

# VLSI Design and FPGA Implementation of N Binary Multiplier Using N-1 Binary Multipliers

L. Keerthana<sup>1</sup>, M. Nisha Angeline<sup>2</sup>

PG Scholar, Master of Engineering in Applied Electronics, Velalar College of Engineering and Technology, Thindal, Erode, India

M.E., (Ph.D), Assistant Professor (SI.Gr.), Department of ECE, Velalar College of Engineering and Technology, Thindal, Erode, India

**Abstract:** In many digital systems, the most important and basic component is multiplier and adder which are recommended for implementing the concepts of DSP systems, arithmetic and logic functions and multimedia applications. In many real time digital applications, power dissipation and hardware size are the major constraints. In this paper, we propose a method that combines a numerical transformation called number splitting with a shift-and-add decomposition scheme. In this design  $N \times N$  bit multiplication is done by using  $(N-1) \times (N-1)$  bit multiplication. The weight reduction and redundant techniques are used to greatly reduce the strength of multiplication. Various multiplier designs are taken and they are compared based on their speed and area. The speed of the system is increased by reducing the size of the hardware, power consumption and path delay. The designs are modeled using VHDL and implemented in Xilinx Spartan FPGA.

**Keywords:** Numerical transformation, weight reduction, redundant technique, shift-and-add decomposition

## 1. Introduction

In modern digital electronics, multiplication is a very common task. Multiplier is the fundamental hardware block in high performance systems such as digital signal processing, FIR filters, multimedia, etc. Multiplier design performs an important role in low power VLSI system design. The performance of the system is examined by the multiplier's performance because generally the multiplier is the slowest element in the system. During partial products, multipliers need very large latency. The conventional multipliers have many stages so that delay is also large. Furthermore, the multiplier consumes large area. Hence, the major design issue of the multiplier is optimizing the speed and area. Three major steps in multiplication

- Recording and generating of partial products.
- Partial products can be reduced by partial product reduction scheme to two rows. and
- The remaining two rows of partial products are added by using carry propagate adder to get the final product.

We study different multipliers, considering from array multiplier to Wallace multiplier. High speed processing system demand is increased, as a result of extending the signal processing applications and computer applications.

## 2. Literature Review

Several different types of multipliers are studied and compared them based on speed, delay, area, power. So that we can judge, which multiplier was best suited for optimizing all the design targets.

### A. Array multiplier

In 1999, the work has been carried out in array architecture, which is a popular technique to implement digital multipliers because of its regular compact structure. High power dissipation is carried out due to switching of large number of gates during multiplication. In this array, multiplier has  $N-1$  adder stages. Therefore if the previous result is obtained,

then only the addition operation is performed. The advantage of array multiplier is easy to design and having regular layout but the speed will be reduced for wide multiplier. Hence, temporal tiling technique is applied to other digital circuits. A temporally tiled array multiplier achieves 50% and 35% improvements in delay and power dissipation compared with conventional array multiplier. In 2014, various integer multiplier designs are taken and are compared based on area and speed. The high performance integer multiplier is suitable for high-speed VLSI system

### B. Braun's array multiplier

Braun array multiplier is similar to array multiplier; it is developed for reducing the delay and power dissipation caused in array multiplier. It can not only achieve the objective for detecting unidirectional faults but also analyse error detection. In 1996, Berger check code is developed for concurrent error detection. Error detection is done using Berger check prediction technique. The performance of the Berger check prediction Braun array multiplier achieves 100% error detection.

### C. Shift and add multiplier

In 2009, low voltage micro power asynchronous multiplier is designed based on shift add structure for power critical applications such as low clock rate. Power dissipation is reduced by using number of low power techniques at the system level and low power circuit designs have been proved. To perform the entire operations for getting the final product, the conventional architecture for shift and add multipliers require many switching activities. So the dynamic power dissipation is more in conventional architecture. By eliminating or reducing the sources switching activity in the Conventional multiplier, low power architecture of multiplier can be derived. In 2014, a low power structure called By pass zero feed a directly (BZ-FAD) for shift and add multiplier is proposed. The switching activity of conventional shift-and-add multiplier is considerably reduced using this method. This BZ-FAD architecture reduces up to 76% of total switching activity

and the power consumption is reduced up to 30% compared with conventional architecture. This method is used for low power applications.

#### D. Wallace multiplier

The Wallace tree multiplier is faster than a simple array multiplier. However, in addition to the large number of adders required, the Wallace tree's wiring is much less regular and more complicated. As a result, Wallace trees are often avoided by designers because of design complexity. Wallace tree styles use a log-depth tree network for reduction. Wallace tree styles are generally avoided for low power applications, since excess of wiring is likely to consume extra power. While subsequently faster than Carry-save structure for large bit multipliers, the Wallace tree multiplier has the disadvantage of being very irregular, which complicates the task of coming with an efficient layout. In 2013, this paper aims at the reduction of power consumption and latency of Wallace tree multiplier. It is an improved version of tree based multiplier architecture. The proposed Wallace tree multiplier is 44.4% faster than conventional Wallace tree multiplier and power consumption is reduced to 11% by using 4:2, 5:2 compressor is used in this proposed work. In 2015, Wallace tree multiplier is proposed which generally reduce partial products and finally carry select adder is used for final carry propagation path. The modified square root carry select adder (MSCSA) is designed using common Boolean logic. The modification is done at gate level to reduce power consumption and area. In 2015, the area is reduced by using energy efficient CMOS full adder. Wallace tree multiplier is designed for implementing high-speed multipliers. It is a three-stage operation, which leads to reduce the number of stages and reduce the number of transistors. Energy efficient full adders are the important in Wallace tree multiplier plays a vital role. It reduces the hardware complexity, which then reduces an area and power ultimately. The reduced complexity Wallace multiplier (RCWM) method greatly reduces the number of adders with 65-75% than standard Wallace multiplier.

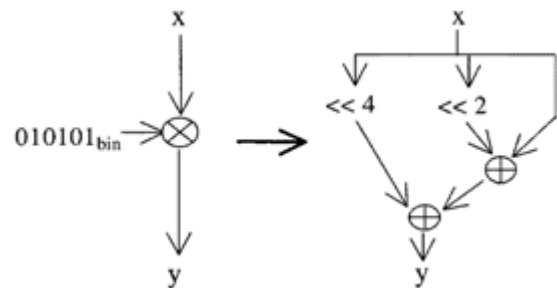
#### E. Booth multiplier

The Original version of Booth's multiplier (Radix - 2) had two drawbacks. The number of add / subtract operations became variable and hence became inconvenient while designing Parallel multipliers. The Algorithm becomes inefficient when there are isolated 1s. In Oct 2013, to improve the performance of pipeline efficiency, high speed and energy efficient VLSBM is proposed. The proposed VLSBM reduces the critical path than the conventional pipelining booth multiplier.

The rest of the paper is organized as follows section III briefly reviews on numerical transformation called number splitting with shift and add decomposition scheme. Section IV describes the architecture description of modes of redundant techniques. Section V, VI describes the experimental results and conclusion.

### 3. Numerical Transformation With Shift and Add Decomposition

The numerical transformation 'globally' changes the constant multiplier and the dataflow graph of the system under design, enabling implementations with fewer shift and add. The decomposition of constant multiplication into efficient implementation with shift and add. Consider an example for shift and add decomposition as shown in Fig 1.



**Figure 1:** Shift and add decomposition.

The number of non-zero digits in the binary representation of the coefficient indicates the required number of shift and add operation.

In this our approach design involves in two steps, 1) decomposition of complex multiplication operation into elementary shift and add operations. 2) Numerically transform the original system to obtain new equivalent architecture with different multiplier coefficients. The number of non-zero digits can be reduced by a novel number representation scheme such as canonical signed-digit (CSD) representation has been used to code multiplier coefficients.

In this work, new multiplier algorithm is proposed based on numerical transformation and VLSI architecture is designed. In this design, the N bit multiplier is reduced to N-1 bit multiplier. The strength of the multiplier is reduced using weight reduction technique. Hence, the system complexity of the multiplier is reduced. The weight of the 4-bit multiplier is 8 their binary value is 1000. Consider two values A=1001 and B=1100 these values are compared with 8. In these, both values are greater than 8. Based on this 4 modes of redundant techniques are classified.

- Both greater than 8 (positive redundant).
- Both lesser than 8 (negative redundant).
- Any one is greater than 8 and other is lesser than 8.
- Combination of all the above modes.

Four types of mode architecture are designed. The 4 bit multiplier can be reduced to 3-bit multiplier by subtracting the given values with 8. Then the last MSB bit is omitted.

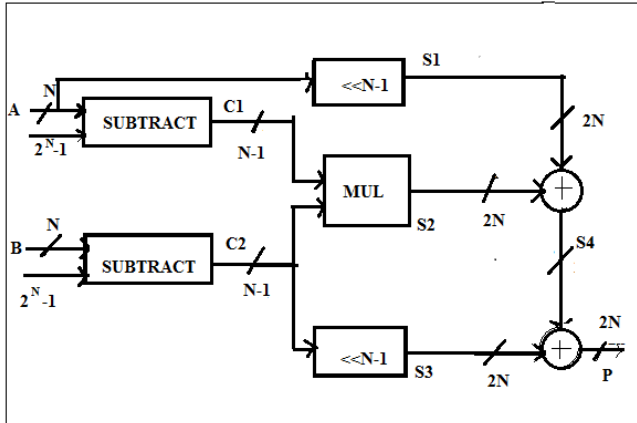
#### 4. Modes Of Redundant Architecture

In this section, four different modes of redundant calculation are carried out. The results are proved theoretically.

##### A. Mode I –Positive redundant

The Fig 2 shows the block diagram of both greater than  $2^{N-1}$ . In this positive redundant algorithm is used. The input values of both A and B should be greater than the numerical

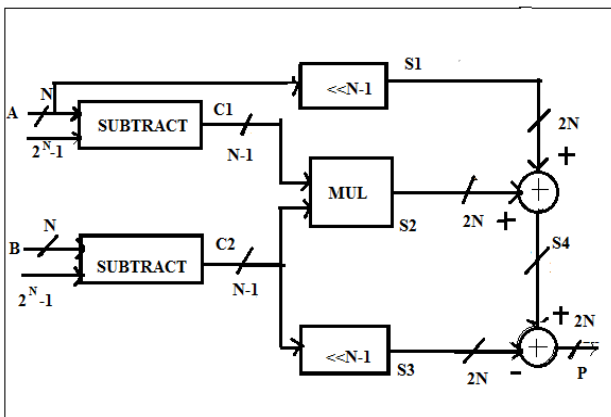
value of  $2^{N-1}$  bit. The input values of A and B is subtracted with the numerical value  $2^{N-1}$ . The redundant derived output is given as a N-1 bit input values to the multiplier. The Multiplier performs the N-1xN-1 multiplication. The value of A is right shifted into N-1bit and then added with output of the multiplier using the adder. The redundant value of B is right shifted into N-1 bit and then added with the output of the adder.



**Figure 2:** Architecture of multiplier for Mode I

**B. Mode II-Negative redundant**

The Fig 3 shows the block diagram of multiplier for negative redundant. In this negative redundant, both inputs are less than  $2^{N-1}$ . The input values of A and B is subtracted from  $2^{N-1}$  to derive redundant. The redundant are given as input values to the multiplier. The Multiplier performs the N-1xN-1 multiplication. The value of A is right shifted into N-1bit and then added with output of the multiplier using the adder. The redundant value of B is right shifted into N-1 bit and then subtracted from the output of the adder.

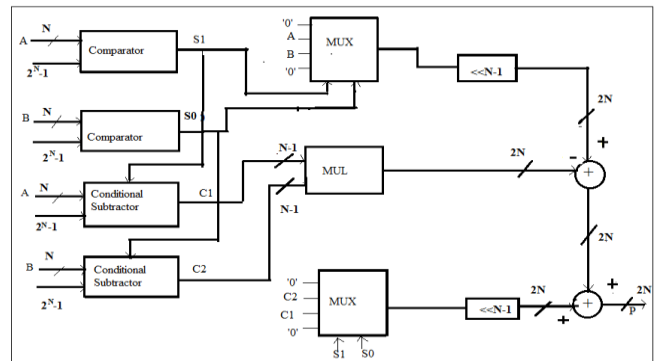


**Figure 3:** Architecture of multiplier for Mode II

**C. Mode III-One negative redundant**

The Fig 4 shows the block diagram of multiplier with +ve and -ve redundant. In this, any one of the input values of A and B should be less than the numerical value of  $2^{N-1}$  bit. The input vales of A and B are compared with numerical value  $2^{N-1}$  then S0,S1 outputs are produced based on this condition the input values of A and B is subtracted with the numerical value  $2^{N-1}$ . The redundant are given as input values to the multiplier. The Multiplier performs the N-1 xN-1 multiplication. Based on the condition value of S0, S1

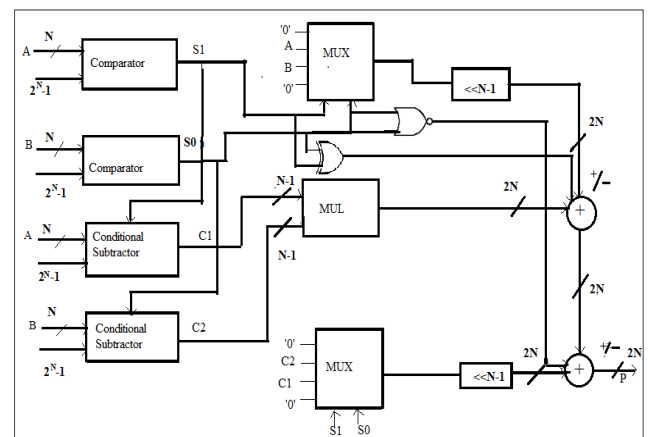
the 4 x 1 Multiplexer performs the operation produces one output it is right shifted into N-1bit and then subtracted with output of the multiplier using the adder. The another mux value is right shifted into N-1 bit ant then added from the output of the adder.



**Figure 4:** Architecture of multiplier for Mode III

**D. Mode IV-Combined Structure Of Multiplier**

The Fig 5 shows the block diagram of combined structure of multiplier. This performs all the three redundant operation. The input vales of A and B are compared with numerical value  $2^{N-1}$  then S0,S1 outputs are produced based on this condition the input values of A and B is subtracted with the numerical value N. The redundant are givenas input values to the multiplier. The Multiplier performs the N-1xN-1 multiplication. Based on the condition value of S0,S1 the 4 x 1 Multiplexer performs the operation produces one output it is right shifted into N-1bit .



**Figure 5:** Combined structure of multipliers

The EX-OR and NOR gate performs the operation having the input value of S0,S1. Based on the EX-OR gate output one of the mux output is subtracted/ added with output of the multiplier using the adder. Then Based on NOR gate output the other mux output is added/subtracted with the output of the adder output .Finally, the partial product output is achieved similar to the binary multiplier output.

**5. Experimental Results**

**Table 1:** Analysis table for various multipliers comparing Delay

Methods	Area (LUTs)			
	4BIT	8BIT	16BIT	32BIT
Array Multiplier	30	125	511	2033
Shift and add multiplier	26	123	508	2044
Braun Multiplier	33	77	294	1194
Wallace Tree Multiplier	27	122	512	2077
Combined multiplier using array	17	191	675	2174
Combined multiplier using shift and add	58	188	687	2368
Combined multiplier using Braun's	61	191	668	2339
Combined multiplier using Wallace	17	93	480	1877

For comparison, we have considered both conventional methods and the proposed multiplier for different bit values in VHDL. The VHDL codes are implemented using Xilinx Vertex FPGA. The power analysis of the gate level structure and delay calculations is conducted using Xilinx ISE tool. The array multiplier is widely used due to its linear structure. It is advantageous for minimum number of bits. The results of table 1 and 2 show the values of the proposed multiplier for different bit values. The multiplier unit is the important basic unit in most of the applications. The array multiplier and shift-and-add multipliers are widely used in most of the applications. In the table 2, the proposed multiplier is compared with existing multipliers for the standard bit sizes 4,8,16 and 32. From the result, it is clear that the proposed method is well suited for higher order bits. In the lower bits, the proposed method doesn't show advantageous result. For the bit size 4, the delay is high for the proposed method. But for the higher order bits, the delay is reduced. Similarly, the number of LUTs is also increased for different bit values. Comparatively, the proposed method reduces the area to the extent of 71%. The delay is reduced to the extent of 77%. By reducing the delay, the speed can also be increased.

**Table 2:** Analysis table for the proposed multiplier comparing area

Methods	Delay in nS			
	4BIT	8BIT	16BIT	32BIT
Array Multiplier	15.269	31.111	62.437	123.387
Shift and add multiplier	15.677	33.840	63.089	124.112
Braun Multiplier	13.088	23.331	62.437	127.776
Wallace Tree Multiplier	12.756	22.863	44.258	87.776
Combined multiplier using array	10.366	34.229	68.913	117.318
Combined multiplier using shift and add	16.921	36.221	63.517	127.250
Combined multiplier using Braun's	17.454	34.229	65.706	150.121
Combined multiplier using Wallace	10.366	20.323	38.258	80.456

**Table 3:** Analysis table for the proposed multiplier comparing power

Methods	Power in mW			
	4BIT	8BIT	16BIT	32BIT
Array Multiplier	0.298	0.312	0.368	0.440
Shift and add multiplier	0.310	0.368	0.501	0.636
Braun Multiplier	0.113	0.149	0.406	0.706
Wallace Tree Multiplier	0.113	0.154	0.375	0.706
Combined multiplier using array	0.127	0.268	0.519	0.723
Combined multiplier using shift-and-add	0.177	0.263	0.575	0.962
Combined multiplier using Braun's	0.133	0.144	0.255	0.272
Combined multiplier using Wallace	0.114	0.150	0.205	0.240

## 6. Conclusion

In this project, various multipliers are studied and also the impact of power dissipation, delay and area have been analyzed. The proposed method increases the speed for higher order bits, the modes of redundant architectures are implemented using VHDL and in Xilinx ISE tool and results thus obtained are analyzed. Various multipliers are implemented and analyzed and compared with one another to select an appropriate architecture for our purpose. The proposed method is not advantageous for minimum number of bits. But the area and delay calculation is very optimal in higher order bits. This method is suitable for higher order bits. The architecture can also be modified to improve the speed further.

## References

- [1] Shivaling S. Mahant-Shetti, Poras T. Balsara, Senior Member, IEEE, and Carl Lemonds, Member, IEEE, Mar 1999, "High Performance Low Power Array Multiplier Using Temporal Tiling" IEEE transactions on very large scale integration (vlsi) systems, vol. 7, no. 1.
- [2] Anjana s and pradeep c, 2014 "High speed integer multiplier designs for reconfigurable systems" International conference on control ,communication and computational technologies (ICCICCT)
- [3] Christian Martyn Jones, Satnam Singh Dlay & Raouf Gorgui Naguib, 1996 "Berger check prediction for concurrent error detection in the Braun array multiplier"
- [4] Bah-Hwee Gwee, Joseph S. Chang, Yiqiong Shi, Chien-Chung Chua, and Kwen-Siong Chong, July 2009, "A Low-Voltage Micropower Asynchronous Multiplier With Shift-Add Multiplication Approach" IEEE transactions on circuits and systems—i: regular papers, vol. 56, no. 7.
- [5] Mottaghi-Dastjerdi M., Afzali-Kusha A., and Pedram M. Feb 2014, "BZ-FAD: A Low-Power Low-Area Multiplier Based on Shift-and-Add Architecture" IEEE Trans On Very Large Scale Integration (Vlsi) Syst, vol. 17, no. 2.
- [6] Huy T. Nguyen and Abhijit Chatterjee, August 2000 "Number-Splitting with Shift-and-Add Decomposition For Power and Hardware Optimization in Linear DSP Synthesis" IEEE transactions on very large scale integration (vlsi) systems, vol. 8, no. 4,

- [7] DamarlaParadhasaradhi, M. Prashanthi, and N Vivek,2014.” Modified Wallace Tree Multiplier using Efficient Square Root Carry Select Adder” IEEE J. Solid-State Circuits, vol. 23, no. 4, pp. 464–484.
- [8] Sandeep Kakde, Shahebaj Khan, Pravin Dakhole, Shailendra Badwaik,2015.” Design of Area and Power Aware Reduced Complexity Wallace Tree Multiplier” International Conference on Pervasive Computing.
- [9] Shin-Kai Chen, Chih-Wei Liu, Member, Ieee, Tsung-Yi Wu, And An-Chi Tsai, October 2013, –Design And Implementation Of High-Speed And Energy-Efficient Variable-Latency Speculating Booth Multiplier (Vlsbm)”IEEE Transactions On Circuits And Systems—I: Regular Papers, Vol. 60, No. 10, 2631.
- [10] Chandrakasan. A and Brodersen .R, Apr.1992, —Low-power CMOS digital design,”IEEE J. Solid-State Circuits, vol. 27, no. 4, pp. 473–484.