4-BIT ARRAY MULTIPLIER

AIM :-

To Design & Implement 4- BIT ARRAY MULTIPLIER using ADDER CIRCUIT Program using Verilog HDL.

TOOL:-

Xilinx ISE 9.2i Version

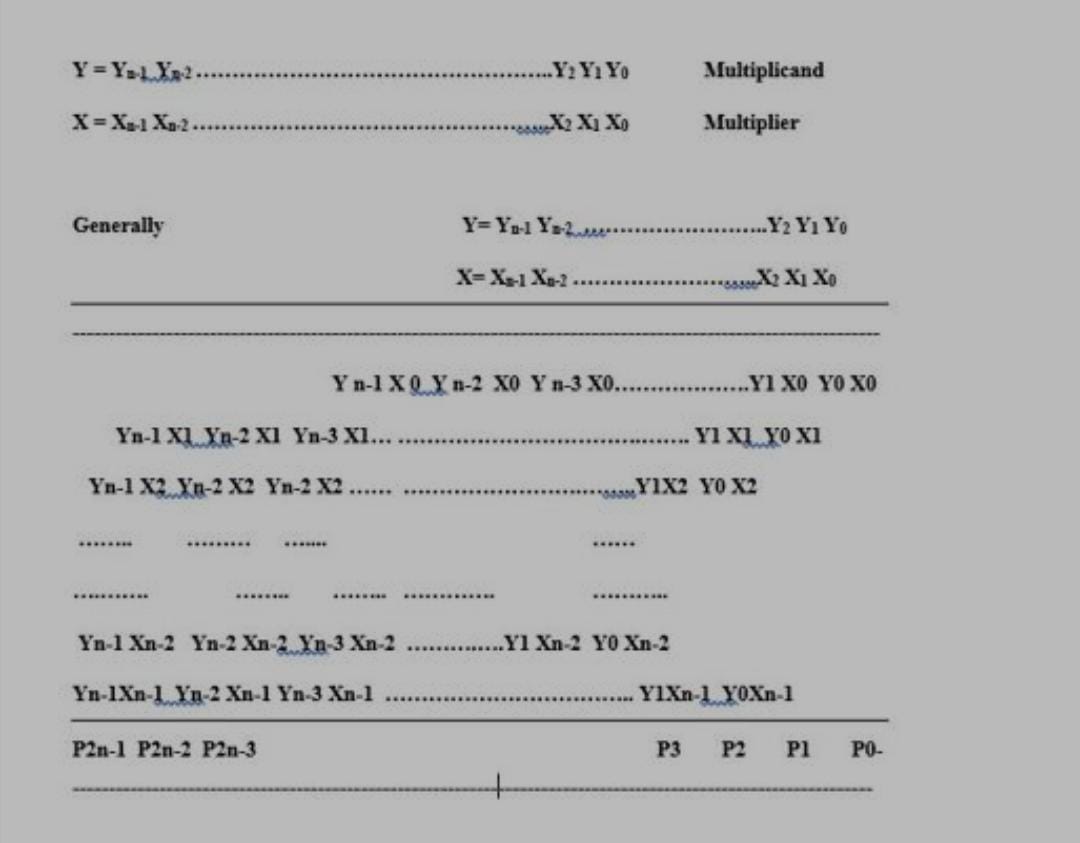
|  |  |
| --- | --- |
| FAMILY | SPARTAN 3 |
| Device | XC3S400 |
| Package | PQ208 |
| Speed | -4/-5 |
| Synthesis | XST(VERILOG/VHDL) |
| Simulator | ISE Simulator |

**Multiplication Algorithm** :-

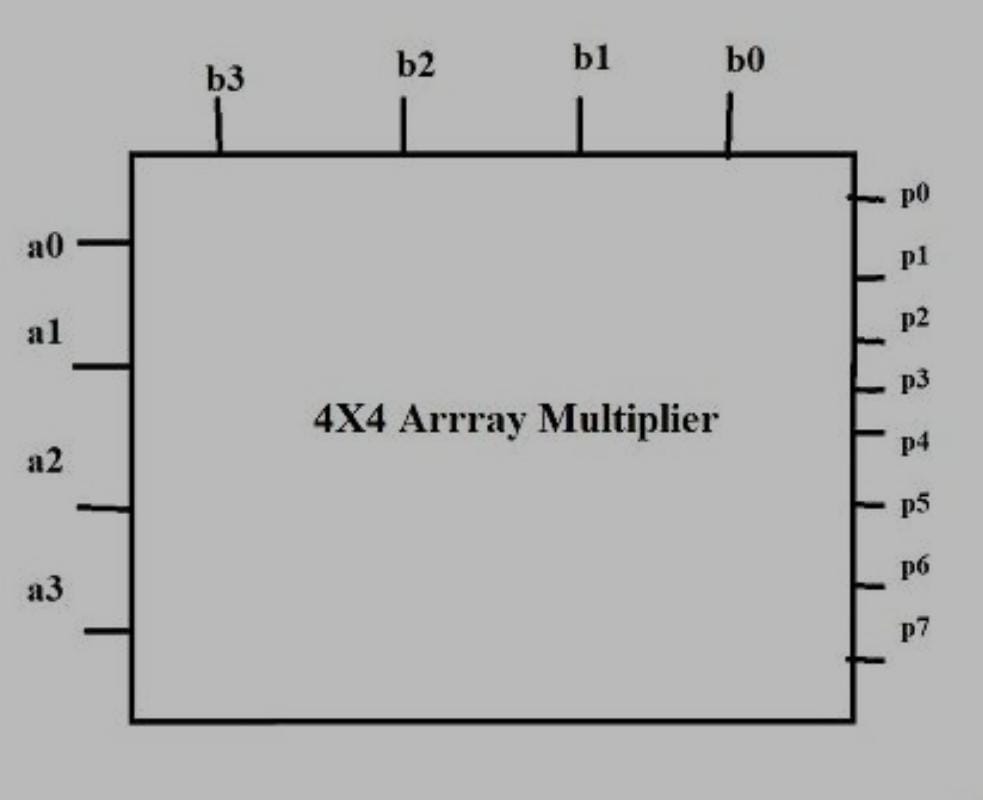
Multiplication process has three main steps:

* Partial product generation
* Partial product reduction
* Final addition.

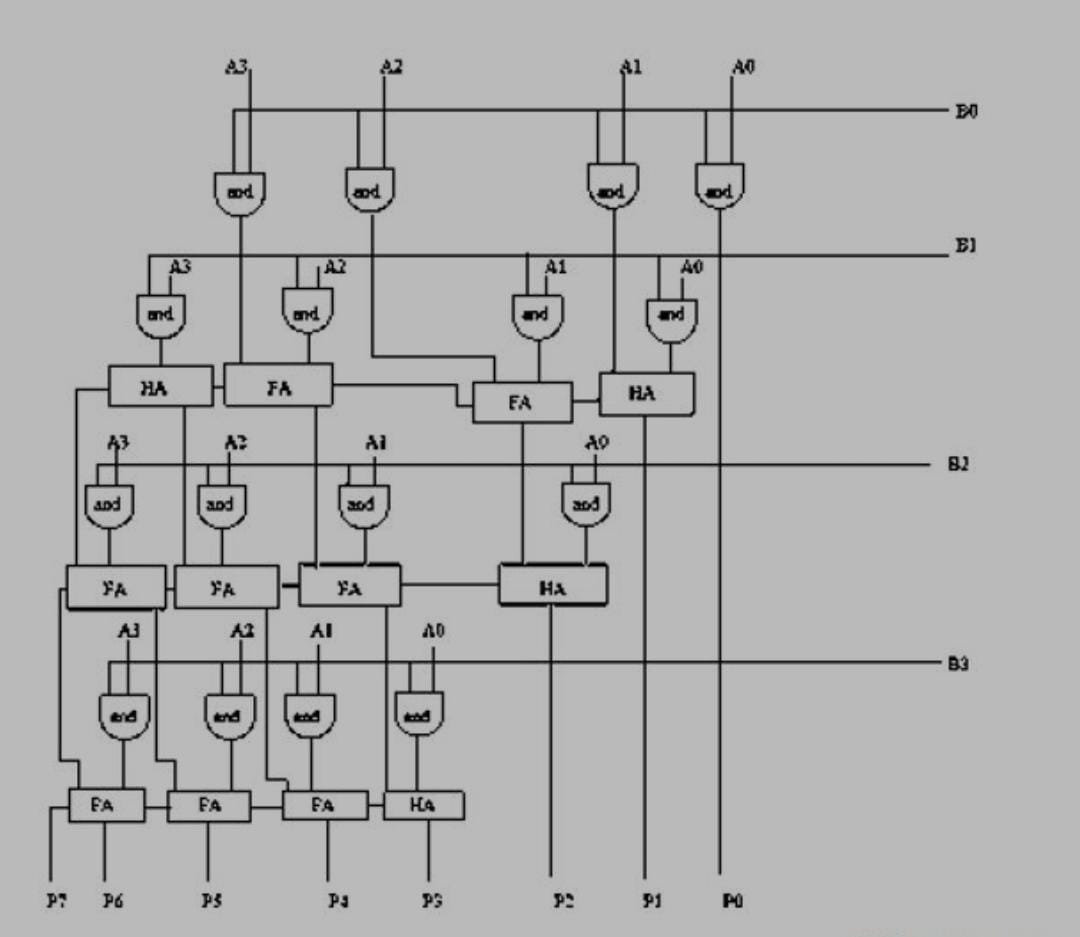
The common multiplication method is the “add and shift “algorithm. The multiplication algorithm for an N-bit multiplier is shown below.



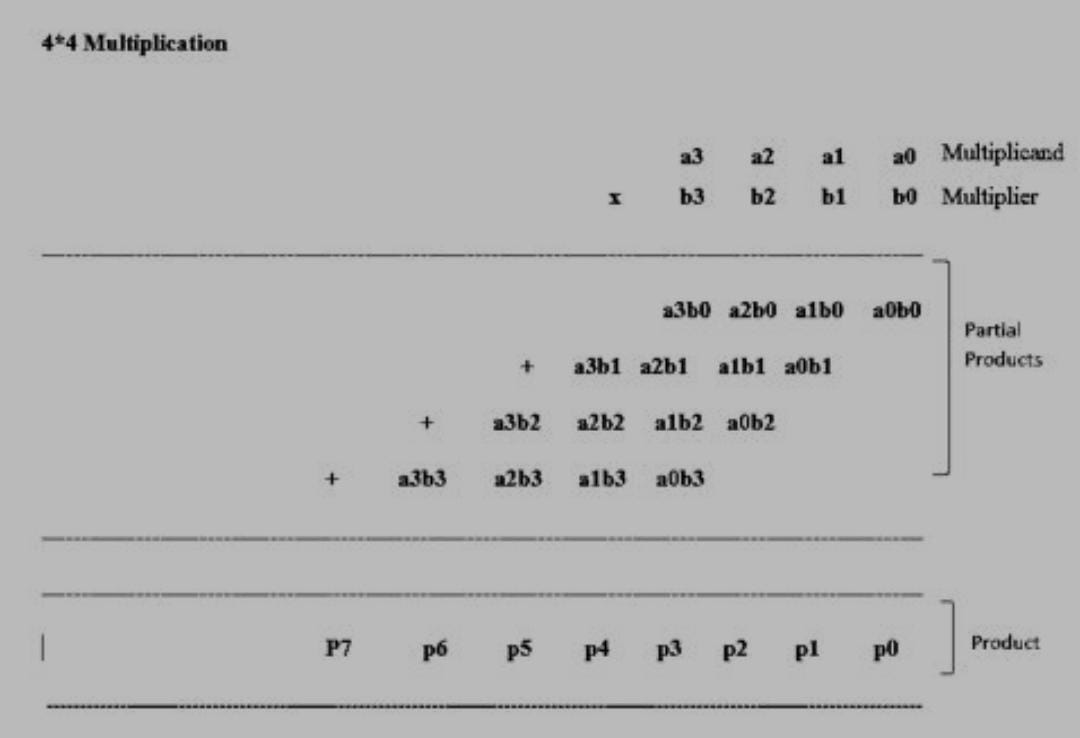
ARCHITECTURE :-



BLOCK DIAGRAM :-



LOGIC :-



SOURCE CODE :-

module Arraymultiplier4bit(product,A,B);

input [3:0]A;

input [3:0]B;

output [7:0]product;

assign product[0]=(A[0]&B[0]);

wire c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16,c17;

HA HA1(product[1],c1,(A[1]& B[0]),(A[0]&B[1]));

FA FA1(c2,c3,A[1]&B[1],(B[0]&A[2]),c1);

FA FA2(c4,c5,(B[1]&A[2]),(B[0]&A[3]),c3);

HA HA2(c6,c7,(B[1]&A[3]),c5);

HA HA3(product[2],c15,c2,(B[2]&A[0]));

FA FA5(c14,c16,c4,(B[2]&A[1]),c15);

FA FA4(c13,c17,c6,(A[2]&B[2]),c16);

FA FA3(c9,c8,c7,(B[2]&A[3]),c17);

HA HA4(product[3],c12,c14,(B[3]&A[0]));

FA FA8(product[4],c11,c13,(B[3]&A[1]),c12);

FA FA7(product[5],c10,c9,(B[3]&A[2]),c11);

FA FA6(product[6],product[7],c8,(A[3]& B[3]),c10);

endmodule

module HA(sout,cout,a,b);

output sout,cout;

input a,b;

assign sout = a^b;

assign cout=(a&b);

endmodule

module FA(sout,cout,a,b,cin);

output sout,cout;

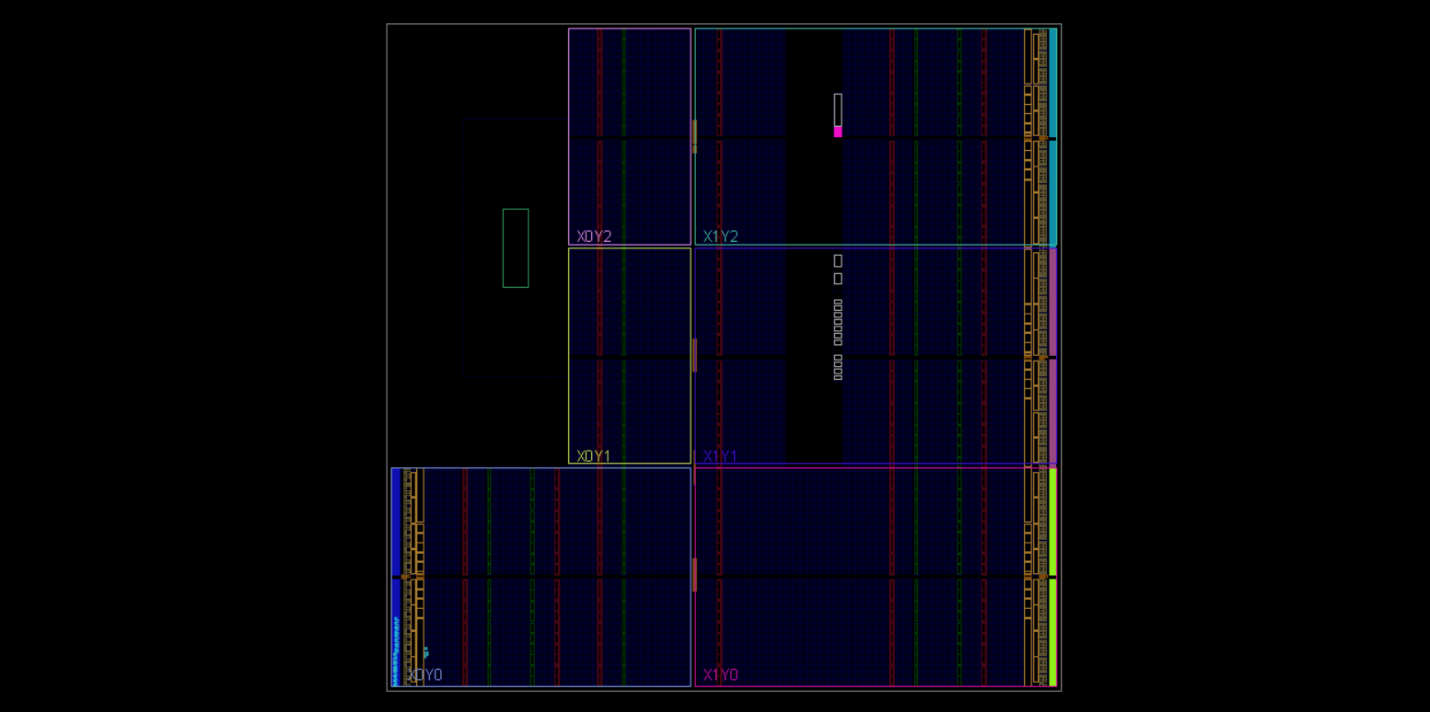
input a,b,cin;

assign sout = (a^b^cin);

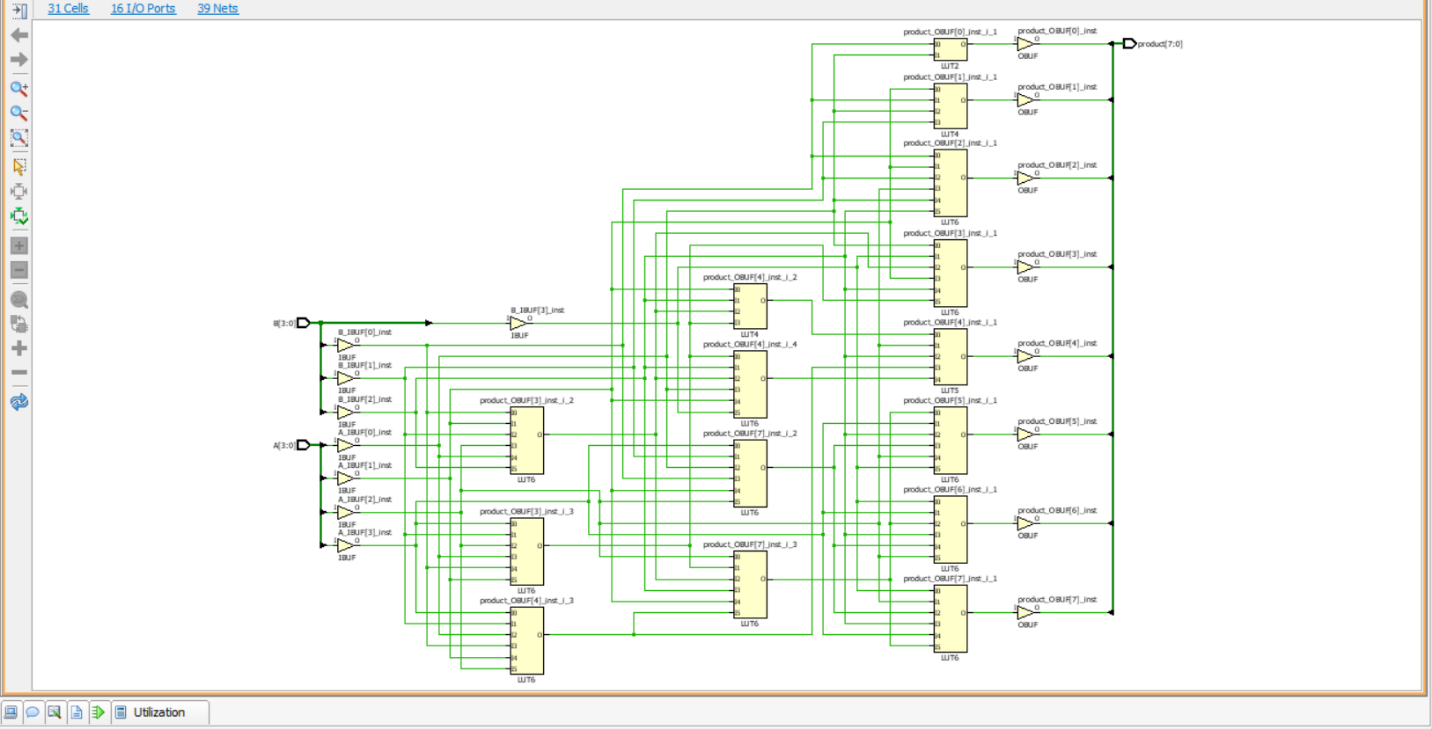
assign cout=((a&b)|(a&cin)|(b&cin));

endmodule

SYNTHESISED CIRCUIT DESIGN :-

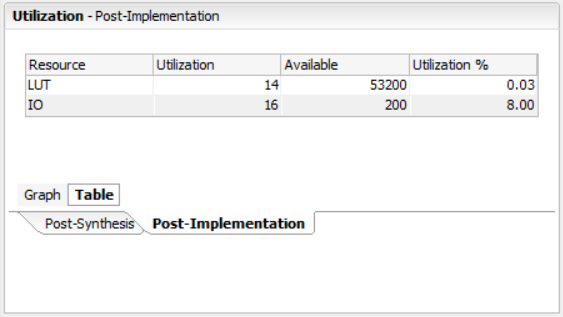


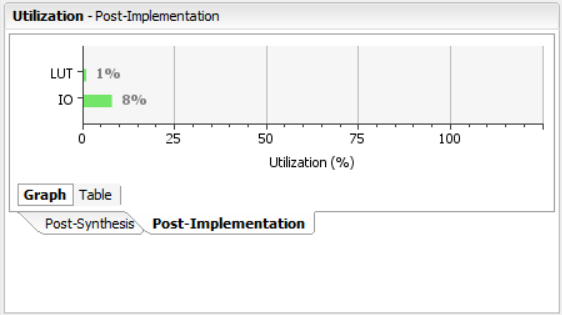
ZED BOARD DESIGN DIAGRAM



SYNTHESIS DIAGRAM

DEVICE UTILIZATION :-



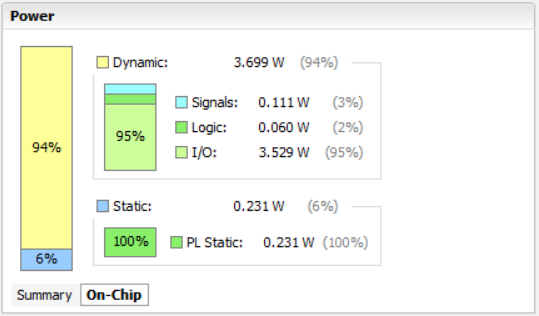


TIMING SUMMARY:-

* COMBINATIONAL DELAYS :-



DEVICE POWER UTILISATION :-



CONCLUSION :-

The 4 – bit array multiplier was designed & Implemented using Verilog HDL