International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 2, Issue 8, August 2015)

Parallel Multiplier Accumulator Based on Radix-2 Modified Booth Algorithm by Using VLSI Architecture

Gade Sreedhar Reddy
M.Tech Student (VLSI-SD)
Dept.of.ECE, GNIT
sridhar.gade467@gmail.com

Mr. N Srinivas
Associate Professor
Dept.of.ECE, GNIT
nsrinivas.gnit@gmail.com

Mr. M Suman Kumar
Associate Professor
Dept.of.ECE, GNIT
sumankumar.gnit@gmail.com

Prof B Kedarnath
HOD-ECE
Dept.of.ECE, GNIT
hodece.gnit@gniindia.org

***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 delayed 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 250, 180 and 130 /xm, and 90 nm standard CMOS library. Based on the theoretical and experimental estimation, we analyzed the results such as the amount of hardware resources, delay, and pipelining scheme. We used Sakurai's alpha power law for the delay modeling. The proposed MAC showed the superior properties to the standard design in many ways and performance twice as much as the previous research in the similar clock frequency. We expect that the proposed MAC can be adapted to various fields requiring high performance such as the signal processing areas.**

***Keywords:* Parallel multiplier, Radix-2, Carry select adder, MAC & Booth algorithm.**

1. INTRODUCTION

Power dissipation is recognized as a critical parameter in modern VLSI design field. To satisfy MOORE'S law and to produce consumer electronics goods with more backup and less weight, low power VLSI design is necessary.

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. In the past multiplication was generally implemented via a sequence of addition subtraction, and shift operations. Multiplication can be considered as a series of repeated additions. The number to be added is the multiplicand, the number of times that it is added is the multiplier, and the result is the product. Each step of addition generates a partial product.

In most computers, the operand usually contains the same number of bits. When the operands are interpreted as integers, the product is generally twice the length of operands in order to preserve the information content. This repeated addition method that is suggested by the arithmetic definition is slow that it is almost always replaced by an algorithm that makes use of positional representation. It is possible to decompose multipliers into two parts. The first part is dedicated to the generation of partial products, and the second one collects and adds them.

2. PROPOSED METHOD

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 delayed 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 250, 180 and 130 /xm, and 90 nm standard CMOS library. Based on the theoretical and experimental estimation, we analyzed the results such as the amount of hardware resources, delay, and pipelining scheme. We used Sakurai's alpha power law for the delay modeling. The proposed MAC showed the superior properties to the standard design in many ways and performance twice as much as the previous research in the similar clock frequency. We expect that the proposed MAC can be adapted to various fields requiring high performance such as the signal processing areas.

3. DEVELOPMENT

The basic multiplication principle is twofold i.e. evaluation of partial products and accumulation of the shifted

International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 2, Issue 8, August 2015)

partial products. It is performed by the successive additions of the columns of the shifted partial product matrix. The 'multiplier' is successfully shifted and gates the appropriate bit of the 'multiplicand'. The delayed, gated instance of the multiplicand must all be in the same column of the shifted partial product matrix. They are then added to form the product bit for the particular form. Multiplication is therefore a multi operand operation. To extend the multiplication to both signed and unsigned numbers, a convenient number system would be the representation of numbers in two's complement forma

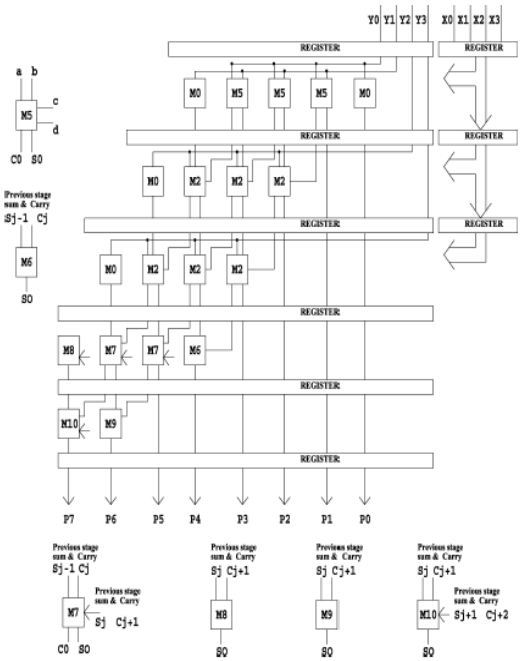


Fig.1. Basic multiplication principle

The MAC (Multiplier and Accumulator Unit) is used for image processing and digital signal processing (DSP) in a DSP processor. Algorithm of MAC is Booth's radix-4 algorithm, Modified Booth Multiplier, 34-bit CSA and improves speed. MIPS was implemented as micro processors and permitted high performance pipeline implementations through the use of their simple register oriented instruction sets.

Although those algorithms (radix-4 algorithm, pipelining, etc) are widely used technique for speeding up each part, the MAC on specific processor cannot be run at 100% efficiency. Due to the reasons of lower speed of MAC, MIPS instruction "mul" (multiplication) takes longer time than any other instruction in our MIPS processor. To improve speed of MIPS, MAC needs to be fast and MIPS must have special algorithm for "mul" instruction. One of the methods we chose was to design multi-clock MAC instead of one-clock MAC which improved the speed of MIPS. In general, the instruction set of MIPS processor includes complex works like multiplication and floating point operation which has multi execution stage. Therefore, system clock of the processor was increased efficiently. We applied 2 stage pipelining to the MAC to MIPS processor and as a result we were able to get the result of matrix multiplication which was used for image processing in our MIPS processor that supports MAC.

4. MULTIPLIER AND ACCUMULATOR UNIT

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. The purpose of this work is, design and implementation of a low power MAC unit with block enabling technique to save power. Firstly, a 1-bit MAC unit is designed, with appropriate geometries that give optimized power, area and delay. The delay in the pipeline stages in the MAC unit is estimated based on which a control unit is designed to control the data flow between the MAC blocks for low power. Similarly, the N-bit MAC unit is designed and controlled for low power using a control logic that enables the pipelined stages at appropriate time. The adder cell designed has advantage of high operational speed, small Gate count and low power.

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. 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 whereWallace 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 .Among them, the architectures based on the Baugh–Wooley algorithm (BWA) have been developed and they have been applied to various digital filtering calculations 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

International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 2, Issue 8, August 2015)

of four steps, as shown in Fig. 1, which shows the operational steps explicitly.

A general hardware architecture of this MAC is shown in Fig. 2. 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.

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

$$X = -2^{N-1}x_{N-1} + \sum_{i=0}^{N-2} x_i 2^i, \qquad x_i \in 0, 1.$$

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

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

$$d_i = -2x_{2i+1} + x_{2i} + x_{2i-1}.$$

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

$$X \times Y = \sum_{i=0}^{N/2-1} d_i 2^{2i} Y.$$

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

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

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 -bit data are multiplied, the number of the generated partial products is proportional to. In order to add them serially, the execution time is also proportional to 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.

Proposed MAC Architecture:

In this section, the expression for the new arithmetic will be derived from equations of the standard design. From this result, VLSI architecture for the new MAC will be proposed. In addition, a hybrid-typed CSA architecture that can satisfy the operation of the proposed MAC will be proposed.

5. RESULTS AND DISCUSSION

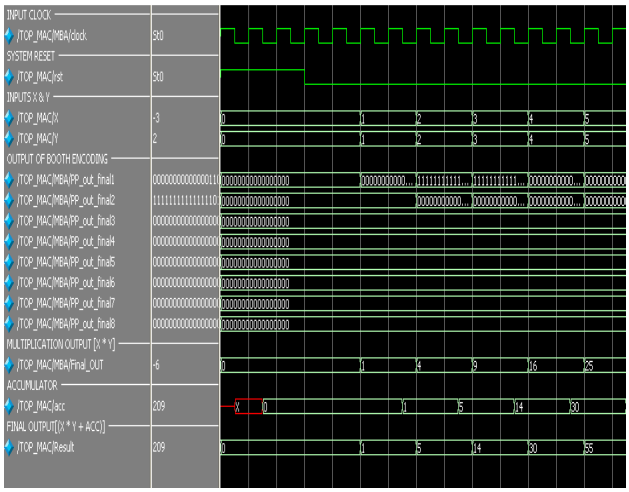


Fig.2. Simulation Results of MAC

The developed MAC design is simulated and verified their functionality. Once the functional verification is done, the RTL model is taken to the synthesis process using the Xilinx ISE tool. In synthesis process, the RTL model will be converted to the gate level netlist mapped to a specific technology library. This MAC design can be synthesized on the family of Spartan 3E. Here in this Spartan 3E family, many different devices were available in the Xilinx ISE tool. In order to synthesis this design the device named as "XC3S500E" has been chosen and the package as "FG320" with the device speed such as "-4". The design of MAC is synthesized and its results were analyzed as follows.

mac1 Project Status			
Project File:	mac1.ise	Current State:	Synthesized
Module Name:	top_tst	• Errors:	No Errors
Target Device:	xc3s500e-4fg320	• Warnings:	22 Warnings
Product Version:	ISE 10.1 - Foundation Simulator	• Routing Results:	
Design Goal:	Balanced	• Timing Constraints:	
Design Strategy:	Xilinx Default (unlocked)	• Final Timing Score:	

mac1 Partition Summary
No partition information was found.

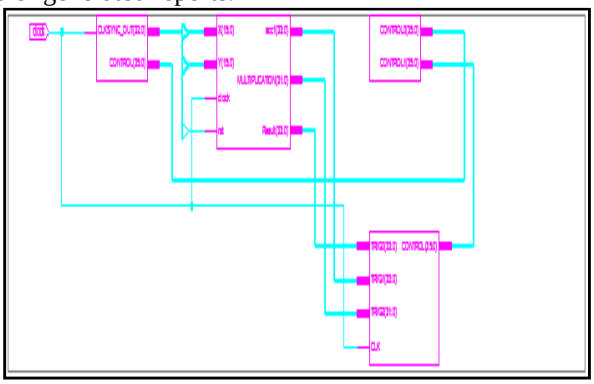
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1182	4656	25%
Number of Slice Flip Flops	678	9312	7%
Number of 4 input LUTs	1863	9312	20%
Number of bonded IOBs	1	232	0%
Number of BRAMs	4	20	20%
Number of GCLKs	2	24	8%

Fig.3. Device utilization summary

International Journal of Advanced Research Foundation
Website: www.ijarf.com, Volume 2, Issue 8, August 2015)

6. SUMMARY

=>The developed MAC design is modelled and is simulated using the Modelsim tool.
=>The simulation results are discussed by considering different cases.
=>The RTL model is synthesized using the Xilinx tool in Spartan 3E and their synthesis results were discussed with the help of generated reports.



7. CONCLUSION

A 16x16 multiplier-accumulator (MAC) is presented in this work. A RADIX -4Modified Booth multiplier circuit is used for MAC architecture. Compared to other circuits, the Booth multiplier has the highest operational speed and less hardware count. The basic building blocks for the MAC unit are identified and each of the blocks is analyzed for its performance. Power and delay is calculated for the blocks. 1-bit MAC unit is designed with enable to reduce the total power consumption based on block enable technique. Using this block, the N-bit MAC unit is constructed and the total power consumption is calculated for the MAC unit. The power reduction techniques adopted in this work. The MAC unit designed in this work can be used in filter realizations for High speed DSP applications.

REFERENCES

[1]. J. J. F. Cavanagh, Digital Computer Arithmetic. New York: McGraw-Hili, 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 Finall19l Draft. ISO/IEC JTClISC29 WG 1.
[4]. O. L. MacSorley, "High speed arithmetic in binary computers," Proc. IRE. vol. 49, pp. 67-91, Jan. 1961.
[5]. S. Waser and M. 1. Flynn, Introduction to Arithmetic for Digital Systems Designers. New York: Holt, Rinehart and Winston,
[6]. A. R. Omondi, Computer Arithmetic Systems. Englewood Cliffs, NJ:
[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. I, pp. 14-17, Feb.