VOL. 24, NO. 1, 2025, 41-46
elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

# Power-Efficient ASIC Implementation of Double SHA-256 for Proof of Work Mining

**Lim Yong Fong**[1], **Shahidatul Sadiah**[1*], **Ab Al-Hadi Ab Rahman**[1], **Muhammad Mun'im Ahmad Zabidi**[1] and **Mohd Usairy Syafiq**[2]

[1]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.
[2]Quality Engineering Research Cluster, UniKL MITEC, Bandar Seri Alam, 81750 Masai, Johor, Malaysia.

[*]Corresponding author: shahidatulsadiah@utm.my

**Abstract:** In this paper, a design space exploration is performed aiming at implementing a low power and high-performance hardware architecture for the double SHA-256 accelerator with optimized message scheduler. Furthermore, this study also explores the power optimization of this architecture by enabling clock gating during logic synthesis and using High Voltage Threshold (HVT) standard cells for layout implementation. The 32nm version of the SAED process design kits (PDK) was used for the ASIC implementation of the double SHA-256 hash function. The result shows that the combination of using HVT standard cell and disabling clock gating achieved the most-balanced trade-off between performance and power consumption. The resulting design could obtain a throughput of 187.9 $Gbps$ at the frequency of 183.5 $MHz$ with power consumption of 131.9 $mW$.

**Keywords:** Double SHA-256, Hash functions, low-power ASIC



*Article History: received 2 November 2024; accepted 12 February 2025; published 30 April 2025*

## 1. INTRODUCTION

Bitcoin uses Proof of Work (PoW) as its consensus mechanism. The main idea is that a computational task which requires intensive resources must be done before a change is proposed to the network. The proposal will be subsequently verified before being committed to deter any potential system abuse by parties with malicious intent. For example, Hashcash [1] was proposed as a system to combat email spam using PoW. The system works by requiring the sender to compute a hash fulfilling a certain target and attach it as a stamp to the header of the email prior to sending it. The computation of the hash discourages the spammers because they must invest a certain amount of CPU power for each email they want to send. However, the recipient can easily verify the stamp as the process costs significantly less for the recipient.

The hash functions used in Hashcash and Bitcoin are SHA-1 and SHA-256 respectively. A hash function generates a fixed-length output based on an input with finite, but arbitrary length. For security applications, the function should be non-reversible where it is infeasible to find an input to generate a given hash value. It also must be resistant to collision where it is difficult to find two distinct values such that they will generate the same hash value. The hash function in PoW demands significant energy consumption from the user who propose a new block to the blockchain network. This lies in the fact that the target requirement imposed on the block's header is difficult to achieve and therefore countless trial and error are needed before the user successfully find a header which satisfies the difficulty target. The machine must continuously crunch the data, and it consumes a great amount of energy in the process. According to Cambridge Bitcoin Electricity Consumption Index [2], as of 18 Sep 2021, the estimated power demand is 11.65 $GW$, and the annualized consumption is around 100 $TWh$, comparable to the electricity consumption of a country.

There have been numerous designs proposed [3]-[9] to increase the throughput of the algorithm, but few focus on optimizing the power consumption of the hash function. Most literature focuses on improving the performance of the hash function because in cryptocurrency, whoever obtains valid hash first will be rewarded while the other must move on and work on other blocks of transactions. Therefore, miners want equipment with the highest performance possible that is profitable when factoring in the electricity costs. The objectives of this project are to implement a double SHA-256 hashing accelerator in ASIC which is used in some variants of PoW consensus algorithms. Then, the design is assessed and evaluated in terms of performance in terms of throughput and power consumption. Improvement of the power consumption of double SHA-256 implementation are made and the trade-off between power and performance is analyzed.

The architecture chosen to be implemented for SHA-256 hash function in this study was the double SHA-256 with compact message expander proposed by Pham et al [9]. This architecture has an optimized message scheduler that enabled the model to exhibit higher throughput but with scarification on the circuit area and the power consumption. In our work, architecture design is implemented in four different flavors using clock gating

technique and standard cell libraries with different threshold voltages to determine the extent of its impact on the design's power consumption.

## 2. BACKGROUND

SHA-256 is a hash algorithm which takes an input of less than 264 bits and generates a 256-bit-long output called message digest [10]. The algorithm is described in three stages: preprocessing, message scheduling, and message compression, as shown in Figure 1. In the preprocessing stage, a bit "1" is first appended to the end of the input message, followed by $k$ zero bits, where $k$ is the non-negative and smallest solution to the equation 448 mod 512 = $l + k + 1$, and $l$ is the length of the input message. Then, a 64-bit block which is the binary number representation of $l$ is appended, which results in a padded message with length of multiples of 512 bits. Next, this message is parsed into $N$ blocks of 512-bit. These 512-bit blocks, $Mi$ will serve as inputs to the message scheduling stage.

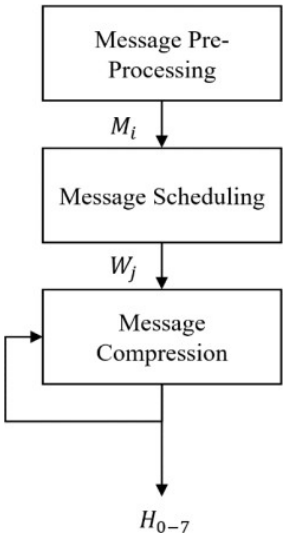


Figure 1. The three stages of SHA-256 hash algorithm.

After pre-processing, the message blocks $Mi$ are processed sequentially. First, eight hash values $H0$ are initialized following the specification in the standard (For the first message block, they are the fractional parts of the square roots of the first eight primes. They are determined differently for the subsequent message blocks, as shown in Figure 3). Then, the message scheduler generates a 32-bit data $Wj$ from the padded message and sends it to the message compression function. The message schedule is used along with constants, rotate right, and shift right functions, and functions to calculate a series of hash values iteratively. This is repeated 64 times for each 512-bit padded message block. The message digest is then determined by the final hash value generated. The flowcharts of message scheduling and message compression stages are presented in Figure 2 and Figure 3 respectively. In these figures, there are six logical functions defined in SHA-256, each operating on a 32-bit values:

$$\sigma 0(x) = \mathrm{ROR}(x, 7) \oplus \mathrm{ROR}(x, 18) \oplus x \gg 3 \quad (1)$$

$$\sigma 1(x) = \mathrm{ROR}(x, 17) \oplus \mathrm{ROR}(x, 19) \oplus x \gg 10 \quad (2)$$

$$S0(x) = \mathrm{ROR}(x, 2) \oplus \mathrm{ROR}(x, 13) \oplus \mathrm{ROR}(x, 22) \quad (3)$$

$$S1(x) = \mathrm{ROR}(x, 6) \oplus \mathrm{ROR}(x, 11) \oplus \mathrm{ROR}(x, 25) \quad (4)$$

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

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

Equations (1) to (6) are the functions defined in the SHA-256 standard to permute a given input repeatedly so that a unique hash corresponding to the input data can be generated.

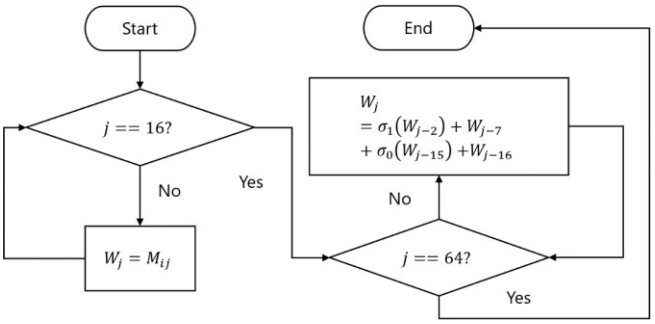


Figure 2. The flowchart of message scheduling stage in SHA-256 hash function.

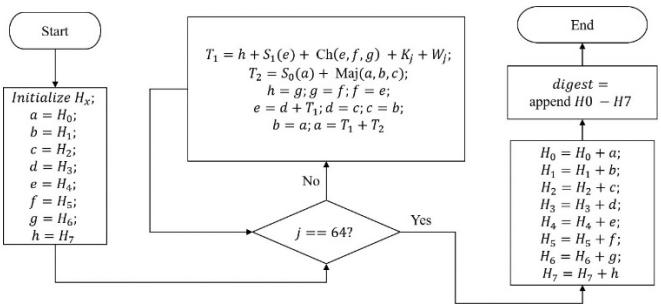


Figure 3. The flowchart of message compression stage in SHA-256 hash function.

## 3. PROOF OF WORK MINING ALGORITHM

For Bitcoin mining, the overview structure of double SHA-256 [5] is shown in Figure 4. The input is the 640-bit-long string block header that consists of six fields: version, previous block header hash, Merkle root hash, time, an encoded version of current target threshold, and nonce. The rest 384 bits of the 1024-bit-long input are the bit padding, as specified by the SHA-256 standard. The Stage-0 and Stage-1 is the first invocation of the hash function, which processes the first 512 bits and the second 512 bits of the input message respectively. Then, the output of Stage-1 SHA-256 is fed into Stage-2 to generate the final message digest. The output of this double SHA-256 function is compared against the target value, if it satisfies the difficulty requirement, then this block is successfully created in the Bitcoin network. Otherwise, the nonce is incremented and the hash is calculated again.

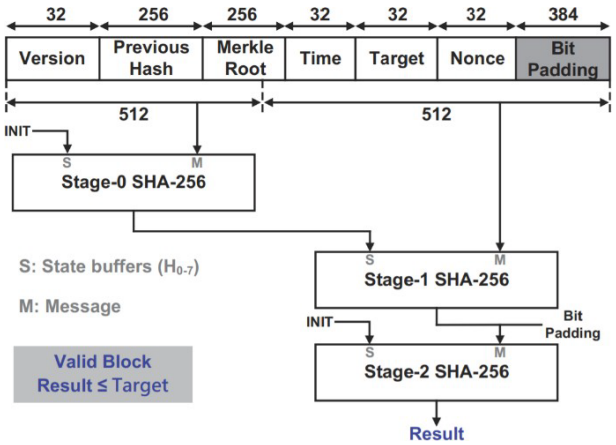


Figure 4. The structure of double SHA-256 in Bitcoin mining [5].

Pham et al. [9] proposed a double SHA-256 hardware architecture with compact message expander focusing on the optimization of message scheduler function. The design consists of three SHA-256 blocks. Each of the SHA-256 blocks comprises a message scheduler and a message compressor. The second and third SHA-256 blocks have 60-round unrolled message expander datapath [11]. By exploiting the input data characteristics of the second and third SHA-256 blocks, four different types of shortened computation circuits are proposed. The design of these four shortened circuits relies on the fact that some parts of the input remain constant throughout the computation, specifically the last 384 bits and the last 256 bits of the second and third SHA-256 blocks' inputs respectively.

## 4. DESIGN OF EXPERIMENTS

To verify the implementation of this work on SAED 32nm standard cell library, we initially incorporate the architecture of a double SHA-256 accelerator with an optimized message scheduler [9]. The technology employed is Synopsys's SAED 32nm static CMOS technology, operating at a nominal voltage of 0.95 $V$. The combination of full loop unrolling, and pipeline registers inserted between every iteration stage would result in a high throughput. By fully unrolling the kernel, the module can perform calculations in all iterations in parallel. Next, deep pipelining would allow different nonce values to be processed in different iteration stages down the pipeline at the same instant as shown in Figure 5. Therefore, these optimizations should allow the design to generate an output every clock cycle. The trade-offs are on the layout area and power due to the growth of resources such as combinational logic and flip-flops.

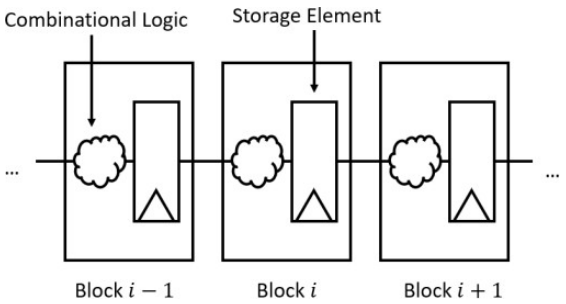


Figure 5. Combination of loop unrolling (as shown by duplicated iteration blocks) and deep pipelining (storage elements in each block) in the message scheduler process.

On top of that, to further optimize the power usage of double SHA-256 accelerator design, clock gating technique and utilization of standard cell libraries with different threshold voltage are explored. Clock gating aims to prevent clock signals from being propagated to components when there is no change in the output [12]. On the other hand, limiting the target library to standard cell with high threshold voltage will help in reducing leakage power, which is becoming increasingly a concern as the process node grows smaller [13]. Therefore, utilization of HVT standard cell library is expected to limit the leakage power while employing clock gating will reduce clock network power and hence the design's total power. The impact of these two variables on the critical path delay and hence the performance of the designs is also investigated. All the four possible different combinations explored in this project are LVT without clock gating, LVT with clock gating, HVT without clock gating, and HVT with clock gating. To specify the standard cell library being used by Design Compiler during synthesis, the target_library application variable is used. The target_library variable tells Design Compiler which library to use for mapping during logic synthesis. On the other hand, the implementation of clock gating in the design is controlled by the presence of gate_clock switch when invoking the compile_ultra command. Once the implementation was done, the data below were collected:

- Area occupied by standard cells ($\mu m2$)
- Number of standard cells (combinational and sequential cells)
- Timing information such as critical path delay ($ns$)
- Internal, leakage and switching power ($mW$)

The first two data, area occupied by standard cells and number of standard cells, are reported, and obtained from ICC. The last two data, critical path delay and power consumption, are collected in PrimeTime. The timing information from the STA report was used to estimate the throughput of the design. A comparison was made between not only the baseline and optimized designs, but also among the four different flavors for each of them.

## 5. EXPERIMENTAL RESULTS AND DISCUSSION

The throughput-optimized version of the SHA-256 hash function [9] were first implemented without clock gating optimization. The target library used in the layout

implementation was LVT standard cell libraries. The implementation result is presented in Table 1. The latency was calculated by measuring the number of clock cycles. In other words, two rounds for the 1024-bit-long input in the first SHA-256 function call (one round for each of the two 512-bit-long message blocks), then another round for the second SHA-256 function call operating on the output of the first SHA-256 function.

Table 1. Implementation result of design [9] on 32nm library.

| Design | Optimized |
|---|---|
| Layout Dimension ($\mu m^2$) | 1441 × 1440 |
| Total Cell Count | 286953 |
| Total Cell Area ($\mu m^2$) | $1.2 \times 10^6$ |
| Total Signal Routing Length ($\mu m$) | $9.5 \times 10^6$ |
| Critical Path Delay ($ns$) | 5.41 |
| Latency (clock cycles) | 188 |
| Operating Frequency ($MHz$) | 196.08 |
| Throughput ($Gbps$) | 200.78 |

With its loops fully unrolled and the datapath pipelined at every stage, the design achieves a throughput of 200.78 Gbps at a clock frequency of 196.08 MHz. This comes at the expense of significant value in both the total area occupied by cells and power consumption. The detailed power report was tabulated in Table 2. Since most of the cells in the design are storage elements, they consume almost half the power.

Table 2. Power report of design [9] implementation in $mW$

| | Total Power |
|---|---|
| Clock Network | 118.4 (19.12%) |
| Register | 289.1 (46.67%) |
| Combinational | 211.9 (34.21%) |

To improve area and power, two optimization techniques are further explored using standard cells with high voltage threshold and employing clock gating during synthesis. The throughput-optimized designs were implemented in different combinations of both techniques and the physical layout information for the four variants of the optimized design is tabulated in Table 3.

Table 3. Implementation result using different combinations of techniques.

| Design | LVT without clock gating | LVT with clock gating | HVT without clock gating | HVT with clock gating |
|---|---|---|---|---|
| Total Cell Count | 286953 | 190950 | 298179 | 203565 |
| Sequential Cell Count | 77273 | 77451 | 77273 | 77414 |
| Combinational Cell Count | 209680 | 113499 | 220906 | 126151 |
| Total Cell Area ($\mu m^2$) | 1173694.43 | 924275.22 | 1278897.59 | 1009293.25 |
| Sequential Cell Area ($\mu m^2$) | 560346.86 | 558018.90 | 564303.12 | 560802.79 |
| Combinational Area ($\mu m^2$) | 613347.57 | 366256.32 | 714594.47 | 448490.46 |
| Total Signal Routing Length ($\mu m$) | 9463175.27 | 7291692.47 | 9650177.92 | 7789681.20 |
| Critical Path Delay ($ns$) | 5.41 | 5.52 | 5.45 | 5.74 |

Enabling clock gating in synthesis decreased the total cell count by 33% on average, and consequently the total area occupied by standard cells reduced by 21%. Changing the standard cell voltage threshold from low to high had a modest effect on the total cell area, which was increased by 8.96% and 8.42% for non-clock-gating and clock-gating designs respectively. The area occupied by sequential cells (i.e., flip-flop) remained relatively constant throughout the four variants, with a difference of only 1.11% between the biggest and smallest values. In contrast, the area occupied by combinational cell was strongly affected by whether the design adopted clock gating or not. Therefore, when clock gating was enabled, the reduction in combination cell count contributed the most to the reduction in total cell area. The reduction in total signal routing length accompanied the decrease in total cell area as most signals did not have to travel long distances to connect the pins.

Table 4. Power consumption ($mW$) of different optimization techniques and its categorization.

| Design | LVT without clock gating | LVT with clock gating | HVT without clock gating | HVT with clock gating |
|---|---|---|---|---|
| Clock Network | 118.4 | 59.3 | 91.8 | 46.6 |
| Register | 289.1 | 289.0 | 12.4 | 13.4 |
| Combinational | 211.9 | 151.1 | 27.7 | 20.8 |
| Net Switching Power | 25.9 | 14.1 | 26.0 | 15.6 |
| Cell Internal Power | 101.5 | 49.6 | 82.0 | 45.2 |
| Cell Leakage Power | 492.0 | 435.7 | 24.0 | 20.0 |
| Total Power | 619.5 | 499.5 | 131.9 | 80.8 |

The breakdown in power consumption by types of cells and power is tabulated in Table 4. It is evident that there was a considerable increase in total power when standard cells with low threshold voltage was used instead of high threshold voltage. The total power consumption when using HVT standard cells was 83.8% and 78.7% lower as compared to using LVT standard cells for the cases with and without clock gating respectively. Register consumed the most power when LVT cells were used in the designs, as high as 57.9% in the design with clock gating enabled. However, in the designs utilizing HVT cells, the power consumption was contributed mostly by cells in the clock network. The significant decrease in the power consumed by register from 289.1 $mW$ to 12.4 $mW$ when switching from LVT to HVT helped to reduce the overall power consumption in the cases without clock gating. By contrast, the implementation of clock gating had marginal effect on the power consumed by the register, whereas it did help in bringing down the power consumed by both clock network cells and combinational cells. With clock gating, the power consumed by the cells in clock network decreased by roughly 50% in both LVT and HVT designs.

The cell leakage power dominated the power consumption in the LVT implementations, while cell internal power contributed the most in the HVT designs. This was expected as a cell with lower threshold voltage leaks more easily than its counterpart with higher threshold voltage. A MOS transistor partially conducts electricity even if the applied voltage is below the threshold voltage, which is known as subthreshold conduction. Unfortunately, because the leakage current is an exponential function of the parameter $VTh$ , decreasing the threshold voltage results in a significant increase of subthreshold current, and consequently, the dramatic increase in leakage power. On the other hand, enabling clock gating in the design helped to reduce both net switching power and cell internal power roughly by half.

To reduce the power consumption without sacrificing the performance, the design with HVT and no clock gating is the best among all the candidates. It has the second lowest critical path delay of 5.45 $ns$ while only consuming 131.9 $mW$. Although it occupies 21% more area by the design with HVT and clock gating, the lower critical path delay warrants the increase in circuit area, due to the performance requirement of the double SHA-256 hash function. The critical path delay of 5.45 $ns$ implies that the maximum operating frequency is around 183.5 $MHz$, so the throughput is calculated as 187.9 $Gbps$.

## 6. CONCLUSION

In this research, a double SHA-256 accelerator with optimized message scheduler was successfully implemented. The design has a deep pipeline and fully unrolled architecture, which allowed it to achieve a throughput of 200.78 $\boldsymbol{Gbps}$ at a frequency of 196 MHz. Due to the extensive resources, the total cell area is significant. Thus, this study explored two optimization techniques which are using high threshold voltage standard cells and employing clock gating during synthesis. In summary, it was found that using HVT but disabling clock gating optimization allowed the designer to obtain a balanced trade-off between power consumption and performance. This combination was able to acquire the second shortest critical path delay of 5.45 ns while reducing the power consumption by 80%. It had a larger area than a corresponding design with clock gating optimization enabled, by 26.7%, nonetheless the increase in area is justified by the performance requirement. As enabling clock gating would increase the critical path delay by 290ps, which may lower the device maximum operating frequency.

## ACKNOWLEDGMENT

The authors would like to thank the Ministry of Higher Education for the funding of this project, with the Fundamental Research Grant number FRGS/1/2020/TK0/UTM/02/62 and Efinix Technology (M) Sdn Bhd for the UTM Contract Research DTD grant with grant number R.J130000.7623.4C669.

## REFERENCES

[1] Back A. Hashcash-a denial of service counter-measure. 2002.

[2] Cambridge Bitcoin Electricity Consumption Index: Cambridge Centre for Alternative Finance; 2021 (cited 2021 Sep 18).

[3] Kim M, Ryou J, Jun S, editors. Efficient hardware architecture of SHA-256 algorithm for trusted mobile computing. International Conference on Information Security and Cryptology; 2008: Springer.

[4] Trusted Platform Module (TPM) Summary: Trusted Computing Group; [cited 2021 Sep 19]. Available from: https://trustedcomputinggroup.org/resource/trusted-platform-module-tpm- summary/.

[5] Wang Y, Wu J, Chen S, Chao MC, Yang C, editors. Micro-Architecture Optimization for Low- Power Bitcoin Mining ASICs. 2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT); 2019 22-25 April 2019.

[6] Vilim M, Duwe H, Kumar R, editors. Approximate bitcoin mining. 2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC); 2016 5-9 June 2016.

[7] Barkatullah J, Hanke T. Goldstrike 1: CoinTerra's First-Generation Cryptocurrency Mining Processor for Bitcoin. IEEE Micro. 2015;35(2):68-76.

[8] Li J, He Z, Qin Y, editors. Design of Asynchronous High Throughput SHA-256 Hardware Accelerator in 40nm CMOS. 2019 IEEE 13th International Conference on ASIC (ASICON); 2019 29 Oct.-1 Nov. 2019.

[9] Pham HL, Tran TH, Phan TD, Le VTD, Lam DK, Nakashima Y. Double SHA-256 hardware architecture with compact message expander for bitcoin mining. IEEE Access. 2020; 8:139634-46.

[10] Pub NF. 180-2. Secure Hash Standard", National Institute of Standards and Technology, US Department of Commerce. 2002.

[11] Suresh VB, Satpathy SK, Mathew SK. Optimized SHA-256 datapath for energy-efficient high-performance Bitcoin mining. Google Patents; 2018.

[12] S TC, Shanmugasundaram N, editors. Clock Gating Techniques: An Overview. 2018 Conference on Emerging Devices and Smart Systems (ICEDSS); 2018 2-3 March 2018.
[13] Abbas Z, Olivieri M. Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells. Microelectronics Journal. 2014;45(2):179-95.