2-bit Multiplier

Block Diagram:

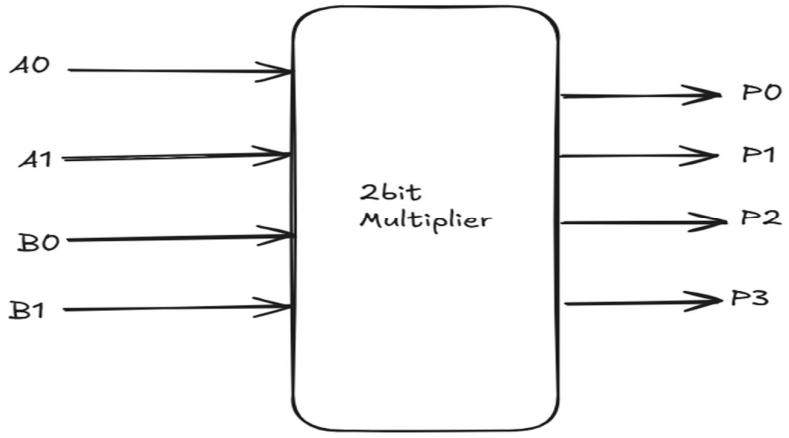

Figure 1: Block diagram

As there carries are generated at intermediary operations, thus we use half adders.

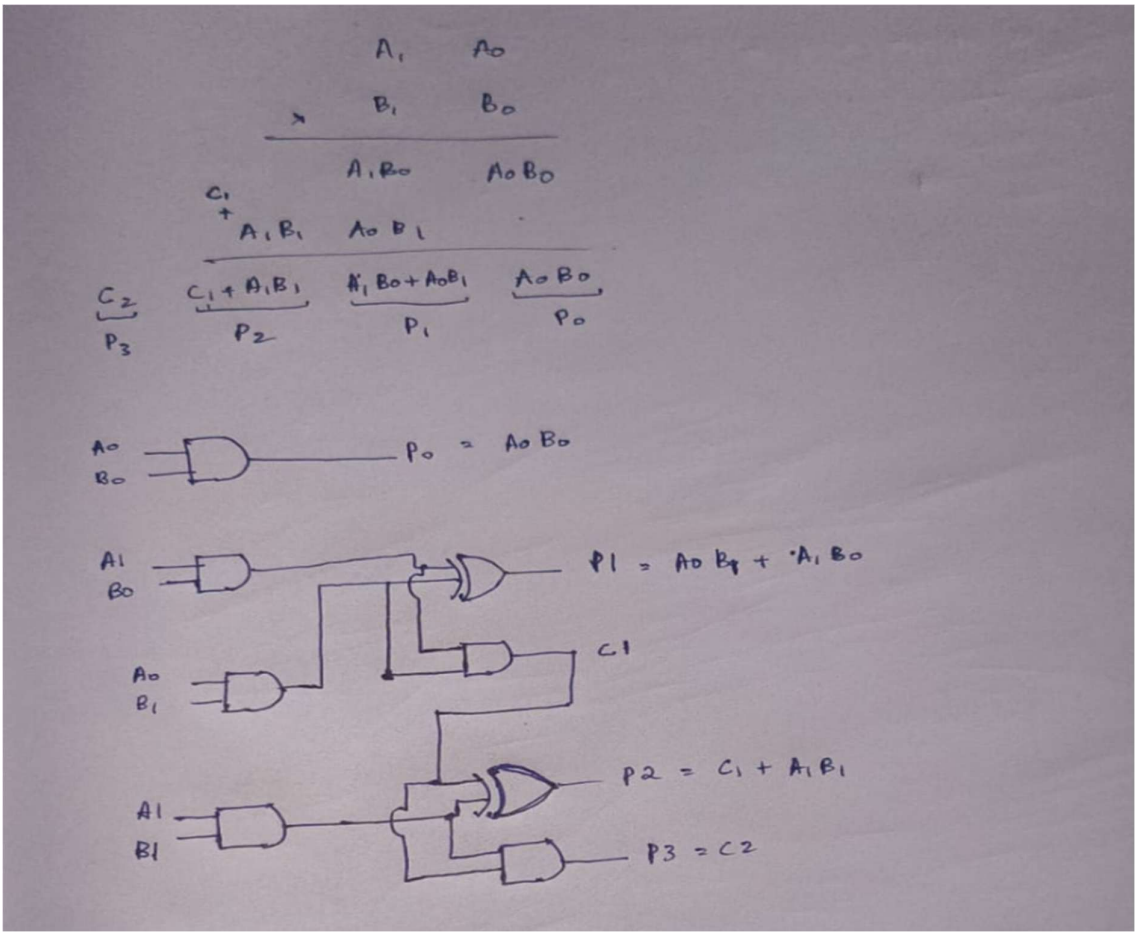

Figure 2: Implementation of 2bit Multiplier

Truthtable:

A1	A0	B1	B0	P3	P2	P1	P0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

K-maps:

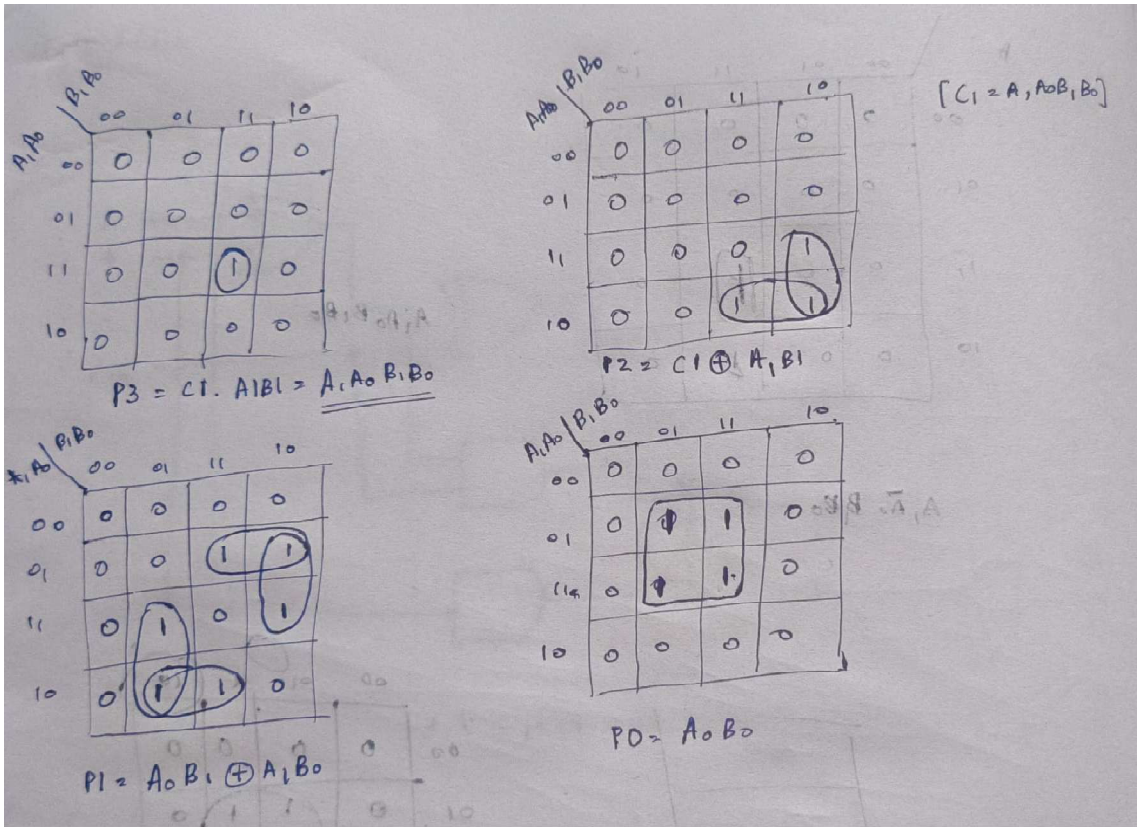

Figure 3: Kmaps for P[3],P[2],P[1],P[0]

Verilog Code:

```
// Verilog code for Half adder
module halfAdder(input A, B, output sum, cout);
assign sum = A^B;
assign cout = A&B;
endmodule

// Verilog code for 2-bit Multiplier
module twobitmultiplier(
   input A0,A1,B1,B0,
   output [3:0] P
   );
   wire X1,X2,X3,C1;
   and(X1,A1,B0);
   and(X2,A0,B1);
   and(X3,A1,B1);
   assign P[0] = A0&B0;
   halfAdder h1(.A(X1),.B(X2),.sum(P[1]),.cout(C1));
   halfAdder h2(.A(C1),.B(X3),.sum(P[2]),.cout(P[3]));
endmodule
```

Schematics:

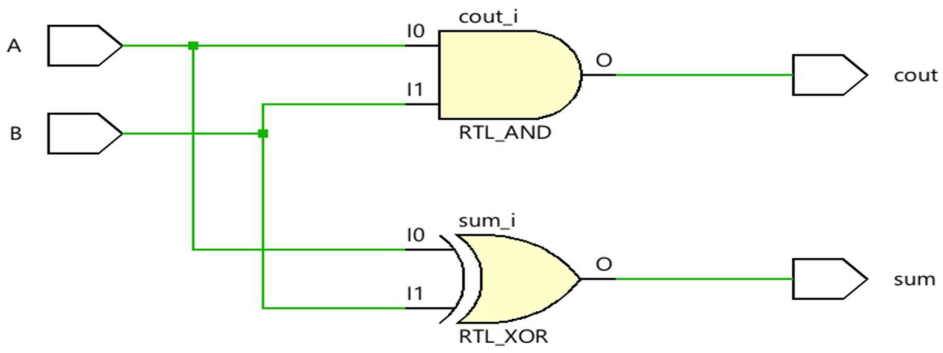

Figure 4: RTL Analysis Schematic - half adder

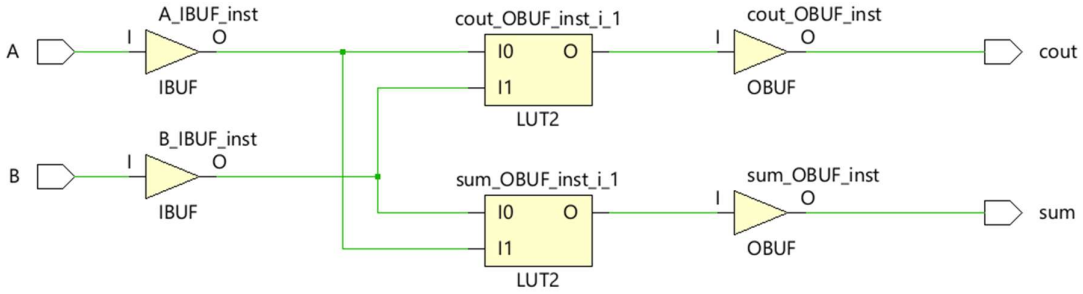

Figure 5: Synthesized schematic - half adder

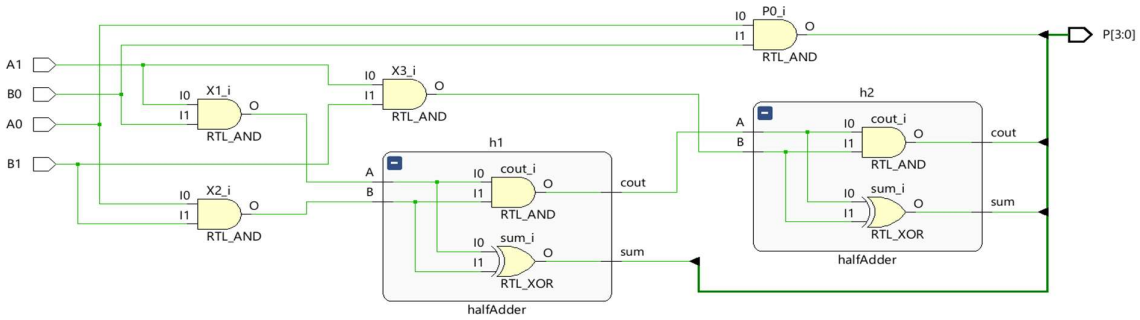

Figure 6: RTL Analysis Schematic – 2bit Multiplier

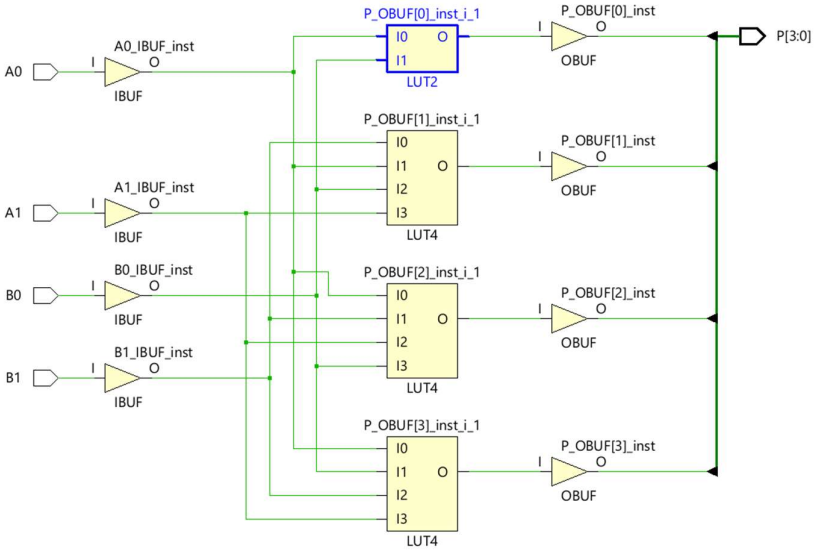

Figure 7: Synthesized schematic - 2bit Multiplier

Look Up Tables:

Cell Properties

P_OBUF[0]_inst_i_1

I1	I0	O=I0 ...
0	0	0
0	1	0
1	0	0
1	1	1

Figure 8: LUT for P[0]

Cell Properties

P_OBUF[1]_inst_i_1

I3	I2	I1	I0	O=I0 & I1 & !I3 + I0 & I1 & !I2 + !I1 & I2 & I3 + !I0 & I2 & I3
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Figure 9: LUT for P[1]

Cell Properties

P_OBUF[2]_inst_i_1

I3	I2	I1	I0	O=I1 & I2 & !I3 + !I0 & I1 & I2
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Figure 9: LUT for P[2]

Cell Properties

P_OBUF[3]_inst_i_1

I3	I2	I1	I0	O=I0 & I1 & I2 & I3
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Figure 10: LUT for P[3]