INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

MULTIOPERAND REDUNDANT HIGHER ORDER COMPRESSOR TREES ON FPGAS

G. SreeReddy[1], T. Kishore[2], P. Mahendra [3]

[1]*PG Student, VLSI, Sree Vidyanikethan Engineering College, Tirupati, Chittoor, A.P, India.*

[2]*Assistant Professor, ECE Dept., Ravindra Engineering College for women, Kurnool, A.P, India.*

[3]*PG Student, VLSI, Sree Vidyanikethan Engineering College, Tirupati, Chittoor, A.P. India.*

Abstract: - Multi-operand adders, which found in parallel multipliers, they consist of the compression trees i.e., 9:2, 11:2, 15:2 and 32:2. In this work redundant adders were used to design parallel multi operand adders for ASIC implementations. The use of redundant adders on Field Programmable Gate Arrays (FPGAs) has been avoided because implementation of Carry Propagate Adders (CPAs) and the area overhead of the redundant adders is very high; delay is more because 'n' numbers of adders are connected. For reducing area-overhead and improve performance purpose we present implementation of carry-save compressor trees on FPGAs instead of redundant adders. The proposed method presents a fast critical path, independent of bit width, with no area overhead compared to CPA trees. This approach defined in a parameterizable HDL code based on CPAs, which compatible with FPGA family or vendor. This paper Analyses the timing performance of carry-save compressor trees in Xilinx virtex-5 family.

Keywords: Field programmable gate arrays, Computer arithmetic, reconfigurable hardware, multi operand addition, redundant representation, carry-save adders.

1. Introduction

Multi-operand addition found in partial product reduction of multipliers, or some combinations of addition and multiplication are frequently used arithmetic operation.The field Programmable gate arrays (FPGAs) are used to implement digital circuits and they have reconfiguration capabilities. FPGAs can be used to implement logical function. To achieve high speed FPGAs of two orders of magnitude are used over a general- purpose processor for arithmetic algorithms [1]. Thus, these devices are selected as the target technology for many applications, such as digital signal processing [2], hardware accelerators [3] cryptography [4].

The structure of an FPGA device is a matrix of configurable logic elements (LEs), each configurable element is consist of one or several n-input lookup tables (N- LUT) and flip-flops. The carry-chain system, which is used to improve the implementation of carry propagate adders (CPAs). It consists of specialized logic to the carry signals, and with help of this logic fast routing done between consecutive LEs, as shown in Fig. 1. So by using fast carry resources in carry chain system we achieve designs with better performance and/or less area requirements as compared to CPA implementation, and for implementing non arithmetic circuits [5], [6]. Multioperand addition appears in many algorithms, such as multiplication, filters and others. For the efficient

implementations, redundant adders are used to reduce the addition time by limiting the length of the carry-propagation chains propagation chains.

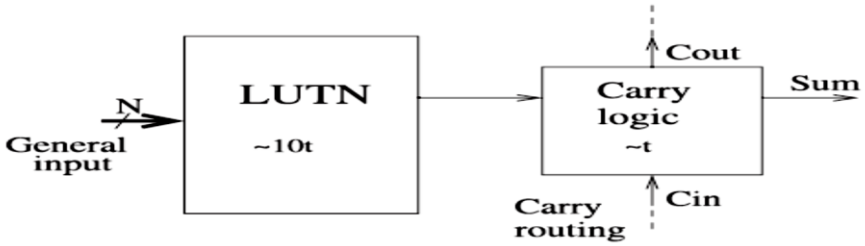


Figure.1. General Scheme of dedicated carry-chain resources included in modern FPGA devices.

In this work we examine the efficient implementation of carry save compressor trees on FPGAs. Section 2 describes previous work on redundant addition on FPGAs. In Section 3 describes implementation of multioperand redundant compressor trees on FPGAs such as 9:2, 11:2, 15:2, 32:2 compressor techniques and analysis of their performance. In Section 4, we compare the results of implementation using different approaches. Finally, the conclusions are presented in Section 5.

2. Related work

The redundant addition on FPGAs optimization can be done by using different approaches:

1. The efficient mapping of isolated redundant adders on an inner structure of FPGAs;

2. Utilizing different heuristics to design compressor trees based on bit counters;

3. Proposing hardware modifications to existing FPGA architectures

4. Specific applications.

In this work based on generalized parallel counters (GPCs) we present compressor tree designs. First, several GPC sizes are selected and they use the inner resources of the target FPGA. Second, to build the specific compressor tree based on a GPCs each proposes a different algorithm, in such a way that an attempt is made to minimize the critical path and/or the area of the tree.

3. CS Compressor trees on FPGAs

In this, to map CS compressor trees on FPGA devices we present different approaches. In this, for general case area and delay analysis are considered. Let us consider a generic compressor tree of Nop input operands with N bit width each. A number of leading guard bits are used for multioperand CS addition.

3.1 9:2 CS Compressor tree

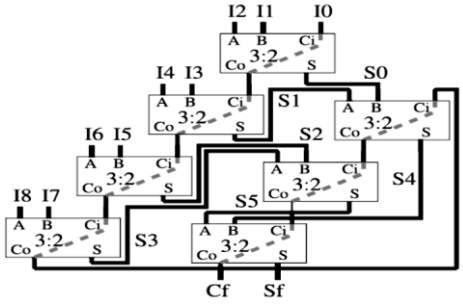


Figure. 2. N-bit width CS 9:2 compressor tree based on a linear array of CSAs.

The 3:2 counter or the 4:2 compressor are the most building blocks to implement 9:2 compressor.3:2 counter is called as full adder and 4:2 compressor built by 2 full adders. We choose 4:2 compressor as building block because it can be implemented on Xilinx FPGAs. The speed of a compressor tree is determined by the number of levels required. In this case, the critical path delay (D) is given by

$$L4{:}2 = [\log 2\ (Nop)]-1$$

$$D = L4{:}2.\ d4{:}2$$

whereL4:2 is the number of levels of the compressor tree and d4:2 is the delay of a 4:2 compressor level.

Fig. 2 shows a 9:2 compressor tree designed by using the proposed linear structure and time based model fig.3 are shown here.9:2 compressor tree designed by using 3:2 compressors here seven 3:2 compressor adders are used. In this compressor tree take 9 inputs and by adding 9 inputs with help of full adder operation we can estimate sum and carry. In this CSA, dashed line between the carry input and output represents the fast carry resources. C_i is used to introduce an input operand, on each CSA C_i is connected to the carry output (Co) of the previous CSA, as shown in Fig. 2. Thus, the whole carry-chain is preserved in 9:2 compressors from the input to the output of the compressor tree (from I_0 to C_f).

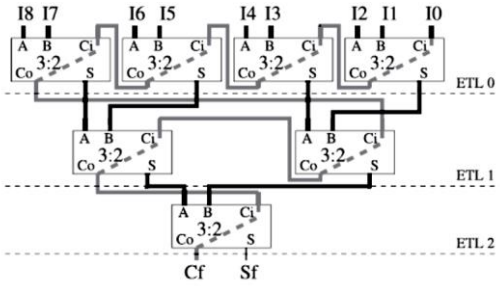


Figure.3. Time model of the proposed CS 9:2 compressor tree.

3.2 11:2 CS Compressor tree

The newest FPGA families Virtex-5/7 or Stratix-II/V, these are used to improve performance of multioperand addition. The number of levels in 11:2 compressors is expressed as

$$L3{:}1 = [\log 3\ (Nop)].$$

This is faster than the binary adders. So, the ternary adder is preferred to implement multi operand parallel addition .11:2 compressor tree can be implemented by using 5:3 compressors. The 5:3 compressor internal tree has three operand input signals, one sum output signal and two carry signals having two inputs and two outputs. One carry signal (cA), which is related to the CSA, The other (cB), which is related to the CPA, they forms a standard carry-chain.

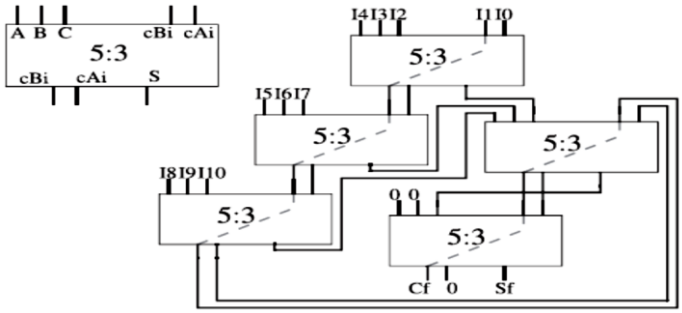


Figure.7. CS 11:2 compressor trees based on a linear array of 5:3 compressors.

As shown in above Fig. 7, 11:2 compressor tree implemented by constructing a linear array of 5:3 CAs. In this compressor tree two additional operands are introduced, the two carry input signals are connected to the corresponding carry outputs of the previous adder, after adding all input operands the sum-words generated and these are added in the same order. In this 11:2 compressor tree five 5:3 compressor adders are used for designing the tree, where as in 9:2 compressor tree seven compressor adders are used.

3.3 15:2 CS Compressor tree

Similar to its 4:2 and 7:2 compressors, the 15:2 compressors as shown in Fig. 8, is capable of adding 16 bits of inputs(x_0-x_{15}) and 4 carry's from the previous stages, at a time. It is constructed from two 7:2 compressors block diagram is show in fig: 8 and architecture is show in fig: 9 Figure

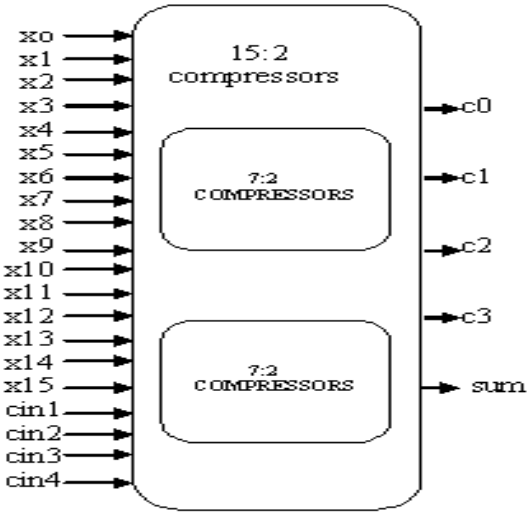


Figure.8. CS 15:2 compressor block diagram

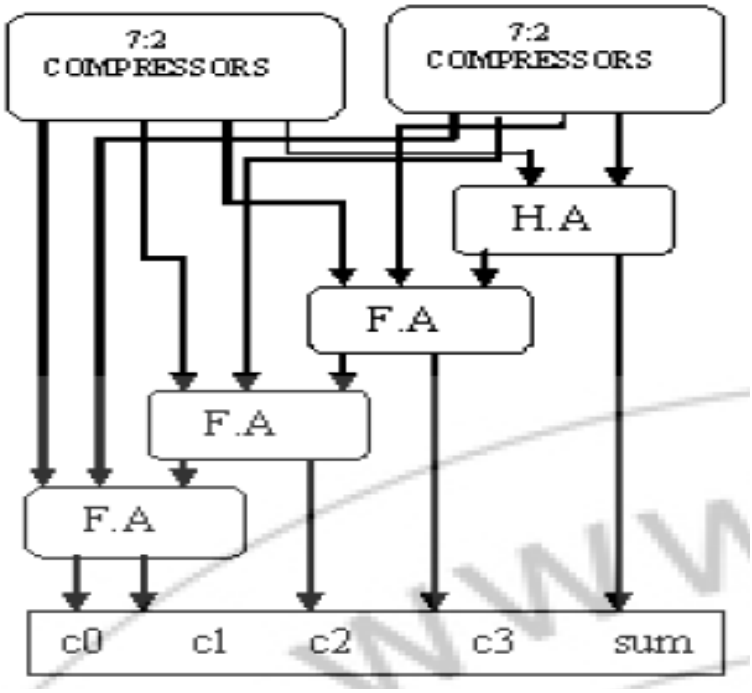


Figure.9. CS 15:2 compressor architecture

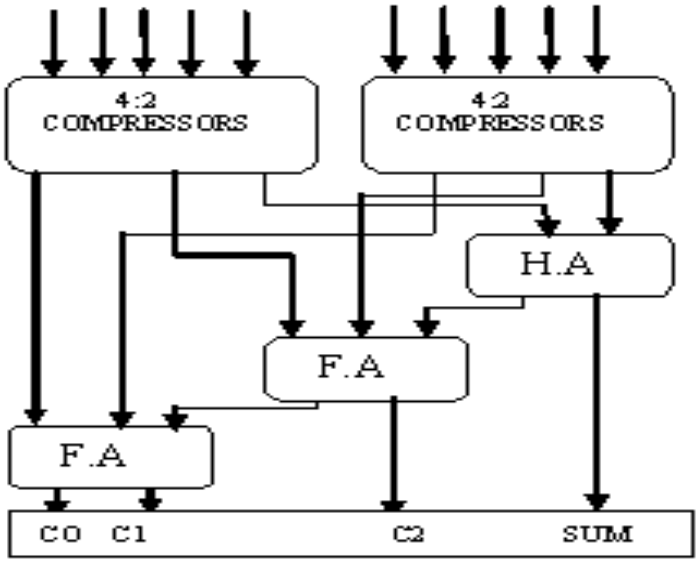
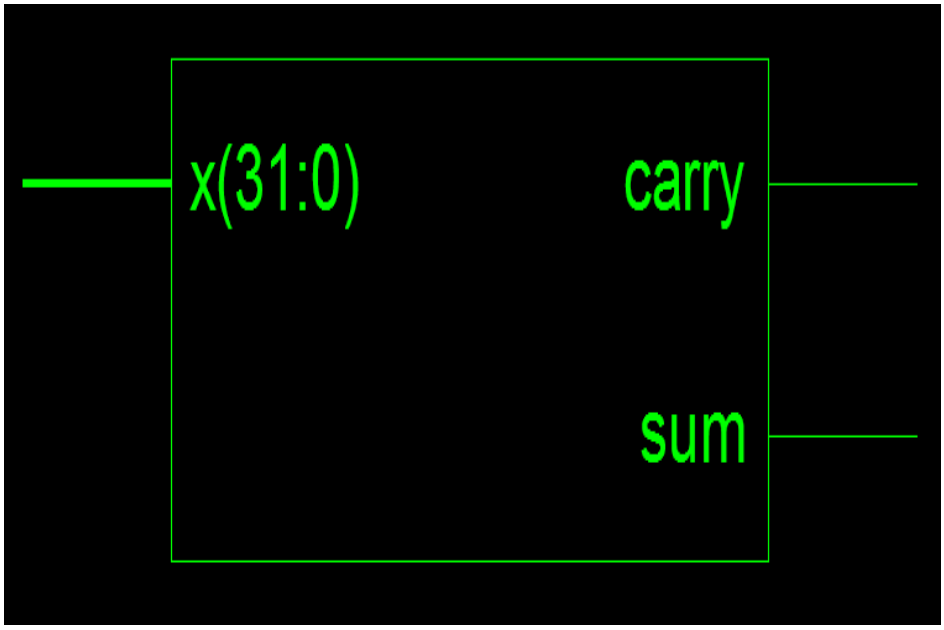


Figure.10.CS 7:2 compressor architecture

Here half adder and full adder are replaced with compressor for the addition of partial products which plays key role to enhance speed of operations. Block diagram and its architecture is as shown in figure. Proposed architectures an equivalent circuit, using full adders and half adders is as shown in fig9. 15:2 compressor is built by using two 7:2 compressors .so 7:2 compressor consist of one half adder and two full adders. The 7:2 compressors built by using two 4:2 compressors as shown in fi 10.7:2 compressors consist of 1 half adder and two full adders. Finally we design 15:2 compressors using 7:2 compressors, 4:2 compressors . 15:2 compressor consist of one half adder and 3 full adders.

3.4 32:2 CS Compressor tree



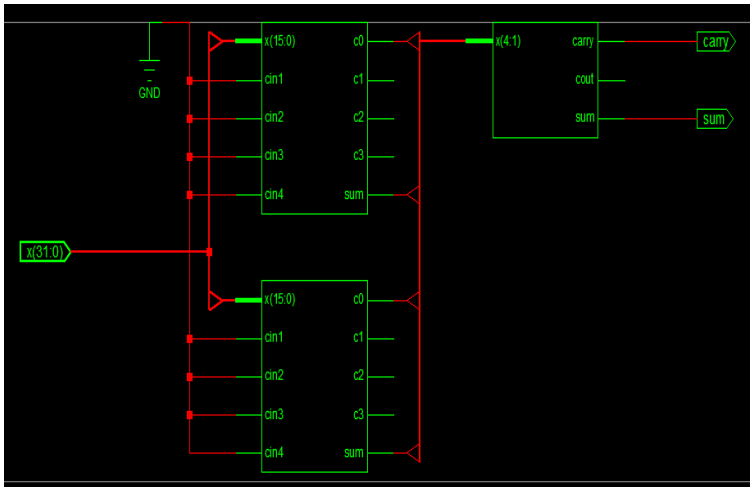


Figure.11.Schematic view of 32:2 compressor tree

The schematic views of 32:2 compressor trees as shown in above. The 32:2 compressor trees can be implemented by constructing array of two 15:2 CAs and one 4:2 CAs. This compressor tree consist of 32 inputs and two outputs sum and carry operands, the outputs of two 15:2 CAs sum and carry words are connected as inputs to 4:2 compressor tree. Finally we can estimate sum and carry for 32:2 compressor adder through usage of 4:2 compressor between two 15:2 compressors.

4. Application oriented design of CS Compressor trees

4.1 4x4 Wallace tree multiplier design

4x4 Wallace tree can be designed by using half adders and 3:2 compressors. This Wallace tree multiplier is arranged in three stages. The multiplication two 4 bit numbers ‘a’ and ‘b’ can be done by using 4x4 Wallace tree .finally we get sum and carry from these we get resultant sequence from p0 to p8 as shown in below figure ,this is multiplication result of 4x4 Wallace tree compressor. Wallace multiplier is used for process for multiplication of two numbers. By using the Wallace method, a three step process is used for the multiplication of two numbers; the bit products are formed. The bit product matrix are reduced to a two row matrix, where sum of the row equal to the sum of bit products, and two resulting rows of the bit product are summed with a fast adder(compressor) to produce a final product.

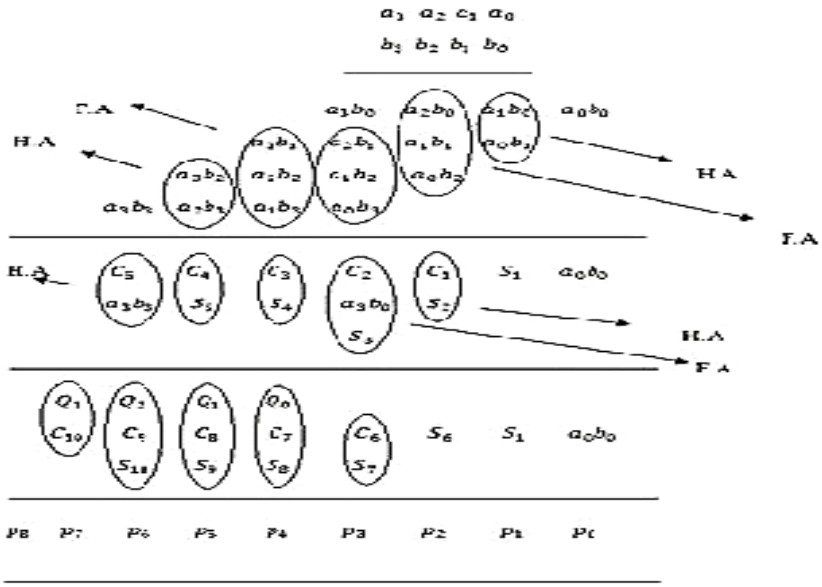


Figure 12.4x4 Wallace tree multiplier

4.2 8x8 Wallace tree multiplier design

8x8 Wallace tree can be designed by using half adders and 3:2 compressors,4:2 compressors,5:2 compressors,7:2 compressors and 9:2 compressors .By using these compressors we can built 8x8 Wallace tree multiplier. This Wallace tree multiplier is arranged in five stages. The multiplication two 8 bit numbers 'a' and 'b' can be done by using 8x8 Wallace tree. Finally we get sum and carry from these we get resultant sequence from p0 to p15 as shown in below figure, this is multiplication result of 8x8 Wallace tree compressors. Wallace multiplier is used for process for multiplication of two numbers.

By using the Wallace method, a three step process is used for the multiplication of two numbers; the bit products are formed. The bit product matrix are reduced to a two row matrix, where sum of the row equal to the sum of bit products, and two resulting rows of the bit product are summed with a fast adder(compressor) to produce a final product.

In the Wallace Tree method, three single bit signals are passed to a one bit full adder which is called a three input Wallace Tree circuit, and the output signal (sum) is supplied to the next stage full adder of the same bit, and the carry output signal thereof is passed to the next stage full adder of the same no of bit, and the carry output signal thereof is supplied to the next stage of the full adder located at a one bit higher position. During the partial product addition stage Compressor is mainly used to reduce the power consumption and the critical path delay is also highly reduced.

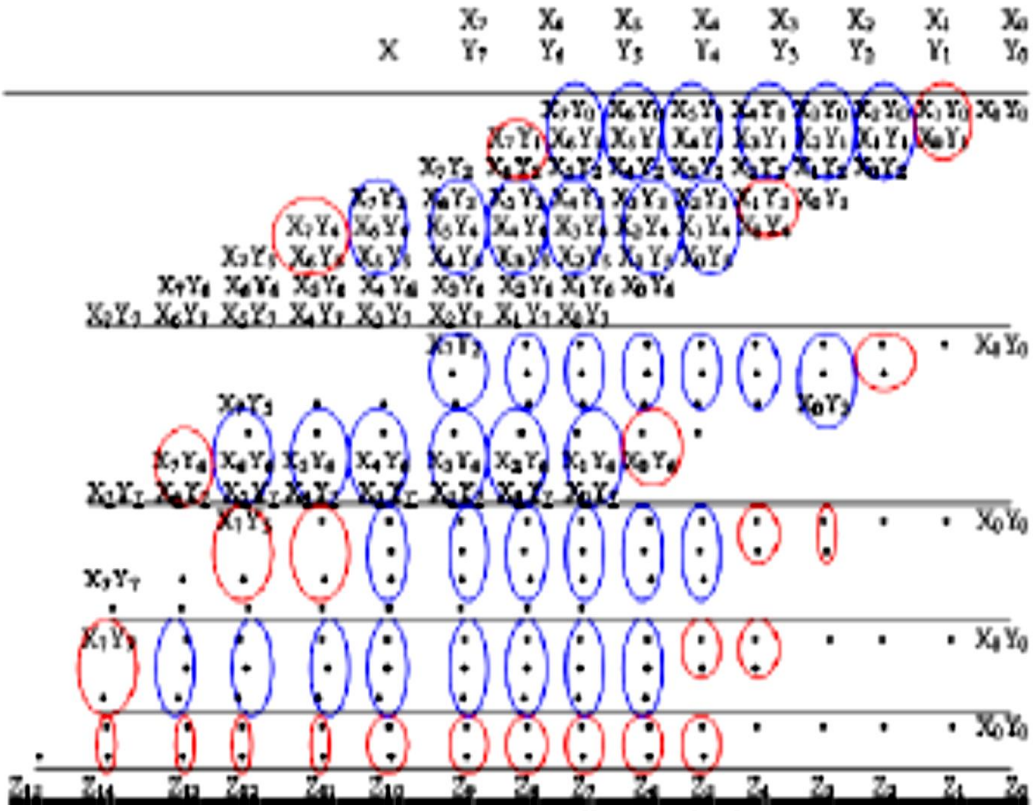


Figure13.8x8 Wallace multiplier

4. Simulation Results

The Proposed system is simulated and verified using Verilog HDL in Xlinix ISE 10.1i for the target device xc3s500e-5fg320. The below simulation results show the outputs of the 15to2 and 32to2 compressor trees.

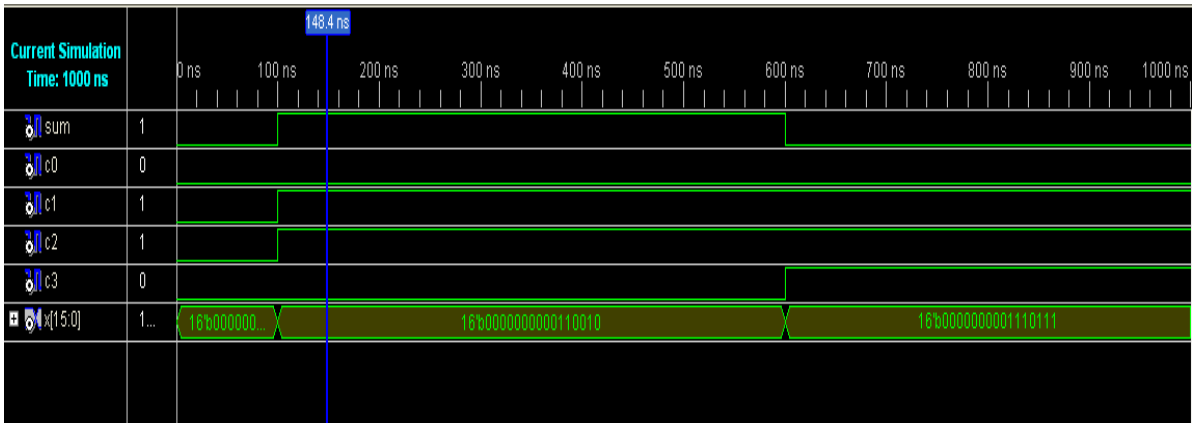


Figure.14. Simulation result for 15to2 compressor tree

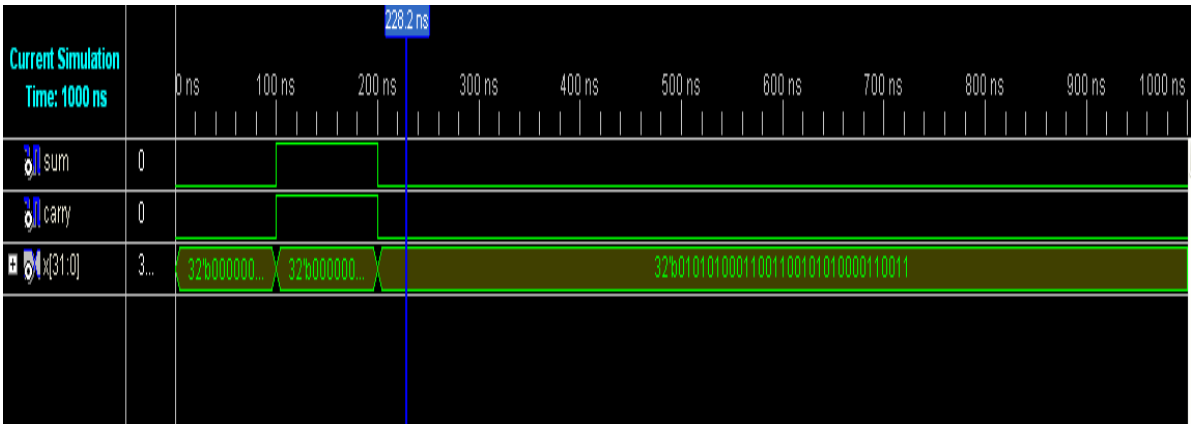


Figure.15. Simulation result for 32to2 compressor tree

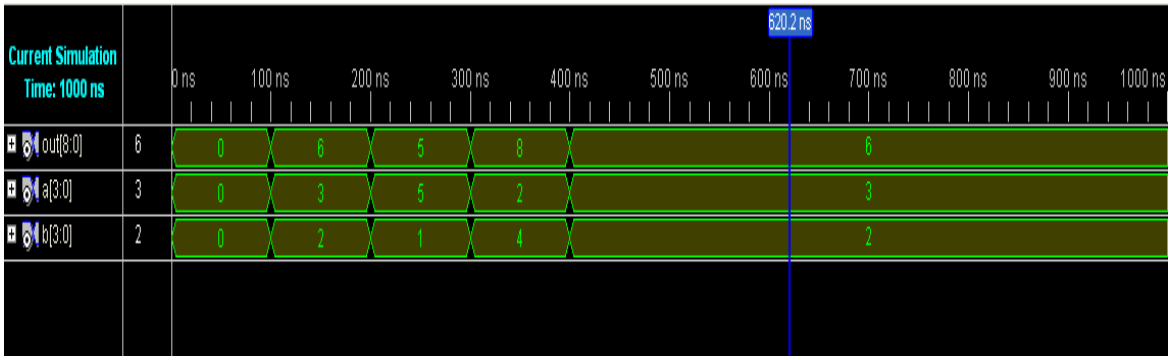


Figure16.simulation result of 4x4 Wallace multiplier using 3t02 compressor

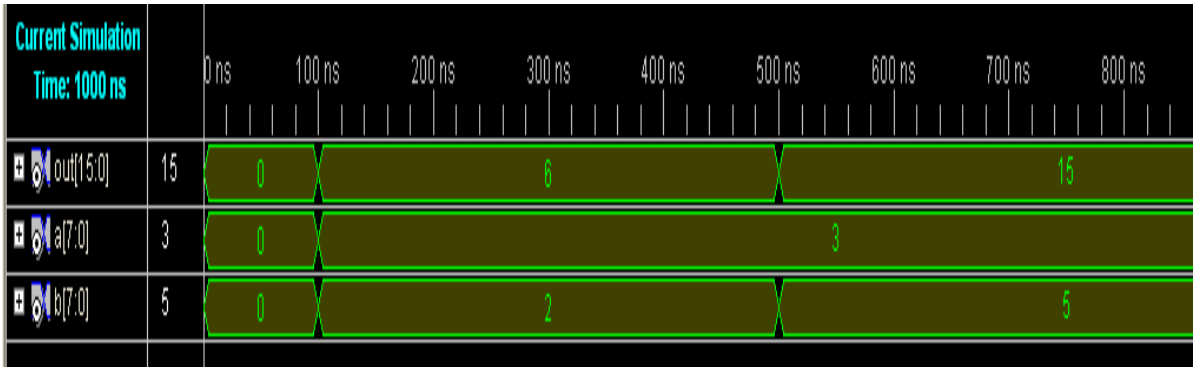


Figure17. Simulation result of 8x8 Wallace tree using 4to2,5to2,7to2,9to2 compressors

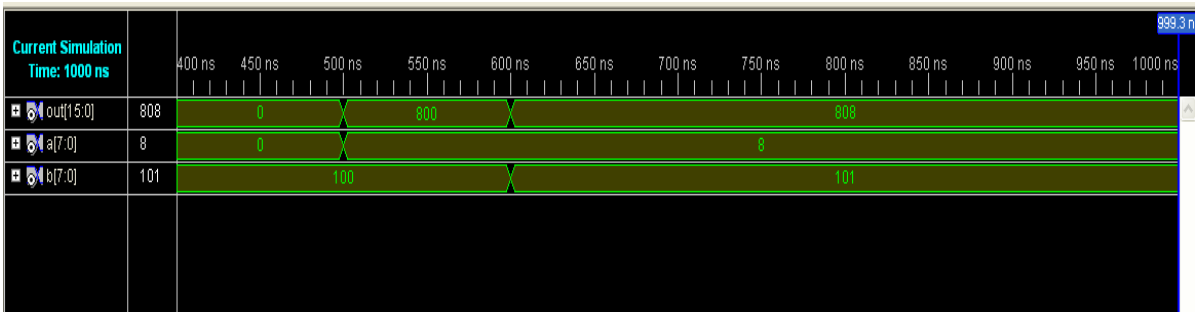


Figure18.simulation result of 8x8 Wallace tree using adders

The synthesis result of 15to2 and 32to2 compressor trees,4x4 Wallace multiplier and 8x8 Wallace multipliers have device utilization and Delay reports shown in the given below Table 1 and Table 2 respectively.

Table 1. Device utilization report

Compressor type	No. Of Slices	No. of 4 input LUTs	No. of IOs
9to2	5	9	11
15to2	12	21	25
32to2	24	43	34

Table 2. Delay comparison between 9to2, 15to2 and 32to2 compressor trees.

Compressor type	Delay (ns)
9to2	8.339
15to2	10.195
32to2	13.943

Table 3.device utilization report of Wallace trees

Wallace tree type	No. Of Slices	No. of 4 input LUTs	No. of IOs
4x4	19	34	17
8x8 with adders	86	150	32
8x8 with compressors	55	96	32

Table 4.delay comparison of Wallace trees 4x4 and 8x8

Wallace tree type	Delay (ns)
4x4	11.628
8x8 with adders	20.929
8x8 with compressors	19.512

5. Conclusion

In this work analyzes the designing compressor trees by using different compressor techniques they are 15:2 and 32:2 compressor arrays. These are used in different applications such as multipliers, digital signal processing, and hardware accelerators. By using these compressor techniques speed can be improved and presents no area overhead, delay can be reduced.

REFERENCES

[1] Javier hormigo,julio villalba , " Multi operand redundant compressors on FPGAs, " IEEE transactions on Computers vol.62, no.10, Oct. 2013

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

[4] 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.

[5] T. PreuBer and R. Spallek, "Enhancing FPGA Device Capabilities by the Automatic Logic Mapping to Additive-Carry Chains," Proc. Int'l Conf. Field Programmable Logic and Applications (FPL), pp. 318-325, 2010.

G. Sree Reddy: He is currently Studying M.tech-VLSI at Sree Vidyanikethan Engineering College, Tirupati. His areas of interest are Digital System Design, Low power VLSI Design.

T. kishore: He is currently working as an Assistant Professor in ECE department of Ravindra College of Engineering College and technology, Kurnool. He has completed M.tech in VLSI Design, in St. Joseph College of engineering, Kurnool. His research areas of interest are RFIC Design, Digital Design, and VLSI Signal Processing.

P. Mahendra: He is currently Studying M.tech-VLSI at Sree Vidyanikethan Engineering College, Tirupati. His areas of interest are Digital System Design, VLSI Design.