International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Design of a High Speed and Area Efficient Novel Adder for AES Applications

A. Radha, K.S.N. Murthy

Abstract--- In the design of VLSI circuits, there is huge power consumption because of circuit complexity. It is known that the demand for portable equipment is rapidly increasing now a days. Recently power efficient circuit designs have been concentrated. In complex arithmetic circuits adder is the most important building block. These are widely used in some other applications also like Central Processing Units, Arithmetic Logic Units and floating-point units. In case of cache memory access and in digital signal processing, these are used for address generation. Adders are most significant in control systems also. The speed of a processor and system accuracy is based on the performance of this adder. Regularly, Ripple Carry Adder is elected for two N-bit numbers adder due to fast design time of these RCAs among various other types of adders. Even though if RCA has fast design time, but it is limited in time because of that each full adder must wait for the carry bits of previous full adder blocks. A carry tree adder is proposed in this paper which is efficiently implemented technique at gate level for decreasing the delay and decreasing the memory usage.

Key Terms — Carry Tree Adder, CLA, RCA, Black Cell, Gray Cell

I. INTRODUCTION

Addition is the major function in the four elementary functions such as multiplication, subtraction and so on. In any digital system, addition is very important operation [4]. It is primary operation which is useful to implement all the other arithmetic functions. For designing a fast, accurate and low power consumed adder [2] directly increases the device speed. This adder can also be used for faster computational applications as well as to improve overall the system life.

The Arithmetic and Logic unit is the main block of digital systems like Digital Signal Processors (DSP), microprocessors, microcontrollers, and other data processing units [6]. In many arithmetic functions, an adder is an important element as a hardware unit for all the other applications. The addition function is also used in various other functions like decoding, encoding and so on. Generally, addition is a function of adding two numbers which produces the output known as the sum and the carry. All the complex adder structures are developed using Half Adder (HA) and Full Adder (FA) only.

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
A. Radha*, Research Scholar, Dept of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur (DT), Andhra Pradesh, India.
K.S.N. Murthy, Professor, Dept of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur(DT), Andhra Pradesh, India.

The complete basic function of adder is constructed using a Half Adder and it can be improved by a Full Adder. The carry bit which is obtained in the addition process is very important in the design of an adder and also decides the speed of the adder. To reduce the time delay of the propagated Carry, multiple adders are designed. Binary adder is one of the significant module of microprocessors. It is not only used to complete addition and subtract functions, but also can be used to achieve multiplication functions and so on. Some of the mostly used adders are CLA [1], Manchester Chain Adder, Carry Select Adder and Parallel Prefix Adder. Designing of an adder consists lot of constraints. The trade-off between the delay and the area is the important factor. Usually adder requires very less area that is why it is easy to implement, but the time taken to give the result is high. To overcome this, advanced techniques are coming in to picture. Now a day's including the speed, the power consumption is also a considerable parameter. So the design of an adder is useful to satisfy all the specifications [5]. Binary addition is the basic function that continuously plays a considerable effect on the modern-day digital system design like control systems and DSP circuits. Various types of adders are available in which each one has its own importance and performance. Selection of an adder is based on the specific use of that. Therefore, binary adders are needed to have fast computation time, high efficiency, less area and low power consumption.

Binary adder [3] is the most important element in any digital system and it determines the performance of that digital system. It is used in many applications like arithmetic and logic units, multipliers, memory addressing units and dividers. Implementation of binary adder with advanced technology improves the overall performance of the device as well as entire system. The main disadvantage of this adder is the carry chain. The number of input bits available at the input of the adder increases the length of the carry chain. To increase the efficiency of the carry propagate adder, it is need to extent the carry chain without eliminating it. So, most of the digital designers now a day's came with high speed adder by advancing the architecture of computer which tends to keep the critical path in many calculations. In this paper we mostly concentrated on the designing of an adder with high speed, device utilization and usage of cell.

Providing security for the data is the main important task in all types of transmission systems. Generating the secret key randomly produces better safety as well as good difficulty in cryptographic algorithms.

There are different symmetric key algorithms exist which uses the similar key for encryption and decryption.Among all the algorithms, Advanced Encryption Standard (AES) algorithm is used in many fields communications for security.In AES, encoding of data involves four rounds of operations.In these operations addition is the most imporatant operation which is done by using Adders.

Adder is the basic building block of the complex arithmetic circuits which increases the consumption of power in VLSI circuits.The speed of the processor and system accuracy is based on the performance of this adder only. To resolve these issues related to system complexity, which reduces the speed and increases the overall time delay of the system performance, the proposed system uses a carry tree adder.

II. EXISTED SYSTEM

The Ripple Carry Adder is the basic adder which adds two N-bit inputs (where N is a positive integer) and gives (N + 1) output bits (N-bit sum and a 1-bit carry). Ripple Carry Adder is developed from N number of Full Adders which are connected together with the carry output bit of first Full Adder tied to the input carry bit of the next Full Adder. The input bits are labeled as A and B, the carry output of each Full Adder is labeled as cout which is the cin of the next full adder stages and the sum bit is labeled as sum. Sum output needs both input bits and carry-in bit before it can be evaluated.

In order to calculate the delay of propagation of this adder, we should keep time delay over every possible combination of inputs as minimum as possible, which is called as critical path. Most important parameter sum output bit can be evaluated if you know the carryout of the previous full adder. In adverse case i.e., if all the carry output bits are ones, then the carry needs to ripple from LSB bit position to the MSB position. For a ripple carry adder, let's take an adverse example in the view of propagation delay, when the input bits changed from 1111 to 1111 and 0000 to 0001, which results an output bit named as sum changes from 01111 to10000.

In the view of VLSI design, Ripple Carry Adder is the basic adder in the implementation. First, we need to implement and layout one full adder cell, then array of N number of these full adder cells are connected together for creating an N-bit RCA. Achievement of high-speed primary full adder cell determines the entire Ripple Carry Adder speed. From the critical path, the carry output time delay of full adder can be reduced. Various methods are available to implement full adder cell to reduce the carry output time delay.

The circuit of a Multiplier is implemented using full adder, which can be cascaded for adding bits which are N in number in parallel. The parallel adder contains "n" number of N bit full adder circuits. In a ripple carry adder, each full adder carry output is the input carry of the following Full Adder stages, which is known as ripple carry due to each carry bit is rippled into the following stages. At any stage of half adder the bits of carry- out and sum are not considered until the input carry of that stage exists in a ripple carry adder due to the logic propagation delay in the internal circuit.

The delay of Propagation is the time difference between the instant of time at which input is applied and instant of time at which output is obtained. For example, for a NOT gate, if the input is "0" then the output of that NOT gate is "1" vice versa. The time taken for the NOT gate to become output of that gate is "0" after the application of logic "1" to the input of the NOT gate is the delay propagation in this example. Likewise, carry propagation delay is the time elapsed among the signal of input carry application and the existence of the output carry. Ripple Carry Adder is represented as shown in below fig1.

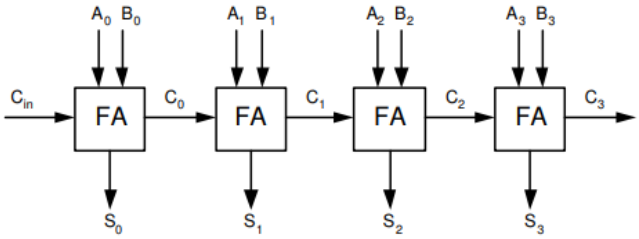

Fig. 1: FOUR BIT RCA

III. PROPOSED SYSTEM

The proposed carry tree adder is utilized for performing the operation of addition. It is looking like structure of a tree for performing the arithmetic operation. Proposed carry tree adder is utilized for higher performance operation of addition. This adder contains Black cells and Gray cells. Each Black cell contains two AND gates and one OR gate. Multiplexer is a combinational circuit which contains multiple inputs and one output. Gray cell contains single AND gate only. Propagate is represented as P_n and it contains single AND gate only which is given in below equation 1. Generate is denoted as G_n and it contains single AND gate and single OR gate which is denoted in below equation 2.

$$P_n=B_n \text{ AND } B_{n-1} \quad (1)$$

$$G_n=A_n \text{ OR } [B_n \text{ AND } A_{n-1}] \quad (2)$$

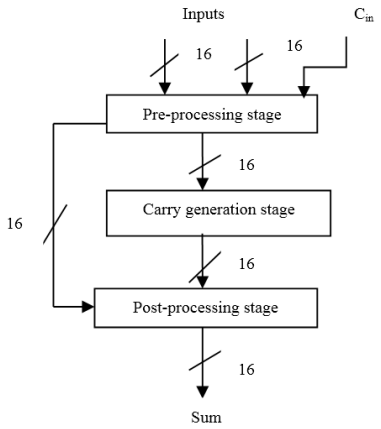

Fig. 2: BLOCK DIAGRAM OF PROPOSED ADDER

Generate is represented as Gn and it also implemented using one OR gate and one AND gate which is given in below equation 3.

Retrieval Number: B1057078219/19©BEIESP
DOI: 10.35940/ijrte.B1057.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$G_{n-1}=A_{n-2}$ OR $[B_{n-2}$ AND $A_{n-1}]$ (3)

The proposed carry tree adder is efficient and delay is reduced. The Novel technique for addition is efficient to increase the speed of the binary addition. This new structure looks like a tree to achieve higher efficiency for arithmetic operations. Motivate to make research on the binary elements produces improvement in the Field programmable gate array [FPGA] devices which are more popular now a days because they improve the microprocessor operation which is used in many applications such as mobile, DSP and telecommunication. The construction of Novel technique for addition contains three stages named as pre- computing stage, Carry Generation Network and post- computing stage as shown in fig 2.

Proposed carry tree adder contains following primary stages, namely

1) Pre-computing
2) Carry Generation Network
3) Post-computing

1)Pre-computing stage: In this stage of calculation two signals such as Propagate and Generate are implemented using a pair of two input bits A and B. Propagate is the "XOR" function of two input bits and Generate uses "AND" function of two input bits. The Propagate (P_n) and Generate (G_n) are represented in following two equations 1 & 2.

$P_n = A_n$ XOR B_n (1)

$G_n = A_n$ AND B_n (2)

2) Carry Generation Network: In this stage, the Proposed adder differ from other PPA's in the structure. So, it forms the performance indicating stage of the adder. It contains the transforming elements and buffer elements. Less number of transforming elements reduces the time delay. This stage generates carry for each bit which is known as Carry Generate (Cg). The Carry Propagate and Carry Generate are used for another process which provides carry at final cell exist in each bit process. The final carry bit provides Sum output of the next bit simultaneously up to the final bit. This carry is used for the next bit sum operation. The Carry Generate and Carry Propagate are represented in the following equations 4 & 5. The output of buffer element is same as that of input. The output of the transforming element is

$C_p = P_{i+1}$ AND P_i (3)

$C_g = G_{i+1}$ OR $(P_{i+1}$ AND $G_i)$ (4)

Where i = 0,1,2,3.....N. The Carry Propagate C_p in above equation 3 is known as Black cell and Carry Generate C_g in above equation 4 is known as Gray cell.

3) Post computing stage: The sum output and the final carry output are evaluated in this stage. It is the final stage of novel technique for addition. The carry output of an initial input bit is XORed with the next bit of Propagate which generates the sum output given in below equation 5.

$S_n = P_n$ XOR C_{n-1} (5)

It is useful in the addition of two sixteen bits. Each and every carry bit go through the post computing step including Propagate gives the last sum output. Initial input bit go through the pre computing step which gives two signals such as Propagate and Generate. This Propagate as well as Generate are used for carry generation to provide Carry Generate and Carry Propagate, which are used in post-computing stage to provide final sum. This step by step process of novel technique for addition is shown in below Fig 3.

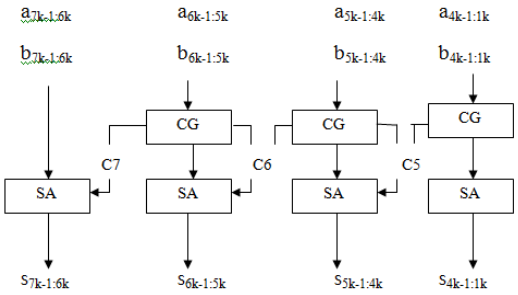

Fig. 3: ERROR TOLERANT ADDER

The representation of block-based adder which is known as Error Tolerant Adder (ETA) is shown in above Fig3. It splits N bit adder into m number of sub adders of equal bit length of k = n=m. The carry generation is represented as CG and sub adder is represented as SA. The carry-in input signal to each sub adder is generated from the previous k bits by a carry generator, while the carry-in bit to each carry generator is logic0, which is essentially truncates the carry chain.

The structure of Novel technique for addition which looks like a structure of tree can be used to achieve high efficient arithmetic operation. This is the high speed adder implemented using minimum number of gates at gate level logic. Therefore, it reduces the time delay and memory utilized in this architecture.

The Novel technique for addition is illustrated in below Fig 4 which increases the speed and reduces the area for the 8-bit addition operation. A_n and B_n are the input bits which concentrates on the signals of generate and propagate by using the operations of XOR and AND. These propagate and generate, which are used to perform the functions of Black cell and Gray cell provides carry C_n. The obtained carry XORed with the Propagate of next bit and gives an output sum.

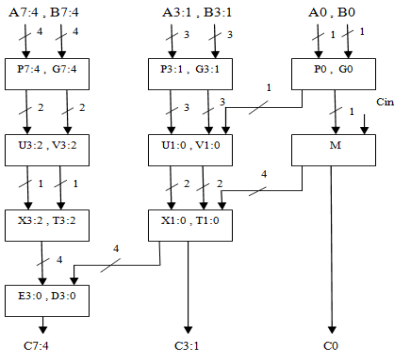

Fig. 4: NOVEL TECHNIQUE FOR ADDITION

This paper proposes the utilization of a proposed carry tree adder for the designing of a fault tolerant adder which is used to handle both correction and detection of errors in the Carry Tree Adder. The proposed system essentially has the same delay when it is realized using FPGA. The proposed Carry Tree is very much abridged, even if it has small Ripple Carry Adders in the implementation. RCA adder is relatively fast when it is realized on Field Programmable Gate Array, without decreasing the speed due to the simplicity in the Carry Tree which maximizes the adder's critical path. This proposed system is illustrated in below fig 5.

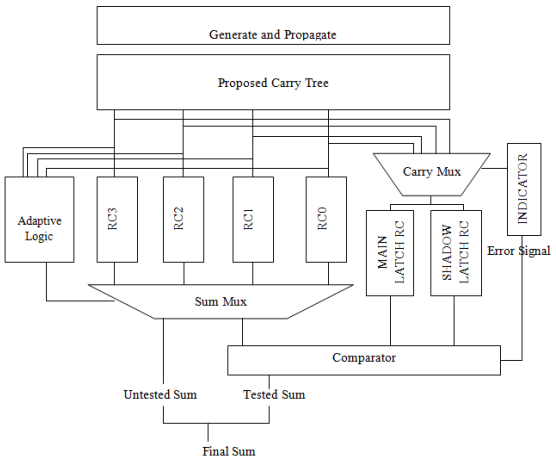

Fig. 5: ARCHITECTURE OF PROPOSED ADDER

The blocks which are labeled as RC0 to RC3 are equally sized Ripple Carry Adders which are used for fault tolerant by including two additional Ripple Carry Adders named as Main Latch RC and Shadow Latch RC as shown in above figure 5. RC0 to RC3, main latch RC and shadow latch RC blocks are similar in bit width. Some multiplexers, control circuits, and comparators are added to RCAs which can be used for the fault tolerant. In each clock cycle one of the adder in the available four adders from RC0 to RC3 can choose for testing by sending its input to both main latch RC and shadow latch RC blocks. The selected RCA (one of the Ripple Carry Adder in RC0 toRC3) and Sum Mux are connected to the Adaptive Logic. RCA under test output is compared with the outputs of two RC latches, an error can be detected as well as corrected.

The control circuit simply contains an indicator with the clock. The indicator turns on a set of multiplexers which determines the input of the main latch RC and shadow latch RC units. Indicator is also used to connect the output of the choosed RCA (any one of RCA from RC0 to RC3) to the comparator to compare its output with the output of two RC Latch units. The validated sum is latched and connected to the remaining sum bits when the clock edge is falling.

Proposed carry tree adder avoids the complexity of wires resulting in smaller area and power consumption. It has maximum logic depth, limited fan-out at each stage. The smart idea of this design is to evaluate prefixes of 2-bit groups first, these are then utilized for finding prefixes of 4-bit groups and in turn used to find the prefixes of 8-bit groups, etc [7]. In this design issue the signals of Propagate and Generate take more stages to calculate [8].

Proposed carry tree adder features with low network complexity comparing with the existed adder. The low network complexity assists to reduce the area of adder resulting in reducing the power consumption also. This feature makes proposed carry tree adder more efficient compared to existing adder, which has more black competition nodes and long wires.

IV. RESULTS

The performance results of our proposed system are shown in below charts, and are compared with related work.

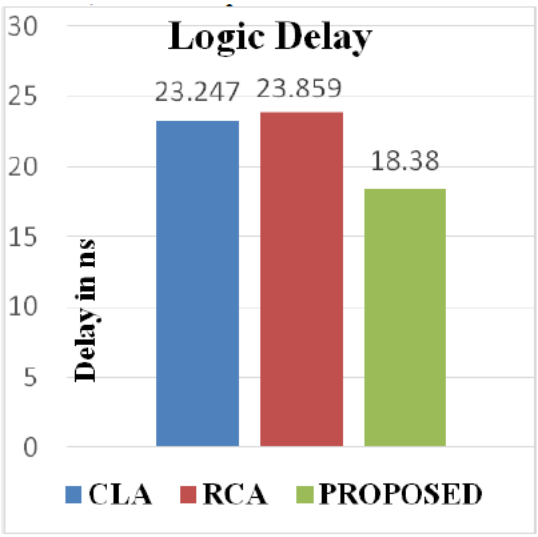

Fig. 6: LOGIC DELAY

Performance comparision of existing system with proposed system in terms of Logic Delay is shown in above column-chart.

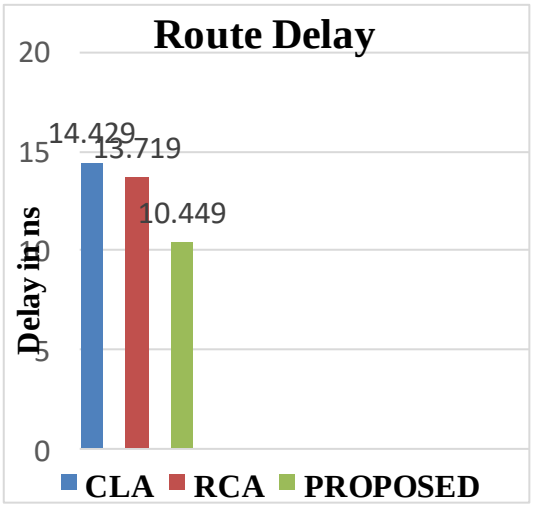

Fig. 7: ROUTE DELAY

Performance comparision of existing system with proposed system in terms of Route Delay is shown in above column-chart.

In Existing system the Ripple Carry Adder (RCA) produce a route delay of 36.5% and with Carry-Look-A

head Adder(CLA) produced a route delay of 38.3%. In our proposed system high performance speed with reduction in a route delay of 36.2% is obatined.

Performance comparision of existing system with proposed system in terms of Total Delay is shown in below coulmn chart.

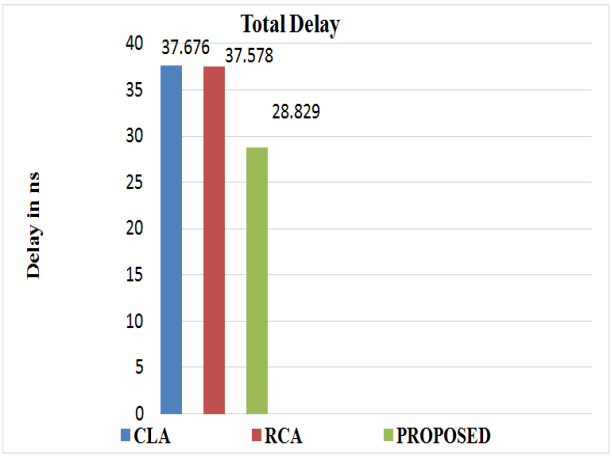

Fig. 8: TOTAL DELAY

In Existing system the Ripple Carry Adder (RCA) produce Total Delay of 37.578ns and with a Carry-Look-A head Adder(CLA) produced a total delay of 37.676ns. In our proposed system high performance speed with reduction in a total delay of 28.829ns is obatined .

In Existing system the Ripple Carry Adder (RCA) used a memory 4536512 KB and with a Carry-Look-A head Adder(CLA) used a memory of 4536512 KB. In our proposed system which uses a memory of 4536552 KB with high performance speed with reduction in a total delay of 28.829ns is shown in fig 9 using pie-chart.

In this proposed method, the Look Up Tables have been minimized by keeping the equal number of buffer clocks as same as in existing system.

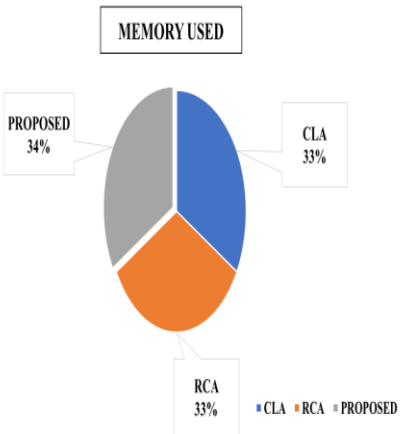

Fig. 9: MEMORY USED

The evaluation speed is optimized by decreasing the number of Look Up Tables used in the proposed algorithm.The generation of substitution byte box will easily creates a state transition matrix in the encryption process with limited number of Look Up Tables, which generates a cipher text with less overall system delay.

V. CONCLUSION

In the proposed paper, a novel technique is developed for addition which is implemented at gate level to decrease the time delay and reduces the area. This technique is implemented with the functions of pre computation, Carry Generation Network and post computation. The pre computation step concentrates on Propagate and Generate. Carry Generation Network step concentrates on carry generation and post computation step concentrates on the final result. It looks like a structure of tree which reduces the number of cells in the Carry Generation step to speed up the binary addition. This proposed adder performs addition function with a better benefit in delay reduction such as a route delay of 10.449ns, logic delay of 18.280ns and total delay of 28.829ns.

REFERENCES

1. Richard P. Brent and H.T. Kung, "*A Regular Layout for Parallel Adders*", IEEE Transactions on Computers Volume 31 Issue 3, March 1982, Pages 260-264.
2. J. Grad and J.E. Stine, "*A Standard Cell Library for Student Projects*", International Conference on Microelectronics Systems Education, pages 98–99. IEEE Computer Society Press 2003, 2003.
3. S. Knowles, "*A family of adders*", Proc. 15th IEEE Symp. Comp. Arith., June 2001, Pages. 277-281.
4. Han, Carlson, "*Fast Area-Efficient VLSI Adders*", IEEE, 1987.
5. G. Dimitrakopoulos and D. Nikolos, "*High-speed parallel-prefix VLSI ling adders*", IEEE Trans. Comput., vol. 54, no. 2, Feb.2005, Pages. 225–231.
6. V. Dave, E. Oruklu, and J. Saniie, "Performance evaluation of flagged prefix adders for constant addition," in Proc. IEEE Int. Conf. Electro/ inf. Technol., 2006, Pages 415–420.
7. GeetaRani, Sachin Kumar, "*Delay Analysis of Parallel-Prefix Adders*", International Journal of Science and Research, Volume 3 Issue 6, June 2014, Pages 2339-2342.
8. Kogge P and Stone H, "*A Parallel Algorithm for the Efficient Solutions of a General Class of Recurrence Relations*", IEEE Transactions on Computers, Vol. C-22, No.8, 1973.