CPE166 Lab 2 Part 1
By: Prof. Pang

Lab 2

Part 1: 3 By 3 Binary Combinational Array Multiplier

The binary combinational multiplier diagram:

$$
\begin{array}{cccccc}
 & & X_2 & X_1 & X_0 & \\
 & \mathbf{x} & Y_2 & Y_1 & Y_0 & \\
\hline
 & & X_2Y_0 & X_1Y_0 & X_0Y_0 & \\
 & X_2Y_1 & X_1Y_1 & X_0Y_1 & & \\
X_2Y_2 & X_1Y_2 & X_0Y_2 & & & \\
\hline
P_5 & P_4 & P_3 & P_2 & P_1 & P_0
\end{array}
$$

Figure 1. 3 by 3 binary combinational multiplication diagram

This lab is to design the above multiplier by using the hardware structure shown below:

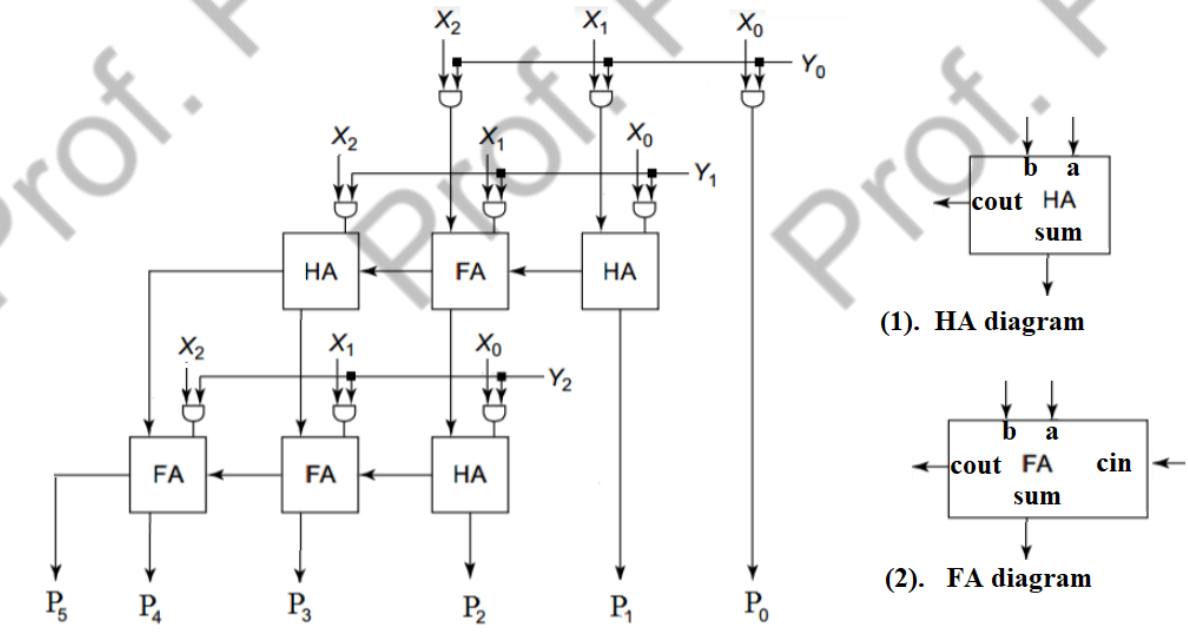


Figure 2. 3 by 3 binary combinational array multiplier hardware structure

CPE166 Lab 2 Part 1
By: Prof. Pang

Lab Procedure

Step 1. Half Adder Design

The half adder adds two 1-bit binary inputs a and b. It generates two outputs, sum signal and carry cout signal.

Inputs		Outputs	
a	b	cout	sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0
Logic equations: cout = a b sum = a ⊕ b			

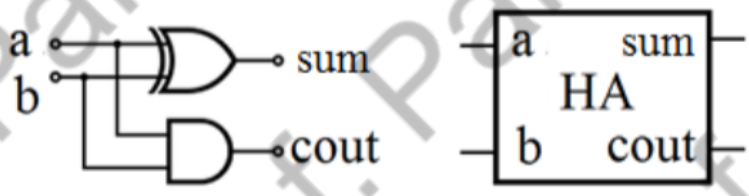


Figure 3. Half adder truth table, schematic and logic symbol

Design the above half adder circuit using Verilog, write testbench and run simulations. Verify that your simulation results are the same as the values listed in the truth table above.

CPE166 Lab 2 Part 1
By: Prof. Pang

Step 2. Full Adder Design

The full adder circuit adds three 1-bit binary inputs a, b and cin. It generates two outputs, sum signal and carry cout signal.

Inputs			Outputs	
a	b	cin	cout	sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Logic equations:

$sum = a \oplus b \oplus cin$

$cout = (a \oplus b) cin + a b$

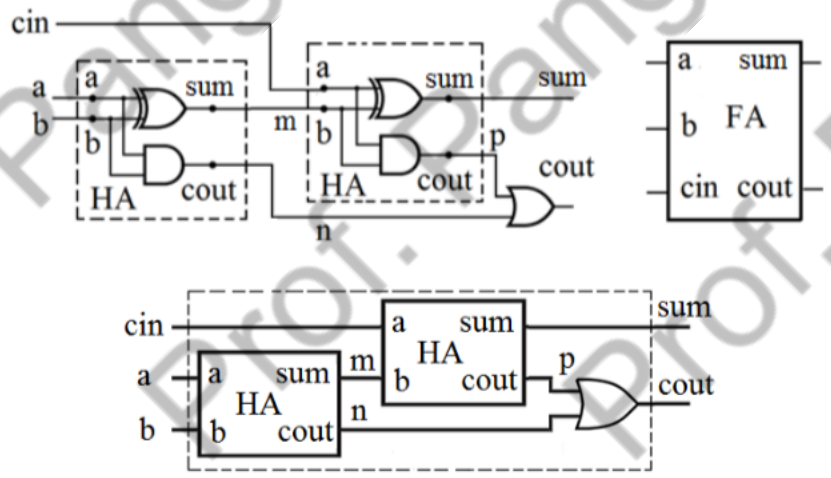


Figure 4. Full adder truth table, schematic and logic symbol

Design the above full adder circuit by using two half adder HA modules, and one OR gate in Verilog, write testbench and run simulations. Verify that your simulation results are the same as the values listed in the truth table above.

CPE166 Lab 2 Part 1
By: Prof. Pang

Step 3. Final Combinational Multiplier Design

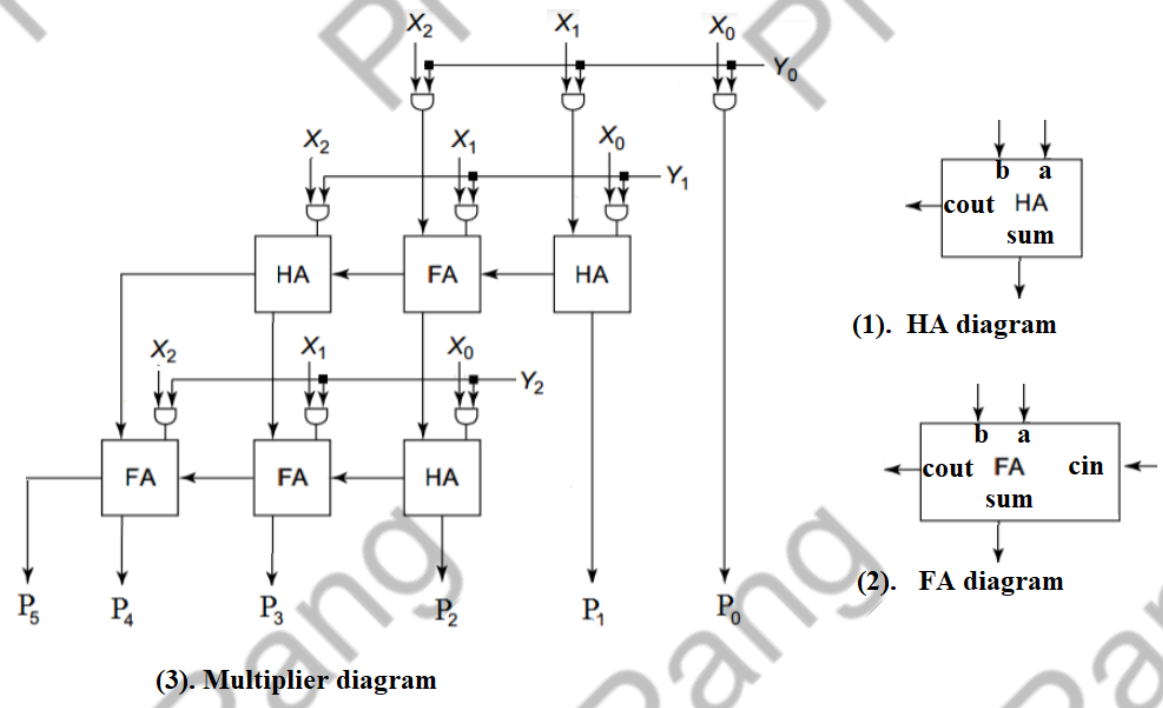


Figure 5. 3 by 3 combinational array multiplier schematic

Design the above multiplier circuit by using nine AND gates, three half adder HA modules and three full adder FA modules in Verilog, write a testbench and run simulation.

Demo Requirement

You need to demonstrate the final simulation waveform of the multiplier design to your lab instructor.

You need to download this design to the FPGA board and demonstrate it to your lab instructor.

CPE166 Lab 2 Part 1
By: Prof. Pang

Note: Before starting this experiment, all the necessary knowledge required to complete this work has been introduced in the CPE166 lecture session. The following examples are for you to refresh your learning.

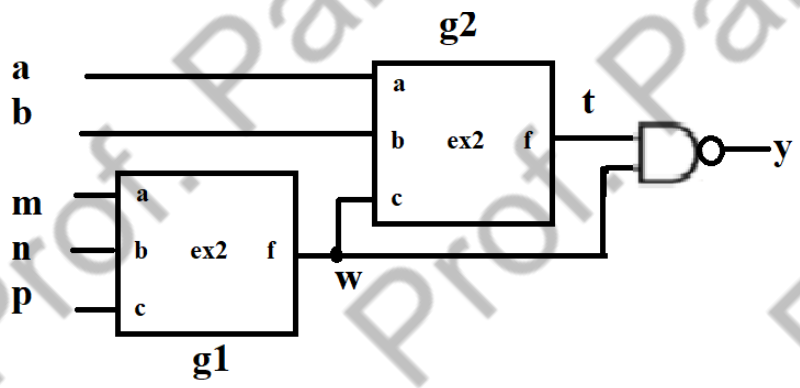
Sample Verilog Codes:

```
module ex1( a, b, f1, f2, f3, f4);

input a, b;

output f1, f2, f3, f4;

assign f1 = a ^ b;              // xor gate

assign f2 = a ~^ b;             // xnor gate

assign f3 = ~ ( a & b );        // nand gate

assign f4 = ~ (a | b);          // nor gate

endmodule


module ex2( a, b, c,  f );

input a, b, c;

wire    m, n;

output f;

assign  m  = ~a ;               // not gate

assign  n=   b & c ;            // and  gate

assign  f  =  m |  n;           // or gate

endmodule


`timescale  1ns/1ps

module ex2_tb;          //testbench for ex2 design

reg      a, b, c;

wire     f;                 // Only a, b, c and f are used for testing.

integer k;

ex2    g1(a, b, c, f);
```

CPE166 Lab 2 Part 1
By: Prof. Pang

```
initial
begin
   $monitor($time, " ns, a=%b, b=%b, c=%b, f=%b", a, b, c, f);
    for (k=0; k<8; k=k+1)
    begin
       {a, b, c} = k;
        #5;
     end
     #5  $stop;
end
endmodule
```

Sample Circuit:



Sample Implementation in Verilog:

```
module sample (a, b, m, n, p, y);
input    a, b, m, n, p;
output    y;
wire     w,  t;
ex2    g1 ( .a (m),  .b(n), .c(p), .f(w) );
ex2    g2  (.a (a),   .b(b),  .c(w), .f (t) );
assign  y =  ~ ( w & t);
endmodule
```