Area Efficient 128-Bit FPGA Architecture of Multiplier using Verilog and Virtex 5 Device

Apoorva Singh[1] , Soumitra S. Pande[2] , Dr. D. Rai[3]

[1]M-Tech Research Scholar, [2]Head PG Studies, [3]Director, IMEC Sagar

IMEC Sagar Department of Electronics & Communication, Sagar

Abstract- The digital multiplier architecture is designed to make faster calculation of logical operations and the processing of information bits depends on the speed of calculations. The logical calculations depend on the algorithm or architecture of the logical circuit. The area utilizations of the directly affects the flow of calculations results on speed of output calculations. In this paper the proposed multiplier architecture is designed and implemented on the FPGA Virtex 5 Device and the programmed in Verilog. The outcomes of the proposed methodology clearly show the improvement in area than previous designs.

Keywords - Adder, Multiplier, Virtex 5, FPGA.

I. INTRODUCTION

Multiplication is one of the basic functions used in digital signal processing (DSP). It requires more hardware resources and processing time than addition and subtraction. In fact, 8.72% of all instructions in a typical processing unit is multiplier. In computers, a typical central processing unit devotes a considerable amount of processing time in implementing arithmetic operations, particularly multiplication operations. Most high performance digital signal processing systems rely on hardware multiplication to achieve high data throughput. Multiplication is an important fundamental arithmetic operation. 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 . Since multiplication dominate the execution time of most DSP algorithms, so there is a need of high speed multiplier. Currently, multiplication time is still that dominant factor in determining the instruction cycle time of a DSP chip. The multiplier is a fairly large block of a computing system.

The speed of the multipliers is greatly improved by properly deciding the number of pipeline stages and the positions for the pipeline registers to be inserted. Pipelines are widely used to improve the performance of digital circuits, since they provide a simple way of implementing parallelism from streams of sequential operations. In a pipelining system, the maximum operating frequency is limited by slowest stage which has the longest delay. A more stages are inserted in the pipeline, each stage becomes shorter, and ideally presents a smaller delay.

Serial Multipliers:

The simplest method to perform multiplication is to add series of partial products. The serial multipliers use a successive addition algorithm. They are simple in structure because both the operands are entered in a serial manner. Therefore, the physical circuit requires less hardware and a minimum amount of chip area. However, the speed performance of the serial multiplier is due to the operands entered sequentially.

Parallel Multipliers:

Most advanced digital systems incorporate a parallel multiplication unit to carry out high speed mathematical operations. A microprocessor requires multipliers in its arithmetic logic unit and a digital signal processing system requires multipliers to implement algorithms such as convolution and filtering. Some examples of parallel multipliers are array multipliers such as Braun multipliers, Booth multipliers and Baugh-Wooley multipliers, as well as the tree multipliers like Wallace multipliers. Array multipliers have a regular layout, although tree

multipliers are generally faster. Parallel multipliers [10] present high-speed performance, but are expensive in terms of silicon area and power consumption because in parallel multipliers both the operands are input to the multiplier in parallel manner.

Array Multipliers :

Array multiplier can be classified into following categories:

- Braun Multiplier
- Booth Multiplier
- Modified Booth Multiplier
- Baugh -Wooley Multiplier

Braun Array Multiplier:

Braun Array multiplier is well known due to its regular structure. It is a simple parallel multiplier that is commonly known as carry save array multiplier. This multiplier is restricted to performing multiplication of two unsigned numbers. It consist of array of AND gates and adders arranged in iterative structure that does not require logic registers.

This is also known as the non-additive multiplier since it does not add an additional operand to result of multiplication [10]. A four-bit Braun multiplier is shown in Figure 2.1. To perform N-bit by N-bit multiplication, the N-bit multiplicand A is multiplied by N-bit Multiplier B to produce product.

II. ADDER DESIGN

Multiplter For Unsigned Data:

Multiplication involves the generation of partial products, one for each digit in the multiplier, as in Figure3.These partial products are then summed to produce the final product. The multiplication of two n-bit binary integers results in a product of up to 2n bits in length [2]. We used the following algorithm to implement the multiplication operation for unsigned data The concept of CSLA is to compute alternative results in parallel and subsequently selecting the correct result with single or multiple stage hierarchical techniques. In CSLA both sum and carry bits are calculated for two alternatives Cin=O and 1.

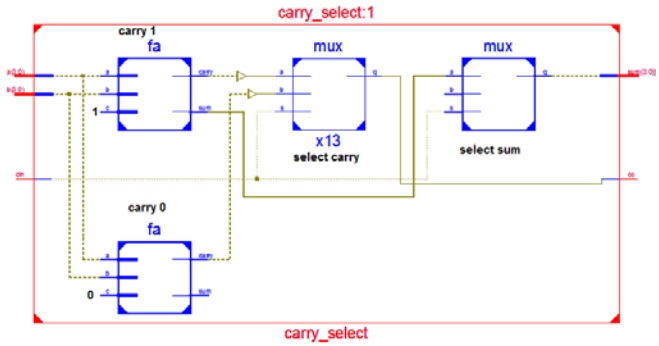

Fig. A Partial Schematic of the Multiplier

Once Cin is delivered, the correct computation is chosen using a mux to produce the desired output. Instead of waiting for Cin to calculate the sum, the sum is correctly output as soon as Cin gets there. The time taken to compute the sum is then avoided which results in good improvement in speed.

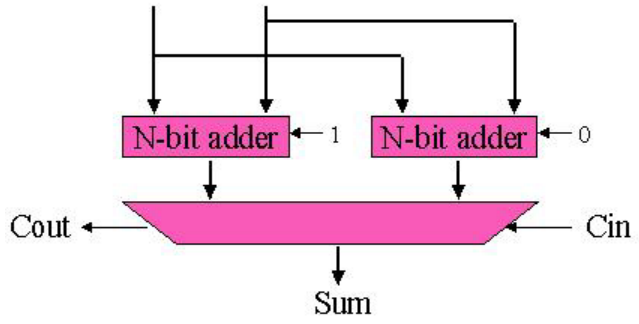

Fig.B Carry Select Adder

III. PROPOSED METHODOLOGY

The proposed methodology utilizes the FPGA Virtex to synthesis the proposed architecture designed in Verilog Language. The designed motivated from the 64-Bit architecture and the modules used are multiplexer, ripple carry adder and full adders. The architecture for the 128 -bit adder design is quite complex but still proposed design managed to achieve the better area utilization than previous designs.

128-Bit Algorithm:

Let the product register size be 128 bits. Let the multiplicand registers size be 128 bits. Store the multiplier in the least significant half of the product register. Clear the most significant half of the product register.

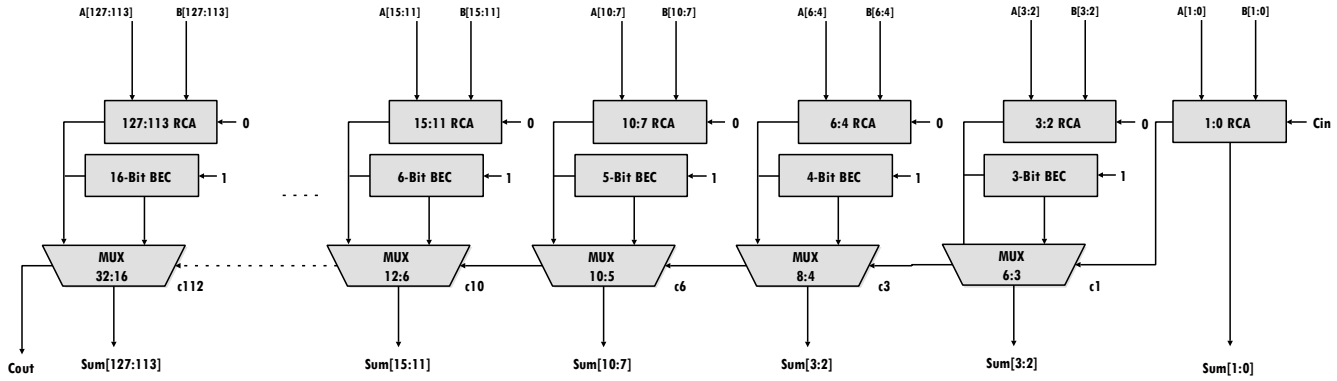

Fig.C 128-Bit Carry Select Adder

Steps of 128 Bit calculations:

a. If the least significant bit of the product register is "1" then add the multiplicand to the most significant half of the product register.
b. Shift the content of the product register one bit to the right (ignore the shifted-out bit.)
c. Shift-in the carry bit into the most significant bit of the product register. Figure 4. Shows a block diagram for such a multiplier [2].

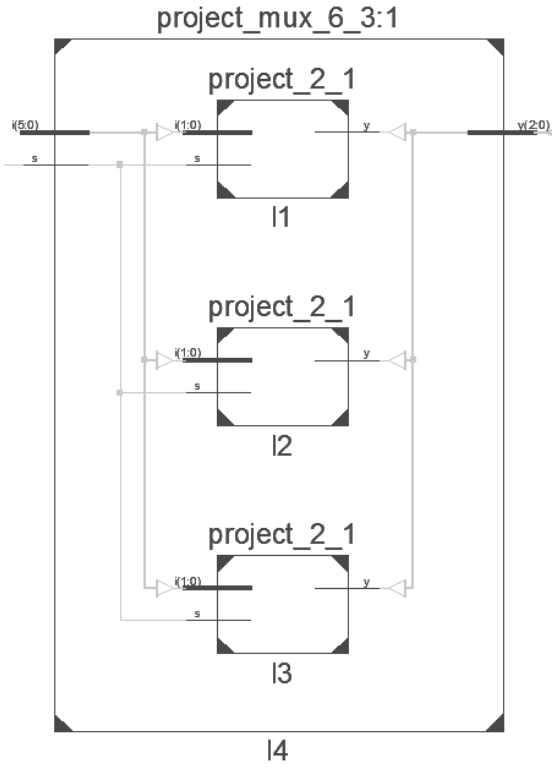

Fig.D Schematic of RTL 3x1 Multiplexer

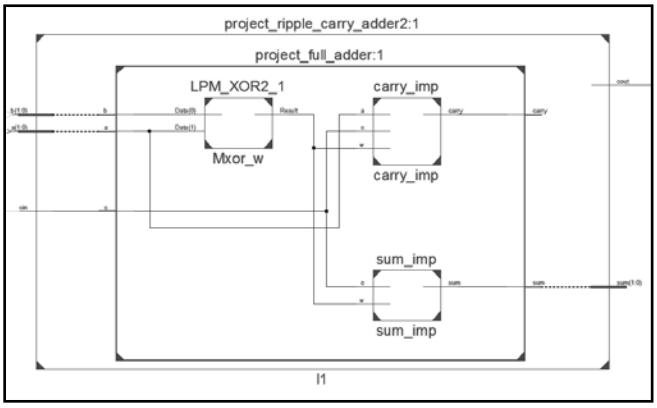

Fig.E Schematic of RTL Ripple Carry Adder

IV. SIMULATION OUTCOMES

The proposed 128-Bit BEC based multiplier architecture is shown in the previous section has better area count required. The calculation of the area count from Slice LUTs is shown here. Basically the Area Count is the number of Logic Cells utilization by the design on particular FPGA device. The implemented Advanced 128-Bit multiplier design is implemented on FPGA Vertex 5 board. The area count summary is given in Table I.

The Logic Cell counts are:

1- Logic Cell = 1.6 Slice LUTs

Table I: Area Count Summary Comparison

Area Count (Logic Cells)			
Logic Utilization	**Used**	**Available**	**Utilization**
Our Work	537	19968	2%
Previous Work	2696	3840	70%

Device Utilization Report is also shown in the below figure.

```
Device utilization summary:
---------------------------
Selected Device : 5vlx20tff323-2

Slice Logic Utilization:
 Number of Slice LUTs:                  336  out of 12480   2%
    Number used as Logic:               336  out of 12480   2%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    336
   Number with an unused Flip Flop:     336  out of   336 100%
   Number with an unused LUT:             0  out of   336   0%
   Number of fully used LUT-FF pairs:     0  out of   336   0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                         386
 Number of bonded IOBs:                 384  out of   172 223%
(*)
```

Fig.F Device Utilization Summary

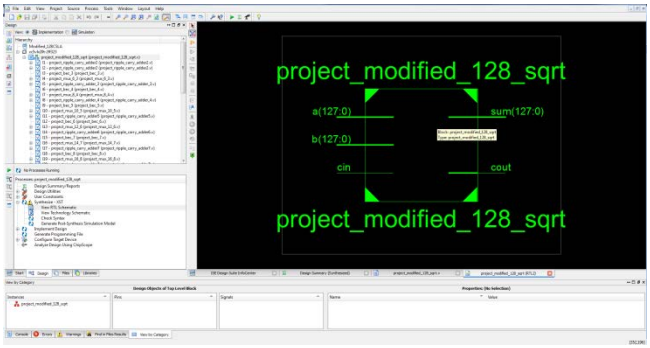

Fig.G RTL Schematic in XILINX User Interface 1

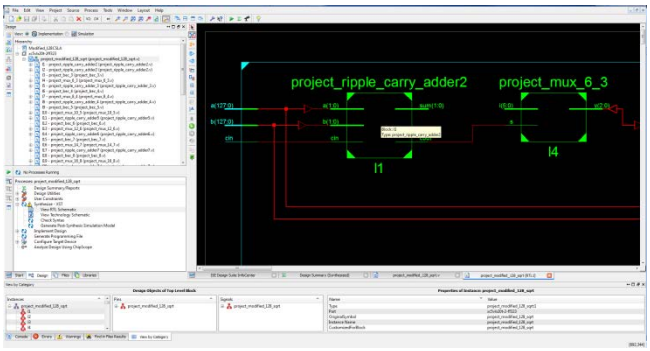

Fig.H RTL Schematic in XILINX User Interface 2

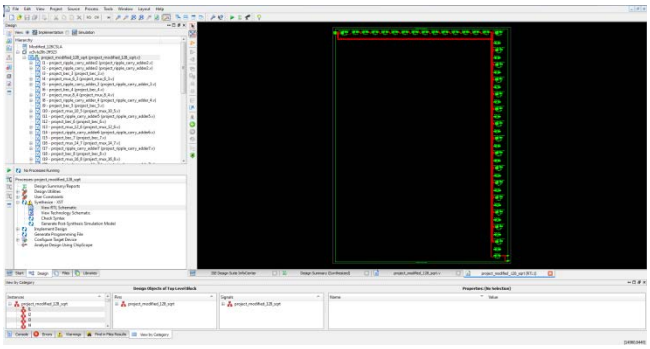

Fig.I RTL Schematic in XILINX User Interface 3

V. CONCLUSION AND FUTURE SCOPE

As shown in the previous sections the proposed multiplier design was good over previous design in terms of area count. In the proposed 128-Bit multiplier is based on the carry select adder with binary to excess converter and its architecture is both area and delay efficient. The synthesis report of the implemented architecture is also given in the previous section. The future extension could the use of different adder architecture which can be carry look ahead adder and the architecture for 128-Bit or higher bit processing module will also show significantly improvement e.g. 256-bit design in terms of area as well as delay.

REFERENCES

[1] Yi Xu; Shu-ming Chen, "Research of transformer based bufferless resonant clock network," *Solid-State and Integrated Circuit Technology (ICSICT), 2010 10th IEEE International Conference on* , vol., no., pp.369,371, 1-4 Nov. 2010

[2] 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.

[3] 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

[4] Grossschadl, J.; Kamendje, G.-A., "A single-cycle (32×32+32+64)-bit multiply/accumulate unit for digital signal processing and public-key cryptography," *Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on* , vol.2, no., pp.739,742 Vol.2, 14-17 Dec. 2003

[5] Schulte, M.J.; Marquette, L.; Krithivasan, S.; Walters, E.G.; Glossner, J., "Combined multiplication and sum-of-squares units," *Application-Specific Systems, Architectures, and*

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

[7] 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.

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

[9] W. Stallings, Computer Organization and Architecture Designing for Peljormance, 71h ed., Prentice Hall, Pearson Education International, USA, 2006, ISBN: 0-13-185644-8.

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

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

[12] Al-Khalili, A.J.; Aiping Hu, "Design of a 32-bit squarer - exploiting addition redundancy," *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on* , vol.5, no., pp.V-325,V-328 vol.5, 25-28 May 2003

[13] 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.

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

[15] Hasan Krad and Aws Yousi("Design and Implementation of a Fast Unsigned 32-bit Multiplier Using VHDL", 2000

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

[17] 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

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

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