IMPLEMENTATION OF AREA EFFICIENT THREE INPUT ADDER WITH HIGH SPEED

[1]Ruhina Fatima, [2]Dr. M Pavithra Jyothi
[1]PG Scholar, M.Tech, Dept of VLSI, Shadan Women's College of Engineering and Technology, Hyderabad, TS.
ruhinawahab15@gmail.com
[2]Professor, Dept of ECE, Shadan Women's College of Engineering and Technology, Hyderabad, TS.

ABSTRACT
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. However, the ripple-carry stage in the CS3A leads to a high propagation delay of O(n). Moreover, a parallel prefix two-operand adder such as Han-Carlson (HCA) can also be used for three-operand addition that significantly reduces the critical path delay at the cost of additional hardware. Hence, a new high-speed and area-efficient adder architecture is proposed using pre-compute bitwise addition followed by carry-prefix computation logic to perform the three-operand binary addition that consumes substantially less area, low power and drastically reduces the adder delay to O($\log_2 n$). The proposed architecture is implemented on the FPGA device for functional validation and also synthesized with the commercially available 32nm CMOS technology library. The post-synthesis results of the proposed adder reported 3.12, 5.31 and 9.28 times faster than the CS3A for 32-, 64- and 128- bit architecture respectively. Moreover, it has a lesser area, lower power dissipation and smaller delay than the HC3A adder. Also, the proposed adder achieves the lowest ADP and PDP than the existing three-operand adder techniques.

INTRODUCTION

Digital systems are highly complex at their most detailed level. They may consist of millions of elements i.e., transistors or logic gates. For many decades, logic schematics served as thengua. Franca of logic design, but not anymore. Today, hardware complexity has grown to such a degree that a schematic with logic gates is almost useless as it shows only a web of connectivity and not functionality of design. Since the 1970s, computer engineers, electrical engineers and electronics engineers have moved toward Hardware description language (HDLs). Digital circuit has rapidly evolved over the last twenty five years .The earliest digital circuits were designed with vacuum tubes and transistors. Integrated circuits were then invented where logic gates were placed on a single chip. The first IC chip was small scale integration (SSI) chips where the gate count is small. When technology became sophisticated, designers were able to place circuits with hundreds of gates on a chip. These chips were called MSI chips with advent of LSI; designers could put thousands of gates on a single chip. At this point, design process is getting complicated and designers felt the need to automate these processes.

With the advent of VLSI technology, designers could design single chip with more than hundred thousand gates. Because of the complexity of these circuits computer aided techniques became critical for verification and for designing these digital circuits. One way to lead with increasing complexity of electronic systems and the increasing time to market is to design at high levels of abstraction. Traditional paper and pencil and capture and simulate methods have largely given way to the described UN synthesized approach. For these reasons, hardware description languages have played an important role in describe and synthesis design methodology. They are used for specification, simulation and synthesis of an electronic system. This helps to reduce the complexity in designing and products are made to be available in market quickly. The components of a digital system can be classified as being specific to an application or as being standard circuits. Standard components are taken from a set that has been used in other systems. MSI components are standard circuits and their use results in a significant reduction in the total cost as compared to the cost of using SSI Circuits. In contrasts, specific components are particular to the system being implemented and are not commonly found among the standard components. The implementation of specific circuits with LSI chips can be done by means of IC that can be programmed to provide the required logic.

LITERATURE SURVEY

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**
Elliptic Curve Cryptography (ECC) has recognized much more attention over the last few years and has time-honored itself among the renowned public key cryptography schemes. The main feature of ECC is that shorter keys can be used as the best option for implementation of public key cryptography in resource-constrained (memory, power, and speed) devices like the Internet of Things (IoT), wireless sensor based

applications, etc. The performance of hardware implementation for ECC is affected by basic design elements such as a coordinate system, modular arithmetic algorithms, implementation target, and underlying finite fields. This paper shows the generic structure of the ECC system implementation which allows the different types of designing parameters like elliptic curve, Galois prime finite field GF(p), and input type. The ECC system is analyzed with performance parameters such as required memory, elapsed time, and process complexity on the MATLAB platform. The simulations are carried out on the 8th generation Intel core i7 processor with the specifications of 8 GB RAM, 3.1 GHz, and 64-bit architecture. This analysis helps to design an efficient and high performance architecture of the ECC system on Application Specific Integrated Circuit (ASIC) and Field Programmable Gate Array (FPGA).Elliptic Curve Cryptography (ECC) has recognized much more attention over the last few years and has time-honored itself among the renowned public key cryptography schemes. The main feature of ECC is that shorter keys can be used as the best option for implementation of public key cryptography in resource-constrained (memory, power, and speed) devices like the Internet of Things (IoT), wireless sensor based applications, etc. The performance of hardware implementation for ECC is affected by basic design elements such as a coordinate system, modular arithmetic algorithms, implementation target, and underlying finite fields. This paper shows the generic structure of the ECC system implementation which allows the different types of designing parameters like elliptic curve, Galois prime finite field GF(p), and input type. The ECC system is analyzed with performance parameters such as required memory, elapsed time, and process complexity on the MATLAB platform. The simulations are carried out on the 8th generation Intel core i7 processor with the specifications of 8 GB RAM, 3.1 GHz, and 64-bit architecture. This analysis helps to design an efficient and high performance architecture of the ECC system on Application Specific Integrated Circuit (ASIC) and Field Programmable Gate Array (FPGA).

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**

Verification of an ECDSA signature requires a double scalar multiplication on an elliptic curve. In this work, we study the computation of this operation on a twisted Edwards curve with an efficiently computable endomorphism, which allows reducing the number of point doublings by approximately 50 percent compared to a conventional implementation. In particular, we focus on a curve defined over the 207-bit prime field F_p with $p = 2^{207} - 5{,}131$. We develop several optimizations to the operation and we describe two hardware architectures for computing the operation. The first architecture is a small processor implemented in 0.13 μm CMOS ASIC and is useful in resource-constrained devices for the Internet of Things (IoT) applications. The second architecture is designed for fast signature verifications by using FPGA acceleration and can be used in the server-side of these applications. Our designs offer various trade-offs and optimizations between performance and resource requirements and they are valuable for IoT applications.

Z. Liu, D. Liu, and X. Zou, **An efficient and flexible hardware implementation of the dual-field elliptic curve cryptographic processor**

Developing a high-speed elliptic curve cryptographic (ECC) processor that performs fast point multiplication with low hardware utilization is a crucial demand in the fields of cryptography and network security. This paper presents field-programmable gate array (FPGA) implementation of a high-speed, low-area, side-channel attacks (SCAs) resistant ECC processor over a prime field. The processor supports 256-bit point multiplication on recently recommended twisted Edwards curve, namely, Edwards25519, which is used for a high-security digital signature scheme called Edwards curve digital signature algorithm (EdDSA). The paper proposes novel hardware architectures for point addition and point doubling operations on the twisted Edwards curve, where the processor takes only 516 and 1029 clock cycles to perform each point addition and point doubling, respectively. For a 256-bit key, the proposed ECC processor performs single point multiplication in 1.48 ms, running at a maximum clock frequency of 177.7 MHz in a cycle count of 262 650 with a throughput of 173.2 kbps, utilizing only 8873 slices on the Xilinx Virtex-7 FPGA platform, where the points are represented in projective coordinates. The implemented design is time-area-efficient as it offers fast scalar multiplication with low hardware utilization without compromising the security level.

S.-R. Kuang, K.-Y. Wu, and R.-Y. Lu, **Low-cost high-performance VLSI architecture for montgomery modular multiplication**

This paper proposes a simple and efficient Montgomery multiplication algorithm such that the low-cost and high-performance Montgomery modular multiplier can be implemented accordingly. The proposed multiplier receives and outputs the data with binary representation and uses only one-level carry-save adder (CSA) to avoid the carry propagation at each addition operation. This CSA is also used to perform operand precomputation and format conversion from the carrysave format to the binary representation, leading to a low hardware cost and short critical path delay at the expense of extra clock cycles for completing one

modular multiplication. To overcome the weakness, a configurable CSA (CCSA), which could be one full-adder or two serial half-adders, is proposed to reduce the extra clock cycles for operand precomputation and format conversion by half. In addition, a mechanism that can detect and skip the unnecessary carry-save addition operations in the one-level CCSA architecture while maintaining the short critical path delay is developed. As a result, the extra clock cycles for operand precomputation and format conversion can be hidden and high throughput can be obtained. Experimental results show that the proposed Montgomery modular multiplier can achieve higher performance and significant area–time product improvement when compared with previous designs.

S.-R. Kuang, J.-P. Wang, K.-C. Chang, and H.-W. Hsu, **Energy-efficient high-throughput montgomery modular multipliers for RSA cryptosystems**

Modular exponentiation in the Rivest, Shamir, and Adleman cryptosystem is usually achieved by repeated modular multiplications on large integers. To speed up the encryption/decryption process, many high-speed Montgomery modular multiplication algorithms and hardware architectures employ carry-save addition to avoid the carry propagation at each addition operation of the add-shift loop. In this paper, we propose an energy-efficient algorithm and its corresponding architecture to not only reduce the energy consumption but also further enhance the throughput of Montgomery modular multipliers. The proposed architecture is capable of bypassing the superfluous carry-save addition and register write operations, leading to less energy consumption and higher throughput. In addition, we also modify the barrel register full adder (BRFA) so that the gated clock design technique can be applied to significantly reduce the energy consumption of storage elements in BRFA. Experimental results show that the proposed approaches can achieve up to 60% energy saving and 24.6% throughput improvement for 1024-bit Montgomery multiplier

R. S. Katti and S. K. Srinivasan, **Efficient hardware implementation of a new pseudo-random bit sequence generator**

In this paper we propose a new linear congruential generator (LCG) based pseudo random bit-sequence generator (PRBG) and its hardware implementation. Linear congruential generators (LCGs) of the form $x_{i+1} = ax_i + b \pmod m$, have been used to generate pseudorandom numbers. However these generators have been known to be insecure. The proposed PRBG couples four such LCGs and is secure. A preliminary proof of security is outlined in this paper. The PRBG generates bit-sequences that pass all NIST pseudo randomness tests. Our PRBG has a very efficient hardware implementation because the modulo operation is with respect to 2n as opposed to $p \times q$ in the Blum-Blum-Shub (BBS) generator, where p and q are large prime numbers. We also show that the hardware implementation can be easily pipelined, thereby increasing the throughput in spite of the hardware having large word-length inputs ($n \geq 128$). A 4-stage pipelined hardware was implemented in VHDL for n = 128 and the synthesized hardware was simulated. Simulation results showed a 2.81 fold increase in throughput (number of pseudo-random bits output per unit time) compared to the non-pipelined version.

A. K. Panda and K. C. Ray, **Modified dual-CLCG method and its VLSI architecture for pseudorandom bit generation**

High implementation complexity of multi-scroll circuit is a bottleneck problem in real chaos-based communication. Especially, in multi-scroll Chua's circuit, the simplified implementation of piecewise-linear resistors with multiple segments is difficult due to their intricate irregular breakpoints and slopes. To solve the challenge, this paper presents a systematic scheme for synthesizing a Chua's diode with multi-segment piecewise linearity, which is achieved by cascading even-numbered passive nonlinear resistors with odd-numbered ones via a negative impedance converter. The traditional voltage mode op-amps are used to implement nonlinear resistors. As no extra DC bias voltage is employed, the scheme can be implemented by much simpler circuits. The voltage-current characteristics of the obtained Chua's diode are analyzed theoretically and verified by numerical simulations. Using the Chua's diode and a second order active Sallen-Key high-pass filter, a new inductor-free Chua's circuit is then constructed to generate multi-scroll chaotic attractors. Different number of scrolls can be generated by changing the number of passive nonlinear resistor cells or adjusting two coupling parameters. Besides, the system can be scaled by using different power supplies, satisfying the low voltage low-power requirement of integrated circuit design.

EXISTING SYSTEM

The three-operand binary addition is one of the critical arithmetic operation in the congruential modular arithmetic architectures [5] – [8] and LCG-based PRBG methods such as CLCG [9], MDCLCG [10] and CVLCG [11]. 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 [9]–[14]. 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

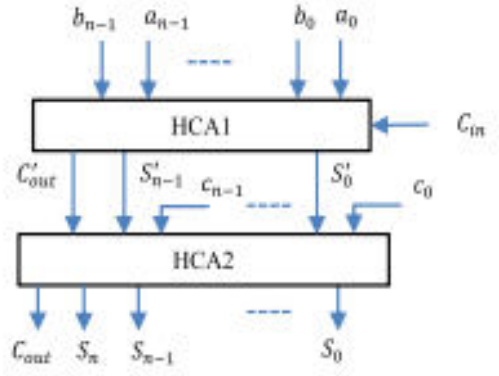
addition. The "carry-out" signal is propagated through the n number of full adders in the ripple-carry stage. Therefore, the delay increases linearly with the increase of bit length. The architecture of the three-operand carry-save adder is shown in Fig. 1 and the critical path delay is highlighted with 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,

$$T_{CS3A} = (n+1)T_{FA} = 3T_X + 2nT_G$$

Similarly, the total area is evaluated as follows,

$$A_{CS3A} = 2nA_{FA} = 4nA_X + 6nA_G$$

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.



Block architecture of HCA-based three-operand adder (HC3A).

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 [16], [17]. 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) [17]. In recent years, various such kind of parallel prefix two-operand adders, i.e., Ling [18], Jackson-Talwar [19], ultra-fast adder [20], hybrid PPFCSL [21] and hybrid Han-Carlson [22] are also discussed in the literature. The ultra-fast adder [20] 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 [22] 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 [22]. 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. 2. The detailed architecture of HCA-based three-operand adder (HC3A) is presented in [15]. 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,

$$T_{HC3A} \approx 4T_X + 4\lfloor \log_2 n \rfloor T_G$$
$$A_{HC3A} \approx (4n+1)A_X + 6\left[n + \left\lceil \frac{n}{2} \right\rceil s - 2^s + 1\right]A_G$$

Here, s = $\log_2 n - 1$. The HCA-based three-operand binary adder (HC3A) [15] 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.

PROPOSED METHODOLOGY

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:

$$S'_i = a_i \oplus b_i \oplus c_i,$$
$$cy_i = a_i \cdot b_i + b_i \cdot c_i + c_i \cdot a_i$$

Stage-2: Base Logic:

$$G_{i:i} = G_i = S'_i \cdot cy_{i-1}, \quad G_{0:0} = G_0 = S'_0 \cdot C_{in}$$
$$P_{i:i} = P_i = S'_i \oplus cy_{i-1}, \quad P_{0:0} = P_0 = S'_0 \oplus C_{in}$$

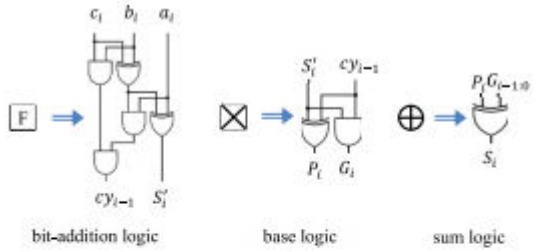
Stage-3: PG (Generate and Propagate) Logic:

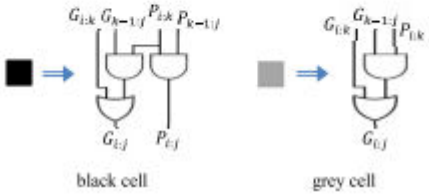
$$G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j},$$
$$P_{i:j} = P_{i:k} \cdot P_{k-1:j}$$

Stage-4: Sum Logic:

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

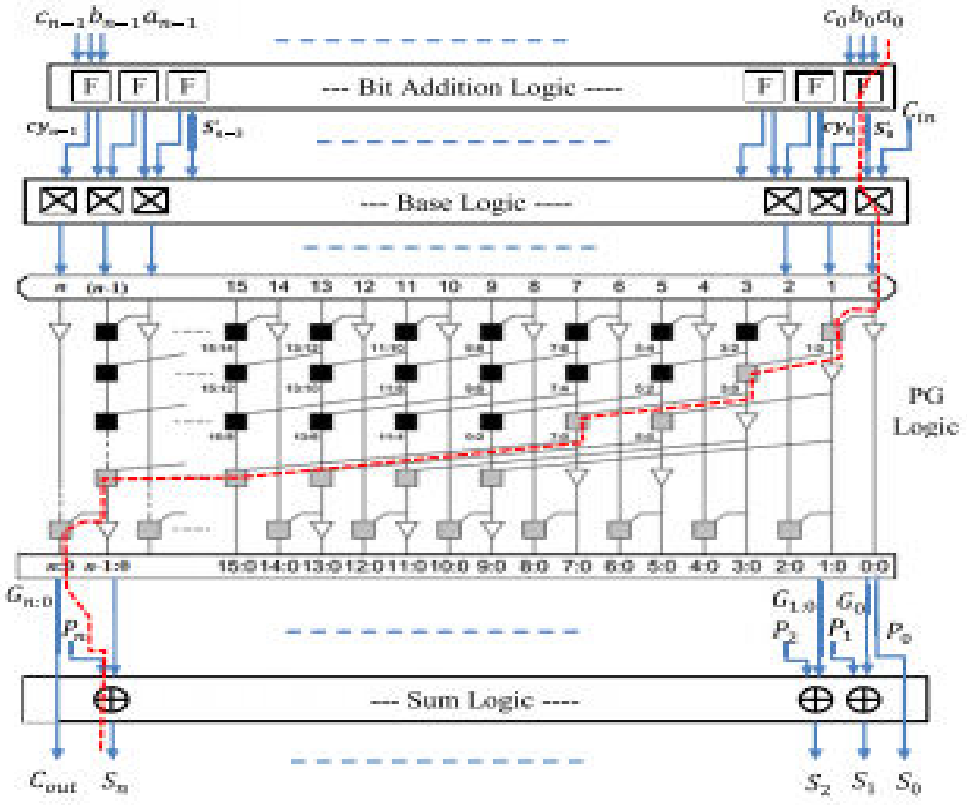
The proposed VLSI architecture of the three-operand binary adder and its internal structure is shown in Fig. 3. The new adder technique performs the addition of three n-bit binary inputs in four different stages. In the first stage (bit-addition logic), the bitwise addition of three n-bit binary input operands





Logical diagram of bit addition, base logic, sum logic, black-cell and grey-cell is performed with the array of full adders, and each full adder computes "sum (S i)" and "carry (cyi)" signals as highlighted in Fig. 3(a). The logical expressions for computing sum (S i) 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 (S i)" 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 Fig. 3(a) and there are n +1 number of saltire-cells in the base logic stage.



Proposed three-operand adder: First order VLSI architecture,

The logic diagram of the saltire-cell is shown in Fig. 3(b), 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 (S 0 · 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 Fig. 3(b) that computes the carry generate Gi: j and propagate Pi: j signals with the following logical expression,

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

Method	Complexity	n- bit	32- bit	64- bit	128- bit
CS3A [4]	Timing	$3T_X + 2nT_G$	$3T_X + 64T_G$	$3T_X + 128T_G$	$3T_X + 256T_G$
	Area	$4nA_X + 6nA_G$	$128A_X + 192A_G$	$256A_X + 384A_G$	$512A_X + 768A_G$
HC3A [15]	Timing	$4T_X + 4\lceil \log_2 n \rceil T_G$	$4T_X + 20T_G$	$4T_X + 24T_G$	$4T_X + 28T_G$
	Area	$(4n+1)A_X + 6\left[n + \left\lceil \frac{n}{2} \right\rceil s - 2^s + 1\right]A_G$	$129A_X + 486A_G$	$257A_X + 1158A_G$	$513A_X + 2694A_G$
HHC3A (Three-operand adder using [22])	Timing	$4T_X + 2\lceil \log_2 n^* + 2 \rceil T_G$	$4T_X + 14T_G$	$4T_X + 16T_G$	$4T_X + 18T_G$
	Area	$(4n+1)A_X + (9n - 3\log_2 n - 2)A_G$	$129A_X + 271A_G$	$257A_X + 556A_G$	$513A_X + 1129A_G$
Proposed	Timing	$4T_X + 2\lceil \log_2 n' + 1 \rceil T_G$	$4T_X + 12T_G$	$4T_X + 14T_G$	$4T_X + 16T_G$
	Area	$(4n+1)A_X + \left[6n + 3s\left\lceil \frac{n}{2} \right\rceil - 3 \times 2^s + 4\right]A_G$	$129A_X + 340A_G$	$257A_X + 772A_G$	$513A_X + 1732A_G$

$s = \lceil \log_2 n - 1 \rceil$, $n' = n - 1, n^* = n - 5$ for $n \geq 8$

Area and Timing Complexity of Three-Operand Adder

The number of prefix computation stages for the proposed adder is (log2 n+1), and therefore, the critical path delay of the proposed adder is mainly influenced by this carry propagate chain. The final stage is represented as sum logic in which the "sum (Si)" bits are computed from the carry generate Gi: j and carry propagate Pi bits using the logical expression, Si = (Pi ⊕ Gi−1:0). The carryout signal (Cout) is directly obtained from the carry generate bit Gn: 0.

A) Comparison of 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 (Tprop) of the proposed three-operand adder architecture can be evaluated as follows:

$$\begin{aligned} T_{prop} &= T_{bitadd} + T_{base} + T_{PG} + T_{sum} \\ &\approx 2T_X + T_X + 2\lceil \log_2 n' + 1 \rceil T_G + T_X \\ &\approx 4T_X + 2\lceil \log_2 n' + 1 \rceil T_G \end{aligned}$$

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

$$\begin{aligned} A_{prop} &= A_{bitadd} + A_{base} + A_{PG} + A_{sum} \\ &\approx (2nA_X + 3nA_G) + (n+1)(A_X + A_G) \\ &\quad + \left[2n + 3s\left\lceil \frac{n}{2} \right\rceil - 3 \times 2^s + 3\right]A_G + nA_X \\ A_{prop} &\approx (4n+1)A_X + \left[6n + 3s\left\lceil \frac{n}{2} \right\rceil - 3 \times 2^s + 4\right]A_G \end{aligned}$$

Here s=[log2n-1] and n'=n-1. . In order to have a comprehensive assessment, the concept of hybrid Han-Carlson two-operand adder given in [22] is extended to develop a three-operand adder architecture. The first order timing-area complexity of this hybrid Han-Carlson three-operand adder

HHC3A is evaluated as

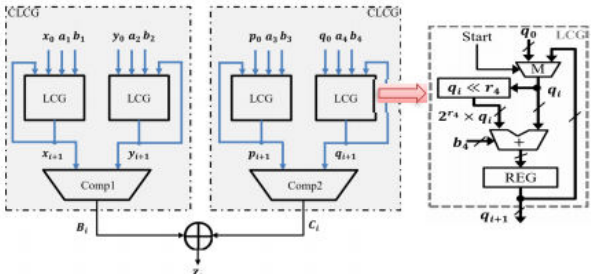
$$\begin{aligned} T_{HHC3A} &= T_{bitadd} + T_{base} + T_{PG} + T_{sum} \\ &\approx 2T_X + T_X + 2\lceil \log_2 n^* + 2 \rceil T_G + T_X \\ &\approx 4T_X + 2\lceil \log_2 n^* + 2 \rceil T_G \\ A_{HHC3A} &= A_{bitadd} + A_{base} + A_{PG} + A_{sum} \\ &\approx (2nA_X + 3nA_G) + (n+1)(A_X + A_G) \\ &\quad + (5n - 3\log_2 n - 3)A_G + nA_X \\ A_{HHC3A} &\approx (4n+1)A_X + (9n - 3\log_2 n - 2)A_G \end{aligned}$$

Here n*=n-5 for n>8.Similarly, the ultrafast two-operand adder given in [20] is also extended to develop the three-operand adder (UF3A), and the area time complexity is evaluated. Table I summarizes the area and timing complexity of the proposed three-operand adder architecture along with the CS3A, HC3A, HHC3A and UF3A three-operand adder architectures. It reports that the proposed adder architecture saves 43.4%, 50.2% and 55.6% AG 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 considerably consumes 10% to 18% less gate area but slower by two gates delay than the proposed adder. On the other hand, ultra-fast two-operand based UF3A is the fastest three operand adder. It is faster than the proposed adder by three gates delay but consumes comparatively two-times large gate area than the HHC3A and proposed three-operand adders.

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. Based on experimental results, the proposed adder has also reported the 6.3% reduction of ADP and 10.7% reduction of PDP when compared to the HHC3A adder for 32- bit architecture.

Thus, the proposed adder not only has less delay but also enhances the energy consumption (PDP) and area-delay product (ADP). Though the standard design performance metrics are compared in Table II, the ADP and PDP are presented graphically for better illustration in Fig. 4(a) and Fig. 4(b) respectively. From the graphical representation, it is clear that ADP and PDP performance of CS3A increase exponentially with respect to increasing in bit size, whereas, these parameters increase linearly with higher slope for HC3A and HHC3A.



MDCLCG architecture

However, in case of proposed adder topology, the ADP and PDP performance parameters increase linearly with lower slope than the other three-operand adders. Hence, the proposed three operand adder involves significantly less ADP and PDP than other three-operand adder techniques. The performance of the proposed adder is further measured by incorporating the design in the modified dual-CLCG (MDCLCG) PRBG method for high data rate lightweight hardware security in the field of IoT applications.

PERFORMANCE OF THE MODIFIED DUAL-CLCG ARCHITECTURE WITH THE PROPOSED THREE-OPERAND ADDER

The hardware security in the field of IoT applications demands stream-cipher based high data rate, lightweight cryptography technique for fastest encryption/ decryption. Key generator or pseudorandom bit generator (PRBG) is the primary component in the stream-cipher based encryption/ decryption. Modified dual-CLCG (MDCLCG) is the most efficient PRBG method amongst the existing PRBG methods which is suitable for stream-cipher based hardware security. However, the security strength of the MDCLCG method linearly depends on the bit size of the congruential modulus. It is polynomial-time unpredictable and secure if n ≥ 32- bits [10]. The hardware architecture of the MDLCG method is based on LCG, as shown in Fig. 5 in which three-operand modulo2n adder is the primary computational arithmetic block. The MDCLCG architecture presented in [10] is developed with four three-operand modulo-2n carry-save adders (CS3A) and two magnitude comparators along with four registers and multiplexers. The longer carry propagation gate delay in CS3A adder influences the performance of MDCLCG architecture with an increase of bit size. Therefore, in this section, the performance metrics of the MDCLCG are measured by replacing the CS3A adder with the HHC3A and proposed adder architectures. By considering the operation of three-operand modulo-2n addition in MDCLCG method, the architecture of the proposed adder is further redesigned. Therefore, the area (AP3OA) and time (TP3OA) complexity of the proposed adder architecture can be evaluated for three-operand modulo-2n addition as follows,

$$A_{P3OA} \approx (4n-2)A_X + \left[6n + 3s\left\lceil\frac{n'}{2}\right\rceil - 3\times 2^s\right]A_G$$
$$T_{P3OA} \approx 4T_X + 2\left\lceil\log_2 n' + 1\right\rceil T_G$$

Here, s = [log2 n – 1] and n' = n − 1. Similarly, the area (Amgc) and time (Tmgc) complexity of the magnitude comparator is evaluated in [10] which is further highlighted as follows

$$A_{mgc} \approx (n-1)\left[9A_G + 4A_N\right]$$
$$T_{mgc} \approx 4T_X + 4\left(\log_2 n\right)T_G$$

Therefore, the area and time complexity of the MDCLCG architecture using the proposed adder and the magnitude comparator can be evaluated as follows,

$$T_{MDCLCG} \approx T_{3oa} + T_{mx}$$
$$\approx 4T_X + 2\lceil \log_2 n' + 2 \rceil T_G$$
$$A_{MDCLCG} = 4(A_{mx} + A_{3oa} + A_{rg}) + 2A_{cmp} + A_X$$
$$\approx (16n - 7)A_X + +4(3n - 2)A_N + 4nA_{FF}$$
$$+ 2\left[27n + 6s\left\lceil \frac{n'}{2} \right\rceil - 6\times 2^s - 5\right] A_G$$

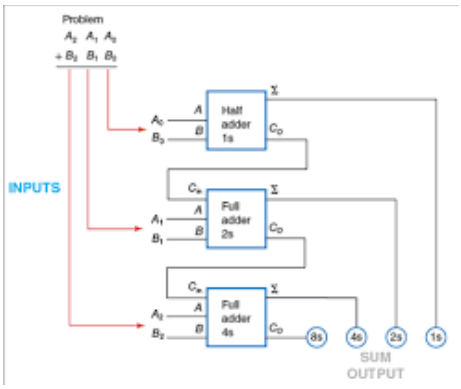
Thus, the critical path delay of the proposed adder based MDCLCG architecture is drastically reduced in the order of O(log2 n), and hence, the overall latency is significantly improved. The 32- bit architecture of MDCLCG method based on the proposed three-operand adder is implemented using Verilog HDL and thereafter synthesized using Synopsys

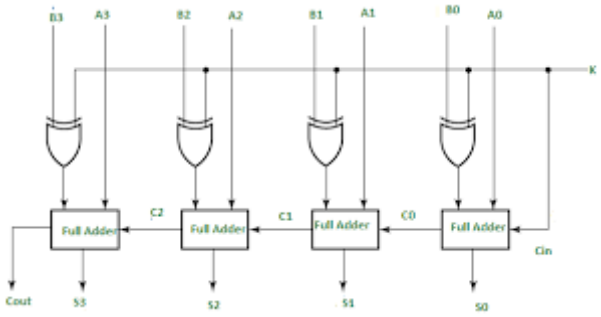
MDCLG arch. using different three-operand adder techniques	Area (μm^2)	Area Ratio	Delay (ns)	Delay Ratio	Power (μW)	Power Ratio	Area × Delay ($\mu m^2 \times$ ns)	ADP Ratio	Power × Delay ($\mu W \times$ ns)	PDP Ratio
CSJA-based MDCLCG	6291.033	0.83	2.13	2.34	177.622	0.85	13399.900	1.95	378.335	1.94
HCSA-based MDCLCG	9037.855	1.19	1.19	1.31	246.018	1.17	10755.047	1.57	292.762	1.49
HHCJA-based MDCLCG	7356.858	0.97	0.98	1.08	202.695	0.97	7209.721	1.05	198.641	1.05
Prop. adder based MDCLCG	7544.041	1.00	0.91	1.00	208.934	1.00	6865.077	1.00	190.129	1.00

BINARY ADDER ARCHITECTURE

A Binary Adder is a digital circuit that performs the arithmetic sum of two binary numbers provided with any length. A Binary Adder is constructed using full-adder circuits connected in series, with the output carry from one full-adder connected to the input carry of the next full-adder. With respect to asymptotic delay time and area complexity, the binary adder architectures can be categorized into four primary classes as given in Table 1. The given results in the table are the highest exponent term of the exact formulas [2], very complex for the high bit lengths of the operands. The first class consists of the very slow ripple carry adder with the smallest area. In the second class, the carry-skip, carry-select and carry increment adders with multiple levels have small area requirements and shortened computation times. From the third class, the carry-lookahead adder and from the fourth class, the parallel prefix and conditional sum adders represent the fastest addition schemes with the largest area complexities. In this section, the circuit structures of the binary adder architectures are given by the set of logic equations, defining single bit cell. In addition, the area and time complexities for each adder architectures based on unit-gate model are given. In this work, we only studied on the four binary adders as the typical structures belong to the different adder classes.

Full **Adder** is a combinational circuit that performs the addition of three **bits** (two significant **bits** and previous carry). ➢It consists of three inputs and two outputs, two inputs are the **bits** to be. added, the third input represents the carry form the previous position.





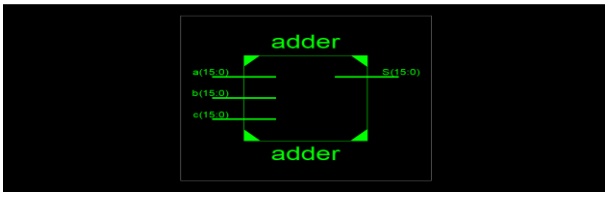
BINARY ADDER/SUBTRACTOR

- When the mode input (M) is at a low logic, i.e. '0', the circuit act as an adder and when the mode input is at a high logic, i.e. '1', the circuit act as a subtractor.
- The exclusive-OR gate connected in series receives input M and one of the inputs B.
- When M is at a low logic, we have B⊕ 0 = B.
- The full-adders receive the value of B, the input carry is 0, and the circuit performs A plus B.
- When M is at a high logic, we have B⊕ 1 = B' and C0 = 1.
- The B inputs are complemented, and a 1 is added through the input carry. The circuit performs the operation A plus the 2's complement of B.

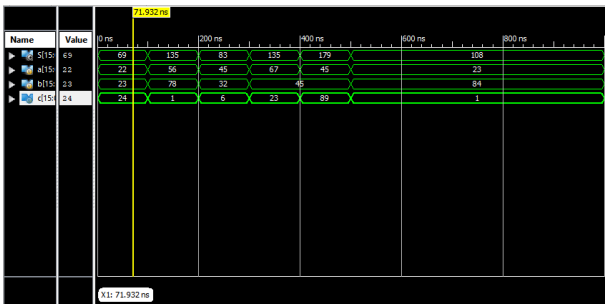
SIMULATION RESULTS

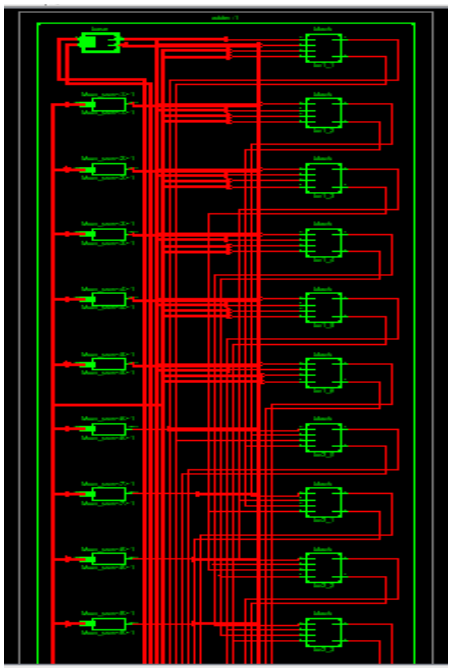
adder Project Status (06/17/2022 - 22:44:31)			
Project File:	three.xise	**Parser Errors:**	No Errors
Module Name:	adder	**Implementation State:**	Placed and Routed
Target Device:	xc7a100t-3csg324	**•Errors:**	No Errors
Product Version:	ISE 14.7	**•Warnings:**	70 Warnings (70 new)
Design Goal:	Balanced	**•Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**•Timing Constraints:**	
Environment:	System Settings	**•Final Timing Score:**	0 (Timing Report)

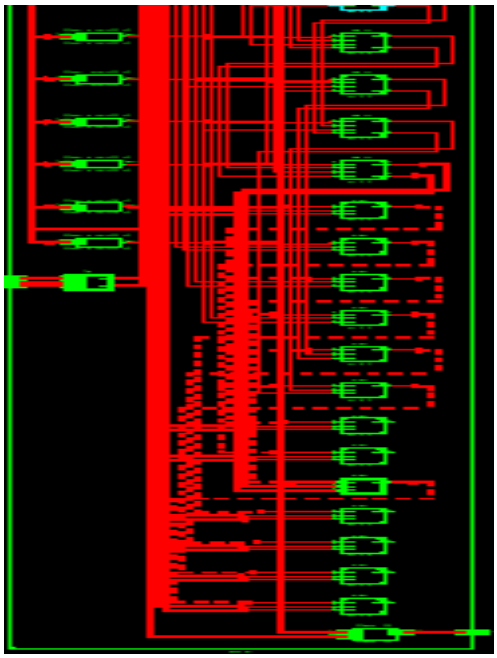
Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	0	126,800	0%	
Number of Slice LUTs	43	63,400	1%	



Device Utilization Summary				
Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	0	126,800	0%	
Number of Slice LUTs	43	63,400	1%	
Number used as logic	43	63,400	1%	
Number using O6 output only	32			
Number using O5 output only	0			
Number using O5 and O6	11			
Number used as ROM	0			
Number used as Memory	0	19,000	0%	
Number used exclusively as route-thrus	0			
Number of occupied Slices	21	15,850	1%	
Number of LUT Flip Flop pairs used	43			
Number with an unused Flip Flop	43	43	100%	
Number with an unused LUT	0	43	0%	
Number of fully used LUT-FF pairs	0	43	0%	
Number of slice register sites lost to control set restrictions	0	126,800	0%	
Number of bonded IOBs	64	210	30%	
Number of RAMB36E1/FIFO36E1s	0	135	0%	
Number of RAMB18E1/FIFO18E1s	0	270	0%	







CONCLUSION

A high-speed area efficient adder technique and its VLSI architecture is proposed to perform the three operand binary addition for efficient computation of modular arithmetic used in cryptography and PRBG applications. The proposed three-operand adder technique is a parallel prefix adder that uses four-stage structures to compute the addition of three input operands. The novelty of this proposed architecture is the reduction of delay and area in the prefix computation stages in PG logic and bit-addition logic that leads to an overall reduction in critical path delay, area-delay product (ADP) and power-delay product (PDP). For the fair comparison, the concept of hybrid Han-Carlson two-operand adder is extended to develop a hybrid Han-Carlson three-operand adder (HHC3A) topology. The same coding style adopted in proposed adder architecture is extended to implement the HHC3A, HC3A and CS3A using Verilog HDL. Further, all these designs are synthesized using commercially available 32nm CMOS technology library to obtain the core area, timing and power for different word size. From the physical synthesis results, this is clear that the proposed adder architecture is 3 to 9 times faster than the correspondingCS3A adder architecture. Moreover, a sharp reduction in area utilization, timing path and power dissipation can be observed in the proposed adder as compared to the HC3A adder. It is also worth noting that the proposed adder has significantly less ADP and PDP compared to other three-operand adder techniques. It has 55.1%, 71.6% and 82.7% reduction on ADP, and 55.1%, 71.8% and 82.9% reduction on PDP over CS3A adder for 32- 64- and 128- bit architectures respectively. Further, a 32- bit MDCLCG architecture presented in literature is designed with the proposed adder architecture by replacing its CS3A three-operand adder architecture, and prototyped on commercially available FPGA device for validating the design on a silicon chip. The performance metrics reported that the proposed adder based MDCLCG architecture is 2.34 times faster than CS3A-based MDCLCG architecture that makes it suitable to develop a high data rate lightweight hardware security system in the field of IoT.

REFERNCES

[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-save adder 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.

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

[16] T. Han and D. A. Carlson, "Fast area-efficient VLSI adders," in Proc. IEEE 8th Symp. Comput. Arithmetic (ARITH), May 1987, pp. 49–56.

[17] D. L. Harris, "Parallel prefix networks that make tradeoffs between logic levels, fanout and wiring racks," U.S. Patent 0225706 A1, Nov. 11, 2004.

[18] H. Ling, "High-speed binary adder," IBM J. Res. Develop., vol. 25, no. 3, pp. 156–166, Mar. 1981.

[19] R. Jackson and S. Talwar, "High speed binary addition," in Proc. Conf. Rec. 38th Asilomar Conf. Signals, Syst. Comput., vol. 2. Pacific Grove, CA, USA, Nov. 2004, pp. 1350–1353.

20] 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.

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

[22] S. MuthyalaSudhakar, K. P. Chidambaram, and E. E. Swartzlander, "Hybrid Han-Carlson adder," in Proc. IEEE 55th Int. Midwest Symp. Circuits Syst. (MWSCAS), Boise, ID, USA, Aug. 2012, pp. 818–821.