ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse156952020.pdf
https://doi.org/10.30534/ijatcse/2020/156952020

Design and Testing of 16 bit Carry Save Adder using Reconfigurable LFSR

Kiran Kumar Mandrumaka[1], M. Mounika[2], Fazal Noorbasha[3]
[1,2]Department of ECE, Anurag University, Hyderabd, Telangana, India
[3]Department of ECE,Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, India
[1,2]E-mail: kirankumarece@anurag.edu.in
[3]E-mail: fazalnoorbasha@kluniversity.

ABSTRACT

In this paper, the design and testing of Carry Save Adder (CSA) using Reconfigurable LFSR is implemented. The ever increasing applications of integrated circuits in the day-to-day useful electronic gadgets are the driving force for the development of low power designs of configurable hardware designs. High speed and low power are the main parameters that are targeted by modern circuit designers. Among the fastest increasing applications, the audio and video signal processing applications are growing at a very high rate. Mobile applications have increased the technological improvements for digital signal processing applications. Arithmetic functions are the very important logical operational unit of any processing unit in digital signal processing applications. The speed of arithmetical computations and performance is among the parameters of any digital hardware design for efficiency improvement. Self-testing ability is another essential aspect of hardware design. This feature offers hardware durability primarily for hardware applications that can be configured. The built-in-self test (BIST) feature helps in the quick diagnosis of the hardware functional authenticity. This paper introduces a carrying save adder with BIST included in it. The work is developed using the Verilog HDL language and implemented in Xilinx Vivado. The concept of self-test design requires a low power test pattern generator (TPG).

Key words: BIST, Linear Feedback Shift Register, Carry Save Adder, Xilinx Vivado

1. INTRODUCTION

Nowadays, processing speed and power consumption can be used to calculate configurable hardware performance. The FPGA is one of the configurable devices for the desired and promising performance of hardware-based on power and speed. In FPGA, the operational execution is based on a combination of hardware resource architecture switching in the internal current direction. The skill-based change in the operating circuit architecture can achieve a hardware-based optimization of any type. A low power system has advantages such as portability of the device, long lifetime, good results, etc. A high-speed processor with low power demand design is the fundamental criteria for modern digital applications. A CSA design is the most important design for digital signal processors. The adder is used in most of the complex data processing applications. Another characteristic needed for self-diagnosis in hardware is the self-testing feature. This function helps to verify the configurable integrated circuit hardware and helps to move the hardware resource to the integrated device in the event of a hardware defect. During self-testing, the hardware is tested using complementary hardware for its practical performance. A basic BIST block diagram is shown in Fig 1. A logic circuit here is a design that is an integrated circuit system design functional component. It carries out a given logical operation on the DATA input in normal operating mode. A random data sequence is generated by the test pattern generator using the BIST Controller control signal when run in self-test mode. The logic circuit performs this test series and compares the generated output of the logic operation with the current output. If the results of the logic function against the test inputs do not match the real output, the output of the comparator shows logic high.

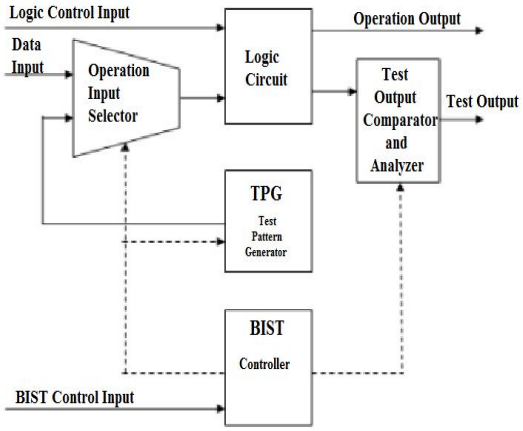

Figure 1. Simple Block Diagram of BIST Design

This condition indicates a fault in the logic circuit hardware. In such cases, configurable hardware re-locates the circuit resources within the integrated circuit to avoid the faulty hardware. It indicates a failure in the hardware of the logic circuit. In such cases, a hardware configuration locates the resources of the circuit inside an integrated circuit to avoid the defective hardware. This design is based on a single pattern of input changes produced by a counter and a grey-code converter to decrease the switching activity. This design

features high speed, low power consumption, and is particularly good for processors where random numbers are needed for uniform distribution. The design of the low power LFSR was carried out based on a low power test pattern generator [3]. This design focuses mainly on how BIST produces test vectors and how power consumption is reduced. This paper reduces the transition by generating the gray-code at a distance of 1-bit. The 16-bit BBS and LFSR PN Sequence Generators were implemented by FPGA [4]. Where the PN generator logic is change the speed in LFSR or by modifying the key used in BBS. The design of a random sequence number with FPGA based N-bit LFSR is proposed in [5].

A simulation study on TPG based 16th-grade primitive polynomials using the Shift Register [6]. This study focuses on a comparative study of various types of LFSR applications for irreducible or primitive polynomials in the 16th degree. In [8], a pseudo-random number generation is presented with the use of the WELL and re-seeding technique. A random number is first generated in this document using the WELL method and its output was analyzed based on Vedic mathematics. The Vedic high-speed multiplier with LFSR is presented on an FPGA-based implementation [9]. The reset method is used to stop the repeat pattern. Several types of research are also carried out on logic operating units using FPGA devices for high-speed applications. A commentary on digital signal processing operations is available in Vedic Mathematics [10]. The TPG is the main element in the design of BIST hardware. Most researchers propose and simulate BIST application circuits to offer power and speed optimized designs based on FPGA implementation. The FPGA implementation for real-time interface applications of BIST enabled UART [11] shows how different UART blocks are verified. A concurrent BIST architecture [12] is proposed for the design of online input vector monitors. This paper aims to observe a group of vectors reaching the inputs of a circuit during regular operation and to use an SRAM architecture to store the relative locations of the vectors reaching the inputs of the circuit. The implementation of the I2C protocol on the FPGA is given in [13], enabled by BIST.

This design allows the self-test of a typical high-speed communication system hardware interface protocol. Low power circuit implementation of BIST-based FPGA logic circuits is the prerequisite for current hardware designs to achieve high speed operating circuits [14-15]. The advanced Low Power BIST [14] architecture and March Intra Word Coupling Fault algorithm BDS orientated. This paper discusses read defects with classic errors and increases design efficiency and testing time for detecting defects. The implementation of low power BIST is taken into account critically. As the test design to implement FPGA with self-test performance, a multiplier with two 4-bit inputs is taken. A low-performance generator design is used to provide the auto-test functionality. An updated architecture is used to formulate the testing pattern by decreasing the number of sequential components compared to the earlier design [16-17]. The present paper is organized as follows: Section-II describes the design of the Test Pattern Generator and CSA that are implemented in the part of this work. Section-III presents simulation and synthesis results. Section-IV presents the conclusion drawn based on the performed design. Finally, the references are mentioned.

2. TEST PATTERN GENERATOR

This work presents the BIST-based approach to carry save adder implementation using configurable hardware. As a test logic design in the present work a 16-bit low power carry save adder design is used. The design of the CSA is carried out using the architectural gate level representation for low power hardware construction. A combination of a gate level is used to create a full-adder and a half-adder style. For the BIST implementation, a test pattern generator with random output value is required. For TPG realization, a low-power modified design of linear-feedback-shift-register (LFSR) is used in this design implementation [18-20].

3. ARCHITECTURE OF LFSR

LFSR is actually a shift register in which each clock signal transfers the data between the registers [18]. Here are some parts of the shift register XORed to create a linear feedback to drive the LFSR entrance bit. Taps are the selected bits that affect the input bit of the prior states. The feedback loop is the XORed chain of such taps depending on the feedback polynomial. A standard degree 'n' polynomial over GF(2) describes an n-bit LFSR internal structure where coefficients h_n imply that a feedback direction exists.
Characteristic polynomial is given by m(x)

$$m(x) = 1 + h_1x_1 + h_2x_2 + \ldots + h_{n-1}x_{n-1} + h_nx_n \quad (1)$$

Based on the feedback polynomial given, LFSR can generate various pseudo random sequences. LFSR arithmetic operations in the polynomial mod 2 field are achieved. This finite field is called Galois field referred to as GF field (p), where p is the number of elements in the field. GF (2) is the binary field over which LFSR performs all arithmetic operations. Any additional operation is thus equal to XOR, and every multiplication takes place as AND. The characteristic polynomial coefficients are either 0 or 1, which demonstrates the existence or absence of a tap location.

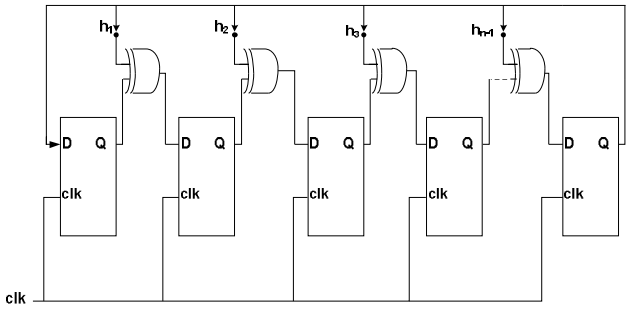

Figure 2. N-bit Modular LFSR Architecture

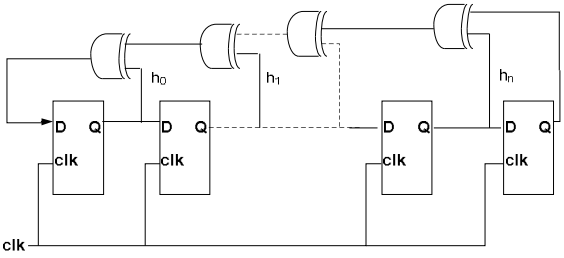

Figure 3. N-bit Standard LFSR architecture

For example, in the feedback polynomial

$$x16 + x14 + x13 + x11 + 1 \quad (2)$$

The term, one represents the input to first flip-flop, and powers of x terms represents the tapped bits. Those polynomials that could divide $1 + x_T$ completely but does not divide any other polynomials of the form 1 + xi are called primitive polynomials, where i<1. The significance of primitive polynomials is that maximum length sequences of T = 2n – 1 can be generated. So these polynomials are fed to LFSR in logic BIST applications. Due to the capability to generate random patterns and because of error detection and correction properties LFSR is commonly used to design the main components in LBIST architecture. Different types of LFSR includes, i) Standard LFSR ii) Modular LFSR iii) Complete LFSR iv) Hybrid LFSR.

A. ***Standard LFSR***: Standard LFSR also called as Fibonacci LFSR or external XOR. LFSR is a shift register XOR gates representing the tap positions of the feedback polynomials are concatenated to produce a new output bit. This single bit output is given as feedback input to the last flip-flop in the structure. Fig.3 shows an N-bit Standard LFSR

B. ***Modular LFSR***: Modular LFSR called as Galois, internal XORs, or one-tomany LFSR is another form LFSR generating the same patterns as that of conventional one. The only difference lies in the connection of taps in feedback polynomials to the flip-flops. The XOR gates are connected immediately after the flip-flop depending on tap positions.

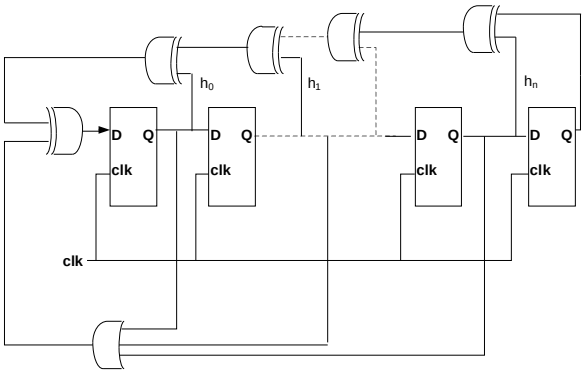

Figure 4. Architecture of Complete LFSR

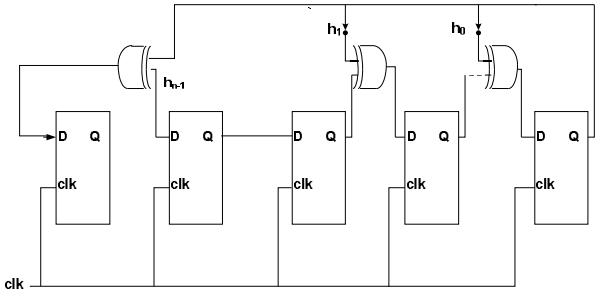

Figure 5. Architecture of Hybrid LFSR

The performance of the gates is fed to the next flip flop immediately. The benefit of this arrangement over the standard type is that the critical path between the two taps includes a limit of one XOR gate compared to another in which all XOR gates are used, depending on how many taps. Thus modular LFSR is faster than standard since XOR operations can be performed one word at a time in relation to the execution. This makes Galois LFSR's software implementation effective. Fig.2 is showing an LFSR module N-bit.

C. Complete LFSR: All LFSR are designed to generate a maximum length sequence of upto 2n -1only. Complete form of architecture is shown in Fig. 4. The presence of an all-zero sate can move LFSR into a locked-state condition. Modified form of LFSR that also include an all zero state is called complete LFSR. Complete LFSR can be constructed from standard LFSR by inserting an XOR gate into the last stage of the LFSR, and a NOR gate with n-1 inputs is used as a zero-detector. Thus a complete LFSR could generate 2n patterns for a primitive polynomial feedback LFSR [18].

D. Hybrid LFSR: A polynomial over GF (2) f(x) = 1+ m(x) + q(x), be said to be fully decomposable if both m(x) and q(x) have no common terms and there exists an integer j such that $q(x) = x_j\, m(x)$, where j<1. Then the corresponding polynomial can be expressed as

$$f(x) = 1 + m(x) + x_j\, m(x) \quad (3)$$

Then a (hybrid) top–bottom LFSR can be constructed using the connection polynomial

$$p(x) = 1 + {}^{x_j} + x_j m(x) \quad (4)$$

When ^xj specifies that the XOR gate is connected to the feedback path with a single input taken from the LFSR j_{th} stage output rather than between stages. Fig.5 shows the Hybrid LFSR architecture. Compared to the modular and standard design, the benefit of the hybrid LFSR design is that the number of XOR gates needed can be reduced to (m-1)/2 hybrid. It makes use of a feedback structure from j_{th} output stage. From the first to $(j-1)_{th}$ registers a modular structure is used. Hybrid LFSR makes small in area, number of gates and speed more efficient.

4. CARRY SAVE ADDER

In Carry Save Adder (CSA), three bits are added parallelly at a time. In this scheme, the carry is not propagated through the stages. Instead, carry is stored in present stage, and updated as addend value in the next stage [2]. Hence, the delay due to the carry is reduced in this scheme. The architecture of CSA is shown in Fig. 6.

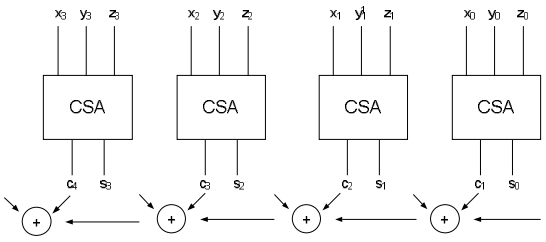

Figure.6 Architecture of CSA

5. SIMULATION AND SYNTHESIS RESULTS

The BIST based adder design in the proposed work is implemented using Verilog Hardware Description Language on Xilinx ISE Tool. The design is simulated for functional performance and power consumption. The simulation wave forms is shown in below Figure.7 and Figure.8

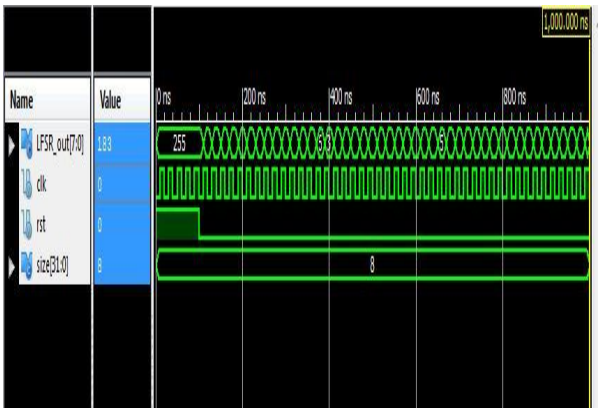

Figure.7 Simulation Result of 8-bit Reconfigurable Hybrid LFSR

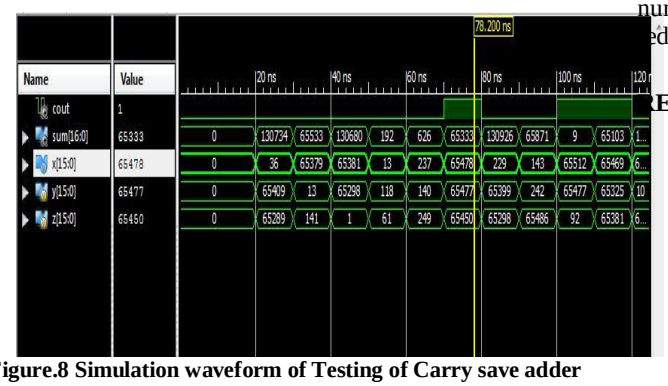

Figure.8 Simulation waveform of Testing of Carry save adder using Reconfigurable Hybrid LFSR

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 1.473ns (Maximum Frequency: 678.771MHz)
   Minimum input arrival time before clock: 2.335ns
   Maximum output required time after clock: 3.668ns
   Maximum combinational path delay: No path found
```

Figure 9: Timing report of Carry save adder using Reconfigurable Hybrid LFSR

```
Device utilization summary:
---------------------------

Selected Device : 6slx4tqg144-3


Slice Logic Utilization:
 Number of Slice Registers:                4  out of   4800     0%
 Number of Slice LUTs:                     1  out of   2400     0%
    Number used as Logic:                  1  out of   2400     0%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:       5
   Number with an unused Flip Flop:        1  out of      5    20%
   Number with an unused LUT:              4  out of      5    80%
   Number of fully used LUT-FF pairs:      0  out of      5     0%
   Number of unique control sets:          1

IO Utilization:
 Number of IOs:                            6
 Number of bonded IOBs:                    6  out of    102     5%

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:                 1  out of     16     6%
```

Figure 10: Area report of Carry save adder using Reconfigurable Hybrid LFSR

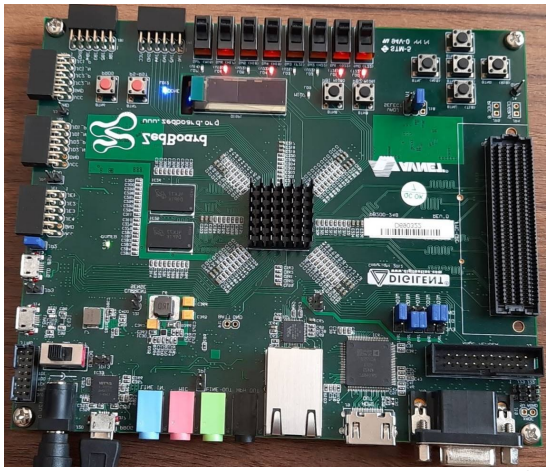

Figure 13: Output on FPGA-zynq board with the result of CSA

6. CONCLUSION

The design and testing of Carry Save Adder (CSA) using Reconfigurable LFSR is implemented on FPGA zynq board. The corresponding test results are shown in the results section. The test patterns are generated with hybrid LFSR and successfully tested the 16 bit carry save adder with less number of LUT's compared with complete LFSR. Delay also reduced with proposed hybrid LFSR.

REFERENCES

[1] A. Kavitha, G. Seetharaman, T. N. Prabhakar and Shrinithi, "Design of Low Power TPG using LP-LFSR", International Conference on Intelligent Systems Modelling and Simulation, IEEE Computer Society, pp.334-338, 2012.

[2] Md. Fokhrul Islam, M. A. Mohd. Ali and Burhanuddin Yeop Majlis, "FPGA Implementation of an LFSR based Pseudorandom Pattern Generator for MEMS Testing", International Journal of Computer Applications, Volume-75 No-11, pp.30-34, August 2013.

[3] Syed. Mujeeb Raheman and M. Basha, "Low Power Linear Feedback Shift Register based Low Power Test Pattern Generator", International Journal and Magazine of Engineering, Technology, Management and Research, Volume-2 Issue-6, pp.579-584, June 2015.

[4] Mohammad Moinuddin and Sadgunn Kumari V, "FPGA implementation of 16-bit BBS and LFSR PN Sequence Generator", International Journal of Scientific Engineering and Technology Research, Volume-4 Issue-31, pp.6082-6084, August 2015.

[5] Babitha P. K., Thushara T. and Dechakka M.P., "FPGA based N-bit LFSR to generate random sequence number", International Journal of Engineering Research and General Science, Volume-3 Issue-3, pp.6-10, May-June 2015.

[6] Swapna S. and Sunilkumar S. Manvi, "Review of LP-TPG using LPLFSR for Switching Activities", International Journal of Advanced Research in Computer Science and Software Engineering, Volume-5 Issue-2, pp.157-160, February 2015.

[7] Mirella Amelia Mioc, "A simulation study of TPG using Shift Register based on 16th Degree Primitive

Polynomials", New Developments in Pure and Applied Mathematics, pp.363-369, ISBN: 978-1-61804-287-3.

[8] V. Divya Bharathi and Arivasanth M., "Generation of Pseudo-Random number by using Well and Re-seeding method", International Journal of Advanced Research in Electronics and Communication Engineering, Volume-4 Issue-3, pp.604-609, March 2015.

[9] Kaustubh M. Gaikwad and Mahesh S. Chavan, "Vedic Mathematics for Digital Signal Processing Operations: A Review", International Journal of Computer Applications, Volume-113 N0-18, pp.10-14, March 2015.

[10] Neha Shaktawat and Ghanshyam Jangid, "FPGA Implementation of High Speed 16-Bits Vedic Multiplier using LFSR", International Journal of Computer Science and Mobile Computing, Volume-4 Issue-8, pp.102- 108, August 2015.

[11] L. Supriya, J. Lingaiah and G. Kalyan, "FPGA Implementation of BIST (Built-in-Self-Test) Enabled UART for Real Time Interface Applications", International Journal of Science, Engineering and Technology Research, Volume-4 Issue-7, pp.2645-2647, July 2015.

[12] S. Abirami, Nikitha S. Paulin, S. Prabhu Venkateshwaran, "A concurrent BIST architecture for online input vector monitoring", International conference on Science, Technology and Management, pp.1411-1488, Feb 2015.

[13] Jayashri H. M. and Shilpa K., "Design and Implementation of I2C with BIST Technique on FPGA", International Journal of Research in Information Technology, Volume-3 Issue-5, pp.428-438, May 2015.

[14] Siva Priya G., Hari Kishore K., Noorbasha F. (2019), 'Static timing analysis and timing violations of sequential circuits', International Journal of Innovative Technology and Exploring Engineering, 8(0), PP.115-121.

[15] Praveen Blessington T., Bhaskara B., Noor Basha F. (2018) , 'Efficient analysis for power modeling on routing topologies in three-dimensional network on chip architectures',Journal of Advanced Research in Dynamical and Control Systems, 10 (4 Special Issue),PP. 1377- 1383

[16] Praveen Blessington, Bhaskara B, Fazal Noorbasha, "Estimation of latency and throughput for three-dimensional network-on-chip architecture" Journal of Advanced Research in Dynamical and Control Systems,Volume 10, Issue 7 Special Issue, 2018, Pages 1353-1359.

[17] Rooban S., Saifuddin S., Leelamadhuri S., Waajeed S. (2019), 'Design of fir filter using wallace tree multiplier with kogge-stone adder', International Journal of Innovative Technology and Exploring Engineering, 8(6), PP.92-96.

[18] K. Hari Kishore, Fazl Noorbasha, Katta Sandeep, D.N.V. Bhupesh, SK. Khadar Imran, K.Sowmya "Linear convolution using UT Vedic multiplier" International Journal of Engineering and Technology(UAE), ISSN No: 2227-524X, Vol No: 7, issue no:2.8, Page No:409-418, March 2018.

[19] Sk.Md.Reyaan, B.Manoj Venkat, G.Y.V.Siva Shankar, K Hari Kishore, Fazal Noorbasha, Y Archana, Shaik Razia "Power Analysis of FIR Filter Design Using APC-OMS Algorithm" Journal of Advanced Research in Dynamical and Control Systems, ISSN No: 1943-023X, Vol No: 12, Issue No: 2, Page No: 1085-1092, March 2020

[20] Soumya N., Sai Kumar K., Raghava Rao K., Rooban S., Sampath Kuma R P., Santhosh Kumar G.N. (2019), '4-bit multiplier design using cmos gates in electric VLSI', International Journal of Recent Technology and Engineering, 8(2), PP.1172-1177