ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

REDUCED COMPUTATION TIME IN FIXED WIDTH BOOTH MULTIPLIER USING BAUGH WOOLEY METHOD

R.Prasanth[1], L.Anbarasu [2] V.Venmathi[3]

PG-Scholar, Dept. of ECE, SNS College of Engineering, Coimbatore, Tamilnadu, India[1]

Assistant professor, Dept. of ECE, SNS College of Engineering, Coimbatore, Tamilnadu, India[2]

PG-Scholar, Dept. of ECE, SNS College of Engineering, Coimbatore, Tamilnadu, India[3]

ABSTRACT: Multiplier plays an important role in digital signal processing systems but it consumes much power and area, In order to reduce the power and area occupied by the multiplier. We opt for fixed width multiplier. Our proposed method is to design fixed-width multiplier using Baugh-Wooley (BW) algorithm. It simplifies a structure of multiplier with the aim of reducing power and increasing performance of system. In this method two topologies are in particularly selected as the most effective ones. The first one is based on a uniform coefficient quantization, while the second topology uses a non uniform quantization scheme. The novel fixed-width multiplier using Baugh-Wooley algorithm exhibit better accuracy with respect to previous solutions the novel fixed-width multiplier topologies exhibit better accuracy with respect to previous solutions, close to the theoretical lower bound. The electrical performances of the proposed fixed-width multi-pliers are compared with previous architectures. It is found that in most of the investigated cases the new topologies are Pareto-optimal regarding the area-accuracy trade-off.

I. INTRODUCTION

In this project we are going to design the fixed-width multiplier using Baugh-Wooley (BW) algorithm. Fixed-width multiplier has the same bit width of input and output. In digital signal processing systems multiplier plays an important role but also it consumes more power and area. In order to reduce the power and area occupied by the multiplier unit fixed-width multipliers are designed. The fixed-width property simplifies the multiplier structure with the aim of improving power and speed.

In fixed width multiplication the product is fixed width to n-bits, the least-significant columns of the product matrix contribute little to the final result. To take advantage of this, fixed width multipliers do not form all of the least-significant columns in the partial-product matrix. Output is the weighted sum of partial-products. By eliminating more columns the area and power consumption of the arithmetic unit are significantly reduced & in many cases the delay also decreases. Area saving of a fixed width multiplier can be achieved by directly truncating n least significant columns and preserving n most significant columns This algorithm is a relatively straightforward way of doing signed multiplications. It uses only an array of full adders and a final Ripple Carry Adder (RCA). The main objective of this project is to design the fixed width BW multiplier using full-width BW multiplier which computes the 2n output as a weighted sum of partial-products. Many applications require an n bits output. A typical example is a digital, computationally intensive, ASIC in which the bit-width at the output of the arithmetic blocks is chosen on the basis of system related accuracy issues. Having a multiplier with an n bits output can also be useful in digital signal processors where the inputs and the outputs of the data path can be stored in registers with the same bit-width. A fixed-width multiplier is an n × n multiplier that computes, with a certain 7yapproximation, the most-significant bits of the product.

BW algorithm is an efficient way to handle the sign bit. BW multiplier uses only full adders. The main aim of this project is to reduce the area and power consumed by the fixed-width signed multiplier which uses both half adders and full adders for implementation.

II. EXISTING FIXED WIDTH MULTIPLIER

Parallel multipliers are fundamental building blocks in multimedia and digital many applications, the inputs and the output of the multiplier have the same bit width. These circuits are denoted in literature as fixed-width multipliers or fixed width multipliers. The most obvious way to design a fixed-width multiplier uses a full-width multiplier, whose

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

output is fixed width/rounded to n bits by n discharging the less-significant bits of the products. The fixed-width property, however, can be exploited to simplify the multiplier structure, with the aim of improving power and speed. Basically, one can discard some of the partial- products in the partial-products array to reduce the circuit complexity, at a price in terms of accuracy. This is the approach pursued in all the fixed-width multiplier architectures proposed in literature.

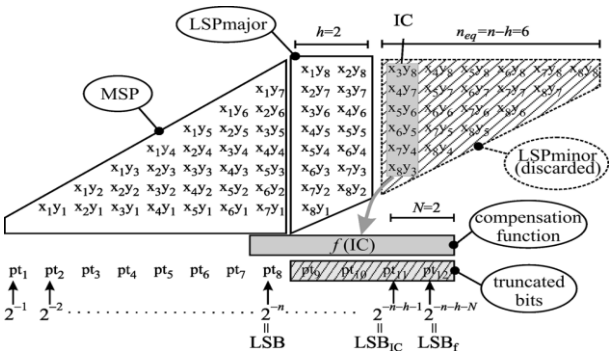

Fig 2.1. Partial-products matrix of a 8x8 bit multiplier,

fixed width with h=2 and using a variable-correction scheme to compute the result.
Many error compensation methods have been proposed in literature. In the simplest approaches the compensation function is a fixed bias. Better accuracy is obtained in the so-called variable-correction (or adaptive) fixed-width multipliers. In these architectures the partial-products in the leftmost column of LSP minor are employed to obtain a probabilistic estimate of the sum of the elements of the LSP minor. In Fig 2.1 the partial-products in the leftmost column of LSP minor are highlighted in gray and are named Input Correction (IC), while f(IC) is the compensation function.

The choice f(IC) of is the critical step in the design of a fixed-width multiplier. This function should be efficiently implemented in hardware, and should provide, at the same time, a good estimate of the sum of the elements of the LSP minor.

III. MINIMUM MEAN SQUARE COMPENSATION FUNCTION

In the following we will consider unsigned multipliers and we will also assume that the inputs of fractional values in (0,1).

$$x=\sum_{i=1}^{n} x_i 2^{-i}; \qquad y=\sum_{i=1}^{n} y_i 2^{-i};$$

The weight of the less-significant bit of x and y is given by

$$LSB=2^{-n}$$

We assume that the less-significant bit of the multiplier output has the same weight of x and y (see Fig 3.1). The x_i and y_i bits are assumed to be independent and uniformly distributed, with a probability 1/2 of being one.
The elements of the IC (see Fig 3.1) will be indicated as follows:

$$IC=[\gamma 1, \gamma 2 \ldots, \gamma\, n_{equ}]$$

Where

$$n_{equ}=n-h$$

$$\gamma_i = x_{h+i} y_{n+1-i}$$

The weight of the IC elements is indicated as LSB_{IC} and is given by

$$LSB_{IC} = LSB\,2^{-h-1} = 2^{-n-h-1}$$

The weight of the less-significant bit used for the calculation of the compensation function is indicated as

$$LSB_f = LSB_{IC}\, 2^{-(N-1)} = 2^{-n-h-N}$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

IV. PROPOSED BAUGH-WOOLEY ALGORITHM

The BW algorithm is a relatively straight forward way of doing signed multiplications. The fixed width multipliers derived from BW algorithm produce n-bit output product with n-bit multiplier and n-bit multiplicand.
It is an efficient way to handle the sign bit. BW multiplier uses only full adders. All bit products are generated in parallel & collected through an array of full adder & final RCA. The creation of the reorganized partial-product array comprises three steps:
i) The Most Significant Bit (MSB) of the first N-1 partial-product rows and all bits of the last partial-product row, except its MSB, are inverted.
ii) A '1' is added to the Nth column.
iii) The MSB of the final result is negated.
Fig 4.1 shows the partial product array diagram for n×n BW multiplier.

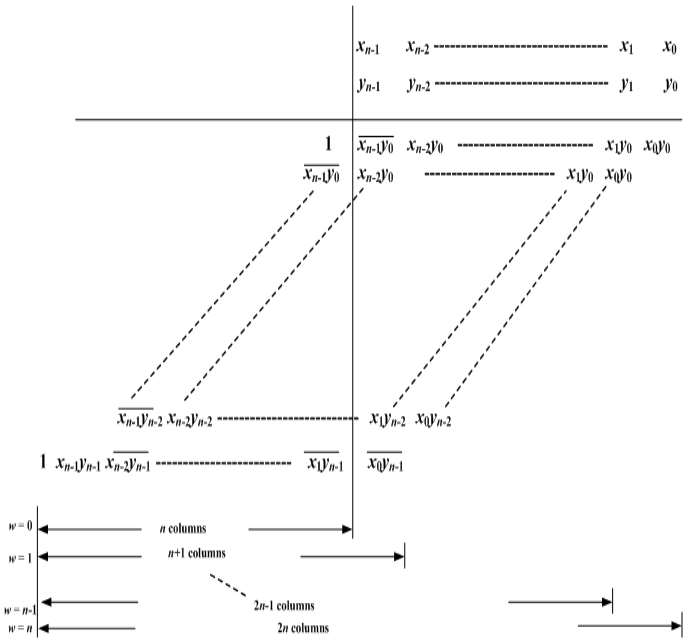

Fig 4.1 Partial Product Array Diagram for n×n BW Multiplier

A. Truncation

Truncation is a method where the least significant columns in the partial product matrix are not formed. The amount of columns not formed in this way, T, defines the degree of truncation and the T Least Significant Bits (LSB) of the product always results in '0'. The algorithm behind fixed width multiplication is the same as when dealing with non fixed width multiplication regardless of the truncation degree.

B. Rounding

Conventionally an n-bit multiplicand and an n-bit multiplier would render a 2n-bit product. Sometimes an n-bit output is desired to reduce the number of stored bits. This is done in two steps.

V. RESULTS

A. Electrical Performances

As we have pointed-out previously, our fixed width multipliers are well suited for effective tree based implementations. This happens for most of the previously proposed fixed width multipliers, use array multiplier with a ripple architecture, that are very slow and power hungry. These architectures can also be implemented with a much more effective carry-save TDM(three dimensional algorithm) reduction method. However, it is worth highlighting that the terms added to the partial-products matrix related to the compensation function are still obtained with a ripple AND/OR network, which increases power dissipation and propagation delay. In order to have a fair comparison among every approach, we have implemented every multiplier by using a carry-save TDM reduction tree followed by a fast carry-propagate adder.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

B Area versus Accuracy Trade-Off

In practical applications the accuracy is generally constrained by system-level consideration. The designer goal is hence to achieve the lowest complexity for a specified accuracy. As we have highlighted in the previous sections, is a trade-off parameter between accuracy and complexity that can be used to obtain an optimal design. Therefore, a comparison of the different fixed width multipliers considering this trade-off between accuracy and complexity can be very useful.TDM are the one the efficient method to improve area than the previous proposed methods. As we have pointed out previously, our fixed-width multi-pliers are well suited for effective tree based implementations.

This happens also for most of the previously proposed fixed-width multipliers. Therefore, in performing a comparison between various architectures, we implemented all the circuits by using a carry-save TDM reduction tree followed by a fast carry-propagate adder. As we have pointed out previously, our fixed-width multi-pliers are well suited for effective tree based implementations. This happens also for most of the previously proposed fixed-width multipliers. Therefore, in performing a comparison between various architectures, we implemented all the circuits by using a carry-save TDM reduction tree followed by a fast The area report of fixed-width Xilinx synthesis tool is shown in Table5.1. The total area is calculated based on the utilization of number of LUT's which is present in the device and the corresponding occupied slices.

Table5.1Device utilization of Fixed-Width linear compensation Multiplier

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	103	1,536	6%
Number of occupied Slices	57	768	7%
Number of Slices containing unrelated logic	0	57	0%
Total Number of 4 input LUTs	103	1,536	6%
Number of IObonded IOBs	27	63	42%

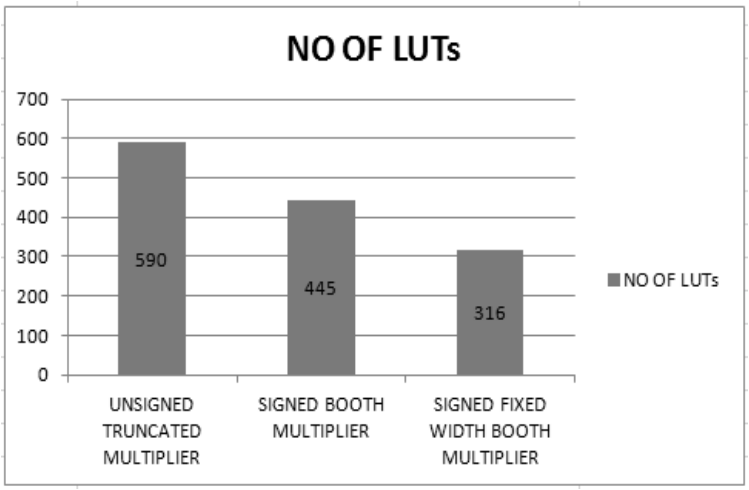

Figure 5.1 Comparative graphs of LUTs

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

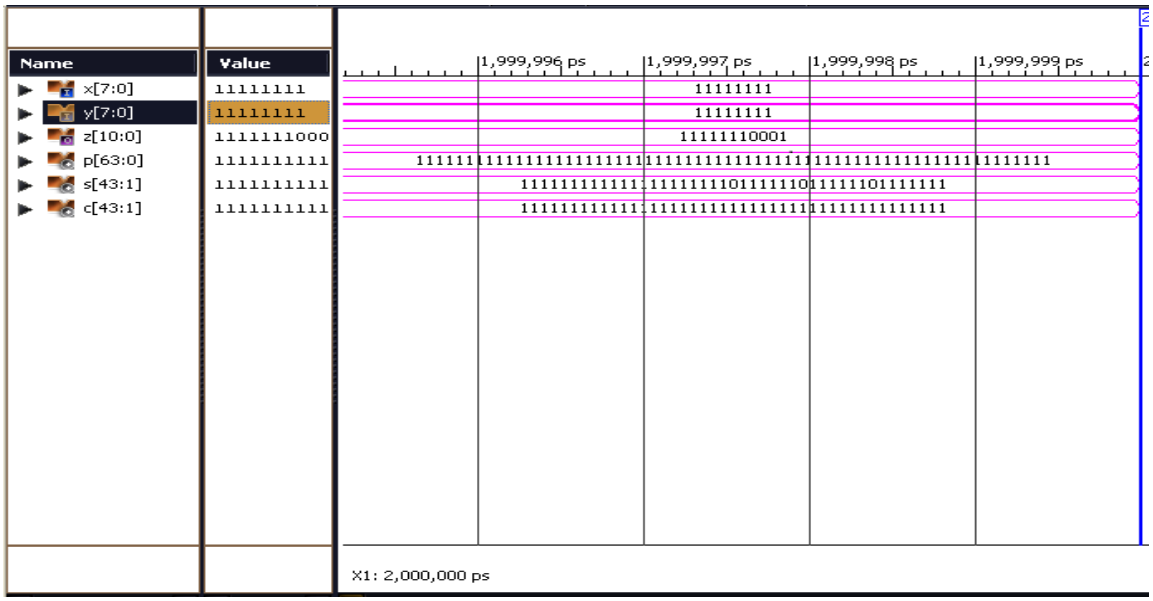

Fig5.2 simulation results of fixed width multiplier

More area than fixed width. The simulation waveform of fixed-width linear multiplier using Xilinx simulation tool is shown in Fig 5.2. Given bits processed

Table 5.2 Proposed Area utilization Report of Fixed-Width Multiplier

Logic Utilization	Used	Available	Utilization
Number of 4 input LUTs	336	1,536	21%
Number of occupied Slices	181	768	23%
Number of Slices containing only related logic	181	181	100%
Number of Slices containing unrelated logic	0	181	0%
Total Number of 4 input LUTs	336	1,536	21%
Number of bonded IOBs	50	63	79%
Average Fan-out of Non-Clock Nets	3.05		

Table 5.3 Area Report of Fixed-Width Multiplier

FIXED WIDTH MULTIPLIER	AREA UTILIZATION
2 BIT ,H=0	2.95
2 BIT ,H=1	3.05
1.5 BIT ,H=0	2.91

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering
Vol. 2, Issue 6, June 2013

Table 5.3 shows the Area report of fixed-width LINEAR COMPENSATION multiplier Xilinx synthesis tool. The total area is calculated based on the utilization of number of LUT's which is present in the device and the corresponding occupied slices. All the parameters are explained and specification is given above. It is clear normal multiplication requires more area than fixed width. The post synthesis implementation data of the 1.5 bits and 2bits fixed-width multipliers the approach using the auxiliary tree yields some advantage with respect to the standard architectures especially for 2 bits multipliers, where the number of duplicated partial-products is higher. For example, for and , the 2 bits fixed-width multiplier using the standard architecture results in increase area with respect to the multiplier of PRESENT Method. the fixed-width multipliers proposed in this paper are more accurate with respect to previously proposed architectures, the improvement being more evident for large.

VI. CONCLUSION

The truncated multiplier design by jointly considering the deletion, reduction, truncation and rounding of the partial product bits during tree reduction is analysed in terms of area and power using the Xilinx tool. The fixed width multiplier a subset of truncated multiplier designs, for a 10*10 unsigned multiplication the area and the power analysis is obtained. The truncated multiplier has a total error of no more than LUT and used in applications that require accurate results. The truncated multiplication is to be compared with extended booth multiplier to analyse the area and power using Xilinx tool. Modification is to done in the multiplier technique to obtain less power consumption.

REFERENCES

[1]. A.G.M. Strollo, E. Napoli, D. De Caro ,N. Petra and V. Garofalo, "Truncated binary multipliers with variable correction and minimum mean square error," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 6, pp. 1312–1325, Jun. 2010.
[2]. B.Parhami, Computer Arithmetic: Algorithms and Hardware Designs.New York: Oxford Univ. Press, 2009.
[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]. E.ESwartzlander,K.C.Bickerstaff and M. Schulte, Jr., "Parallel reduced area multipliers," J. VLSI Signal Process., vol. 9, no. 3, pp. 181– 191, 1995.
[5]. Hou-Jen Ko and Shen-Fu Hsiao" Design and Application of Faithfully Rounded and Truncated Multipliers With Combined Deletion, Reduction, Truncation, and Rounding" IEEE Transactions On Circuits And Systems—II: Express Briefs, Vol. 58, No. 5, May 2011.
[6]. J.-P. Wang, S.-R. Kuang, and S.-C. Liang, "High-accuracy fixed-width modified booth multipliers for lossy applications," in IEEE Trans. Very Large Scale Integr. (VLSI) Syst., Jan. 2011, vol. 19, no. 1, pp. 52–60.
[7]. L. Dadda, "Some schemes for parallel multipliers," Alta Frequenza, vol. 34, pp. 349–356, 1965.redmultipliers," J. VLSI Signal Process., vol. 9, no. 3, pp. 181– 191, 1995.