International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Design and Analysis of Low Power High Speed Area Efficient Multipliers using Compressors on FPGA

Ch. Naga Srinivasa Rao[1], K. V. B. Chandra Sekhar Rao[2]

[1,2]Department of ECE, Swarnandhra College of Engineering & Technology., Narsapur, A.P, India

Abstract: ***The main theme of the paper is to design Compressor Based Low Power high speed and Area Efficient Multipliers on FPGA. In order to perform higher order multiplications more number of adders are required for the partial product addition. Special kind of adders are introduced which are capable of adding five/six/seven/eight/nine bits per decade with which we can reduce the number of adders and these special kind of adders are called as compressors. In order to develop higher order compressors, the combination of XOR gates and MUX circuits along with the binary counter property is contrasted with the conventional design. By using these compressors we can reduce the vertical critical paths. In this paper we present efficient implementation of multipliers with compressors on FPGA. When compared to carry propagate adders (CPA), high speed compressors provide fast critical path, independent of bit width with practically no area overhead. Design of such compressors will reduce the stage delays, transistor count and power delay product (PDP) and the results are verified in SPARTAN 3 FPGA.***

Keywords: FPGA, Multiplier, Carry propagate Adder, Compressor, counter

1. Introduction

The most important issues in designing a VLSI chip are power consumption, area and speed of operation. Multipliers play a key role in digital signal processing and various other applications. The architecture of the multiplier consists of three stages: 1. Generation of partial products 2. Reduction of partial products 3. Final addition.

During the stage of partial products addition, multipliers consume high amount of power and delay. More number of adders are required for higher order multiplications to perform partial product addition [1], [2]. The number of adders can be minimized by using different high order compressors.

In order to implement digital circuits the use of field programmable gate arrays (FPGAs) has been grown in recent years. Due to their reconfiguration capabilities these devices are allowed in parallel computing. For intensive arithmetic algorithms FPGAs achieve speedups of two orders of magnitude over general purpose processor [3]. Hence these are selected as target technology for many applications like digital signal processing [4], cryptography [5] hardware accelerations [6] etc.

The general architecture of an FPGA is a matrix of configurable logic elements (LEs) each logic element is surrounded by interconnection resources. Each configurable element consists of one or several n-input lookup tables (N-LUTs) and flip-flops. However In the modern FPGA architecture, specialized circuitry such as dedicated multipliers and RAM blocks are augmented with the array of LEs. The intensive use of these new elements will reduce the performance gap between ASIC & FPGA Implementations. Carry chain system is one of the resources used to implement the carry propagate adders (CPAs) which mainly consists of additional specialized logic to deal with carry signals and specific fast routing lines between consecutive LEs and is shown n figure 1.

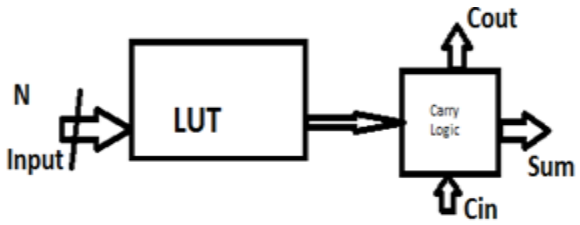

Figure 1: Modern FPGA device including dedicated carry chain resources.

Most of the modern FPGA devices include these resources from low cost one to high end families, compared to the implementation of general resources the proposed resource will accelerate the carry propagation by more than one order of magnitude. For the implementation of non arithmetic circuits [7], [8] many studies has been demonstrated the importance of using this resource in order to achieve designs with better performance and/or less area requirements. Different approaches for the optimization of redundant addition on FPGA are:

1) Proposing hardware modifications to the existing FPGA architectures [9]
2) Efficient mapping of isolated redundant adders in the inner structure of FPGAs [10], and
3) Based on the bit counters property Utilizing different heuristics to design compressor trees [11].

2. Various Compressor Architectures

A. 3-2 COMPRESSOR

3-2 compressor consists of three inputs X1, X2, X3 and generates two outputs sum and carries bits. Its block diagram is shown below.

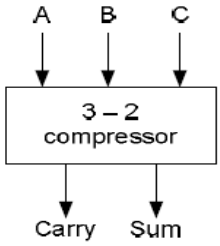

Figure 2: 3-2 compressor

This can be governed by the following equation

$$A + B + C = Sum + 2 * Carry \quad (1)$$

The architectures of traditional and enhanced version of 3-2 compressors along with sum and carry expressions are shown below.

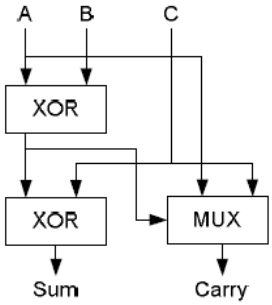

Figure 3: 3-2 Compressors Traditional architecture.

$$Sum = A \oplus B \oplus C \quad (2)$$

$$Carry = (A \oplus B) * C + \overline{(A \oplus B)} * A \quad (3)$$

B. 4-2 Compressor

The 4-2 compressor consists of 5 inputs A, B, C, D and Cin and will generate three outputs Sum, Carry, and Cout as shown in the figure 5.

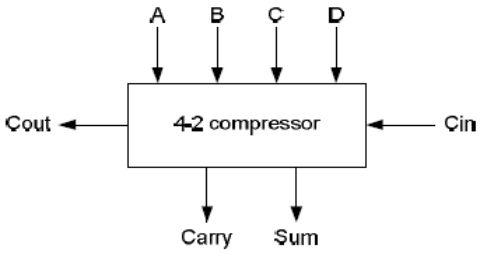

Figure 4: 4-2 compressor

In this the four inputs A, B, C, D and the output sum will have the same weight. The cin input is the output from a previous lower significant compressor and the output Cout is for the compressor in the next significant stage. The 4-2 compressor can be implemented with two full adders which are connected serially as shown in figure 6.

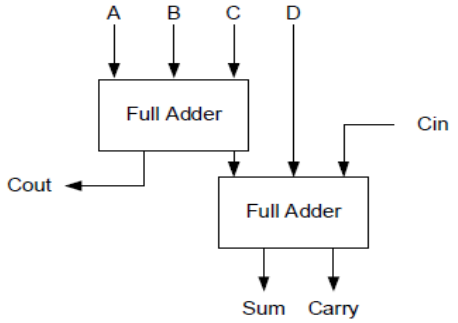

Figure 5: 4-2 Compressors with full adders

This can be governed by the following equation

$$A+B+C+D+Cin = Sum+2*(Carry +Cout) \quad (4)$$

The architectures of traditional and enhanced version of 4-2 compressors along with sum carry and Cout expressions are shown below.

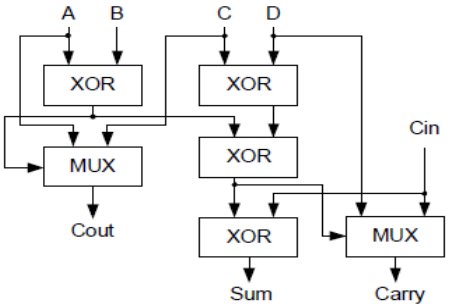

Figure 6: Traditional architecture of 4-2 compressor

$$Sum = A \oplus B \oplus C \oplus D \oplus Cin \quad (5)$$

$$Cout = (A \oplus B) * C + \overline{(A \oplus B)} * A \quad (6)$$

$$Carry = (A \oplus B \oplus C \oplus D) * Cin + \overline{(A \oplus B \oplus C \oplus D)} * D \quad (7)$$

3. Design of Multiplier Using Compressors

3.1 Array Multiplier

Multiplication is a mathematical operation; simply it is a process of adding an integer to itself in a specified number of times. The major steps involved in the multiplication are:

- Generation of partial product
- Reduction of partial product
- Final stage addition

In general case for the multiplication of an n-bit multiplicand with an m-bit multiplier, produce m number of partial products and the product is formed by n+m bits long. In order to perform N-bit/N-bit multiplication the N-bit multiplicand X is multiplied by N-bit multiplier Y to produce product Z. the unsigned binary numbers X and Y can be expressed as

$$X = \sum_{I=0}^{n} Xi2^{i} \quad (8)$$

$$Y = \sum_{j=0}^{n} Yj2^{j} \quad (9)$$

The product of X and Y can be represented with Z and it can be written in the following form

$$Z = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} Xi\ Yj\ 2^{i+j} \quad (10)$$

For the n*n multiplier, n(n-1) adders and n^2 AND gates are required. In the case of simple array each row of 3-2 compressor will adds a partial product to the partial sum which will generate a new partial sum and a sequence of carries. The delay of an array will depends on the depth of the array. Therefore the summing time of the simple array is N-2 of [3:2] compressors delay. Where N is the number of partial products.

3.2 8×8 Multiplier Design with 5:3 Compressors

The architecture of an 8×8 bit multiplier is shown in figure 9.

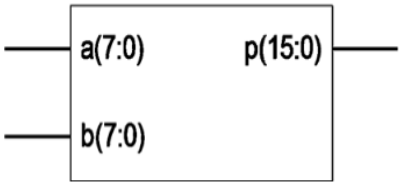

Figure 7: 8×8 bit multiplier

The Wallace tree architecture for an 8×8 bit multiplier is shown in fig 10. In this four stages are required to perform partial product. Different adders and compressors are used carefully to reduce the number of stages and to generate minimum number of outputs.

Now consider the column number four in figure 8, In which four bits are added at the first stage. These 4 bits could be added by using one full adder and it will produce two outputs. Instead of using this we proposed to use 4-2 compressor so that in the next stage we will have only 2 bits instead of 3 bits which in turns reduce the number of bits in the next stage. Now consider, column 5 in which five bits are added at the first stage. These 5 bits can be added with normal 3-2 compressors, it will produce 4 bits to the next stage of operation.

Instead of using normal compressors we propose traditional compressors which are designed with the combination of XOR gates and MUX circuits along with the binary counter property is contrasted with the conventional design. Using these high speed compressors we can reduce the vertical critical path which is independent of bit width, practically no area overhead when compared to carry propagate arrays (CPAs) on FPGAs.

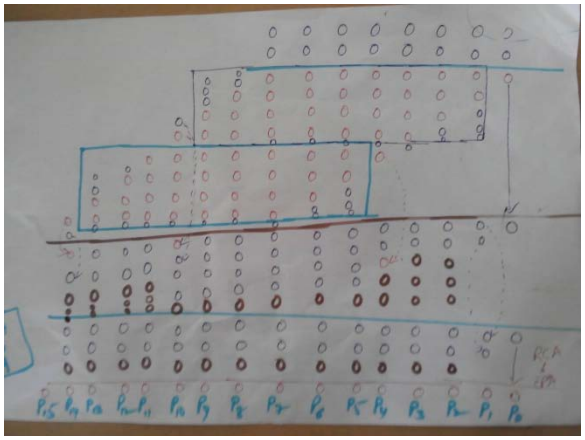

Figure 8: 8 bit multiplier with compressors

4. Performance Evaluation/ Results

The simulation results for an 8-bit multiplier in Xilinx ISE Simulator are shown in figure 9.

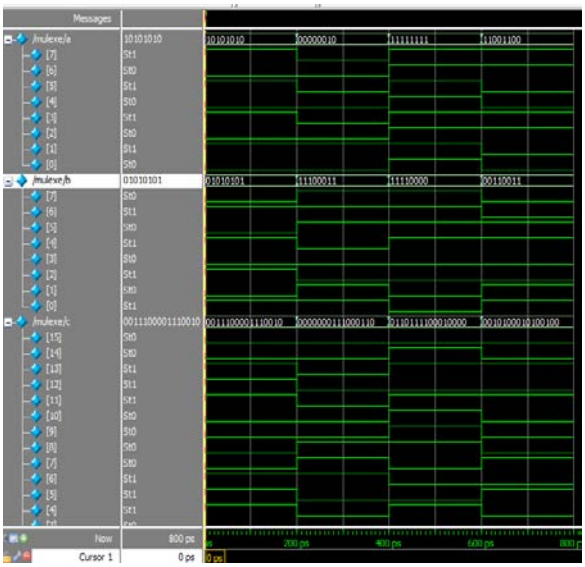

Figure 9: Simulated Waveforms

The RTL schematic for an 8-bit multiplier based on high speed compressors is shown in figure 10.

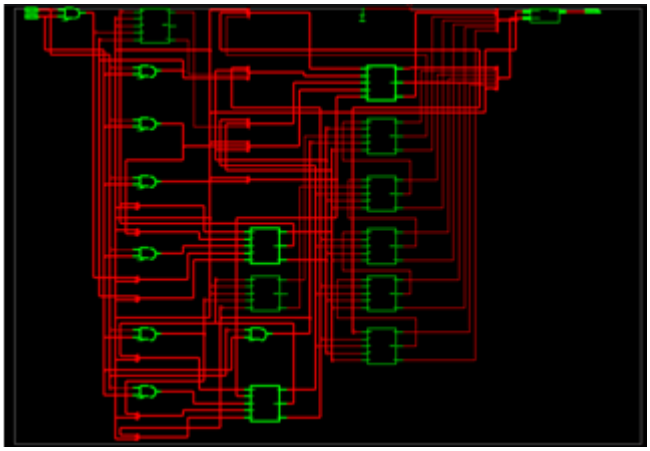

Figure 10: RTL Schematic

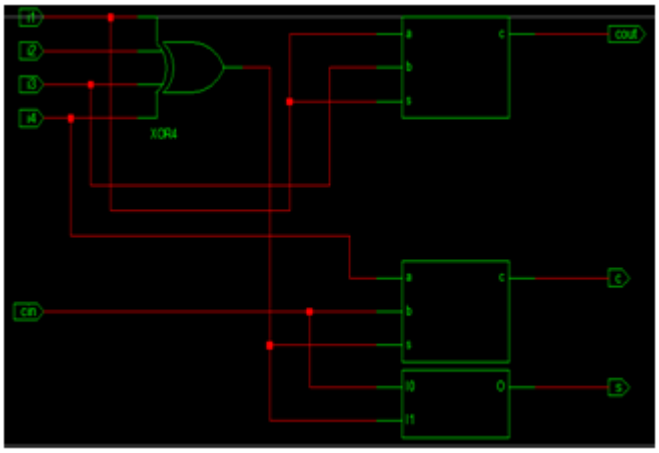

Figure 11: Multiplexer based Compressor Design

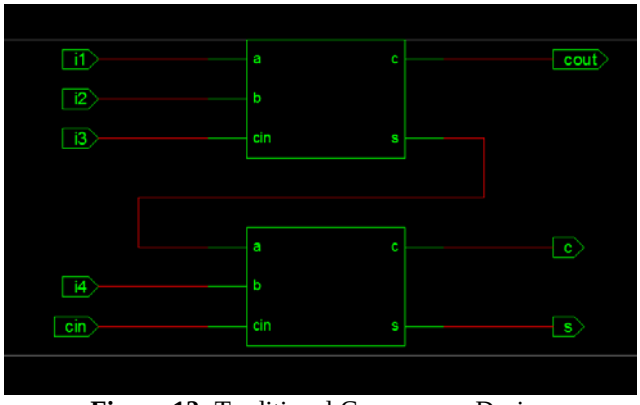

Figure 12: Traditional Compressor Design

As shown in the table 1 we summarize the different comparisons on power, delay, and power delay product (PDP), out of which the compressor based on Mux, produces low vertical Critical path delay results in reduction of power delay product.

Table 1: Power Comparison

S. No	Multiplier	Delay (ns)	Power (mw)	PDP (pws)
1	Array Multiplier	25.256	40	1010.2
2	Traditional Compressor	23.823	39	929.09
3	Mux based Compressor	21.63	40	865.48

As shown in the table 2 we summarize the comparisons on number of slices, and lookup tables (LUTs), out of which the compressor based on MUX, occupies less number of slices and LUTs which in turns reduces the area.

Table 2: Area Comparison

S. No	Multiplier	Slices used	Total slices	Used 4bit LUT's	Total LUT's
1	Array Multiplier	69	768	128	1536
2	Traditional Compressor	85	768	153	1536
3	MUX based Compressor	63	768	119	1536

5. Conclusion

In order to speed up the multiplication process different high speed compressors are used. Use of these high speed compressors in multipliers will simultaneously not only reduce the vertical critical path delay but also reduce the number of stage operations. By introducing the concept of high speed compressors on FPGA will reduce the no of slices, LUTs, PDP and increase the speed of operation by reducing the vertical critical path. Different comparisons are made in terms of Area, delay; Power and PDP are shown in the above Table1 and Table2

6. Future Scope

In order to perform higher order multiplications different compressors like 6-2, 7-2, and 9-2 compressors will help to reduce the area; power consumption and vertical critical path delay, and increases the speed of operation.

References

[1] S. F. Hsiao, M. R. Jiang, and J. S. Yeh, "Design of high speed low-power 3-2 counter and 4-2 compressor for fast multipliers," *Electron. Lett, vol. 34, no. 4, pp. 341–343*, 1998
[2] H. T. Bui, Y. Wang, and Y. Jiang, "Design and analysis of low-power 10-transistor full adders using novel XOR–XNOR gates," *IEEE Trans. Circuits Syst. II, vol. 49, pp. 25–30, Jan. 2002.*
[3] B. Cope, P. Cheung, W. Luk, and L. Howes, "Performance Comparison of Graphics Processors to Reconfigurable Logic: A Case Study," IEEE Trans. Computers, vol. 59, no. 4, pp. 433-448, Apr. 2010.
[4] S. Dikmese, A. Kavak, K. Kucuk, S. Sahin, A. Tangel, and H. Dincer, "Digital Signal Processor against Field Programmable Gate Array Implementations of Space-Code Correlator Beam former for Smart Antennas," IET Microwaves, Antennas Propagation, vol. 4, no. 5,pp. 593 599, May 2010.
[5] H. Lange and A. Koch, "Architectures and Execution Models for Hardware/Software Compilation and their System-Level Realization," IEEE Trans. Computers, vol. 59, no. 10, pp. 1363- 1377, Oct. 2010.
[6] S. Roy and P. Banerjee, "An Algorithm for Trading off Quantization Error with Hardware Resources for MATLAB-based FPGA Design," IEEE Trans. Computers, vol. 54, no. 7, pp. 886-896, July 2005.

[7] T. Guneysu, T. Kasper, M. Novotny, C. Paar, and A. Rupp, "Cryptanalysis with COPACOBANA," IEEE Trans. Computers, vol. 57, no. 11, pp. 1498-1513, Nov. 2008.
[8] I. Kuon and J. Rose, "Measuring the Gap between FPGAs and ASICs," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 2, pp. 203-215, Feb. 2007.
[9] G. Cardarilli, S. Pontarelli, M. Re, and A. Salsano, "On the Use of Signed Digit Arithmetic for the New 6-Inputs LUT Based FPGAs," Proc. IEEE 15th Int'l Conf. Electronics, Circuits and Systems (ICECS), pp. 602-605, 2008.
[10] M. Frederick and A. Somani, "Beyond the Arithmetic Constraint: Depth-Optimal Mapping of Logic Chains in LUT-Based FPGAs," Proc. ACM/SIGDA Int'l Symp. Field Programmable Gate Arrays, pp. 37-46, 2008.
[11] J.-L. Beuchat and J.-M. Muller, "Automatic Generation of Modular Multipliers for FPGA Applications," IEEE Trans. Computers, vol. 57, no. 12, pp. 1600-1613, Dec. 2008.

Author Profile

CH. Naga Srinivasa Rao, Pursuing M.Tech in Swarnandhra College of Engineering and Technology, Narsapur, AP. His area of interest is Digital signal processing (DSP).

Mr. K. V B Chandra Sekhar Rao, Assoc. Professor in the Department of Electronics and Communication Engineering, Swarnandhra College of Engineering and Technology, Narsapur , AP. His research areas are based on VLSI, and Embedded Systems.