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 VLSI Architecture of Three-Operand Binary Adder

[1]Pappu Deepesh, [2]Pavurala Chaitanya, [3]Pathan Iklas Khan,[4]Mungara V Sasidhar Yadav, [5]Dr.K. Murali

[1]UGScholar, [2]UGScholar, [3]UGScholar, [4]UGScholar, [5]Professor
[12345] Department of Electronics and Communication Engineering (ECE),
[12345]Narayana Engineering College, Nellore, India

Abstract: The main objective Arithmetic and logic unit has been the most significant unit in any electronic devices. In the recent advancement, for an arithmetic and logic unit to be significant it needs to have an efficient algorithmic operation such as Multiplications and addition. Three-operand binary adder is the basic functional unit to perform the modular arithmetic in various cryptography and pseudorandom bit generator (PRBG) algorithms. Carry save adder (CS3A) is the widely used technique to perform the three-operand addition, Ladner-Fischer adder (LFA) and Han-Carlson adder (HCA). Analysis of the adder performance Moreover, In the proposed Three-operand adder that significantly reduces the critical path delay. Hence, a new high-speed adder architecture is proposed using pre-compute bitwise addition followed by KOGGE-STONE adder to perform the three-operand binary addition that consumes substantially low power and drastically reduces the adder delay .

IndexTerms **– XILINX ISE , VLSI , Verilog , FPGA**

I. INTRODUCTION

VLSI systems are constantly developing and a greater number of transistors are being incorporated in a single IC every two years so that they can perform better and fast. As the number of bits a system can handle increases, delay increases and performance parameters like area and power are in turn affected. Optimizing area, delay and power of VLSI systems have always been a challenge. The thing as a designer can do is to bring about any change in the performance parameters of any individual component the digital system comprises so that it can have an overall improvement in performance. One such individual component is the adders which are used in digital system which can be used to perform complex addition. Multiplication, subtraction and division which can be done with the help of a simple adder unit. Hybrid adders are combination of two or more adders of same or different type, a hybrid adder can be designed in such a way that the individual adder comprised within the adder overcomes the limitation of the other adder used, say, a combination of two adders such that one adder overcomes the limitation of other adder and prove to be advantageous in complex computations. Another type of high-speed adders is the parallel prefix adders, they can perform addition in a parallel fashion (compute the bits parallelly) making them one among the high-speed adders. A combination of these highspeed adders can bring about an unimaginable improvement in the performance of digital systems in terms of operating speed, frequency of operation, delay etc. Investigating combinations of different adders and comparing their performance parameters is the main goal of this research work.An adder is a digital circuit that performs addition of numbers. In many computers and other kinds of processors adders are used in the arithmetic logic units or ALU.

They are also used in other parts of the processor, where they are used to calculate addresses, table indices, increment and decrement operators, and similar operations.

II RELATED WORK

THREE-OPERAND BINARY ADDER TECHNIQUES

The three-operand binary addition is one of the critical arithmetic operation in the congruential modular arithmetic architectures and LCG-based PRBG methods . It can be implemented either by using two stages of two-operand adders or one stage of three-operand adder. Carry-save adder (CSA) is the commonly used technique to perform the three-operand binary addition . It computes the addition of three operands in two stages. The first stage is the array of full adders. Each full adder computes “carry” bit and “sum” bit concurrently from three binary input ai, bi and ci . The second stage is the ripple-carry

adder that computes the final n-bit size "sum" and one-bit size "carry-out" signals at the output of three-operand addition. The "carry-out" signal is propagated through the n number of full adders in the ripple-carry stage. Therefore, the delay increases linearly with the increase of bit length. The architecture of the three-operand carry-save adder is shown in Fig. 1 and the critical path delay is highlighted with a dashed line. It shows that the critical path delay depends on the carry propagation delay of ripple carry stage and is evaluated as follows,

TCS3A = (n+1)TFA = 3TX + 2nTG

Similarly, the total area is evaluated as follows,

ACS3A = 2nAFA = 4nAX + 6nAG

Here, AG and TG indicate the area and propagation delay of basic 2-input gate (AND/OR/NAND/NOR) respectively. AX and TX indicate the area and propagation delay of 2-input XOR gate respectively. The major drawback of the CS3A is the larger critical path delay which increases with an increase of bit length. This critical propagation path delay influences the overall latency of the congruential modular arithmetic based cryptography and PRBG architectures, where three-operand adder is the primary component.

Hence, to shorten the critical path delay, two stages of parallel prefix two-operand adder can also be used. In literature, parallel prefix or logarithmic prefix adders are the fastest twooperand adder techniques. These adder techniques have six different topologies, such as Brent-Kung, Sklansky, Knowles, Ladner-Fischer, Kogge-Stone (KS) and Han-Carlson (HC). Among these, Han-Carlson is the fastest one when bit size increases (i.e. n > 16). In recent years, various such kind of parallel prefix two-operand adders, i.e., Ling, Jackson-Talwar, ultra-fast adder, hybrid PPFCSL and hybrid Han-Carlson are also discussed in the literature. The ultra-fast adder is reported as the fastest one, and it is even faster than the Han-Carlson by three gates delay. However, it consumes comparatively two times large gate area than the Han-Carlson adder. On the other hand, the hybrid Han-Carlson adder is designed with two Brent-Kung stages each at the beginning and the end, and with Kogge-Stone stages in the middle. This resultant a slightly higher delay (two gates delay) than the HanCarlson adder, with a 10% to 18% reduction in the gate complexity. Essentially, the Han-Carlson adder provides a reasonably good speed at low gate complexity as compared to other existing two-operand adder techniques. It has the lowest area delay product (ADP) and power-delay product (PDP) among all. Thus, the three-operand addition can be performed using Han-Carlson adder (HCA) in two stages, as shown in Fig. The detailed architecture of HCA-based three-operand adder (HC3A) is presented . The maximum combinational path delay of HC3A depends on the propagate chain, i.e. the number of black-grey cell stage in the PG logic of Han-Carlson adder and is evaluated as follows,

The HCA-based three-operand binary adder (HC3A) greatly reduces the critical path delay in comparison with the three-operand carry-save binary adder. However, the area increases with increase of bit length in the order of O(n log2 n). Therefore, to minimize this trade-off between area and delay, a new high-speed, area-efficient three-operand adder technique and its efficient VLSI architecture is proposed in the next section.

II. PROPOSED SYSTEM

The proposed VLSI architecture of the three-operand binary adder and its internal structure is shown in Figure. The new adder technique performs the addition of three n-bit binary inputs in four different stages. In the first stage (bit-addition logic), the bitwise addition of three n-bit binary input operands is performed with the array of full adders, and each full adder computes "sum (Si)" and "carry (cyi)" signals as highlighted in Figure. The logical expressions for computing sum (Si) and carry (cyi) signals are defined in Stage-1, and the logical diagram of the bit-addition logic is shown in Fig. 3(b).

In the first stage, the output signal "sum (Si)" bit of current full adder and the output signal "carry" bit of its right-adjacent full adder are used together to compute the generate (Gi) and propagate (Pi) signals in the second stage (base logic).The computation of Gi and Pi signals are represented by the "squared saltire-cell" as shown in Figure and there are n +1 number of saltire-cells in the base logic stage. The logic diagram of the saltire-cell is shown in Figure, and it is realized by the following logical expression,

$$G_{i:i} = G_i = S_i' \cdot cy_{i-1};$$
$$P_{i:i} = P_i = S_i' \oplus cy_{i-1}$$

The external carry-input signal (Cin) is also taken into consideration for three-operand addition in the proposed adder technique. This additional carry-input signal (Cin) is taken as input to base logic while computing the G0 (S0 · Cin) in the first saltire-cell of the base logic. The third stage is the carry computation stage called "generate and propagate logic" (PG) to pre-compute the carry bit and is the combination of black and grey cell logics. The logical diagram of black and grey cell is shown in Figure that computes the carry generate Gi: j and propagate Pi: j signals with the following logical expression,

Gi: j = Gi:k + Pi:k · Gk−1: j,

Pi: j = Pi:k · Pk−1: j

kogge-stone Adder:

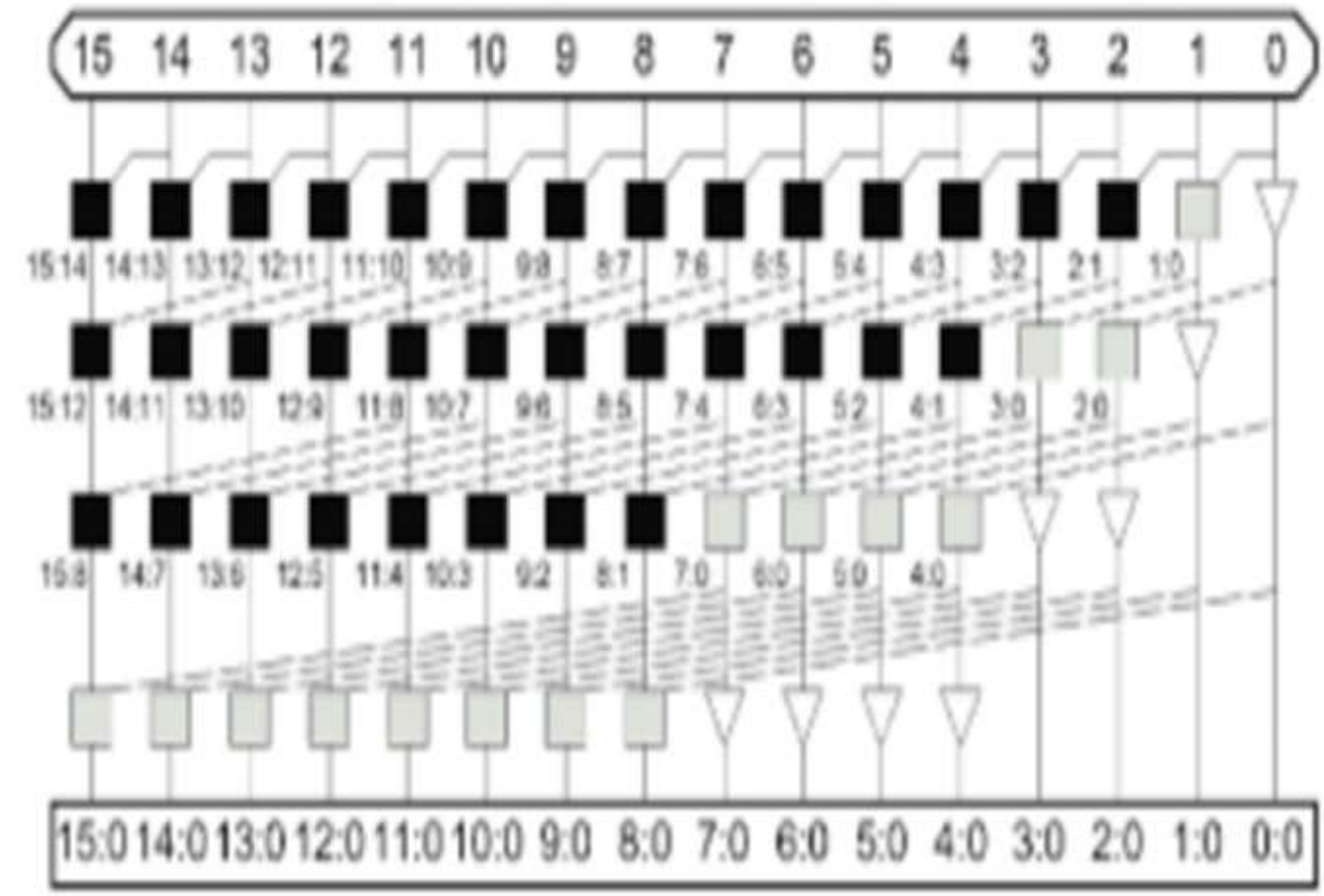
The kogge-stone adder is a parallel prefix adder, known for its regular structure and efficient carry propagation mechanism. It is designed to enable high-speed addition of binary numbers by utilizing parallelism in the computation. The adder is suitable for applications where speed is a critical factor, such as in digital signal processing (DSP) circuits.

The addition process involves following three stages:

1.Pre-processing stage.

2.Carry generation network.

3.Post-processing stage.



Design Simulation:

Verifying Functionality using Behavioral Simulation Create a test bench waveform containing input stimulus you can use to verify the functionality of the counter module. The test bench waveform is a graphical view of a test bench. Create the test bench waveform as follows:

1. Select the counter HDL file in the Sources window.
2. Create a new test bench source by selecting Project → New Source.
3. In the New Source Wizard, select Test Bench WaveForm as the source type, and type counter_tbw in the File Name field.
4. Click Next.
5. The Associated Source page shows that you are associating the test bench waveform with the source file counter. Click Next.
6. The Summary page shows that the source will be added to the project, and it displays the source directory, type, and name. Click Finish.
7. You need to set the clock frequency, setup time and output delay times in the Initialize Timing dialog box before the test bench waveform editing window opens. The requirements for this design are the following:

♦ The counter must operate correctly with an input clock frequency = 25 MHz.

♦ The DIRECTION input will be valid 10 ns before the rising edge of CLOCK

♦ The output (COUNT_OUT) must be valid 10 ns after the rising edge of CLOCK. The design requirements correspond with the values below. Fill in the fields in the Initialize Timing dialog box with the following information:

♦ Clock High Time: 20 ns.

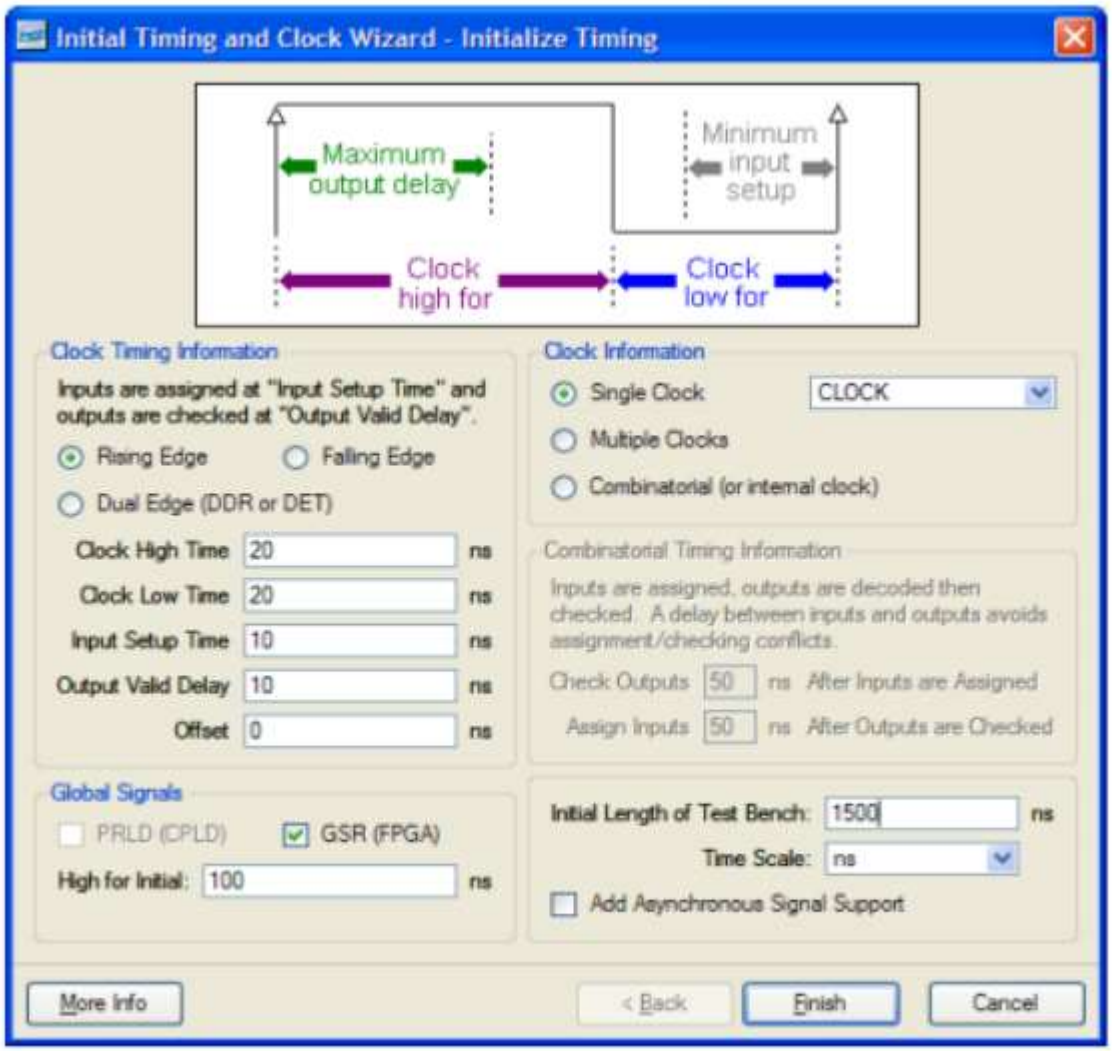
♦ Clock Low Time: 20 ns.

♦ Input Setup Time: 10 ns.

♦ Output Valid Delay: 10 ns. ♦ Offset: 0 ns.

♦ Global Signals: GSR (FPGA) Note: When GSR(FPGA) is enabled, 100 ns. is added to the Offset value automatically.

♦ Initial Length of Test Bench: 1500 ns.



Click Finish to complete the timing initialization. 9. The blue shaded areas that precede the rising edge of the CLOCK correspond to the Input Setup Time in the Initialize Timing dialog box. Toggle the DIRECTION port to define the input stimulus for the counter design as follows:

♦ Click on the blue cell at approximately the 300 ns to assert DIRECTION high so that the counter will count up.

♦ Click on the blue cell at approximately the 900 ns to assert DIRECTION low so that the counter will count down.

III. RESULTS

The below figure-1, is the RTL Schematic diagram of Three Operand Adder for Low Power Application. Here a, b, care the inputs of the Three Operand Adder and S is the outputs.

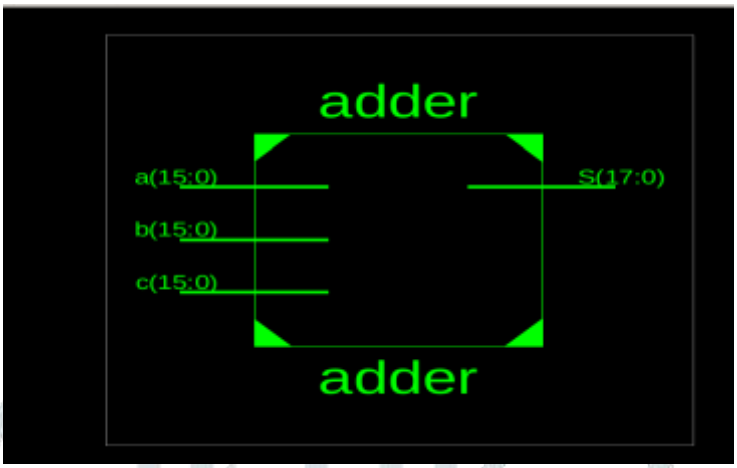


Fig 1 Schematic Diagrams Three Operand Adder for Low Power Application.

FIG: 2 RTL Internal diagram of Three Operand Adder

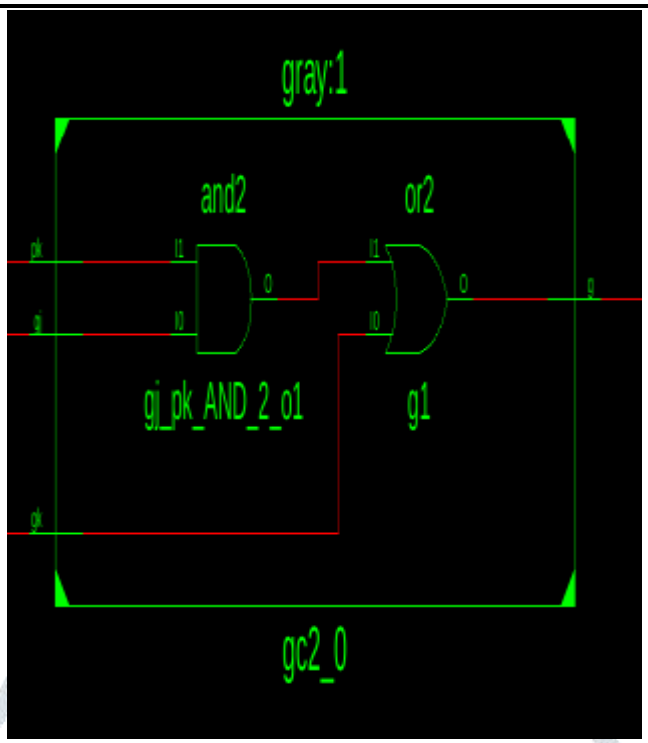


FIG: 3 RTL Internal diagram of Gray cell

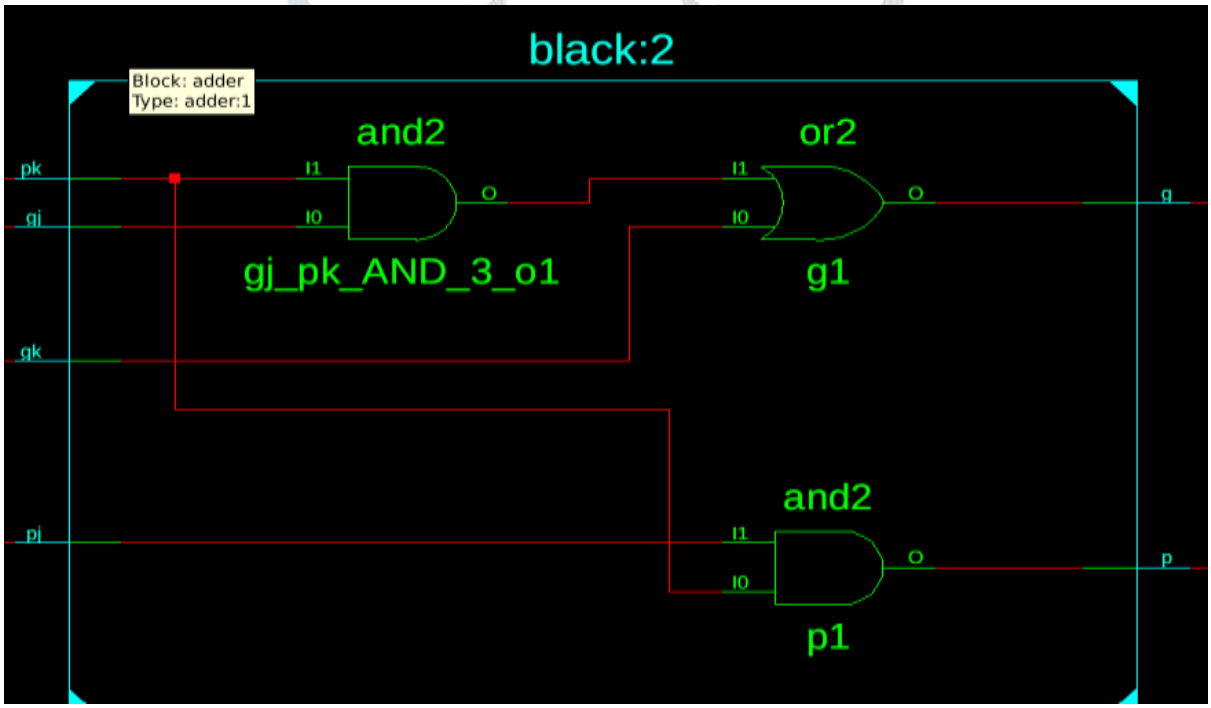


FIG: 4 RTL Internal diagram of Black Cell.

SIMULATION RESULT OF Three Operand Adder for Low Power Application.

FIG:5 Simulation result of Three Operand Adder for Low Power Application.

Fig 5 shows the output waveform of proposed adder where we performed addition of 16 bit input data of b c and gives the output is s.

IV CONCLUSIONS

The high-speed VLSI architecture for the three-operand binary adder demonstrates superior performance in terms of speed and area efficiency, making it suitable for applications requiring rapid arithmetic operations in compact hardware. With its optimized power consumption and reduced circuit complexity, the proposed design offers a promising solution for high-performance computing systems.

VII 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] B. Parhami, Computer Arithmetic: Algorithms and Hardware Design. New York, NY, USA: Oxford Univ. Press, 2000.

[5] P. L. Montgomery, "Modular multiplication without trial division," Math. Comput., vol. 44, no. 170, pp. 519–521, Apr. 1985.

[6] S.-R. Kuang, K.-Y. Wu, and R.-Y. Lu, "Low-cost high-performance VLSI architecture for montgomery modular multiplication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 2, pp. 434–443, Feb. 2016.

[7] S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, "Energy-efficient high-throughput montgomery modular multipliers for RSA cryptosystems," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 11, pp. 1999–2009, Nov. 2013.

[8] S. S. Erdem, T. Yanik, and A. Celebi, "A general digit-serial architecture for montgomery modular multiplication," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 5, pp. 1658–1668, May 2017.

[9] R. S. Katti and S. K. Srinivasan, "Efficient hardware implementation of a new pseudo-random bit sequence generator," in Proc. IEEE Int. Symp. Circuits Syst., Taipei, Taiwan, May 2009, pp. 1393–1396.

[10] A. K. Panda and K. C. Ray, "Modified dual-CLCG method and its VLSI architecture for pseudorandom bit generation," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 3, pp. 989–1002, Mar. 2019.

[11] A. Kumar Panda and K. Chandra Ray, "A coupled variable input LCG method and its VLSI architecture for pseudorandom bit generation," IEEE Trans. Instrum. Meas., vol. 69, no. 4, pp. 1011–1019, Apr. 2020.

[12] N. Weste and K. Eshraghian, Principles of CMOS VLSI Design—A Systems Perspective. Reading, MA, USA: Addison-Wesley, 1985.

[13] T. Kim, W. Jao, and S. Tjiang, "Circuit optimization using carry-saveadder cells," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 17, no. 10, pp. 974–984, Oct. 1998.

[14] A. Rezai and P. Keshavarzi, "High-throughput modular multiplication and exponentiation algorithms using multibit-scan–multibit-shift technique," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 9, pp. 1710–1719, Sep. 2015.