RESEARCH ARTICLE OPEN ACCESS

Design of High Speed Architecture of Parallel MAC Based On Radix-2 MBA

Syed Anwar Ahmed, MD Salahuddin
HOD, Department of Electronics & Communication Engineering Shaaz College of engineering and technology
Asst. Prof., Department of Electronics & Communication Engineering Azad College of engineering and technology

ABSTRACT
The multiplier and multiplier-and-accumulator (MAC) are the essential elements of the digital signal processing such as filtering, convolution, transformations and Inner products. Parallel MAC is frequently used in digital signal processing and video/graphics applications. Fast multipliers are essential parts of digital signal processing systems. The speed of multiply operation is of great importance in digital signal processing as well as in the general purpose processors today, especially since the media processing took off. The MAC provides high speed multiplication and multiplication with accumulative addition. This paper presents a combined process of multiplication and accumulation based on radix-4 & radix-8 booth encodings. In this Paper, we investigate the method of implementing the Parallel MAC with the smallest possible delay. Enhancing the speed of operation of the parallel MAC is a major design issue. This has been achieved by developing a CLA adder for parallel MAC.

Keywords: Radix-4 and Radix-8 based Booth multiplier; modified booth algorithm (MBA); carry save adder (CSA) tree; carry look ahead adder(CLA); computer arithmetic; digital signal processing (DSP); multiplier and-accumulator (MAC); Adders.

I. Introduction

In this paper, the various parallel MAC architectures is studied and then a design of parallel MAC based on modified booth encodings such as radix-4 & radix-8 booth encoder and some final adder such as carry look ahead adder(CLA) is implemented and their performance characteristics is compared. A Digital multiplier is the fundamental component in general purpose microprocessor and in DSP [1], [2]. Compared with many other arithmetic operations multiplication is time consuming and power hungry. Thus enhancing the performance and reducing the power dissipation are the most important design challenges for all applications in which multiplier unit dominate the system performance and power dissipation. The one most effective way to increase the speed of a multiplier is to reduce the number of the partial products. Although the number of partial products can be reduced with a higher radix booth encoder, but the number of hard multiples that are expensive to generate also increases simultaneously.

In the majority of digital signal processing (DSP) applications the critical operations are the multiplication and accumulation. Real-time signal processing requires high speed and high throughput Multiplier-Accumulator (MAC) unit that consumes low power, which is always a key to achieve a high performance digital signal processing system.

In general, a multiplier uses Booth's algorithm and array of full adders (FAs), or Wallace tree instead of the array of FA's., i.e., this multiplier mainly consists of the three parts: Booth encoder, a tree to compress the partial products such as Wallace tree, and final adder. Because Wallace tree is to add the partial products from encoder as parallel as possible, its operation time is proportional to, where is the number of inputs.

To increase the speed and performance, many parallel MAC architectures have been proposed. Parallelism in obtaining partial products is the most common technique used in the above implemented architecture. There are two common approaches that make use of parallelism to enhance the multiplication performance. The first one is reducing the number of partial product rows and second one is the carry-save-tree technique to reduce multiple partial product rows as two "carry-save" redundant forms. An architecture was proposed in [4] to provide the tact to merge the final adder block to the accumulator register in the MAC operator to provide the possibility of using two separate N/2-bit adders instead of one N-bit adder to accumulate the N–bit MAC results. The most advanced types of MAC has been proposed by Elguibaly [8] in which accumulation has been combined with the carry save adder (CSA) tree that compresses partial products and thus reduces the critical path. Later on a new architecture for a high-speed MAC is proposed by Seo and Kim [9]. The difference between the two is that the latest one carries out the accumulation by feeding back the final CSA output rather than the final adder results.

The rest of the paper is organized as follows. In section second, an introduction to the general MAC is given along with basic MAC algorithms. In section third, entire process of parallel MAC based on radix-4 and radix-8 booth encodings [7] is explained. Section four shows implementation result and the characteristics of parallel MAC based on both of the booth encodings. Finally, the conclusion will be given in section five in which we provide summary of our proposed approach and discuss scope of future extensions.

II. GENERAL MAC STRUCTURE

In this section, we discuss basic MAC operation. Basically, multiplier operation can be divided into three operational steps. The first one is booth encoding to generate the partial products. The second one is adder array or partial product compression and the last one is final addition in which final multiplication result is obtained.

If the multiplication process is extended to accumulate the multiplied result, then MAC consists of four steps. General hardware architecture for MAC is shown in Figure 1. It executes the multiplication operation by multiplying input multiplier X and input multiplicand Y. After that current multiplication result is added to the previous multiplication result Z as accumulation step.

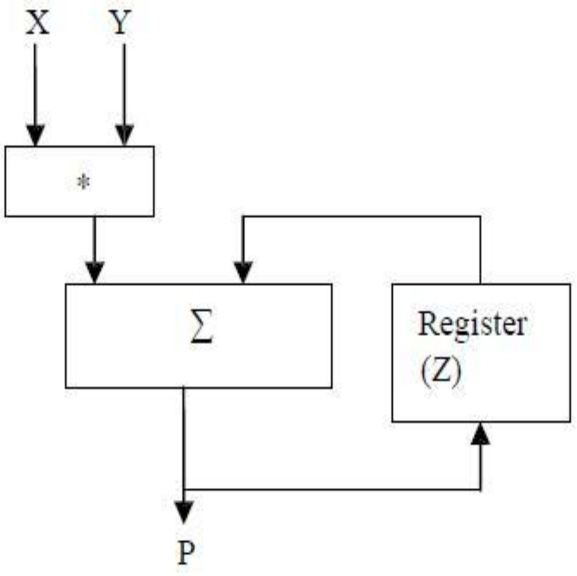


Fig. 1. Hardware architecture of general MAC

The N-bit 2's compliment number can be expressed as

$$X = -2^{N-1}x_{N-1} + \sum_{i=0}^{N-2} x_i 2^i, \quad x_i \in \{0,1\}$$ …………………………(1)

If (1) is expressed in base -4 type redundant sign digit form in order to apply the radix-2 booths algorithm,

$$X = \sum_{i=0}^{N/2-1} d_i 4_i$$ …………………………….(2)

$$d_i = -2x_{2i+1} + x_{2i} + x_{2i-1}$$ ………………………(3)

If (2) is used, multiplication can be expressed as

$$X*Y = \sum_{i=0}^{N/2-1} d_i 2^{2i} Y$$ ……………………………(4)

If these equations are used, the afore-mentioned multiplication-accumulation results can be expressed as

$$P = X*Y + Z$$

$$P = \sum_{i=0}^{N/2-1} d_i 2^i Y + \sum_{i=0}^{2N-1} z_i 2^i$$ …..………………..

(5)

3.1 DERIVATION OF MAC ARITHMETIC

If an operation of multiplication of two N-bit numbers and accumulating them into a 2N-bit number is considered, then the critical path is totally determined by accumulation operation. If pipeline scheme is applied for each step, then the delay of last accumulator must be reduced by combining the accumulation operation with the CSA function.

The aforementioned concept is applied to (5) to express the presented MAC arithmetic. Then, the multiplication would be transferred to a hardware architecture that complies with the proposed concept, in which the feedback value for accumulation will be modified and expanded for the new MAC.

First, if the multiplication is decomposed and rearranged, it becomes

$$X*Y = d_0 2Y + d_1 2^2 Y + d_2 2^4 Y + \dots d_{\frac{N}{2}-1} 2^{N-2} Y$$ …………………… (6)

If (6) is divided into the first partial product, sum of the middle partial products, and the final partial product, it can be re expressed as (7). The reason for separating the partial product addition as (7) is that three types of data are fed back for accumulation, which are the sum, the carry, and the pre added results of the sum and carry from lower bits.

$$X*Y = d_0 2Y + \sum_{i=1}^{N/2-2} d_i 2^{2i} Y + d_{\frac{N}{2}-1} 2^{N-2} Y$$ ……..…

(7)

Now, the proposed concept is applied to Z in

(5), if Z is first divided into upper and lower bits and rearranged, (8) will be derived. The first term of the right-hand side in (8) corresponds to the upper bits. It is the value that is fed back as the sum and the carry. The second term corresponds to the lower bits and is the value that is fed back as the addition result for the sum and carry

$$Z=\sum^{N}{}_{i=0}{}_{-1} z_i 2^i+\sum^{2}{}_{N-1} z_i 2^i \qquad \text{………………………(8)}$$

The second term can be separated further into the carry term and sum term as

$$\sum_{i=N}^{2N-1} z_i \quad = \sum_{i=0}^{N-1} z_{N+i} \qquad = \sum_{i=0}^{N-2}(c_i+s_i)2^i 2^N \qquad \text{……………..(9)}$$

Thus (8) is finally separated into three terms as

$$Z=\sum^{N}{}_{i=0}{}_{-1} z_i 2^i+\sum^{N}{}_{-2} c_i 2^i 2^N+\sum^{N}{}_{-2} s_i 2^i 2^N \qquad \text{………(10)}$$

If (7) and (10) are used, the MAC arithmetic in (5) can be expressed as

$$P=(d_0 2Y+\sum_{2}{}_{i=0}^{\frac{N}{}-2} d_i\, 2^{2i}Y+\ d_{\frac{N}{2}-1} 2^{N-}Y)+(\sum_{i=0}^{N-1}$$

……………. (11)

If each term of (11) is matched to the bit position and rearranged, it can be expressed as (12), which is the final equation for the proposed MAC. The first parenthesis on the right is the operation to accumulate the first partial product with the added result of the sum and the carry. The second parenthesis is the one to accumulate the middle partial products with the sum of the CSA that was fed back. Finally, the third parenthesis expresses the operation to accumulate the last partial product with the carry of the CSA.

$$P=\left(d_0 2Y+\sum^{N}{}_{=0}{}_{-1} z_i 2^i\right)+\left(\sum_{\frac{i=0}{2}}^{N} d_i\, 2^{2i}Y+\sum_{i=0}^{N-2} c_i\, 2^i 2^N\right)$$

$$+\left(d_{\frac{N}{2}-1} 2^{N-2}Y+\sum_{i=0}^{N-2} s_i\, 2^i 2^N\right) \qquad \text{………..(12)}$$

3.2 MAC ARCHITECTURE

The hardware architecture of the proposed MAC satisfying the aforementioned equations is shown in Figure 2. The n-bit MAC inputs, X and Y, are converted into group of partial products by passing through the booth encoders based on radix-4 and radix-8 booth algorithms.

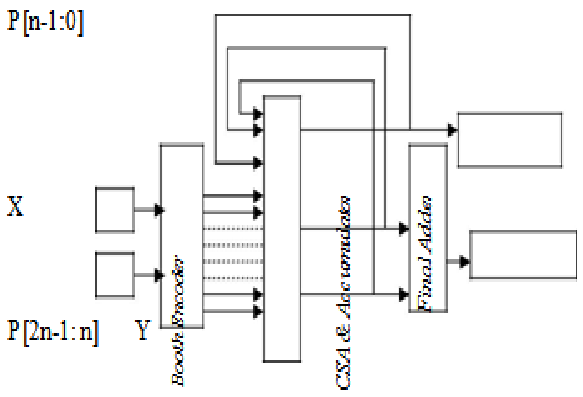


Fig. 2. Hardware architecture of proposed MAC

After that accumulation is carried out along with addition of partial products in CSA and Accumulator. As a result n-bit S, C and, Z are generated. These three values are fed back and used for accumulation operation. P [2n-1: n] is generated by adding S and C in the final adder which would be either CLA (carry look ahead adder) or RCA (ripple carry adder) or Kogge stone adder. At last, by combining P [2n-1: n] with P [n-1: 0] we get the final result.

3.3 BOOTH ENCODERS

The modified Booth's algorithm based on a radix-4, generally called Booth-2 [7] is the most popular approach for implementing fast multipliers using parallel encoding [1]. It uses a digit set {0, ±1, ±2} to reduce the number of the partial products to n'= [(n+1)/ 2]. Radix- 4 encoding start by appending a zero to the right of multiplier LSB. Triplets are taken beginning at position x –1 and continuing to the MSB with one bit overlapping between adjacent triplets.

Table 1. Radix-4 booth encoding

Yn+1Yn	Yn-1	Zn	Partial Product
0 0	0	0	0
0 0	1	1	1× Multiplicand
0 1	0	1	1× Multiplicand
0 1	1	2	2× Multiplicand
1 0	0	-2	-2×
1 0	1	-1	1× Multiplicand
1 1	0	-1	1× Multiplicand
1 1	1	0	0

This recoding scheme applied to a parallel multiplier halves the number of partial products so the multiplication time and the hardware requirements decrease. Radix-8 recoding [5], [7] applies the same algorithm as radix-4, but now in this we take quartets of bits instead of triplets.

To implement the radix-4 table in hardware three different types of encoded signals are generated, this signals are then provided as an inputs to the partial product generator, thus partial product generator produces the partial product based on the encoded signals. The truth table can be realized using k-maps.

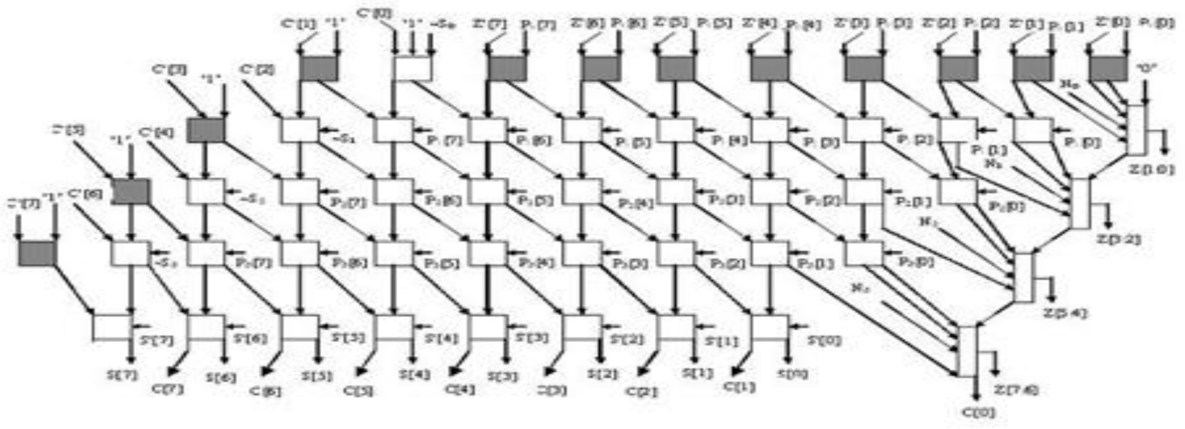
Table 2. Radix-4 booth encoding in hardware

Yn+1	Yn	Yn-1	Zn	X_a	X_b	neg
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	2	0	1	0
1	0	0	-2	0	1	1
1	0	1	-1	1	0	1
1	1	0	-1	1	0	1
1	1	1	0	0	0	0

The Booth-3 scheme is based on a radix-8 encoding to reduce this number to n' = [(n+1)/3]. All digit sets {0, ±1, ±2, ±3, ±4} are obtained by simple shifting and complementary operations, except generation of the multiple 3X, which is computed by an adding and shifting operation, 3X = 2X+X and -3X can be generated by complement 3X. Radix-4 booth encoding that generates partial products is shown in Table 1 respectively.

3.4 PROPOSED CSA AND ACCUMULATOR ARCHITECTURE

The architecture of CSA and accumulator based on radix-4 modified booth encoder is shown in Figure 3 which performs 8×8 bit MAC operation. In Figure 3, Ni is to compensate the 1′s complement number into 2′s complement, Si is to specify the sign expansion of the corresponding partial product and (~Si) is 1′s complement of Si. S[i] and C[i] corresponds to the *i*th bit of the feedback sum and carry whereas Z[i] is the *i*th bit of the sum of lower bits for each partial product. S′[i], C′[i] and Z′[i] are the previous result for accumulation. In addition, Pj[i] is the *i*th bit of *j*th partial product.



Proposed CSA consists of full adders, half adders and carry look ahead adders as shown in Figure 3. In this, series of CLA adders are used to generate the lower 8-bits of the final result. The white square in Figure 3 represents an FA and the black square is a half adder (HA). The rectangular symbol with five inputs is a 2-bit CLA with a carry input.

The critical path in this CSA is determined by the 2-bit CLA. It is also possible to use FAs to implement the CSA without CLA. However, if the lower bits of the previously generated partial product are not processed in advance by the CLAs, the number of bits for the final adder will increase. When the entire multiplier or MAC is considered, it degrades the performance.

3.5 FINAL ADDER

Final adders are digital circuits that compute the addition of variable binary strings of equivalent or different size. The carry look-ahead adder is used for final addition of last two rows obtained from the carry save adder tree. In carry look-ahead adder, carry signal is calculated in advance based on input signals. The result is a reduced carry propagation time.

III. EXPERIMENTAL RESULTS

In this section, the MAC is implemented and analyzed. The 8*8 bit parallel MAC based on both the booth encodings (i.e. Radix-4 booth encoding and Radix-8 booth encoding) and some final adders (such as CLA) is designed in Verilog and the functionalities of the algorithms are verified by cadence simulator and the layouts of the circuits are designed using cadence Encounter tool.

The proposed MAC was implemented in register-transfer level (RTL) using verilog hardware descriptive language (HDL). The designed circuits were synthesized using the design compiler from Cadence. The medium library (180nm) for manufacturing digital semiconductors was used.

In table 3, the gate count for the proposed MAC and [8] is compared, as shown in the table 3, the gate count for our architecture is slightly smaller than that in [8].

Table 3. Estimation of gate size by synthesis

nm	CSA		Booth Encoder	Final Adder		Total	
	[8]	Proposed		[8]	Proposed	[8]	proposed
180	1581	844	160	120	132	2510	2091

The power consumption of designed circuits is measured and summarized in table 4.

Table 4. Power consumption

nm	CSA	Booth Encoder	Final Adder	Total
180	43628nW	348nW	3916nW	41251nW

IV. CONCLUSION

An 8x8 multiplier-accumulator (MAC) is presented in this work. A Radix 4 Modified Booth multiplier circuit is used for MAC architecture. Compared to other circuits, the Booth multiplier has the highest operational speed and less hardware count. This algorithm is competitive with other more commonly used algorithms when used for high performance implementations. Secondarily, this thesis has shown that algorithms based upon the Radix-4 Booth partial product method are distinctly superior in performance when compared to Radix-8 Booth encoded method. This work can be utilized in any of the following such as in DSP applications, Numerical co-processor, Calculators, Filtering, and Modulation & Demodulation etc. As the summation networks and partial product generation logic results in higher delay and most of the area of a MAC. Thus in future, some more techniques and advancement needs to be done which further improves the performance of MAC. Also some measures should be taken which minimize the area consumption.

REFERENCES

[1] J. J. F. Cavanagh, “Digital Computer Arithmetic”, New York: McGraw- Hill, 1984.
[2] O. L. MacSorley, “High speed arithmetic in binary computers,” Proc. IRE, vol. 49, pp. 67–91, Jan. 1961.
[3] C. S. Wallace, “A suggestion for a fast multiplier”, IEEE Trans. Electron Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.
[4] Fayed and M. Bayoumi, “A merged multiplier-accumulator for high speed signal processing application”, Proc. ICASSP, vol. 3, pp. 3212–3215, 2002.
[5] Yajuan He and Chip-Hong Chang, “A New redundant binary booth encoding for fast 2n-bit multiplier design”, IEEE Transaction on circuit and systems, Vol. 56, No.6, June 2009.
[6] R. Cooper, “Parallel architecture modified Booth multiplier”, Proc. Inst. Electr. Eng. G, vol. 135, pp. 125–128, 1988.
[7] Marc Hunger and Daniel Marienfeld, “New self checking booth multiplier”, Int. J. Appl. Math, Comput. Sci., Vol.18, No.3, 319–328, 2008.
[8] F. Elguibaly, “A fast parallel multiplier–

accumulator using the modified Booth algorithm", IEEE Trans. Circuits Syst., vol. 27, no. 9, pp. 902–908, Sep. 2000.

[9] Young-Ho Seo and Dong-Wook Kim, "A New VLSI Architecture of Parallel Multiplier Accumulator Based on Radix-2 Modified Booth Algorithm", IEEE trans. on VLSI Systems, Vol.18 No. 2, Feb. 2010. R.

[10] E. Ladner and M. J. Fischer, "Parallel Prefix Computation", Journal of the ACM, vol.27, No.4, October 1980, 831-838.

[11] Virtex generation configuration user guide, Virtex-II Pro™ FPGA families, DS083- 1 (v1.0) January 31, 2002, available at www.xilinx.com.

MD SALAHUDDIN received the B.Eng. in Electronics and Communication Engineering from Osmania University, Hyderabad, AP, INDIA in 2010, and the M.Tech. in VLSI System Design from JNT University, Hyderabad, AP, INDIA in 2012. His research interests include Wireless Communication, Microelectronics, Analog VLSI, Image processing, Embedded Systems.

SYED ANWAR AHMED received the B.Tech. in Electronics and & Communication Engineering from Jawaharlal Technological University, Hyderabad, AP, INDIA in 2008, and the M. Tech. in VLSI System Design from JNT University, Hyderabad, AP, INDIA in 2012. His research interests include Analog Electronics, Electromagnetic Theory, Analog VLSI, Image processing, Embedded Systems.