УДК 621.374

Д. В. Гаврілов

АПАРАТНА РЕАЛІЗАЦІЯ SHA АЛГОРИТМІВ НА РІЗНИХ FPGA ТА ПОРІВНЯННЯ ШВИДКОДІЇ

Вінницький національний технічний університет

Анотація

У цій роботі представлено порівняння алгоритму поточного стандарту SHA-2 та кандидату SHA-3 Blake з точки зору апаратної ефективності в сучасних Intel FPGA. Алгоритм реалізований з використанням декількох платформ, які засновані на концепціях згортання, розгортання і конвеєрної обробки. Досліджується відношення швидкодії і пропускної здатності до кінцевого об'єму у матриці. Реалізації алгоритму порівнюються на основі їх загальної продуктивності і ідентифікуються характерні особливості кожної з них, що є важливим з точки зору створення апаратної структури.

Ключові слова: SHA; FPGA; Altera; Intel; Quartus; згортання; розгортання; конвеєрна обробка; хешування.

Abstract

This paper presents a comparison of the current SHA-2 standard algorithm and SHA-3 candidate Blake in terms of hardware efficiency in modern Intel FPGA. The algorithm is implemented with the use of several platforms, which are based on the concepts of folding, unrolling and pipelining. The relation of speed and bandwidth to the finite volume in a matrix is studied. Implementations of the algorithm are compared on the basis of their overall performance and identify the specific features of each of them, which is important for implementation in hardware structure.

Keywords: SHA; FPGA; Altera; Intel; Quartus; folding; unrolling; pipelining; hash.

Вступ

Відкриті конкурси вибору вдосконаленого стандарту шифрування (Advanced Encryption Standard – AES), стали методом відбору криптографічних стандартів у всьому світі. Чотири критерії, які беруться до уваги при оцінці кандидатів в таких конкурсах: безпека, програмна продуктивність, апаратна продуктивність і гнучкість. Хоча безпека зазвичай визнається в якості найбільш важливого критерію оцінки, це міра, яку найскладніше оцінити протягом відносно короткого періоду часу, відведеного для більшості конкурсів. Наступною є оцінка продуктивності в програмному і апаратному забезпеченні.

Дана робота буде сконцентрована на порівнянні продуктивності апаратних засобів одного з SHA-3 кандидатів у конкурсі, організованого Державним інститутом стандартів і технологій (National Institute of Standards and Technology – NIST). Особливістю підходу є дослідження декількох апаратних реалізацій архітектур алгоритмів, з наступним аналізом продуктивності з точки зору компромісу між пропускною здатністю з швидкодією та використанням апаратного ресурсу кандидату в SHA-3 та поточного стандарту SHA-2. Дослідження проводиться з використанням двох апаратних платформ FPGA: Stratix III і Stratix IV від Intel (Altera) [1 – 3].

Результати дослідження

Відомо безліч атак протоколу SHA-1, одна з них це атака, що застосовувалась для знаходження колізії хешів [4]. NIST вирішив розробити новий стандарт хешування, який буде більш надійним і заслуговуючим довіру алгоритмом хешування. Щоб домогтися цього переходу, NIST провів відкритий конкурс, започаткований в першому кварталі 2007 року [5, 6].

Через два роки після оголошення, 64 кандидата представили свої алгоритми хешування, 51 з них були допущені NIST до участі в першому турі. Критерії NIST, використані для оцінки кандидатів першого раунду, були: безпека, вартість, продуктивність і алгоритми реалізації програмного забезпечення. На цьому етапі продуктивність апаратних реалізацій не розглядалася.

У другому кварталі 2009 року NIST організувала конференцію для оголошення 14 кандидатів, які пройшли у другий тур. Потім у другому кварталі 2010 року NIST провів другу конференцію по оголошенню переможців, які пройшли в третій тур. Третій тур є останнім туром для цього конкурсу з 5 кандидатами. Критерії, використані для оцінки кандидатів усіх раундів, були однаковими. Але в

третьому раунді було додано критерій пов'язаний з апаратною реалізацією, тому останні 5 кандидатів були реалізовані апаратно [5, 6].

Основні параметри SHA-2 та кандидату SHA-3 показано в табл. 1 [1].

Таблиця 1 – Основні параметри 256-бітових і 512-бітових варіантів кандидату SHA-3 Blake і стандарту SHA-2

Алгоритм	256-біт			512-біт		
	Розмір блоку, байт	Розмір стану, байт	Кількість раундів, r	Розмір блоку, байт	Розмір стану, байт	Кількість раундів, r
Blake	512	512	14	1024	1024	16
SHA-2	512	256	64	1024	512	80

На рис. 1 показана детальна структура FPGA системи, для оцінки продуктивності апаратної реалізації. Криптографічна схема на FPGA складається з інтерфейсного блоку, який курує введенням і виведенням, і основного функціонального блоку, який виконує процес хешування. Деяким кандидатам SHA-3, необхідно зберігати вхідне повідомлення в процесі хешування, для цього використовується регістр повідомлення [7].

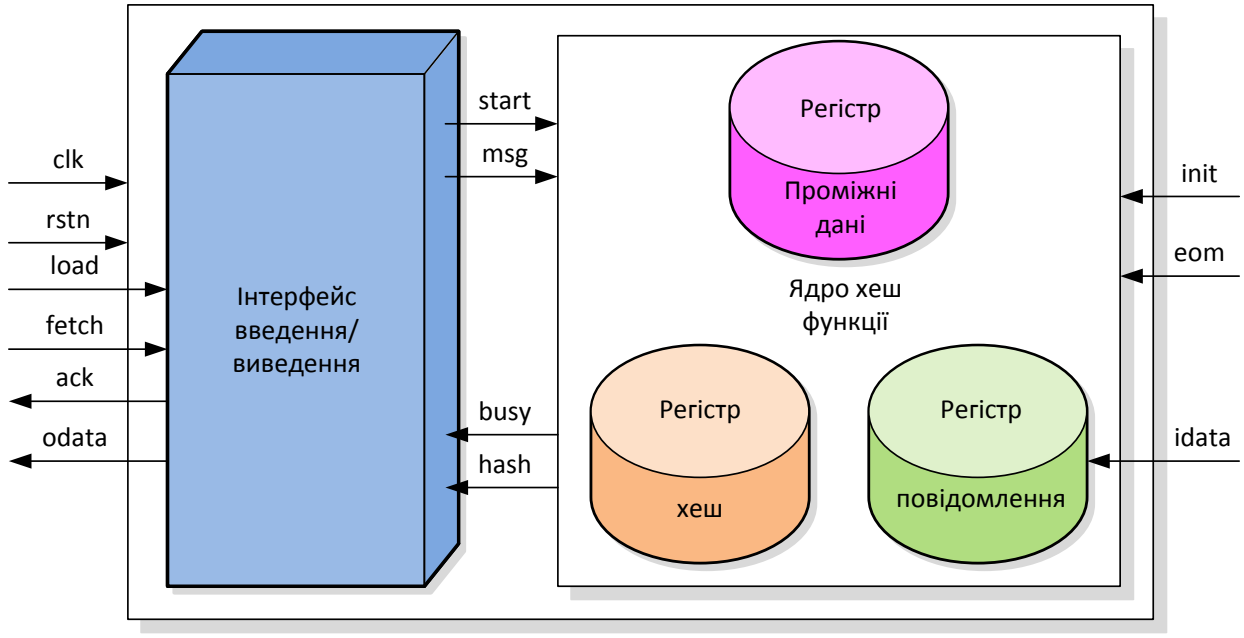

Рис. 1. Структура FPGA системи, для оцінки продуктивності апаратної реалізації

Серед фіналістів кандидатом, який може отримати значну користь від горизонтальної згортки, є Blake. Раунд Blake складається з двох горизонтальних шарів однакових функцій G, розділених тільки перестановкою. Реалізуючи тільки один шар у комбінаційній логіці, можна легко досягти горизонтальної згортки у два рази. Додатково, кожна функція G має симетричну структуру вздовж горизонтальної осі і може бути легко згорнута горизонтально з коефіцієнтом 2. В результаті досягається коефіцієнт згортання 4 для всього раунду. Принципи апаратних реалізацій хеш-функцій наведені на рис. 2 [1].

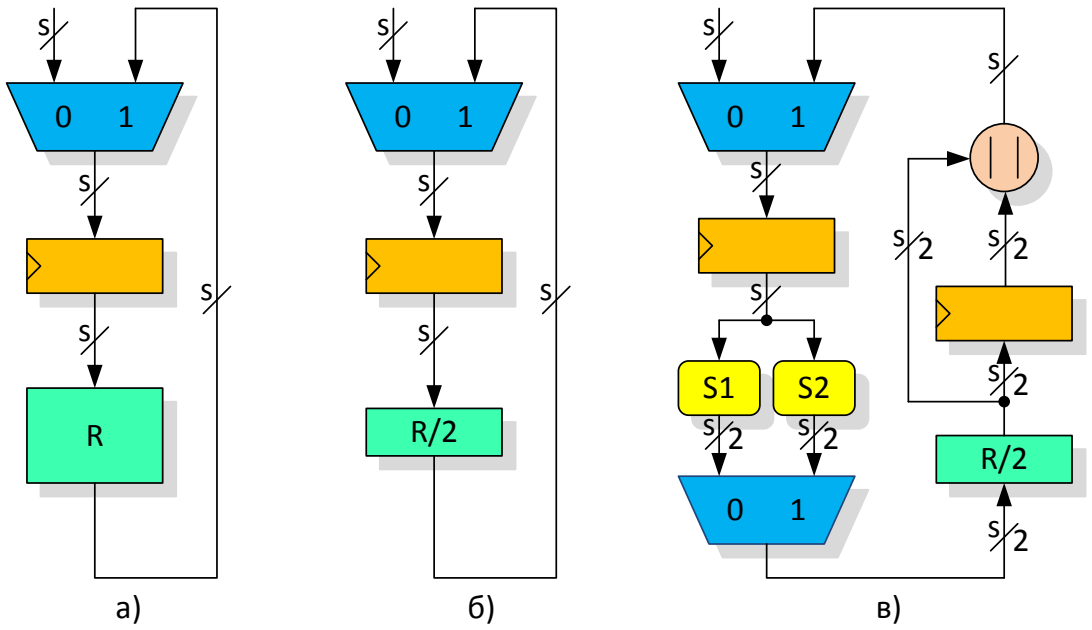

Рис. 2. Апаратні реалізації хеш-функції: а) основний ітеративний: x1, б) згорнутий горизонтально в 2 рази /2(h), в) згорнутий вертикально в 2 рази /2(v). R - раунд, S1, S2 - функції вибору

Практично у всіх відомих застосуваннях хеш-функцій, повідомлення, що обробляються є відносно короткими, зазвичай менше 1500 байт і декілька повідомлень (пакетів) є доступними для обробки блоків хешування одночасно. Наприклад, в найбільш розповсюджених протоколах безпеки Інтернету, таких як IPSec, SSL і WLAN (802.11), вхідні дані для хеш-блоків представляють собою пакети. Максимальний розмір пакета для Інтернету обмежений так званим максимальним блоком передачі (MTU). Типовий розмір MTU для Ethernet мереж становить 1500 байт. Максимальний блок для IPv4 ще менше і становить 576 байт.

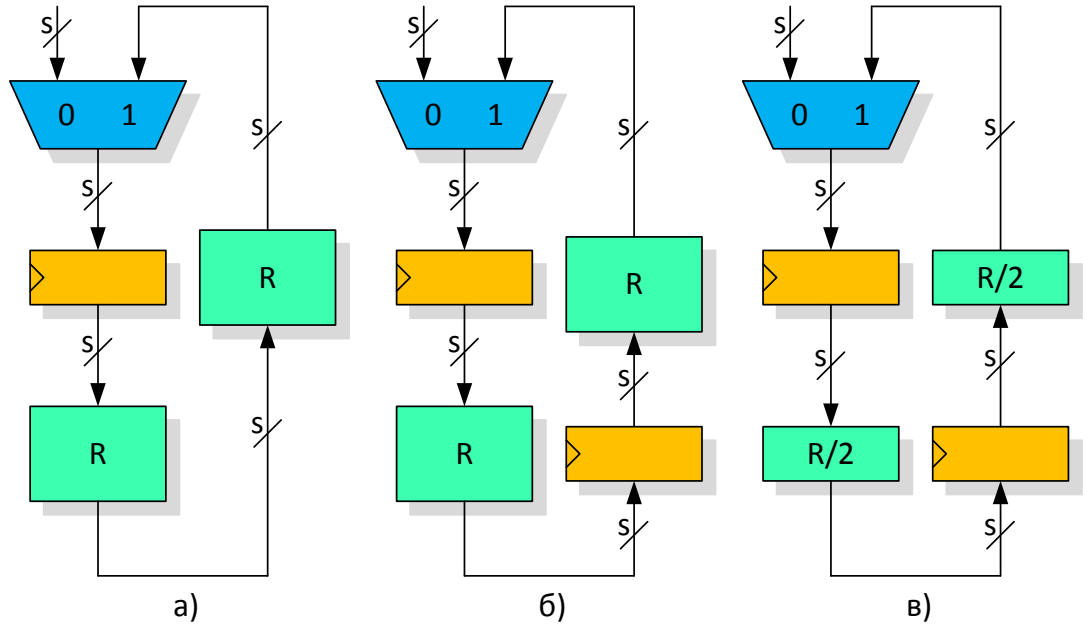

Рис. 3. Апаратні реалізації хеш-функції a) розгортаються з коефіцієнтом 2: x2, b) розгортаються з коефіцієнтом 2 з двома стадіями конвеєра: x2-PPL2, c) базова ітераційна з 2 стадіями конвеєра: x1-PPL2

В результаті, як показано на рис. 3, а, в стандартному Інтернет вузлі до 80% пакетів мають розмір 576 байт або менше, а всі пакети мають розмір 1500 байт або менше. Такі невеликі розміри блоків

даних означають, що сотні таких пакетів можуть бути легко буферизовані у вузлах обробки у вигляді черг пакетів (як показано на рис. 3, б), при цьому не буде внесено суттєвої затримки в загальний час проходження пакету від джерела до отримувача.

Порівняльні результати апаратно реалізації наведені в табл. 2 і 3 [1].

Таблиця 2 – Результати для 256-бітних варіантів кандидату SHA-3 і SHA-2, реалізованих з використанням досліджуваних архітектур і сімейств FPGA Stratix III і Stratix IV від Altera

Реалізація	Stratix III			Stratix IV		
	T	A	T/A	T	A	T/A
Blake						
/4(h)/4(v)	370	915	0.40	378	915	0.41
/4(h)	1708	3153	0.54	1747	3157	0.55
/2(h)	2151	3603	0.60	2302	3605	0.64
/2(h)-PPL2	3149	4571	0.69	3471	4570	0.76
xl	2195	4745	0.46	2305	4742	0.49
/2(h)-PPL4	4894	5080	**0.96**	5312	5049	**1.05**
xl-PPL2	4487	5420	0.83	4704	5431	0.87
xl-PPL4	7524	6273	**1.20**	8186	6278	**1.30**
SHA-256						
x1	1654	988	**1.67**	1744	988	**1.77**

Позначення: T - пропускна здатність, A - ефективність використання внутрішньої структури FPGA, T/A - відношення пропускної здатності до ефективності використання внутрішньої структури FPGA. Найкращі значення відношення пропускної здатності до ефективності використання внутрішньої структури FPGA і найкращі архітектури позначені жирним.

Для Blake дві кращі реалізації з точки зору відношення пропускної здатності до ефективності використання внутрішньої структури FPGA: x1-PPL4, тобто базова схема з 4-ма конвеєрами, а також /2(h)-PPL4, тобто архітектура з одночасною горизонтальної згорткою в 2 рази і 4-ма конвеєрами. Висока продуктивність першої пов'язана із симетричною структурою раунду, що дозволяє легко розділити канал передачі даних на дві добре збалансовані ступені конвеєра. Хороша продуктивність другої з цих реалізацій пов'язана зі значним зниженням складності функції BLAKE PERMUTE в результаті горизонтальної згортки на 2. Дві менш успішні архітектури включають x1-PPL2 і /2(h)-PPL2.

Для SHA-2 жоден з наведених методів не застосовується. Реалізація цієї функції достатня компактна, тому використання внутрішньої структури FPGA є ефективним. Кращим способом прискорити цю функцію - використовувати кілька незалежних блоків SHA-2, що працюють паралельно. Дану архітектуру позначимо через MUn, де n позначає кількість одиниць хешу.

Таблиця 3 – Результати для 512-бітних варіантів кандидату SHA-3 і SHA-2, реалізованих з використанням досліджуваних архітектур і сімейств FPGA Stratix III і Stratix IV від Altera

Реалізація	Stratix III			Stratix IV		
	T	A	T/A	T	A	T/A
Blake						
/4(h)/4(v)	485	1664	0 29	546	1675	0.33
/4(h)	2230	6137	0 36	2477	6161	0.40
/2(h)	2905	7127	0.41	3288	7128	0.46
/2(h)-PPL2	4033	8960	0.45	4780	8962	0.53
xl	2947	9251	0.32	3310	9268	0.36
/2(h)-PPL4	5535	9698	**0.57**	7521	9703	**0.78**
xl-PPL2	5549	10616	0.52	6222	10627	0.59
xl-PPL4	5647	12100	0.47	6379	12100	0.53
SHA-256						
x1	2146	2072	**1.04**	2399	2073	**1.16**

Приклад реалізації модуля тестової перевірки мовою Verilog у САПР Quartus Prime наведено на рис. 4 [8]

```
	blakeminer #(.comm_clk_frequency(comm_clk_frequency)) uut
		(
		.CLOCK_50(clk), .RxD(RxD), .TxD(TxD), .LED(led), .HEX0(h0), .HEX1(h1),
.HEX2(h2), .HEX3(h3), .HEX4(h4), .HEX5(h5)
		);

	// TEST DATA (diff=1) NB target, nonce, data, midstate (shifted from the msb/left
end) - GENESIS BLOCK
	reg [415:0] data =
	416'h000007ff555555404053081c1fcc5552c44c33134e94e7889601c1de46f746e751d100c832a56
9ef69ab9f4736b9db3e3e918f62;
	//const char golden_nonce[] = "000187a2";
	reg 		serial_send = 0;
	wire 	serial_busy;
	reg [31:0]	data_32 = 0;
	reg [31:0]	start_cycle = 0;

	serial_transmit #(.comm_clk_frequency(comm_clk_frequency), .baud_rate(baud_rate))
sertx (.clk(clk), .TxD(RxD), .send(serial_send), .busy(serial_busy), .word(data_32));

	// BLAKE rx_done is at 43500ns with loadnonce a few cycles later
	// TUNE this according to comm_clk_frequency so we send a single getwork (else it
gets overwritten with 0's)
	parameter stop_cycle = 7020;		// For comm_clk_frequency=1_000_000 [TODO
REDUCE FOR BLAKE, but 7020 is OK]
	// parameter stop_cycle = 0;		// Use this to DISABLE sending data
	always @ (posedge clk)
	begin
		serial_send <= 0;			// Default
		// Send data every time tx goes idle (NB the !serial_send is to prevent
serial_send
		// going high for two cycles since serial_busy arrives one cycle after
serial_send)
		if (cycle > 5 && cycle < stop_cycle && !serial_busy && !serial_send &&
data!=0)
		begin
			serial_send <= 1;
			data_32 <= data[415:384];
			data <= { data[383:0], 32'd0 };
			start_cycle <= cycle;		// Remember each start cycle (makes
debugging easier)
		end
	end

endmodule
`endif
```

Рис. 4. Модуль тестової перевірки мовою Verilog у САПР Quartus Prime

Висновки

У цій роботі було проведено дослідження швидкодіючих апаратних реалізацій для кандидату SHA-3 Blake і стандарту SHA-2. Досліджувані архітектури мають будову, що базується на горизонтальній згортці, вертикальній згортці, розгортках, конвеєрній обробці і паралельній обробці з використанням декількох незалежних блоків. Кожна архітектура була реалізована з використанням швидкодіючих родин FPGA: Stratix III і Stratix IV від Altera. На підставі отриманих результатів було визначено найбільш ефективну апаратну реалізацію для кожного з досліджуваних алгоритмів, виходячи з найкращого співвідношення пропускної здатності до ефективності використання ресурсу FPGA.

У випадку кандидату Blake найбільш ефективною архітектурою виявилася конвеєрна архітектура. Оптимальне число конвеєрів для алгоритму Blake склала чотири. Результати для всіх досліджених функцій і найбільш успішних апаратних реалізацій були представлені на графіках відношення пропускної здатності до ефективності використання ресурсу FPGA.

Blake є алгоритмом з самою високою гнучкістю і найбільшим числом можливих апаратних реалізацій. Алгоритм можна легко розширити по горизонталі і вертикалі в два і чотири рази. Це також єдиний алгоритм, що має ефективну архітектуру, яка меншою базової ітеративної архітектури SHA-2. Також Blake - алгоритм, який може істотно виграти від використання вбудованих блоків пам'яті Altera FPGA.

Наступне дослідження буде включати в себе експериментальне тестування всіх апаратних реалізацій з використанням обчислювальних платформ на основі FPGA Xilinx і Altera, оснащених високошвидкісним інтерфейсом PCI Express [8 – 10].

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. E. Homsirikamol, M. Rogawski, K. Gaj, and G. Mason, “Comparing hardware performance of round 3 SHA-3 candidates using multiple hardware architectures in Xilinx and Altera FPGAs,” in Ecrypt II Hash Workshop 2011, 2011. [Online]. Available: http://www.ecrypt.eu.org/ hash2011/proceedings/hash201107.pdf.
2. Y. Jararweh, L. Tawalbeh, H. Tawalbeh and A. Moh’d, "Hardware Performance Evaluation of SHA-3 Candidate Algorithms," Journal of Information Security, Vol. 3 No. 2, 2012, pp. 69-76. doi: 10.4236/jis.2012.32008.
3. Gaj K., Homsirikamol E., Rogawski M. (2010) Fair and Comprehensive Methodology for Comparing Hardware Performance of Fourteen Round Two SHA-3 Candidates Using FPGAs. In: Mangard S., Standaert FX. (eds) Cryptographic Hardware and Embedded Systems, CHES 2010. CHES 2010. Lecture Notes in Computer Science, vol 6225. Springer, Berlin, Heidelberg. DOI 10.1007/978-3-642-15031-9_18.
4. Xiaoyun Wang and Yiqun Lisa Yin and Hongbo Yu, Finding Collisions in the Full SHA-1, Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings, Lecture Notes in Computer Science, Springer, vol. 3621, pp. 17-36, doi: 10.1007/11535218_2, https://iacr.org/archive/crypto2005/36210017/36210017.pdf
5. NIST, “Cryptographic Algorithm Validation Program,” 2010. http://csrc.nist.gov
6. NIST, “Secure Hashing,” 2011. http://csrc.nist.gov
7. M. Knezevic et al., "Fair and Consistent Hardware Evaluation of Fourteen Round Two SHA-3 Candidates," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 5, pp. 827-840, May 2012. doi: 10.1109/TVLSI.2011.2128353
8. Николай Ковач, "Майнер с алгоритмом Blake," https://marsohod.org/projects/proekty-dlya-platy-marsokhod3/363-blake
9. Кофанов В. Л. Лабораторний практикум з цифрових пристроїв на основі САПР Quartus II [Текст] : навчальний посібник / В. Л. Кофанов, О. В. Осадчук, Д. В. Гаврілов. – Вінниця : УНІВЕРСУМ- Вінниця, 2007. – 167 с.
10. Кофанов В. Л. Лабораторний практикум з дослідження цифрових пристроїв на основі САПР MAX+PLUS II [Текст] : лабораторний практикум / В. Л. Кофанов, О. В. Осадчук, Д. В. Гаврілов. – Вінниця : УНІВЕРСУМ-Вінниця, 2006. – 200 с.

Гаврілов Дмитро Володимирович — канд. техн. наук, доцент, доцент кафедри радіотехніки, Вінницький національний технічний університет, Вінниця, email: havrilov@vntu.edu.ua

Havrilov Dmytro — Cand. Sc. (Eng), Associate Professor of the Department of Radio-Frequency Engineering, Vinnytsia National Technical University, Vinnytsia, email: havrilov@vntu.edu.ua