© 2019 IJSRSET | Volume 6 | Issue 2 | Print ISSN: 2395-1990 | Online ISSN : 2394-4099
Themed Section : Engineering and Technology
DOI : https://doi.org/10.32628/IJSRSET196238

Implementation of Reversible Circuits in Modular Adders

A. K. Shaneel Kumar Reddy, M. Guru Sai, T. Bharath Reddy
ECE Department, Panimalar Institute of Technology, Chennai, Tamil Nadu, India

ABSTRACT

Reversible logic is a computing paradigm that has attracted significant attention in recent years due to its properties that lead to ultra-low power and reliable circuits. Reversible circuits are fundamental, for example, for quantum computing .Since addition is a fundamental operation, designing efficient adders is a cornerstone in the research of reversible circuits. Residue Number Systems (RNS) has been as a powerful tool toprovide parallel and fault-tolerant implementations of computations where additions and multiplications are dominant. In this paper, for the first time in the literature, we propose the combination of RNS and reversible logic. The parallelism of RNS is leveraged to increase the performance of reversible computational circuits. Being the most fundamental part in any RNS, in this work we propose the implementation of different adders, namely ripple carry adder, carry save adder, carry look ahead adder using reversible logic. Analysis and comparison with different adders for modular addition are designed using reversible gates with minimum overhead in comparison to regular Brunt Kung modulo-adders.

Keywords : Residue Number System, Reversible Circuits, Modular Adder, Parallel-Prefix Adder

I. INTRODUCTION

Researchers in academia and industry believe that Moore's law is ending, and even newly delivered deep-submicrontransistors are not significantly more efficient than their previous generations. Therefore, new computing paradigms should be investigated in order to overcome the predicted performance wall which will be reached in 2020. This rebooting of computing has to be based on novel methods at different computing levels of design abstraction, including arithmetic andcircuit level, in order to address the challenges of the emerging applications such as deep convolutional neural network (DNN) and internet-of-things (IOT). Residue Number System (RNS) is one unconventional number system that can provide fast and low-power implementation of additions and multiplications.RNS is a different approach of dealing and representing numbers that provide parallelism at arithmetic level. This number system has been applied to achieve parallel and efficient implementations for asymmetric cryptographic and digital signal processing (DSP). RNS is used now a days to achieve also energy-efficient and high-performance implementation of various emerging applications, such as deep neural networks communication networks and cloud storage. However, current implementations of RNS systems, on ASICs and FPGAs, are based on the CMOS technology, which is reaching its limit. Alternative methods and technologies, such as Nano electronic, are considered to be used. One of these alternatives is Reversible Computing (RC), which can provide ultra-low power computational circuits.In 1961 Rolf Ladner showed that the loss of information in a logical computing device resulted in KTlog2 joules of heat energy dissipation, where k is Boltzmann's constant and T the absolute temperature at which computation is performed. Later in 1973 C.H. Bennet showed that the dissipated energy is directly

correlated to the number of information bits lost and if the computation can be made reversible the energy dissipation would not occur. Reversible computation can be performed in a system only when the system comprises of reversible circuits.

II. RESIDUE NUMBER SYSTEM

A residue numeral system (RNS) is a numeral system representing integers by their values modulo several pair-wise co-prime integers called the moduli. This representation is allowed by the Chinese remainder theorem, which asserts that, if *N* is the product of the moduli, there is in an interval of length *N*, exactly one integer having any given set of modular values. The arithmetic of a residue numeral system is also called multi-modular arithmetic. A residue numeral system is defined by a set of *k* integers { m 1 , m 2 , m 3 , … , m k } , called the *moduli*, which are generally supposed to be pairwise coprime (that is, any two of them have a greatest common divisor equal to one), referred to as the *moduli*. Residue number systems have been defined for non-co prime moduli, but are not commonly used because of worse properties. Therefore, they will not be considered in the remainder of this article.

An integer *x* is represented in the residue numeral system by the set of its remainders

{ x 1 , x 2 , x 3 , … , x k }

Under Euclidean division by the moduli. That is x i = x mod m i , and 0 ≤ x i < m i for every *i* . The first step to architect a RNS is to select moduli set according to the target application constraints and requirements. The moduli set consists of pair-wise relatively prime numbers {*m*1, m2,*mn*}, being the dynamic range the sequence of integers that can be uniquely represented in RNS, i.e. [0, *M*-1] with M=*m*1×*m*2×…×*mn*. In order to decrease the complexity of hardware realization of RNS-based arithmetic, usually near power-of-two moduli are adopted, such as 2*n*-1, 2*n* and 2*n*+1.

Among these moduli, the simplest one to deal with is the 2*n*, which does not require any specific modular arithmetic, just the circuits for binary arithmetic. Apart from that, the most frequent co-prime number in moduli sets for RNS is 2*n*-1, since moduli 2*n*+1 is more complex and its representation requires on additional bit.

Typical RNS moduli sets are {2*n*-1, 2*n*+*k*, 2*n*+1}, {2*n*-1, 2*n*, 2*n*+1-1}, {2*n*-1, 2*n*, 2*n*+1, 22*n*+1-1}, {2*k*, 2*n*-1, 2*n*+1, 2*n*+1-1} and {2*n*-1, 2*n*, 2*n*+1, 2*n*+1-1, 2*n*-1-1}. The main arithmetic blocks of RNS are the forward converter, the modular arithmetic in the channels, and the reverse converter. The forward converter translates the weighted binary number (*X*) to the residues (*xi*'s), according to the moduli.

$$X \xrightarrow{Forward\ Conversion} (x_1, x_2, \ldots, x_n)$$

Where

$$x_i = X \bmod m_i = |X|_{m_i} \quad for\ i = 1 \ldots n$$

Note that *mod* indicates the remainder of de integer division of *X* by m_i. Then, considering two numbers *A* and *B* as follows:

$$A = (a_1, a_2, \ldots, a_n)$$
$$B = (b_1, b_2, \ldots, b_n)$$
$$S = A \bullet B = (s_1, s_2, \ldots, s_n)$$

where

$$s_i = |a_i \bullet b_i|_{m_i}, \bullet \in \{+, -, \times\}$$

Finally, a reverse converter maps the results in the RNS domain to the regular weighted representation, by using, for example, the Chinese remainder theorem (CRT). Other RNS operations such as sign detection, magnitude comparison and overflow handling are optional, according to the target application, and harder to perform in the RNS domain. It should be mentioned that general division cannot directly be performed in RNS, but division by a constant, one of the moduli of the set, i.e. scaling, is easier to perform.

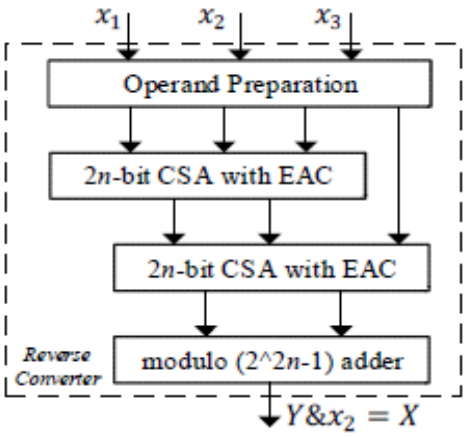


Fig 1. The full reverse converter for the moduli set $\{2n\text{-}1, 2n+k, 2n+1\}$

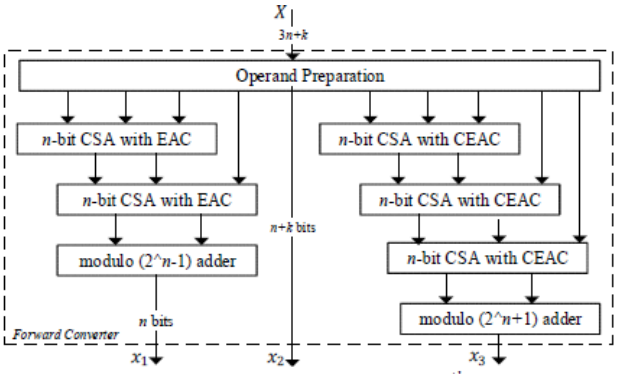


Fig 2. The forward converter for the moduli set $\{2n\text{-}1, 2n+k, 2n+1\}$

Most of the mentioned RNS operations are implemented using 3-to-2 carry-save adders (CSAs) with end-around carries (EACs) and 2-to-1 modular adders. A full hardware design of RNS with moduli set $\{2n\text{-}1, 2n+k, 2n+1\}$ is reported , and herein forward and reverse converters for this moduli set are depicted in Figs. 1 and 2, respectively. It can be observed that CSAs and carry-propagate modulo $2n$-1 adders are the components required to implement a full RNS architecture, since arithmetic in a channel also requires modulo adders and multipliers. Thus, to have an efficient modular adder is fundamental for RNS-based application.

III. MODULO-ADDITION BASED ON UNIVERSAL GATES

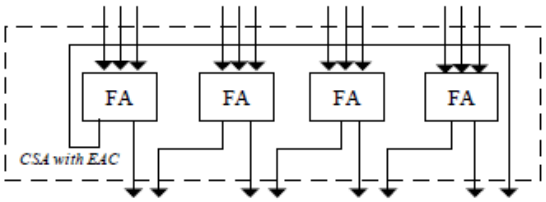


Fig 3. The 4-bit CSA with EAC structure

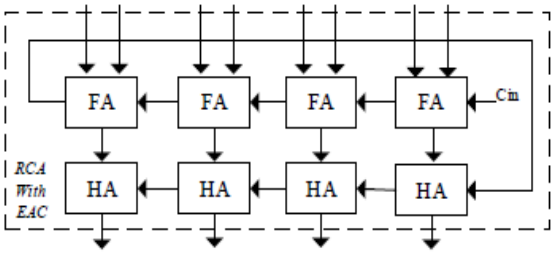


Fig 4. The 4-bit modulo [2^n-1] adder based on RCA method namely with EAC

The CSA with EAC consists of independent full adders (FAs) which just combine the three inputs into two carry-save output vectors, as shown in Fig. 3. Its delay is just the delay of a single FA, while the overall area linearly depends on the width of the operands. End-around carry is complemented. Modular carrypropagate adders can be designed based on different architectures, from low-cost ripple-carry adders (RCAs) (Fig. 4) to fast parallel-prefix adders (PPAs). The PPAs architectures can provide a good trade-off between circuit's parameters, being popular in RNS arithmetic circuits.Reversible computing provide Reliable and low power design, high performance circuits synchronous with speed and processing power. 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.

It again Improve computational efficiency this can be done by building circuits which reduce energy from state will save energy. Reversible computing will also lead to improvement in energy efficiency. It increases portability of device to reduce element size to atomic

size. It has incurred more hardware cost, but power cost and performance are dominant than hardware cost. Hence need of reversible computing cannot be ignored in computing era.

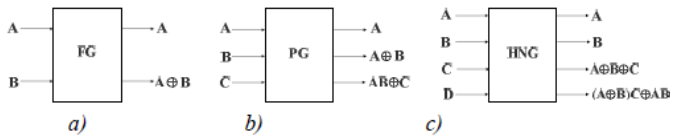


Fig 5. The reversible gates: a) Feynman; b) Peres and c) HNG

IV. MODULAR ADDER DESIGN USING REVERSIBLE CIRCUITS

The CSA is a 3-to-2 compression unit that is very popular for regular arithmetic as well as in RNS architectures due to its speed and cost. According to Fig. 3, a CSA can be built by using *n* FAs for adding three *n*-bit operands. According to, the HNG reversible gate can be used to realize a FA by setting the fourth input of HNG to the zero-logic level, as shown in Fig. 6.

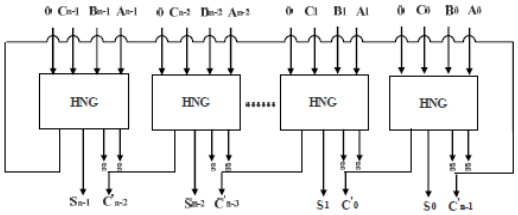


Fig 6. The CSA with EAC using HNG reversible gates

Requires *n* FAs and *n* HAs in the first and second levels, respectively. Similar to CSA, FAs can be realized with HNG gates. Besides, the Peres reversible gate can be used to implement a HA, where the third input bit is set to Furthermore, the total constant inputs and garbage outputs are 2*n* and 3*n*, respectively, since one of the inputs ofHNG and Peres gates is zero, and also two and one outputs of HNG and Peres gates, respectively, are not used

Similar to CSA, FAs can be realized with HNG gates. Besides, the Peres reversible gate canbe used to implement a HA, where the third input bit is set to zero, as shown in Fig. 7.

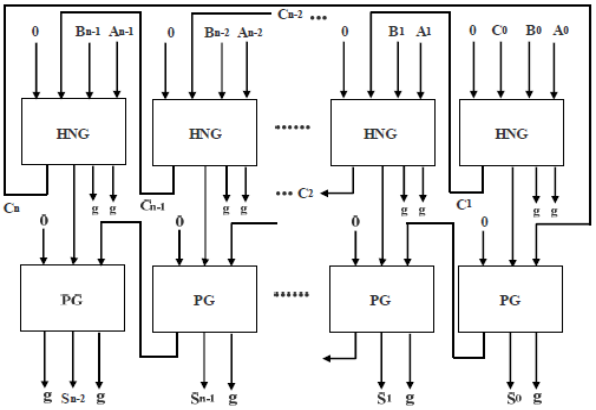


Fig 7. The RCA with EAC using HNG and Peres reversible gates

The heart of any PPA adder is a carry-computation network which consists of black and gray cells, as depicted in Fig. 8. There is different carry-computation networks which can be used for designing PPAs, The reversible implementation of different PPAs are presented in, and it is shown that the Brent-Kung adder has the least quantum cost among the prefix structures. Due to this, the Brent-Kung adder is also selected herein as the basis to design modulo 2*n*-1 adder with reversible gates.

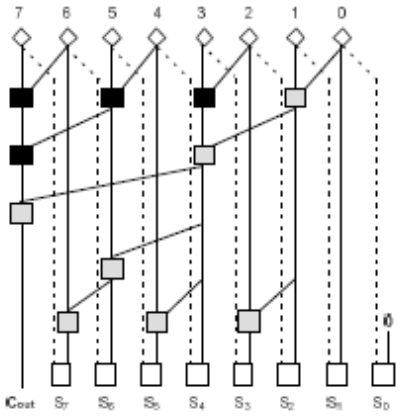


Fig 8. The regular parallel-prefix adder structure based on (*left side*) Brent- Kung

The regular Brent-Kung adder into a modulo 2*n*-1 adder, by inserting a row of black cells to add the carry-out, i.e. the end-around carry's as shown in Fig. 9.

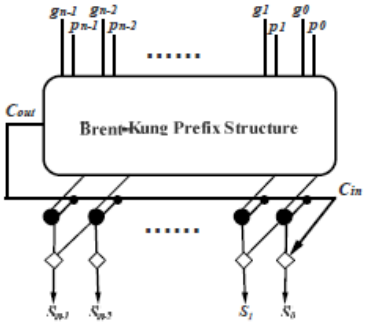


Fig 9. The modulo 2n-1 Brent-Kung prefix adder architecture

V. MODULO ADDERS WITH PROPOSED ADDERS (CLA, CARRY SELECT, AND CARRY SKIP)

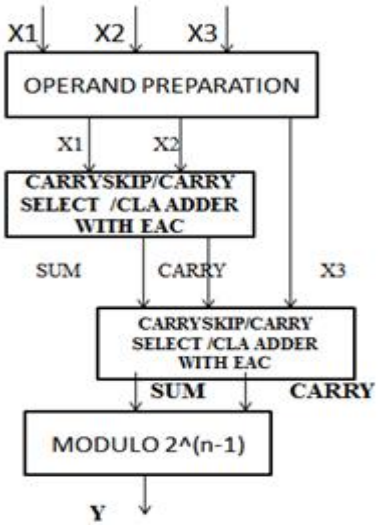


Fig 10. The full reverse converter moduli set

Further proposing an modulo addition based on normal gates and reversible logic implementing

- carry skip adder,
- Carry select adder and
- Carry look ahead adder comparing which is the best adder among them.

VI. CARRY SKIP ADDER IN MODULO ADDITION:

A Carry Skip Adder consists of a simple ripple carry adder with speed up carry chain called a skip chain. The chain defines the distribution of ripple carry blocks, which compose the skip carry blocks, which compose the skip adder.

Skip Carry adder is divided into blocks, where a special circuit detects quickly if all the bits to be added are different.Thus 4 carry skip adder blocks are made to get the 16 bit carry skip adder with normal gates.

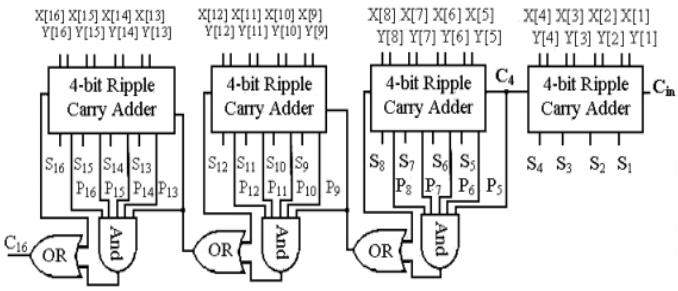


Fig 11. Carry skip adder

VII. CARRY SKIP ADDER WITH REVERSIBLE GATES:

In carry skip adder, there are 4 rca, in each rca there are 4 full adders, where in reversible gates, these full adders are replaced by HNG gate.

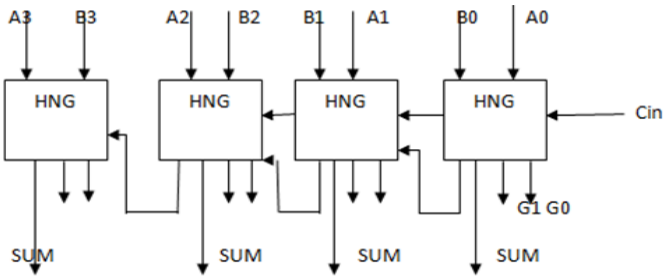
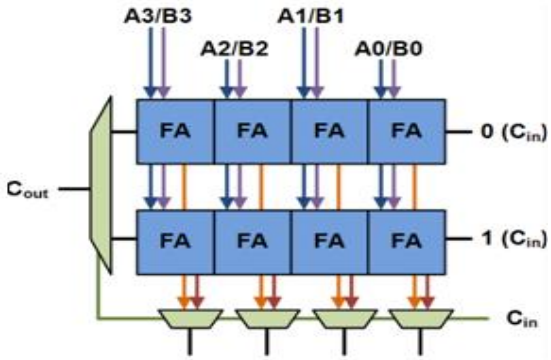


Fig 11. Reversible carry skip adder

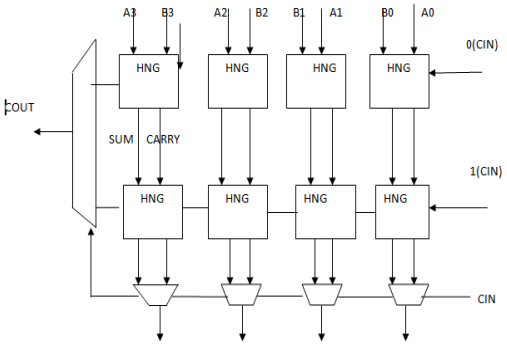
VIII. CARRY SELECT ADDER

The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders), in order to perform the calculation twice, one time with the assumption of the carry-in being zero and the other assuming it will be one. After the two results are calculated, the correct sum, as well as the correct carry-out, is then selected with the multiplexer once the correct carry-in is known.

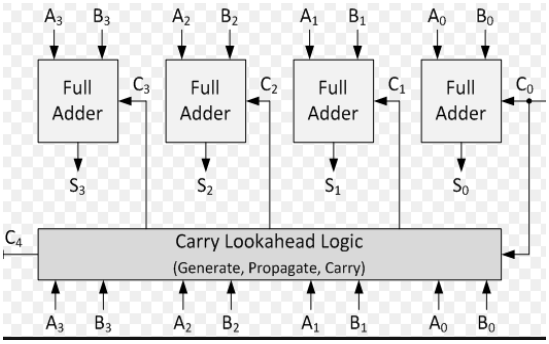


Carry select adder-block diagram

Carry select adder with reversible gates



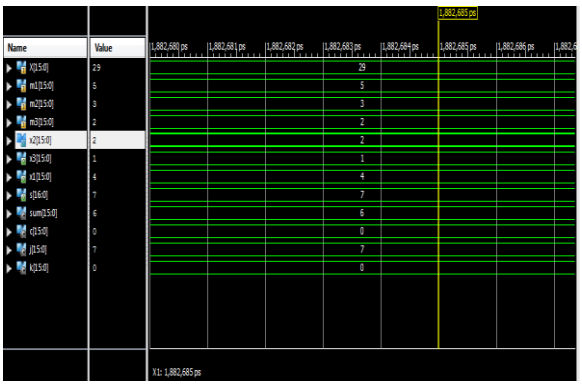
IX. IMPLEMENTATION OF CLA:



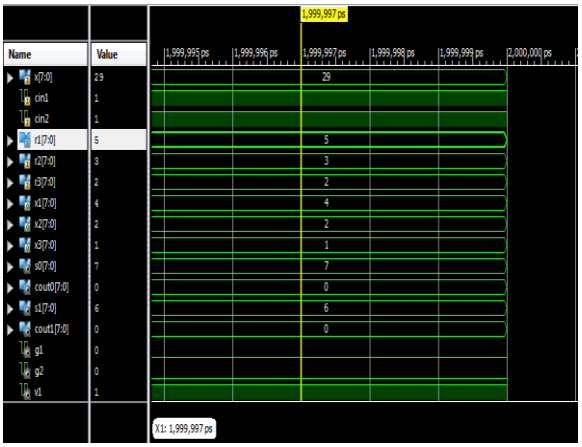
A **carry-look ahead adder** (CLA) or **fast adder** is a type of adder used in digital logic. A carry-look ahead adder improves speed by reducing the amount of time required to determine carry bits. It can be contrasted with the simpler, but usually slower, ripple-carry adder(RCA), for which the carry bit is calculated alongside the sum bit, and each stage must wait until the previous carry bit has been calculated to begin calculating its own sum bit and carry bit. The carry-look ahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger-value bits of the adder. The Kogge–Stone adder (KSA) and Brent–Kung adder (BKA) are examples of this type of adder.

X. RESULTS AND IMPLEMENTATION:

Modulo adder using normal gates



Modulo adder using reversible gates



DESCRIPTION	AREA ANALYSIS	POWER ANALYSIS	TIMING ANALYSIS
CARRYSKIP-UNIV	1300 LUT's	327.245 mW	64.178 ns
CARRYSKIP-REVERS	74 LUT's	324.01 mw	20.213.488 ns
CARRYSELECT-UNIV	1296 LUT's	328.20 mW	64.345 ns
CARRYSELECT-REVERS	1296 LUT's	325.29 mW	41.009 ns
CARRY LOOK AHEAD ADDER-UNIV	1283 LUT's	346.89mW	57.826ns
CARRY LOOK AHEAD ADDER-REVERS	29 LUT's	344.14mw	12.347ns

XI. CONCLUSION

From the result and analysis, we conclude that modulo addition using reversible gates consume low power than the universal gates and timing delay also

gets reduced by using reversible gates. And from the results, we prove that carry skip adder consumes less power in modulo addition. Thus future's VLSI deals with reversible gates.

XII.REFERENCES

[1]. T. M. Conte, E.P. DeBenedictis, P.A. Gargini, and E. Track, "RebootingComputing: The Road Ahead," Computer, vol. 50, no. 1, pp. 20-29, 2017.

[2]. M. Alioto (Ed.), Enabling the Internet of Things: From IntegratedCircuits to Integrated Systems, Springer , 2017.

[3]. A.S.Molahosseini, L.Sousa and C.H. Chang (Eds.), Embedded Systems Design with Special Arithmetic and Number Systems,Springer, 2017.

[4]. C. H. Chang, A. S. Molahosseini, A. A. EmraniZarandi, and T. F. Tay, "Residue Number Systems : A New Paradigm to Datapath Optimizationfor Low-Power and High-Performance Digital Signal Processing Applications," IEEE Circuits and Systems Magazine, vol. 15, no. 4, pp. 26-44, 2015.

[5]. L. Sousa, S. Antão, and P. Martins, "Combining Residue Arithmetic toDesign Efficient Cryptographic Circuits and Systems," IEEE Circuits andSystems Magazine, vol. 16, no. 4, pp. 6-32, 2016.

Cite this article as :

A. K. Shaneel Kumar Reddy, M. Guru Sai, T. Bharath Reddy, "Implementation of Reversible Circuits in Modular Adders", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), ISSN : 2456-3307, Volume 6 Issue 2, pp. 135-141, March-April 2019. Available at doi : https://doi.org/10.32628/IJSRSET196238

Journal URL : http://ijsrset.com/IJSRSET196238