International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 12
August 2016

A Survey on Modular & hybrid multiplication using carry saves adder

Mr. M. GANESH KUMAR[1] & LAVANYA B[2]

1Associate Professor Dept. of ECE, Svr Engineering College Nandyal Mail: - mgani411@gmail.com

[2]PG-Scholar Dept. of ECE, Svr Engineering College Nandyal Mail: - lavanyab.vss@gmail.com

Abstract

Low power and High-speed computing systems have been very much demand in recent years, because of the fast growing technologies in scientific computing applications. Designing a low power and high speed multiplier will have a large impact on applications like Image Processing, Convolution, Fast Fourier Transform, and Filtering and in microprocessors. The sutra named Urdhva-Triyagbhyam (Vertically and Cross wise) from the Ancient Indian Vedic Mathematics used in multiplication process since it has a unique way of calculations. Urdhvatiryakbhyam Sutra is most efficient Sutra giving minimum delay for multiplication of all types of numbers, either small or large. By using this sutra the partial products and sums are generated in one step which reduces the design of architecture in processors. Also, building an ALU using Vedic Multiplier is less complex when compared to Montgomery modular multiplier. In Montgomery modular multiplierwhile implementing two 32 bit values will produce a partial output. In the proposed Vedic Multiplier while implementing two16 bit values will produce a complete 32 bit output. As a result of this algorithm the speed of the computation process is increased and the computing time is reduced due to decrease of path delay compared to the Montgomery modular multiplier. The multiplier that has been used in a Vedic Multiplier built using urdhvaTiryakbhyam Sutra and has been fitted into the MAC design. The design of the Vedic Multiplier is performed in Veriloglanguage and the tool used for simulation is Xilinx Spartan 3.

Keywords: Modular multiplication, carry save adder, hybrid multiplication.

1. Introduction

In VLSI design power, speed and area are the most often used measures for determining the performance of the VLSI design. Arithmetic is the oldest and most elementary branch of Mathematics. The name Arithmetic comes from the Greek word. Arithmetic is used by almost everyone, for tasks ranging from simple day to day work like counting to advanced science and business calculations. As a result, the need for a faster and efficient Arithmetic Unit in

computers has been a topic of interest over decades. The work presented makes use of Vedic Mathematics and goes step by step, by first designing a Vedic Multiplier, then a Multiply Accumulate Unit and then finally an Arithmetic module which uses this multiplier and MAC unit.

The four basic operations in elementary arithmetic are addition, subtraction, multiplication and division. Multiplication basically is the mathematical operation of scaling one number by another. Talking about today's engineering world, multiplication based operations are some of the frequently used Functions, currently implemented in many Digital Signal Processing (DSP) applications such as Convolution, Fast Fourier Transform, filtering and in Arithmetic Logic Unit (ALU) of Microprocessors. Since multiplication is such a frequently used operation, it's necessary for a multiplier to be fast and power efficient and so, development of a fast and low power multiplier has been a subject of interest over decades. The demand for high speed processing has been increasing as a result of expanding computer and signal processing applications. 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 delay and power consumption are very essential requirements for many applications. Two most common multiplication algorithms followed in the digital hardware are array multiplication algorithm and Booth multiplication algorithm. The computation time taken by the array multiplier is comparatively less because the partial products are calculated independently in parallel. Booth multiplication is another important multiplication algorithm.

In many DSP algorithms, the multiplier lies in the critical delay path and ultimately determines the performance of algorithm. The speed of multiplication operation is of great importance in DSP as well as in general processor. In the past multiplication was implemented generally with a sequence of addition, subtraction and shift operations. Minimizing power consumption and delay for digital systems involves optimization at all levels of the design. This optimization means choosing the optimum Algorithm for the situation, this being the highest level of design, then the circuit styles the topology and finally the technology used to implement the digital circuits. Depending upon the arrangement of the components, there are different types of multipliers available. Particular multiplier architecture is chosen based on the application.

The word "Vedic" is derived from the word "Veda" which means the store house of all knowledge. Vedic mathematics is mainly based on 16 Sutras (or aphorisms) dealing with various branches of mathematics like arithmetic, algebra, geometry etc.The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). 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.

The multiplier is based on an algorithm UrdhvaTiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. UrdhvaTiryakbhyam Sutra 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 and then, concurrent addition of these partial products can be done. Thus parallelism in generation of partial products and their summation is obtained using UrdhavaTiryakbhyam. The algorithm can be generalized for n x n bit number. The design starts first with Multiplier design that is 2x2 bit multiplier. Here, "UrdhvaTiryakbhyam Sutra" or "Vertically and Crosswise Algorithm" for multiplication has been effectively used to develop digital multiplier architecture.

2. Related Work

2.1 VEDIC MULTIPLIER:

The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware.

URDHVA TIRYAKBHYAM SUTRA

The multiplier is based on an algorithm UrdhvaTiryakbhyam (Vertical & Crosswise) of ancient Indian Vedic Mathematics. UrdhvaTiryakbhyam Sutra is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and crosswise". Urdhvatiryakbhyam Sutra 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 and then, concurrent addition of these partial products can be done. Thus parallelism in generation of partial products and their summation is obtained using rdhavaTiryakbhyam. Urdhvatiryakbhyam Sutra is first applied to the binary number system and is used to develop digital multiplier

architecture. This is shown to be very similar to the popular array multiplier architecture. 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.

Thus the multiplier will require the same amount of time to calculate the product and hence is independent of the clock frequency. The net advantage is that it reduces the need of microprocessors to operate at increasingly high clock frequencies. While a higher clock frequency generally results in increased processing power, its disadvantage is that it also increases power dissipation which results in higher device operating temperatures. By adopting the Vedic multiplier, microprocessors 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 has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other multipliers. Therefore it is time, space and power efficient. It is demonstrated that this architecture is quite efficient in terms of silicon area/speed. Firstly, least significant bits are multiplied which gives the least significant bit of the 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 product and the carry is added in the output of next stage sum obtained by the crosswise and vertical multiplication and addition of three bits of the two umbers from least significant position. Next, all the four bits are processed with crosswise 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. The same operation continues until the multiplication of the two MSBs to give the MSB of the product. For example, if in some intermediate step, we get 110, then 0 will act as result and 11 as the carry. It should be clearly noted that carry may be a multi-bit number. From here we observe one thing, as the number of bits goes on increasing, the required stages of carry and propagate also increase and get arranged as in ripple carry

adder.

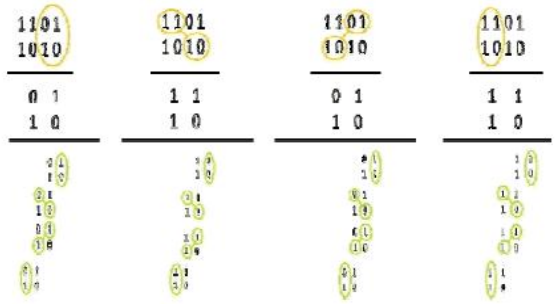


Fig 2 Better Implementation of UrdhvaTiryakbhyam for Binary numbers.

Vedic multiplier is faster than Montgomery modular multiplier. As the number of bits increases from 8x8 bits to 16x16 bits, the timing delay is greatly reduced for Vedic multiplier as compared to Montgomery modular multiplier. Vedic multiplier has the greatest advantage as compared to Montgomery multipliers over gate delays and regularity of structures.

In most of the DSP algorithms, the performance of the algorithm is based on the path delay of the multiplier. The speed of multiplication is very important in DSP as well as in general processors. In the early period, multiplications were implemented generally with a sequence of shift and add operations. The Montgomery modular multiplier has extra clock cycles to complete for completing one modular multiplication so it has more delay. This vedic multiplier has minimum cycles to complete one multiplication. Thus veidc multiplier shows the highest speed among conventional multipliers. It has this advantage than others to prefer a best multiplier.

3. Implementation

The design starts first with Multiplier design that is 2x2 bit multiplier. Here, "UrdhvaTiryakbhyam Sutra" or "Vertically and Crosswise Algorithm" for multiplication has been effectively used to develop digital multiplier architecture. This algorithm is quite different from the traditional method of multiplication that is to add and shift the partial products. This Sutra shows how to handle multiplication of a larger number (N x N, of N bits each) by breaking it into smaller numbers of size (N/2 = n, say) and these smaller numbers can again be broken into smaller numbers (n/2 each) till we reach multiplicand size of (2 x 2). Thus, simplifying the whole multiplication process. A simple digital multiplier architecture based on the UrdhvaTiryakbhyam (Vertically and Cross wise) Sutra of Vedic Mathematics is presented. An improved technique for low power and high speed multiplier of two binary numbers (16 bit each) is developed. An algorithm is proposed and implemented on 16nm CMOS technology. For Multiplier, first the basic blocks that are the 2x2 bit multipliers have been made and then, using these blocks, 4x4 block has been made and then using this 4x4 block, 8x8 bit block and then finally 16x16

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 12
August 2016

bit Multiplier has been made. The device selected for synthesis is Device family Spartan 3E, device is xc3s500, package fg320 with speed grade -4.So let's start from the synthesis of a 2x2 bit multiplier.

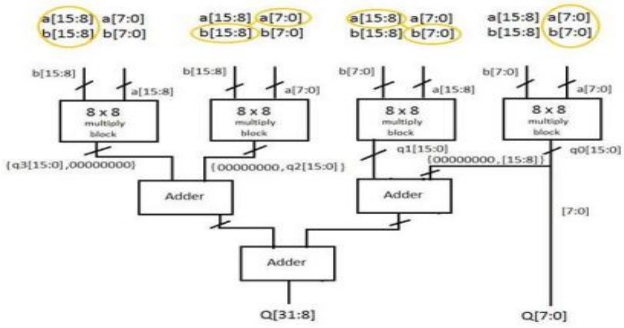
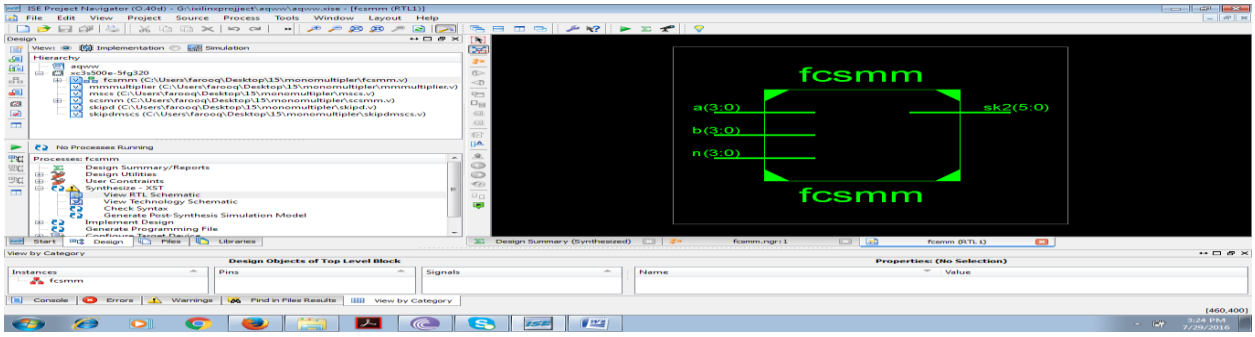
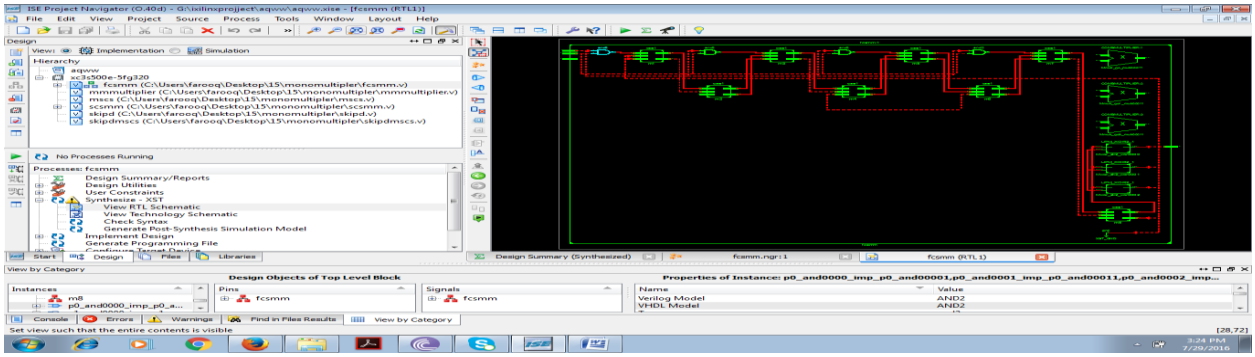


Fig 3 Block diagram of 16x16 multiply block.

4. Experimental Work





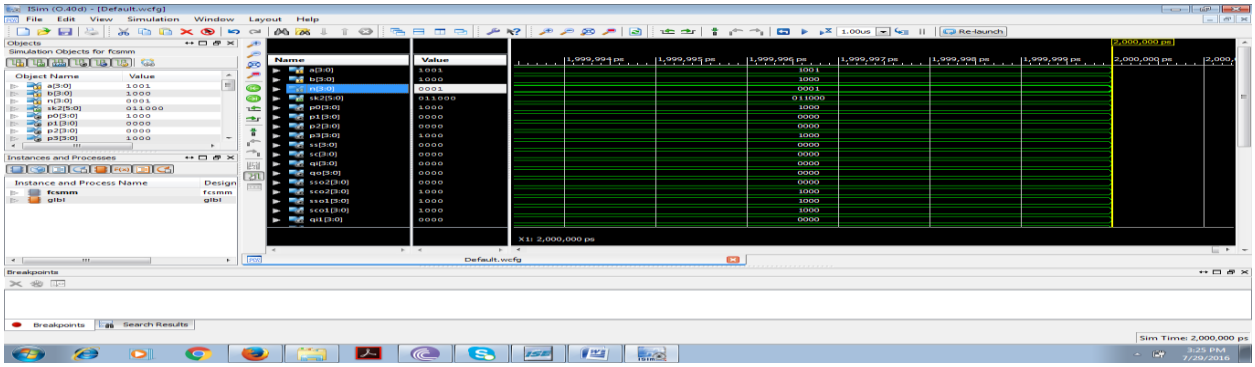


Fig 4 Simulation result of Montgomery MM.

From the above simulation result using Montgomery algorithm is clearly shown that, with two 32 bit input and a constant value. It generates a partial output by using carry skip and zero detector technique. It produced a power of 0.013mw.

5. Conclusion

FCS-based multipliers maintain the input and output operands of the Montgomery MM in the carry-save format to escape from the format conversion, leading to fewer clock cycles but larger area than SCS-based multiplier. To enhance the performance of Montgomery MM while maintaining the low hardware complexity, montgomery has modified the SCS based Montgomery multiplication algorithm and proposed a low-cost and high performance Montgomery modular multiplier. The vedic multiplier used one level CCSA architecture and skipped the unnecessary carry-save addition operations to largely reduce the critical path delay and required clock cycles for completing one MM operation. Experimental results of Montgomery multiplication shows that the produced output is 0.013mW hence in the proposed approach of vedic multiplier the result will be enhanced to reduce the power and increase the speed and decrease the delay.

6. References

[1]. Amberg.P, Pinckney.N, and Harris.D.M, "Parallel high-radix Montgomery multipliers," in Proc. 42nd Asilomar Conf. Signals, Syst., Comput. , Oct. 2008, pp. 772–776.

[2]. Bunimov.V, Schimmler.M, and Tolg.B, "A complexity-effective version of Montgomery's algorihm," in Proc. Workshop Complex. Effective Designs, May 2002.

[3]. Gang.F, "Design of modular multiplier based on improved Montgomery algorithm and systolic array," in Proc.1st Int. Multi-Symp. Comput. Comput.Sci,vol. 2. Jun. 2006, pp. 356–359.

[4]. Hong.J.H and Wu C.W, "Cellular-array modular multiplier for fast RSA public-key cryptosystem based on modified Booth's algorithm," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. , vol. 11, no. 3, pp. 474–484, Jun. 2003.

[5]. Han.J, Wang,W, Huang.W, Yu.Z , and Zeng.X, "Parallelization of radix-2 Montgomery multiplication on multicore platform," IEEE

Trans.Very Large Scale Integr. (VLSI) Syst. , vol. 21, no. 12, pp. 2325–2330, Dec. 2013.

[6]. Koblitz.N, "Elliptic curve cryptosystems,"Math. Comput. , vol. 48, no. 177, pp. 203–209, 1987.

[7]. Rivest.R.L, Shamir.A, and Adleman.L, "A method for obtaining digital signatures and public-key cryptosystems,"Commun. ACM, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[8]. Miller.V.S, "Use of elliptic curves in cryptography," in Advances in Cryptology . Berlin, Germany: Springer-Verlag, 1986, pp. 417–426.

[9]. Montgomery.P.L, "Modular multiplication without trial division," Math. Comput. , vol. 44, no. 170, pp. 519–521, Apr. 1985.

[10]. Kim.Y.S, Kang.W.S, and Choi.J.R, "Asynchronous implementation of 1024-bit modular processor for RSA cryptosystem," in Proc. 2nd IEEE Asia Pacific Conf. ASIC , Aug. 2000, pp. 187–190

[11]. Kuang.S.R, Wang.W.S, Chang.K.S, and Hsu.W.S, "Energy-efficient high-throughput Montgomery modular multipliers for RSA cryptosystems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. , vol. 21, no. 11, pp. 1999–2009, Nov. 2013.

[12]. C. McIvor, M. McLoone, and J. V. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques," IEE Proc.- Comput. Digit. Techn. , vol. 151, no. 6, pp. 402–408, Nov. 2004.

[13]. A. Miyamoto, N. Homma, T. Aoki, and A. Satoh, "Systematic design of RSA processors based on high-radix Montgomery multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. , vol. 19, no. 7, pp. 1136–1146, Jul. 2011.