IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676, p-ISSN: 2320–3331, Volume 8, Issue 2 (Nov.-Dec.), PP 57-61
www.iosrjournals.org

High Speed Signed multiplier for Digital Signal Processing Applications

[1]D. Naresh, [2]Dr. Giri Babu Kande
[1]PG Student (M.Tech VLSI), Dept. Of ECE, Vasireddy Venkatadri Ins. Tech., Nambur, Guntur, AP, India
[2] Professor & Head, Dept. Of ECE, Vasireddy Venkatadri Ins. Tech., Nambur, Guntur, AP, India

Abstract: *New VLSI circuit architectures for addition and multiplication modulo (2n - 1) and (2n + 1) are proposed that allow the implementation of highly efficient combinational and pipelined circuits for modular arithmetic. Fast multiplication is very ¬important in DSPs for convolution, Fourier transforms etc. A fast method for multiplication based on ancient Indian Vedic mathematics is proposed in this paper. Among the various methods of multiplications in Vedic mathematics, Urdhva tiryakbhyam is discussed in detail. Urdhva tiryakbhyam is a general multiplication formula applicable to all cases of multiplication. Apart from the Vedic another multiplier with parallel prefix adder architecture is well suited to realize fast end around carry adders used for modulo addition which is known as radix-8 booth multiplier is introduced .a comparative result is analyzed in the implementation.*

Keywords: *Radix Multiplier, Baugh-Wooley Multiplier, Carry Save array Multiplier.*

I. Introduction

Multiplication is an important fundamental function in arithmetic operations. Multiplication-based operations such as Multiply and Accumulate(MAC) and inner product are among some of the frequently used Computation- Intensive Arithmetic Functions (CIAF) currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform(FFT), filtering and in microprocessors in its arithmetic and logic unit . Since multiplication dominates the execution time of most DSP algorithms, so there is a need of high speed multiplier. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip.

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 multiplier circuit has been a subject of interest over decades. Reducing the time delay and power consumption are very essential requirements for many applications. This work presents different multiplier architectures. Multiplier based on Vedic Mathematics is one of the fast and low power multiplier.

Minimizing power consumption for digital systems involves optimization at all levels of the design. This optimization includes the technology used to implement the digital circuits, the circuit style and topology, the architecture for implementing the circuits and at the highest level the algorithms that are being implemented. Digital multipliers are the most commonly used components in any digital circuit design. They are fast, reliable and efficient components that are utilized to implement any operation. Depending upon the arrangement of the components, there are different types of multipliers available. Particular multiplier architecture is chosen based on the application.

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.

There have been many algorithms proposals in literature to perform multiplication, each offering different advantages and having tradeoff in terms of speed, circuit complexity, and area and power consumption. The multiplier is a fairly large block of a computing system. The amount of circuitry involved is directly proportional to the square of its resolution i.e. a multiplier of size n bits has n2 gates. For multiplication algorithms performed in DSP applications latency and throughput are the two major concerns from delay perspective. Latency is the real delay of computing a function, a measure of how long the inputs to a device are stable is the final result available on outputs. Throughput is the measure of how many multiplications can be performed in a given period of time; multiplier is not only a high delay block but also a major source of power dissipation. That is why if one also aims to minimize power consumption, it is of great interest to reduce the delay by using various delay optimizations

In algorithmic and structural levels, a lot of multiplication techniques had been developed to enhance the efficiency of the multiplier; which encounters the reduction of the partial products and/or the methods for their partial products addition, but the principle behind multiplication was same in all cases.

Vedic arithmetic is the ancient system of Indian mathematics which has a unique technique of calculations based on 16 Sutras (Formulae). "Urdhva-tiryakbyham" is a Sanskrit word means vertically and crosswise formula is used for smaller number multiplication.

Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers that are multiplied. It is the standard technique used in chip design, and provides significant improvements over the "long multiplication" technique.

The radix-8 Booth encoding reduces the number of partial products to which is more aggressive than the radix-4 Booth encoding. However, in the radix-8 Booth encoded modulo 2n-1 multiplication, not all modulo-reduced partial products can be generated using the bitwise circular-left-shift operation and bitwise inversion. Particularly, the hard multiple |+3X|2n-1 is to be generated by an n -bit end-around-carry addition of X and 2X.

When applying Booth encoding to a k-bit digit, the resulting encoded digit value is in the range [-2k□1,2k□1]. For radix 8, k=3 and the encoded multiplier digit is in the range [-4, 4]. The implementation of some values like -3 and 3 increases the complexity of the design.

II. IMPLEMENTATION

Vedic Multiplication:

To illustrate this multiplication scheme, let us consider the multiplication of two decimal numbers (325 * 738). Line diagram for the multiplication is shown in fig 1. 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 7result bit and all other bits act as carry for the next step. Initially the carry is taken to be zero. To make the methodology more clear, an alternate illustration is given with the help of line diagrams in figure where the dots represent bit „0□ or „1□.

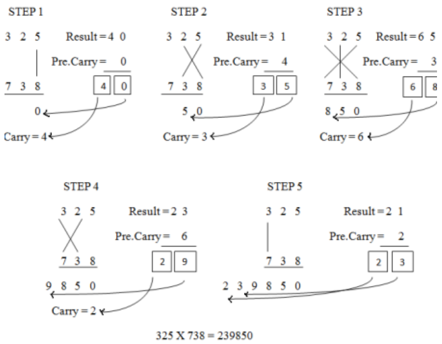


Figure 1 Multiplication of two decimal numbers by Urdhva Tiryakbhyam

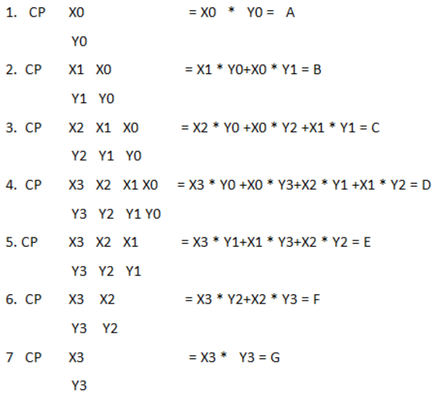
Algorithm for 4 x 4 bit Vedic multiplier Using Urdhva Tiryakbhyam (Vertically and crosswise) for two Binary numbers:

8 Parallel Computation Methodology:

Multiplication Process:

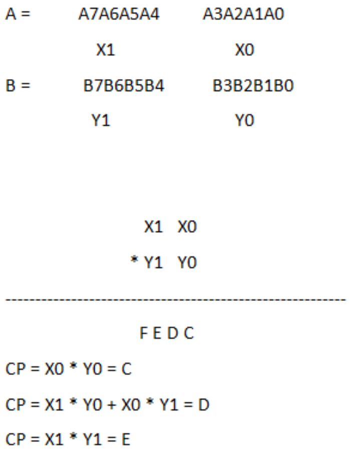
```
                X3   X2   X1   X0        Multiplicand

                Y3   Y2   Y1   Y0        Multiplier

--------------------------------------------------------------------

   H   G    F    E    D    C    B    A

  ---------------------------------------------------------------------

  P7   P6   P5   P4   P3   P2   P1   P0        Product

---------------------------------------------------------------------
```

Multiplication Process (continued):



```
1.  CP    X0                 = X0   *   Y0 =   A
           Y0
2.  CP    X1   X0            = X1 * Y0+X0 * Y1 = B
           Y1   Y0
3.  CP    X2   X1   X0       = X2 * Y0 +X0 * Y2 +X1 * Y1 = C
           Y2   Y1   Y0
4.  CP    X3   X2   X1 X0    = X3 * Y0 +X0 * Y3+X2 * Y1 +X1 * Y2 = D
           Y3   Y2   Y1 Y0
5. CP     X3   X2   X1       = X3 * Y1+X1 * Y3+X2 * Y2 = E
           Y3   Y2   Y1
6.  CP    X3    X2           = X3 * Y2+X2 * Y3 = F
           Y3    Y2
7   CP    X3                 = X3 *   Y3 = G
           Y3
```

Algorithms for 8 X 8 Bit Multiplication Using Urdhva Triyakbhyam (Vertically and crosswise) for two Binary numbers:



```
A =      A7A6A5A4      A3A2A1A0
           X1              X0
B =       B7B6B5B4      B3B2B1B0
          Y1              Y0


                X1   X0
              * Y1   Y0
-----------------------------------------------------------
               F E D C
CP = X0 * Y0 = C
CP = X1 * Y0 + X0 * Y1 = D
CP = X1 * Y1 = E
```

Where CP = Cross Product.

To illustrate the multiplication algorithm, let us consider the multiplication of two binary numbers $a_3a_2a_1a_0$ and $b_3b_2b_1b_0$. As the result of this multiplication would be more than 4 bits, we express it as... $r_3r_2r_1r_0$. Line diagram for multiplication of two 4-bit numbers is shown in Fig2.2. which is nothing but the mapping of the Fig.2.2. in binary system. For the simplicity, each bit is represented by a circle. Least significant bit r_0 is obtained by multiplying the least significant bits of the multiplicand and the multiplier. The process is followed according to the steps shown in Fig2.2.

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 numbers 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 is10again 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 bit (referred as rn) and 11 as the carry (referred as cn). It should be clearly noted that cn may be a multi-bit number.

Booth Implementation:

Let $X = \sum_{i=0}^{n-1} x_i \cdot 2^i$ and $Y = \sum_{i=0}^{n-1} y_i \cdot 2^i$ represent the multiplicand and the multiplier of the modulo 2^n-1 multiplier, respectively. The radix-8 Booth encoding algorithm can be viewed as a digit set conversion of four consecutive overlapping multiplier bits $y_{3i+2}y_{3i+1}y_{3i}(y_{3i-1})$ to a signed digit, $d_i, d_i \in [-4, 4]$, for $i = 0, 1, \ldots, \lfloor n/3 \rfloor$. The digit set conversion is formally expressed as

$$d_i = y_{3i-1} + y_{3i} + 2y_{3i+1} - 4y_{3i+2} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

Where

Modulo-reduced multiples for the radix-8 booth encoding

Table 1 Modulo Reduced Multiples of X

d_i	$\lvert d_i \cdot X\rvert_{2^n-1}$	d_i	$\lvert d_i \cdot X\rvert_{2^n-1}$
+0	$\underbrace{0\cdots0}_{n}$	−0	$\underbrace{1\cdots1}_{n}$
+1	X	−1	$\overline{X}$
+2	$CLS(X, 1)$	−2	$CLS(\overline{X},1)$
+3	$\lvert +3X\rvert_{2^n-1}$	−3	$\lvert -3X\rvert_{2^n-1}$
+4	$CLS(X, 2)$	−4	$CLS(\overline{X},2)$

Table I summarizes the modulo-reduced multiples of X for all possible values of the radix-8 Booth encoded multiplier digit, di, where CLS(X, J) denotes a circular-left-shift of X by j bit positions. Three unique properties of modulo 2^n-1 arithmetic that will be used for simplifying the combinatorial logic circuit of the proposed modulo multiplier design are reviewed here.

1) Property 1: The modulo 2^n-1 reduction of –X can be implemented as the -bit one's complementation of the binary word as follows:

$$|-X|_{2^n-1} = 2^n - 1 - X = \overline{X}. \quad \text{………….. (2)}$$

2) Property 2: For any nonnegative integer, the periodicity of an integer power of two over modulus can be stated as follows :

$$\left|2^{n\cdot s+i}\right|_{2^n-1} = \left|\left|2^{n.s}\right|_{2^n-1} \cdot \left|2^i\right|_{2^n-1}\right|_{2^n-1} = \left|2^i\right|_{2^n-1}. \quad \text{……….. (3)}$$

Property 2 ensures that the modulo 2^n-1 reduction of binary exponents can be implemented with no logic cost. As a corollary, the modulo 2^n-1 reduction of the product of a binary word X and an integer power of two, 2j, is equivalent to CLS(X,J). This property can be formally expressed as Property 3.

3) Property 3 : For j<n

$$\left|2^j X\right|_{2^n-1} = \sum_{i=0}^{n-j-1} x_i \cdot 2^{i+j} + \sum_{i=n-j}^{n-1} x_i \cdot 2^{i+j-n} = CLS(X,j). \quad \text{……………. (4)}$$

In Table IV, the modulo 2^n-1 reduction for $d_i \in \{\pm0, \pm1, \pm2, \pm4\}$ are replaced by simple bitwise inversion and bitwise circular-left-shift of X using Properties 1 and 3, respectively.

In contrast, the carry propagation addition of X and 2X is unavoidable in the computation of the hard multiple $|+3X|_{2^n-1}$. Numerous modulo 2^n-1 adders employing parallel-prefix structure with additional prefix operators for the end-around-carry

III. Results and Conclusions

The behavioral simulation waveforms for the Vedic and Radix-8 multiplier are shown in figure 2 and figure 3. The proposed Radix-8 Multiplier may be used in DSP applications because it gives better performance in terms of power, delay and PDP. The proposed adder based multiplier can be used in high speed application because of its less power dissipation and delay. Also, this multiplier has the minimum number of nonzero partial products based on the CSD number property. The number of add/subtract operations is further reduced through the use of bypass techniques. Thus, the complexity of the hardware implementation is dramatically reduced as compared to conventional methods, including modified Booth recoding and competing CSD recoding techniques. This approach achieves an overall speedup as well as reduced power consumption which is particularly critical in mobile multimedia applications. We analyzed the architecture and compared it to the previous fast architecture extracted in terms of area, speed, and power consumption. We found that the new architecture has a better performance than the previous ones.

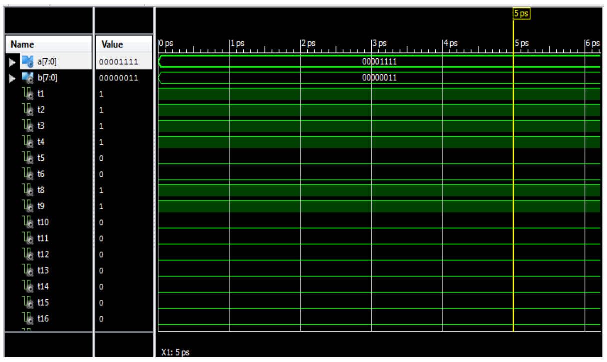


Figure 2 Behavioral simulation waveform for the Vedic Multiplier

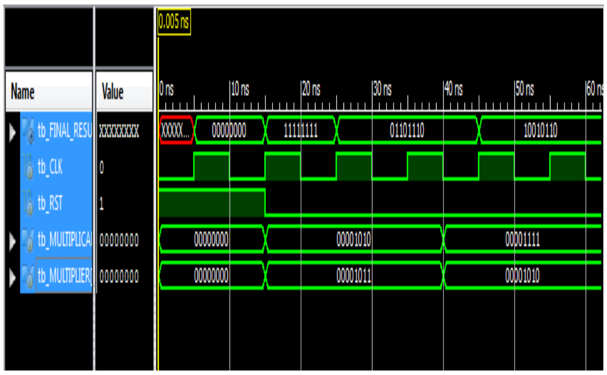


Figure 3 Behavioral simulation waveform for the Radix-8 Multiplier

Acknowledgements

The authors would like to thank the anonymous reviewers for their comments which were very helpful in improving the quality and presentation of this paper.

References

[1] Y K Yamanaka, T Nishidha, T Saito, M Shimohigashi, and K Shimizu, A. Hitachi Ltd., Tokyo " A 3.8-ns CMOS 16x16-b multiplier using complementary pass-transistor logic," IEEE Journal of Solid-State Circuits, vo1.25, no 2,pp.388-95, 1990.

[2] R Zimmermann and W Fichtner, Fellow, IEEELow-Power" Logic Styles: CMOS Versus Pass-Transistor Logic," IEEE Journal Of Solid-State Circuits, vol. 32, no. 7,pp.1079- 90,1997.

[3] D Markovic, B Nikolic, and V G Oklobdzija, "A general method in synthesis of pass-transistor circuits," Microelectr. J, vol. 31,pp. 991-8,2000.

[4] C H Chang, J Gu, and M Zhang, "A review of O.J8-mm full adder performances for tree structured arithmetic circuits," IEEE Trans. Very Large Scale Integr. (VLSI) Syst. Vol. 13 pp. 686-95, 2005.

[5] Massimo Alioto, Member, IEEE, and Gaetano Palumbo, Senior Member, IEEE, "Analysis and Comparision on Full Adder Block in Submicron Technology," IEEE Transactions On Very Large Scale Integration (VLSI) Systems, vol. 10, no. 6, pp. 806-23, 2002.

[6] L Sousa and R Chaves, "A universal architecture for designing effiCient modulo 2n+ 1 multipliers, " IEEE Trans. Circuits Syst.I: Regular Papers, vol. 52, pp.1166-78, 2005.

[7] T Oscal, C Chen, S Wang, and Y W Wu, "Minimization of switching activities of Partial Products for Designing LowPower Multipliers, "IEEE Transaction on Very Large Scale Integration (VLSI) Systems,vol. II, no. 3, pp. 418-433,2008.

[8] A. G. Dempster and M. D. Macleod, "Constant integer multiplication using minimum adders," IEE Proc.: Circuits, Devices, Syst., vol. 141, no. 5, pp. 407-413, Oct. 1994.

[9] Y. Voronenko and M. Puschel, "Multiplierless multiple constant multiplication," ACM Transactions on Algorithms, vol. 3, iss. 2, article no. 11, May 2007.

[10] Oscar Gustafsson, "A Difference Based Adder Graph Heuristic for Multiple Constant Multiplication Problems," Int'l Symp. Circuits and Syst., New Orleans, LA, pp. 1097-1100, May 2007.