

Optimized high performance multiplier using Vedic mathematics

Pradeep M C¹, Dr. Ramesh S²

¹(Department of Electronics and Communication Engineering, Dr. Ambedkar Institute of Technology, India)

²(Department of Electronics and Communication Engineering, Dr. Ambedkar Institute of Technology, India)

Abstract: Multiplication is the commonly used operations in a Central Processing Unit (CPU). The performance of the CPU depends on multiplier which may be slower and may consume significant amount of power. This work presents a low power and high speed multiplier architecture using Vedic mathematics technique. The work also proves the efficiency of Urdhava Tiryakbhyam sutra of Vedic mathematics which shows a difference between actual process of multiplication and Vedic multiplication. Carry Save Adder (CSA) is used in the architecture to have reduced delay. The proposed multiplier circuit is synthesized using Xilinx 13.1 version tool for Field Programmable Gate Array (FPGA) flow and Cadence 12.10 version tool for Application Specific Integrated Circuit (ASIC) flow for the analysis of dynamic power consumption and propagation delay and the design is simulated using Modelsim 6.5 version tool for functional verification.

Keywords: ASIC flow, CSA, FPGA flow, Vedic mathematics, Urdhava Tiryakbhyam sutra

I. Introduction

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable amount of power. Therefore low-power multiplier design has been an important part in low- power Very Large Scale of Integration (VLSI) system design. Multiplication is the process of adding a number of Partial Products (PP). Multiplication algorithms differ in terms of PP generation and PP addition to produce the final result [1]. Higher throughput arithmetic operations are important to achieve the desired performance in many real time signal and image processing applications [2]. A multiplier is one of the key hardware blocks in most Digital Signal Processing (DSP) systems [3][4]. Typical DSP applications where a multiplier plays an important role include digital filtering, digital communications and spectral analysis. Many current DSP applications are targeted at portable, battery-operated systems, so that power dissipation becomes one of the primary design constraints. Since multipliers are rather complex circuits and must typically operate at a high system clock rate, reducing the delay of a multiplier is an essential part of satisfying the overall design.

A multiplier block can be implemented by using many algorithms. The two most common multiplication algorithms followed in digital hardware are Array multiplication and Booth multiplication [5]. Vedic multiplication algorithm is gaining reputation in the recent years. Vedic mathematics is the name given to the ancient system of mathematics, which was rediscovered from ancient Indian scriptures between 1911 and 1918. The Vedic mathematics reduces the typical calculations in conventional mathematics to very simple one [6]. This is so because the Vedic formulae are claimed to be based on the natural principles on which the human mind works. This makes the use of Vedic mathematics very attractive.

This paper is organized as follows. In section 2, the overview of related work is briefly reviewed. In section 3, the proposed Vedic multiplier architecture is discussed. The performance of proposed Vedic multiplier architecture is compared with existing Vedic multiplier architecture with results and discussion in section 4. Finally, a brief conclusion is given in section 5.

II. Related Work

Vedic mathematics is a part of four Vedas (books of wisdom). It is a part of Stapatya-Veda (book of civil engineering and architecture), which is an upa-Veda (supplement) of Atharva Veda. It gives explanation of several mathematic terms including arithmetic, geometry, trigonometry, factorization and even calculus [7].

His holiness Jagadguru Shankaracharya Bharathi Krishna Teerthaji Maharaj(1884-1960) put all his work together and gave it's mathematical explanation while discussing it for various application. Vedic mathematics deals with several basic as well as complex mathematic operations, especially methods of basic arithmetic [8] are extremely simple and powerful. The system of Vedic mathematics is based on 16 sutras (or aphorisms) - formulae and 13 up-sutras or corollaries [9].

One of the sutras of Vedic mathematics implied for multiplication is Urdhava Tiryakbhyam (vertical and cross wire) [7] which is also the foundation of the proposed design. It is based on a concept through which the generation of all Partial Products (PP) can be done with the concurrent addition of these PPs. The parallelism

in generation of PPs and their summation is obtained by vertical and cross wire multiplication and addition. According to this algorithm a 4×4 bit multiplication can be carried out in the following way.

- 1) Firstly least significant bits are multiplied which gives the Least Significant Bit (LSB) of the product (vertical).
- 2) 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 (cross wire). The sum gives second bit of the product and the carry is added in the output of the next stage sum obtained by the cross wire and vertical multiplication and addition of three bits of two numbers from least significant position.
- 3) Next, all the four bits are processed with cross wire multiplication and addition to give the sum and carry. The sum is the corresponding bit of the product and the carry is again added to the next stage multiplication and addition of three bits except the LSB.
- 4) The same operation continues until the multiplication of two most significant bits to give the Most Significant Bit (MSB) of the product.

An illustration is given with the help of line diagrams in Fig.1.

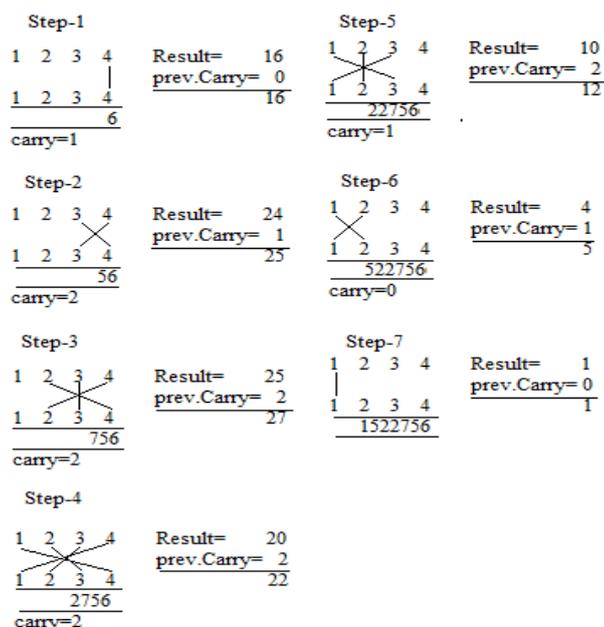


Figure.1: Multiplication of $1234 \times 1234 = 1522756$ by urdhava tiryakbhyam sutra with line diagram.

The beauty of Vedic multiplier is that here Partial Product Generation (PPG) and additions are done concurrently. Hence, it is well adapted to parallel processing. This feature makes it more attractive for binary multiplications. This, in turn, reduces delay. One such Vedic multiplier was proposed in [10]. The architecture of $n \times n$ multiplier proposed in [10] using Vedic mathematics is shown in Fig.2. To get final product, one n -bit Carry Save Adder (CSA), one $(n+1)$ -bit binary adder and one n -bit binary adder are used. In this referred paper, the n -bit CSA is used to add three n -bit operands, i.e. concatenated n -bit $((n/2)$ zeros & most significant $(n/2)$ output bits of right hand most of $n \times n$ multiplier module) as shown in Fig.2 and two n -bit operands we get from the output of two middle $n \times n$ multiplier modules. It may be noted that the outputs of the CSA (sum and carry) are fed into a $(n+1)$ -bit binary adder to generate $(n+1)$ -bit sum, as desired.

It may be reiterated the fact that the first $[(n/2)-1$ to $0]$ -bit final product is directly obtained from rightmost $n \times n$ multiplier module. Next $[(n/2)$ to $(n-1)]$ -bit is obtained from least significant $(n/2)$ -bits of $(n+1)$ -bit sum obtained from the $(n+1)$ -bit binary adder. Finally, as shown in Fig.2, the n -bit output of the left most $n \times n$ multiplier module and concatenated n -bits $((n/2)-1)$ zeros & the most significant three bits of $(n+1)$ -bit sum are fed into an n -bit binary adder. The sum produced by n -bit binary adder gives the remaining $[(2n-1)$ to $n]$ -bit final products. The referred Vedic multiplier can be used to reduce delay.

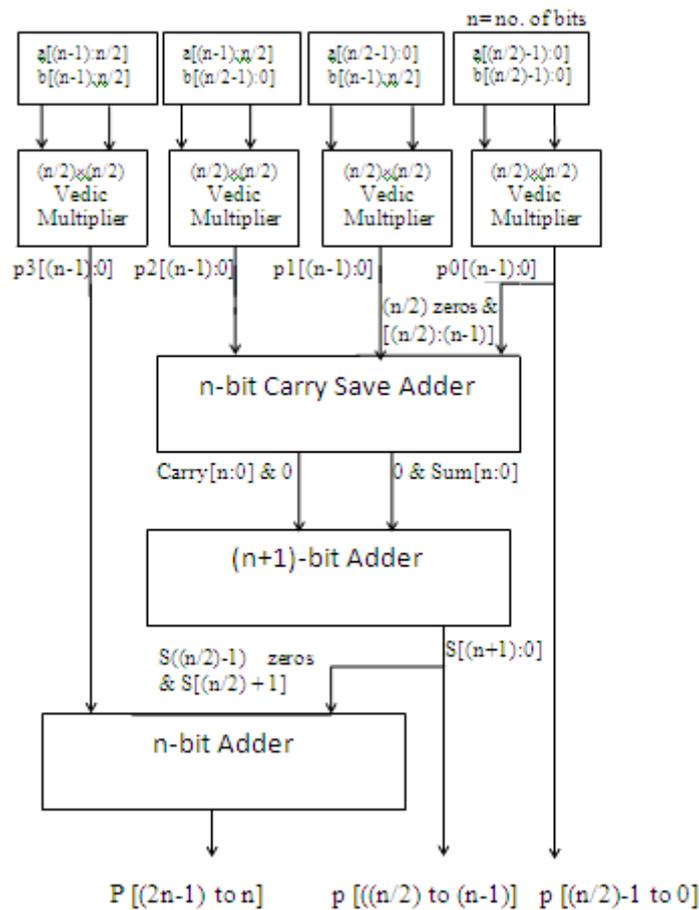


Figure.2: Block diagram of multiplier architecture proposed in [10].

III. Proposed Vedic Multiplier Architecture

The proposed Vedic multiplier is designed using Urdhava Tiryakbhyam sutra. The Partial Products (PP) of multiplier using Urdhava Tiryakbhyam sutra is shown in Fig.3. As shown in Fig.3 the PPs are grouped into four $(n/2)$ multiplier modules and they are added using Carry Save Adder (CSA) to produce the final multiplier products. The block diagram of Urdhava multiplier is shown in Fig.4. Three input CSA is used in the architecture. The first input is obtained by taking $[(n-1) \text{ to } (n-(n/2))]$ -bit result of the first multiplier module (rightmost $n \times n$ multiplier) and taking fourth multiplier module (leftmost $n \times n$ multiplier) result and concatenating them. The second and third input is obtained by taking second and third multiplier module (middle $n \times n$ multipliers) results and concatenating each of them with two zeros at the Most Significant Bit (MSB) side to make it $(n+ (n/2))$ -bit for addition. First $[(n-(n/2) + 1) \text{ to } 0]$ -bit product is obtained by taking $[n-(n/2) + 1 \text{ to } 0]$ -bit result of first multiplier module directly. While the remaining resultant bits $[(2n-1) \text{ to } (n-(n/2))]$ is obtained by the sum produced by CSA. Since only CSA is used in the architecture there is a considerable amount of reduction in dynamic power consumption and overall propagation delay than the work proposed in [10].

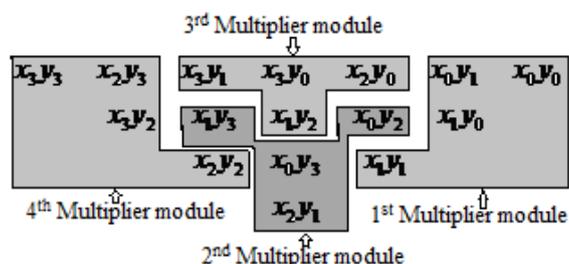


Figure.3: 4x4 Vedic multiplier partial products using urdhva tiryakbhyam sutra.
n=no. of bits

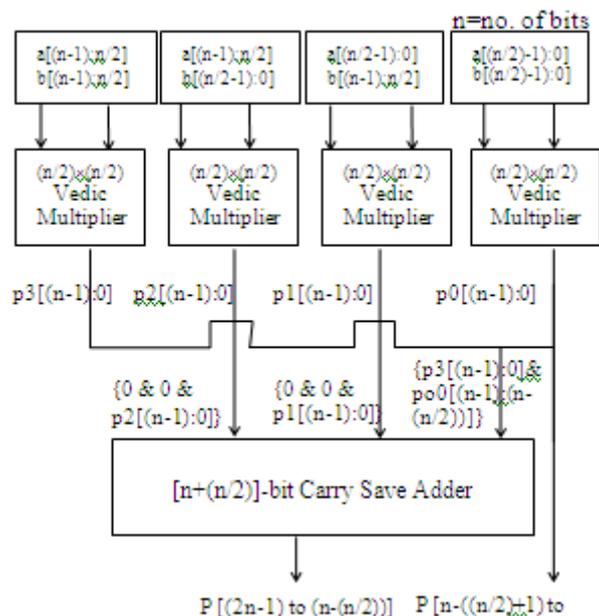


Figure.4: Block diagram of proposed vedic multiplier architecture.

IV. Results and Discussion

Multiplier for 4-bit and 8-bit were designed for both existing [10] and optimized methods. The designed Vedic multiplier were simulated using Modelsim tool of version 6.5 for functional verification and synthesized using Cadence RTL compiler tool of version 12.10 with 180nm standard cell technology library and Xilinx tool of version 13.1 (Vertex 7 family with speed grade of -1) for dynamic power and propagation delay analysis. The simulation results for the proposed 4-bit and 8-bit Vedic squarer is shown in Fig.5 and Fig.6. Simulation results in Fig.5 and Fig.6 are shown for various possible input combinations. As shown Fig.5 ‘a’ and ‘b’ are two 4-bit inputs and ‘p’ is the output (product of two inputs ‘a’ and ‘b’) which results in 8-bit binary number. Similarly as shown in Fig.6 ‘a’ and ‘b’ are two 8-bit inputs and ‘p’ is the output which results in 16-bit binary number. Block diagram of 4-bit and 8-bit optimized Vedic multiplier are shown in Fig.7 and Fig.8. As shown in block diagram ‘a’ and ‘b’ are the input given to multiplier module and ‘p’ is output of multiplier module, ‘m1’, ‘m2’, ‘m3’ and ‘m4’ are multiplier modules and ‘s1’ is the adder module.

The performance of the proposed multiplier design for 4-bit and 8-bit is shown in Table [1, 2, 3 and 4]. Comparison is made between the existing Vedic multiplier architecture [10] and proposed Vedic multiplier architecture. The comparison results in Table [1, 2, 3 and 4] shows that the proposed multiplier architecture not only consumes less power but also performs high speed than multiplier design in [10].

Table 1: Synthesis Result of 4-Bit Multiplier in ASIC Flow

Parameters	Propagation Delay (ns)	Dynamic Power (mw)
Existing[10]	2.694	0.0066
Optimized	2.118	0.0067
% Improvement	21.38	-1.51

Table 2: Synthesis Result of 8-Bit Multiplier in ASIC Flow

Parameters	Propagation Delay (ns)	Dynamic Power (mw)
Existing[10]	7.254	0.0473
Optimized	4.826	0.0445
% Improvement	33.47	5.91

Table 3: Synthesis Result of 4-Bit Multiplier in FPGA Flow

Parameters	Propagation Delay (ns)	Dynamic Power (mw)
Existing[10]	5.445	5.12
Optimized	5.118	4.70
% Improvement	6.00	8.20

Table 4: Synthesis Result of 8-Bit Multiplier in FPGA Flow

Parameters	Propagation Delay (ns)	Dynamic Power (mw)
Existing[10]	11.351	14.40
Optimized	9.022	12.84
% Improvement	20.51	10.83

Messages					
/opmultiplier4_bitstr_tb/a	1110	0011	0100	0101	0110
/opmultiplier4_bitstr_tb/b	1111	1000			
/opmultiplier4_bitstr_tb/p	11010010	00011000	00100000	00101000	00110000

Figure.5: Simulation results of 4-bit vedic multiplier.

Messages					
/opmultiplier8_bitstr_tb/a	11111110	10000011		10000100	
/opmultiplier8_bitstr_tb/b	11111111	10000100			
/opmultiplier8_bitstr_tb/p	1111110100000010	0100001110001100		0100010000010000	

Figure.6: Simulation results of 8-bit vedic multiplier.

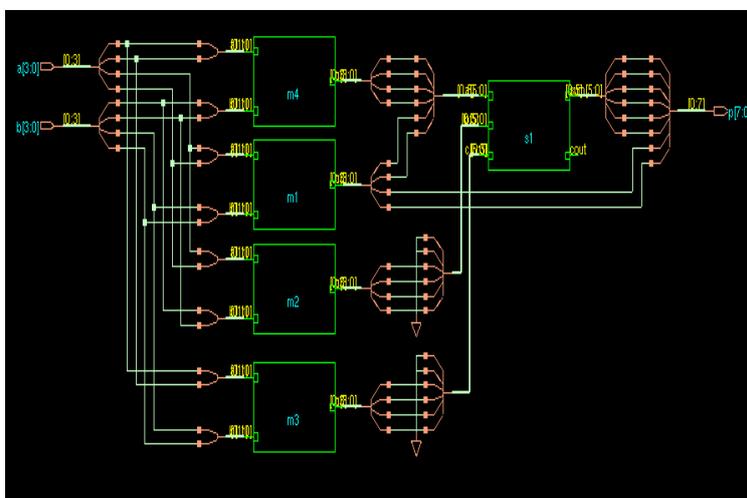


Figure.7: Block diagram of 4-bit optimized vedic multiplier architecture.

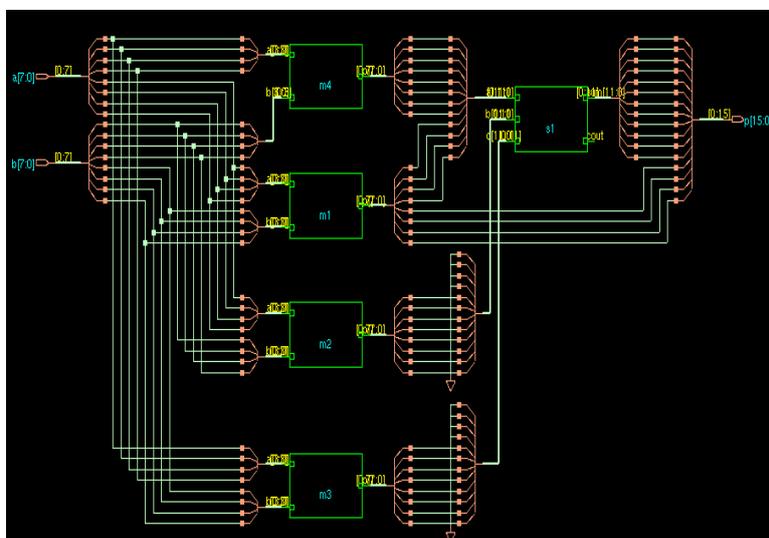


Figure.8: Block diagram of 8-bit optimized vedic multiplier architecture.

V. Conclusion

This work presents a novel binary multiplier design based on the sutra of ancient Indian Vedic mathematics which is highly suitable for high speed arithmetic circuits which have wide application in VLSI signal processing applications. The results shows that as width of multiplier increase the performance also increases which makes the multiplier design highly modular and design complexity gets reduced by using Vedic method. The proposed Vedic multiplier design is simulated and synthesized for 4-bit and 8-bit. The proposed Vedic multiplier results show that for the optimized 8-bit squarer the overall propagation delay is reduced by 33.47% and dynamic power by 5.91% for ASIC flow and similarly 22.25% and 20% for FPGA flow when compared with existing Vedic multiplier architecture [10].

References

- [1]. Reto Zimmermann, Lecture notes on computer arithmetic: principles, architecture and design (Integrated Systems Laboratory, ETH Zurich, March 1999).
- [2]. Sunder S. kidambi, Fayez el-Guibaly and Andreas Antoniou, Area efficient multipliers for digital signal processing applications: IEEE transactions on circuits and systems-II: Analog and Digital Signal Processing, vol. 43, no. 2, February 1996, pp. 90-95.
- [3]. Johnny Pihl and Einar J. Aas, A multiplier and squarer generator for high performance DSP applications: IEEE 39th Midwest symposium on Circuits and Systems, Ames, IA, vol 1, 18-21 Aug 1996, pp. 109-112.
- [4]. Akhalesh K, Itawadiya, Rajesh Mahle, Vivek Patel and Dadan Kumar, Design a DSP operations using vedic mathematics: IEEE International Conference on Communications and Signal Processing (ICCSP), Melmaruvathur, 3-5 April 2013, pp. 897-902.
- [5]. Nick Carter, Schaum's outline of theory and problems of computer architecture (The McGraw-Hill Companies Inc. Indian Special Edition 2009).
- [6]. Parth Mehta and Dhanashri Gawali, Conventional versus vedic mathematical method for hardware implementation of a multiplier: IEEE International Conference on Advances in Computing, Control, & Telecommunication Technologies, Trivandrum, pp. 640-642, 28-29 Dec 2009.
- [7]. A.P Nicholas, K.R Williams and J Pickles, Applications of the vedic mathematics sutra: vertically and crosswire (Inspiration books, Third revised edition, The Vedic mathematics research group, 2010).
- [8]. A.P Nicholas, J Pickles and K Williams, Introductory lectures on vedic mathematics (Polytechnic of North London, July 1982).
- [9]. www.vedicmaths.com
- [10]. kabiraj Sethi and Rutuparna Panda, An improved squaring circuit for binary numbers, International journal of advanced computer science and applications, vol.3 , No.2, 2012, 111-105.