DOI: 10.64302/joshusc.v31n1e.1340

THIẾT KẾ BỘ XỬ LÝ ĐA LÕI TĂNG TỐC CHO THUẬT TOÁN SHA-256 SỬ DỤNG BỘ NHỚ CỤC BỘ VÀ TÍNH TOÁN SONG SONG TOÀN PHẦN

Phan Văn Đại*, Khổng Thị Thu Thảo

Khoa Điện, Điện tử & Công nghệ vật liệu, Trường Đại học Khoa học, Đại học Huế

*Email: vandai2995@gmail.com

Ngày nhận bài: 17/9/2025; ngày hoàn thành phản biện: 12/10/2025; ngày duyệt đăng: 16/10/2025

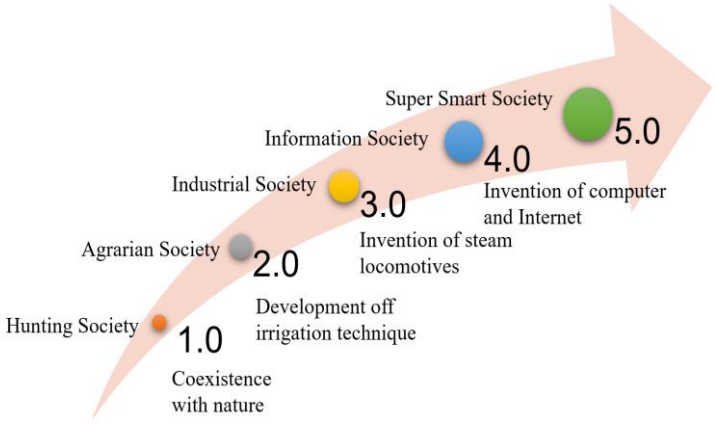
TÓM TẮT

SHA-256 là một trong những thuật toán dùng để kiểm chứng tính toàn vẹn dữ liệu được sử dụng rộng rãi nhất hiện nay. Để đạt hiệu năng cao, nhiều hệ thống triển khai SHA-256 trực tiếp trên phần cứng. Tuy nhiên, tốc độ xử lý thường bị giới hạn do khối lượng tính toán lớn. Bên cạnh đó, thuật toán đòi hỏi nhiều vòng lặp trên cùng một dữ liệu, dẫn đến việc phải truyền dữ liệu liên tục giữa bộ tăng tốc và bộ nhớ ngoài khi không có bộ nhớ cục bộ. Để giải quyết vấn đề này, bài báo đề xuất một kiến trúc ALU kết hợp tính toán song song toàn phần với các tầng pipeline, nhờ đó nâng cao tốc độ xử lý SHA-256. Ngoài ra, một khối bộ nhớ cục bộ được bố trí gần ALU để giảm thiểu truy cập bộ nhớ ngoài trong quá trình lặp tính toán. Nhằm đạt tốc độ băm cao hơn, thiết kế còn được mở rộng thành bộ tăng tốc SHA-256 đa lõi dựa trên SoC. Kết quả thực nghiệm cho thấy bộ tăng tốc của chúng tôi đạt mức cải thiện cao nhất, với tốc độ xử lý tăng gấp 31,2 lần và hiệu suất phần cứng tăng gấp 12,42 lần so với các công trình trước đây.

Từ khóa: SHA-256, tính toán song song, bộ nhớ cục bộ.

1. MỞ ĐẦU

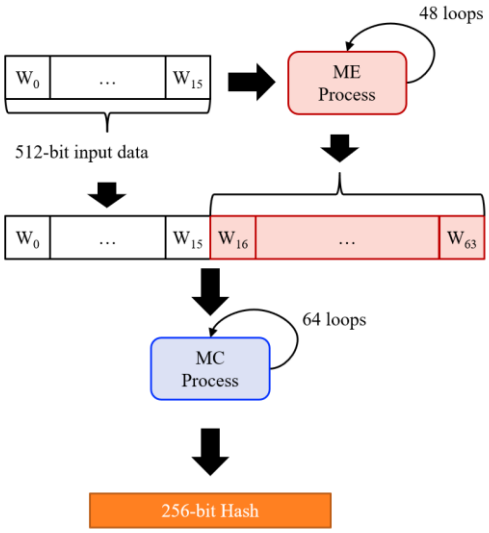
Khái niệm Xã hội 5.0, được Nhật Bản đề xuất trong Kế hoạch Cơ bản về Khoa học và Công nghệ lần thứ 5 [1], hướng đến xây dựng một xã hội siêu thông minh nơi các công nghệ như AI, IoT và Blockchain được tích hợp sâu vào mọi lĩnh vực, mang lại cuộc sống tiện nghi và phát triển kinh tế bền vững. Mô hình này đề cao việc kết nối không gian vật lý và không gian mạng, trong đó dữ liệu từ cảm biến và thiết bị thông minh được thu thập, xử lý bằng AI và Machine Learning, rồi phản hồi trở lại để tối ưu các hoạt động thực tế. Tuy nhiên, việc truyền tải lượng dữ liệu khổng lồ liên tục đặt ra thách thức lớn về an toàn và bảo mật thông tin.



Hình 1. Xã hội 5.0 được tập trung phát triển [16].

Trong bối cảnh đó, hàm băm SHA được sử dụng rộng rãi trong các ứng dụng như DSA, HMAC và Blockchain nhằm đảm bảo tính bảo mật và quyền riêng tư [2]. SHA hoạt động bằng cách ánh xạ dữ liệu thành giá trị băm không thể đảo ngược, trong đó SHA-256 là thuật toán phổ biến nhất hiện nay. Do đó, nghiên cứu và phát triển bộ tăng tốc SHA-256 trở thành một hướng quan trọng nhằm đáp ứng nhu cầu bảo mật và hiệu năng cao trong kỷ nguyên Xã hội 5.0.

2. CƠ SỞ LÝ THUYẾT



Hình 2. Tổng quan thuật toán SHA-256.

Thuật toán băm mật mã SHA (Secure Hash Algorithm) là một họ hàm băm được phát triển và công bố bởi Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (National Institute of Standards and Technology - NIST). Phiên bản đầu tiên, SHA-0, được giới

thiệu vào năm 1993, sau đó được cải tiến thành SHA-1 vào năm 1995. Họ thuật toán SHA-2, bao gồm các biến thể SHA-224, SHA-256, SHA-384 và SHA-512, được NIST công bố vào năm 2001 [15]. SHA-256 được ứng dụng rộng rãi trong các lĩnh vực như Blockchain, mã xác thực thông điệp và giao thức mã hóa. Thuật toán tạo ra giá trị băm 256-bit từ dữ liệu đầu vào bất kỳ, gồm hai giai đoạn: (i) tiền xử lý chia dữ liệu thành các khối 512-bit và (ii) tính toán băm với 64 vòng lặp, gồm Bộ mở rộng thông điệp (ME) và Bộ nén thông điệp (MC) như minh họa trong Hình 2.

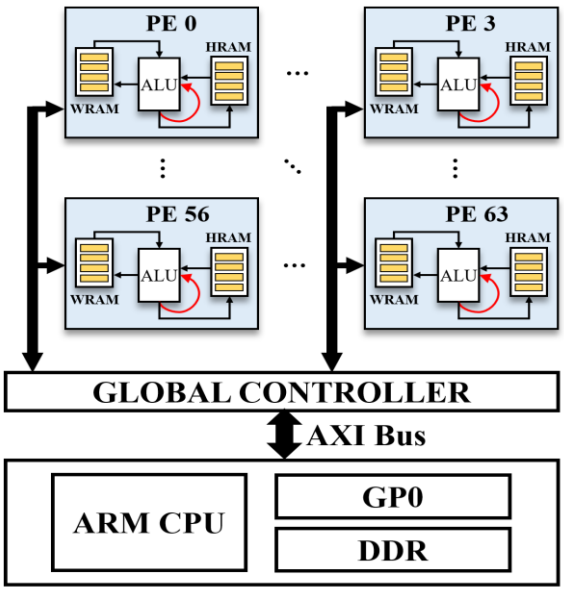
Một số nghiên cứu trước đây đã tùy chỉnh đường truyền dữ liệu (datapath) cho dữ liệu phản hồi của ALU [3], hoặc tái sắp xếp thứ tự các phép toán trong giai đoạn tính toán băm [4]. Các phương pháp này giúp giảm truy cập bộ nhớ không cần thiết và rút ngắn đường tới hạn, từ đó nâng cao hiệu năng. Bên cạnh đó, một số thiết kế đã sử dụng kỹ thuật pipeline như các khối tính toán độc lập cho MC, hoặc áp dụng phương pháp unrolling trong các vòng lặp ngoài của MC [5][6], giúp cải thiện đáng kể thông lượng. Tuy nhiên, các hướng tiếp cận trên vẫn xử lý tuần tự ME và MC ở các đơn vị riêng biệt, đồng thời tập trung vào hiệu năng lõi xử lý mà chưa hỗ trợ bộ nhớ cục bộ.

Trong nghiên cứu này, ALU được thiết kế để thực hiện tính toán song song hoàn toàn cho cả MC và ME, nhằm loại bỏ độ trễ do quá trình tính toán ME. Ngoài ra, năm tầng tính toán pipeline được tích hợp trong ALU để tăng thông lượng. Mỗi phần tử xử lý (Processing Element – PE) còn được trang bị bộ nhớ cục bộ, thanh ghi dịch và bộ điều khiển, nhằm hỗ trợ lưu trữ và xử lý khối lượng dữ liệu lớn một cách hiệu quả. Kiến trúc đề xuất này đã giảm đáng kể thời gian truy cập bộ nhớ của ALU trong quá trình xử lý dữ liệu cường độ cao.

3. MÔ HÌNH ĐỀ XUẤT

3.1. Tổng quan hệ thống

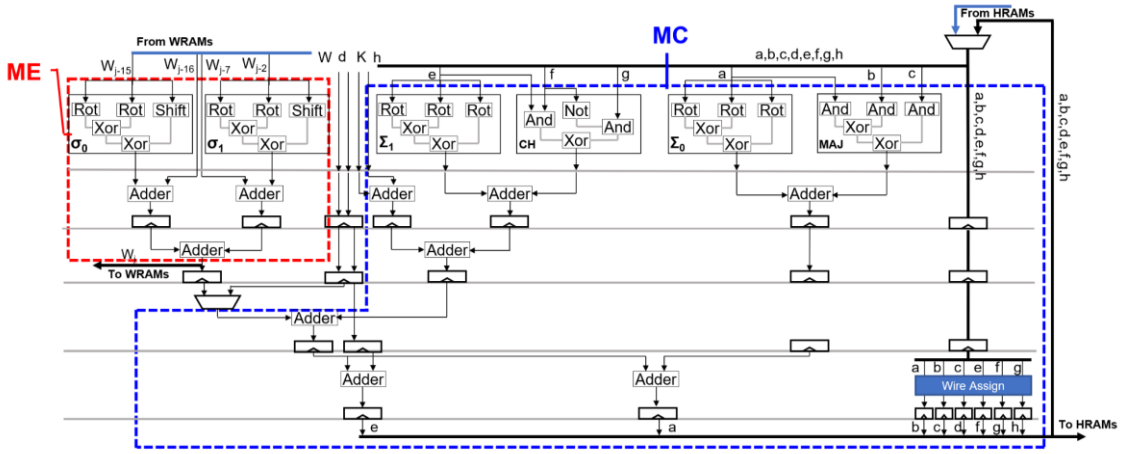
Kiến trúc hệ thống (Hình 3) gồm bộ tăng tốc SHA-256 được đề xuất và CPU chính, kết nối qua giao diện AXI trong môi trường nhúng. CPU, DDR và các cổng I/O được đóng gói trong khối xử lý PS, còn bộ tăng tốc gồm mảng các phần tử xử lý (PE) và bộ điều khiển toàn cục kết nối trực tiếp với từng PE. Thiết kế tương thích SoC giúp bộ tăng tốc dễ dàng tích hợp vào các hệ thống nhúng có hỗ trợ AXI, đồng thời giảm độ trễ nhờ sử dụng nhiều bộ nhớ cục bộ (multi-local memories) cho phép xử lý liên tục.



Hình 3. Mô hình hệ thống đề xuất

CPU điều khiển toàn hệ thống, thực hiện khởi tạo, truyền dữ liệu giữa bộ nhớ ngoài và bộ tăng tốc, cũng như nhận kết quả. Do truyền dữ liệu qua bus chia sẻ tốn thời gian, bộ nhớ cục bộ trong bộ tăng tốc được dùng để giảm chờ đợi. Thành phần chính là các PE thực hiện tính toán băm SHA-256, mỗi PE gồm ALU pipeline, bộ điều khiển, bộ nhớ cục bộ (WRAM, HRAM), thanh ghi dịch và giao diện AXI. Bộ điều khiển toàn cục quản lý truyền dữ liệu và ánh xạ đến từng PE.

ALU pipeline chứa toàn bộ toán tử logic và số học cho 64 vòng lặp SHA-256, được sắp xếp pipeline để tăng tốc độ. Hai bộ nhớ WRAM và HRAM lưu dữ liệu đầu vào/ra gần ALU giúp giảm độ trễ truy xuất, cho phép PE xử lý liên tục mà không cần truy cập bộ nhớ ngoài. Mỗi PE có thể xử lý bốn thông điệp trong 64 chu kỳ, đạt độ trễ trung bình 67 chu kỳ. Khi cấu hình tối đa 64 PE, bộ tăng tốc có thể xử lý đồng thời 64×4×L hàm băm mà vẫn duy trì pipeline và song song trên toàn hệ thống, giúp tăng đáng kể hiệu năng tổng thể



Hình 4. Kiến trúc ALU được đề xuất

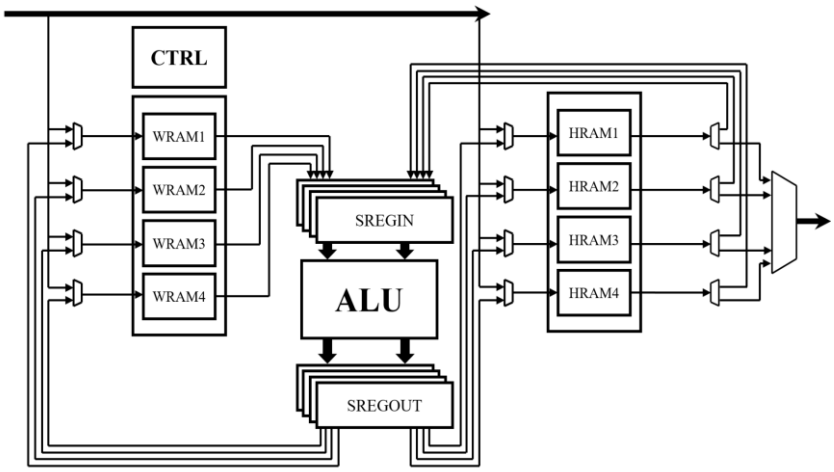
3.2. Kiến trúc ALU dạng pipeline

Kiến trúc ALU (Hình 4) gồm hai khối chính: Mở rộng thông điệp (ME) và Nén thông điệp (MC). Trong các thiết kế trước, SHA-256 tính tuần tự gây độ trễ cao; thiết kế này áp dụng rescheduling để rút ngắn đường tới hạn và giảm độ trễ. Hai giá trị A và E phụ thuộc vào T1 và T2, nên ALU tập trung tối ưu hai hàm này. Các phép logic (Σ_0, Σ_1, Maj, Ch, σ_0, σ_1) được thực hiện song song, còn các phép cộng được phân bố hợp lý giữa các chu kỳ, giúp mỗi vòng lặp chỉ cần 4 chu kỳ, với pipeline 4 luồng song song để tăng tốc độ xử lý.

Trong ME, chu kỳ đầu thực hiện xoay, XOR và dịch; chu kỳ sau dùng hai bộ cộng để tạo Wj, hoàn tất sau hai chu kỳ, rồi gửi đến bộ nhớ cục bộ và MC. MC xử lý song song: thực hiện Σ_0, Σ_1, Maj, Ch ở chu kỳ đầu, sau đó tính T1, T2 và cập nhật A–H, hoàn tất một vòng trong chu kỳ thứ tư.

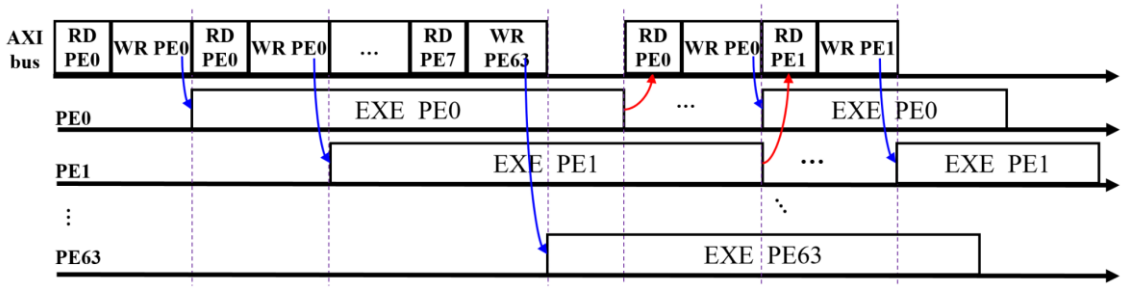
Cả ME và MC chạy 64 vòng trong một ALU. ME chia đầu vào 512-bit thành 16 khối đầu và tính 48 khối tiếp theo từ Wj−16, Wj−15, Wj−7, Wj−2 trong WRAM, kết quả được truyền và lưu song song giữa WRAM và HRAM. Nhờ xử lý song song 4 luồng pipeline, ALU tận dụng tối đa tài nguyên phần cứng, tăng hiệu năng và thông lượng, trong khi kết quả cuối được lưu vào HRAM sau 64 vòng.

3.3. Cấu trúc đa bộ nhớ cục bộ



Hình 5. Cấu trúc toàn bộ phần tử xử lý được thiết kế với đa bộ nhớ cục bộ

Trên các nền tảng thực tế, hệ thống gồm nhiều mô-đun kết nối và đồng bộ, trong đó bộ tăng tốc SHA-256 chỉ là một phần. Do băng thông giữa các mô-đun và CPU hạn chế, việc truyền dữ liệu với bộ nhớ ngoài chậm hơn truyền nội bộ, gây gián đoạn xử lý. Để khắc phục, bộ tăng tốc dùng bộ nhớ cục bộ lưu dữ liệu tạm, đảm bảo xử lý liên tục. Với kiến trúc ALU pipeline 4 luồng song song, bộ tăng tốc có 4 khối bộ nhớ ME và 4 khối MC, đều là dual-port RAM hỗ trợ đọc/ghi đồng thời, đáp ứng yêu cầu truyền dữ liệu nhanh. Hình 5 minh họa kiến trúc Processing Element (PE) gồm hai bộ nhớ chính: HRAM và WRAM, mỗi loại có 4 khối phục vụ 4 luồng song song. HRAM lưu giá trị băm khởi tạo và kết quả (16×L phần tử, 32×16×L bit), còn WRAM lưu giá trị W_0–W_{63} trong ME (64×L phần tử, 32×64×L bit).



Hình 6. Phân bố thời gian khi chạy đa lõi

Trong quá trình hoạt động, giá trị băm khởi tạo được ghi vào HRAM, kết quả ME lưu dần vào WRAM qua 64 vòng, rồi kết quả cuối được ghi lại vào HRAM. MUX đầu vào/ra cho phép chọn linh hoạt giữa ALU, bộ nhớ cục bộ và bộ nhớ ngoài, trong khi Controller điều phối luồng dữ liệu và truy cập bộ nhớ qua bus AXI. Nhờ chồng lấn

(overlapping) giữa đọc, xử lý và ghi, PE duy trì hoạt động liên tục, giảm độ trễ, tiết kiệm năng lượng và tăng hiệu năng bộ tăng tốc SHA-256.

3.4. Bộ tăng tốc đa lõi hiệu năng cao

Các ứng dụng hiện nay có yêu cầu đa dạng — từ tiết kiệm năng lượng đến hiệu năng cao — nên bộ tăng tốc cần khả năng cấu hình số lượng PE hoạt động. Bộ tăng tốc được CPU điều khiển qua thanh ghi điều khiển, gồm hai chế độ: cấu hình và truyền dữ liệu. Ở chế độ cấu hình, CPU có thể bật/tắt PE hoặc nhận tín hiệu hoàn tất xử lý. Ở chế độ truyền dữ liệu, thanh ghi điều khiển xác định vị trí bộ nhớ, chọn khối RAM (4 khối), loại WRAM/HRAM và PE cần thao tác. Nhờ đó, CPU có thể đọc, ghi hoặc điều khiển linh hoạt từng PE.

Mỗi PE hoàn thành một vòng SHA-256 trong 64 chu kỳ; với 64 PE, trung bình chỉ cần 1 chu kỳ cho mỗi giá trị băm. Bộ nhớ WRAM và HRAM riêng biệt giúp 64 PE xử lý liên tục mà không cần truy cập bộ nhớ ngoài, đạt hiệu suất tối đa 64×4×L thông điệp song song (với L là số thông điệp đầu vào). CPU điều phối dữ liệu theo chu trình WR → EXE → RD; trong khi một PE giao tiếp với bus AXI, các PE khác vẫn hoạt động bình thường như minh họa ở Hình 6. Nhờ pipeline và xử lý song song, hệ thống đạt tốc độ cao và tận dụng tối đa tài nguyên phần cứng.

4. KẾT QUẢ THÍ NGHIỆM VÀ ĐÁNH GIÁ

4.1. Thiết lập thí nghiệm và xác thực trên phần cứng thực tế

Để kiểm chứng tính đúng đắn, nhóm nghiên cứu triển khai nền tảng SoC gồm Host PC và bo mạch Xilinx UltraScale+ ZCU102 kết nối qua UART. Host PC sử dụng Vivado và SDK để cấu hình hệ thống và lập trình phần mềm nhúng. Trên ZCU102, PS (chứa ARMv8 và DDRAM) giao tiếp với PL (chứa bộ tăng tốc SHA-256) qua bus AXI, đồng thời ILA được dùng để quan sát dạng sóng hoạt động.

Ứng dụng trong SDK (ngôn ngữ C) thực hiện cấu hình số lượng PE, ghi dữ liệu đầu vào, và kích hoạt bộ tăng tốc bằng lệnh start. Kết quả được kiểm tra qua ILA hoặc phản hồi trên terminal SDK.

Thử nghiệm với cấu hình một PE và nhiều PE song song cho thấy toàn bộ kết quả đều trùng khớp, chứng minh bộ tăng tốc SHA-256 hoạt động chính xác và ổn định trên nền tảng SoC.

Bảng 1. Kết quả đánh giá

Device	**Design**	**Research**	**Freq. (MHz)**	**Slice/ LUT**	**FFs**	**Cycle**	**Throughput (Mbps)**	**Hardware Efficiency (Kbps/LUT)**
Virtex 2	ALU	[3]	35	431	-	280	65	150
		[7]	104	1117	-	65	820	734
		[8]	133	1373	-	68	109	735
		[9]	134	502	-	129	532	1509
		Our	**249**	**1064**	**1897**	**64**	**1989**	**1864**
	PE w/o RAM	[10]	136	779	-	490	75	96
		Our	**3330**	**5568**	**64**	**1584**	**475**	
Virtex E	Full PE	[11]	3588	1261	-	520	87	69
		Our	**85**	**3158**	**5230**	**64**	**677**	**215**
Virtex 4	ALU	[3]	50	422	-	280	91	217
		[4]	256	979	-	66	1984	2027
		[12]	171	610	-	65	1345	2205
		[9]	222	485	-	129	881	1817
		Our	**355**	**1061**	**1895**	**64**	**2842**	**2678**
Virtex 5	ALU	[3]	64	139	-	280	118	847
		[9]	272	273	-	129	1080	3956
		Our	**411**	**771**	**1895**	**64**	**3288**	**4264**
	PE w/o RAM	[13]	179	2796	-	65	1410	505
		Our	**374**	**2881**	**5563**	**64**	**2992**	**4262**
	64 PE	**Our**	**290**	**451545**	**339399**	**1**	**148480**	**328**
Virtex 6	ALU	[14]	271	960	-	68	2040	2125
		Our	**525**	**773**	**1895**	**64**	**4196**	**5429**
	64 PE	**Our**	**348**	**451763**	**399335**	**1**	**178176**	**394**
Virtex 7	Full PE	[6]	181	6367	25190	101	917	144
		Our	**403**	**3102**	**5385**	**64**	**3224**	**1039**
	64 PE	**Our**	**406**	**451132**	**339083**	**1**	**207872**	**460**

4.2. Kết quả thực nghiệm tổng hợp (Synthesis) trên FPGA

Mạch được thiết kế và kiểm chứng chức năng bằng testbench trên Modelsim với ba trường hợp: khối ALU, một PE hoàn chỉnh và nhiều PE xử lý song song. Để so sánh với các kiến trúc SHA-256 hiện có, bộ tăng tốc được tổng hợp trên nhiều FPGA, gồm Virtex-2, -E, -4, -5, -6, và -7. Các tiêu chí so sánh gồm tần số, lượng tài nguyên logic, tốc độ xử lý và hiệu suất phần cứng.

Do sự khác biệt giữa các công trình (chỉ ALU, toàn bộ kiến trúc, hoặc không có bộ nhớ), nhóm nghiên cứu đã tổng hợp mạch theo nhiều cấu hình (ALU riêng, toàn bộ kiến trúc, 1 PE và 64 PE) để đảm bảo tính công bằng. Kết quả cho thấy hiệu năng vượt trội ở mọi cấu hình. Băng thông xử lý (Throughput) được tính theo công thức (1) dựa trên kết quả tổng hợp từ công cụ Vivado:

$$\textbf{Throughput} = (\textbf{blocksize} \times \textbf{Fmax}) / \#\textbf{clockcycle} \quad (1)$$

Trong đó:

- blocksize: Kích thước một block dữ liệu đầu vào (512 bits đối với SHA-256)
- Fmax: Tần số hoạt động tối đa của thiết kế sau khi tổng hợp
- #clockcycle: Số chu kỳ cần thiết để xử lý hoàn chỉnh một block

Hiệu suất phần cứng (Hardware Efficiency) được định nghĩa như công thức (2):

$$\textbf{Hardware Efficiency} = \textbf{Throughput} / \textbf{LUTs} \quad (2)$$

Trong đó LUTs (Look-Up Tables) là số lượng tài nguyên logic được sử dụng trên FPGA, phản ánh mức độ tối ưu của thiết kế.

Cụ thể, trên Virtex-2, ALU đạt 1864 Kbps ở 249 MHz, cao hơn 29.2× về thông lượng và 12.42× về hiệu suất phần cứng so với các công trình [3], [7], [8], [9], nhờ kiến trúc pipeline và kỹ thuật tái sắp xếp toán tử giúp giảm độ trễ đường truyền. Trên Virtex-E, PE hoàn chỉnh đạt thông lượng và hiệu suất cao hơn lần lượt 7.78× và 3.1× so với [11]. Với Virtex-4, thông lượng và hiệu suất cao hơn tối đa 31.2× và 12.34×. Trên Virtex-5, hiệu suất phần cứng của ALU và PE cao hơn tối đa 8.4×. Trên Virtex-6 và Virtex-7, hiệu suất phần cứng tăng 2.55× và 9.09× so với [14] và [6].

Kết quả tổng hợp khẳng định bộ tăng tốc SHA-256 được đề xuất đạt hiệu năng và hiệu suất phần cứng vượt trội trên nhiều nền tảng FPGA.

5. KẾT LUẬN

Trong bài báo này, chúng tôi đề xuất một bộ tăng tốc đa lõi cho thuật toán SHA-256 nhằm nâng cao hiệu năng tính toán băm. Bằng cách triển khai cơ chế tính toán song

song giữa ME và MC, độ trễ của ALU trong mỗi vòng lặp được giảm đáng kể. Bên cạnh đó, bộ nhớ cục bộ và bộ điều khiển trong PE chịu trách nhiệm điều phối luồng dữ liệu, qua đó loại bỏ việc truyền dữ liệu thường xuyên giữa bộ tăng tốc và bộ nhớ ngoài. Kiến trúc đề xuất, bao gồm tám PE kết hợp với CPU, đã được hiện thực hóa trên nền tảng Xilinx UltraScale+ ZCU102. Kết quả thực nghiệm cho thấy thiết kế này đạt hiệu năng vượt trội so với các giải pháp hiện tại.

TÀI LIỆU THAM KHẢO

[1]. Shiroishi, Yoshihiro, Uchiyama, Kunio, Suzuki, Norihiro (2018). Society 5.0: For human security and well-being.

[2]. W. Stallings (2017). Cryptography and Network Security: Principles and Practice, 7th edition.

[3]. R. García, I. Algredo-Badillo, M. Morales-Sandoval, C. Feregrino-Uribe, and R. Cumplido (2014). A compact FPGA-based processor for the secure hash algorithm SHA-256, Computers & E.E. 40(1), pp. 194-202.

[4]. Yimeng Chen, Shuguo Li (2020). A high-throughput hardware implementation of SHA-256 algorithm, IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-4.

[5]. Raffaele Martino, Alessandro Cilardo (2019). A flexible framework for exploring, evaluating, and comparing SHA-2 designs, IEEE Access 7, pp. 72443-72456.

[6]. M. Kammoun, M. Elleuchi, M. Abid, and M. S. BenSaleh (2020). FPGA-based implementation of the SHA-256 hash algorithm, IEEE Inter. Conf. on Design & Test of Integrated Micro & Nano-Systems, pp. 1-6.

[7]. I. Algredo-Badillo, C. Feregrino-Uribe, R. Cumplido, and M. Morales Sandoval (2011). Novel hardware architecture for implementing the inner loop of the SHA-2 algorithms. 14th Euromicro Conference on Digital System Design, pp. 543–549

[8]. R. P. McEvoy, F. M. Crowe, C. C. Murphy, and W. P. Marnane. Optimisation of the SHA-2 family of hash functions on FPGAs (2006). IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06), pp. 317–322.

[9]. M. M. Wong, V. Pudi, and A. Chattopadhyay. Lightweight and high-performance SHA-256 using architectural folding and 4-2 adder compressor (2018). 2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), pp. 95–100

[10]. M. Kim, J. Ryou, and S. Jun. Efficient hardware architecture of SHA-256 algorithm for trusted mobile computing (2009). International Conference on Information Security and Cryptology, pp. 240–252.

[11].K. K. Ting, S. C. L. Yuen, K. H. Lee, and P. H. W. Leong. An FPGA-based SHA-256 processor (2002). International Conference on Field Programmable Logic and Applications, pp. 577–585.

[12].Meelu Padhi and Ravindra Chaudhari. An optimized pipelined architecture of SHA-256 hash function (2017). 7th International Symposium on Embedded Computing and System Design (ISED), pages 14.

[13].C. Jeong and Y. Kim. Implementation of efficient SHA-256 hash algorithm for secure vehicle communication using FPGA (2014). International SoC Design Conference (ISOCC), pp. 224–225, Nov.

[14].M. D. Rote, N. Vijendran, and D. Selvakumar. High–performance SHA-2 core using the round pipelined technique (2015). IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), pp. 1–6, Jul.

[15]. National Technical Information Service, U.S. Department of Commerce/NIST, Springfield, VA, USA (2002). FIPS 180-2—Secure Hash Standard. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.180-4.pdf.

[16].Yoshihiko Nagasato, Takashi Yoshimura, and Ryo Shinozaki. Realizing Society 5.0: Expectations from Japanese Business. [Online] Available:https://www.jstage.jst.go.jp/article/jsim/38/1/38_3/_pdf

HIGH PERFORMANCE MULTICORE SHA-256 ACCELERATOR USING FULLY PARALLEL COMPUTATION AND LOCAL MEMORY

Phan Van Dai*, Khong Thi Thu Thao

University of Sciences, Hue University

*Email: vandai2995@gmail.com

ABSTRACT

Ensuring data integrity is crucial in modern computing systems. Among the commonly used algorithms for integrity verification, SHA-256 is one of the most prevalent. To achieve high performance, many systems implement SHA-256 in hardware. However, its processing throughput is often constrained by the intensive computations involved. In addition, the algorithm involves repeated iterations over the same data, which can lead to frequent transfers between the accelerator and off-chip memory when local storage is not utilized. To address these issues, this paper introduces an ALU architecture that integrates fully parallel computation with pipelined layers, thereby improving the processing speed of SHA-256. Furthermore, a local memory unit is placed close to the ALU to reduce off-chip memory access during iterative operations. For even higher hash rates, we extend the design to a multicore SoC-based SHA-256 accelerator. Experimental results on an FPGA-based SoC (Xilinx UltraScale+ ZCU102) demonstrate up to 31.2× higher throughput and 12.42× higher hardware efficiency compared with representative prior designs.

Keywords: SHA-256, parallel computation, local memory.