Efficient Delay Architecture of 32-Bit Multiplier Using Recursive Adder

Eti Awasthi[1], Prof. Ashish Raghuwanshi[2]

Mtech. Scholar[1], Research Guide[2]

Department of Electronics and Communication Engineering, IES College, Bhopal

Abstract - Every digital circuit has some logic to perform which is basically an algorithm of frequent operations of addition, subtraction, division and multiplication to achieve a specific task. The repeated addition operation is nothing but multiplication. The speed of this repeated operation is highly depends on the structure of the adder and multiplier operation highly depends on the architecture of the adder. This paper shows the proposed optimum adder design and multiplier utilizing the adder architecture with better speed. The improved design is of 32-bit which has a delay of 38.739ns for performing multiplication operation which is 39% less than previous 32-bit multiplier design. The proposed recursive adder based multiplier design is implemented on Virtex 7 FPGA device.

Keywords - Recursive addition, 32-bit Multiplier, Delay, Area.

I. INTRODUCTION

As the performance of processors has increased, the demand for high speed arithmetic blocks has also increased. With clock frequencies approaching 1 GHz, arithmetic blocks must keep pace with the continued demand for more computational power. The purpose of this work is to present methods of implementing high speed binary multiplication. In general, both the algorithms used to perform multiplication, and the actual implementation procedures are addressed. The emphasis of this work is on minimizing the latency, with the goal being the implementation of the fastest multiplication blocks possible.

The design of high-speed and low-power VLSI architectures need efficient arithmetic processing units, which are optimized for the performance parameters, namely, speed and power consumption. Adders are the key components in general purpose microprocessors and digital signal processors. They also find use in many other functions such as subtraction, multiplication and division. As a result, it is very pertinent that its performance augers well for their speed performance. Furthermore, for the applications such as the RISC processor design, where single cycle execution of instructions is the key measure of performance of the circuits, use of an efficient adder circuit becomes necessary, to realize efficient system performance. Additionally, the area is an essential factor which is to be taken into account in the design of fast adders. Towards this end, high-speed, low power and area efficient addition and multiplication have always been a fundamental requirement of high-performance processors and systems. The major speed limitation of adders arises from the huge carry propagation delay encountered in the conventional adder circuits, such as ripple carry adder and carry save adder.

Low power and high-throughput circuitry design are playing the challenging role for VLSI designer. For real-time signal processing, a high speed and high throughput MAC unit is always a key to achieve a high performance digital signal processing system. A regular MAC unit consists of multipliers and accumulators that contain the sum of the previous consecutive products. The main motivation of this work is to investigate various multiplier and adder architectures which are suitable for implementing Low power, area efficient and high speed MAC unit.

Multipliers play an important role in today's digital signal processing and various other applications. With advances in technology, many researchers have tried and are trying to design multipliers which offer either of following high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier, thus making them suitable for various high speed, low power, and compact VLSI implementation. The common multiplication method is adding and shift algorithm. Multiplication is a mathematical operation at its simplest is an abbreviated process of adding an integer to itself, a specified number of times. A number (multiplicand) is added to itself a number of times as specified by another number (multiplier) to form result (product).Multiplication hardware often consume much time and area compared to other arithmetic operations. Digital signal processors use a multiplier/MAC unit as a basic building block and the algorithms they run are often multiply-intensive. Multiplication-based operations such as multiply and Accumulate (MAC) are currently implemented in many Digital Signal Processing (DSP) applications such as convolution, Fast Fourier Transform (FFT), filtering and in microprocessors in its arithmetic and logic unit.

II. CARRY SELECT ADDER

The carry-select adder can be simply define as a combination of two Ripple Carry Adders (RCA).Carry select adders are one of the other popular architectures which show improved performance over ripple carry adders. As in ripple carry adders they are popular for their regular layout structure. These adders basically consist of blocks where each block executes two additions. One assumes that the input carry is '1' and the other assumes that the input carry is '0'. The input carry signal '0' generates a block generate signal and the input carry signal '1' generates a block propagate signal which are used to produce the carry out signal for the subsequent block which selects the appropriate set of sum bits.

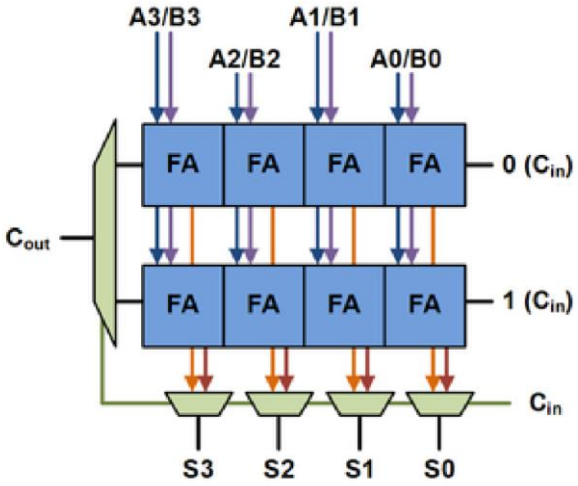

Figure 2.1 Carry Select Adder.

The carry select adder is category of conditional sum adder. Conditional sum adder works on some condition. Sum and carry are calculated by assuming input carry as 1 and 0 prior the input carry comes. When actual carry input arrives, the actual calculated values of sum and carry are selected using a multiplexer. The conventional carry select adder consists of k/2 bit adder for the lower half of the bits i.e. least significant bits and for the upper half i.e. Most Significant Bits (MSB's) two k/2 bit adders. In MSB adder one adder assumes carry input as one for performing addition and another assumes carry input as zero.

The carry out calculated from the last stage i.e. least significant bit stage is used to select the actual calculated values of output carry and sum. The selection is done by using a multiplexer. The basic concept of CSLA is shown in fig 2.1.

Addition is a fundamental operation for any digital system, digital signal processing or control system.

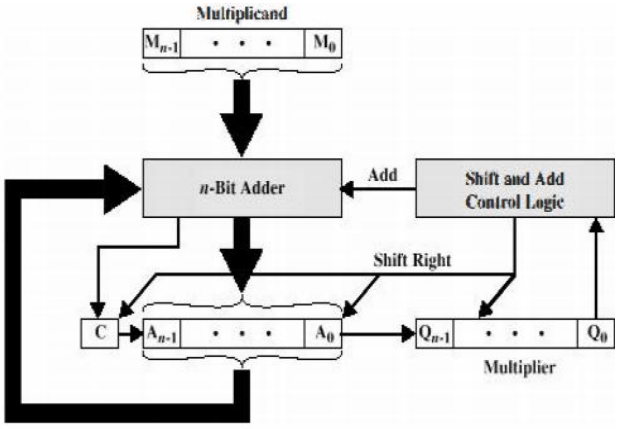

Figure 3.1 Multiplier of two n-bit values.

A fast and accurate operation of a digital system is greatly influenced by the performance of the resident adders. Adders are also very important component in digital systems because of their extensive use in other basic digital operations such as subtraction, multiplication and division. Hence, improving performance of the digital adder would greatly advance the execution of binary operations inside a circuit compromised of such blocks. The performance of a digital circuit block is gauged by analyzing its power dissipation, layout area and its operating speed.

III. PROPOSED ARCHITECTURE

Implementation and synthesis of proposed design has been completed on Xilinx ISE design suite. Figure 3.1 shows the schematic representation of proposed work. Recursive adder carry select adder CSLA architecture as shown in figure 3.1 has been implemented to design a delay efficient architecture of 32-Bit Multiplier.
RTL schematic of recursive adder design has been shown in figure 3.2. There are 8 blocks of 4 bit adders are used to design a 32 Recursive CSLA adder block. This adder can be used for the construction of add and shift multiplier which have lowest area, high speed and minimum power consumption.

As considering CSLA there is considerable area loss which can be avoided by using Recursive CSLA. The figure 3.3 describes the working of add and shift multiplier using the Recursive CSLA Adder.

The main logic of CSLA is to compute alternative results in parallel and subsequently selecting the correct result by using mux according to the control bit.

The extra time taken to compute the sum (because of propagation delay) is then avoided which results in good improvement in speed. The architecture of multiplier RTL schematic has been shown in Figure 3.3.

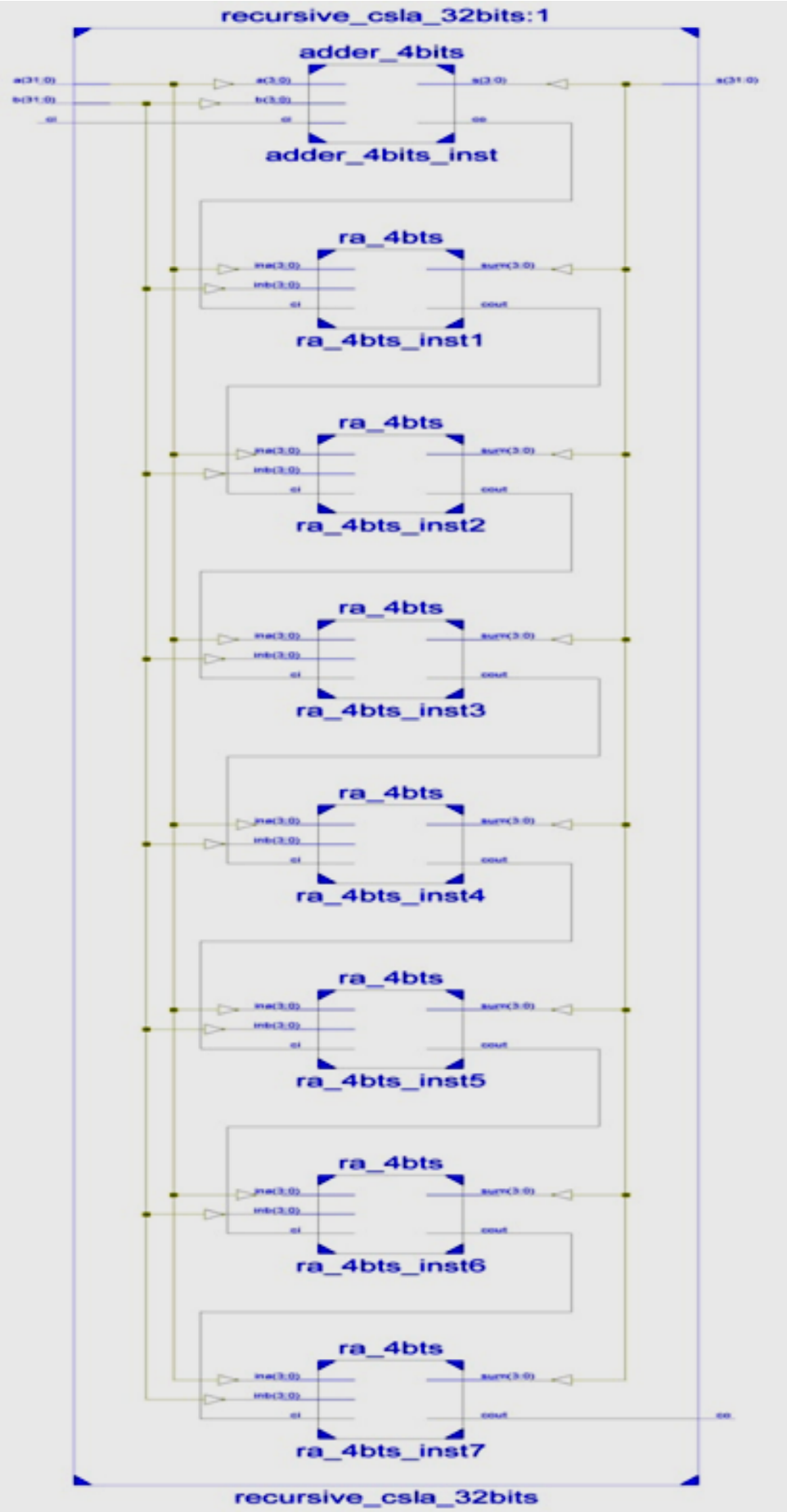

Figure 3.2 RTL Schematic of Recursive Adder Design.

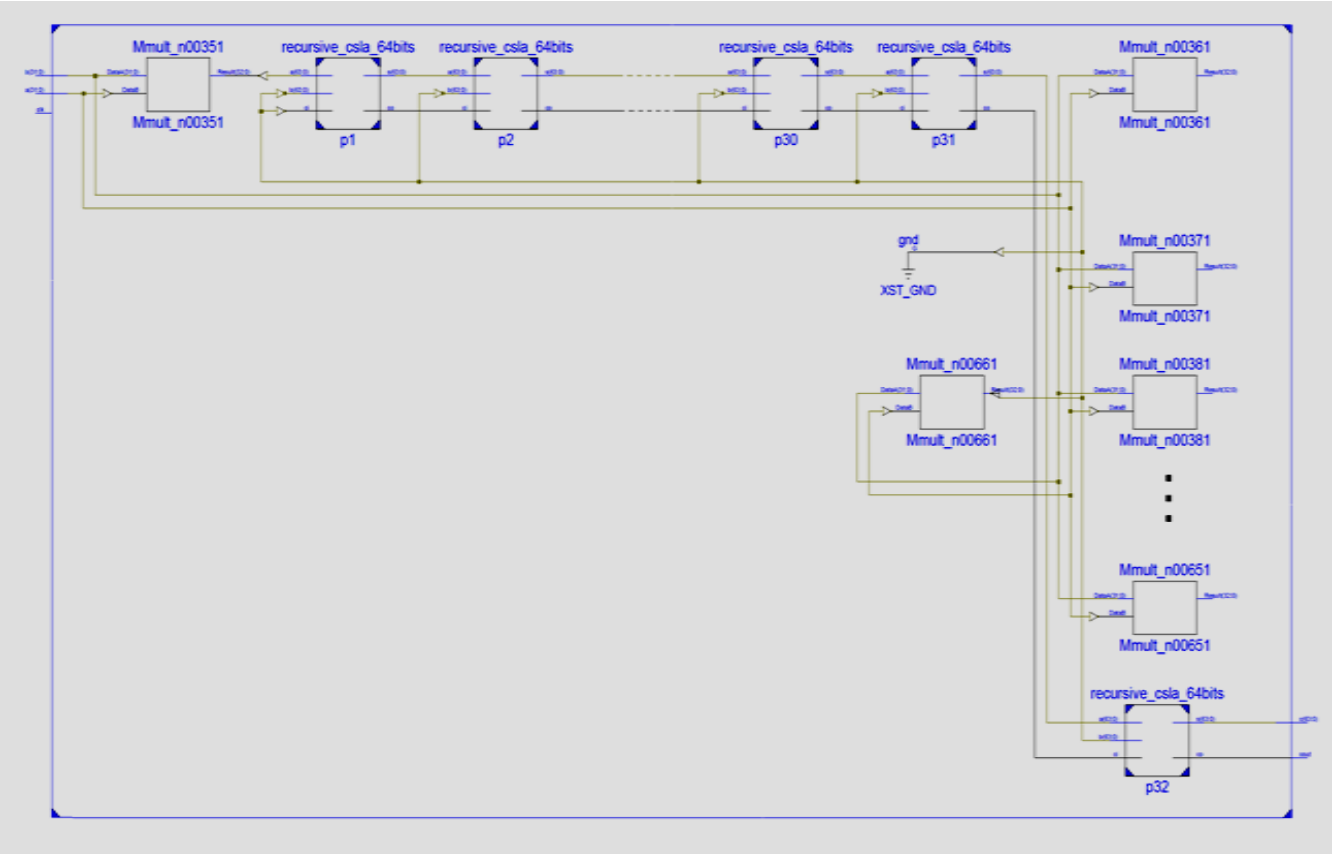

Figure 3.3 RTL Schematic of Proposed Multiplier Design.

IV. SYNTHESIS RESULTS

The synthesis of the proposed design is done on the XILINX 13.1 using Virtex 7 FPGA device. The synthesis outcomes as test bench waveforms are shown in the below figure. The proposed 32-Bit multiplier design is explained in the previous work is having lower delay profile as well as the area occupied. The details of the delay and area are given in this work. The device utilization summary is given in Table 1.

In Fig. 4.1 the test bench of the proposed multiplier architecture is shown where three different calculations are done.

In this, waveforms, as shown in figure 4.1 timing diagrams, the design summary and for CSLA based multipliers. The HDL code for both multipliers, CSLA, is generated.

The performance analysis of the area and delay is shown in the table 1 also the parameters are compared with existing work verifies good performance compared to existing work.

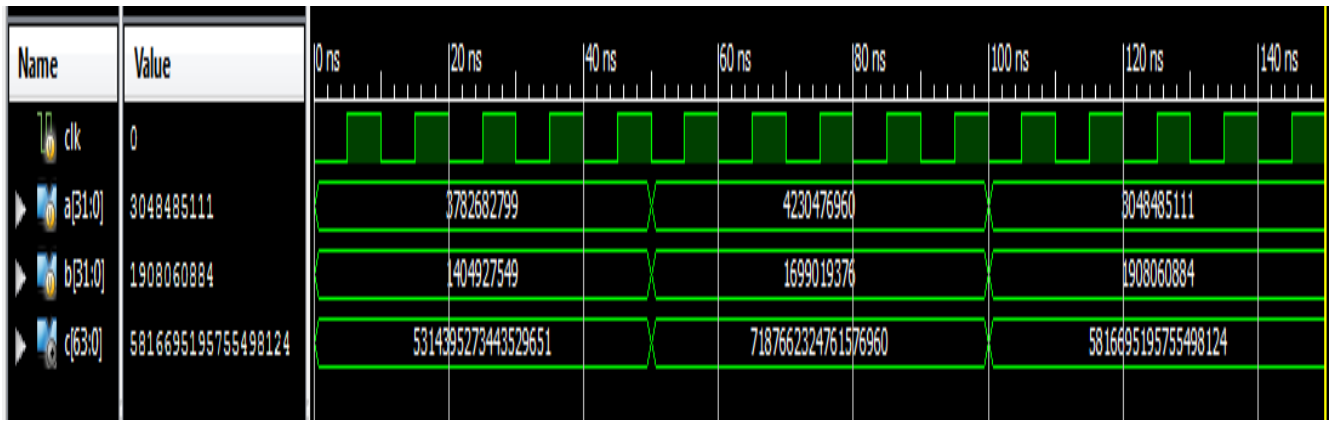

Figure 4.1 Test Bench Waveforms of the Proposed 32-Bit Multiplier

Table 1: Comparison of Delay and Area Utilization

	Delay	**Area**	**Delay Area Product**
Previous Work	99.50 ns	2039	202880.5
Proposed Work	38.74 ns	2769	107271.06

V. CONCLUSION AND FUTURE SCOPES

The proposed design is implemented using Verilog for 32-bit unsigned multiplier with recursive adder. Verilog was used to model and synthesis our multiplier. Using recursive logic improves the overall performance of the multiplier. Thus a 39% less delay and 52% area delay product reduction is possible with the use of the recursive adder based 32-bit unsigned multiplier than existing CSLA based 32 bit unsigned multiplier. The future extension could the use of same adder architecture to implement higher bit sized architectures for example64-bit or 128-Bit which will significantly improved in terms of area as well as delay.

REFERENCES

[1] Vijayalakshmi, V.; Seshadri, R.; Ramakrishnan, S., "Design and implementation of 32 bit unsigned multiplier using CLAA and CSLA," in 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] W. Stallings, Computer Organization and Architecture Designing for Peljormance, ed., Prentice Hall, Pearson Education International, USA, 2006, ISBN: 0-13-185644-8.

[3] L. F. Wakerly, Digital Design-Principles and Practices, 4th ed. , Pearson Prentice Hall, USA, 2006. ISBN: 0131733494

[4] P. Asadi and K. Navi, "A novel highs-speed 54-54 bit multiplier", Am. J Applied Sci., vol. 4 (9), pp. 666-672. 2007.

[5] P. S. Mohanty, "Design and Implementation of Faster and Low Power Multipliers", Bachelor Thesis. National Institute of Technology, Rourkela, 2009.

[6] S. Brown and Z. Vranesic, Fundamentals of Digital Logic with VHDL Design, 2nd ed. , McGraw-Hill Higher Education, USA, 2005. ISBN: 0072499389.

[7] J. R. Armstrong and F.G. Gray, VHDL Design Representation and Synthesis, 2nd ed. , Prentice Hall, USA, 2000. ISBN: 0-13-021670-4.

[8] Z. Navabi, VHDL Modular Design and Synthesis of Cores and Systems, 3rd ed. , McGraw-Hill Prof essional, USA, 2007. ISBN: 9780071508926.

[9] P. C. H. Meier, R. A. Rutenbar and L. R. Carley, "Exploring Multiplier Architecture and Layout for low Power", CIC'96, 1996.

[10] Software Simulation Package: Direct VHDL, Version 1.2, 2007, Green Mounting Computing Systems, Inc., Essex, VT, UK.

[11] Hasan Krad and Aws Yousi("Design and Implementation of a Fast Unsigned 32-bit Multiplier Using VHDL", 2010.

[12] 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.