ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
*A Peer Reviewed Open Access International Journal*

# Efficient Multi-Operand Adders in VLSI Technology

**K.Priyanka**
M.Tech-VLSI,
CMR Institute of Technology,
Hyderabad, T.S, India.

**D.Chandra Mohan**
Assistant Professor,
CMR Institute of Technology,
Hyderabad, T.S, India.

**Dr.S.Balaji, M.E, Ph.D**
Dean,
Department of ECE,
CMR Institute of Technology,
Hyderabad, T.S, India.

## Abstract:

This paper presents different approaches to the efficient implementation of generic carry-save compressor trees on FPGAs. They present a fast critical path, independent of bit width, with practically no area overhead compared to CPA trees. Along with the classic carry-save compressor tree, we present a novel linear array structure, which efficiently uses the fast carry-chain resources. This approach is defined in a HDL code based on CPAs, which makes it compatible with any FPGA (field programmable gate arrays) family or vendor. A detailed study is provided for a wide range of bit widths and large number of operands.

## Keywords:

Compressor, CPA, Linear array structure, Mullti-operand addition

## I.INTRODUCTION:

As the scale of integration keeps growing, more and more sophisticated signal processing systems are being implemented on a VLSI chip. These signal processing applications not only demand great computation capacity but also consume considerable amount of energy. While performance and Area remain to be the two major design tolls, power consumption has become a critical concern in today‘s VLSI system design[1]. The need for low-power VLSI system arises from two main forces. First, with the steady growth of operating frequency and processing capacity per chip, large currents have to be delivered and the heat due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these portable devices. Addition usually impacts widely the overall performance of digital systems and a crucial arithmetic function. In electronic applications adders are most widely used. Applications where these are used are multipliers, DSP to execute various algorithms like FFT, FIR and IIR. So, speed of operation is the most important constraint to be considered while designing multipliers. Due to device portability miniaturization of device should be high and power consumption should be low. Devices like Mobile, Laptops etc. require more battery backup. So, a VLSI designer has to optimize these three parameters in a design. These constraints are very difficult to achieve so depending on demand or application some compromise between constraints has to be made. Ripple carry adders exhibits the most compact design but the slowest in speed. Whereas carry look ahead is the fastest one but consumes more area. Carry select adders act as a compromise between the two adders. In 2002, a new concept of hybrid adders is presented to speed up addition process by Wang et al. that gives hybrid carry look-ahead/carry select adders design. In 2008, low power multipliers based on new hybrid full adders is presented.

## II.NEED FOR LOW POWER DESIGN:

The design of portable devices requires consideration for peak power consumption to ensure reliability and proper operation. However, the time averaged power is often more critical as it is linearly related to the battery life. There are four sources of power dissipation in digital CMOS circuits: switching power, short-circuit power, leakage power and static power.Low Voltage Power consumption is linearly proportional to voltage swing (Vs) and supply voltage (Vdd) as indicated in Eq. (2.5). For most CMOS logic families, the swing is typically rail-to-rail. Hence, power consumption is also said to be proportional to the square of the supply voltage, Vdd. Therefore, lowering the Vdd is an efficient approach to reduce both energy and power, presuming that the signal voltage swing can be freely chosen. This is, however, at the expense of the delay of circuits. The delay, td, can be shown to be proportional to. The exponent is between 1 and 2. It tends to be closer to 1 for MOS transistors that are in deep sub-micrometer region, where carrier velocity saturation may occur.

The current technology trends are to reduce feature size and lower supply voltage. Lowering Vdd leads to increased circuit delays and therefore lower functional throughput. Smaller feature size, however, reduces gate delay, as it is inversely proportional to the square of the effective channel length of the devices. In addition, thinner gate oxides impose voltage limitation for reliability reasons. Hence, the supply voltage must be lowered for smaller geometries. The net effect is that circuit performance improves as CMOS technologies scale down, despite of the Vdd reduction. Therefore, the new technology has made it possible to fulfill the contradicting requirements of low power and high throughput.

## III.EXISTING:

In this paper, we study the efficient implementation of multi-operand redundant compressor trees in modern FPGAs by using their fast carry resources. Deign Implementation the classic design of a multi-operand CS compressor tree attempts to reduce the number of levels in its structure. The 3:2 counters or the 4:2 compressors are the most widely known building blocks to implement it. We select a 4:2 compressor as the basic building block, because it could be efficiently implemented on Xilinx FPGAs. The implementation of a generic CS compressor tree requires [Nop/2]-1, 4:2 compressors (because each one eliminates two signals), whereas a carry-propagate tree uses. To optimize the use of the carry resources, we propose a compressor tree structure similar to the classic linear array of CSAs. ]. However, in our case, given the two output words of each adder (sum-word and carry-word), only the carry-word is connected from each CSA to the next, whereas the sum words are connected to lower levels of the array.First, the two regular inputs on each CSA are used to add all the input operands (Ii). When all the input operands have been introduced in the array, the partial sum-words (Si) previously generated are then added in order (i.e., the first generated partial sums are added first).

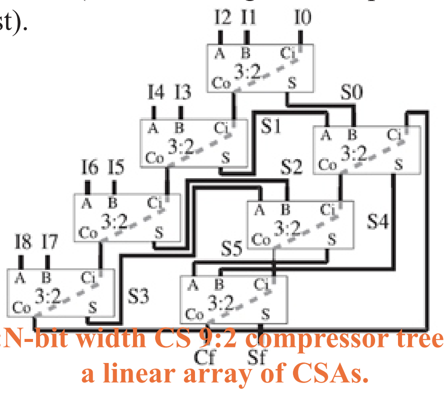


Figure 1:N-bit width CS 9:2 compressor tree based on a linear array of CSAs.

## IV.PROPOSED:

In this section, we present different approaches to efficiently map CS compressor trees on FPGA devices. In addition, approximate area and delay analysis are conducted for the general case. Let us consider a generic compressor tree of Nop input operands with N bit width each. We also assume the same bit width for input and output operands. Thus, input operands should have previously been zero or sign extended to guarantee that no overflow occurs. A detailed analysis of the number of leading guard bits required for multi operand CS addition is provided.

## Applications:

- Digital signal processors
- Microprocessors
- Controllers

## Objective:

The area overhead of the redundant adders when they are implemented on FPGAs Present a fast critical path, independent of bit width, with practically no area overhead compared to CPA trees.In this 18:2 compressor is used so that the main theme is even though we are increasing the multiple bits or whether the architecture is increasing or not the delay will gets reduced and increases efficiency. The main point is even though architecture increases the compressors which re of different types used inside the architecture is very efficient for total design .so, that's why even though architecture was increasing delay is reducing and increasing the performance.

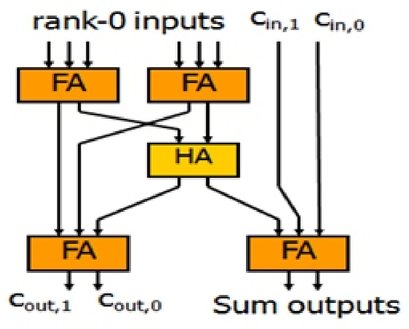
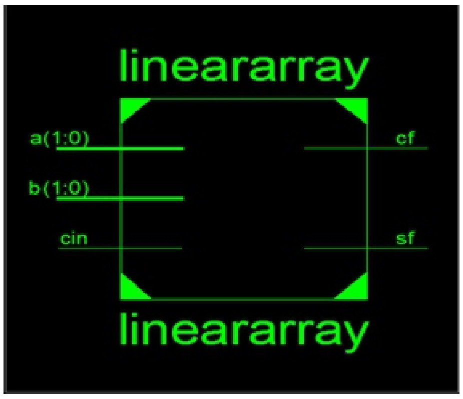


Figure 2: 6:2 Compressor based Adder

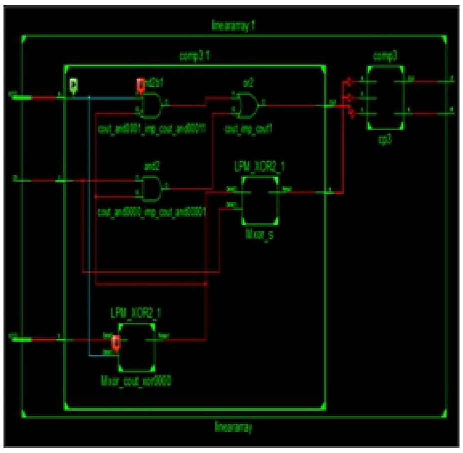
ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
*A Peer Reviewed Open Access International Journal*

## V.IMPLEMENTATION RESULTS & COMPARISION:

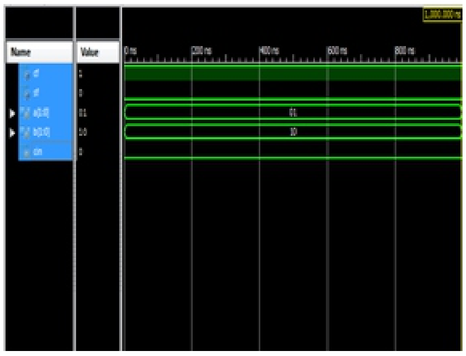
To measure the effectiveness of the designs presented in this paper, we have developed two generic VERILOG modules implementing the proposed compressor tree structures: First, the linear array implemented by using CPAs (binary and ternary) and, second, the 4:2 compressor tree using the design of the compressor presented in [28]. Both modules provide the output result in CS format and allow the selection of different parameters such as: The number of operands (Nop), the number of bits per operand (N), and the basic building blocks (i.e., binary or ternary adder) for the linear array. For the purposes of comparison, similar modules, which implement classic adder tree structures based on binary CPAs and ternary CPAs, have also been developed. All these modules were simulated using ISIMULATOR and they were synthesized using Xilinx ISE 14.2, targeting Spartan-3A, Virtex-4, and Virtex-5 devices.



a)

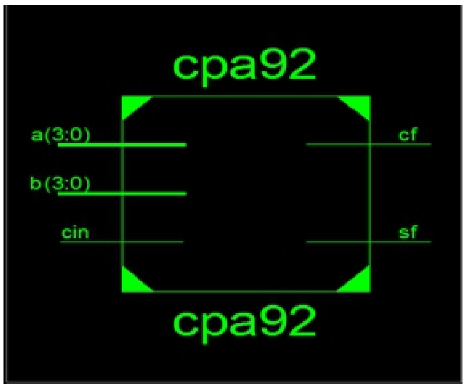


b)

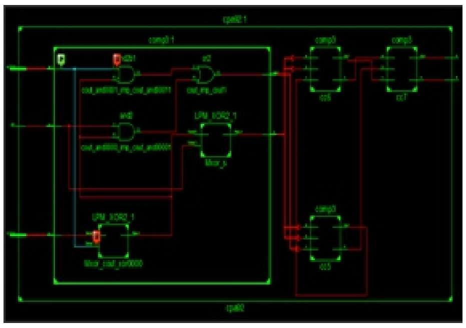


c)

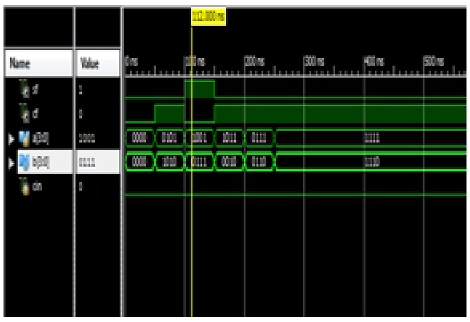
Figure 3: Existing system for Linear array a) Block Diagram, b) RTL Schematic, c) Waveform



a)

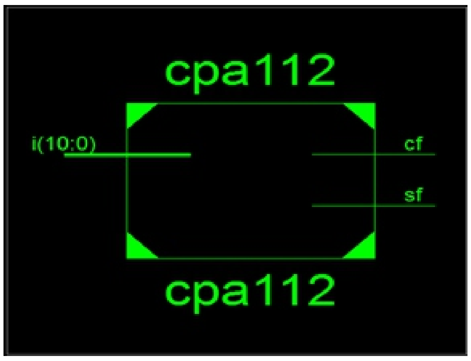


b)

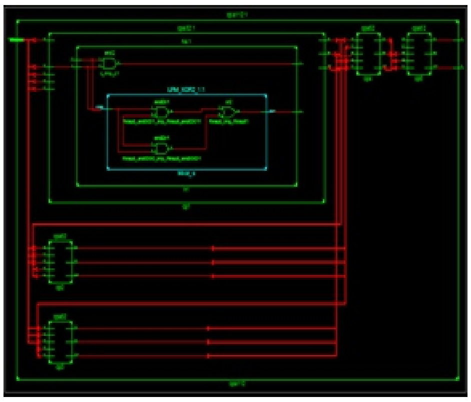


c)

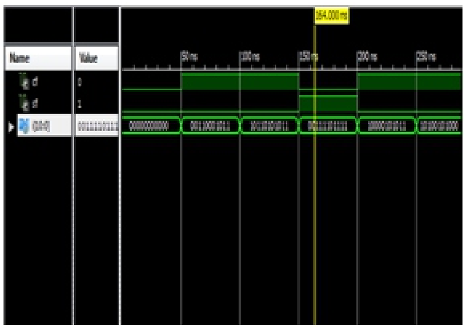
Figure 4: Existing system for Linear CPA a) Block Diagram, b) RTL Schematic c) Waveform



a)

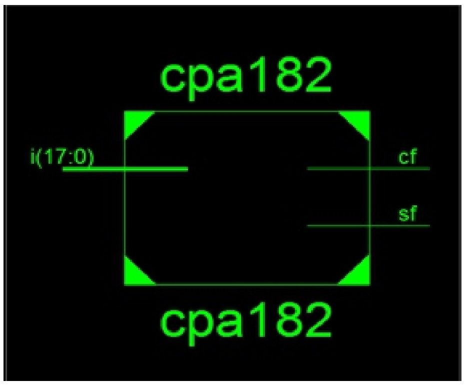


b)

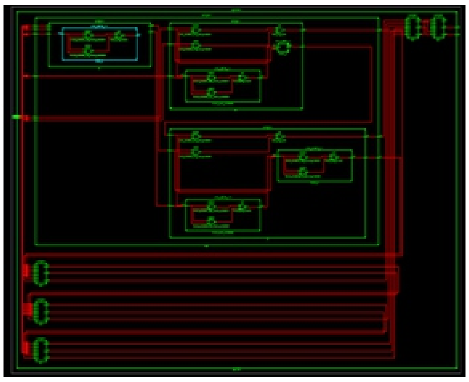


c)

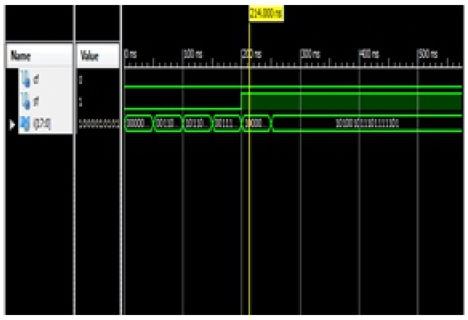
Figure 5: Existing System using 11:2 Compressor a) Block Diagram, b) RTL Schematic c) Waveform.



a)



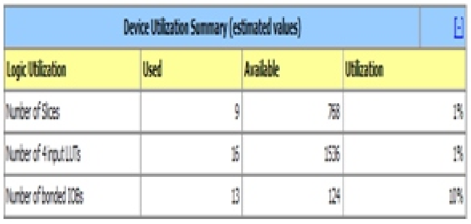
b)



c)

Figure-6: Proposed Method using 18:2 a) Block Diagram, b) RTL Schematic, c) Waveform.

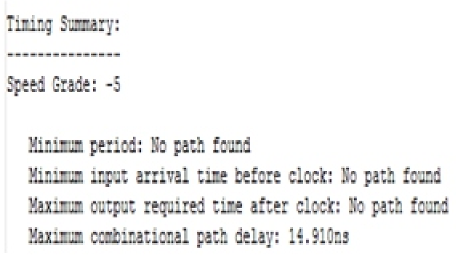
## Area & Delay Reports:



Device Utilization Summary (estimated values)

| Logic Utilization | Used | Available | Utilization |
|---|---|---|---|
| Number of Slices | 9 | 768 | 1% |
| Number of 4 input LUTs | 16 | 1536 | 1% |
| Number of bonded IOBs | 13 | 124 | 10% |

a)

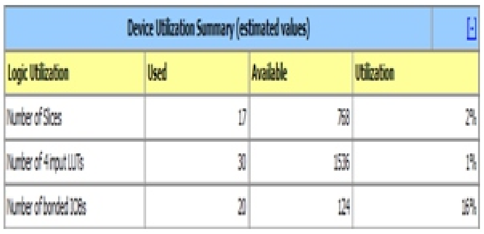


```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 14.910ns
```

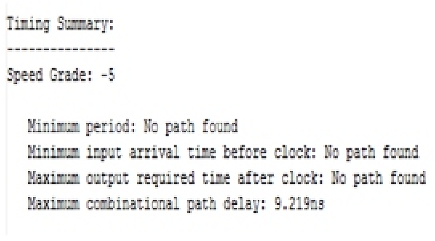
b)

Figure 6: Existing Method using 11:2 a) Area, b) Delay



| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| Logic Utilization | Used | Available | Utilization |
| Number of Slices | 17 | 768 | 2% |
| Number of 4 input LUTs | 30 | 1536 | 1% |
| Number of bonded IOBs | 20 | 124 | 16% |

a)



```
Timing Summary:
---------------
Speed Grade: -5

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 9.219ns
```

b)

**Figure7: Proposed Method using 18:2 a) Area, b) Delay**

## VI.CONCLUSION:

Efficiently implementing CS compressor trees on FPGA, in terms of area and speed, is made possible by using the specialized carry-chains of these devices in a novel way. Similar to what happens when using ASIC technology, the proposed CS linear array compressor trees lead to marked improvements in speed compared to CPA approaches and, in general, with no additional hardware cost. Furthermore, the proposed high-level definition of CSA arrays based on CPAs facilitates ease-of-use and portability, even in relation to future FPGA architectures, because CPAs will probably remain a key element in the next generations of FPGA. We have compared our architectures, implemented on different FPGA families, to several designs and have provided a qualitative and quantitative study of the benefits of our proposals.

## REFERENCES:

[1] B. Cope, P. Cheung, W. Luk, and L. Howes, "PerformanceComparison of Graphics Processors to Reconfigurable Logic: A Case Study," IEEE Trans. Computers, vol. 59, no. 4, pp. 433-448, Apr. 2010.

[2] 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 Beamformer for Smart Antennas," IET Microwaves, Antennas Propagation, vol. 4, no. 5,pp. 593-599, May 2010.

[3] 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.

[4] F. Schneider, A. Agarwal, Y.M. Yoo, T. Fukuoka, and Y. Kim, "A Fully Programmable Computing Architecture for Medical Ultrasound Machines," IEEE Trans. Information Technology in Biomedicine, vol. 14, no. 2, pp. 538-540, Mar. 2010.

[5] J. Hill, "The Soft-Core Discrete-Time Signal Processor Peripheral [Applications Corner]," IEEE Signal Processing Magazine, vol. 26, no. 2, pp. 112-115, Mar. 2009.

[6] J.S. Kim, L. Deng, P. Mangalagiri, K. Irick, K. Sobti, M. Kandemir, V. Narayanan, C. Chakrabarti, N. Pitsianis, and X. Sun, "An Automated Framework for Accelerating Numerical Algorithms on Reconfigurable Platforms Using Algorithmic/Architectural Optimization," IEEE Trans. Computers, vol. 58, no. 12, pp. 1654- 1667, Dec. 2009.

[7] 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.