

Designing of Various types of multipliers and their implementation on FPGA

Hemanshi Chugh
 Department of Electronics and Communication
 Dr. Akhilesh Das Gupta Institute of Technology and Management
 New Delhi, India
 hemanshi.chugh@gmail.com

Abstract - This paper presents the detailed study of four different multiplier architectures based on Vedic Multiplier, Booth encoded Wallace tree multiplier, Baugh Wooley multiplier and Braun multiplier. Here these architectures are implemented for 4, 8 bits. All the multipliers are coded in Verilog HDL and simulated in Model SIM and implemented on Xilinx Spartan 3E FPGA board. All multipliers are then compared for their performance based on LUTs and path delays. Furthermore, some modifications to speed up the overall multiplier architecture are also presented in this report. Results show that in case of signed multipliers Baugh Wooley gives better results as compared to Booth encoded Wallace tree. And in case of unsigned multipliers, Braun multiplier using carry skip adder shows better results than Vedic multiplier.

Index terms- Vedic Multiplier, Booth encoded Wallace tree multiplier, Baugh Wooley multiplier, Braun multiplier

I. INTRODUCTION

Multiplication is the most crucial fundamental function in arithmetic operations. Multiplication is the basic building block for several DSP processors, Image processing and many others. Hence, a multiplier accounts to a fairly large block of a computing system.

The demand for high speed processing has been increasing as a result of expanding signal processing applications. Higher throughput arithmetic operations are important to achieve the desired performance in many real-time signal and image processing applications. Reducing the time delay and power consumption are very essential requirements for many applications. Since multiplication dominates the execution time of most DSP algorithms, so there arises a need of high speed multiplier. The four different types of multipliers are presented in this report, each offering different advantages and having trade off in terms of speed, circuit complexity, area and power consumption. Particular multiplier architecture can be chosen based on the application.

Multiplier architectures fall into two categories: "signed" and "unsigned" multipliers.

Different multipliers are discussed in below sections.

VEDIC MULTIPLIER

The multiplier is based on an algorithm **Urdhva Tiryakbhyam** (Vertical & Crosswise) of ancient Indian Vedic Mathematics.[1] The algorithm can be generalized for $n \times n$ bit number. This algorithm shows how to handle multiplication of a larger number ($N \times N$, of N bits each) by breaking it into smaller numbers of size ($N/2 = n$, say) and these smaller numbers can again be broken into smaller numbers ($n/2$ each) till 2×2 basic multiplier block. Hence, the whole multiplication process is simplified with ease. The basic building block of the Vedic multiplier architecture is as shown in figure 1.

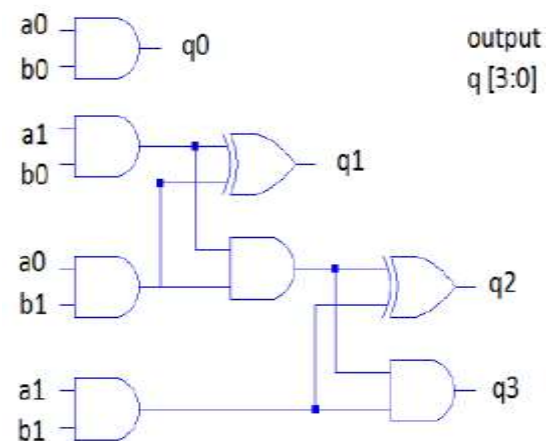


Fig.1: 2 x 2 Vedic Multiplier.

The 2x2 blocks are instantiated in the design for 4 x 4 multiplier (Fig. 2). The adders used in a 4x4 multiplier are 4 bit and 6 bit adders. The partial product addition in Vedic multiplier is realized using ripple carry adder[8], carry select adder and carry skip adder technique to observe variations.

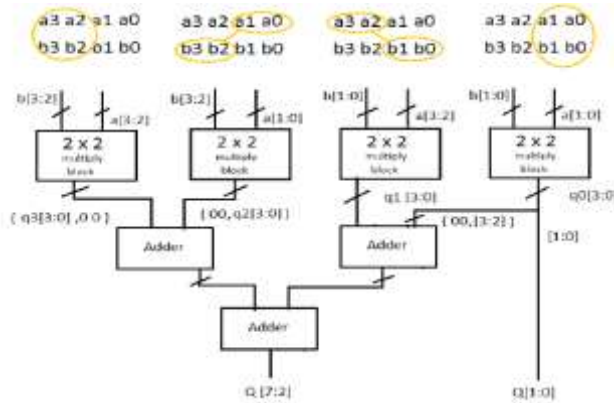


Fig.2: 4x4 Vedic Multiplier.

The 4x4 multiplication is done in a single line in Urdhva Tiryagbhyam sutra, whereas in shift and add (conventional) method, four partial products have to be added to get the result. Thus, by using Urdhva Tiryagbhyam Sutra in binary multiplication, the number of steps required calculating the final product will be reduced and hence there is a reduction in computational time and increase in speed of the multiplier.

Likewise, an 8x8 multiplier is designed using 4x4 blocks, 8bit and 12bit adders.

Modification in Vedic Multiplier

A modification was done in the architecture where a 4x4 multiplier was directly implemented using 9 full adders and a 5 bit input special adder. The architecture uses three stages of addition as shown.

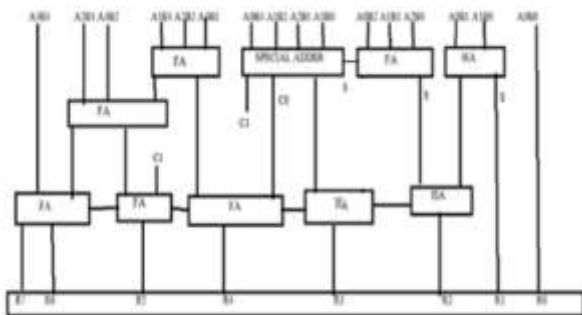


Fig.3: 4x4 Modified Vedic Multiplier.

Adder Design

The special Adder (Fig. 4) in the above architecture, has the following logic:

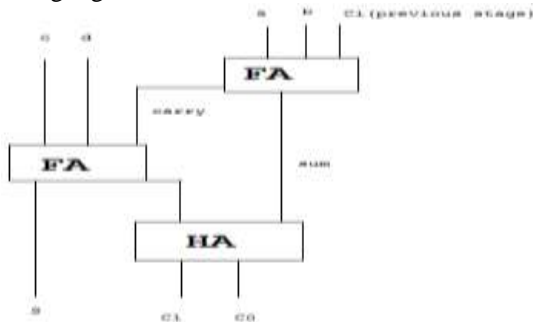


Fig.4: 5 input adder

The adder reduces the total number of full adders in the conventional $4 * 4$ Vedic multiplier. Thus effectively reducing delay.

BOOTH ENCODED WALLACE TREE MULTIPLIER

The Booth Encoded Wallace Tree Multiplier[2] is the best known algorithm for signed as well as unsigned multiplication for multiplying large numbers. Multiplication involves two main steps: the generation of partial products[3] and accumulation of all the partial products. Hence, the speed of multiplication can be increased by either deducting the number of partial products and/or speeding up the accumulation of partial products

In this paper, the number of partial products are reduced to $(MD/2 - 1)$ partial products using booth encoded algorithm[9], high speed computation is achieved using Wallace tree algorithm[4] and the final addition of the carry propagations is done using Ripple carry adder, carry skip adder and carry select adder.

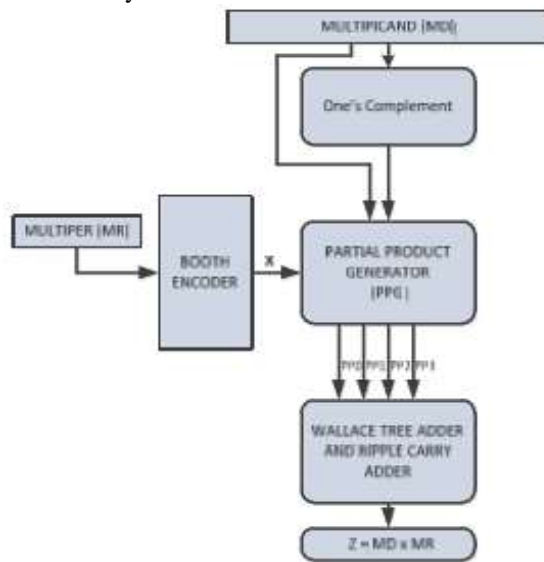


Fig.5: Booth Encoded Wallace Tree Multiplier

BAUGH WOOLEY MULTIPLIER

The Baugh-Wooley multiplication[5] is one of the efficient methods to handle the sign bits and is used effectively for multiplication of both signed and unsigned numbers. This approach has been developed in order to design regular multipliers used productively in digital filters.

Baugh-Wooley Multiplication on signed operands makes sure that the signs of all partial products are positive using 2's complement representation. Thus each partial product has to be sign extended to the width of the final product in order to form a correct sum for the adder chain. Hence, The Partial Products are adjusted such that negative sign moves to the last step, which in turn maximizes the regularity of the multiplication array.

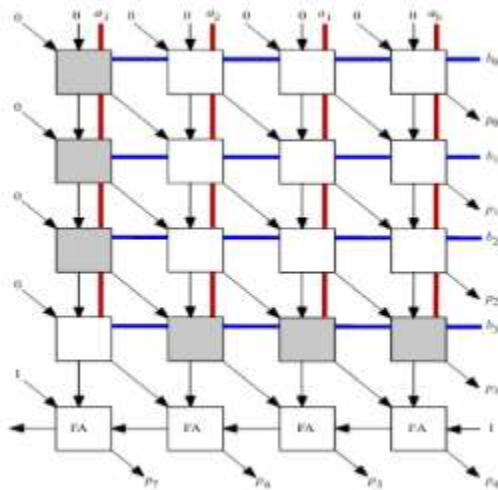


Fig 6: 4x4 Baugh Wooley Multiplier

The block representation of 4x4 Baugh Wooley multiplier[6] is as shown above. The white cells represent the positive bit multiplication whereas the grey cells represent negative bits representation.

The structure of the white and grey cell is as shown below

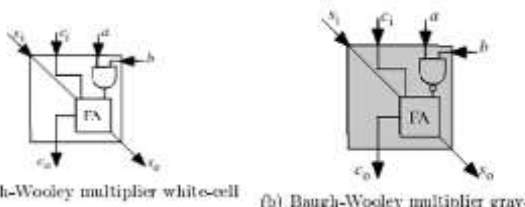


Fig 7: Baugh-Wooley Cells

However, the conventional Baugh Wooley architecture uses a ripple carry adder[8] in the final stage. A modified Baugh Wooley architecture is presented by replacing the ripple carry adder of the final stage by an N bit linear carry select and carry skip adder.

BRAUNS MULTIPLIER

Brauns multiplier is a parallel array multiplier, generally also called as carry save array multiplier.

The basic structure consists of an array of AND gates and adders arranged in an iterative manner. An N*N bit Brauns multiplier is constructed with n (n-1) adders, n² AND gates and (n-1) rows of Carry Save Adder.

The Braun Bypassing techniques[7] help in reducing dynamic power than the array multiplier.

In Row Bypassing multiplier, if the multiplier bit b_j is zero then the addition operations in the j-th row can be bypassed; thus directly providing (j-1)th row outputs directly to (j+1)th row. In a column bypassing based Braun multiplier, if the multiplicand bit a_i is zero, then the addition operation in (i+1)-th row can be bypassed.

The row bypassing and column bypassing techniques only reduce the critical path delay, but also substantially reduce the overall power dissipation as unrequired transitions don't take place. A low power multiplier with Row and Column bypassing can be obtained by merging the above two techniques and simplifying the full adder. This is shown in Fig. 8.

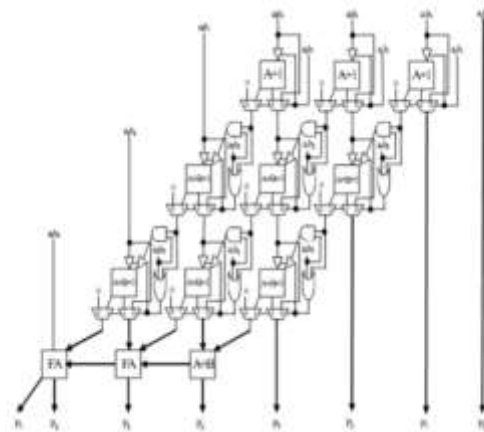


Fig 8: Braun Row and Column Bypassing Technique. A modified Braun architecture is compared by using the ripple carry adder[8], carry select and carry skip adder in the final stage.

RESULTS AND DISCUSSIONS

Vedic Multiplier

The comparison results for Vedic multiplier are shown in table 1 and table 2:

| 4x4 | VEDIC USING CSkA | VEDIC USING RCA | VEDIC USING CSLA | VEDIC ON MODIFICATION |
|------------|------------------|-----------------|------------------|-----------------------|
| No. Of LUT | 31 | 31 | 31 | 37 |
| No. Of CLB | 16 | 16 | 16 | 19 |
| Delay(ns) | 9.221 | 9.221 | 9.213 | 9.275 |

Table 1: Vedic 4x4 multiplier results

| 8x8 | VEDIC USING CSkA | VEDIC USING RCA | VEDIC USING CSLA | VEDIC ON MODIFICATION |
|------------|------------------|-----------------|------------------|-----------------------|
| No. Of LUT | 176 | 173 | 177 | 197 |
| No. Of CLB | 88 | 87 | 89 | 99 |
| Delay(ns) | 16.474 | 16.368 | 17.323 | 16.401 |

Table 2: Vedic 8x8 multiplier results

| 16x16 | VEDIC USING CSKA | VEDIC USING RCA | VEDIC USING CSLA | VEDIC ON MODIFICATION |
|-------------|------------------|-----------------|------------------|-----------------------|
| No. Of LUT | 843 | 811 | 825 | 907 |
| No. Of CLBs | 422 | 406 | 413 | 454 |
| Delay(ns) | 26.638 | 26.646 | 29.157 | 26.279 |

Table 3: Vedic 16x16 multiplier results

Results show that our modification on Vedic multiplier is efficient for higher bit multiplication. It increases the area but reduces the delay.

Booth Encoded Wallace Tree Multiplier

| 4X4 | BOOTH ENCODED WALLACE TREE (RCA/CSKA) | BOOTH ENCODED WALLACE TREE (CSLA) |
|-------------|---------------------------------------|-----------------------------------|
| No. Of LUT | 41 | 55 |
| No. Of CLBs | 21 | 28 |
| Delay(ns) | 10.810 | 8.985 |

Table 4: 4x4 Booth Encoded Wallace Tree multiplier results

| | BEWT USING RCA | BEWT USING CSKA | BEWT USING CSLA |
|-------------|----------------|-----------------|-----------------|
| No. Of LUT | 187 | 185 | 177 |
| No. Of CLBs | 94 | 93 | 89 |
| Delay (ns) | 15.629 | 16.571 | 17.469 |

Table 5: 8 x 8 Booth Encoded Wallace Tree multiplier results

Results show that booth encoded Wallace tree using ripple carry adder gives the minimum delay as compared to other modifications.

Baugh Wooley Multiplier

| 4X4 | BAUGH WOOLEY (RCA/CSKA/CSLA) |
|-------------|------------------------------|
| No. Of LUT | 29 |
| No. Of CLBs | 15 |
| Delay(ns) | 9.407 |

Table 6: 4 x 4 Baugh Wooley multiplier results

| | B.W USING CSLA | B.W USING CSKA | B.W USING RCA |
|-------------|----------------|----------------|---------------|
| No. Of LUT | 131 | 133 | 124 |
| No. Of CLBs | 66 | 69 | 62 |
| Delay (ns) | 14.541 | 15.110 | 15.434 |

Table 7: 8 x 8 Baugh Wooley multiplier results

The results show that Baugh Wooley architecture using carry select adder gives the minimum delay with slight increase in area.

Braun Multiplier

| | ROW BYPASSING | COLUMN BYPASSING | ROW AND COLUMN BYPASSING (RCA) | ROW AND COLUMN BYPASSING (CSKA) | ROW AND COLUMN BYPASSING (CSLA) |
|-------------|---------------|------------------|--------------------------------|---------------------------------|---------------------------------|
| NO. OF LUTs | 206 | 132 | 121 | 126 | 132 |
| CLBs | 103 | 66 | 61 | 63 | 66 |
| DELAY(ns) | 19.301 | 14.527 | 17.301 | 15.426 | 15.445 |

Table 8: Braun multiplier results

The results show that column bypassing technique gives the minimum delay among all modifications.

CONCLUSION

The results based on parameters area and delay has been observed for all the multipliers with various last stage adders.

In case of signed multipliers

1. Baugh Wooley multiplier gives better results as compared to booth encoded Wallace tree.
2. Carry select adder reduces the delay.
3. Ripple carry adder minimizes the area.

Thus for faster computation, i.e. in encryption, Baugh Wooley multiplier with carry select adder in the last stage is preferred.

In case of unsigned multipliers,

1. As the number of input bits increases, Vedic using carry skip adder shows better results in terms of delay as compared to ripple carry adder and carry select adder.
2. Our modification for small inputs is not successful, but as input bit size increases, delay reduces, but area increases significantly.
3. Braun consumes less area and power. Delay of Vedic and Braun is almost equal. Thus for small inputs, we can prefer Braun multiplier.
4. Braun using carry skip adder shows the best results amongst all.

REFERENCES

[1] Prof J M Rudagil, Vishwanath Amble, VishwanathMunavalli , Ravindra Patil, Vinaykumar Sajjan, “**Design and Implementation of Efficient Multiplier using Vedic Mathematics**”, Proc. of int. Conf, on Advances in Recent Technologies in Communication and Computing 2011.

[2] Rahul D Kshirsagar, Aishwarya.E.V., Ahire Shashank Vishwanath, P Jayakrishnan , “**Implementation of Pipelined Booth Encoded Wallace Tree Multiplier Architecture**”, 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE) ,IEEE.

[3] Minu Thomas M. Tech Electronics & Communication Engineering (VLI&ES), “**Design and Simulation of Radix-8 Booth Encoder Multiplier for Signed and Unsigned Numbers**”.

[4] C. S. Wallace, “**A Suggestion for a Fast Multiplier**”, Electronic Computers, IEEE Transactions, vol.13, Page(s): 14-17, Feb. 1964.

[5] Indrayani Patle, Akansha Bhargav, Prashant Wanjari, “**Implementation of Baugh-Wooley Multiplier Based on Soft-Core Processor**”, IOSR Journal of Engineering (IOSRJEN) e-ISSN: 2250-3021, p-ISSN: 2278-8719 Vol. 3, Issue 10 (October. 2013), ||V3|| PP 01-0.

[6] Jipsa Antony, Jyotirmoy Pathak, “**design and implementation of high speed baugh wooley and modified booth multiplier using cadence rtl**”, IJRET: International Journal of Research in Engineering and Technology ISSN: 2319-1163 | pISSN: 2321-7308.

[7] Anitha R, Alekhya Nelapati, Lincy Jesima W, V. Bagyaveereswaran, “**Comparative Study of High performance Braun’s Multiplier using FPGAs**”, IOSR Journal of Electronics and Communication Engineering (IOSRJECE) ISSN: 2278-2834 Volume 1, Issue 4 (May-June 2012), PP 33-37.

[8] Kungching, Chen (2005). “**Types of adders**”.

[9] Sukanya B, Kothainachiar S, “**Booth Encoded Area Efficient Parallel Tree Reduced Multipliers**”, International Journal of Communications and Engineering, Volume 03– No.3, Issue: 04, March 2012.

[10] Premananda B.S., Samarth S. Pai*, Shashank B. *, Shashank S. Bhat*, “**Design and Implementation of 8-Bit Vedic Multiplier**”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (An ISO 3297: 2007 Certified Organization) Vol. 2, Issue 12, December 2013.

[11] Ms. Madhu Thakur ,Prof. Javed Ashraf., “**Design of Braun Multiplier with Kogge Stone Adder & It’s Implementation on FPGA**”, International Journal of Scientific & Engineering Research, Volume 3, Issue 10, October-2012.

[12] Pramodini Mohanty, Rashmi Ranjan, “**An Efficient Baugh-Wooley Architecture for Both Signed & Unsigned Multiplication**”, International Journal of Computer Science & Engineering Technology (IJCSSET), ISSN: 2229-3345, Vol. 3 No. 4 April 2012.

[13] Snehal R Deshmukh, Dept of E&TC SSGMCOE Shegaon, India (MS) ,Prof. Dinkar L Bhombe Dept of E&TC SSGMCOE Shegaon, India (MS), “**High Performance Multiplier using Booth Algorithm**”, International Journal of Engineering Research & Technology (IJERT) Vol. 3 Issue 4, April – 2014

[14] P. S. Tulasiram*, D. Vaithyanathan, R. Seshasayanan,” **Implementation of Modified Booth Recoded Wallace Tree Multiplier for fast Arithmetic Circuits**”, International Journal of Advanced Research in Computer Science and Software Engineering , ISSN: 2277 128X , Volume 4, Issue 10, October 2014.