IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 24, 2020, accepted July 15, 2020, date of publication July 28, 2020, date of current version August 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3012581

Double SHA-256 Hardware Architecture With Compact Message Expander for Bitcoin Mining

HOAI LUAN PHAM[1], (Graduate Student Member, IEEE), THI HONG TRAN[1], (Member, IEEE), TRI DUNG PHAN[1], VU TRUNG DUONG LE[2], DUC KHAI LAM[2], AND YASUHIKO NAKASHIMA[1], (Senior Member, IEEE)

[1]Graduation School of Information Science, Nara Institute of Science and Technology (NAIST), Ikoma 630-0192, Japan
[2]Computer Engineering Department, University of Information and Technology-Vietnam National University, Ho Chi Minh City 700000, Vietnam

Corresponding author: Thi Hong Tran (hong@is.naist.jp)

This work was supported by the Japan Science and Technology Agency (JST) under the Strategic Basic Research Programs PRESTO Grant number 2019A039.

ABSTRACT In the Bitcoin network, computing double SHA-256 values consumes most of the network energy. Therefore, reducing the power consumption and increasing the processing rate for the double SHA-256 algorithm is currently an important research trend. In this paper, we propose a high-data-rate low-power hardware architecture named the compact message expander (CME) double SHA-256. The CME double SHA-256 architecture combines resource sharing and fully unrolled datapath technologies to achieve both a high data rate and low power consumption. Notably, the CME algorithm utilizes the double SHA-256 input data characteristics to further reduce the hardware cost and power consumption. A review of the literature shows that the CME algorithm eliminates at least 9.68% of the 32-bit XOR gates, 16.49% of the 32-bit adders, and 16.79% of the registers required to calculate double SHA-256. We synthesized and laid out the CME double SHA-256 using CMOS 0.18 μm technology. The hardware cost of the synthesized circuit is approximately 13.88% less than that of the conventional approach. The chip layout size is 5.9 $mm \times 5.9$ mm, and the correctness of the circuit was verified on a real hardware platform (ZCU 102). The throughput of the proposed architecture is 61.44 Gbps on an ASIC with Rohm 180nm CMOS standard cell library and 340 Gbps on a FinFET FPGA 16nm Zynq UltraScale+ MPSoC ZCU102.

INDEX TERMS Bitcoin mining, SHA-256, unrolling, ASIC.

I. INTRODUCTION

Bitcoin is the most popular cryptocurrency and was invented by Satoshi Nakamoto in 2008 [1], [2]. Leveraging blockchain technology, Bitcoin uses a distributed public ledger to record all transactions without any third party [3]. Each block added to the public distributed ledger is created by hashing a 1024-bit message, including a version number, a hash of the previous block, a hash of the Merkle root, timestamp, target value, and a nonce. In the 1024-bit message, the nonce must be valid to create a hashing output smaller than the specified target value. Therefore, miners relentlessly seek valid nonces when adding new blocks. The process of finding a valid nonce is called Bitcoin mining [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Huang.

In Bitcoin mining, the double SHA-256 algorithm is used to compute the hash value of the bitcoin block header, which is a 1024-bit message. The use of double SHA-256 protects against the length extension attack [5]. Technically, SHA-256 consists of a message expander (ME) and a message compressor (MC). During the SHA-256 operation, the ME expands the 512-bit input message into 64 chunks of 32-bit data. The MC compresses these 64 32-bit data chunks into a 256-bit hashed output.

Most of the energy consumption required for maintaining the Bitcoin network stems from calculating double SHA-256 values. Therefore, reducing the hardware cost and energy consumption of the SHA-256 circuit is a popular research trend. In [6], the authors optimized the double SHA-256 operation for Bitcoin mining from an algorithmic perspective, but no hardware design was available to evaluate

the power consumption. From a hardware perspective, [7]–[22] proposed solutions to improve SHA-256. For instance, the authors of [7] employed the carry-save adder to improve the computation time of the critical path, which increased the maximum frequency and processing rate, while [8]–[12] used pipeline technology to improve the SHA-256 throughput. A cache memory technique was presented in [13] to reuse data, minimize the critical paths, and reduce the number of memory accesses for SHA-256 processing. The authors of [14] adopted the unfolding technique to reduce the computing latency for SHA-256. The authors of [15] proposed using a 7-3-2 array compressor to reduce the critical path delay for SHA-256. The carry-save adders technique is used in [16] to reduce the latency of additions in the SHA-256 algorithm. The authors of [17] used a combination of techniques such as carry-save-adders and pipelines to increase the performance of SHA-256. Pipeline and unrolled techniques are presented in [18] and [19] to increase the throughput of SHA-256. The authors of [20]–[22] presented a SHA-256 implementation on an FPGA for performance evaluation, with no technique optimization. Despite providing improvements in terms of hardware cost and power consumption, the hardware circuits developed in [7]–[22] have low processing rates because they require several (up to 64) clock cycles to compute a single 256-bit hash value.

To be applicable for Bitcoin mining, a SHA-256 circuit needs not only efficient hardware and power cost but also a high processing rate. To reach a high processing rate, the authors in [23] proposed the fully unrolled SHA-256 datapath for Bitcoin mining hardware. Additionally, the fully unrolled SHA-256 datapath can be designed to run on an application-specific integrated circuit (ASIC) [24], which can reach even higher processing rates. However, because an ASIC implementation of a fully unrolled datapath has high power consumption and hardware costs, [25]–[28] proposed eliminating an 8-round unrolled datapath in the double SHA-256 architecture to reduce the chip area. Furthermore, several technical solutions, such as carry-save adders and optimized message compressor (MC) architectures have been proposed and applied to reduce the hardware and power costs.

In this study, we propose a new approach for reducing the hardware cost and power consumption of high processing rate fully unrolled SHA-256 architecture. We analyze the characteristics of the 1024-bit input data of double SHA-256 and propose compact message expander (CME) algorithms that significantly reduce the hardware cost required to compute the message expander (ME) process of SHA-256. In addition, we propose a CME double SHA-256 accelerator architecture that adopts the proposed CME algorithms to reduce the power consumption. Our architecture generates one 256-bit hash value per clock cycle. We implemented the proposed double SHA-256 accelerator architectures in ASIC CMOS 0.18 μm technology to demonstrate their energy efficiency. The Verilog code and synthesized results of the experiment are publicly available from GitHub.

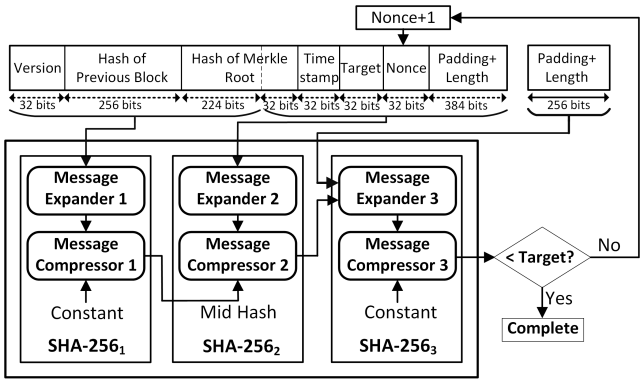

FIGURE 1. Overview architecture of double SHA-256 in Bitcoin Mining.

The remainder of this paper is organized as follows. Section II presents a preliminary study. Section III describes our proposed CME double SHA-256 architecture, and the CME algorithms and hardware circuits are explained in detail. Section IV reports our evaluation in terms of theory, ASIC, and FPGA experiments. Finally, Section V concludes the paper.

II. PRELIMINARIES

A. DOUBLE SHA-256 ARCHITECTURE FOR BITCOIN MINING

Fig. 1 shows the overview architecture of double SHA-256 applied for Bitcoin mining. The input to the double SHA-256 process is a 1024-bit message, which includes a 32-bit *version*, a 256-bit *hash of the previous block*, a 256-bit *hash of the Merkle root*, a 32-bit *timestamp*, a 32-bit *target*, a 32-bit *nonce*, and 384 bits of *padding*. The 1024-bit message is split into two 512-bit message parts; then SHA-256$_1$ calculates a hash value of the first 512-bit message, and SHA-256$_2$ computes a hash value of the final 512-bit message. Due to the double SHA-256 requirement, the 256-bit hash output from SHA-256$_2$ must be compressed into the final 256-bit hash by using SHA-256$_3$. In the Bitcoin mining process, the final 256-bit hash output from SHA-256$_3$ is compared to the target value. If the final hash is smaller than the target value, the valid 32-bit *nonce* is specified, and a new Bitcoin block is successfully created. Otherwise, the 32-bit *nonce* is increased by one and the double SHA-256 circuit recomputes to find a new hash value. This process is repeated until the 256-bit hash of SHA-256$_3$ meets the target requirement.

Computation inside all three blocks (SHA-256$_1$, SHA-256$_2$, and SHA-256$_3$) follows the SHA-256 algorithm, which has two processes: a message expander (ME) and a message compressor (MC).

Algorithm 1 shows the ME process, which expands the 512-bit input message into 64 chunks of 32-bit data W_j ($0 \leq j \leq 63$). In the first 16 rounds, the ME parses the 512-bit message into 16 32-bit data chunks (denoted as W_j, $j = 0$ to 15 where j is the round index). In the final 48 rounds, the ME calculates 48 chunks of 32-bit data W_j ($17 \leq j \leq 63$). Three 32-bit adders and two logical functions $\sigma_0(x)$ and $\sigma_1(x)$ are

Algorithm 1 Message Expander (ME)

- For j from 0 to 15 {
  $W_j = M_j$ }
- For j from 16 to 63 {
  $W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$ }

needed to compute each W_j ($17 \leq j \leq 63$) value. Fig. 2 shows the conventional circuit C required to calculate W_j ($17 \leq j \leq 63$), in which the logical functions $\sigma_0(x)$ and $\sigma_1(x)$ are respectively defined as follows:

$$\sigma_0(x) = S^7(x) \oplus S^{18}(x) \oplus R^3(x) \quad (1)$$

$$\sigma_1(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x) \quad (2)$$

Algorithm 2 shows the MC process, which compresses the 64 chunks of W_j ($0 \leq j \leq 63$) into a 256-bit hash value. The process involves three main steps: *initialization*, *loop*, and *add*. In the *initialization* step, eight internal hash values (denoted as a, b, c, d, e, f, g, h) are assigned to eight initial hashes H_1, H_2,...,H_8 defined by the SHA-256 algorithm. In the *loop* step, the internal hash values a, b, c, d, e, f, g, h are calculated and updated through 64 loops. To compute a, b, c, d, e, f, g, h in each loop, logical functions such as $\Sigma_0(x)$, $\Sigma_1(x)$, $Ch(x, y, z)$, and $Maj(x, y, z)$ are used.

$$\Sigma_0(x) = S^2(x) \oplus S^{13}(x) \oplus S^{22}(x) \quad (3)$$

$$\Sigma_1(x) = S^6(x) \oplus S^{11}(x) \oplus S^{25}(x) \quad (4)$$

$$Ch(x, y, z) = (x \wedge y) \oplus (\neg x \wedge z) \quad (5)$$

$$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z) \quad (6)$$

In the *add* step, the final hash is computed by adding the initial hashes H_1, H_2,...,H_8 to the final internal hashes a, b, c, d, e, f, g, h resulting from the 64 loops.

Algorithm 2 Message Compressor (MC)

(1) Initialization:
$a = H_1; b = H_2; c = H_3; d = H_4; e = H_5; f = H_6; g = H_7; h = H_8$

(2) Loop:
For j from 0 to 63 {
- $T_1 = h + \Sigma_1(e) + Ch(e, f, g) + K_j + W_j$
- $T_2 = \Sigma_0(a) + Maj(a, b, c)$
- h = g; g = f; f = e; e = d + T_1; d = c; c = b; b = a; a = $T_1 + T_2$ }

(3) Add:
$HO_1 = a + H_1; HO_2 = b + H_2; HO_3 = c + H_3;$
$HO_4 = d + H_4; HO_5 = e + H_5; HO_6 = f + H_6;$
$HO_7 = g + H_7; HO_8 = h + H_8;$

B. THE PROTOTYPE DOUBLE SHA-256 ARCHITECTURE

To be applicable for Bitcoin mining, double SHA-256 hardware should provide a high processing rate. The current optimal solution is to develop and implement a double SHA-256 accelerator in ASIC chips. In [23], the authors proposed

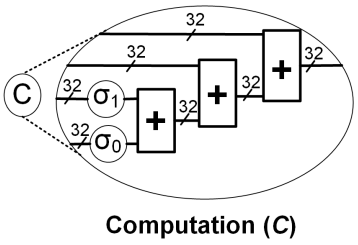

FIGURE 2. Conventional circuit C required for message expander (ME).

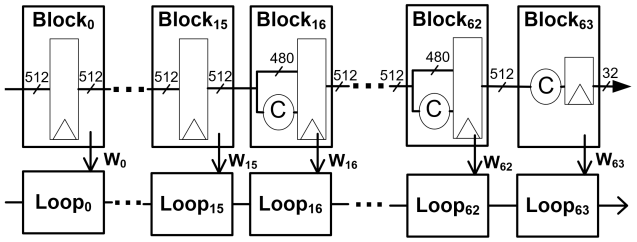

FIGURE 3. The Prototype 64-round unrolled datapath architecture for ME and MC processes of each SHA-256 circuit.

an ASIC-based double SHA-256 accelerator that implemented *ME* and *MC* processes in a fully unrolled datapath for high processing. Technically, the fully unrolled SHA-256 datapath enables the 64 rounds of *ME* and *MC* to run in parallel and be pipelined.

Fig. 3 illustrates a prototype SHA-256 architecture with 64-round unrolled datapaths for the *MC* and *ME* processes. The unrolled *ME* datapath is denoted as $Block_j$ ($j = 0, \ldots, 63$), while the unrolled *MC* datapath is denoted as $Loop_j$ ($j = 0, \ldots, 63$).

Because the goal of this study is to optimize the *ME* process, we focus specifically on a hardware implementation for *ME*. For the first 16 blocks (i.e., $Block_j$ ($j = 0, \ldots, 15$)), each ME block requires a 512-bit register (or 16 32-bit registers) to pipeline and store the 16 W_j ($j = 0, \ldots, 15$) values. For the last 48 blocks, i.e., $Block_j$ ($j = 16, \ldots, 63$), each block needs a 512-bit register (or 16 32-bit registers) and C circuits (Fig. 2) to compute W_j ($j = 16, \ldots, 63$). As shown in Fig. 1, the double SHA-256 accelerator for Bitcoin mining requires three individual SHA-256 circuits. This means that the accelerator must implement $48 \times 3 = 144$ C circuits (in the 16^{th} to 63^{th} blocks of SHA-256_1, SHA-256_2, and SHA-256_3). Thus, it is necessary to both optimize the C circuit and reduce the number of C circuits required for double SHA-256.

C. THE OPTIMIZED DOUBLE SHA-256 ARCHITECTURE

The prototype double SHA-256 accelerator has high power consumption because the fully unrolled datapath results in a large chip area. To reduce the power consumption, [25]–[28] proposed the optimized double SHA-256 accelerator, in which a 64-round unrolled datapath is optimized into a 60-round unrolled datapath.

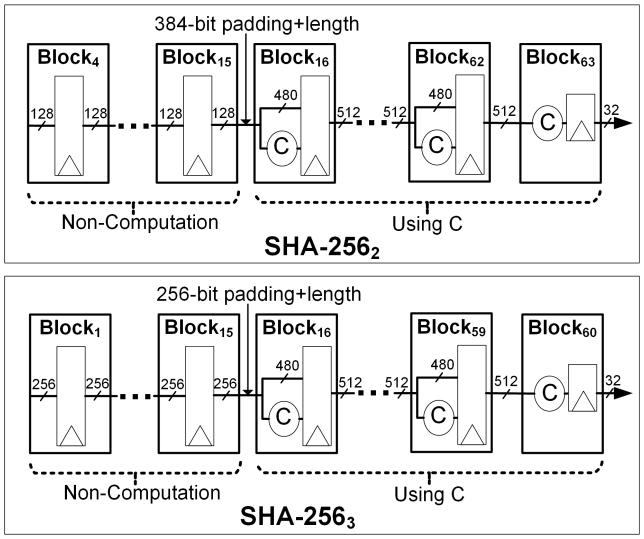

FIGURE 4. The optimized 60-round unrolled datapath architecture for the ME process of SHA-256$_2$ and SHA-256$_3$.

Fig. 4 shows a schematic diagram of the 60-round unrolled *ME* datapath used in SHA-256$_2$ and SHA-256$_3$. In SHA-256$_2$, the 60-round unrolled *ME* datapath includes rounds 4 to 63 (denoted as $Block_j$ ($j = 4, \ldots, 63$)). In SHA-256$_3$, the 60-round unrolled ME datapath includes rounds 1 to 60 (denoted as $Block_j$ ($j = 1, \ldots, 60$)). Consequently, 8 ME blocks are eliminated compared with the prototype architecture mentioned above.

III. THE PROPOSED CME DOUBLE SHA-256 ARCHITECTURE

A. ARCHITECTURAL OVERVIEW

In Bitcoin mining, the 512 bits of data input to SHA-256$_1$ does not change frequently because it does not include the 32-bit *nonce* field. Conversely, the 512 bits of data input to SHA-256$_2$ are updated frequently because of the changing value of the *nonce* field. Whenever the output of SHA-256$_2$ changes, SHA-256$_3$ also needs to be recomputed. Because the *nonce* field has 32 bits, each computation of SHA-256$_1$ requires SHA-256$_2$ and SHA-256$_3$ to recompute their values up to 2^{32} times.

Therefore, we propose the CME double SHA-256 accelerator architecture, as shown in Fig. 5. To achieve a high processing rate as well as efficient hardware and power cost, we implement a resource-sharing architecture for SHA-256$_1$ and a fully unrolled datapath architecture for SHA-256$_2$ and SHA-256$_3$. The SHA-256$_1$ has a single $Block_{0-63}$ circuit for calculating W_j ($j = 0, \ldots, 63$) and a single $Loop_{0-63}$ circuit for calculating the internal hashes a, b, c, d, e, f, h in 64 clock cycles. Each clock cycle computes one W_j value and updates the internal hash one time.

Similar to the conventional optimized double SHA-256 architecture, our SHA-256$_2$ has 60-round unrolled datapaths ($j = 4, \ldots, 63$), and our SHA-256$_3$ has 60-round

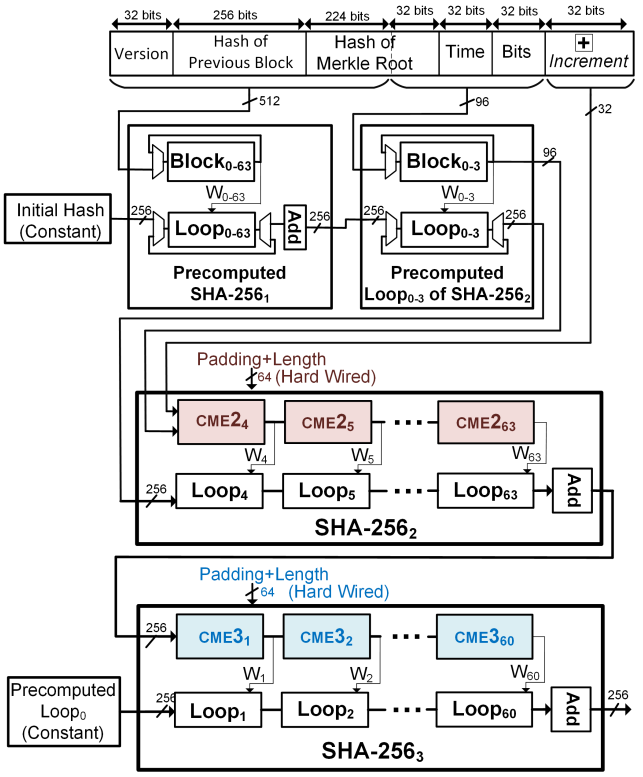

FIGURE 5. Block diagram of the proposed CME double SHA-256 accelerator for Bitcoin mining.

unrolled datapaths ($j = 1, \ldots, 60$). To reduce the hardware and power costs of SHA-256$_2$ and SHA-256$_3$, we propose using CME algorithms and their equivalent hardware circuits. In Fig. 5, the CME for SHA-256$_2$ is denoted as CME2$_j$ ($j = 4, \ldots, 63$), and the CME for SHA-256$_3$ is denoted as CME3$_j$ ($j = 1, \ldots, 60$).

Using pipelined and parallel operations, SHA-256$_2$ and SHA-256$_3$ can produce an output hash every clock cycle. However, the resource-sharing SHA-256$_1$ circuit produces one hash value every 64 clock cycles. The low processing rate of the SHA-256$_1$ circuit does not affect the final processing rate of the CME double SHA-256 accelerator because one SHA-256$_1$ output value can be used to calculate SHA-256$_2$ and SHA-256$_3$ up to 2^{32} times. The final processing rate of the CME-based double SHA-256 is one 256-bit hash value per clock cycle.

In the following subsections, we explain our proposed CME algorithms and the equivalent hardware designs.

B. COMPACT MESSAGE EXPANDER (CME) ALGORITHM

We propose the CME algorithms by analyzing the characteristics of the input data of SHA-256$_2$ and SHA-256$_3$.

1) CME FOR SHA-256$_2$

As seen in Fig. 1, the 512 bits of data input to SHA-256$_2$ include a 32-bit *Merkle root hash*, a 32-bit *time stamp*, a 32-bit *target*, a 32-bit *nonce*, and a 384-bit *padding+length*

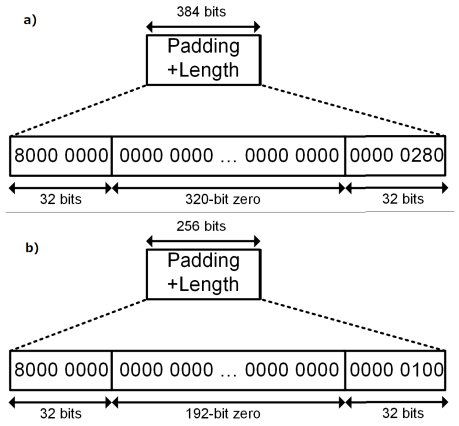

FIGURE 6. Contents of the padding+length field for SHA-256$_2$ (a), and SHA-256$_3$ (b).

field. It is worth noting that most of the content of the *padding+length* field consists of zeros (refer to Fig. 6a).

Assume that the 512 bits of data are separated into 16 32-bit words M_j ($j = 0, \ldots, 15$). The CME operation for SHA-256$_2$ is illustrated in Algorithm 3. The algorithm processes the data in 64 loops. During the first 16 loops, W_j ($j = 0, \ldots, 15$) are assigned to M_j ($j = 0, \ldots, 15$). The values of W_j ($j = 5, \ldots, 14$) are all zero because they are equivalent to the zero values of the *padding+length* field. In addition, W_4 and W_{15} are constants. During the last 48 loops, the CME calculates W_j ($j = 16, \ldots, 63$) by using (7):

$$W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}. \quad (7)$$

The logical functions $\sigma_0(x)$ and $\sigma_1(x)$ are shown in (1) and (2), respectively.

Utilizing the zeros or constant values of W_j ($j = 4, \ldots, 15$), we can optimize the calculation of (7). For example, the W_{16} calculation can be analyzed as follows:

$$\begin{aligned} W_{16} &= \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0 \\ &= 0 + 0 + \sigma_0(W_1) + W_0 \\ &= \sigma_0(W_1) + W_0 \end{aligned} \quad (8)$$

Note that $W_{14} = 0$ and $W_9 = 0$. By comparing (7) with (8) for calculating W_{16}, it can be seen that the logical function $\sigma_1(x)$ and two 32-bit adders have been eliminated.

The computations of $W_j (j = 17, \ldots, 63)$ are analyzed and optimized similarly. The final results are shown in Algorithm 3.

2) CME FOR SHA-256$_3$

The 512 bits of input data to SHA-256$_3$ include the 256-bit hash output from SHA-256$_2$ concatenated with a 256-bit *padding+length* field. The value of the first 32 bits of padding is $32'h80000000$, while the value of the last 32 bits *padding+length* is $32'h00000100$. The remaining values are all zeros (refer to Fig. 6b).

We divide the 512-bit input data into 16 32-bit words M_j ($j = 0, \ldots, 15$). The CME operation for SHA-256$_3$ is illustrated in Algorithm 4. It processes the data in 64 loops. In the first 16 loops, W_j ($j = 0, \ldots, 15$) are assigned to M_j ($j = 0, \ldots, 15$). The values of W_j ($j = 9, \ldots, 14$) are all zero because they are equivalent to the zero values of the *padding+length* field. In addition, W_8 and W_{15} are constants. In the last 48 loops, CME calculates W_j ($j = 16, \ldots, 63$) using (7).

Utilizing the zero or constant characteristics of W_j ($j = 8, \ldots, 15$), we optimize the calculation of (7) for calculating $W_j (j = 16, \ldots 63)$. The final results are shown in Algorithm 4.

Utilizing Algorithms 3 and 4, we can significantly reduce the number of 32-bit adders and the number of logical func-

Algorithm 3 Compact Message Expander in SHA-256$_2$

- For j from 0 to 3 {
  $W_j = M_j$ }
- $W_4 = 32$'h80000000
- For j from 5 to 14 {
  $W_j = 32$'h00000000 }
- $W_{15} = 32$'h00000280
- $W_{16} = \sigma_0(W_1) + W_0$
- For j from 17 to 19 {
  $W_j = \sigma_1(W_{j-2}) + \sigma_0(W_{j-15}) + W_{j-16}$ }
- $W_{20} = \sigma_1(W_{18}) + W_4$
- $W_{21} = \sigma_1(W_{19})$
- For j from 22 to 29 {
  $W_j = \sigma_1(W_{j-2}) + W_{j-7}$ }
- $W_{30} = \sigma_1(W_{28}) + W_{23} + \sigma_0(W_{15})$
- For j from 31 to 63 {
  $W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$ }

Algorithm 4 Compact Message Expander in SHA-256$_3$

- For j from 0 to 7 {
  $W_j = M_j^{(i)}$ }
- $W_8 = 32$'h80000000
- For j from 9 to 14 {
  $W_j = 32$'h00000000 }
- $W_{15} = 32$'h00000100
- $W_{16} = \sigma_0(W_1) + W_0$
- For j from 17 to 21 {
  $W_j = \sigma_1(W_{j-2}) + \sigma_0(W_{j-15}) + W_{j-16}$ }
- For j from 22 to 23 {
  $W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$ }
- $W_{24} = \sigma_1(W_{22}) + W_{17} + W_8$
- For j from 25 to 29 {
  $W_j = \sigma_1(W_{j-2}) + W_{j-7}$ }
- $W_{30} = \sigma_1(W_{28}) + W_{23} + \sigma_0(W_{15})$
- For j from 31 to 63 {
  $W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$ }

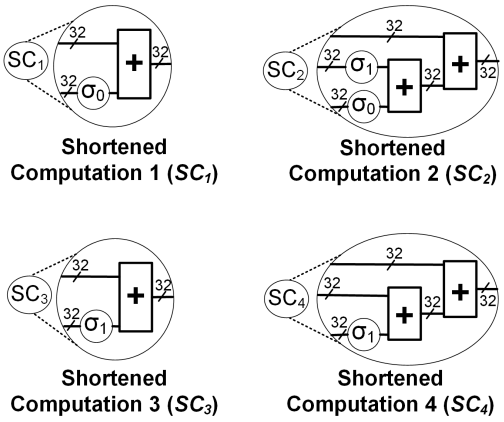

FIGURE 7. The proposed shortened computation circuits: SC_1, SC_2, SC_3, and SC_4 for the CME process.

tions $\sigma_0(x)$ and $\sigma_1(x)$ required to calculate W_{16} to W_{63} in SHA-256_2 and SHA-256_3.

C. CME HARDWARE CIRCUITS

From Algorithm 3 and 4, we propose four types of shortened computation (SC) circuits as shown in Fig. 7. Compared with the traditional C circuit shown in Fig. 2, the proposed SC_1 eliminates two 32-bit adders and the logical function $\sigma_1(x)$; SC_2 eliminates one 32-bit adder; SC_3 eliminates two 32-bit adders and the logical function $\sigma_1(x)$; and SC_4 eliminates one 32-bit adder and the logical function $\sigma_0(x)$. Note that eliminating either $\sigma_0(x)$ or $\sigma_1(x)$ also eliminates two 32-bit rotations, one 32-bit shift, and two 32-bit XOR circuits.

Based on the C circuit shown in Fig. 2 and the four types of SC circuits shown in Fig. 7, we develop hardware architectures for the CME processes of SHA-256_2 and SHA-256_3 as shown in Fig. 8 and Fig. 10, respectively.

The proposed CME circuit for SHA-256_2 (Fig. 8) is divided into three phases. Phase 1 includes $CME2_4$ to $CME2_{19}$. Each operation requires a 128-bit register (or four 32-bit registers) to store and pipeline W_0 to W_3. In phase 1, instead of using the conventional C circuit in Fig. 2, the SC_1 and SC_2 circuits in Fig. 7 are implemented to reduce hardware costs. Phase 2 includes $CME2_{20}$ to $CME2_{30}$, for which the SC_2 and SC_3 circuits are appropriately implemented (refer to algorithm 3). Phase 3 includes $CME2_{31}$ to $CME2_{63}$, and the C circuit is implemented in all the blocks of this phase.

The three phases are classified based on the characteristics of the datapath bit width. In phase 1, the datapath bit-width is constant (128 bits). The 384-bits of W_4 to W_{15} are fixed constants. Hence, phase 1 do not need to store and pipeline W_4 to W_{15}. In phase 2, W_{20} to W_{30} must be stored and pipelined. Thus, the datapath bit-width in phase 2 is appropriately increased from 160 bits to 480 bits. In phase 3, the datapath bit-width of $CME2_{31}$ to $CME2_{57}$ is 512 bits without optimization. To eliminate unnecessary values of W_j in subsequent blocks, the datapath bit-width of $CME2_{57}$ to $CME2_{63}$ appropriately reduces from 480 bits to 32 bits. To understand the reason for the datapath bit-width adjustment,

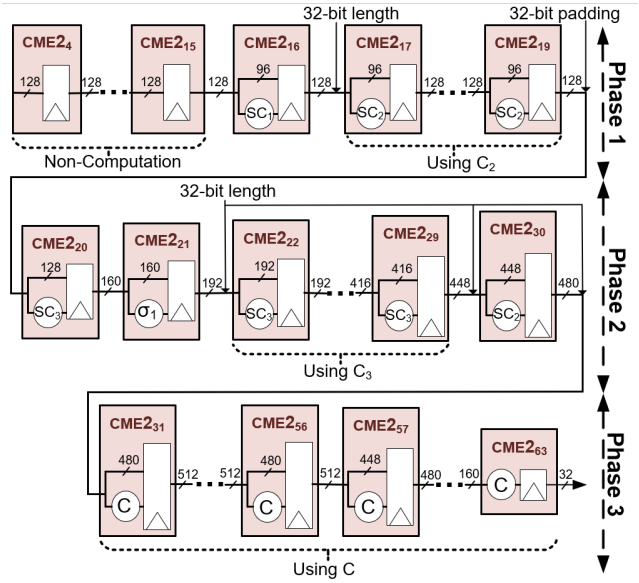

FIGURE 8. Block diagram of the 60-round unrolled datapath CME2 process for SHA-256_2.

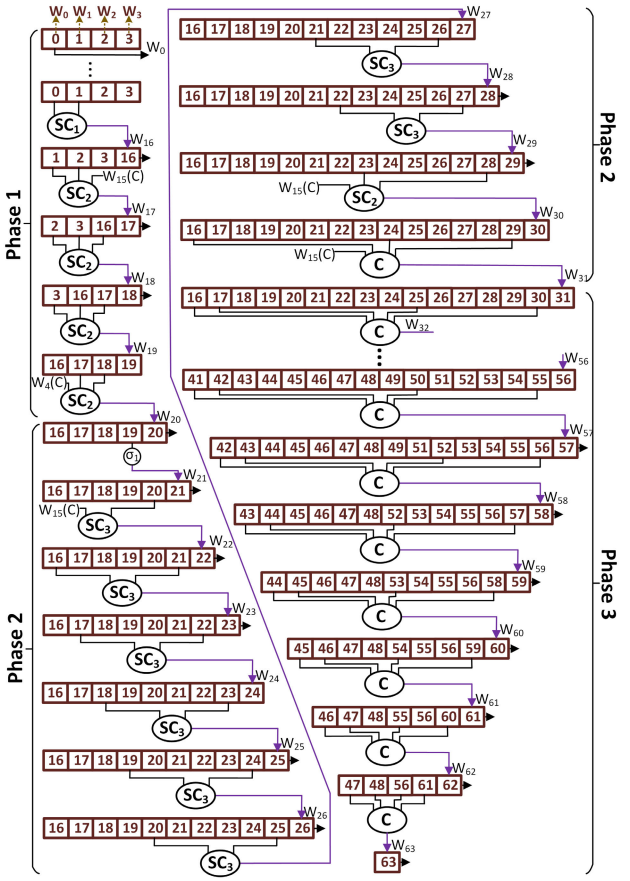

FIGURE 9. Detailed computational circuit of the CME2 process for SHA-256_2.

we show the detailed data flow and computational circuit of the CME2 process in Fig. 9. In this figure, the number represents the j index of W_j. For example, we need four 32-bit

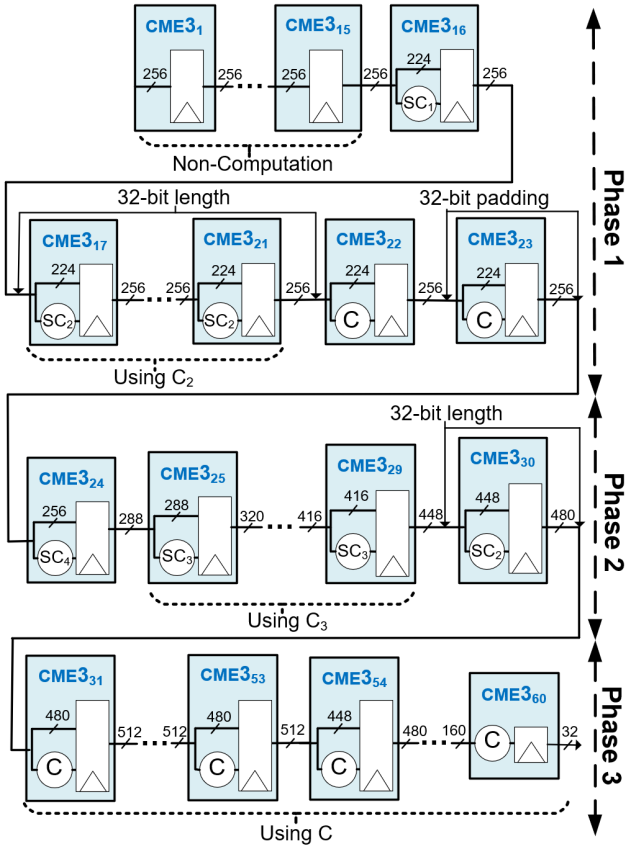

FIGURE 10. Block diagram of the 60-round unrolled datapath CME3 process for SHA-256$_3$.

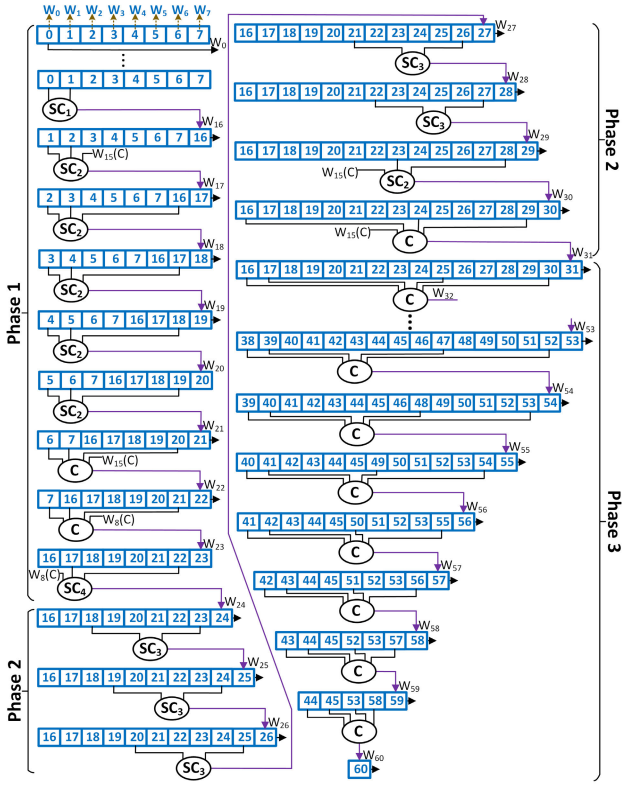

FIGURE 11. Detailed computational circuit of the CME3 process for SHA-256$_3$.

registers (equivalent to 128 bits) to store W_0 to W_3 in blocks CME2$_4$ to CME2$_{15}$. As another example, CME2$_{32}$ needs sixteen 32-bit registers ($16 \times 32 = 512$ bits) to pipeline store 16 values of W_j (j=16, 17,...,31), which are required for the calculation of its following blocks.

Similarly, the proposed CME circuit for SHA-256$_3$ has three phases (Fig. 10). Phase 1 includes CME3$_1$ to CME3$_{23}$. Because of the zero and constant property of input data W_8 to W_{15}, all blocks of phase 1 have the same datapath of 256 bits only (which is required to pipeline store eight 32-bit values W_0 to W_7). A large number of registers are thus eliminated. In this phase, circuits SC$_1$, SC$_2$, and C are appropriately implemented (refer to algorithm 4). Phase 2 includes blocks from CME3$_{24}$ to CME3$_{30}$. Circuits SC$_4$, SC$_3$, and SC$_2$ are appropriately implemented (refer to algorithm 4). Phase 3 includes blocks from CME3$_{31}$ to CME3$_{60}$. We do not implement blocks from CME3$_{61}$ to CME3$_{63}$ because we can detect early whether the final hash is smaller than the target value without waiting for results from CME3$_{61}$ to CME3$_{63}$. Circuit C is implemented in all blocks.

Three phases are classified based on the characteristics of the datapath bit-width. In phase 1, the datapath bit-width is constant (256 bits).The 256-bits of W_8 to W_{15} are fixed constants and do not need to be stored and pipelined in phase 1. In phase 2, W_{24} to W_{30} must be stored and pipelined. Therefore, the datapath bit-width of CME3$_{24}$ to CME3$_{30}$ is appropriately increased from 288 bits to 480 bits. In phase 3, the datapath bit-width of CME3$_{31}$ to CME3 $_{53}$ is 512 bits without optimization. The datapath bit-width of CME3$_{54}$ to CME3 $_{60}$ is reduced from 480 bits to 32 bits. To prove that the datapath bit-width adjustment is appropriate, we show the detailed data flow and the computational circuit of the CME3 process in Fig. 11. In this figure, the number represents the j-th index of W_j. For example, each block from CME3$_0$ to CME2$_{15}$ requires eight 32-bit registers (equivalent to $8 \times 32 = 256$ bits) to store W_0 to W_7. These values are required to calculate the blocks from CME3$_{16}$ to CME2$_{22}$. As another example, block CME3$_{59}$ requires five 32-bit registers ($5 \times 32 = 160$ bits) to store W_{44}, W_{45}, W_{53}, W_{58}, and W_{59}, which are required for the CME3$_{60}$ calculation.

IV. EVALUATION

In this section, we evaluate the efficiency of the CME method when it is applied in the CME double SHA-256 accelerator. We evaluate the performance from three aspects: theory, ASIC, and FPGA experimental results.

A. THEORETICAL REVIEW

For comparison purposes, we developed three hardware circuits, all of which follow the architecture proposed in Fig. 5. The three circuits differ only in how they implement the ME processes of SHA-256$_2$ and SHA-256$_3$. The first circuit (named Prototype double SHA-256) was proposed in [23] and mentioned in section II-B. The second circuit (named

TABLE 1. Theory comparison: hardware-resource required for ME process.

Stage	Resource	Architecture		
		Prototype	Optimized	**Proposed**
$SHA\text{-}256_2$	32-bit adders	144	144	**117**
	32-bit XOR gates	192	192	**170**
	32-bit Rotation	240	240	**217**
	32-bit Shift	48	48	**38**
	32-bit registers	1009	801	**651**
$SHA\text{-}256_3$	32-bit adders	144	135	**116**
	32-bit XOR gates	192	180	**166**
	32-bit Rotation	240	225	**210**
	32-bit Shift	48	45	**39**
	32-bit registers	1009	825	**697**
$SHA\text{-}256_2$ + $SHA\text{-}256_3$	32-bit adders	288	279	**233**
	32-bit XOR gates	384	372	**336**
	32-bit Rotation	480	465	**427**
	32-bit Shift	96	93	**77**
	32-bit registers	2018	1620	**1348**

Optimized double SHA-256) was proposed in [25]–[27], and [28], and is mentioned in Section II-C. The last circuit is our proposed CME double SHA-256.

Table 1 shows the theoretical hardware resources required by the three architectures in terms of the number of adders, XOR gates, rotations, shifts, and registers. In Table 1, $SHA\text{-}256_2$ and $SHA\text{-}256_3$ are the evaluation targets because they are the most hardware-intensive parts.

Compared to the prototype and optimized architectures, the proposed architecture respectively decreases the total number of 32-bit adders by approximately 19.1% and 16.49%, the total number of 32-bit XOR gates by approximately 12.5% and 9.68%, and the total number of 32-bit rotation operations, by approximately 11% and 8.17%.

In addition, the proposed architecture reduces the total number of 32-bit shift operations by approximately 19.8% and 17.2% compared to the prototype and optimized architectures, respectively.

Notably, the proposed architecture eliminates 33.2% and 16.79% of the total number of registers compared to the prototype and optimized architectures, respectively.

B. ASIC EXPERIMENT

1) AREA AND POWER APPROACH

To ensure a fair comparison, the three double SHA-256 circuits were coded in Verilog and synthesized in an ASIC using the Synopsys Design Compiler with the Rohm $0.18\mu m$ CMOS standard cell library [29]. Table 2 shows the synthesized area of the three architectures. Note that the total area is the sum of the combinational and non-combinational area (registers), as well as other types of circuits, including buff/Inv, wires, etc. The total area of the proposed

TABLE 2. Practical comparison: The ASIC synthesized area of three *double SHA-256* architectures.

Design	Combinational Area (μm^2)	Non-combinational Area (μm^2)	Total Area (μm^2)
Prototype	5,741,967	6,716,928	13,823,243
Optimized	5,480,520	6,428,772	13,224,169
Proposed	**4,876,523**	**5,409,863**	**11,388,986**

	Prototype	Optimized	**Proposed**
Cell Internal Power (mW)	158	151	**133**
Net Switching Power (mW)	108	105	**95**

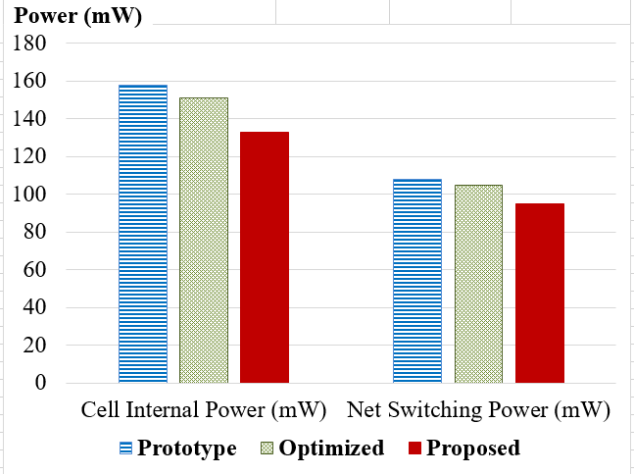

FIGURE 12. The ASIC synthesis power of the prototype, the optimized, and the proposed double SHA-256 accelerators.

CME double SHA-256 is smaller by 17.6% and 13.9% compared to the prototype and optimized architectures, respectively.

Fig. 12 summarizes the energy consumption of the three architectures obtained from the ASIC synthesis results. In terms of cell internal power, the proposed double SHA-256 circuit consumes 133 mW, which is a reduction of 15.82% and 11.92% compared to the prototype and optimized architectures, respectively. In terms of net switching power, the proposed CME double SHA-256 circuit consumes 95 mW, which constitutes reductions of 12.04% and 9.52% compared to prototype and optimized architectures, respectively. These energy consumption reductions are due to the smaller hardware circuit, which matches our expectations.

Based on the timing report of ASIC synthesis, the maximum frequency of the three architectures is 60 MHz. This means that the architectures achieve throughput of 1024 bits $\times$ 60 MHz $=$ 61.44 Gbps.

In addition, we successfully laid out the proposed CME double SHA-256 circuit in ASIC technology with the Rohm $0.18\mu m$ CMOS standard cell library. Fig. 13 shows the chip layout, and Fig. 14 shows the chip energy distribution map. The size of the chip layout is 5.9 mm $\times$ 5.9 mm.

TABLE 3. A comparison of ASIC synthesis results.

Technology	Design	Frequency (MHz)	No. of Cycles		Throughput R_d (Gbps)	Area (μm^2)	Hardware Efficiency E_d (Kbps/μm^2)
			Single SHA-256	Double SHA-256			
SMIC 0.18μm	[15]	208	65	195	1.092	211,955	5.16
UMC 0.18μm	[16]	302	66	198	1.562	185,256	8.44
TSMC 0.18μm	[17]	380	65	195	1.995	562,704	3.54
	Prop.	**92**	**-**	**1**	**94.208**	**9,038,148**	**10.42**

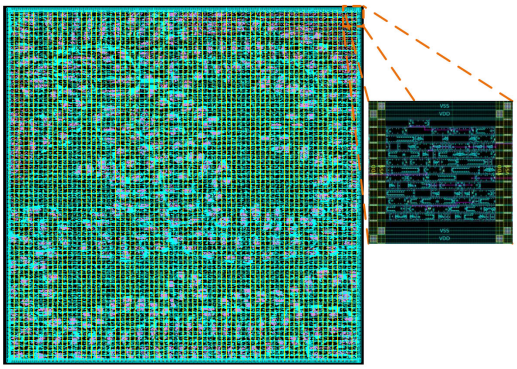

FIGURE 13. Post-layout circuit of the proposed CME double SHA-256 accelerator.

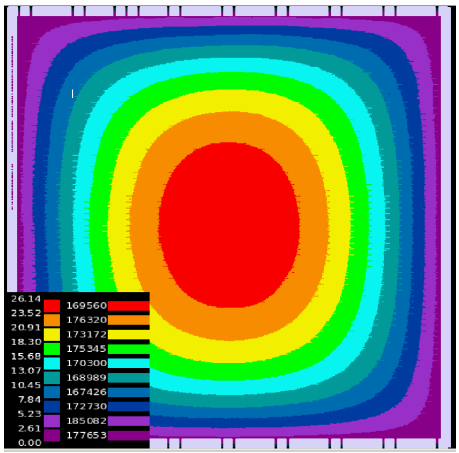

FIGURE 14. Energy distribution map of the post-layout *CME double SHA-256* circuit.

2) PROCESSING RATE AND HARDWARE EFFICIENCY APPROACH

In this experiment, we prove that the ASIC design of our proposed CME double SHA-256 architecture outperforms previous works in terms of processing rate and hardware efficiency. To ensure a fair comparison, we also synthesized our architecture in ASIC TSMC 0.18μm technology using the CMOS standard cell library. We then compare our results with the previous works in [15], [16], and [17].

The comparison is shown in Table 3. It is worth noting that the designs of [15], [16], and [17] are single SHA-256 circuits. To be applied to Bitcoin mining, these circuits must repeat their calculations three times to generate a double SHA-256 hash value from the 1024-bit input message. The number of cycles required to compute the double SHA-256 (denoted by C_d) is thus triple the number of cycles required to compute a single SHA-256 (denoted by C_s); refer to (9).

$$C_d = 3 \times C_s \tag{9}$$

Then, we calculate the processing rate for double SHA-256 (R_d) by using (10). The *BlockSize* is 1024 bits.

$$R_d = \frac{BlockSize \times Frequency}{C_d} \tag{10}$$

From the R_d and area results, the hardware efficiency for double SHA-256 (denoted by E_d) is computed by (11).

$$E_d = \frac{R_d}{Area} \tag{11}$$

Table 3 summarizes the synthesized area results, the calculated processing rate, and the hardware efficiency. The processing rate (R_d) and hardware efficiency (E_d) of our proposed architecture are significantly improved compared to those of the works in [15], [17], and [16]. The numerical results are as follows.

In terms of processing rate (R_d), our CME double SHA-256 architecture is faster than the designs proposed in [15], [16], and [17] by 86, 60, and 47 times, respectively.

In terms of hardware efficiency (E_d), our CME double SHA-256 architecture improves the efficiency by 102%, 23%, and 194% compared to the designs in [15], [16], and [17], respectively.

3) FPGA SYNTHESIS RESULTS

To ensure a fair comparison with other existing SHA-256 architectures, such as [18]–[21], [22], and [23], we synthesized the proposed CME double SHA-256 circuit on four Xilinx FPGA boards, including Kintex UltraScale (XCKU5P-ffva676-3-e), Virtex 7(XC7VX1140T-FLG 1926-2), Artix 7 (XC7A200T-FBG484-1), and Zynq UltraScale+ ZCU102 (XCZU9EG-FFVB1156-2-e).

The results are shown in Table 4. It is worth noting that the existing architectures in [18]–[22] and [23] are single SHA-256 architectures that must repeat the computation three times to generate a double SHA-256 hash value for Bitcoin mining. Thus, the number of clock cycles required

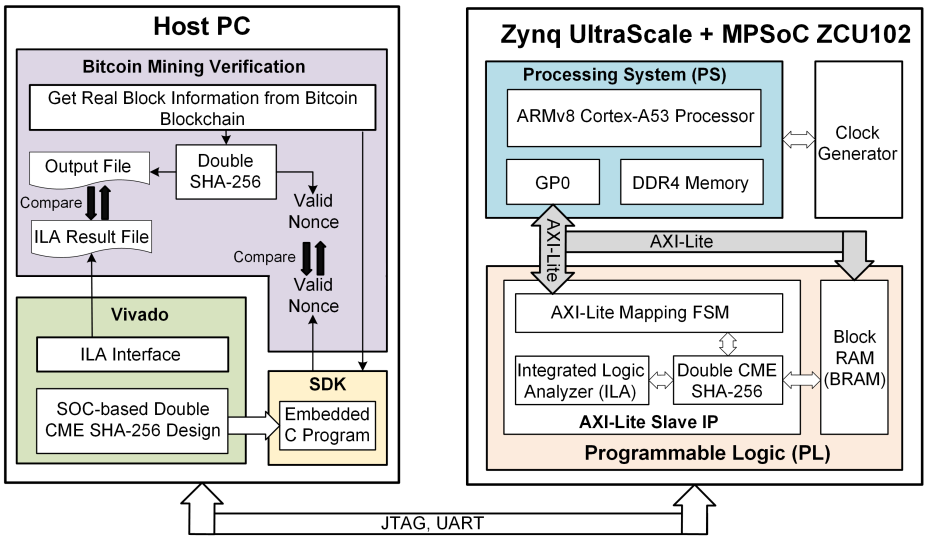

FIGURE 15. Diagram of the SoC-based CME double SHA-256 system for implementation and verification.

TABLE 4. A comparison of FPGA synthesis results.

FPGA Device	Design	Frequency (MHz)	No. of Cycles		Throughput R_d (Gbps)	Area		Hardware Efficiency E_d (Mbps/LUT)
			Single SHA-256	Double SHA-256		LUTs	FFs	
Kintex Ultrascale+ XCKU5P	[18]	364.3	16	48	7.772	9,316	4,188	0.83
	[19]	377.9	16	48	8.062	4,986	4,312	1.62
	[23]	428.1	-	1	438.374	49,144	54,674	8.92
	Prop.	**428.1**	**-**	**1**	**438.374**	**46,053**	**52,428**	**9.52**
Virtex 7 XC7VX1140T	[20]	367	65	195	1.928	350	-	5.5
	[21]	196.1	65	195	1.03	1,505	-	0.68
	[23]	263.2	-	1	269.516	49,104	54,674	5.48
	Prop.	**263.2**	**-**	**1**	**269.516**	**46,013**	**52,428**	**5.86**
Artix 7 XC7A200T	[22]	174	65	195	0.914	1,359	1,618	0.68
	[23]	132.5	-	1	135.168	49,104	54,674	2.76
	Prop.	**132.5**	**-**	**1**	**135.168**	**46,013**	**52,428**	**2.94**
Zynq Ultrascale+ ZCU102	[23]	374.2	-	1	383.18	49,145	54674	7.8
	Prop.	**374.2**	**-**	**1**	**383.18**	**46,013**	**52,428**	**8.32**

to compute a double SHA-256 is tripled. We focus on evaluating the hardware efficiency (Mbps/LUT) of the single and double SHA-256 architectures in this subsection. In general, the proposed CME double SHA-256 outperforms the existing SHA-256 architectures in terms of hardware efficiency. The numerical results are as follows.

On the Kintex UltraScale FPGA, the hardware efficiency (Mbps/LUT) of the proposed CME double SHA-256 architecture is enhanced by 1,047% (9.52 vs. 0.83), 488% (9.52 vs. 1.62), and 7% (9.52 vs. 8.92) compared to the hardware efficiencies of the architectures in [18], [19], and [23], respectively.

On the Virtex 7 FPGA, the hardware efficiency of the proposed architecture is enhanced by 7% (5.86 vs. 5.5), 762% (5.86 vs. 0.68), and 7% (5.86 vs. 5.48) compared to the the hardware efficiencies of the architectures in [20], [21], and [23], respectively.

On the Artix 7 FPGA, the hardware efficiency of the proposed architecture is enhanced by 332% (2.94 vs. 0.68) and 7% (2.94 vs. 2.76) compared to the the hardware efficiencies of the architectures in [22] and [23], respectively.

On Zynq UltraScale+ ZCU102 FPGA, the hardware efficiency of the proposed architecture is enhanced by 7% (8.32 vs. 7.8) compared to the hardware efficiency of the architecture in [23].

C. FPGA EXPERIMENT

1) FUNCTIONAL VERIFICATION ON A REAL SoC HARDWARE PLATFORM

To prove that the circuit operates correctly not only in the software simulation tool but also on real hardware, we built a System on Chip (SoC) platform to execute the proposed CME double SHA-256 circuit. The SoC platform overview is shown in Fig. 15.

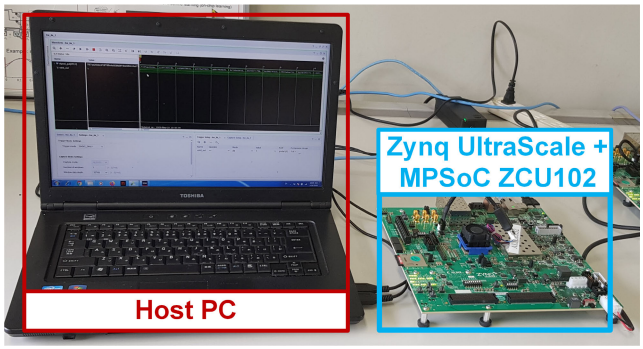

FIGURE 16. Real-world experiments with the SoC-based CME double SHA-256 devices.

TABLE 5. Execution time of double SHA-256 on different hardware platforms.

Device	Execution time (millisecond)				
	100,000 outputs	200,000 outputs	300,000 outputs	400,000 outputs	500,000 outputs
CPU Core i7-6950X@3.50GHz (10 cores), Memory128GB	150	310	460	620	770
CPU XEON6144x2 (16cores) + V100, Memory 256GB	140	290	380	560	740
GPU NVIDIA Tesla V100-PCIE, Memory 16GB	27	64	82	110	140
FPGA Zynq UltraScale+ MPSoC ZCU102 (333MHz clock input)	0.3	0.6	0.9	1.2	1.5

The platform includes two primary components: a host PC and a Zynq UltraScale+ ZCU102 evaluation board. The host PC exchanges data with the ZCU102 board via JTAG and UART cables.

The ZCU102 board includes an ARMv8 microprocessor, a programmable logic (PL), and a clock generator. Our developed circuit, CME double SHA-256, is embedded in the PL of ZCU102. The PL also has block ram (BRAM) and an integrated logic analyzer (ILA). We used the BRAM to store the valid nonce value for Bitcoin mining and ILA to monitor the outputs of the CME double SHA-256 circuit. The maximum operating frequency of the ZCU102 board is 333 MHz.

The host PC consists of a Vivado, a Software Development Kit (SDK), and a Bitcoin Mining Verification (BMV) program. Vivado is a software suite for SoC development. We use the Vivado suite to design and load the SoC-based system onto the Zynq UltraScale+ ZCU102 board. Moreover, the Vivado helps to export the outputs of the CME double SHA-256 circuit in the ZCU102 into an ILA result file for verification by the BMV program. The SDK is intended for the development of embedded software applications for SoC systems. We use the SDK to embed the real block information from the Bitcoin blockchain network onto our SoC-based system. The BMV is a C-code program that verifies the correctness of the embedded CME double SHA-256 circuit. The BMV executes a double SHA-256 on the host PC and compares the results with the outputs of the CME double SHA-256 circuit.

TABLE 6. Hash rate and power consumption of several SHA-256 architectures.

Research	Device	Hash rate (MHash/s)	Power (W)
[30]	CPU Intel I7-2600K@ 3.3GHz, 4 Cores	1.9	-
	GPU NVIDIA GeForce GTX 550 TI	20.4	-
[31]	CPU Intel Core i7-990x@ 3.46GHz, 6 Cores	33	-
	GPU NVIDIA GeForce GTX 570	155	-
[32]	CPU Intel Core i7 i7-950@ 3.06GHz	18.9	150
	GPU ATI Radeon HD 5770	214.5	108
	ASIC Block Erupter Sapphire	333	2.55
This work	**FPGA Zynq UltraScale+ ZCU102 (333MHz)**	**333**	**2.49**

The abovementioned SoC system has been used to thoroughly verify the correctness of the CME double SHA-256 circuit at different operating frequencies, such as 333 MHz (maximum frequency) and 200 MHz. All the cases result in 100% accuracy, which proves that the proposed CME double SHA-256 architecture works correctly in a real hardware platform. The maximum processing rate of the circuit on the ZCU102 board is 333 MHash/s (or 333 MHz $\times$ 1024 bit/CLK $=$ 340.992 Gbps).

Fig. 16 shows an image of the SoC evaluation platform, which includes a host PC (Toshiba Satellite B652 / G Core i5 3320M 2.6GHz / 4GB) and the UltraScale+ ZCU102 evaluation board.

2) PROCESSING-RATE EVALUATION ON A REAL HARDWARE PLATFORM

In this subsection, we evaluate the processing rate and power consumption of the proposed CME double SHA-256 on real hardware platform ZCU102 to prove that our architecture outperforms other high-performance platforms, including CPUs, GPUs, and the existing SHA-256 architectures.

Table 5 shows the execution time of the double SHA-256 algorithm on several hardware platforms, including a CPU, GPU, and FPGA. To compute the same number of hashes (e.g., 500.000 hashes) the proposed architecture running on the FPGA ZCU102 requires only 1.5 ms, while the CPU i7-6950X, CPU XEON 6144, and GPU Tesla V100 require 770 ms, 740 ms, and 140 ms, respectively,

which means that the proposed architecture reduces the execution time by 513 times, 493 times, and 93 times, respectively.

Table 6 summarizes the hash rate and power consumption from several studies that reported double SHA-256 results. As the table shows, the hash rate of our proposed architecture running on an FPGA is significantly higher than those of the works in [30] and [31]. Although [32] was executed on an ASIC and our architecture was executed on an FPGA, our architecture still achieves the same hash rate but consumes less power.

V. CONCLUSION

Bitcoin mining is an important process in keeping the Bitcoin network secure; however, it consumes massive amounts of energy. To reduce the power consumption and increase the processing rate of the Bitcoin mining process, we proposed a CME double SHA-256 hardware circuit in this paper. The architecture includes three SHA-256 circuits in which the first circuit (SHA-256_1) is a resource-sharing architecture while the last two circuits (SHA-256_2 and SHA-256_3) are fully unrolled datapath architectures. The combination of these two types of architecture results in a high processing rate but low hardware costs. Specifically, we propose several compact message expander (CME) algorithms and associated hardware architectures to further reduce the power consumption and hardware costs. Our proposed circuit generates one 256-bit hash value per clock cycle. We thoroughly verified and evaluated the proposed circuit on both ASIC and FPGA platforms. The experimental results showed that the proposed circuit outperforms other high-performance CPU and GPU platforms for computing double SHA-256 values. The proposed circuit also outperforms existing works with specific hardware circuits for computing the double SHA-256 values. The double SHA-256 circuit was laid out on the ASIC with Rohm 0.18 μm CMOS standard cell library, resulting in a chip size of 5.9 $mm \times 5.9$ mm and the throughput of 61.44 Gbps. The circuit is also proven to work correctly in a real hardware platform (ZCU102), achieving a processing rate of 340.992 Gbps.

Blockchain is not only the Bitcoin network. Blockchain technology is outgrowing in its potential to be applied in many fields of life, such as smart health care, autonomous cars, and supply chains. Other blockchain networks may employ not only SHA-256 but also other cryptography hash functions, such as SHA-512 or SHA-3. Therefore, developing a flexible and programmable accelerator that can compute several hash functions is a future need. By developing a low-cost low-power-consumption blockchain accelerator, we help to enhance the security and decentralized features of the blockchain network. Therefore, we believe that developing a blockchain accelerator that can compute multiple cryptography hash functions at low cost and with low power consumption will be an important research trend in the near future.

APPENDIX

The Verilog code and the synthesized results of the prototype, optimized, and proposed architectures can be found at https://github.com/archlab-naist/Double-CME-SHA256/

REFERENCES

[1] P. D. DeVries, "An analysis of cryptocurrency, bitcoin, and the future," *Int. J. Bus. Manage. Commerce*, vol. 1, no. 2, pp. 1–9, 2016.

[2] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*. Manubot, Nov. 2019.

[3] M. Rahouti, K. Xiong, and N. Ghani, "Bitcoin concepts, threats, and machine-learning security solutions," *IEEE Access*, vol. 6, pp. 67189–67205, 2018.

[4] H. Vranken, "Sustainability of bitcoin and blockchains," *Current Opinion Environ. Sustainability*, vol. 28, pp. 1–9, Oct. 2017.

[5] J. Taskinsoy, "Bitcoin and turkey: A good match or a perfect storm," *SSRN Electron. J.*, Oct. 2019, doi: 10.2139/ssrn.3477849.

[6] N. T. Courtois, M. Grajek, and R. Naik, "Optimizing SHA256 in bitcoin mining," in *Proc. Int. Conf. Cryptogr. Secur. Syst.*, Berlin, Germany: Springer, 2014, pp. 131–144.

[7] X. Zhang and H. Hu, "Optimization of hash function implementation for bitcoin mining," in *Proc. 3rd Int. Conf. Mechatronics Eng. Inf. Technol. (ICMEIT)*, 2019.

[8] M. D. Rote, V. N, and D. Selvakumar, "High performance SHA-2 core using the round pipelined technique," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Jul. 2015, pp. 1–6.

[9] L. Dadda, M. Macchetti, and J. Owen, "The design of a high speed ASIC unit for the hash function SHA-256 (384, 512)," in *Proc. Design, Autom. Test Eur. Conf. Exhib.*, Feb. 2004, pp. 70–75.

[10] M. Padhi and R. Chaudhari, "An optimized pipelined architecture of SHA-256 hash function," in *Proc. 7th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Dec. 2017, pp. 1–4.

[11] X. Zhang, R. Wu, M. Wang, and L. Wang, "A high-performance parallel computation hardware architecture in ASIC of SHA-256 hash," in *Proc. 21st Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2019, pp. 52–55.

[12] S. B. Suhaili and T. Watanabe, "Design of high-throughput SHA-256 hash function based on FPGA," in *Proc. 6th Int. Conf. Electr. Eng. Informat. (ICEEI)*, Nov. 2017, pp. 1–6.

[13] R. García, I. Algredo-Badillo, M. Morales-Sandoval, C. Feregrino-Uribe, and R. Cumplido, "A compact FPGA-based processor for the secure hash algorithm SHA-256," *Comput. Electr. Eng.*, vol. 40, no. 1, pp. 194–202, Jan. 2014.

[14] H. Michail, G. Athanasiou, A. Kritikakou, C. Goutis, A. Gregoriades, and V. Papadopoulou, "Ultra high speed SHA-256 hashing cryptographic module for ipsec hardware/software codesign," in *Proc. Int. Conf. Secur. Cryptogr. (SECRYPT)*, Jul. 2010, pp. 1–5.

[15] L. Bai and S. Li, "VLSI implementation of high-speed SHA-256," in *Proc. IEEE 8th Int. Conf. ASIC*, Oct. 2009, pp. 131–134, doi: 10.1109/ASICON.2009.5351591.

[16] S. Tillich, M. Feldhofer, M. Kirschbaum, P. Thomas, S. Jörn-Marc, and S. Alexander, "High-speed hardware implementations of BLAKE, blue midnight wish, CubeHash, ECHO, Fugue, Grøstl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, and Skein," *IACR Cryptol. ePrint Arch.*, vol. 510, 2009.

[17] F. Opritoiu, S. L. Jurj, and M. Vladutiu, "Technological solutions for throughput improvement of a secure hash algorithm-256 engine," in *Proc. IEEE 23rd Int. Symp. Design Technol. Electron. Packag. (SIITME)*, Oct. 2017, pp. 159–164.

[18] R. Martino, and A. Cilardo, "A flexible framework for exploring, evaluating, and comparing SHA-2 designs," *IEEE Access*, vol. 7, pp. 72443–72456, 2019.

[19] R. Martino, and A. Cilardo, "A configurable implementation of the SHA-256 hash function," in *Proc. Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput.*, Cham, Switzerland: Springer, Nov. 2019, pp. 558–567.

[20] F. Kahri, B. Bouallegue, M. Machhout, and R. Tourki, "An FPGA implementation and comparison of the SHA-256 and blake-256," in *Proc. 14th Int. Conf. Sci. Techn. Autom. Control Comput. Eng. STA*, Dec. 2013, pp. 152–157, doi: 10.1109/STA.2013.6783122.

[21] R. Florin and R. Ionut, "FPGA based architecture for securing IoT with blockchain," in *Proc. Int. Conf. Speech Technol. Hum.-Comput. Dialogue (SpeD)*, Oct. 2019, pp. 1–8, doi: 10.1109/SPED.2019.8906595.

[22] D. K. N. and R. Bhakthavatchalu, "Parameterizable FPGA implementation of SHA-256 using blockchain concept," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2019, pp. 0370–0374, doi: 10.1109/ICCSP.2019.8698069.
[23] J. Barkatullah and T. Hanke, "Goldstrike 1: CoinTerra's first-generation cryptocurrency mining processor for bitcoin," *IEEE Micro*, vol. 35, no. 2, pp. 68–76, Mar. 2015, doi: 10.1109/MM.2015.13.
[24] M. Vilim, H. Duwe, and R. Kumar, "Approximate bitcoin mining," in *Proc. 53rd Annu. Design Autom. Conf. DAC*, Jun. 2016, pp. 1–6.
[25] V. B. Suresh, S. K. Satpathy, and S. K. Mathew, "Energy-efficient bitcoin mining hardware accelerators," U.S. Patent 10 313 108 B2, Jun. 4, 2019.
[26] V. B. Suresh, S. K. Satpathy, and S. K. Mathew, "Optimized SHA-256 datapath for energy-efficient high-performance Bitcoin mining," U.S. Patent 1 014 098 B2, Nov. 27, 2018.
[27] V. B. Suresh, S. K. Satpathy, and S. K. Mathew, "Bitcoin mining hardware accelerator with optimized message digest and message scheduler datapath," U.S. Patent 2 018 008 642 A1, Mar. 29, 2018.
[28] V. Suresh, S. Satpathy, R. Kumar, M. Anders, H. Kaul, A. Agarwal, S. Hsu, R. Krishnamurthy, V. De, and S. Mathew, "A 250Mv, 0.063J/Ghash bitcoin mining engine in 14nm CMOS featuring dual-vcc Sha256 datapath and 3-Phase latch based clocking," in *Proc. Symp. VLSI Circuits*, Jun. 2019, pp. C32–C33.
[29] T. H. Tran, S. Kanagawa, D. P. Nguyen, and Y. Nakashima, "ASIC design of MUL-RED Radix-2 pipeline FFT circuit for 802.11ah system," in *Proc. IEEE Symp. Low-Power High-Speed Chips (COOL CHIPS XIX)*, Apr. 2016, pp. 1–3.
[30] J. Anish Dev, "Bitcoin mining acceleration and performance quantification," in *Proc. IEEE 27th Can. Conf. Electr. Comput. Eng. (CCECE)*, May 2014, pp. 1–6, doi: 10.1109/CCECE.2014.6900989.
[31] M. Bedford Taylor, "The evolution of bitcoin hardware," *Computer*, vol. 50, no. 9, pp. 58–66, 2017, doi: 10.1109/MC.2017.3571056.
[32] S. Ghimire and H. Selvaraj, "A survey on bitcoin cryptocurrency and its mining," in *Proc. 26th Int. Conf. Syst. Eng. (ICSEng)*, Dec. 2018, pp. 1–6, doi: 10.1109/ICSENG.2018.8638208.

HOAI LUAN PHAM (Graduate Student Member, IEEE) received the bachelor's degree in computer engineering from the University of Information Technology-Vietnam National University Ho Chi Minh City (VNU-HCM), Vietnam, in 2018. He is currently pursuing the M.S. degree with the Nara Institute of Science and Technology (NAIST), Japan. His research interests include blockchain technology and cryptography.

THI HONG TRAN (Member, IEEE) received the bachelor's degree in physics and the master's degree in microelectronics from the Vietnam National University-Ho Chi Minh University of Science (VNU-HCMUS), Vietnam, in 2008 and 2012, respectively, and the Ph.D. degree in information science from the Kyushu Institute of Technology, Japan, in 2014. Since 2015, she has been with the Nara Institute of Science and Technology (NAIST), Japan, as a Full Time Assistant Professor. Her research interests include digital hardware circuit design, algorithm relate to wireless communication, communication security, blockchain technologies, SHA-2, SHA-3, cryptography, and so on. She is a Regular Member of the IEEE, IEICE, REV-JEC, and so on.

TRI DUNG PHAN received the Engineer degree in computer engineering (hardware design) from Vietnam National University Ho Chi Minh City-University of Information Technology (VNUHCM-UIT), in 2019. He is currently pursuing the M.S. degree in Nara Institute of Science and Technology (NAIST), Japan. His research interests include secure hash algorithm (SHA) in Hardware Design, such as FPGA and ASIC design.

VU TRUNG DUONG LE received the bachelor's degree in IC and hardware design from the Vietnam National University Ho Chi Minh City-University of Information Technology, in 2020. His research interests include blockchain technologies, deep learning, cryptography, and so on.

DUC KHAI LAM received the B.E. and M.S. degrees from the University of Science, Vietnam National University Ho Chi Minh City (VNU-HCM), in 2006 and 2011, respectively, and the Ph.D. degree from the Kyushu Institute of Technology, Japan, in 2016. He is currently with the University of Information Technology, VNU-HCM, as a Lecturer and a Researcher. His research interests include wireless communication systems, digital signal processing, ASIC, and VLSI design.

YASUHIKO NAKASHIMA (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in computer engineering from Kyoto University, in 1986, 1988, and 1998, respectively. He was a Computer Architect with the Computer and System Architecture Department, Fujitsu Ltd., from 1988 to 1999. From 1999 to 2005, he was an Associate Professor with the Graduate School of Economics, Kyoto University. Since 2006, he has been a Professor with the Graduate School of Information Science, Nara Institute of Science and Technology. His research interests include computer architecture, emulation, circuit design, and accelerators. He is a Fellow of IEICE, a Senior Member of IPSJ, and a member of the IEEE CS and ACM.