Area and Power Efficient Wallace Multiplier

Mamatha V S[1] Babitha S[2]
[1]M. Tech. Student [2]Associate Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]Don Bosco Institute of Technology, Bangalore India

Abstract— Multiplication can be measured as a chain of repeated additions. The number which is to be added is called the multiplicand, the number of times which is added is called the multiplier, and the result being given is known as the product. The necessary operations associated in multiplication consist of generating and adding the partial product. As an outcome, to speed up the whole multiplication technique, these two most important steps must be optimized. (a) Creation of partial products (b) partial product addition using speed adders. The modified Wallace multiplier uses MUX realized from transmission gate for partial product generation and also for addition achieving significant improvement in power and area.

Key words: Transmission Gate Based MUX, Kogge Stone Adder, Wallace Tree

I. Introduction

Multiplier is most used element in mathematical operations and is one of the crucial hardware blocks in most of the digital and high performance systems such as digital signal processors, process controllers, computers, graphic engines and microprocessors. With the latest progresses in technology, many moderates significantly. The Wallace tree fundamentally multiplies two unsigned integers. The conventional Wallace tree multiplier architecture comprises of an AND array for computing the partial products, a carry save adder for adding the partial products so obtained using kogge stone arrangement in the final stage of addition.

In the proposed architecture, partial product generation is done using transmission gate based MUX. Final addition using half adders and full adders are also realized by transmission gate based MUX replacing basic gates. Modified Wallace multiplier consumes less area compared to the conventional Wallace multiplier, reduced area Wallace multiplier, which is compared using kogge stone adder for addition of partial products. The main aim is to offer an area and power efficient Wallace multiplier.

Nevertheless, the statement remains that area and speed are two contradictory enactment limitation. Innovation in speed may result in large area. The delay is decreased marginally as the stages of operation are same in the above compared modifications of the Wallace tree multiplier. The only way for reducing area is by decreasing number of transistors and that is made by using transmission gate based MUX. Whereas in conventional Wallace multiplier it uses a chain of AND gates performing Wallace tree generation.

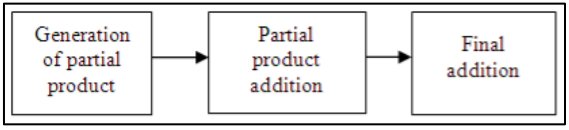

Fig. 1: Block diagram of multiplier architecture

Block diagram of a multiplier consists of three stages as shown in Fig.1, generation of partial products, addition of the generated partial product and the final addition in the last stage when two rows of sum and carry are generated. Parallel multipliers are the most rapid multiplier type. Many type of architecture exists while designing the parallel multiplier. One of those is Wallace multiplier. As system performance is concluded by performance of multiplier. Hence multiplier plays a very major role while designing any VLSI circuits.

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have large area, long latency and consume considerable power. Therefore low-power multiplier design has been an important part in low- power VLSI system design. Power refers to number of Joules dissipated over a certain amount of time whereas energy is the measure of the total number of Joules dissipated by a circuit. In digital CMOS design, the well-known power-delay product is commonly used to assess the merits of designs. In a sense, this can be shown as power × delay = (energy/delay) × delay = energy, which implies delay is irrelevant. The literature review will emphasis on the thorough study carried; intricate on many methods in which this will assist the design and build at end. The performances of the exiting multiplier schemes are limited by the time to do a carry propagate addition. Carry propagate adds are relatively slow, because of the long wires needed to propagate carries from low order bits to high order bits. Probably the single most important advance in improving the speed of multipliers, pioneered by Wallace, is the use of carry save adders (CSAs also known as full adders or 3-2 counters), to add three or more numbers in a redundant and carry propagate free manner. By applying the basic three input adder in a recursive manner, any number of partial products can be added and reduced to 2 numbers without a carry propagate adder. A single carry propagate addition is only needed in the final step to reduce the 2 numbers to a single, final product. The general method can be applied to trees and linear arrays alike to improve the performance.

II. Previous Architecture

In the conventional 8 bit Wallace tree multiplier design, more number of addition operations is required. Using the carry save adder, three partial product terms can be added at a time to form the carry and sum. The sum signal is used by the full adder of next level. The carry signal is used by the adder involved in the generation of the next output bit, with a resulting overall delay proportional to log3/2n, for n number of rows. In the first and second stages of the Wallace structure, the partial products do not depend upon any other values other than the inputs obtained from the AND array. However, for the immediate higher stages, the final value (PP3) depends on the carry out value of previous stage. This operation is repeated for the consecutive stages. Hence, the major cause of delay is the propagation of the carry out from the previous stage to the next stage. In conventional Wallace tree structure, the total number of stages in the critical path sums up to 13. Each full adder accounts for a latency of 2. Therefore, the total latency of the given structure when calculated is 26. The latency count gets added by one, when considering the AND array, thus resulting in a total latency.

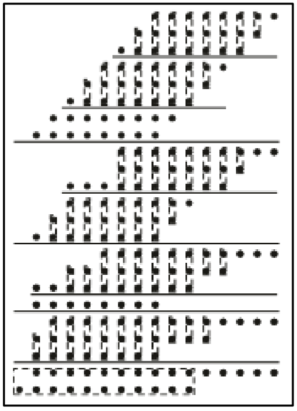

Fig. 2: Traditional Wallace multiplier 8 – bit

The partial products are readjusted in a reverse pyramid style which makes it easy to analyse the tree for efficient reduction. The number of stages for RCW multiplier remains the same as that of TW multiplier. RCW tries to reduce the partial product tree using only full adders. Half adders are used only where they are necessary to satisfy the number of rows in a stage. This approach allows RCW multiplier to reduce the area of the reduction process. However, RCW multiplier uses a much larger final adder as compared to TW multiplier.

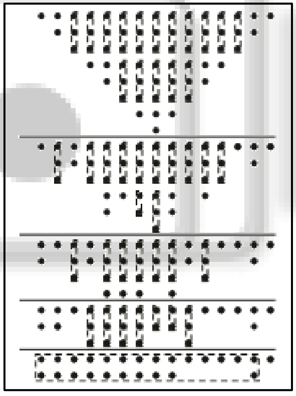

Fig. 3: Reduced complexity Wallace multiplier 8 – bit

III. Proposed Architecture

Our proposed architecture aims to reduce the overall latency. This leads to increased speed and reduced power consumption.

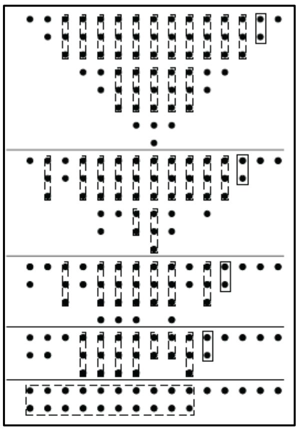

Fig. 4: Proposed Wallace multiplier 8 – bit

In this section, we proposed a modification in the RCW multiplier to further reduce its area by reducing the size of the final adder. PW multiplier has the same number of stages and the same rule for maximum number of rows in a stage. An 8-bit PW reduction process is shown in Fig. 4. We have to design first an AND gate, 1 bit half adder, 1 bit full adder realized by MUX all sub modules are checked for functionality before combined into a module and combine them to build Wallace tree multiplier. Both half adders and full adders are used as a combination to form a carry save adder and arranged in kogge stone manner

PW uses an additional half adder in each stage in order to reduce the size of the final adder. The algorithm scans from the right side and starts the reduction by using a half adder when it finds the first column where the number of elements is greater than one. The additional half adders are shown in solid boxes at each stage of PW multiplier in Fig. 4. Compared with RCW in Fig. 3, the introduction of half adders in Fig. 4 makes the final adder in PW “less wide”, namely, a smaller size. This is because, in each stage, the half adder that we introduced computes the final product bit for that particular column of the partial product tree. Therefore, the size of the required final adder is decreased by one in each stage. The least significant bit (LSB) of the product, $P0$, is produced by the partial product generation block by computing $A0 \times B0$. The partial product generation using array of AND gate is replaced by an array of transmission gate based MUX as shown in Fig. 5

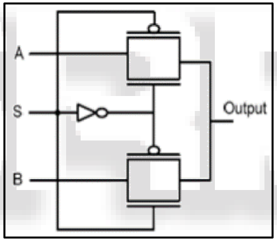

Fig. 5: Transmission gate based MUX

In the first stage of the reduction process, product bit P1 is computed by using the additional half adder. In the second stage, $P2$ is computed. Similarly, stage 3 and stage 4 compute the product bits $P3$ and $P4$, respectively. Thus, when the partial product tree is reduced to two rows, five LSBs ($P4 - P0$) of the product are already computed as shown in Fig. 4. This is achieved at the expense of an increased area for reduction process due to the insertion of additional half adders in the PW. However, the effect of additional half adders realized from transmission gate based MUX is very small as compared to the area saved by half adders derived from basic gates and also reducing the final adder size in the same way. The Half adder and Full adder realization using mux is shown in Fig. 6 and 7 respectively.

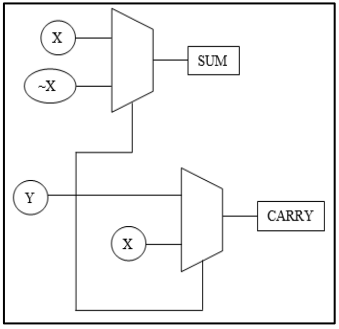

Fig. 6: Half adder realization using MUX

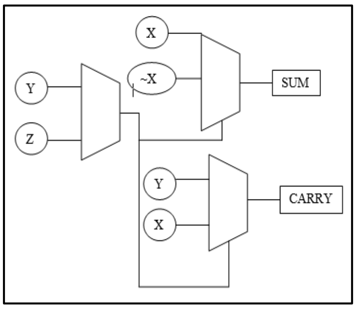

Fig. 7: Full adder realization using MUX

Therefore, the overall area of the PW is less than that of the RCW. The PW has the smallest final adder as compared to all other multipliers. All the multipliers need the same number of logic levels to implement the final adder, which means that all the multipliers will have almost the same delay. The half adders and Full adders which are realized by basic gates are replaced by half adders and full adders realized by transmission gate based MUX.

In full adder the sum and carry expression are specified below

$$
\begin{aligned}
\text{Sum} &= X \oplus Y \oplus Z \\
&= (X \oplus Y) \oplus Z \\
&= (\bar{X}Y + X\bar{Y}) \oplus Z \\
&= (\overline{\bar{X}Y + X\bar{Y}})\, Z + (\bar{X}Y + X\bar{Y})\, \bar{Z} \\
&= (\bar{X}\bar{Y} + XY)\, Z + (X\bar{Y} + X\bar{Y})\, \bar{Z} \\
&= \overline{XY}Z + XYZ + \bar{X}Y\bar{Z} + X\overline{YZ} \\
&= X\,(\bar{Y}Z + \overline{YZ}) + \bar{X}\,(\bar{Y}Z + Y\bar{Z}) \\
&= (\overline{Y \oplus Z})\, X + (Y \oplus Z)\, \bar{X}
\end{aligned}
$$

$$
\begin{aligned}
\text{Carry} &= XY + YZ + ZX \\
&= Z\,(Y + X) + XY \\
&= Z\,(Y + X) + (Y + \bar{Y}) + XY \\
&= Z\,(Y + X\bar{Y}) + XY \\
&= YZ + X\bar{Y}Z + XY \\
&= Y\,(Z + X) + X\bar{Y}Z \\
&= Y\,(Z + X)\,(Z + \bar{Z}) + X\bar{Y}Z \\
&= Y\,(Z + X\bar{Z}) + X\bar{Y}Z \\
&= YZ + X\bar{Z}Y + X\bar{Y}Z \\
&= YZ\,(Y + \bar{Y})\,(Z + \bar{Z}) + X\bar{Z}Y + X\bar{Y}Z \\
&= Y\,(\overline{YZ} + YZ) + X\,(Y\bar{Z} + \bar{Y}Z) \\
&= Y\,(\overline{Y \oplus Z}) + X(Y \oplus Z)
\end{aligned}
$$

Fig. 9: Full adder the sum and carry expression

The blue circle defines full adder and the red circle indicates half adder in the Fig. 8. And also shows where full adder and half adder to be considered.

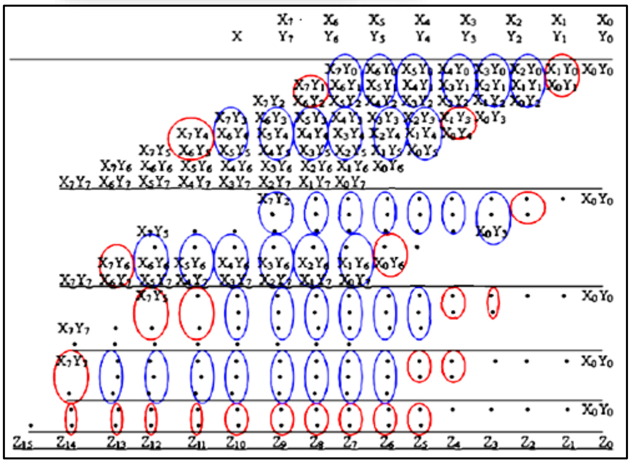
X_7 X_6 X_5 X_4 X_3 X_2 X_1 X_0
X Y_7 Y_6 Y_5 Y_4 Y_3 Y_2 Y_1 Y_0
X_0Y_0
Z_{15} Z_{14} Z_{13} Z_{12} Z_{11} Z_{10} Z_9 Z_8 Z_7 Z_6 Z_5 Z_4 Z_3 Z_2 Z_1 Z_0

Fig. 8: Description of usage of adders

Table 1 shows how the full adder works with the inputs and sum and carry generation possibilities.

Y & Z both are 0 or 1	Sum = X
Y & Z either 0 or 1	Sum = ~X
Y & Z both are 0 or 1	Carry = Y
Y & Z either are 0 or 1	Carry = X

Table 1: working of MUX based Full adder

The third step of the Wallace tree-based multipliers is to add the remaining two rows using a fast adder. Some of the most widely used parallel-prefix adders used for high speed operations are Kogge-Stone, Sklansky, and Brent-Kung. These adders use the same tree topology but differ in terms of logic levels, fan out, and interconnect wires. We used Kogge-Stone adder in both conventional and modified multipliers discussed in this paper. The logic levels for implementation of an*N*-bit Kogge-Stone adder can be calculated by using Logic Levels = [log2(*N*)].

The performance comparison conventional Wallace multiplier and modified Wallace multiplier is detailed below in Table.2 in terms of power, delay and number of transistors.

8 Bits	CWM	PWM
No of stages	4	4
Delay (ns)	10ns	10ns
Avg. Power (mW)	14.25	5.18
No. of transistors	2544	1552

Table 2: Performance of 8 bit wallace tree multiplier

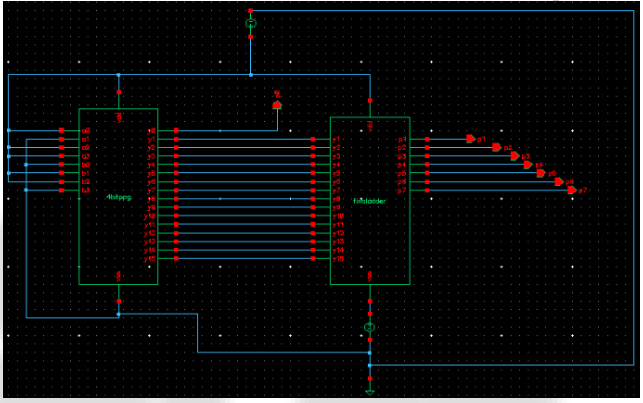

Fig. 9: Implementation of 8 bit Modified Wallace Multiplier

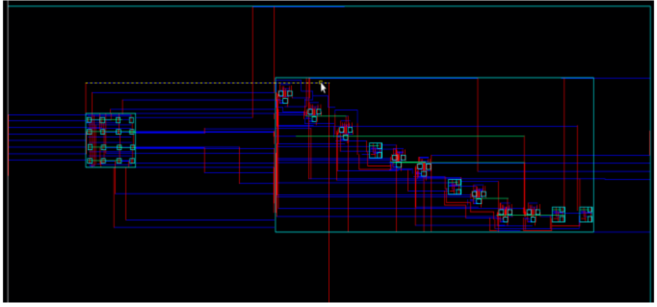

Fig. 10: Layout of 4 bit Wallace Multiplier

All the multipliers are synthesized using cadence using 90nm technology, supply voltage 1.8V, temperature 25 ^{0}C. The designs can be optimized for area, power and delay

IV. Conclusion

This paper presents a method to reduce area and power without any delay difference as the stages remains almost same. Design is synthesized using 90nm technology the synthesis results are verified and as expected has the smallest area compared to conventional.

References

[1] Kokila Bharti Jaiswal, Nithish Kumar V, Pavithra Seshadri and Lakshmi narayanan G 2015 3rd International Conference on Signal Processing,

Communication and Networking (ICSCN)978-1-4673-6823-0/15/$31.00 ©2015 IEEE.

[2] K. S. Ganesh Kumar1, J. Deva Prasannam2, M. Anitha Christy International Journal of Emerging Technology and Advanced Engineering Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Issue 3, March 2014.

[3] International Journal of Applied Engineering Research ISSN 0973-4562 Volume 9, Number 21(2014) pp.8889-8900 © Research India Publications http://www.ripublication.com

[4] M. Jayaprakash, M. Peer Mohamed, Dr. A.Shanmugam Design and Analysis of Low Power and Area Efficient Multiplier. International Journal of Electrical, Electronics and Mechanical Controls. ISSN (Online): 2319-7501. Volume 3 Issue 1 January 2014. Manuscript ID: ISSN23197501-V3I1M2-012014.

[5] P. Jagadeesh, S. Ravi, and K. H. Mallikarjun, “Design of high performance 64 bit mac unit,” in Proceedings of the International Conference on Circuits, Power and Computing Technologies (ICCPCT ’13), pp. 782–786, March 2013.

[6] Shahebaj Khan, Sandeep Kakde, YogeshSuryanvanshi, “VLSI implementation of reduced Complexity Wallace Multiplier Using Energy Efficient CMOS Full Adder,” IEEE International Conference on Computational Intelligence and Computing Research, 2013.

[7] A Novel Low Power and High Speed Wallace Tree Multiplier for RISC Processor “978-1-4244-8679-3/11/$26.00 ©2011IEEE”.

[8] R. S. Waters and E. E. Swartzlander, “A reduced complexity wallace multiplier reduction,” IEEE Transactions on Computers, vol. 59, no. 8, pp. 1134–1137, 2010.