**Sou. Sushila Danchand Ghodawat Charitable Trust’s**

**Sanjay Ghodawat Group of Institutions, Atigre**



**Sanjay Ghodawat Institute**

**SEMESTER : VI ACADEMIC YEAR: 2016 - 2017**

**DEPARTMENT: Electronics and Telecommunication.**

**CLASS : Third Year [B Div]**

# PROJECT NAME: 5 Bit Array Multiplier

# Project By:

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Name** | **Roll No.** |
| 1. | Likhita S. Reddy | 15 |
| 2. | Shreeya S. Bidkar | 80 |
| 3. | Saishruti S. Kore | 14 |
| 4. | Sawali J. Sawant | 27 |

Prof. Sharanabasappa.R.R Dr. Mrs. Chougule.S.R

(Project In charge) (H.O.D, E&TC)

**1.INTRODUCTION:**

Multipliers play an important role in today’s digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets – high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

The common multiplication method is “add and shift” algorithm. In parallel multipliers

number of partial products to be added is the main parameter that determines the performance of

the multiplier. To reduce the number of partial products to be added, Modified Booth algorithm

is one of the most popular algorithms. To achieve speed improvements Wallace Tree algorithm

can be used to reduce the number of sequential adding stages. Further by combining both

Modified Booth algorithm and Wallace Tree technique we can see advantage of both algorithms

in one multiplier. However with increasing parallelism, the amount of shifts between the partial

products and intermediate sums to be added will increase which may result in reduced speed,

increase in silicon area due to irregularity of structure and also increased power consumption due

to increase in interconnect resulting from complex routing.

On the other hand “serial-parallel”multipliers compromise speed to achieve better performance for area and power consumption The selection of a parallel or serial multiplier actually depends on the nature of application. In this lecture we introduce the multiplication algorithms and architecture and compare them in terms of speed, area, power and combination of these metrics.

Array Multipliers :

Array multiplier is well known due to its regular structure. Multiplier circuit is based on add shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit.

The partial product are shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder. N-1 adders are required where N is the multiplier length.

**2.PROGRAM:**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.NUMERIC\_STD.ALL;

entity arraymul is

Port ( x,y : in bit\_VECTOR (4 downto 0);

p : out bit\_VECTOR (9 downto 0));

end arraymul;

architecture Behavioral of arraymul is

signal c1,c2,c3,c4:bit\_vector(4 downto 0);

signal s1,s2,s3,s4:bit\_vector(4 downto 0);

signal xy0,xy1,xy2,xy3,xy4:bit\_vector (4 downto 0);

component fulladder

Port ( x,y,cin : in bit;

cout,sum: out bit);

end component;

component halfadder

Port ( x,y : in bit;

cout,sum : out bit);

end component;

begin

xy0(0) <= x(0) and y(0); xy1(0) <= x(0) and y(1);

xy0(1) <= x(1) and y(0); xy1(1) <= x(1) and y(1);

xy0(2) <= x(2) and y(0); xy1(2) <= x(2) and y(1);

xy0(3) <= x(3) and y(0); xy1(3) <= x(3) and y(1);

xy0(4) <= x(4) and y(0); xy1(4) <= x(4) and y(1);

xy2(0) <= x(0) and y(2); xy3(0) <= x(0) and y(3);

xy2(1) <= x(1) and y(2); xy3(1) <= x(1) and y(3);

xy2(2) <= x(2) and y(2); xy3(2) <= x(2) and y(3);

xy2(3) <= x(3) and y(2); xy3(3) <= x(3) and y(3);

xy2(4) <= x(4) and y(2); xy3(4) <= x(4) and y(3);

xy4(0) <= x(0) and y(4);

xy4(1) <= x(1) and y(4);

xy4(2) <= x(2) and y(4);

xy4(3) <= x(3) and y(4);

xy4(4) <= x(4) and y(4);

p(0) <=xy0(0);

p(1) <=s1(0);

p(2) <=s2(0);

p(3) <=s3(0);

p(4) <=s4(0);

p(5) <=s4(1);

p(6) <=s4(2);

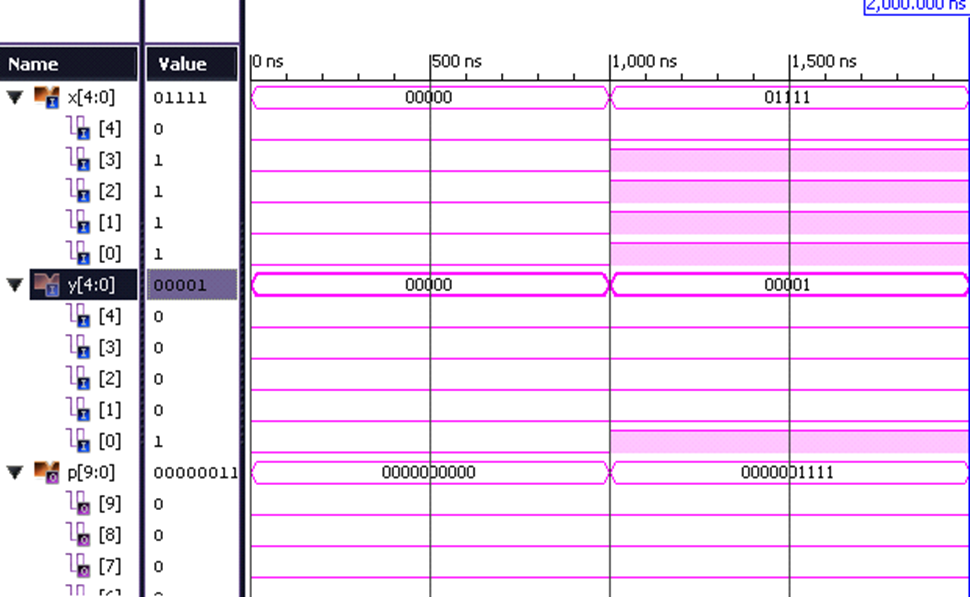
p(7) <=s4(3);

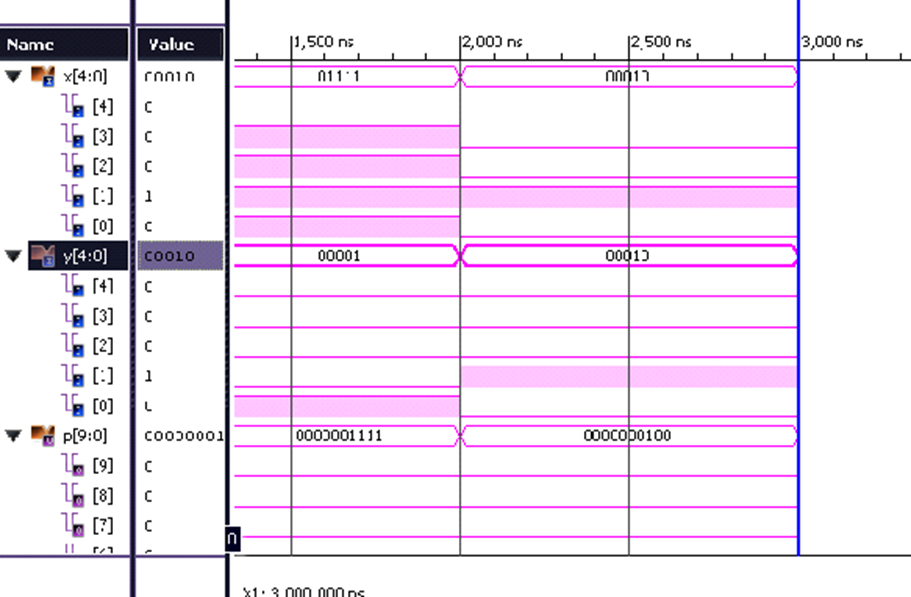
p(8) <=s4(4);

p(9) <=c4(4);

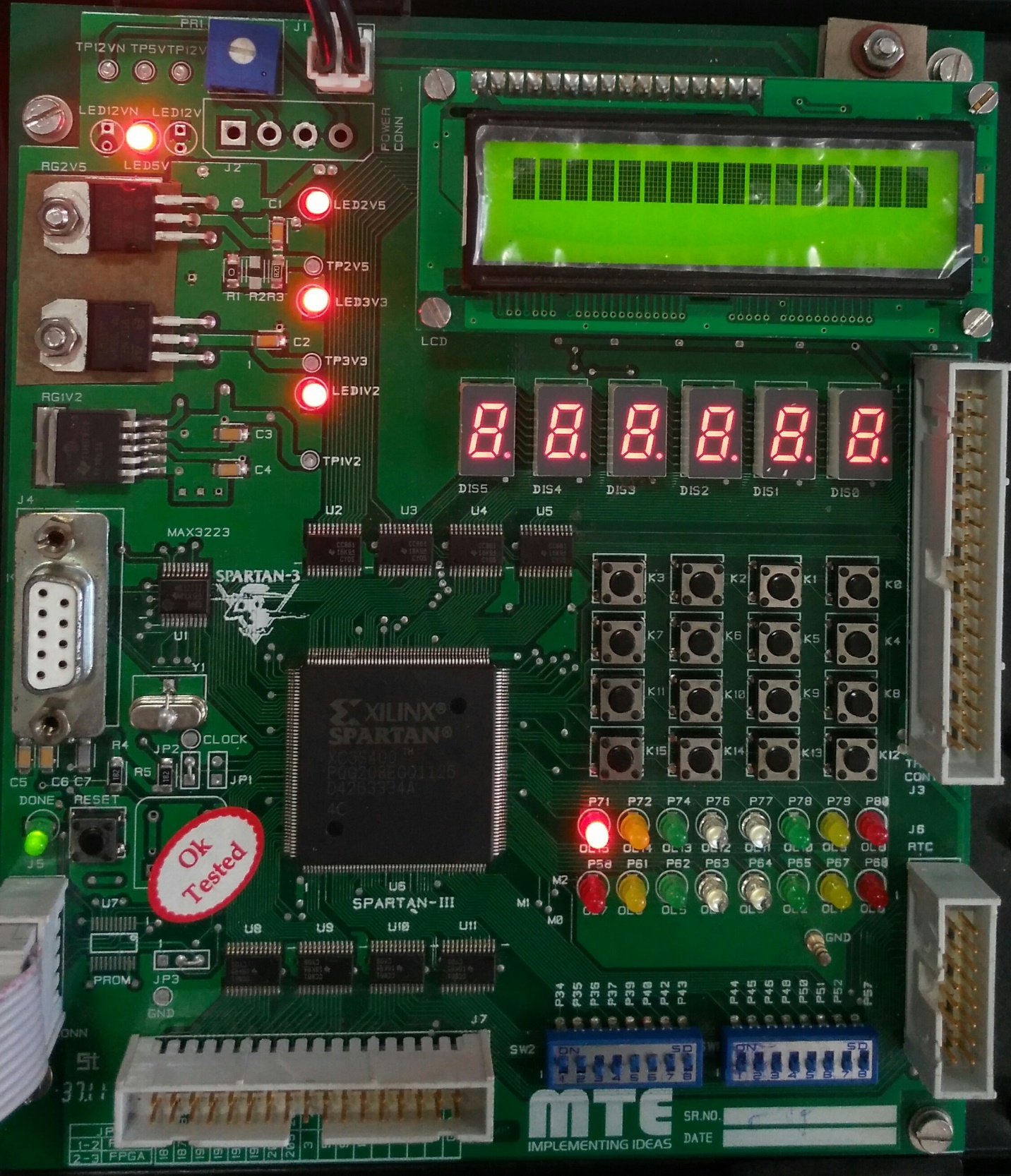
end Behavioral;

3.simulation output:

****

****

4.hardware implementation:



**5.ADVANTAGE:**

1. An array multiplier─ a multiplication method in which an array of identical cells generates new partial product and accumulation of it at the same time.

2. We can use pipelines at each level

• Result from the adder can be latched at each level and used as input for next level adder

circuit.

3. The delay is logarithmically proportional to the bit size of multiplicand and multiplier if we use the high speed array multiplier circuit

**6.DISADVANTAGE:**

1. Large number of logic gates required to design an array multiplier.

2. To design array multiplier we need many logic gates.

3. The design is complicated.

7.CONCLUSION:

• multiplication circuit becomes fast by array

multiplier.

• a multiplication method in which an array

of identical cells generates new partial

product and accumulation of it at the same

time.