IMPLEMENTATION OF THREE-OPERAND BINARY ADDER USING MODIFIED KOOGLE STONE ADDER

Y. MOUNIKA 1, P. SURYA PRAKASH REDDY 2

1 PG SCHOLAR, DEPT OF ECE, GATES INSTITUTE OF TECHNOLOGY, GOOTY, AP, INDIA.
2 ASSISTANT PROFESSOR, DEPARTMENT OF ECE, GATES INSTITUTE OF TECHNOLOGY, GOOTY, AP, INDIA.

ABSTRACT: 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. 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.

1. INTRODUCTION
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 multiplication and 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 congruential modular arithmetic operation. The most efficient approach to implement the modular multiplication and exponentiation is the Montgomery algorithm whose critical operation is based on the three-operand binary addition. The three-operand binary addition is also a primary arithmetic operation in the linear congruential generator (LCG) based pseudo-random bit generators (PRBG) such as coupled LCG (CLCG) [9], modified dual-CLCG (MDCLCG) and coupled variable input LCG (CVLCG) [11]. Modified dual-CLCG (MDCLCG) is the most secure and highly random PRBG method among all the LCG-based and other existing PRBG methods. It is polynomial-time unpredictable and secure if \( n \geq 32 \)-bits. Therefore, the security of the MDCLCG enhances with the increase of operand size. However, the area and critical path delay increases linearly since its hardware architecture consists of four three-operand modulo-2n adders, two comparators, four multiplexers area. Hence, the performance of the MDCLCG can be improved by the efficient implementation of the three-operand adder.

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 \( O(n \log_2 n) \) [15]. 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. 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.

2. LITERATURE SURVEY
Kogge P, Stone H proposed A parallel calculation for the productive arrangement of a general class Recurrence relations. A mth-request repeat issue is characterized as the calculation of the arrangement \( x_1, x_2, ..., x_N \), where \( x_i = f(i|x_i-1, ..., x_i-m) \) for some capacity \( f \). This paper utilizes a system called recursive multiplying in a calculation for illuminating a huge class of repeat issues on parallel PCs, for example, the Illiac IV. Recursive multiplying includes the part of the calculation of a capacity into two similarly complex subfunctions whose assessment can be performed at the same time in two separate processors. Progressive part of every one of these subfunctions spreads the calculation over more processors. This calculation can be connected to any repeat condition of the structure \( x_i = f(b_i, g(ai, xi-1)) \) where \( f \) and \( g \) are capacities that fulfill certain distributive and affiliated like properties. In spite of the fact that this repeat is first request, all straight mth-request repeat conditions can be thrown into this structure. Reasonable applications incorporate direct repeat conditions, polynomial assessment, a few nonlinear issues, the assurance of the greatest or least of \( N \) numbers, and the arrangement of tridiagonal straight conditions. The subsequent calculation...
registers the whole arrangement x1, ..., xN in time corresponding to \([\log_2 N]\) on a PC with N-overlap parallelism. On a sequential PC, calculation time is relative to N.

3. 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 CVLCLG [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 \(a_i\), \(b_i\) and \(c_i\). 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.

Fig.1. Three-operand carry-save adder (CS3A).

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 is the primary component.

4. PROPOSED SYSTEM
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 two operand 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 Han-Carlson adder, with a 10% to 18% reduction in the gate complexity [22].

Fig.1. Block level architecture of HCA-based three-operand adder (HC3A).

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 areadelay 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 combinatorial 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. 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 \(O(n \log_2 n)\). 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.

Fig.2. Proposed three-operand adder; (a) First order VLSI architecture,

![Diagram of VLSI architecture](image)

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:

\[ S_i = a_i \land b_i \land c_i \land \neg c_i \]

\[ c_{i-1} = a_i \land b_i \land c_i \land \neg c_i \]

Stage-2: Base Logic:

\[ G_i = G_{i-1} \land c_{i-1} \land \neg G_{i-1} \land \neg c_{i-1} \]

\[ P_i = P_{i-1} \land c_{i-1} \land \neg P_{i-1} \land \neg c_{i-1} \]

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

\[ G_j = G_k \land P_j \land P_k \land G_{k+1} \]

\[ P_j = P_k \land P_{k+1} \land G_j \]

Stage-4: Sum Logic:

\[ S_i = (P_{i} \land G_i) \lor S_0 \land P_0 \land \neg C_0 \]

The logical expression of all these four stages are defined as follows,

Stage-1: Bit Addition Logic:

\[ S_i = a_i \land b_i \land c_i \land \neg c_i \]

\[ c_{i-1} = a_i \land b_i \land c_i \land \neg c_i \]

Stage-2: Base Logic:

\[ G_i = G_{i-1} \land c_{i-1} \land \neg G_{i-1} \land \neg c_{i-1} \]

\[ P_i = P_{i-1} \land c_{i-1} \land \neg P_{i-1} \land \neg c_{i-1} \]

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

\[ G_j = G_k \land P_j \land P_k \land G_{k+1} \]

\[ P_j = P_k \land P_{k+1} \land G_j \]

Stage-4: Sum Logic:

\[ S_i = (P_{i} \land G_i) \lor S_0 \land P_0 \land \neg C_0 \]

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 is performed with the array of full adders, and each full adder computes “sum (S_i)” and “carry (cyi)” signals as highlighted in Fig. 2. 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.

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. 2 and there are n+1 number of saltire-cells in the base logic stage. The logic diagram of the saltire-cell is shown in Fig. 3(b), and it is realized by the following logical expression,

\begin{align*}
Gi = & Gi = S_i \cdot cyi - 1; \\
Pi = & Pi = S_i \cdot cyi - 1
\end{align*}

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 \cdot 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 that computes the carry generate Gi: j and propagate Pi: j signals with the following logical expression,

\begin{align*}
Gi: & j = Gi: k + Pi: k \cdot Gi: k - 1; j \\
Pi: & j = Pi: k \cdot Pi: k - 1; j
\end{align*}

The number of prefix computation stages for the proposed adder is \(\log_2 n + 1\), 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.

5. MODIFIED PREFIX ADDER
It is readily apparent that a key advantage of the tree-structured adder is that the critical path due to the carry delay is on the order of \(\log_2 2N\) for an N-bit wide adder. The arrangement of the prefix network gives rise to various families of adders. For a discussion of the various carry-tree structures. For this study, the focus is on the prefix adder, known for having minimal logic depth and fanout. Here we designate BC as the black cell which generates the ordered pair in equation (1); the gray cell (GC) generates the left signal only, following . The regularity of the prefix adder network has built in redundancy which has implications for fault-tolerant designs. The sparse prefix adder adder, shown in Fig 1(b), is also studied. This hybrid design completes the summation process with a 4 bit RCA allowing the carry prefix network to be simplified.

The construction of the first level of the prefix tree of this adder is similar to the construction of prefix adder. The main structural difference begins from the second level of the prefix tree. At the second level of the prefix tree, the groups of two schematic nodes are formed, at the 3rd level – groups compose four schematic nodes and at the 4th level – groups including 8 schematic nodes, etc. This adder first computes gi and hi signals for the first stage. Then at the first level of prefix tree, Gi:k and Pi:k signals of 2-bit are computed at the same time, and then, it...
computes $G_i^k$ and $P_i^k$ signals for pairs of columns, then for blocks of 4, then for blocks of 8, then 16, and so on until the final $G_i^k$ signal for every column is known. Finally, at the last stage this adder computes the sums together with the generated signals obtained from the previous prefix computation stage. The number of levels of the prefix tree corresponds to $(\log_2 n)$ and the number of schematic nodes.

6. RESULT

7. CONCLUSION & FUTURE SCOPE

In this project, 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 CS3A using Verilog HDL. Further, all these designs are synthesized, area, timing and power. From the physical synthesis results, this is clear that the proposed adder architecture is 3 to 9 times faster than the corresponding CS3A 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.

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.

REFERENCES