International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 03 | Mar 2021 www.irjet.net p-ISSN: 2395-0072

AN EFFICIENT FIR FILTER DESIGN USING THREE-OPERAND BINARY ADDER

Ms. D. SATHYAKALA[1], Ms. M. MATHUMATHI[2], Ms. T. DHEIVA PRIYA[3], Ms. S. DEEPIKA[4], Ms. K. KAVIYASRI[5]

[1]Assistant Professor, Dhanalakshmi Srinivasan Engineering College (Autonomous)
[2,3,4,5]Dhanalakshmi Srinivasan Engineering College (Autonomous)

Abstract — *FIR filter is very important part in digital word because of its characteristics such as linearity and stability. An area efficiency of the FIR filters can be achieved by reducing the count of either multiplier unit or adder unit since they are the basic building blocks of the FIR filter. This paper represents the VLSI architecture of the FIR filter using the three operand binary adder techniques. The basic functional unit in modular arithmetic is Three-operand binary adder. New architecture is proposed to improve speed and area-efficient adder. To perform three-operand binary addition, pre-compute bitwise addition and followed by carry- prefix computation logic also used. It is extensively reduces area, power and delay. Also, we can achieve the lowest ADP and PDP. Hence we can reduce the size and power consumption of the FIR filter can be reduced by using the three operand binary adder.*

***Index Terms*—FIR filter, Three-operand adder, carry save adder (CSA), Modular Arithmetic.**

1. INTRODUCTION

Finite impulse response (FIR) filters are used in Digital Signal Processing applications. Accuracy in filter Designing is based on the Multiplication and accumulation of filter coefficients. Filters are digital filter whose response to the unit filter (Unit Sample Function) is finite in duration. This is in contrast to Infinite impulse response (IIR) filters whose response to unit impulse is infinite in duration FIR filter can be implemented using either recursive or non-recursive techniques, but usually non recursive technique are used

FIR filter is Finite IR filter and IIR filter is Infinite IR filter. FIR filters are non-recursive. That is, there is no feedback involved. Whereas an IIR filter is recursive. There is feedback involved .The impulse response of an FIR filter will eventually reach zero. The impulse response of an IIR filter may very well keep "ringing" ad- infinitum. IIR filters may be designed to accurately simulate "classical" analog filter responses whereas FIR filters, in general, cannot do this. FIR filter has linear phase and easily control whereas IIR filter has no particular phase and difficult to control FIR filter is stable and IIR filter is unstable .FIR filter depend only on I/P whereas IIR filter depend upon both I/P and O/p .FIR filter consist of only zeroes and IIR filter consist of poles & zeroeos.

2. EXISTING SYSTEM

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 multiplicationand 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 modular arithmetic operation. The most efficient approach to implement the modular multiplication and exponentiation is based on three-operand binary addition.

The three-operand binary addition can be carried out either by using two two-operand adders or one three-operand adder. Carry-save adder (CS3A) is the area-efficient and widely adopted technique to perform the three-operand binary addition in the modular arithmetic used in cryptography algorithms and PRBG methods However, the longer carry propagation delay in the ripple-carry stage of CS3A seriously influences the performance of the MDCLCG and other cryptography architectures on IoT based hardware devices. In order to shorten the critical path delay, a parallel prefixed two-operand adder such as Han-Carlson (HCA) can also be used for three-operand binary addition. It reduces the critical path delay in the order of $O(\log_2 n)$ but increases the area in the order of $O(n \log_2 n)$.

Therefore, it is necessary to develop an efficient VLSI architecture to carry out the fast three-operand binary addition with minimum hardware resources. Hence, a new high-speed area-efficient adder technique is proposed using pre-compute bitwise addition followed by carry-prefix computation logic to perform the three-operand addition in this paper that consumes considerably

less gate area while minimizing the propagation delay in comparison to the HCA-based three-operand adder (HC3A). Furthermore, the proposed adder architecture is implemented with the Verilog HDL, and then synthesized with commercial available 32nm CMOS technology library. Also, the area-delay and power-delay products of the proposed adder technique are measured and compared with respect to the existing CS3A and HC3A three-operand adder techniques.

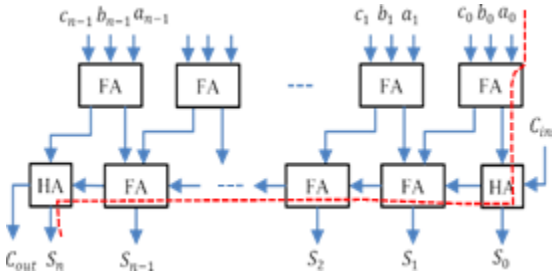


Fig. No.1 Carry Save Adder

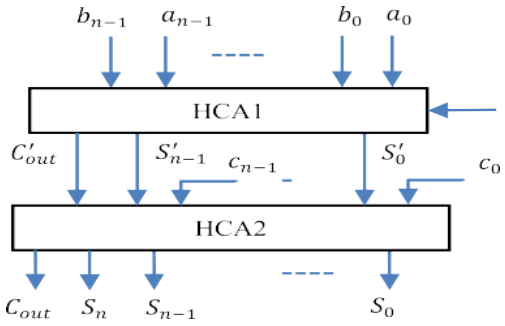


Fig. No. 2 HCA-based three-operand adder (HC3A)

3. SYSTEM ANALYSIS

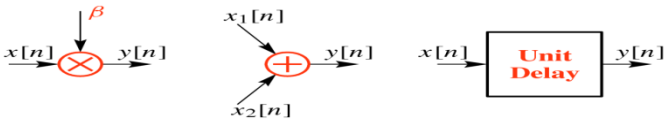
The goal of our project is to create a 9 tap FIR filter using three operand binary adders in modelsim. The goal is to get familiar with the tool chain and create the necessary components using Xilinx's Core Generator. The equation to implement a FIR filter is given as:

$y(n)=x(n)*h(n)= \sum_{k=0}^{l-1} hk]x[n-1]$

A. Implementation of FIR Filters

The implementation of an FIR requires three basic building blocks

- Multiplication
- Addition
- Delay



Multiplier: y[n]=βx[n]

In a DSP system the multiplier must be fast and must have sufficient precision to support the desired application A high quality filter will in general require more multiplications than one of lesser quality, so throughput suffers if the multiplier is not fast, There are classes of filters that do not require multiplies. FIR filters having 50 coefficients or more are not that uncommon.

Adder: y[n]=x1[n]+x2[n]

Signal addition is a very basic DSP function. In an FIR filter additions are required in combination with multiplications, hence DSP microprocessors feature multiply- accumulate (MAC) units Adders generally operate with just two inputs at a time

Unit Delay: y[n]=x[n-1]

The unit delay provides a one sample signal delay .A sample value is stored in a memory slot for one sample clock cycle, and then made available as an input to the next processing stage. An M-unit delay requires M memory cells (note each memory cell must store say B-bits) configured as a shift register (B-bits wide)

B. Proposed Three Operand adder Architecture:

This section presents a new adder technique and its VLSI architecture to perform the three-operand addition in modular arithmetic. The proposed adder technique is a parallel prefix adder. However, it has four-stage structures instead three-stage structures in prefix adder to compute the addition of three binary input operands such as bit-addition logic, base logic, PG (propagate and generate) logic and sum logic. The logical expression of all these four stages are defined as follows,

Stage-1: Bit Addition Logic:

$$S_i^r = a_i \oplus b_i \oplus c_i,$$

$$cy_i = a_i \cdot b_i + b_i \cdot c_i + c_i \cdot a_i$$

Stage-2: Base Logic:

$$Gi{:}i = Gi = S_I^r \cdot cy_{i-1},\ G0{:}0 = G0 = S_0^r \cdot Cin$$

$Pi{:}i = Pi = S_i^r \oplus cy_{i-1}$, $P0{:}0 = P0 = S_0^r \oplus Cin$

Stage-3: PG (Generate and Propagate) Logic:

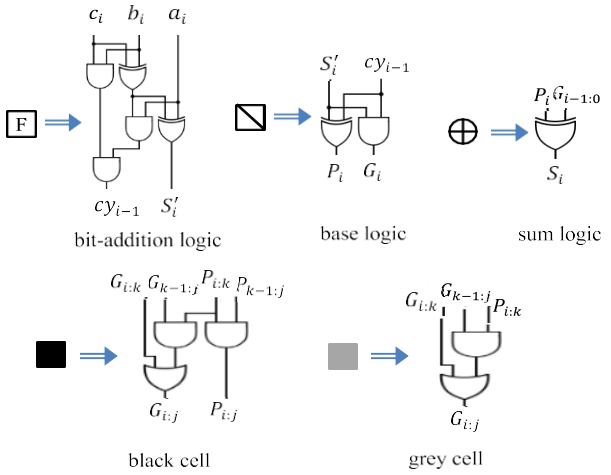
$$G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j},$$

$$P_{i:j} = P_{i:k} \cdot P_{k-1:j}$$

Stage-4: Sum Logic:

$$S_i = (P_i \oplus G_{i-1:0}), \quad S_0 = P_0, \; C_{out} = G_{n:0}$$

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 (S_i^r)" and "carry (cy_i)" signals as highlighted in. The logical expressions for computing sum (S_i^r) In the first stage, the output signal "sum (S_i^r)" bit of current full adder and the output signal "carry" bit of its right-adjacent full adder are used together to compute the generate (G_i) and propagate (P_i) signals in the second stage (base logic). The computation of G_i and P_i signals are represented by the "squared saltire-cell" and there are n 1 number of saltire-cells in the base logic stage. The logic diagram of the saltire-cell is shown in and it is realized by the following logical expression,



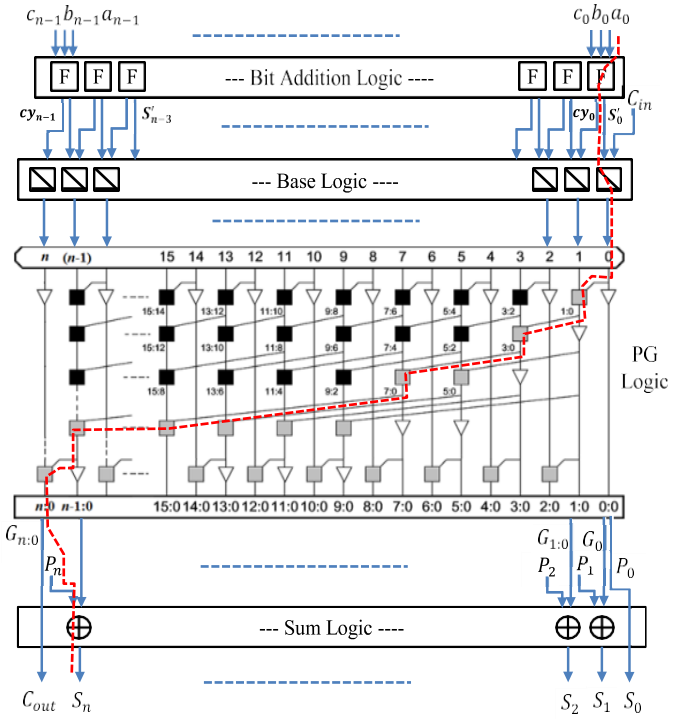


Fig. No. 3. Different stages of Three Operand adder Architecture

$Gi{:}i = G_i = S_i^r \; cy_{i-1}; Pi{:}i = P_i = S_i^r \oplus cy_{i-1}$

The external carry-input signal (C_{in}) is also taken into consideration for three-operand addition in the proposed adder technique. This additional carry-input signal (C_{in}) is taken as input to base logic while computing the G_0 $(S_0^r \; C_{in})$ 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 andis the combination of black and grey cell logics. The logical diagram of black and grey cell is shown in Fig. 3(b) that computes the carry generate $G_{i\,j}$ and propagate $P_{i\,j}$ signals with the following logical expression,

$$G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j},$$

$$P_{i:j} = P_{i:k} \cdot P_{k-1:j}$$

C. Physical Synthesis results of the Proposed Adder

For the fair comparison, the same coding style using Verilog HDL is adopted for designing the HHC3A, CS3A and HC3A and the proposed three-operand adders. Further, all these designs are synthesized using Synopsys Design Compiler in same SAED 32nm CMOS technology

library to obtain the core area, timing and power for different word size.

The physical synthesis analysis metrics comprised of maximum combinational gate delay, core area, power consumption, Area-Delay-Product (ADP) and Power-Delay-Product (PDP). The ratio entries are considered as 1.00 for the proposed adder architecture, and simple divisions obtain the rest of ratios. It is observed that the proposed adder is 3.12, 5.31 and 9.28 times faster than the CS3A adder for 32-, 64- and 128- bit operand size respectively.

Similarly, it is 1.54, 1.57 and 1.60 times faster than the HC3A adder for the same bit lengths. Moreover, it saves the core area of 21.2%, 24.1% and 26.6% as compared to the HC3A adder architecture for 32-, 64- and 128- bit word size respectively. In addition to this, it consumes 25.3%, 26.8%, and 28.2% less power as compared to the HC3A for 32-, 64- and 128- bit architectures.

However, it is slower by 12% to 15% than the proposed adder for same word lengths. It is worth mentioning that both the HHC3A and proposed adder architectures have significantly less Area-Delay-Product (ADP) and Power-Delay-Product (PDP) compared to the existing three-operand adder techniques. The proposed adder has 55.1%, 71.6% and 82.7% reduction on ADP over CS3A adder for 32-, 64- and 128- bit architectures respectively. Similarly, it has 48.9%, 51.7% and 54.2% reduction on ADP over HC3A adder for the same bit lengths. Moreover, it has also reported a similar amount of reduction on PDP over CS3A and HC3A adder architectures.

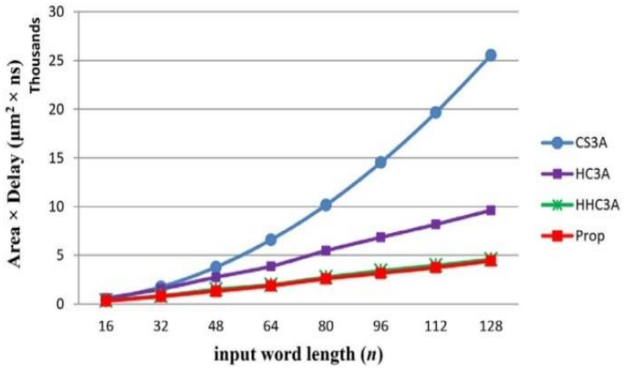


Fig. No. 4 Area-Delay-Product

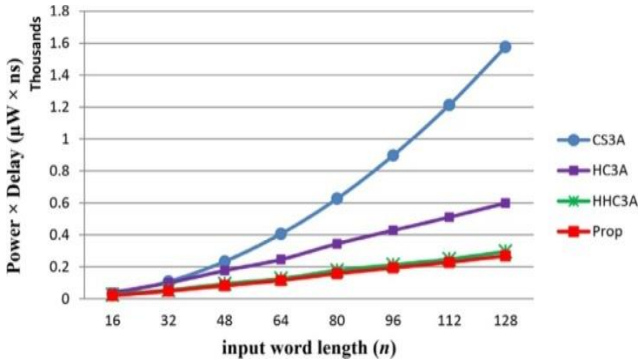


Fig. No. 5 Power-Delay-Product

D. Area and Timing complexity

The structure of the proposed adder consists of four main stages, namely bit-addition, base, PG and sum logics. The adder performance mainly depends on the number of prefix stages of PG logic. Therefore, the maximum propagation gate delay (T_{prop}) of the proposed three-operand adder architecture can be evaluated as follows:

$$T_{prop} = T_{bitadd} + T_{base} + T_{PG} + T_{sum}$$

$$\approx 2T_X T_X 2 \log n^r 1 T_G T_X$$

$$\approx 4T_X + 2 \log_2 n^r + 1 T_G$$

Similarly, the hardware area (A_{prop}) of the proposed three- operand adder can be estimated as:

$$A_{prop} = A_{bitadd} + A_{base} + A_{PG} + A_{sum}$$

$$\approx (2nA_X + 3nA_G) + (n + 1)(A_X + A_G)$$

$$+ 2n + 3s - 3\times 2^s + 3 A_G + nA_X$$

$$A_{prop} \approx (4n + 1)A_X + 6n + 3s \; \underline{'n'} - 3 \times 2^s + 4 A_G$$

Here, $s \log_2 n$ 1 and $n^r n$ 1.

The first order timing-area complexity of this hybrid Han-Carlson three-operand adder.

(HHC3A) is evaluated as follows:

$$T_{HHC3A} = T_{bitadd} + T_{base} + T_{PG} + T_{sum}$$

$$2T_X T_X 2 \log n^* 2 T_G T_X$$

$$\approx 4T_X + 2 \log_2 n^* + 2 T_G$$

$$A_{HHC3A} = A_{bitadd} + A_{base} + A_{PG} + A_{sum}$$

$$\approx (2nA_X + 3nA_G) + (n + 1)(A_X + Ag)$$

It reports that the proposed adder architecture saves 43.4%, 50.2% and 55.6% A_G gate area than HC3A adder architecture for 32-, 64- and 128- bit operand size respectively while reducing the critical delay. Moreover, the critical delay of the proposed adder is significantly reduced, i.e., 81.2%, 89.1% and 93.7% as compared to the CS3A adder for 32-, 64- and 128- bit size. It is also observed that the hybrid Han-Carlson based HHC3A adder.

E. Proposed FIR Filter Using Three Operand Adder

Finite impulse response (FIR) filters are used in Digital Signal Processing applications. Accuracy in filter Designing is based on the Multiplication and accumulation of filter coefficients. Filters are digital filter whose response to the unit filter (Unit Sample Function) is finite in duration. This is in contrast to Infinite impulse response (IIR) filters whose response to unit impulse is infinite in duration FIR filter can be implemented using either recursive or non-recursive techniques, but usually non recursive technique are used.

FIR filter is Finite IR filter and IIR filter is Infinite IR filter. FIR filters are non-recursive. That is, there is no feedback involved. Whereas an IIR filter is recursive. There is feedback involved .The impulse response of an FIR filter will eventually reach zero.

The impulse response of an IIR filter may very well keep "ringing" ad- infinitum. IIR filters may be designed to accurately simulate "classical" analog filter responses whereas FIR filters, in general, cannot do this. FIR filter has linear phase and easily control whereas IIR filter has no particular phase and difficult to control FIR filter is stable and IIR filter is unstable .FIR filter depend only on I/P whereas IIR filter depend upon both I/P and O/p .FIR filter consist of only zeroes and IIR filter consist of both poles and zeroes.

In order to reduce area, we will reduce the no of adders used in the finite impulse response filter, instead of using the normal adder we can use the proposed three operand adder in the FIR filter, here we are using the 9 tap FIR filter. By using the three operand binary adder we can reduce the area of the adder used in the circuit

In modelsim we will include all the three operand adder file and then compile the proposed FIR filter, simulate the file provide the input values from 0 to 8 because we are using 9 tap FIR filters give clock value as 1 and reset value as 0 press the RESET button for each value we get output.

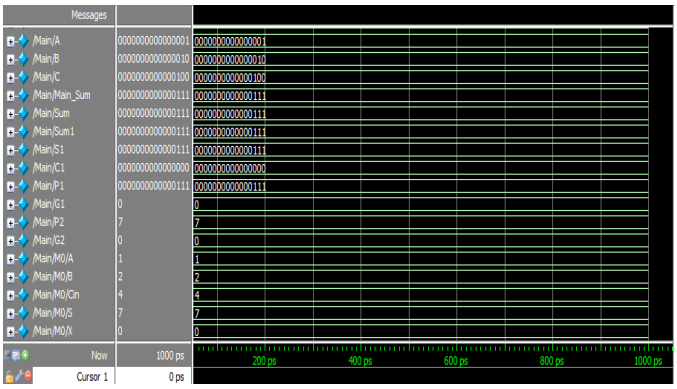


Fig. No. 6 Proposed Three operand Adder waveform

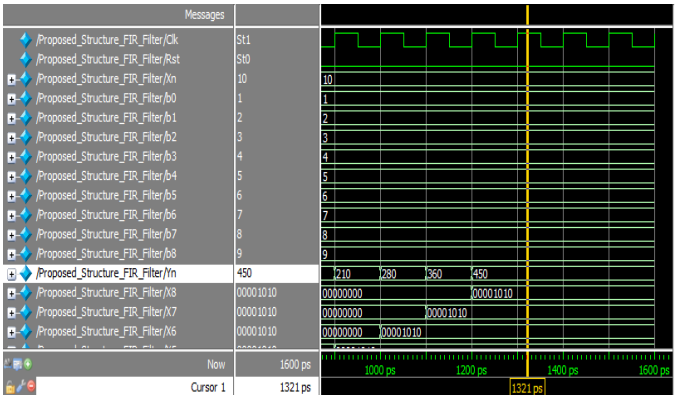


Fig. No. 7 Proposed FIR filter using Three operand Adder waveform

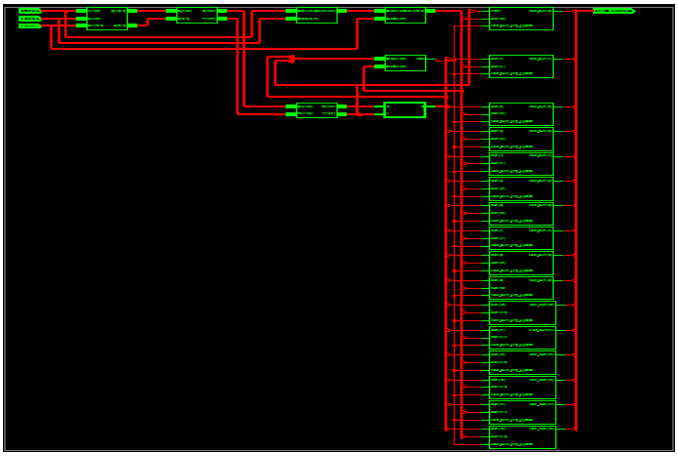


Fig. No. 8 FIR filter using three operand Adder RTL Schematic

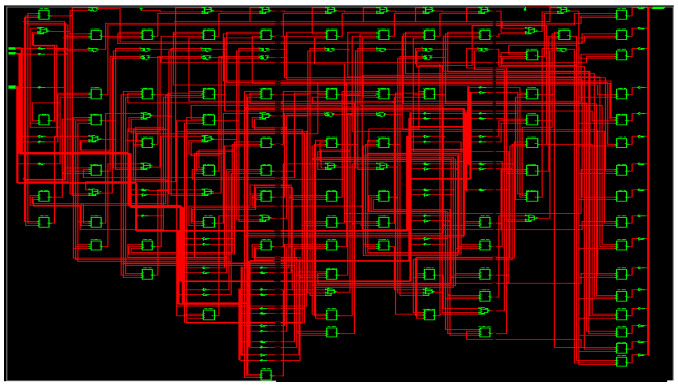


Fig. No. 9 FIR filter using three operand Adder Technology Schematic

4. CONCLUSION

A FIR filters is implemented with three operand adder. This three operand binary adder is used in digital signal processing, area efficient system like satellite, mobile phones. It is suitable to develop the high data rate light weight hardware security system in the field of IoT. It is designed to reduced area and power by reducing the count of adder unit. Based on this implemented the FIR filter the Area Delay Product and Power Delay Product are reduced. The path delay is achieved as 5.684ns which is compared to the others adders it is very smaller. So that it provides less area and power. In future we can do in hardware security system.

REFERENCES:

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] A. K. Panda and K. C. Ray, "Design and FPGA prototype of 1024- bit Blum-Blum-Shub PRBG architecture," in Proc. IEEE Int. Conf. Inf. Commun. Signal Process. (ICICSP), Singapore, Sep. 2018, pp. 38–43.

[9] K. S. Pandey, D. K. B. N. Goel, and H. Shrimali, "An ultra-fast parallel prefix adder," in Proc. IEEE 26th Symp. Comput. Arithmetic (ARITH), Kyoto, Japan, Jun. 2019, pp. 125–134.

[10] F. Jafarzadehpour, A. S. Molahosseini, A. A. Emrani Zarandi, and L. Sousa, "New energy-efficient hybrid wide-operand adder architec- ture," IET Circuits, Devices Syst., vol. 13, no. 8, pp. 1221–1231, Nov. 2019.