

# An Analysis on Designing Issues of Unsigned Multiplier Using CLAA and CSLA

Sweta Sankhediya<sup>1</sup>, Asst. Prof. Yogesh Mishra<sup>2</sup>, Prof. Sachin Bandewar<sup>3</sup>

<sup>1</sup>M-Tech Research Scholar, <sup>2</sup>Research Guide, <sup>3</sup>HOD,

Department of Electronics & Communication Engineering, SSSCE, Bhopal

*Abstract-* A multiplier is one of the key hardware blocks in most digital and high performance systems such as digital signal processors, microprocessors & FIR filters. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following- high speed, regularity of layout, low power consumption, and hence less area or even combination of them in multiplier. By designing the suitable for various high speed, low power, and compact. Area and speed are two conflicting constraints. Subsequently improving speed results always in larger areas. In this review paper numerous researches have been analyzed and we try to find out the best trade off solution among the both of them. As we know multiplication goes in two basic steps.

*Keywords-* CSLA, CLAA, Area, Delay & Array Multiplier.

## I. INTRODUCTION

As the scale of integration keeps growing, many sophisticated signal processing systems are being designed on a VLSI chip. The signal processing applications not only demand great computation capacity but also consume considerable amount of energy. As performance and Area remain to be the two major design tolls, power consumption has turned out to be a critical concern in today's VLSI system design [11]. The need for low-power VLSI system arises from two main forces. The steady growth of operating frequency and processing capacity per chip, huge currents have to be delivered and the heat due to large power consumption must be removed by proper cooling methods. Battery life in portable electronic devices is restricted. Low power plan directly leads to prolonged operation time in these portable devices.

Multiplication is a fundamental operation in most signal processing algorithms. Multipliers have big area, long latency and consume considerable power. Consequently low-power multiplier design has been an important part in low-power VLSI system design. There has been wide work on low-power multipliers at technology, physical, logic levels and circuit. The system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest element in the system. In

addition, it is generally the most area consuming. Therefore, optimizing the speed and area of the multiplier is a major design issue. Though, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. Accordingly, a whole spectrum of multipliers with different area- speed constraints has been designed with fully parallel.

Parallel Multipliers at one end of the spectrum and fully serial multipliers at the other end. In between digit are serial multipliers where single digits consisting of several bits are operated on. The multipliers have moderate performance in both speed and area. Though, existing digit serial multipliers have been plagued by complicated switching systems and/or irregularities in design. Radix  $2^n$  multipliers which operate on digits in a parallel fashion instead of bits bring the pipelining to the digit level and avoid most of the above problems. The structures are modular and iterative. The pipelining done at the digit level brings the benefit of constant operation speed irrespective of the size of the multiplier. Clock speed is only determined by the digit size which is already fixed before the design is implemented.

### *Power Optimization*

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 the digital CMOS design, well-known power-delay product is commonly used to assess the merits of designs. In a sense, this can be shown as  $\text{power} \times \text{delay} = (\text{energy}/\text{delay}) \times \text{delay} = \text{energy}$ , which implies delay is irrelevant [13].

### *Low-Power Multiplier Design*

Multiplication consists of three steps: generation of (PPG) partial products, reduction of (PPR) partial products, and finally (CPA) carry-propagate addition. Generally there are sequential and combinational multiplier implementations. Only consider the combinational case here because the scale of integration now is large enough to accept parallel multiplier implementations in digital VLSI designs. Different

multiplication algorithms vary in the approaches of PPR, PPG, and CPA. For PPG, radix-2 is the easiest. To reduce the no. of PPs and consequently [13] reduce the area/delay of PP reduction, one operand is typically recoded into high-radix digit sets. The majority one is the radix-4 digit set  $\{-2, -1, 0, 1, 2\}$ . For PPR, two alternatives exist: reduction by rows and performed by an array of adder, and reduction by columns, performed by an array of counters. The ultimate CPA requires a fast adder scheme because it is on the critical path. Final CPA is postponed if it is advantageous to keep redundant results from PPG for further arithmetic operations.

*Language & Tools Used*

XILINX can be used for our programming. We may consider the VHDL as our primary language. For the test bench waveforms also we also used Xilinx to write our own test benches. Model Synthesis work Map report all features in Xilinx helped us a lot. We used Xilinx’s XPower Estimator (XPE) tool in order to calculate power consumed in any arithmetic circuit. For calculation of the power using Xilinx’s XPE we need to generate the map report file in XILINX which will be saved in the same directory with an extension “.mp”. But in the later part of the project we used SYNOPSIS tool for finding out Power and delay and Area calculations

*Adders*

Addition is the most common and often used arithmetic operation on microprocessor, digital signal processor, particularly digital computers. As well it serves as a building block for synthesis all other arithmetic operations. Consequently, regarding the efficient implementation of an arithmetic unit, binary adder structures become a very critical hardware unit. In any book on computer arithmetic, it looks that there exists a large number of different circuit architectures with different performance characteristics and widely used in the practice [14].

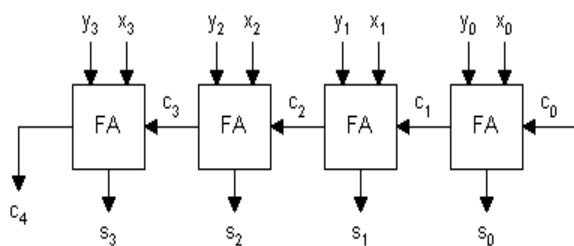


Fig. 1.1 A 4-bit Ripple Carry Adder

*Ripple Carry Adders (RCA)*

The well known adder architecture, ripple carry adder has been composed of cascaded full adders for n-bit adder, as shown in given figure.1. It is constructed by cascading full adder blocks in series. Carry out of one step is fed directly to the carry-in of the next step. For an n-bit parallel adder it necessitates n full adders [12].

- Not very efficient when large number bit numbers are used.
- Delay increases linearly with bit length.

*Delay*

Delay from Carry-in to Carry-out is more important than from A to carry-out or carry-in to SUM, because of the carry-propagation chain will determine the latency of the whole circuit for a Ripple-Carry adder. The Consideration of the above worst-case signal propagation path we can thus write the following equation. For a k-bit RCA most awful case path delay is [11].

$$T_{RCA-kbit} = T_{FA}(x_0, y_0, c_0) + (k - 2) \cdot T_{FA}(C_{in}, C_i) + T_{FA}(C_{in}, S_{k-1})$$

II. SYSTEM MODEL

*Carry Select Adders (CSLA)*

In Carry select adder scheme, blocks of bits are added in two ways: one assuming a carry-in of 0 and the other with a carry-in of 1.

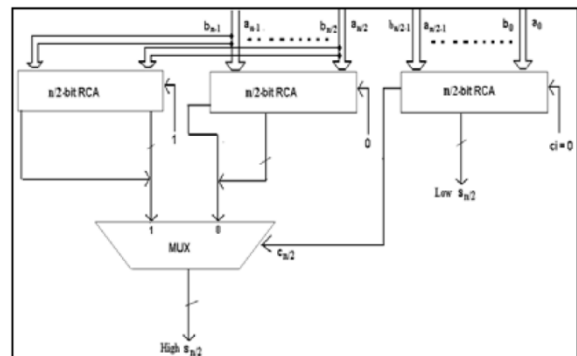


Fig. 1.2 A Carry Select Adder with 1 level using n/2- bit RCA

That results in two precomputed sum and carry-out signal pairs later as the block’s true carry-in (ck) becomes known, the correct signal pairs are selected. In general multiplexers are used to propagate carries.

- Because of multiplexers larger area is required.
- Have a lesser delay than Ripple Carry Adders (half delay of RCA).
- Hence we always go for Carry Select Adder while working with smaller no of bits.

### Carry Look Ahead Adders

Carry Look Ahead Adder (CLA) can produce carries faster due to carry bits generated in parallel by an additional circuitry whenever inputs change. This method uses carry bypass logic to speed up the carry propagation [13].

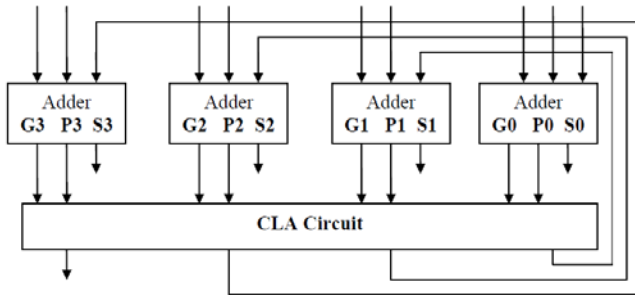


Fig. 1.3 4-BIT CLA Logic equations

Let  $a_i$  and  $b_i$  be the augends and addend inputs,  $c_i$  the carry input,  $s_i$  and  $c_{i+1}$ , the sum and carry-out to the  $i$ th bit positions. If the auxiliary functions,  $p_i$  and  $g_i$  called propagate and generate signals.

- As we increase the no of bits in the Carry Look Ahead adders, complexity raises due to the no. of gates in the expression  $C_{i+1}$  increase. So practically it's not desirable to use the traditional CLA shown above because it increase the Space required and the power too.
- Instead we will use here Carry Look Ahead adder (less bits) in levels to create a larger CLA. Generally smaller CLA may be taken as a 4-bit CLA. Consequently it can be defined the carry look ahead over a group of 4 bits. Therefore now redefine terms carry generate as [Group Generated Carry]  $g[i, i+3]$  and carry propagate as [Group Propagated Carry]  $p[i, i+3]$  which are defined below.

### III. LITERATURE REVIEW

Vijayalakshmi, V., Seshadri, R. & Ramakrishnan, S.[1] Investigated the comparison of the VLSI design of the carry look-ahead adder (CLAA) based 32-bit unsigned integer multiplier and the VLSI design of the carry select adder (CSLA) based 32-bit unsigned integer multiplier. The CLAA based multiplier uses the delay time of 99ns for performing multiplication operation where as in CSLA based multiplier also uses nearly the same delay time for multiplication operation. But the area desired for CLAA multiplier is reduced to 31% by the CSLA based multiplier to complete the multiplication operation.

Mishra, P., Aniruddha, A.K. and Nidhi, A. [2] presented the multiplier architectures proposed in literature seek to reduce power dissipation by reduction of effective switching activity or reduction in effective switched capacitance. They characterize a family of unsigned integer multiplier architectures by considering three multipliers for each and every of the 8, 16 and 32 bit word length cases and compare their performance against the Array and Wallace multiplier architectures used for similar class of applications. Authors observed in their experiments that the proposed multiplier can have a power advantage of up to 34.92% as compared to Array multiplier when subjected to image data, with the same area characteristic and a very marginal trade off in speed.

Qingzheng Li, Guixuan Liang and Bermak, [3] proposed a novel unified implementation of unsigned multiplication using a simple sign-control unit together with a line of multiplexers. The proposed method has been demonstrated through CMOS implementation of a 32-bit unsigned multiplier. Reported outcomes show the proposed unified unsigned implementation is very compact with only 0.45% silicon area overhead. The critical path delay of the proposed multiplier is 3.13 ns.

Gurumurthy, K.S. and Prahallad, M.S., [4] discussed the "Array of Array" multiplier which is a derivative of Braun Array Multiplier. The main benefit of "Array of Array" multipliers is its inherent ability to reduce both time and space complexity with intermediate relative performance. In this research work a  $16 \times 16$  unsigned 'Array of Array' multiplier circuit is designed with hierarchical structuring, it is optimized using Vedic Multiplication Algorithm "Urdhva Triyagbhyam" and Karatsuba-Ofman algorithm. The presented algorithm is useful for math coprocessors in the field of computers. The presented multiplier implementation shows large fall in average power dissipation and in time delay as compared to Booth encoded radix-4 multiplier.

Asati, A. and Chandrashekhar, [5] performed the implementation of multipliers are preferred for smaller operand sizes due to their simpler VLSI design. The proposed multiplier design shows large reduction in propagation delay and the average power consumption (at 20 MHz) as compared to 16-bit Booth encoded. The whole transistor count, maximum instantaneous power, core area, leakage power and total routing length have been estimated.

Perri, S., Staino, G. and Corsonello, P. [6] worked on a high-speed parallel multiplier based on 3-bit-scan without overlapping bits. The presented multiplier is capable to elaborate both signed and unsigned operands and it is

suitable for both full-custom and standard-cells based VLSI designs.

Shah, F.A., Jamal, H. and Khan, M.A. [7] worked on a low power programmable FIR filter based on partitioned multipliers. The Architecture chosen for design is conventional direct form. Which is Power efficient technique similar to unsigned multiplication and reduction of switching activity are used.

#### IV. PROBLEM FORMULATION

Regarding the circuit area complexity in the adder architectures, the (RCA) ripple-carry adder in the first class is the most efficient one, but the carry select adder in the fourth class with highest complexity is the least efficient one. Bearing in mind the circuit delay time, Carry Select Adder is the fastest one for every n-bit length, as a result the shortest delay. Otherwise, Ripple Carry Adder (RCA) is the slowest one, because of the long carry propagation.

As a term Area-Delay Product which gave us the clear picture of the space-time tradeoff. It is valuable to note that while we consider all the adders discussed above Ripple Carry adders and Carry Select Adders are the two sides of the spectrum. Since, while Ripple Carry Adders have a smaller area and lesser speed, in contrast to the Carry Select adders have high speed (nearly twice the speed Ripple Carry Adders) and occupy a larger area. But Carry Look Ahead Adder (CLA) has a proper balance between both the Area occupied and Time required. Therefore amongst the three, Carry Look Ahead Adder has the least AREA DELAY PRODUCT.

#### V. CONCLUSIONS

After all of this we can try to improve power efficiency of circuits. After reviewing all research papers we want to modify one of the research methods to reach proper and efficient results. After the comparison all we came to a conclusion that Carry Select Adders are best suited for situations where Speed is the only criteria. Equally Ripple Carry Adders are best suited for Low Power Applications. But amongst all Carry Look Ahead Adder had the least Area-Delay product that tells us, it is appropriate for situation where both low power and fastness is main criteria.

#### REFERENCES

- [1]. Vijayalakshmi, V.; Seshadri, R.; Ramakrishnan, S., "Design and implementation of 32 bit unsigned multiplier using CLAA and CSLA," *Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), 2013 International Conference on*, vol., no., pp.1,5, 7-9 Jan. 2013.
- [2]. Mishra, P.; Aniruddha, A.K.; Nidhi, A.; Kishore, J.K., "Low power unsigned integer multiplier for digital signal processors," *India Conference (INDICON), 2012 Annual IEEE*, vol., no., pp.059,064, 7-9 Dec. 2012.
- [3]. Qingzheng Li; Guixuan Liang; Bermak, A., "A High-speed 32-bit Signed/Unsigned Pipelined Multiplier," *Electronic Design, Test and Application, 2010. DELTA '10. Fifth IEEE International Symposium on*, vol., no., pp.207,211, 13-15 Jan. 2010.
- [4]. Gurumurthy, K.S.; Prahalad, M.S., "Fast and power efficient 16x16 Array of Array multiplier using Vedic Multiplication," *Microsystems Packaging Assembly and Circuits Technology Conference (IMPACT), 2010 5th International*, vol., no., pp.1,4, 20-22 Oct. 2010.
- [5]. Asati, A.; Chandrashekhar, "A high-speed, hierarchical 16x16 array of array multiplier design," *Multimedia, Signal Processing and Communication Technologies, 2009. IMPACT '09. International*, vol., no., pp.161,164, 14-16 March 2009.
- [6]. Perri, S.; Staino, G.; Corsonello, P., "Parallel Multipliers using 3-Bit-Scan without Overlapping Bits," *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*, vol., no., pp.1211,1214, 24-27 Nov. 2007.
- [7]. Shah, F.A.; Jamal, H.; Khan, M.A., "Reconfigurable Low Power FIR Filter based on Partitioned Multipliers," *Microelectronics, 2006. ICM '06. International Conference on*, vol., no., pp.87,90, 16-19 Dec. 2006.
- [8]. Guoping Wang; Shield, J., "The efficient implementation of an array multiplier," *Electro Information Technology, 2005 IEEE International Conference on*, vol., no., pp.5 pp.,5, 22-25 May 2005.
- [9]. Bandapati, S.K.; Smith, S.C.; Choi, M., "Design and characterization of convention self-timed multipliers," *Design & Test of Computers, IEEE*, vol.20, no.6, pp.26,36, Nov.-Dec. 2003.
- [10]. Na Tang; Jian-hui Jiang; Lin, K., "A high-performance 32-bit parallel multiplier using modified Booth's algorithm and sign-deduction algorithm," *ASIC, 2003. Proceedings. 5th International Conference on*, vol.2, no., pp.1281,1284 Vol.2, 21-24 Oct. 2003.
- [11]. P. Asadi and K. Navi, "A novel high-speed 54-54 bit multiplier", *Am. J Applied Sci.*, vol. 4 (9), pp. 666-672. 2007.
- [12]. W. Stallings, *Computer Organization and Architecture Designing for Performance*, 7th ed., Prentice Hall, Pearson Education International, USA, 2006, ISBN: 0-13-185644-8.
- [13]. I. F. Wakerly, *Digital Design-Principles and Practices*, 4th ed., Pearson Prentice Hall, USA, 2006. ISBN: 0131733494
- [14]. A. Sertbas and R.S. Ozbey, "A performance analysis of classified binary adder architectures and the VHDL simulations", *J Elect. Electron. Eng.*, Istanbul, Turkey, vol. 4, pp. 1025-1030, 2004.