DAY #12

30 DAYS OF VERILOG

AIM – TO IMPLEMENT 4 Bit Multiplier

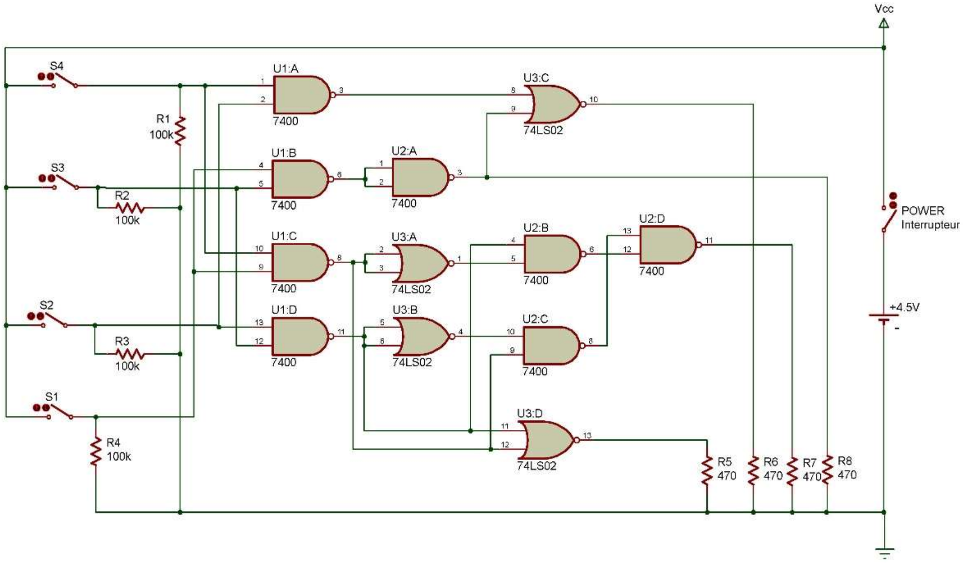
A **4-bit multiplier** is a circuit that allows you to multiply two binary numbers, each consisting of 4 bits.

1. **Functionality**:
   - A 4-bit multiplier takes two 4-bit binary numbers (the **multiplier** and the **multiplicand**) and produces an 8-bit output (the **product**).
   - The bit size of the product is equal to the sum of the bit sizes of the multiplier and multiplicand.
2. **Circuit Diagram**:
   - The circuit diagram for a 4-bit multiplier typically involves logic gates (such as AND gates, XOR gates, etc.) to perform the multiplication.
3. **Working Principle**:
   - The 4-bit multiplier processes each pair of corresponding bits from the multiplier and multiplicand.
   - It performs partial multiplications for each bit position and accumulates the results to obtain the final product.
4. **Example**:
   - Suppose we want to multiply two 4-bit binary numbers: A=1101 (multiplier) and B=1010 (multiplicand).
   - The product 'P' can be calculated as follows:

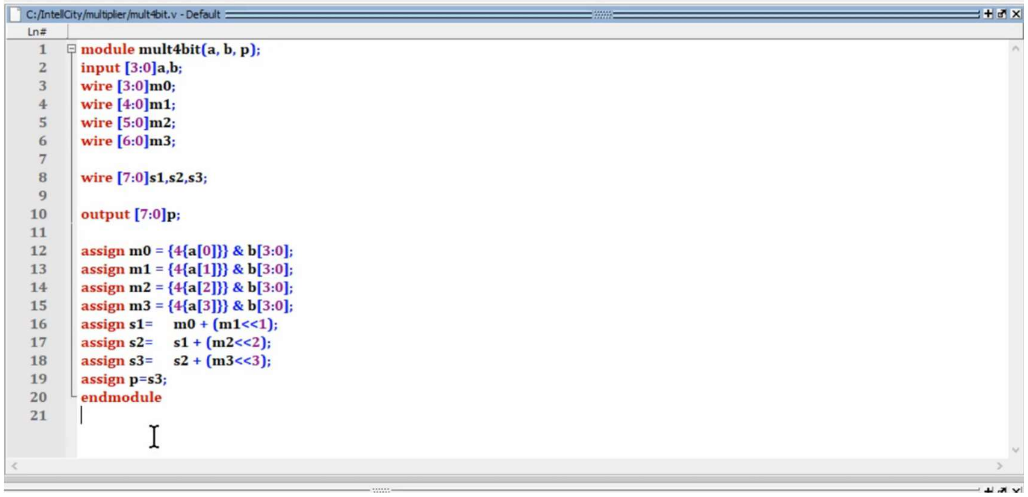
     - P = A X B
     - P = 1101 X 1010
     - P = 11110110 (in binary)
     - P = 246 (in decimal)

5. **Applications**:
   - 4-bit multipliers are commonly used in digital signal processing, communication systems, and arithmetic circuits.
   - They play a crucial role in various electronic devices and systems.

SCHEMATIC –



CODE –



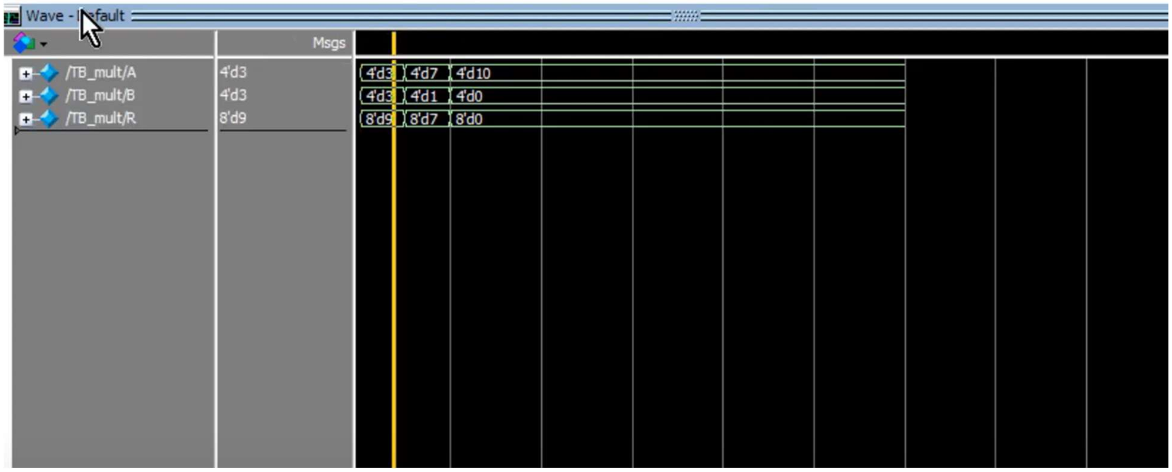
```
module mult4bit(a, b, p);
input [3:0]a,b;
wire [3:0]m0;
wire [4:0]m1;
wire [5:0]m2;
wire [6:0]m3;

wire [7:0]s1,s2,s3;

output [7:0]p;

assign m0 = {4{a[0]}} & b[3:0];
assign m1 = {4{a[1]}} & b[3:0];
assign m2 = {4{a[2]}} & b[3:0];
assign m3 = {4{a[3]}} & b[3:0];
assign s1=    m0 + (m1<<1);
assign s2=    s1 + (m2<<2);
assign s3=    s2 + (m3<<3);
assign p=s3;
endmodule
```

WAVEFORM –


Wave - Default
Msgs
/TB_mult/A
4'd3
/TB_mult/B
4'd3
/TB_mult/R
8'd9
4'd3
4'd7
4'd10
4'd3
4'd1
4'd0
8'd9
8'd7
8'd0