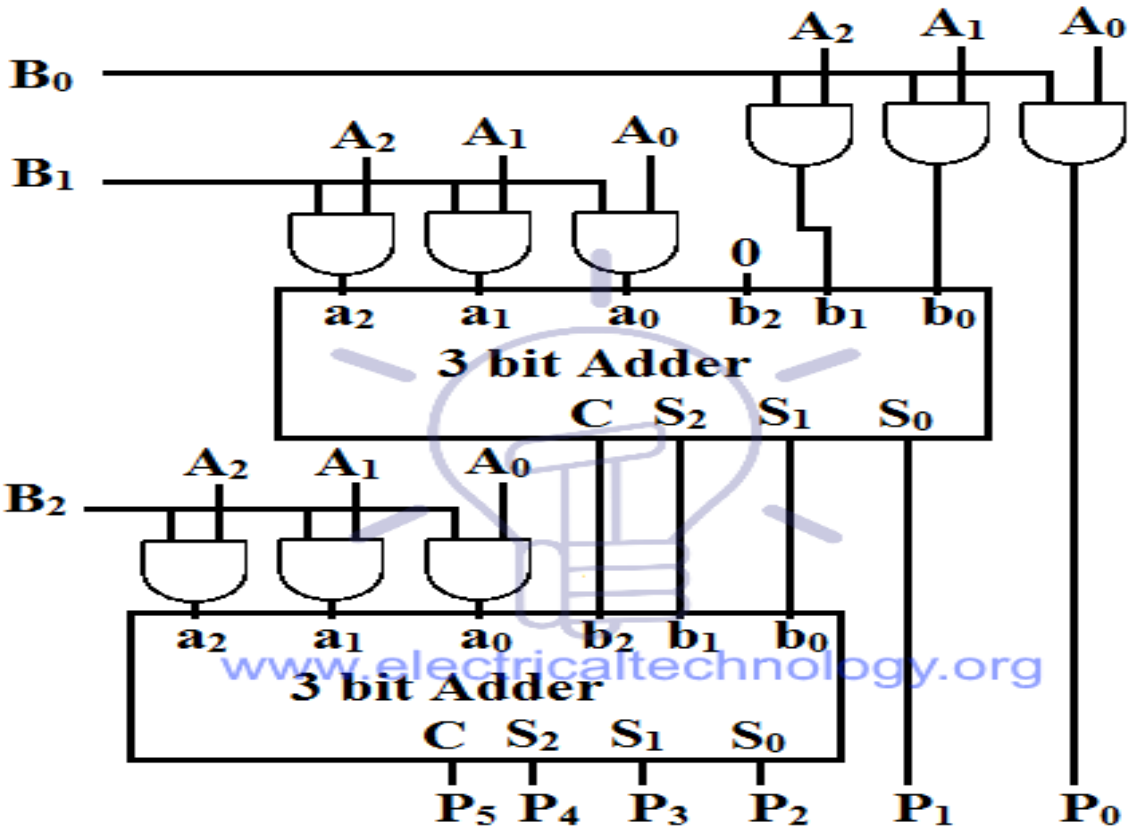
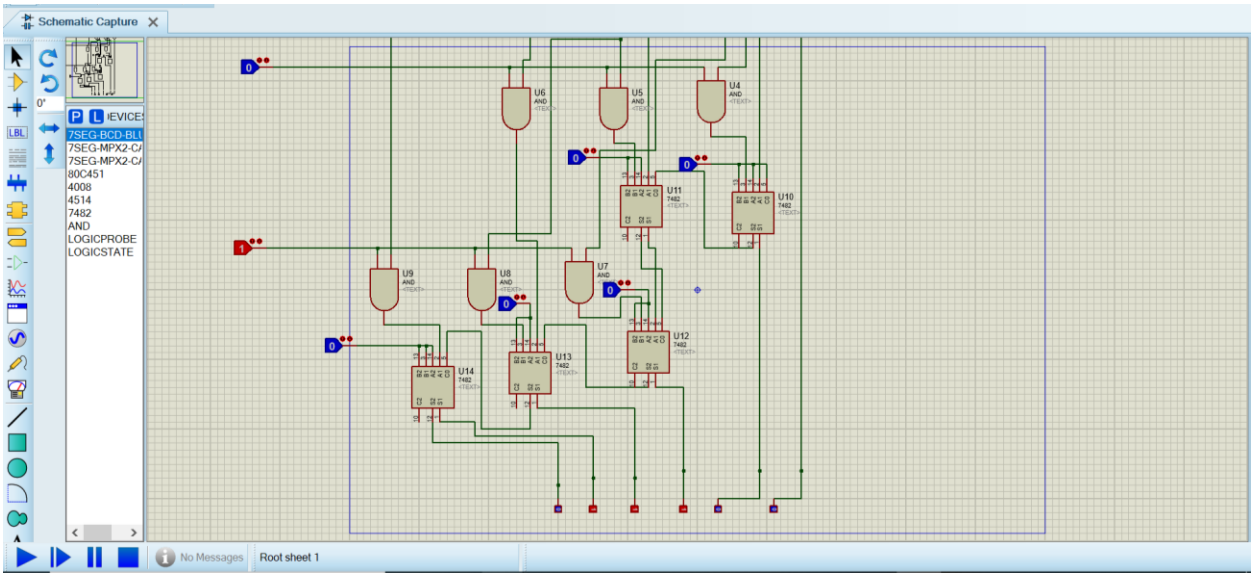


Diagram:



PROTEUS IMPLIMENTATION:



CODE:

```
module bit_multiplier(A0,A1,A2,B0,B1,B2,C0,C1,C2,C3,C4,C5);
output C0,C1,C2,C3,C4,C5;
input A0,A1,A2,B0,B1,B2;
wire w1,w2,w3,w4,w5,w6,w7,w8,w9,w11,w12,w13,w14,w15,w16,w17,w18,w19,w20,w21,w22,w23,w24;

//operation on ist order of  bits


and G1(C0,A0,B0);
and G2(w1,A1,B0);
and G3(w2,A2,B0);
//operation on 2nd order
and G4(w3,A0,B1);
and G5(w4,A1,B1);
and G6(w5,A2,B1);
// Adding firt order
xor G7(C1,w1,w3);
and G8(w6,w1,w3);
xor G9(w7,w4,w2);
and G10(w8,w4,w2);
xor G11(w9,w7,w6);
and G12(w10,w7,w6);
or G13(w11,w10,w8);
xor G14(w12,w11,w5);
and  G15(w13,w11,w15);
//for 3rd order multiplication
and G16(w14,A0,B2);
and G17(w15,A1,B2);
and G18(w16,A2,B2);
//FOR THE LAST SUM OF ALL BITS
xor G19(C2,w9,w14);
and G20(w17,w9,w14);
xor G21(w18,w12,w15);
and G22(w19,w12,w15);
xor G23(C3,w18,w17);
and G24(w20,w18,w17);
or G25(w21,w19,w20);
xor G26(w22,w16,w13);
and G27(w23,w16,w13);
xor G28(C4,w22,w21);
and G29(w24,w22,w21);
or  G30(C5,w24,w23);

endmodule

module multiplier_3x3;
wire  C0,C1,C2,C3,C4,C5;
reg A0,A1,A2,B0,B1,B2;

bit_multiplier bunny_25(A0,A1,A2,B0,B1,B2,C0,C1,C2,C3,C4,C5);
initial
begin
 A0=1'b1; A1=1'b1; A2=1'b1; B0=1'b1; B1=1'b1; B2=1'b1;
end

endmodule
```

MODELSIM IMPLEMENTATION:

```
Ln#
 1  module test_bench();
 2
 3  reg [2:0]A,B;
 4  wire [5:0]R;
 5
 6  multiplier mml(.a(A), .b(B), .p(R));
 7
 8  initial
 9  begin
10          A=3'b011; B=3'b100;
11          #50;
12          A=3'b101; B=3'b101;
13          #50;
14          A=3'b110; B=3'b110;
15          #50;
16          A=3'b111; B=3'b111;
17          #50;
18  end
19
20  endmodule
21
```

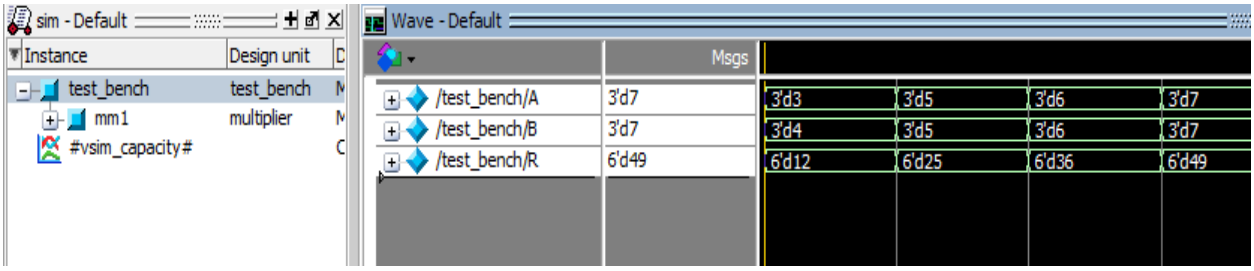
TEST BENCH

```
Ln#
 1  module multiplier (a,b,p);
 2  input [2:0]a,b;
 3
 4  wire [2:0]m0;
 5  wire [3:0]ml;
 6  wire [4:0]m2;
 7
 8  wire [5:0]sl,s2;
 9  output [5:0]p;
10
11  assign m0= {3{a[0]}} & b[2:0];
12  assign ml= {3{a[1]}} & b[2:0];
13  assign m2= {3{a[2]}} & b[2:0];
14
15  assign sl= m0 + (ml<<1);
16  assign s2= sl + (m2<<2);
17
18  assign p=s2;
19  endmodule
20
```

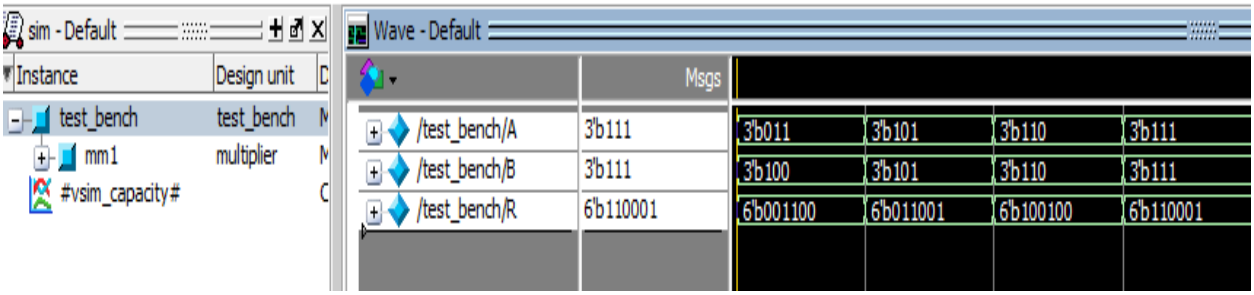
MULTIPLIER

WAVE SIMULATION:

SIMULATION RESULT AS UNSIGNED DECIMAL NUMBERS:



SIMULATION RESULT AS BINARY NUMBERS:



APPLICATIONS:

- Binary multipliers are applications like computers, mobiles, high speed calculators and some general purpose processors require binary multipliers.
- These multiplier logic circuits are implemented on integrated circuits with various pin configurations.
- These ICs are used in several applications, particularly in various microprocessors used for computers, controlling devices, calculators, mobiles, digital signal processors (DSPs), etc.

REFERENCES:

https://photoscissors.com/tutorials/online/change-background

https://bit.ly/32uPYkK