Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 7, July 2016, pg.304 – 311

NEW REALIZATION AND IMPLEMENTATION BOOTH MULTIPLIER

Farheen Sultana
M. Tech, PG Student
faru2k8@gmail.com

A.Venkat Reddy
Associate Professor
venkat7641@gmail.com

Abstract— In this paper, we proposed a new architecture of multiplier-and-accumulator (MAC) for high-speed arithmetic. By combining multiplication with accumulation and devising a hybrid type of carry save adder (CSA), the performance was improved. Since the accumulator that has the largest delay in MAC was merged into CSA, the overall performance was elevated. The proposed CSA tree uses 1's-complement-based radix-2 modified Booth's algorithm (MBA) and has the modified array for the sign extension in order to increase the bit density of the operands.

The CSA propagates the carries to the least significant bits of the partial products and generates the least significant bits in advance to decrease the number of the input bits of the final adder. Also, the proposed MAC accumulates the intermediate results in the type of sum and carry bits instead of the output of the final adder, which made it possible to optimize the pipeline scheme to improve the performance. The proposed architecture was synthesized with and performance of the entire calculation. Because the multiplier requires the longest delay among the basic operational blocks in digital system, the critical path is determined by the multiplier, in general. For high-speed multiplication, the modified radix-4 Booth's algorithm (MBA) is commonly used.

I. INTRODUCTION

WITH the recent rapid advances in multimedia and communication systems, real-time signal processings like audio signal processing, video/image processing, or large capacity data processing are increasingly being demanded. The multiplier and multiplier-and-accumulator (MAC) [1] are the essential elements of the digital signal processing such as filtering, convolution, and inner products. Most digital signal processing methods use nonlinear functions such as discrete cosine transform (DCT) [2] or discrete wavelet transform (DWT) [3].

Because they are basically accomplished by repetitive application of multiplication and addition, the speed of the multiplication and addition arithmetics determines the execution speed number of inputs. It uses the fact that counting the number of 1's among the inputs reduces the number of outputs into. In real implementation, many (3:2) or (7:3) counters are used to reduce the number of outputs in each pipeline step. The most effective way to increase the speed of a multiplier is to reduce the number of the partial products because multiplication proceeds a series of additions for the partial products. To reduce the number of calculation steps for the partial products, MBA algorithm has been applied mostly where Wallace tree has taken the role of increasing the speed to add the partial products. To increase the speed of the MBA algorithm, many parallel multiplication architectures have been researched [11]–[13]. Among them, the architectures based on the Baugh–Wooley algorithm (BWA) have been developed and they have been applied to various digital filtering calculations [14]–[16].

One of the most advanced types of MAC for general-purpose digital signal processing has been proposed by Elguibaly [17].

It is an architecture in which accumulation has been combined with the carry save adder (CSA) tree that compresses partial products. In the architecture proposed in [17], the critical path was reduced by eliminating the adder for accumulation and de creasing the number of input bits in the final adder. While it has a better performance because of the reduced critical path compared to the previous MAC architectures, there is a need to improve the output rate due to the use of the final adder results for accumulation. An architecture to merge the adder block to the accumulator register in the MAC operator was proposed in [18] to provide the possibility of using two separate/2-bit adders instead of one-bit adder to accumulate the bit MAC results. Recently, Zicari proposed an architecture that took a merging technique to fully utilize the 4–2 compressor [19]. It also took this compressor as the basic building blocks for the multiplication circuit. In this paper, a new architecture for a high-speed MAC is proposed. In this MAC, the computations of multiplication and accumulation are combined and a hybrid-type CSA structure is proposed to reduce the critical path and improve the output rate. It uses MBA algorithm based on 1's complement number system. A modified array structure for the sign bits is used to increase the density of the operands. A carry look-ahead adder (CLA) is inserted in the CSA tree to reduce the number of bits in the final adder. In addition, in order to increase the output rate by optimizing the pipeline efficiency, intermediate calculation results are accumulated in the form of sum and carry instead of the final adder outputs.

II. Significance of MAC

Digital Signal Processing (DSP) is used in a wide range of applications such as speech and audio coding, image processing and video, pattern recognition, sonar and so on. In real time Very Large Scale Integration (VLSI) implementation of the DSP instruction, the system requires hardware architecture which can process input signal samples as they received. Most of the DSP computation involves the use of multiply and multiply accumulate operations and therefore Multiplier Accumulator (MAC) unit is very important in DSP applications.
A Digital multiplier is the fundamental component in general-purpose microprocessors and in digital signal processors. In addition, the multiply-accumulate (MAC) operation is very prevalent in many scientific and engineering applications. It is very easy to find such operation in signal processing algorithms and matrix arithmetic. The multiplier speed is usually the bottleneck that determines the depth of pipelining in the ALU or how fast a processor can run. The current trend in ALU design is to implement the addition and multiplication operations using one hardware component.
MAC = multiplication + Accumulation

2.1 Multiplier Background:

Multiplier circuits are found in virtually every computer, cellular telephone, and digital audio/video equipment. In fact, essentially any digital device used to handle speech, stereo, image, graphics, and multimedia content contains one or more multiplier circuits. The multiplier circuits are usually integrated within microprocessor, media co-processor, and digital signal processor chips. These multipliers are used to perform a wide range of functions such as address generation, Discrete Cosine Transformations (DCT), Fast Fourier Transforms (FFT), multiply-accumulate, etc. Thus multipliers play a critical role in processing audio, graphics, video, and multimedia data.
Multiplication operation is carried out by using different types of multipliers.
1. Serial multiplier
2. Parallel multiplier

i) Booth encoding
ii) Modified booth encoding

2.2 Binary serial multiplier:

A Binary serial multiplier is an electronic hardware device used in digital electronics or a computer or other electronic device to perform rapid multiplication of two numbers in binary representation. It is built using binary adders. The basic hardware required for binary multiplier is shown in Fig. 2.1.
The rules for binary multiplication can be stated as follows:
i) If the multiplier digit is a 1, the multiplicand is simply copied down and represents the product.
ii) If the multiplier digit is a 0 the product is also 0.
For designing a multiplier circuit, the following four points should be kept in mind.
1. It should be capable of identifying whether a bit is 0 or 1?
2. It should be capable of shifting left partial products.
3. It should be able to add all the partial products to give the products as sum of partial products.
4. It should examine the sign bits. If they are alike, the sign of the product will be a positive, if the sign bits are opposite product will be negative. The sign bit of the product stored with above criteria should be displayed along with the product.
From the above discussion it is clear that it is not necessary to wait until all the partial products have been formed before summing them. In fact the addition of partial product can be carried out as soon as the partial product is formed.

2.3 Parallel Multiplier:

Parallel multipliers can be implemented using two methods to increase the speed of the multiplier. They are
1. Booth encoding Algorithm
2. Modified Booth Algorithm (MBA)

Booth multiplication is a technique that allows for smaller, faster multiplication circuits, by recoding the numbers. It is the standard technique used in chip design and provides significant improvements over long multiplication technique.
Shift and Add:
A standard approach that might be taken by a novice to perform multiplication is "shift and add", or normal "long multiplication". That is, for each column in the multiplier, shift the multiplicand to the appropriate number of columns and multiply it by the value of the digit in that column of the multiplier, to obtain a partial product. The partial products are then added to obtain the final result.
With this system, the number of partial products is exactly the number of columns in the multiplier.

Unlike RCA, CLA & CSL adders, the Carry Save Adder realizes concurrent addition of multiple operands, which is a basic requirement of multiplication. To increase the speed of the addition process many times, Carry Save Adder architectures are used.
In this technique, the carry output from the bit i during step j is applied to carry input for bit i+1 during next step j+1. After addition of product components in the last row, one more step is required in which the carries are allowed to ripple from the least to most significant bit. This technique does not save any hardware but it reduces the propagation delay substantially.

2.4 Hybrid Adder:

Hybrid adder is a combination of any two adders. It is used in high speed applications. The hybrid adder generally consists of two carry look-ahead adders and a multiplexer. Adding two n-bit numbers with a hybrid adder is done with two adders (therefore two carry look ahead adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known. The propagation delay is less for hybrid adder and at the same time it occupies larger area compared to the other adders.

2.5 Overview of MAC:

In this section, basic MAC operation is introduced. A multiplier can be divided into three operational steps. The first is radix-2 Booth encoding in which a partial product is generated from the multiplicand X and the multiplier Y. The second is adder array or partial product compression to add all partial products and convert them into the form of sum and carry. The last is the final addition in which the final multiplication result is produced by adding the sum and the carry. If the process to accumulate the multiplied results is included, a MAC consists of four steps, as shown in

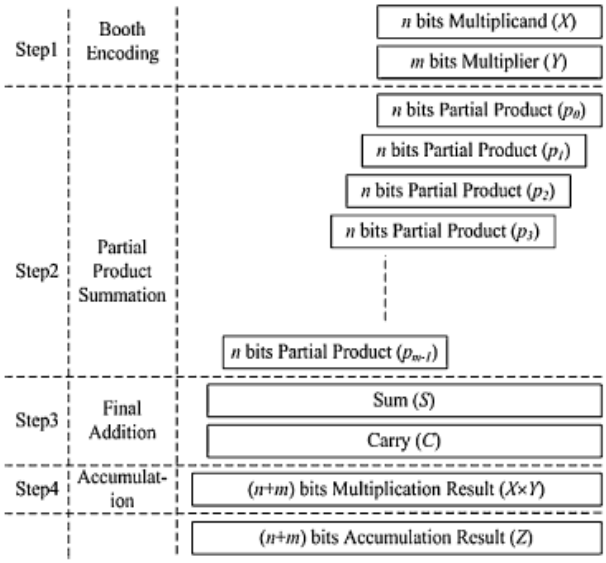

Fig 1 hardware architecture of this MAC

General hardware architecture of this MAC is shown in Fig. 1. It executes the multiplication operation by multiplying the input multiplier X and the multiplicand Y. This is added to the previous multiplication result Z as the accumulation step.

$P = X \times Y + Z$

The N-bit 2's complement binary number X can be expressed as

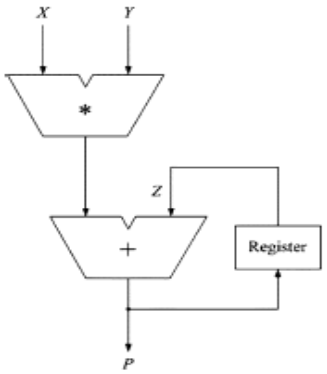

Fig. 2. Hardware architecture of general MAC

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

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

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

Each of the two terms on the right-hand side of (5) is calculated independently and the final result is produced by adding the two results. The MAC architecture implemented by (5) is called the standard design.

If N-bit data are multiplied, the number of the generated partial products is proportional to N. In order to add them serially, the execution time is also proportional to N. The architecture of a multiplier, which is the fastest, uses radix-2 Booth encoding that generates partial products and a Wallace tree based on CSA as the adder array to add the partial products. If radix-2 Booth encoding is used, the number of partial products, i.e., the inputs to the Wallace tree, is reduced to half, resulting in the decrease in CSA tree step. In addition, the signed multiplication based on 2's complement numbers is also possible. Due to these reasons, most current used multipliers adopt the Booth encoding.

2.7 Parallel MAC structure:

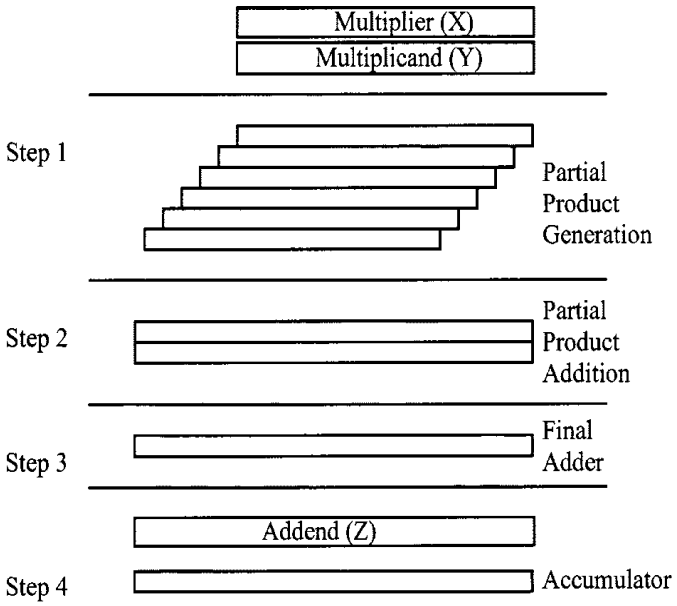

Fig : 3 general structure of a parallel MAC

the general structure of a parallel MAC for multiplying two numbers X and Y and adding the result to Z. The partial products in the figure can be generated using any multiplication algorithm using bit-serial, serial-parallel, or full-parallel techniques, as explained below. The required tasks, organized from fastest to slowest, are the following.

1) Partial-Product Generation: This can be achieved using several techniques such as the BWA, the Booth algorithm (BA), or the MBA. For an n-bit multiplier, the number of summands is n for BWA, <= n/2 for BA, and n/2 for MBA. In addition to the encoding step, BA and MBA algorithms also require generation of the two's complement of the multiplier which introduces extra delay. The delay for two's-complement generation is not trivial, but has been consistently neglected in most of the proposed designs.

2) Partial-Product Addition: This is done using a carry ripple adder for serial-parallel multipliers. For parallel multipliers, the addition is accomplished using carry-save techniques, Wallace trees, or summand skip. However, the last two techniques require irregular wiring and extra hardware.

3) Final Adder: When the number of partial products is reduced to sum and carry words, a final adder is required to generate the multiplication result. The number of bits of the final adder is the sum of the number of bits of the

multiplier and multiplicand. Thus, the data path width is usually doubled and the delay of this stage is most severe. In this thesis, we use a mix of 2-bit and 4-bit CLA's to reduce the delay and area requirements.

4) Accumulator: The final adder produces a double-precision result of 2n bits that must be added to the accumulator content, which are also 2n-bits wide. Typical microprocessors will complete the multiplication operation and follow that with a double-precision accumulation operation. Needless to say, this is delay intensive, since both the multiplier and accumulator will each have a delay that is almost 2n times the delay of a one-bit full adder. Recently, it was realized that the MAC operation can be merged to make the MAC operation take as much delay as a regular multiply operation.

To increase the speed of the MBA algorithm, several parallel MAC architectures have been researched.

III. SIMULATION RESULTS

This chapter gives the simulation results of all parallel MAC architectures.

3.1 FPGA Simulation results:

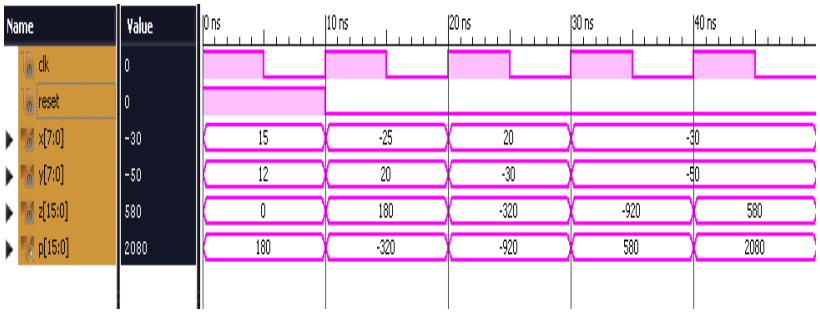

Fig. 4. simulation waveform of the standard design

In the above figure, the inputs are chosen randomly. Initially the multiplicand(x) and the multiplier(y) are given as 15, 12 respectively. Z[15:0] is the previous MAC result, and initially it is set to 0. Current MAC result is stored in p(p = x * y + z), so 15 * 12 + 0 = 180 is stored in the p register. For the second clock cycle, the values of x, y chosen as -25 and 20 respectively. Now previous MAC result 180 is stored in z register. So p = -25 * 20 + 180 = -320 is stored in the p register. This process repeats.

3.2 Elguibaly's parallel MAC architecture:

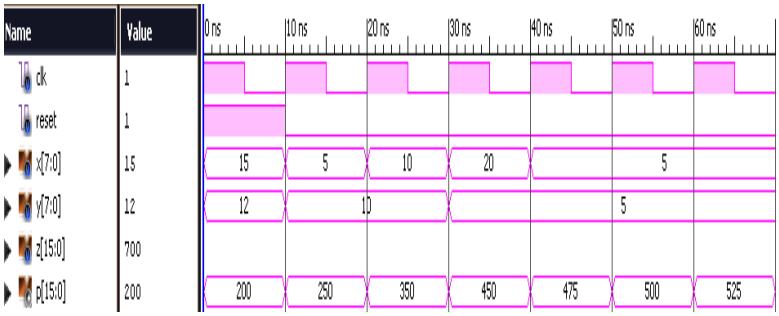

Fig. 5 simulation waveform for the Elguibaly's parallel MAC architecture

The same procedure is applicable for Elguibaly's and proposed parallel MAC architectures also.

3.3 Proposed parallel MAC architecture:

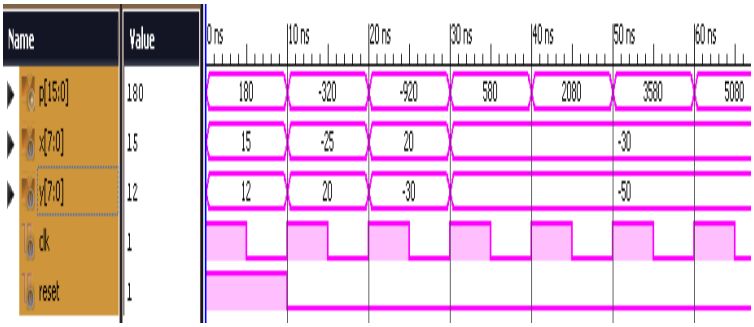

Fig. 6 simulation waveform for the proposed parallel MAC architecture

3.4 Elguibaly's parallel MAC architecture with 2-stage pipelining:

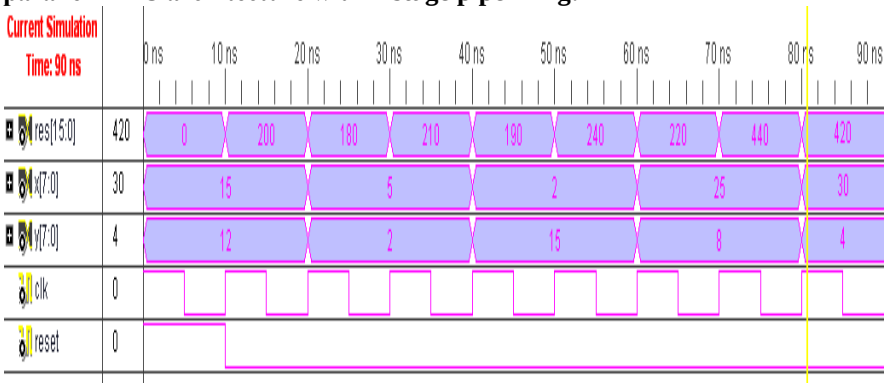

Fig. 7 simulation waveform for the modified Elguibaly's architecture

3.5 Proposed parallel MAC architecture with 2-stage pipelining:

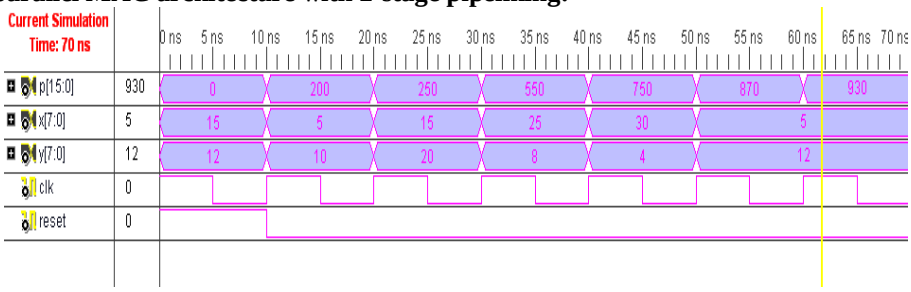

Fig. 8 simulation waveform for the modified proposed architecture

REFERENCES

[1] J. J. F. Cavanagh, *Digital Computer Arithmetic*. New York: McGraw- Hill, 1984.
[2] *Information Technology-Coding of Moving Picture and Associated Autio, MPEG-2 Draft International Standard*, ISO/IEC 13818-1, 2, 3, 1994.
[3] *JPEG 2000 Part I Fina1119l Draft*, ISO/IEC JTC1/SC29 WG1.
[4] O. L. MacSorley, "High speed arithmetic in binary computers," *Proc. IRE*, vol. 49, pp. 67–91, Jan. 1961.
[5] S. Waser and M. J. Flynn, *Introduction to Arithmetic for Digital Systems Designers*. New York: Holt, Rinehart and Winston, 1982.

[6] A. R. Omondi, *Computer Arithmetic Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[7] A. D. Booth, "A signed binary multiplication technique," *Quart. J. Math.*, vol. IV, pp. 236–240, 1952.
[8] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Trans. Electron Comput.*, vol. EC-13, no. 1, pp. 14–17, Feb. 1964.
[9] A. R. Cooper, "Parallel architecture modified Booth multiplier," *Proc. Inst. Electr. Eng. G*, vol. 135, pp. 125–128, 1988.
[10] N. R. Shanbag and P. Juneja, "Parallel implementation of a 4□4-bit multiplier using modified Booth's algorithm," *IEEE J. Solid-State Circuits*, vol. 23, no. 4, pp. 1010–1013, Aug. 1988.
[11] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54□54 regular structured tree multiplier," *IEEE J. Solid-State Circuits*, vol. 27, no. 9, pp. 1229–1236, Sep. 1992.
[12] J. Fadavi-Ardekani, "M□N Booth encoded multiplier generator using optimizedWallace trees," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no. 2, pp. 120–125, Jun. 1993.
[13] N. Ohkubo, M. Suzuki, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, and Y. Nakagome, "A 4.4 ns CMOS 54□54 multiplier using pass-transistor multiplexer," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 251–257, Mar. 1995.

