International Journal of Electronics and Communication Engineering (IJECE)
ISSN 2278-9901
Vol. 2, Issue 2, May 2013, 33-42

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

A HIGH SPEED AND LESS AREA MULTIPLIER FOR HIGH SPEED PROCESSOR BY SDT TECHNIQUE

M. SABARINATHAN[1], G. OMPRAKASH[2], S. RAMACHANDRAN[3] & V. BALAMURUGAN[4]

[1,2,3]Student, M.Tech VLSI Design, Sathyabama University, Tamil Nadu, India

[4]Assistant Professor, ECE Department, Sathyabama University, Tamil Nadu, India

ABSTRACT

A high speed processor depends greatly on the multiplier as it is one of the key hardware blocks in most digital signal processing systems as well as in general processors. This paper presents a high speed 8x8 bit Vedic multiplier architecture which is quite different from the Conventional method of multiplication. The most significant aspect of the proposed method is that, the developed multiplier architecture is based on Vertical and Crosswise structure of Ancient Indian Vedic Mathematics. It generates all partial products and their sum in one step. This also gives chances for modular design where smaller block can be used to design the bigger one. So the design complexity gets reduced for inputs of larger no of bits and modularity gets increased. The modified Vedic multiplier is coded in Balsa Hardware Description Language which uses a syntax directed translation technique, synthesized and simulated using same EDA (Electronic Design Automation) tool.

KEYWORDS: Balsa, Carry Save Adder (CSA), Multiplication, Syntax Directed Translation (SDT), Urdhva Tiryagbhyam Sutra, Vedic Mathematics, and Vedic Multiplier (VM)

INTRODUCTION

Multipliers are extensively used in Microprocessors, DSP and Communication applications. For higher order multiplications, a huge number of adders are to be used to perform the partial product addition. The need of high speed multiplier is increasing as the need of high speed processors are increasing. Higher throughput arithmetic operations are important to achieve the desired performance in many real time signal and image processing applications. One of the key arithmetic operations in such applications is multiplication and the development of fast multiplier circuit has been a subject of interest over decades. Reducing the time delay and power consumption are very essential requirements for many applications.

In the past multiplication was implemented generally with a sequence of addition, subtraction and shift operations. Two most common multiplication algorithms followed in the digital hardware are array multiplication algorithm and Booth multiplication algorithm. Due to the importance of digital multipliers in DSP, it has always been an active area of research. Vedic mathematics is the name given to the ancient system of mathematics, which was rediscovered from the ancient Indian scriptures between 1911 and 1918 by Jagadguru Swami Sri Bharati Krishna Tirthaji (1884-1960), a scholar of Sanskrit, mathematics, history and philosophy. The whole of Vedic mathematics is based on 16 Vedic sutras, which are actually word formulae describing natural ways of solving a whole range of mathematical problems [1]. The paper is organized as follows. Section 2 describes the basic information about the Balsa synthesis system. Section 3 describes the basic methodology of Vedic multiplication technique. Section 4 describes the modified multiplier architecture based on Vedic multiplication and the generalized algorithm for NXN bit Vedic multiplier. Section 5 describes the design and

implementation of Vedic multiplier module in Balsa HDL. Section 6comprises of Result and Discussion in which device utilization summary and computational path delay obtained for the modified Vedic multiplier (after synthesis) is discussed. Finally Section 7 comprises of Conclusion.

BALSA SYNTHESIS SYSTEM

The syntax-directed synthesis of asynchronous circuits is based on the compilation of a high-level description into a communicating network of predesigned modules. Each language construct is mapped one-to-one into a network of components that implement it [7]. The structure of the resulting circuit is hence directly related to the description style, giving designers a high degree of flexibility to optimize circuits at the description language level. The Balsa synthesis system uses the syntax-directed translation to generate a handshake circuit [8] from a description written in the Balsa language. A handshake circuit is a communicating network of handshake components connected point-to-point using handshake channels. Channels carry data from one component to another under the control of a request-acknowledge handshake protocol. Each handshake component has a parameterized gate-level implementation, and a gate-level netlist can be directly generated for a handshake circuit by applying the appropriate parameters, e.g., data width, to each component instance. A large-scale handshake circuit can then be constructed by the composition of small handshake components that are implemented in isolation.

VEDIC MULTIPLICATION TECHNIQUE

The use of Vedic mathematics lies in the fact that it reduces the typical calculations in conventional mathematics to very simple one. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works. Vedic Mathematics is a methodology of arithmetic rules that allow more efficient speed implementation. It also provides some effective algorithms which can be applied to various branches of engineering such as computing.

Urdhva Tiryagbhyam Sutra

The proposed Vedic multiplier is based on the "Urdhva Tiryagbhyam" sutra (algorithm) [3]. These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware. It is a general multiplication formula applicable to all cases of multiplication. It literally means "*Vertically and Crosswise*". It is based on a novel concept through which the generation of all partial products can be done with the concurrent addition of these partial products. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Due to its regular structure, it can be easily layout in microprocessors and designers can easily circumvent these problems to avoid catastrophic device failures. The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier based on this sutra has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other conventional multipliers.

Multiplication of Two Decimal Numbers

To illustrate this scheme, let us consider the multiplication of two decimal numbers 252 x 846 by Urdhva-Tiryakbhyam method as shown in Figure1. The digits on the both sides of the line are multiplied and added with the carry from the previous step. This generates one of the bits of the result and a carry. This carry is added in the next step and

hence the process goes on. If more than one line are there in one step, all the results are added to the previous carry. In each step, least significant bit acts as the result bit and all other bits act as carry for the next step. Initially the carry is taken to be zero.

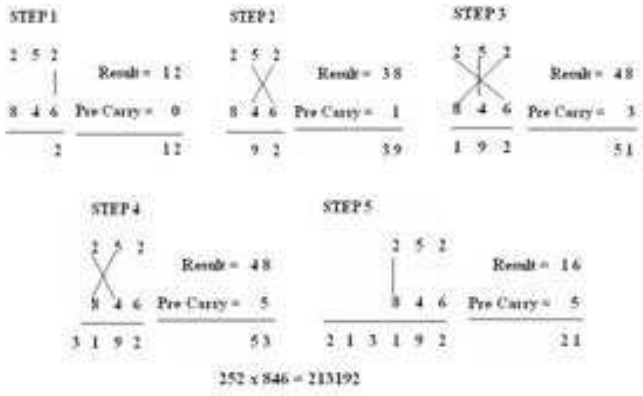

Figure 1: Multiplication of Two Decimal Numbers-252 X 846 [5]

THE MODIFIED MULTIPLIER ARCHITECTURE

The hardware architecture of 2X2, 4x4 and 8x8 bit Vedic multiplier module are displayed in the below sections. Here, "Urdhva-Tiryagbhyam" (*Vertically and Crosswise*) sutra is used to propose such architecture for the multiplication of two binary numbers. The beauty of Vedic multiplier is that here partial product generation and additions are done concurrently. Hence, it is well adapted to parallel processing. The feature makes it more attractive for binary multiplications. This in turn reduces delay, which is the primary motivation behind this work.

Vedic Multiplier for 2x2 Bit Module

The method is explained below for two, 2 bit numbers A and B where $A = a1a0$ and $B = b1b0$ as shown in Figure 2. Firstly, the least significant bits are multiplied which gives the least significant bit of the final product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with, the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the final product and the carry is added with the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and carry becomes the fourth bit of the final product.

$s0 = a0b0$; ---------- (1)

$c1s1 = a1b0 + a0b1$; ---------- (2)

$c2s2 = c1 + a1b1$; ---------- (3)

The final result will be $c2s2s1s0$. This multiplication method is applicable for all the cases.

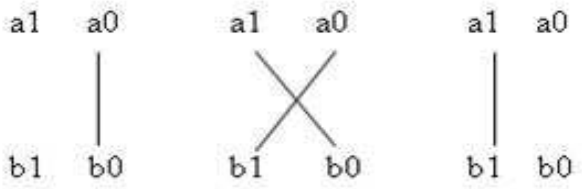

Figure 2: The Vedic Multiplication Method for Two 2-Bit Binary Numbers [5]

The 2X2 Vedic multiplier module is implemented using four input AND gates & two half-adders which is displayed in its block diagram in Figure 3. It is found that the hardware architecture of 2x2 bit Vedic multiplier is same as the hardware architecture of 2x2 bit conventional Array Multiplier [2]. Hence it is concluded that multiplication of 2 bit binary numbers by Vedic method does not made significant effect in improvement of the multiplier's efficiency. Very precisely we can state that the total delay is only 2-half adder delays, after final bit products are generated, which is very similar to Array multiplier. So we switch over to the implementation of 4x4 bit Vedic multiplier which uses the 2x2 bit multiplier as a basic building block. The same method can be extended for input bits 4 & 8. But for higher no. of bits in input, little modification is required.

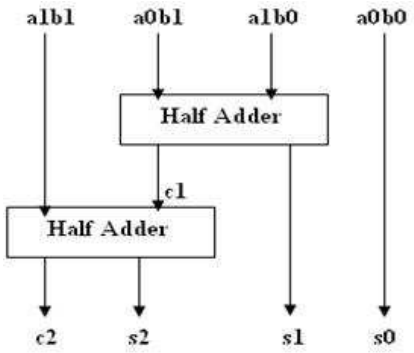

Figure 3: 2X2 Vedic Multiplier [5]

Vedic Multiplier for 4 X 4 Bit Module

The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multiplier modules as discussed in Figure 3. Let's analyze 4x4 multiplications, say A = A3 A2 A1 A0 and B = B3 B2 B1 B0. The output line for the multiplication result is – S7 S6 S5 S4 S3 S2 S1 S0.To understand the concept of, Block diagram of 4X4 Vedic Multiplier is shown in Figure 5. To get the product (s7, s6, s5, s4, s3, s2, s1, s0), four 2X2 bit Vedic Multiplier (Figure 3) and three 4 bit Carry save adder (CSA) are required. The modified Vedic multiplier can be used to reduce delay. Early literature speaks about Vedic multipliers based on array multiplier structures. On the other hand, we proposed a new architecture, which is efficient in terms of speed. The arrangements of Carry Save Adders shown in Figure 5, helps us to reduce delay. Interestingly, 8x8 Vedic multiplier modules are implemented easily by using four 4x4 multiplier modules.

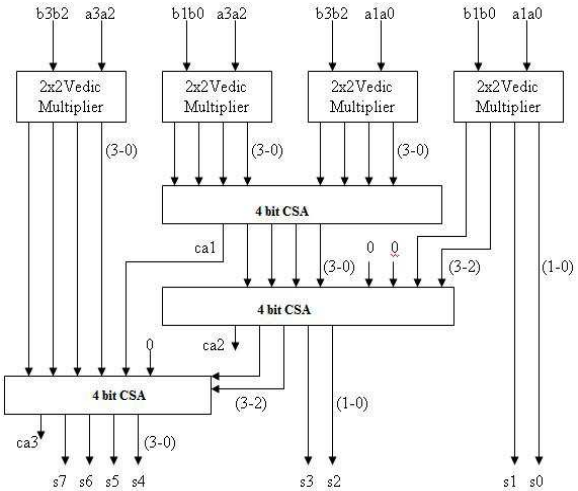

Figure 4: Modified 4X4 Vedic Multiplier

Vedic Multiplier 8X8 Bit Module

The 8x8 bit Vedic multiplier module as shown in the block diagram in Figure 6 can be easily implemented by using four 4x4 bit Vedic multiplier modules as discussed in the previous section. Let's analyze 8x8 multiplications, say A= A7 A6 A5 A4 A3 A2 A1 A0 and B= B7 B6 B5 B4 B3 B2 B1B0. 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. The 16 bit product can be written as:

P = A x B

= (AH-AL) x (BH-BL)

= AH x BH + (AH x BL + AL x BH) + AL x BL

The outputs of 4X4 bit multiplier are added accordingly to obtain the final product. Here total three 8 bit Carry Save Adder (CSA) are required as shown in Figure 5.

Generalized Algorithm for NXN Bit Vedic Multiplier

Step 1: Divide the multiplicand A and multiplier B into two equal parts, each consisting of [N to (N/2) +1] bits and [N/2 to 1] bits respectively, where first part indicates the MSB and other represents LSB.

Step 2: Represent the parts of A as AM and AL, and parts of B as BM and BL. Now represent A and B as AM AL and BM BL respectively.

Step 3: For A X B, we have general format as shown in Figure 6

Step 4: The individual multiplications product can be obtained by the partitioning method and applying the basic building blocks.

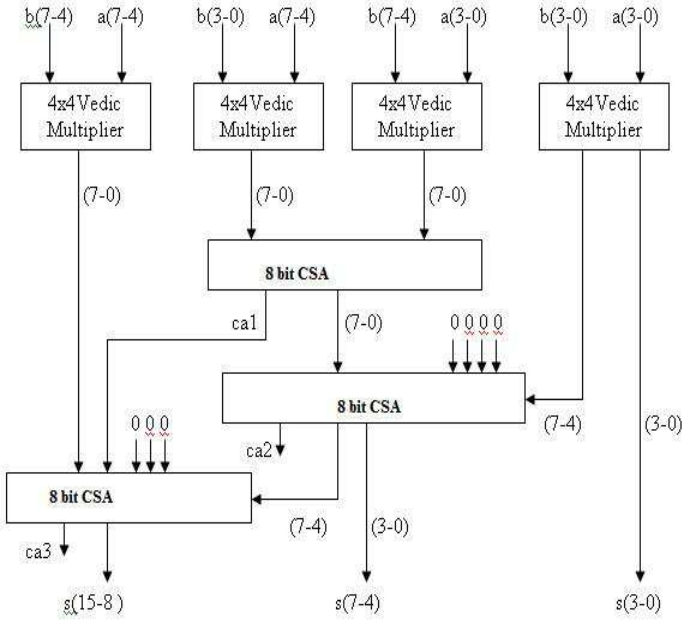

Figure 5: Modified 8X8 Vedic Multiplier

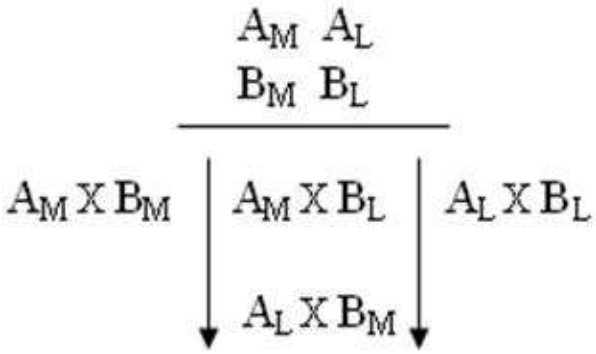

Figure 6: Generalized Representation of Vedic Multiplier [5]

By adopting the above generalized algorithm we can implement Vedic Multiplier for any number of bits say 16, 32, 64, and so on, as per the requirement. Therefore, it could be possible to implement this Vedic multiplier in the ALU (Arithmetic Logic Unit) which will reduce the computational speed drastically & hence improves the processors efficiency.

IMPLEMENTATION IN BALSA

In this work, 8X8 bit Vedic Multiplier is designed in Balsa Hardware Description Language. Logic synthesis and simulation was done using an EDA (Electronic Design Automation) tool in Balsa [6]. The performance of the circuit (delay calculation) is evaluated on Xilinx device. In Balsa HDL we can also evaluated a circuit area cost [4]. For example Figure 8. 8X8 bit Vedic multiplier input, the multiplier a=”00000011” and multiplicand b=”00000011” and we get 16 bit output =”0000000000001001”.

The Figure 8 Shown the Handshake circuit graph representation produced by the Balsa compiler (Balsa C).This handshake graph is composed of handshake component which are elegantly described in Balsa library [4]. The Balsa produce a breeze cost which is used to estimate the area cost of the circuit. By using the carry save adder instead of ripple carry adder the circuit area will be reduced which is shown in the table 1.

RESULTS AND DISCUSSIONS

Table 1 shows the comparison of 8X8 bit conventional multiplier [5] with ours modified Vedic multiplier in terms of circuit cost area (mm^2) produced by Balsa tool and computational path delay produced by Xilinx in nanoseconds (ns).

Table 1: Comparison of 8X8 Multiplier (ns)

Parameter	Multiplier using Ripple Carry Adder(Existing)	Multiplier using Carry Save Adder(Modified)
Balsa Cost (area) (mm^2)	446171.75	318188.75
Delay Before clock (ns)	2.796	2.611
Delay after clock (ns)	4.283	4.283
Path Delay	9470.536	8718.140

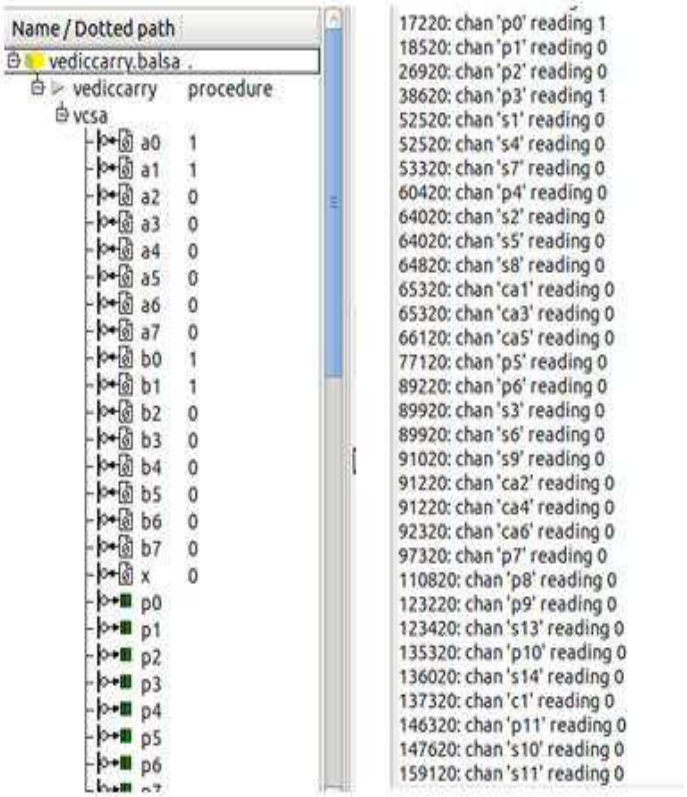

Figure 7: Simulation Result 8X8 Vedic Multiplier

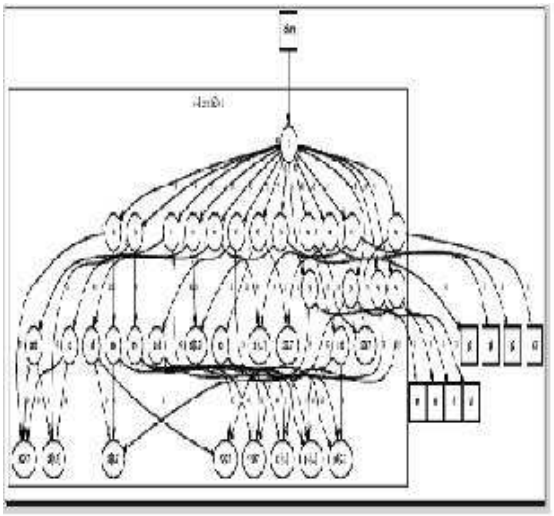

Figure 8: Handshake Circuit Graph

CONCLUSIONS

This paper presents a highly efficient method of multiplication – "Urdhva Tiryakbhyam Sutra" based on Vedic mathematics. It gives us method for hierarchical multiplier design and clearly indicates the computational advantages offered by Vedic methods. The computational path delay for proposed 8x8 bit Vedic multiplier is found to be 8718.140 ns. Hence our motivation to reduce delay is finely fulfilled. Therefore, we observed that the Vedic multiplier is much more efficient than conventional multiplier [5] in terms of execution time (speed).

ACKNOWLEDGEMENTS

We are heartly thankful to our Guide, Head, Department of Electronics and Communication, Sathyabama University, and our friends for his kind supports.

REFERENCES

1. Jagadguru Swami, Sri Bharati Krisna, Tirthaji Maharaja, "Vedic Mathematics or Sixteen Simple Mathematical Formulae from the Veda, Delhi (1965)", Motilal Banarsidas, Varanasi, India, 1986.
2. M. Morris Mano, "Computer System Architecture", 3rd edition, Prientice-Hall, New Jersey, USA, 1993, pp. 346-348.
3. H. Thapliyal and H.R Arbania. "A Time-Area-Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics"
4. Sune Fallgaard Nielsen, Jens Sparsø and Jan Madsen,"Behavioral Synthesis of Asynchronous Circuits Using Syntax Directed Translation as Backend" ieee transactions on very large scale integration (vlsi) systems, vol. 17, no. 2, february 2009.
5. Pushpalata Verma, K. K. Mehta "Implementation of an Efficient Multiplier based on Vedic Mathematics Using EDA Tool " International journal of Engineering and Advanced Technology (IJEAT) pp-2012.
6. Doug Edwards, Andrew Bardsley,Lilian Janin, Luis Plana & Will Toms Balsa :tutorial guide Version V3.5 – Printed: 19/5/06
7. Ren-Der Chen,Yu-Cheng Chou, Wan-Chen Liu "Comparative Design of Floating-Point Arithmetic Units Using the Balsa Synthesis System" 2011 International Symposium on Integrated Circuits.
8. K. van Berkel, Handshake Circuits - An asynchronous architecture for VLSI programming. Cambridge University Press, 1994.

AUTHOR'S DETAILS

Mr M. Sabarinathan obtained his B.E (Electronics and communication Engineering) in 2009 from Maamallan Institute of Technology, Chennai and he is pursuing his M.TECH in VLSI Design from Sathyabama University, Chennai India.

Mr. G.Omprakash obtained his B.E (Electronics and communication Engineering) in 2011 from I.F.E.T College of Engineering, Villupuram and he is pursuing his M.TECH in VLSI Design from Sathyabama University, Chennai India.

Mr S.Ramachandran obtained his B.E (Electronics and communication Engineering) in 2009 from Thangavelu Engineering College, Chennai and he is pursuing his M.TECH in VLSI Design from Sathyabama University, Chennai India.

Mr.V.Balamurugan obtained his B.E (Electronics and communication Engineering) in 2004 from Muthyammal Engineering College,Nammakal and M.TECH in VLSI Design from Sathyabama University in 2006.At present Research scholar and Faculty of ECE Department in Sathyabama University,