www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3, Issue 10 October, 2014 Page No. 8957-8959

Variable Truncated Multiplier with Low Power

***Athira Prasad*[1]*, Robin Abraham*[2]**

[1]Ilahia College of Engineering and Technology,
Cochin ,Kerala, India.
athiraprasad09@gmail.com

[2]Asst Professor. Ilahia College of Engineering and Technology ,
Cochin ,Kerala, India
robymec@gmail.com

Abstract: ***Multipliers play a significant part in the DSP architecture. Truncation helps in the reduction of power consumption by disabling a portion of the partial product. In this paper different multipliers are made to truncate the partial products variably using a control bit and then compared for power and delay. The comparison results show that the Dadda multiplier shows a small improvement in power and Baugh Wooley is better in terms of delay.***

Keywords: Truncated Multiplier, Baugh Wooley Multiplier, Variable Truncation, Dadda Multiplier.

1. Introduction

Multipliers have become inevitable with the advancement of communication. In order to enable the implementation of complex algorithms in DSP architectures the advancing VLSI play a significant role. High speed, Low power consumption, layout regularity, reduction in area, time and delay are the major concerns while implementing multipliers which represent the backbone of a DSP system.

Multipliers developed are mainly of fixed width ie; for a N x N bit multiplication, the output attained is 2N bit product. Thus we know that we can have only a fixed width output and cannot control the length of the bits. In order to control the length of the bits according to the precision, required for the application, we can make the fixed width multipliers to variable truncated multipliers.

Truncation is the process of disabling or skipping a portion of the partial product to reduce the consumption of power. This process is merely a cut-off digit. In this paper different fixed width parallel multipliers are modified to multipliers with variable truncation and they are compared for power, and delay.

The Paper is designed as follows. Section 2 presents the Baugh Wooley multiplier, Section 3 presents the modified Baugh Wooley multipliers which has slight modification from the earlier described Baugh Wooley multiplier. Section 4 presents the Dadda multiplier and Section 5 briefly describes the comparisons and simulation results of the different variable multipliers. The final section 6 represents the conclusion.

2. Baugh Wooley Multiplier

2.1 Two's Complement Multiplication

This requires an N-bit fractional multiplicand and an N-bit fractional multiplier as the inputs of the multiplier. Consider

$$X= -x_{n-1}\,2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i$$

$$Y = -y_{n-1}2^{n-1} + \sum_{j=0}^{n-2} y_j 2^j \qquad (1)$$

Where X and Y are the multiplier and multiplicand and x_i, y_i ϵ 0, 1.

The 2N-bit product will be maintained in full precision as

P = X x Y

$$=\left(-x_{n-1}\,2^{n-1} + \sum_{i=0}^{n-2} x_i 2^i\right)\left(-y_{n-1}2^{n-1} + \sum_{j=0}^{n-2} y_j 2^j\right)$$

$$= x_{n-1}y_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} x_i y_j 2^{i+j} -$$

$$2^{n-1}\ \sum_{i=0}^{n-2} x_i y_{n-1} 2^i - 2^{n-1}\sum_{j=0}^{n-2} x_{n-1} y_j 2^j \qquad (2)$$

Applying Baugh Wooley algorithm [7] to the partial product matrix generates results of all-positive partial product bits in a final matrix such as

$$P = x_{n-1}y_{n-1}2^{2n-2} + \sum_{i=0}^{n-2}\sum_{j=0}^{n-2} x_i y_j 2^{i+j}$$

$$+ 2^{n-1}\sum_{i=0}^{n-2}\overline{x_i y_{n-1}}2^i + 2^{n-1}\sum_{j=0}^{n-2}\overline{x_{n-1} y_j}2^j - 2^{2n-1} + 2^n \quad (3)$$

When the partial product matrix of the parallel multiplier is obtained, the final product result can be generated by combining the partial product with the final result. The basic structure for a 5 x 5 bit multiplier is shown in Fig.1.

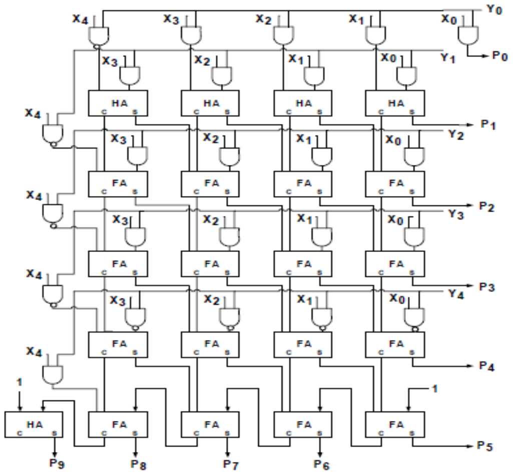

Figure.1 5 x 5 bit Baugh Wooley Multiplier

2.2 Truncated Multipliers

Truncated Multiplier has the advantage of reducing power consumption in the DSP systems. It is most commonly used in systems where least significant part of partial product can be skipped or disabled which leads to low power consumption, area and timing. Here the partial product is split into two sections namely the Least Significant Part (LSP) and Most Significant Part (MSP). The LSP is disabled or avoided to get the truncated output. The product of a full-width Multiplier can be described as $P_{full} = S_{MSP} + S_{LSP}$, where S_{MSP} represents the sum of the partial product bits belonging to the MSP and S_{LSP} is the sum of the bits belonging to LSP. Generally, the output of the fixed-width N x N truncated multiplier is represented as $P_{full\text{-}rounded} = t_N (S_{MSP} + S_{LSP} + LSB/2)$, where LSB represents the Least Significant Bit and t_N (x) represents the truncation of an operand x by eliminating its lowest bits. Bits are discarded to maintain bit-width of N-bits.

2.3 Variable Truncated Multiplier

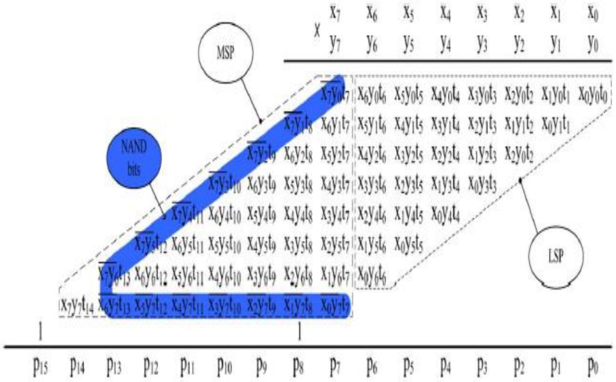

Figure.2 8 x 8 bit Variable Truncated Multiplier

Variable Truncated Multiplier [1] effectively helps in adjusting the output width of the multiplier. This follows a column based strategy which flexibly truncates the partial product as per the requirement thus controlling the power and time constraints. Truncation may be different for different applications. This architecture is flexible and work differently for different applications. The applications may differ from each other as some may require a high precision output and some others may require low power, time or area. Thus Variable Truncated Multiplier can be employed as a general purpose multiplier. The concept of the Variable Truncated Multiplication for an 8 x 8 bit multiplier is illustrated in Fig.2. The general representation of the resultant of variable truncated multiplier is

$$P_{VTM} = 2^{-1} + 2^{-N} + \sum_{i=0}^{N-2}\sum_{j=0}^{N-2} t_{(i+j)} \, . \, ppt_{[i,j]} \, . \, 2^{i+j-2N} \qquad (4)$$

Where $t_{(i+j)}$ represents the control bit and $ppt_{[i,j]}$ represents the partial product matrix. Partial product terms for the Baugh Wooley implementation is

$$ppt_{[i,j]} = \begin{cases} x_i \cdot y_j & \text{if } 0 < i < (N-2) \text{ and } 0 < j < (N-2) \\ \overline{x_i \cdot y_j} & \text{if } 0 < i < (N-2) \text{ and } j = (N-1) \\ \overline{x_i \cdot y_j} & \text{if } i = (N-1) \text{ and } 0 < j < (N-2) \\ x_i \cdot y_j & \text{if } i = (N-1) \text{ and } j = (N-1) \end{cases} \qquad (5)$$

where x_i and y_j represents the input bits X and Y of the multiplier. In this the 2-input AND gate is replaced with the 3-input AND gate in which the third input is the control bit $t_{(i+j)}$ .This helps in attaining column-wise controllability which in turn makes the fixed-width multiplier a variable multiplier. To obtain a full precision multiplier all the bits within t is made to be 1 i.e. the control bit t = 0x7FFF and inorder to obtain a half partial product matrix the value of t = 0x7F00.

3. Modified Baugh Wooley Multiplier

Conventional Baugh Wooley architecture makes use of AND and NAND gates. This gates can be replaced with OR and NOR gates [4] which is possible by applying the DeMorgan's theorem.

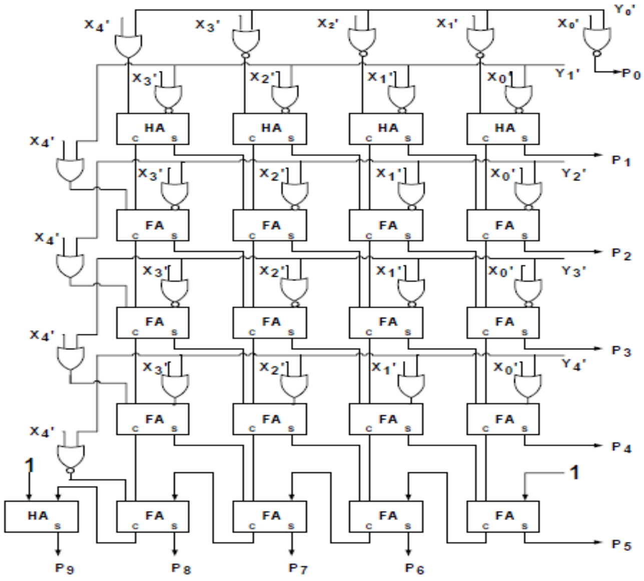

Figure 3 5 x 5 bit Modified Baugh Wooley Architecture

DeMorgan's law states that

$$\overline{A+B} = (\bar{A}.\bar{B})$$

$$\overline{A.B} = (\bar{A}+\bar{B}) \qquad (6)$$

By applying the DeMorgan's law we find that the inputs given for the OR operation is the complement of that given for the AND operation. Fig. 3 shows the modified structure of Baugh Wooley architecture. This multiplier is made variable by adding a control bit similar to that used in the variable Baugh Wooley architecture. The two input OR and NOR gates are made to three input OR and NOR gates in which the third bit is the control bit which helps in controlling the truncation. When compared to AND/OR gates which make use of 6 transistors for the architecture, the NAND/NOR gates use only 4 transistors.

4. Dadda Multiplier

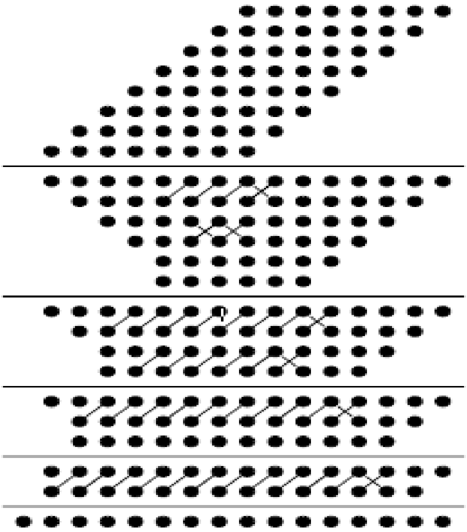

Figure. 4 8 bit Dadda Multiplier

Dadda multiplier, which consists of three stages, belongs to the group of fastest multipliers. The first stage generates the partial product matrix using N^2 AND gates. The 8 x 8 Dadda Multiplier dot diagram is represented in Fig.4. Each dot denotes the partial products [2], [3]. The partial product is reduced to a height of two in the second stage. This stage uses the column compression procedure. When the matrix is reduced to two, adder is used to obtain the final product. The step involved in the Dadda multiplier is the same as that is used in the conventional Dadda multiplier. In the modified Dadda multiplier a control bit is added to obtain variable truncation. The procedure involved in obtaining the result is as follows

- Let k_1 = 2 and repeat k_{j+1} = floor (1.5 x k_j) for increasing values of j. k_j is the height of the matrix for the j^{th} stage. This step is continued until the largest j is reached where there exists at least one column in the present stage of the matrix with more dots than k_j. Using this equation we get k_1=2, k_2 =3, k_3=4, k_4=6, k_5=9 and so on.
- Every column having heights greater than k_j, are reduced to a height of k_j using either half adder or full adder.
- The reduction procedure comes to halt if the height of the matrix becomes two.

5. Comparison and simulation results

Table-1: Comparison between different variable truncated multipliers

Multipliers	*Power (mW)*	*Delay (ns)*
Baugh Wooley Multiplier	77.72	3.192
Modified Baugh Wooley Multiplier	77.86	4.344
DADDA Multiplier	77.65	4.296

Baugh Wooley Multiplier, Modified Baugh Wooley Multiplier and Dadda Multiplier modified to variable truncated multipliers were compared for their power and delay and the result is shown in Table I. The simulation was performed using ModelSim SE PLUS 6.2b and synthesis results were obtained from Xilinx 12.3.

6. Conclusion

In this paper various fixed width multiplier architectures were made variable truncating multipliers. The comparison results show Dadda multiplier exhibits a small improvement in power but have larger delay than others. Wallace and Baugh Wooley multipliers exhibit smaller delay.

References

[1] Manuel de la Guia Solaz, Richard Conwayand WeiHan,"A Flexible Low Power DSP With a Programmable Truncated Multiplier" IEEE transactions on circuits and systems—I : regular papers, vol. 59, no. 11, November 2012

[2] M. Schulte, J. Stine, and J. Jansen, "Reduced power dissipation through truncated multiplication," in Proc. IEEE Alessandro Volta Memorial Workshop Low-Power Design, 1999, pp. 61–69.

[3] W. J. Townsend, E. E. Swartzlander, Jr., and J. Abraham, "A comparison of Dadda and Wallace multiplier delays," in Proc. SPIE Symp.Adv. Signal Process. Algorithms, Archit., Implementations, San Diego, CA, 2003, vol. 5205.

[4] Ronak Bajaj, Saransh Chhabra, Sreehari Veeramachaneni and M B Srinivas,"A Novel, Low-Power Array Multiplier Architecture", International Institute of Information Technology-Hyderabad and Birla Institute of Technology and Sciences (BITS) Pilani, Hyderabad Campus

[5] P.C.Franklin, M.Ramya," Resource Efficient Reconfigurable Processor for DSP Applications," International Journal of Innovative Research in Science, Engineering and Technology,Volume 3, Special Issue 3, March 2014

[6] L.-D. Van and J.-H. Tu, "Power-efficient pipelined reconfigurable fixed-width baugh-wooley multipliers," IEEE Trans. Comput., vol. 58, no. 10, pp. 1346–1355, Oct. 2009.

[7] Baugh-Wooley Multiplier. [online]. Available : http://www.ece.uvic.ca/~fayez/courses/ceng465/lab_465/project2/multiplier.pdf

[8] Truncated Binary Multipliers with minimum mean square error: analytical characterization, circuit implementation and applications by ValeriaGarofalo, [online]. Available: http://www.fedoa.unina.it/3904/1/Garofalo_Thesis.pdf.