JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Design of High Speed Multi-Operand RCA Based Binary Tree Adder Using Reversible Logic Gates

[1]B. Nishanth Kumar, [2]T.SR.CH.Murthy, [3]M.Kiran Kumar, [4]Amrita Sajja

[1]Research Scholar, [2,3,4]Assistant Professor
[1,2,3,4]Electronics and Communication Engineering,
Anurag University, Hyderabad, India.

Abstract: Addition is one of the most basic operations performed in all computing units, including microprocessors and digital signal processors. It is also a basic unit utilized in various complicated algorithms of multiplication and division. Efficient implementation of an adder circuit usually revolves around reducing the cost to propagate the carry between successive bit positions. Hence, a new high-speed and area efficient adder architecture is proposed using proposed logic to perform the multi-operand binary addition that consumes substantially less area, low power and drastically reduces the adder delay. Further, as an enhancement of this project reversible logic gates are introduced to optimise constrained parameters.

IndexTerms **- Multi operand adder, Latency, Density, Ripple carry adder, Reversible logic, Propagation.**

I. INTRODUCTION

Besides technological scaling, advances in the field of computer architecture have also contributed to the exponential growth in performance of digital computer hardware. The flip-side of the rising processor performance is an unprecedented increase in hardware and software complexity. Increasing complexity leads to high development costs, difficulty with testability and verifiability, and less adaptability. The challenge in front of computer designers is therefore to opt for simpler, robust, and easily certifiable circuits. Computer arithmetic, here plays a key role aiding computer architects with this challenge. It is one of the oldest sub-fields of computer architecture. The bulk of hardware in earlier computers resided in the accumulator and other arithmetic/logic circuits. Successful operation of computer arithmetic circuits was taken for granted and high performance of these circuits has been routinely expected. This context has been changing due to various reasons.TO ACHIEVE optimal system performance while maintaining physical security, it is necessary to implement the cryptography algorithms on hardware . Modular arithmetic such as modular exponentiation, modular multiplication and modular addition is frequently used for the arithmetic operations in various cryptography algorithms . Therefore, the performance of the cryptography algorithm depends on the efficient implementation of the congruential modular arithmetic operation. The most efficient approach to implement the modular multiplication and exponentiation is the Montgomery algorithm whose critical operation is based on multi-operand binary addition . The area, delay and energy efficiency of the BTA depends on the performance of adders used in the structure. There are various types of adders such as ripple carry adder (RCA), carry-look-ahead adder (CLA), parallel prefix adder (PPA), carry-select adder (CSLA) and carry-skip adder (CSKA), which can be used to develop the BTA structure, where each adder has its own tradeoffs between area, delay and energy consumption. Among these adders, RCA has less area and energy with higher delay, whereas other adders (fast adders) having less delay with higher area and energy. On the other hand, area and energy are the two major constraints in most of the systems which demand the addition of a large number of operands should be hardware and energy efficient. Therefore, RCA is mostly preferred over fast adders in the BTA structure but its higher delay limits the overall speed of the system.

II. LITERATURE SURVEY

The Multi-Operand Adders are generally implemented in two methods i.e Array Adders and Adder Tree structure. In Array Adder structure, two operands are added and output is added with third operand and continues the chain of addition until to get final sum output. It requires ‘K’ number of adder levels for addition of ‘K’ operands. But in case of Adder Tree structure the number of levels to add ‘K’ operands is less than that of Array Adders. It groups ‘K’ number of operands into sets of two operands. All the sets are added parallel in one level. The sum outputs from first level again grouped into sets of two operands and perform addition. This process continues until to get two operands and added in last level to obtain final sum. In each level it reduces number of operands to half. Therefore it requires log2 K levels. The Adder Tree structure is faster than Adder Array structure with same resources consumed by both configurations. But the Array Adder is having regular routing than Adder Tree structure. The Ripple Carry Adder (RCA) or Carry Look Ahead Adder (CLA) are two general Carry Propagate Adders used in the above methods i.e. Array Adder, Adder Tree is Carry Propagate Adder. The delay of their CPA depends on bit length of operand. For N-bit operand the of RCA proportional to N and for CLA it is proportional tolog2 N. To reduce the delay these adders where implemented on FPGA by using dedicated carry chains [8]. The RCA on FPGA using fast carry chain is simpler than any other

CPA topologies at an expense of high hardware cost [9]. The pipelining technique can be applied more effectively RCA [1]. The delay of Adder Tree using CPA is high due to carry propagation along the bit length. Carry Save Adder tree is used as another approach for implementing Multi-Operand Adders. Here the carry is directly propagated to next level instead of propagating in the same as in case of CPA. The advantage of Carry Save Adder (CSA) tree is utilized in ASIC implementation due to flexible routing. The critical path delay can be minimized by optimizing the interconnection between Full Adders. But to implement on FPGA the Ripple Carry Adder tree is preferred than CSA adder tree. When CSA tree is implemented on FPGA it become slower than RCA tree due to routing delay of CSA. However, a straightforward implementation on FPGAs [6] roughly requires double hardware than a carry ripple adder, and does not exploit the fast carry chain to improve speed.

III. EXISTING METHOD

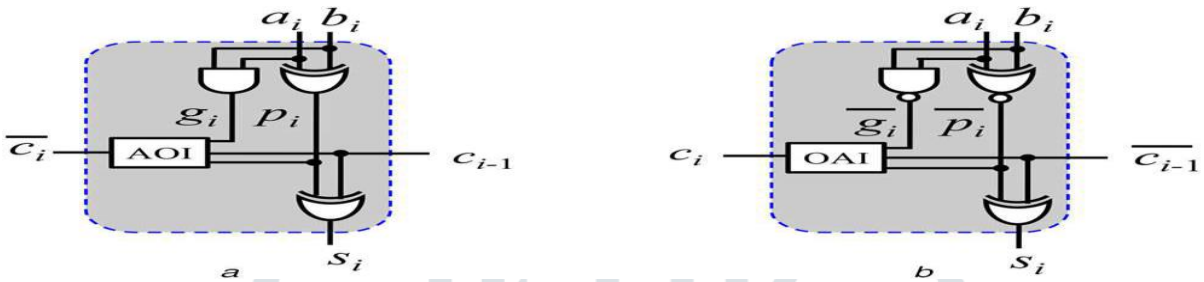

Fig.1 Logic diagram of Existing (a) AOI-LC, (b) OAI-LC

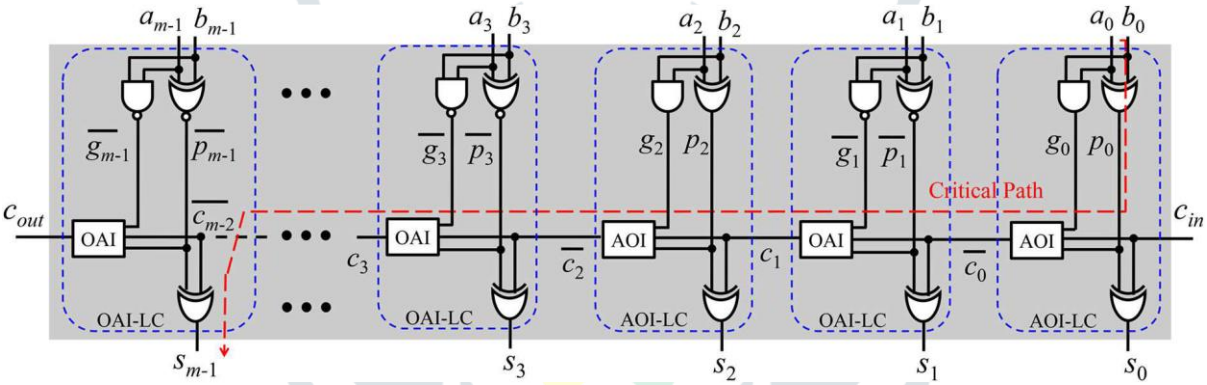

Fig.2 Existing *m-bit RCA design*

The two 1-bit logic cells (AOI-LC and OAI-LC) are derived using the proposed logic formulation given in (9)–(11) and (12)–(14), and shown in Fig. 1. The AOI-LC takes inputs a_i, b_i and c_{i-1}, and computes sum (s_i) and intermediate carry-out c_i signals while the OAI-LC computes sum (s_i) and carry-out (c_i) signals using 1-bit input signals a_i, b_i and c_{i-1}. Using AOI-LC and OAI-LC modules, an *m*-bit RCA design is proposed as shown in Fig. 2(for even value of *m*). In the proposed RCA design, the AOI-LC and OAI-LC modules are connected alternatively because AOI-LC generates carry-out in the complement form while the OAI-LC generates normal carry-out. It takes *m*-bit inputs (*a* and *b*) and initial carry-in (c_{in}) to compute the sum (*s*) and carry-out c_{out} signals, where the output carry is equal to c_{m-1}. If the value of *m* is odd then the proposed RCA includes AOI-LC in the MSB bit position which generates the carry-out signal in the complement form. Therefore, an actual output carry is obtained by complementing the carry-out signal of the AOI-LC placed in the MSB position.

Table 1 Area and delay of CMOS logic cells (TCBN65GPLUS TSMC 65 nm core library databook)

Gates	Cell name	Delay, ps	Area (transistor count)
AND	AN2D0	26.15	6
OR	OR2D0	28.5	6
NOR	NR2D0	13.95	4
NAND	ND2D0	12.75	4
NOT	INVD0	9.4	2
XOR	XOR2D0	42	12
XNOR	XNR2D0	40.85	12
AOI	AOI21D0	18.65	6
AO	AO21D0	33.45	8
OAI	OAI21D0	17.55	6

This subsection presents the comparison of the proposed RCA with the existing RCA design on the basis of the theoretical estimation of delay and logic complexities. The approach is considered for theoretical estimation of area and delay complexities of the proposed and existing RCA designs. For theoretical calculation, area and delay of two-input gates (AND, OR, NAND, NOR, XOR, XNOR), NOT gate and three-input complex gates (AOI, OAI, AO) are taken from TCBN65GPLUS TSMC 65 nm core library data book and listed in Table 1. By using Table 1, the expressions of area (in transistor count) and delay (in pico-second) for existing RCA are derived and given below:

$$A_{RCA} = mA_{FA} = 38\,\mathrm{m}$$
$$T_{RCA} = (m-1)T_{AO} + 2T_{XOR} = 33.45\,\mathrm{m} + 50.55$$

where A_{RCA} and A_{FA} denote the areas of RCA and FA, respectively; T_{RCA}, T_{AO} and T_{XOR} are the delays of RCA, AO and XOR gates, respectively. Similarly, the area and delay relations for the proposed RCA design are derived and given as:

$$A_{prop} = \left\lceil \frac{m}{2} \right\rceil A_{AOI-LC} + \left\lfloor \frac{m}{2} \right\rfloor A_{OAI-LC} = 36\left\lceil \frac{m}{2} \right\rceil + 34\left\lfloor \frac{m}{2} \right\rfloor$$

$$T_{prop} = \left\lfloor \frac{m}{2} \right\rfloor T_{AOI} + \left(\left\lceil \frac{m}{2} \right\rceil - 1\right) T_{OAI} + 2T_{XOR}$$

$$= 18.65\left\lfloor \frac{m}{2} \right\rfloor + 17.55\left\lceil \frac{m}{2} \right\rceil + 66.45$$

where $m/2$ and $m/2$ represent the smallest positive integer greater than or equal to $m/2$ and the largest positive integer less than or equal to $m/2$, respectively; A_{prop}, A_{AOI-LC} and A_{OAI-LC} denote the areas of the proposed RCA, AOI-LC and OAI-LC, respectively; and T_{prop}, T_{AOI} and T_{OAI} are the delays of the proposed RCA, AOI and OAI gates, respectively. For the odd value of m, the output carry is generated in the complemented form. Therefore, one NOT gate is used to achieve the output carry in the normal form and the calculated area includes two more transistors (one NOT gate).

Drawbacks of existing system:

- Delay is maximum and hardware complexity of the circuit is more.
- Area of the design and computational time is high which leads to higher power consumption. More combinational path delay along with more latency is main drawbacks in existing method.

IV. PROPOSED METHOD:

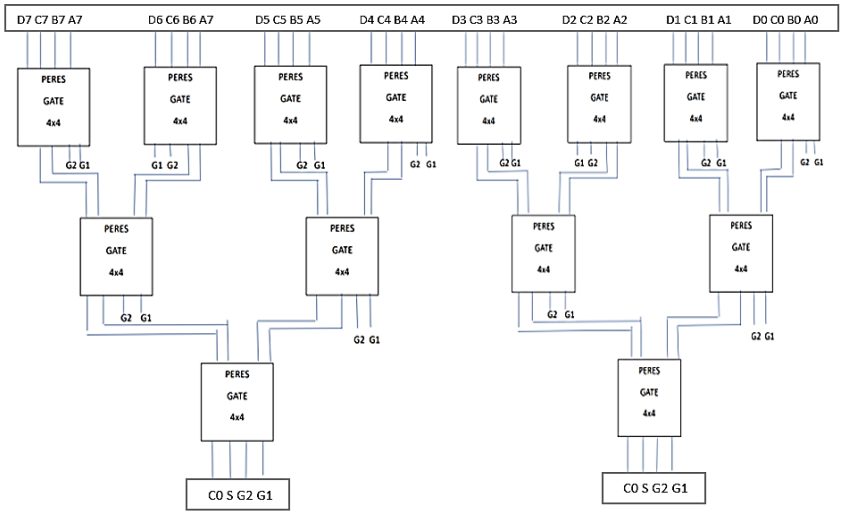

Fig.3 BTA Structure for 8 operands, each operand 4bit using peres gate

- **REVERSIBLE LOGIC GATES:**

High-performance chips releasing large amounts of heat impose practical limitation on how far can we improve the performance of the system. Reversible circuits that conserve information, by un computing bits instead of throwing them away, will soon offer the only physically possible way to keep improving performance. Reversible computing will also lead to improvement in energy efficiency. Energy efficiency will fundamentally affect the speed of circuits such as nano circuits and therefore the speed of most computing applications. To increase the portability of devices again reversible computing is required. It will let circuit element sizes to reduce to atomic size limits and hence devices will become more portable. Although the hardware design costs incurred in near future may be high but the power cost and performance being more dominant than logic hardware cost in today's computing era, the need of reversible computing cannot be ignored.

A. Reversible Function:

The multiple output Boolean function $F(x1; x2; :::; xn)$ of n Boolean variables is called reversible if:a. The number of outputs is equal to the number of inputs; b. Any output pattern has a unique pre-image. In other words, reversible functions are those that perform permutations of the set of input vectors

B. Reversible logic gate:

Reversible Gates are circuits in which number of outputs is equal to the number of inputs and there is a one to one correspondence between the vector of inputs and outputs. It not only helps us to determine the outputs from the inputs but also helps us to uniquely recover the inputs from the outputs.

- ***Peres Gate***

Fig.4 shows a 3*3 Peres gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R). The output is defined by P = A, Q = A ^ B and R=AB ^ C. Quantum cost of a Peres gate is 4. In the proposed design Peres gate is used because of its lowest quantum cost.

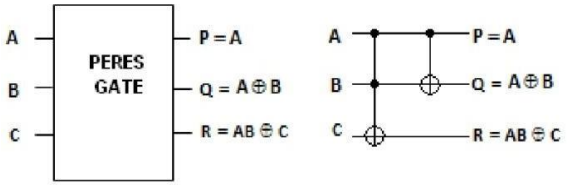

Fig 4: Peres gate

Table 2: Truth table of peres gate

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

A full- adder using two Peres gates is as shown in Fig.5. The quantum realization of this shows that its quantum cost is 8 two Peres gates are used

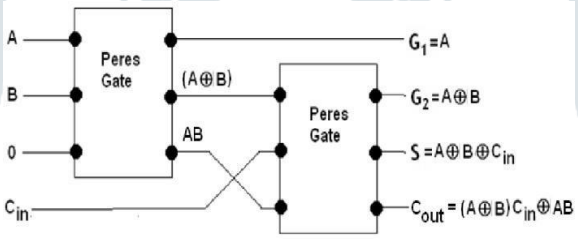

Fig 5: Full adder using two Peres gates

A single 4*4 reversible gate called PFAG gate with quantumcost of 8 is used to realize the circuit.

V. RESULTS

Simulation results are shown in Fig.6 and comparison proposed model with existing listed in table.3

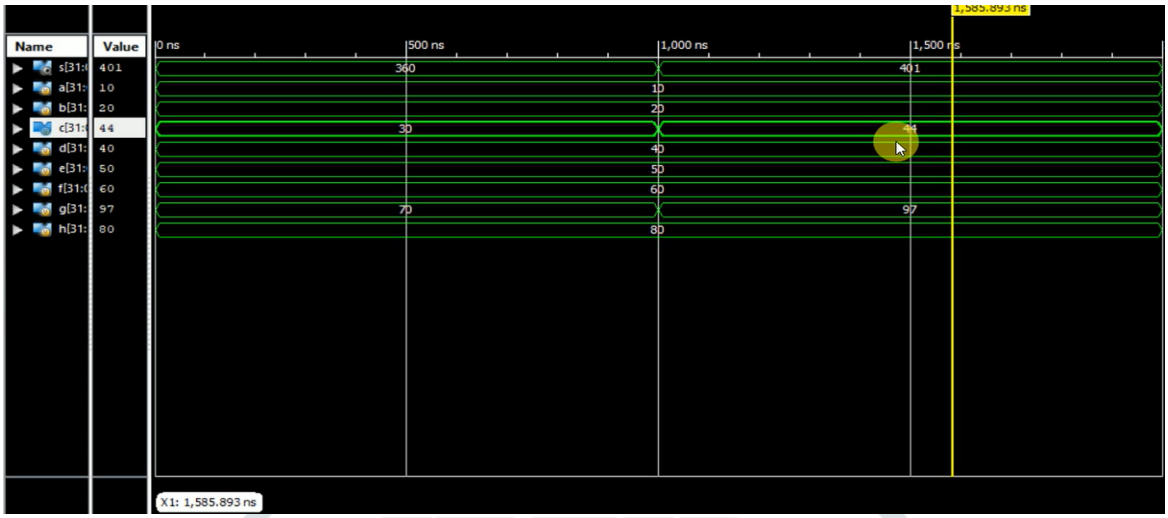

Fig.6: Proposed simulation result

Table.3 Comparison table for existing and proposed

	Existing work	Proposed work
AREA(LUT'S)	443	430
POWER(mw)	0.84	0.81
DELAY(ns)	62.102	24.024

CONCLUSION

In terms of tolerance for faults in reversible circuit designs, this thesis makes a wide range of contributions. Existing research on reversible fault tolerance mostly addresses fault testing. We have outlined the necessary steps and methods to implement fault tolerance in reversible logic in this paper. Probabilities of missing or inactive gate faults are used to calculate the voter circuit failure probability. Our team has come up with a series of equations that, given the likelihood of gate faults, may be used to determine the likelihood of circuit failure. After a certain amount of trials, our results show that the voter circuit failure probability surpasses 50% when the gate fault probabilities reach 0.003% or higher. Future inventors of reversible logic gates may find the simulation findings.

REFERENCES

[1] M. M. Islam, M. S. Hossain, M. K. Hasan, M. Shahjalal, and Y. M. Jang, "FPGA implementation of high-speed area-efficient processor for elliptic curve point multiplication over prime field," *IEEE Access*, vol. 7, pp. 178811–178826, 2019.

[2] Z. Liu, J. GroBschadl, Z. Hu, K. Jarvinen, H.Wang, and I. Verbauwhede, "Elliptic curve cryptography with efficiently computable endomorphisms and its hardware implementations for the Internet of Things," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 773–785, May 2017.

[3] Z. Liu, D. Liu, and X. Zou, "An efficient and flexible hardware implementation of the dual-field elliptic curve cryptographic processor," *IEEE Trans. Ind. Electron.*, vol. 64, no. 3, pp. 2353–2362, Mar. 2017.

[4] M. Haghparast, M. Mohammadi, K. Navi, M. Eshghi," Optimized Reversible Multiplier Circuit, Journal of Circuits, Systems and Computers", 18(2), 2009, pp 311-323.

[5] Lenin Gopal, Nor SyahirMohd Mahayadin, Adib kabir Chowdhury, Alpha Agape Gopalai, Ashutosh Kumar Singh, "Design and Synthesis of Reversible Arithmetic and Logic Unit", International Conference on Communication and Control Technology (I4CT 2014), 2014, pp 289-293.

[6] Mehrdad Khatir , Alireza Ejlali and Amir Moradi, " Improving the energy efficiency of reversible logic circuits by the combined use of adiabatic styles," Integration-the VLSI journal, Sciencedirect, Vol 44,pp 11-21, 2011.

[7] Huynh Tan Hoi, " Design and Analysis of Full Adders using Reversible Logic," Journal of VLSI Circuits and Systems, Vol 2, Issue 2, 2021.
[8] Sujata S.Chiwande and Pravin K.Dakhole, " VLSI Design of Power Efficient Carry Skip Adder using TSG & Fredkin Reversible gate," IEEE International Conference on Devices, Circuits and Systems (ICDCS), 2012.
[9] Sujata S.Chiwande and Dr.P.K.Dakhole, "Design and Analysis of Low Power Full Adder using Reversible logic," IEEE 6th International Conference on Electronics, Communication and Aerospace Technology(ICEAC), 2022.
[10] Gowthami.P and RVS Satyanarayana, "Design of Digital Adder Using Reversible Logic," International Journal of Engineering research and Applications, Vol 6, Feb, 2016. [11] Jagadeesh Pujar, Sithara Raveendran, Trilochan Panigrahi, Vasantha M.H and Nithin Kumar Y.B, "Design and Analysis of Energy Efficient Reversible Logic based Full Adder,"IEEE 62nd International Midwest Symposium on Circuits and Systems (MWSCAS), 2019.
[12] Jagannatha KB, D.Divya, Kavana S.Reddy, Pallavi K Desai and Sevanthi S, "ASIC Design of Reversible Full Adder Circuits," International Conference on Computing,Electronics and Electrical Technologies(ICCEET), 2012.
[13] Vivek Singh and K.B. Ramesh, "Introduction to Reversible Logic Gates and its Operations," International Journal for Research in Applied Science and Engineering Technology(IJRASET), Vol 10, Feb, 2022.
[14] Maheshwari.A, Gukan. V, Jeeva Anandh.R and Kumar A, "Modular Adder Design based on Reversible Toffoli CLA," IEEE 7th International Conference on Computing Methodologies and Communication(ICCMC)," 2023.
[15]Lihui Ni, Zhijin Guan and Wenying Zhu, " A General Method of Constructing the Reversible Full Adder," IEEE 3rd International Symposium on Intelligent Information Technology and Security Informatics," 2010.
[16] Kento Oonishi, Tomoki Tanaka, Shumpei Uno, Takahiko Satoh, Rodney Van Meter and Noboru Kunihiro, " Efficient Construction of a Control Modular Adder on a Carry-Lookahead Adder using Relative-phase Toffoli Gates," IEEE Transactions on Quantum Engineering, Vol 3, 2022.