International Journal of Electronics and
Communication Engineering (IJECE)
Vol.1, Issue 1 Aug 2012 34-40
© IASET

International Academy of Science,
Engineering and Technology
Connecting Researchers; Nurturing Innovations
IASET

FPGA IMPLEMENTATION OF DIFFERENT MULTIPLIER ARCHITECTURES

MANJULA B M & CHIRAG SHARMA

Department of Electronics and communication Engineering, NMIT Yelahanka Bangalore, India

ABSTRACT

In this paper 2 different multiplier architectures are implemented in Xilink FPGA and compared for their performance. Here these architectures are implemented for 4,8,16 bit

Based on various speed- up schemes for binary multiplication, a comprehensive overview of different multiplier architectures are given in this report. In addition , it is found that booth multiplier is faster than array multiplier.

KEYWORDS: Multiplier, Computer Arithmetic Algorithm, FPGA.

INTRODUCTION

Multiplication is one of the basic functions used in digital signal processing (DSP). It requires more hardware resources and processing time than addition and subtraction. In fact, 8.72% of all instructions in a typical processing unit is multiplier. The multiplier is a fairly large block of a computing system. The amount of circuitry involved is directly proportional to square of its resolution i.e., a multiplier of size of n bits has O (n^2) gates. . This paper presents various multiplier architectures. Multiplier architectures fall generally into two categories i.e., “tree” multipliers and “array” multipliers. Tree multipliers add as many partial products in parallel as possible and therefore, are very high performance architectures.

OBJECTIVES AND TOOLS USED

Project Objectives

The main objective of this project is design and implementation of Array and Booth multiplier for different bit sizes.

Tools Used

Simulation Software: ISE 9.2i is used for design and implementation and ModelSim 6.1e is used for modeling and simulation.

Hardware Used

Xilinx vertex 2p (Family), XC2VP30 (Device), FG (Package) FPGA device.

TYPES OF MULTIPLIER

Array Multiplier

Multiplication is a mathematical operation that at its simplest is an abbreviated process of adding an integer to itself a specified number of times.

. Multiplication involves three main steps:

- Partial product generation
- Partial product reduction
- Final addition

For the multiplication of an n-bit multiplicand with an m-bit multiplier, m partial products are generated and product formed is n + m bits long.

To perform N -bit by N-bit multiplication the N-bit multiplicand A is multiplied by N-bit multiplier B to produce product. The unsigned binary numbers A and B can be expressed as:

$$A = \sum_{i=0}^{n} Ai2^{i} \qquad \text{........(1)}$$

$$B = \sum_{j=0}^{n} Bj\ 2^{j} \qquad \text{----(2)}$$

The product of A and B is P and it can be written in the following form

$$P = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} A_i B_j 2^{i+j} \qquad \text{-----(3)}$$

A n*n multiplier requires n(n-1) ADDERS and n^2 AND gates.

In the simple array, each row of [3:2] compressors adds a partial product to the partial sum, generating a new partial sum and a sequence of carries. The delay of the array depends on the depth of the array. Therefore, the summing time for the simple array isN-2[3:2] Compressor delays, where N is the number of partial products. The array multiplier originates from the multiplication parallelogram. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial products are shifted according to their bit orders and then added.

Architecture

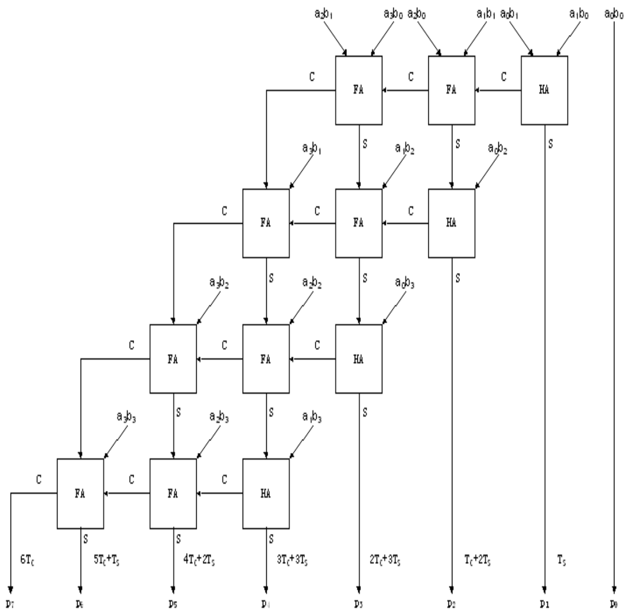


Fig .1 Architecture of 3x3Array Multiplier

Advantages

First advantage of the array multiplier is that it has a regular structure. Since it is regular, it is easy to layout and has a small size. . A second advantage of the array multiplier is its ease of design for a pipelined architecture.

Limitations

Major limitation of array multiplier is its size. As operand sizes increase, arrays grow in size at a rate equal to the square of the operand size.

BOOTH MULTIPLIER

Major limitation of array multiplier is its size. As operand sizes increase, arrays grow in size at a rate equal to the square of the operand size ,hence speed of multiplier reduces .In order to increase the speed of multiplier booth algorithm is used. The Booth multiplier makes use of Booth encoding algorithm in order to reduce the number of partial products by considering two bits of the multiplier at a time, thereby achieving a speed advantage over other multiplier architectures. This algorithm is valid for both signed and unsigned numbers. It accepts the number in 2's complement form, based on radix-2 computation.

Block diagram

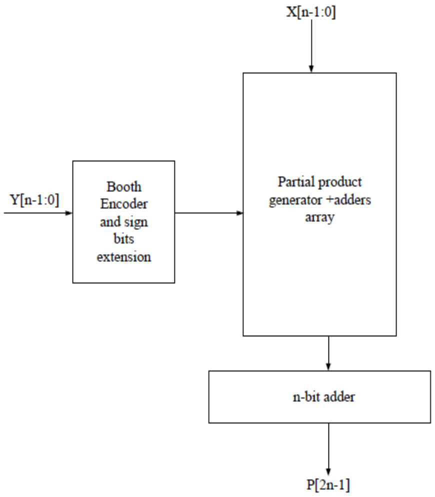

X[n-1:0]
Y[n-1:0]
Booth Encoder and sign bits extension
Partial product generator +adders array
n-bit adder
P[2n-1]

Fig .2 Block Diagram of Booth Multiplier

Booth Algorithm

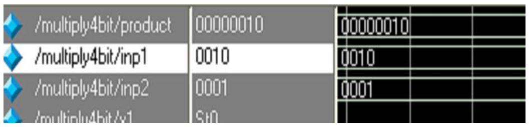
This method can be used to multiply two 2’s complement number without the sign bit extension. Booth observed that when strings of '1' bits occur in the multiplicand the number of partial products can be reduced by subtraction. Table 1 shows the booth algorithm operation.

Table -1. Booth Table

X_i	X_{i-1}	Booth code
0	**0**	**0**
0	**1**	**1**
1	**0**	**-1**
1	**1**	**0**

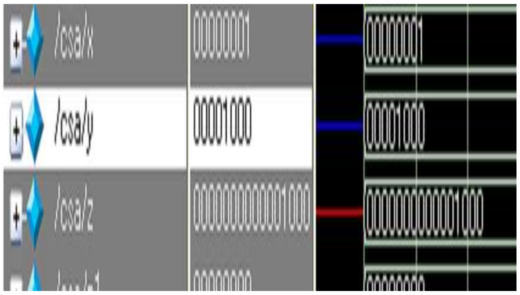
Booth classified group of bits into beginning, middle or end of run. String of zeros avoids arithmetic, so these can be left alone. Booth algorithm changed the original algorithm by looking at two bits of multiplier in contrast to the old algorithm that looks at only one bit at a time. New algorithm has four cases, depending on the values of two bits. Let us assume that the pair of bits examined consists of current bit and bit to right. Second step is to shift the product right.

RESULTS



```
/multiply4bit/product   00000010   00000010
/multiply4bit/inp1      0010       0010
/multiply4bit/inp2      0001       0001
/multiply4bit/x1        St0
```

Fig. 3 Simulation Result of 4x4 Array Multiplier



```
/csa/x   00000001           00000001
/csa/y   00001000           00001000
/csa/z   0000000000001000   0000000000001000
```

Fig. 4 Simulation Result of 8x8 Booth Multiplier

Graphical Representation

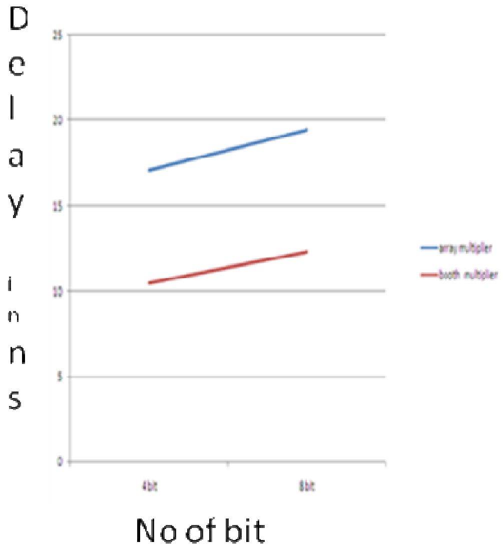


Fig. 5 Graphical Representation of Multiplier Delay V/S Number of Bits.

CONCLUSIONS

The designs of 4-bit and 8-bit Array and Booth multiplier have been implement on Xilinx vertex 2p(family), XC2VP30 (device),FF896 FPGA. The computation delay for 4 bit Array multiplier is 17.45ns. In addition, computation delays obtained for 8-bit Array multiplier is19.33ns and 4-bit & 8-bit Booth multiplier are 10.44 ns and 12.33ns respectively. But here Booth speed is not doubled

COMPARISON

Table- 2 Comparison Table

Multiplier type	Speed	Circuit complexity	Layout	Area
Array	Low	Simple	Regular	Large
Booth	High	Complex	Irregular	Medium

Synthesis Report of 4-Bit Array Multiplier

Selected device	3s400tq144-5
Number of Slices	17 out of 3584 0%
Number of 4 input LUTs	30 out of 7168 0%
Number of IOs	16
Number of bonded IOBs	16 out of 97 16%

Synthesis Report of 8-Bit Multiplier

Selected device	3s400tq144-5
Number of Slices	72 out of 7168 1%
Number of 4 input LUTs	126 out of 7168 0%
Number of bonded IOBs	132 out of 97 32%

Synthesis report of Booth Multiplier

Selected device	3s400tq144-5
Number of Slices	114 out of 3594 3%
Number of 4 input LUTs	200 out of 7168 2%
Number of IOs	-
Number of bonded IOBs	32 out of 97 32%

REFERENCES

1. Weste, Neil H.E. Eshraghian, and Kamran, “CMOS VLSI Design.

2. Parhami Behrooz . computer arithmetic algorithm and hardware design.

3. Wikipedia.

4. Roy Kaushik,Yeo and Kiat-Seng “ Low Power vlsi Sub systems” Mc-Graw Hill, pp. 124-141

5. Yin-Tsung,jin-Fa Lin,Ming –Hwa Sheu“Low power multipliers using Enhanced row bypassing Schemes”

6. G. Bohlender and T. Teufel. Computer Arithmetic: Scientific Computation and Programming Languages, chapter BAPSC: A

7. Decimal Floating-Point Processor for Optimal Arithmetic,pages 31–58. B. G. Teubner, 1987.

8. Shiann-Rong Kuang”Design of Power-Efficient Configurable Booth Multiplier” Circuits and Systems I: Regular Papers, IEEE Transactions

9. R. D. Kenney, M. J. Schulte, and M. A.Erle. A High-Frequency Decimal Multiplier. International Conference on Computer Design

10. M. A. Song, L. D. Van, C. C. Yang, S. C. Chiu, and S. Y. Kuo, "A framework for the design of error-aware power-efficient fixed-width Booth multipliers," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2005, pp. 81-84, Kobe, Japan