Available online at http://www.bretj.com

International Journal of Current Engineering Sciences

ISSN: 2320-9046

RESEARCH ARTICLE

International Journal of Current Engineering Sciences - Vol.5, Issue, 3, pp. 4-7, March, 2015

IMPLEMENTATION OF MULTIPLIER BASED ON VERTICAL AND CROSSWIRE ALGORITHM USING KOGGE STONE ADDER

Priscilla Jennifer. J, Sivasankari. Y, Shalini. R , Sathya. R and G.Sivajanane

Department of Electronics and Communication Engineering Sri Manakula Vinayagar Engineering College, Pondicherry, India

A R T I C L E   I N F O

Article History

Received 8th, February, 2015
Received in revised form 20th, February, 2015
Accepted 10th, March, 2015
Published online 28th, March, 2015

Key words:

FPGA, Kogge stone adder, Quartus II, VHDL.

A B S T R A C T

Multipliers play a very important role in today's arithmetic operations. The existing 8X8 multiplier architecture using Vertical and Crosswire Algorithm consists of 4X4 multipliers along with ripple carry adder for addition. The ripple carry adder poses challenge in terms of propagation delay due to its consecutive addition. This project ensures reduction in propagation delay in comparison with the existing multiplier architecture. We are proposing an architecture which makes use of Kogge stone adder instead of carry ripple adder reduces the delay of the multiplier to a further extent. The architecture is to be modelled using VHDL (Very High speed integrated circuit Hardware Description Language) in Altera Quartus II software environment and implemented in Altera Acex FPGA. This multiplier finds application in Digital Signal Processing (DSP) and in Arithmetic and Logic Unit (ALU) of microprocessor and microcontrollers.

INTRODUCTION

A Binary Multiplier is an electronic circuit used in Digital electronics such as computers, mobile phones which is used to multiply two binary numbers .It is built using binary adders. A variety of computer arithmetic techniques can be used to implement a digital multiplier. Digital multipliers are indispensable in the hardware implementation of many important functions such as fast Fourier transforms (FFTs) and Multiply and Accumulate (MAC).This has made them the core components of all the Digital Signal Processor(DSP) [1]. In general there are two multiplication algorithm they are array multiplication and booth multiplication algorithm. The array multipliers take less computation time because the partial products are calculated independently in parallel. Booth multiplication is another important multiplication algorithm. Large Booth arrays having large partial sum and partial carry registers are required for high speed multiplication and exponential operations. The objective of this paper is to design and implement a delay efficient architecture of 8X8 multiplier using 4X4 multipliers as building blocks, along with adders units.

8X8 Multiplier

```
          1111 0000
        X 1000 0111

          0110 1001        = 1111 X 0111
        0111 1000          = 1111 X 1000
      0000 0000            = 0000 X 0111
    0000 0000              = 0000 X 1000
-------------------------------------------
  0111011000100000
```

Now the 8 bit binary multiplication of two numbers is considered. The multiplicand and multiplier are 11110000_2 and 10000111_2. These correspond to the decimal numbers 224 and 135 respectively.

As a check,we see that 224 X 135 = 30,240 , which is equal to 0111011000100000_2.

The hardware implementation follows directly from this observation. It requires four 4X4 multipliers and three ripple carry adders. In this paper we are using Kogge Stone adder by replacing ripple carry adder for reducing the delay level of the 8X8 multiplier.

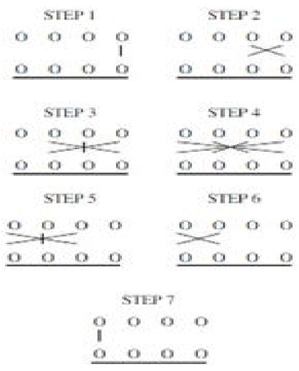
Vertical and Crosswire Algorithm

We use this Vertical and Crosswire Algorithm to binary systems as it strikes a different in the process of multiplication In particular, we develop an efficient 4X4 digital multiplier that calculates the partial products in parallel and hence the computation time involved is less. The architecture of the designed multiplier comes out to be very similar to that of the popular array multiplier and hence it should be noted that the computational delay is less like in array multiplier.

The Vertical and crosswire multiplication is explained below and the corresponding equations are derived. This concept of multiplication is implemented in 4X4 multiplier.Fig.1.which is the basic building block of 8X8 multiplier which reduces both the area and delay of the 8X8 multiplier architecture. But still the delay can be

Corresponding author:* **Priscilla Jennifer. J
Department of Electronics and Communication Engineering Sri Manakula Vinayagar Engineering College, Pondicherry, India

reduced further by using Kogge Stone adder which is been discussed in this paper.



For inputs a_3 a_2 a_1 a_0 and b_3 b_2 b_1 b_0:
$P_0 = a_0b_0$
$C_0P_1= a_0b_1+a_1b_0$
$C_1P_2= C_0+a_0b_2+a_2b_0+a_1b_1$
$C_2P_3= C_1+a_3b_0+a_0b_3+a_1b_2+a_2b_1+C_{00}$
$C_3P_4= C_2+a_3b_1+a_1b_3+a_2b_2+C_{01}+C_{10}$
$C_4P_5= C_3+a_3b_2+a_2b_3+C_{11}+C_{20}$
$C_5P_6= C_4+a_3b_3+C_{21}$

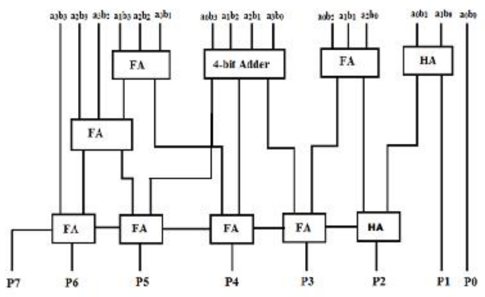


Fig. 1 4X4 multiplier using vertical and crosswire algorithm

Existing architecture of 8x8 Multiplier

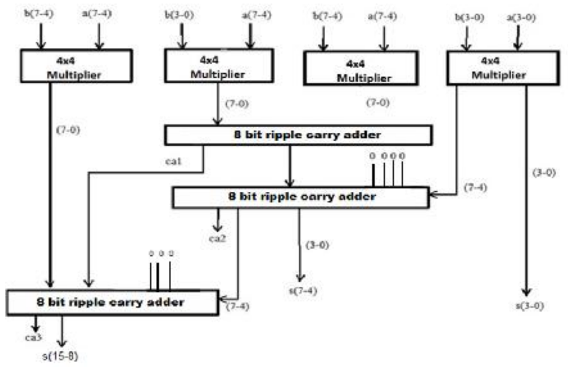


Fig. 2 8X8 multiplier architecture using ripple carry adder

Thus in the existing 8X8 multiplier Fig.2, there are four 4X4 multiplier which is implemented by using a vertical and crosswire algorithm and three ripple carry adders which has the disadvantage.The ripple carry adder experiences propagational delay as the output has to wait for the previous carry and thus it adds to the delay of the overall architecture.

Let's analyze 8X8 multiplications, say
A = A7 A6 A5 A4 A3 A2 A1 A0 and
B = B7 B6 B5 B4 B3 B2 B1 B0.

The output line for the multiplication result will be of 16 bits as – S15 S14 S13 S12 S11 S10 S9 S8 S7 S6 S5 S4 S3 S2 S1 S0.

Let's divide A and B into two parts, say the 8 bit multiplicand A can be decomposed into pair of 4 bits AH-AL. Similarly multiplicand B can be decomposed into BH-BL. Using the fundamental of Vertical and crosswire multiplication, taking four bits at a time and using 4 bit multiplier block as discussed we can perform the multiplication. The outputs of 4X4 bit multipliers are added accordingly to obtain the final product. Here three 8 bit Ripple-Carry Adders are required as shown in Fig.2.

Proposed Architecture

In this proposed system Fig.3.,we are using kogge stone adder by replacing ripple carry adder. In the ripple carry adder propogational delay is present, in order to reduce this delay we are going for a different type of adder called the kogge stone adder.

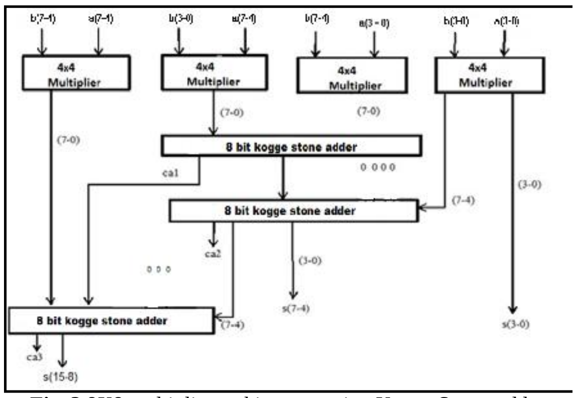


Fig 3 8X8 multiplier architecture using Kogge Stone adder

8 Bit Kogge Stone Adder

KSA Fig.4. is a parallel prefix form of carry look ahead adder. It generates carry in O (log n) time and is considered as the fastest adder and is widely used in the industry for high performance arithmetic circuits. In KSA, carries are computed fast by computing them in parallel at the cost of increased area. The Kogge-Stone adder has low logic depth, high node count, and minimal fan out. While a high node count implies a larger area, the low logic depth and minimal fan out allow for faster performance.

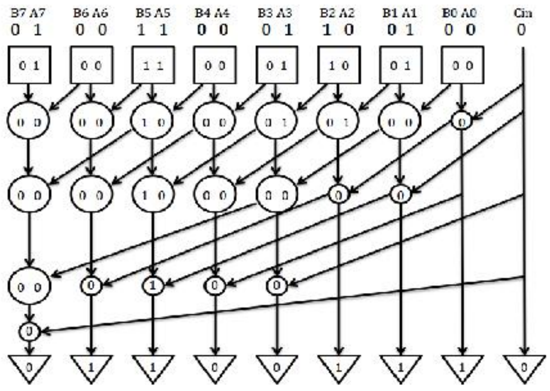


Fig. 4 8 bit Kogge Stone adder

The complete functioning of KSA can be easily comprehended by analyzing it in terms of three distinct parts:

Pre processing

This step involves computation of generate and propagate signals corresponding to each pair of bits in A and B. These signals are given by the logic equations below:

pi = Ai xor Bi
gi = Ai and Bi

Carry look ahead network

This block differentiates KSA from other adders and is the main force behind its high performance. This step involves computation of carries corresponding to each bit. It uses group propagate and generate as intermediate signals which are given by the logic equations below:

Pi:j = Pi:k+1 and Pk:j
Gi:j = Gi:k+1 or (Pi:k+1 and Gk:j)

Post processing

This is the final step and is common to all adders of this family (carry look ahead). It involves computation of sum bits. Sum bits are computed by the logic given below:

Si = pi xor Ci-1

The schematic of KSA is implemented by using following building blocks :

Bit propagate and generate

This block implements the following logic:

Gi = Ai AND Bi
Pi = Ai XOR Bi

Group propagate and generate

This block implements the following logic:

G2 = G1 OR (G0 AND P1)
P2 = P1 AND P0

The propagational delay of the two adders used in existing system and proposed system is given in tabular column below:

Ripple carry adder	Kogge stone adder
n	log(2n)

RESULTS

Delay Comparison

The Tabular column given below gives the delay comparison between 8X8 multiplier using Ripple carry Adder and 8X8 multiplier using Kogge Stone adder. The proposed multiplier shows a 11% improvement in delay than the existing one.

8X8 multiplier using ripple carry adder	8X8 multiplier using Kogge Stone adder
55.700ns	49.800ns

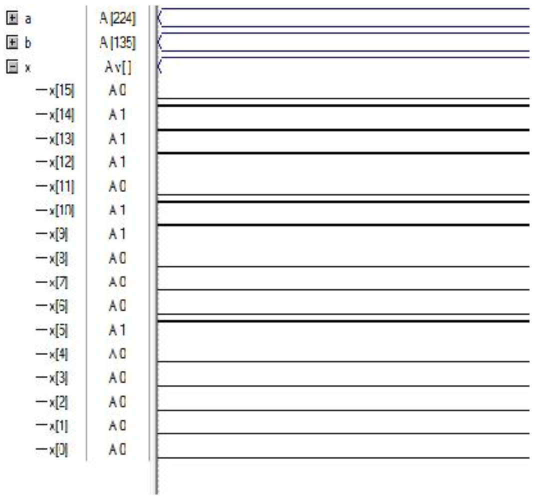
Simulation Output

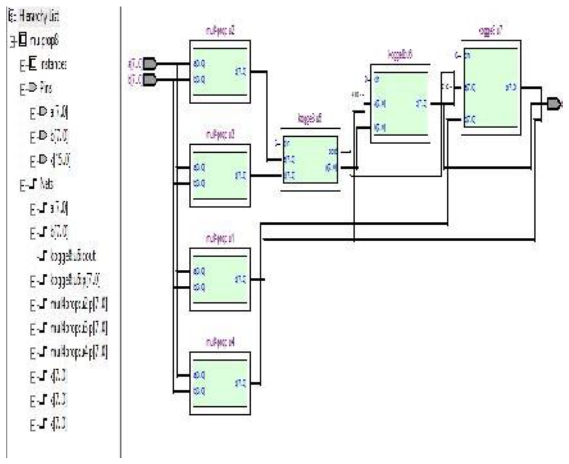
In this sample simulation output , the first input from a0 to a7 is taken as 224 and b0 to b7 is taken as 135. The multiplication output obtained is 30,240 in binary form.

a (input 1)	-	11100000
b (input 2)	-	10000111
x (output)	-	0111011000100000

RTL Synthesis Results

The RTL synthesis is done using Quartus II for the proposed 8X8 multiplier using Kogge Stone Adder. The first four blocks consists of 4X4 multiplier which is based on Vertical and Crosswire Algorithm explained in previous chapters. The other three blocks are 8 bit Kogge Stone adder.





FPGA Implementation

The binary files are generated in the Quartus II software and these binary files are configured and the 8X8 multiplier using Kogge Stone adder is implemented in Altera Acex FPGA. The prototyping is done for both the multiplier architecture which uses ripple carry adder and Kogge stone adder and the output is viewed.

CONCLUSION

Thus a delay efficient method of multiplication based on Vertical and Crosswire Algorithm using Kogge Stone adder is implemented in Altera Acex FPGA. From the simulation results it is seen that the computational path delay for 8X8 bit multiplier using ripple carry adder is 55.700 ns and the computational path delay for 8X8 bit multiplier using Kogge Stone adder is 48.900 ns. Hence we can say that the multiplier using Kogge Stone adder is much more efficient than the multiplier using Kogge Stone adder in terms of delay.

Future Scope

In future , it is possible to adopt this multiplier

Architecture instead of the existing multiplier architectures as the overall speed of the 8 bit multiplication is enhanced. It can also be designed using Brent Kung adder instead of Kogge Stone adder since Brent Kung adder has area less than that of Kogge stone adder.

References

1. Adilakshmi Siliveru , M.Bharathi,"Design of Kogge- Stone and Brent - Kung adders using Degenerate Pass Transistor Logic".2013
2. Bipin,Sakshi,"Design of multiplier using regular partial products".2013
3. Digital Logic with VHDL design by Stephen Brown.
4. Digital Systems Principles and Applications by Ronald J. Tocci.
5. G.Jaberipur and A. Kaivani,"Binary - coded decimal digit multipliers".2007
6. Giridhari Muduli, Siddhartha Kumar Dash, Bibhu Datta Pradhan ,Manas Ranjan Jena.“Design of Digital Multiplier with Reversible Logic by Using the Ancient India Vedic Mathematics Suitable for Use in Hardware of Cryptosystems”. 2014
7. Harpreet Singh Dhillon, Abhijit Mitra,"A Reduced - Bit Multiplication Algorithm for Digital Arithmetic".2008
8. Jagadish.K.N, Nagaraj.C. “High Speed Signed Multiplier For Digital Signal Processing Applications".2013
9. Sumit Vaidya, Deepak Dandekar. “Delay–Power Performance Comparison of Multipliers in Vlsi circuit Design”. 2010
10. Vaijyanath Kunchigi, Linganagooda Kulkarni, Subhash Kulkarni,"High Speed and Area Efficient Vedic Multiplier".2014

How to cite this article:

Priscilla Jennifer. J., implementation of multiplier based on vertical and crosswire algorithm using kogge stone adder. International Journal of Current Engineering Sciences - Vol.5, Issue, 3, pp. 4-7, March, 2015

****