IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 14, 2020, accepted November 4, 2020, date of publication November 16, 2020,
date of current version November 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3038219

FPGA-Based Lightweight Hardware Architecture of the PHOTON Hash Function for IoT Edge Devices

MOHAMMED AL-SHATARI[1], (Graduate Student Member, IEEE),
FAWNIZU AZMADI HUSSIN[1], (Senior Member, IEEE),
AZRINA ABD AZIZ[1], (Senior Member, IEEE),
GUNAWAN WITJAKSONO[2], (Senior Member, IEEE),
AND XUAN-TU TRAN[3], (Senior Member, IEEE)

[1]Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia
[2]Department of Information Technology, BRI Institute of Technology and Business, Jakarta 12550, Indonesia
[3]SISLAB, VNU University of Engineering and Technology, Vietnam National University, Hanoi 123106, Vietnam

Corresponding author: Mohammed Al-Shatari (m.alshatari@gmail.com)

This work was supported by the Yayasan UTP Fundamental Research Grant (YUTP-FRG) under Grant 015LC0-140.

ABSTRACT The design of cryptographic engines for the Internet of Things (IoT) edge devices and other ultralightweight devices is a crucial challenge. The emergence of such resource-constrained devices raises significant challenges to current cryptographic algorithms. PHOTON is an ultra-lightweight cryptographic hash function targeting low-resource devices. The currently implemented hardware architectures of PHOTON hash function utilize a large amount of resources and have low operating frequencies with a low rate of throughputs. Maximum operating frequency and throughput of PHOTON architecture can be improved but at the cost of larger area utilization. Therefore, to improve the area-performance trade-offs of PHOTON hash function, an iterative architecture is implemented in this work. The concern is with the most lightweight version of PHOTON hash function with the hash size of 80 bits. It is implemented and verified on several Xilinx and Altera Field Programmable Gate Array (FPGA) devices using their synthesis and simulation tools. Low-cost and high-processing FPGA devices were both considered. The design is optimized for performance, whereas the area utilization is also taken into consideration. The overall performance and logic utilization are benchmarked with the existing implementations. The results show an improvement rate of 10.26% to 51.04% in the speed performance and a reduction rate of 7.55% to 60.64% in area utilization compared to existing implementations of PHOTON hash functions.

INDEX TERMS FPGA, hardware security, lightweight cryptography, low power, PHOTON Hash function, sponge construction.

I. INTRODUCTION

In our daily life, lightweight devices such as IoT edge devices and RFID cards are increasingly used in many applications either to grant access to private data or to be used in monitoring and control. These trending technologies have raised new challenges for cryptographers where private data could be leaked and modified through these low-processing devices resulting in high costs. Therefore, these devices should be integrated with cryptographic engines and secured with a high level of authentication to avoid such cases. Conventional and high-processing hash functions do not suit such constrained devices. As a result, several lightweight authentication schemes were proposed [1]–[4] and implemented in hardware and software on diverse platforms [5]–[10] with some applied cryptanalysis [11]–[13]. The internal structure of these hash function schemes is mainly focusing on the area-performance trade-offs with different size of the message digest. They can be hardware-oriented or software-oriented, where some of them are compact in hardware and efficient in software too. The processing capabilities of the current lightweight hash functions are low due to

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

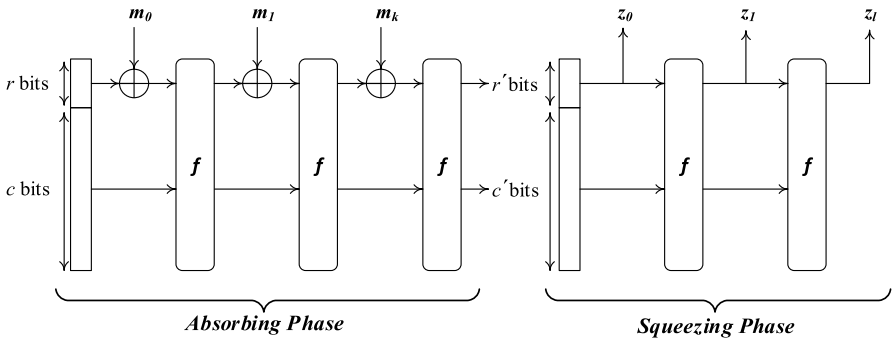

FIGURE 1. Sponge construction.

the constraints of their applications resources. However, these devices need to process real-time data. Therefore, area-performance trade-offs should be considered. PHOTON is a lightweight hash function proposed by J. Guo *et al.* [4]. It is compact in hardware and efficient in software. Its permutation is similar to LED block cipher [14] as they are both proposed by the same group, but with different sizes of datapath and state dimensions. LED has an AES-like permutation which can be designed in hardware with a tiny area [15]. In the original paper of PHOTON hash function [4], the architecture was hardware-oriented and explored in Application Specific Integrated Circuit (ASIC) with the gate equivalent as the main parameter for area utilization. Lately, the design space exploration of PHOTON hash function was proposed by [5], [16], [17] on FPGA targeting different FPGA devices. The main limitation of the existing PHOTON architectures is the large utilization of logic area compared to the achieved frequency and throughput. In this work, PHOTON architecture is designed and implemented with the focus on high performance while considering the logic utilization too.

The contribution of this article is in the implementation of an efficient hardware architecture of PHOTON hash function achieving higher speed performance with smaller logic utilization than the existing designs. The algorithm of PHOTON hash function was designed in a way where all the permutation modules of a single round are executed in one clock cycle to achieve higher throughput. The linear feedback shift register (LFSR) used to generate the round constants is also utilized as a counter and a controller of the rounds. The intensive computation of the *MixColumns* module is reduced with the use of Look-Up Tables (LUTs) instead of Galois multiplication. The usage of LFSR and LUTs significantly reduces the utilization of logic resource.

II. PHOTON ALGORITHM

The algorithm of PHOTON hash function was designed with the main goal to achieve lightweight and low-area utilization. The architecture of PHOTON algorithm is based on Sponge construction, as shown in Figure 10, and introduced by Bertoni *et al*. [18]. It consists of two phases; the absorbing phase where the message m is fully absorbed and processed through the permutation function f, and the squeezing phase where the output hash z is squeezed and generated. The structure of the internal round permutation f is AES-like with slight differences to allow for low area implementation. PHOTON algorithm is described in their original paper [4]. They defined five variants of PHOTON hash function distinguished by the size of the hash output ($80 \leq n \leq 256$) and the input and output bitrates r and r', respectively. Therefore, the function is indicated as PHOTON-$n/r/r'$. This also results in different security levels, performance and logic utilization. These different configurations of PHOTON are illustrated in Table 1.

Bertoni *et al.* [18] also extended Sponge construction to use different input and output rates for better security [19]. The internal state of PHOTON ($t = c + r$) is interpreted in a two-dimensional way, where c is the capacity and r is the rate. Therefore, the state size t depends on the capacity and rate to form the matrix dimensions ($d \times d$) with the cell size s. The parameter d determines the number of rows and columns in the two-dimensional matrix representation (d^2 cells) while the number of bits per cell is defined by the parameter s, where $s \in \{4, 8\}$, and thus $t = sd^2$. The input message m is permuted with the input rate r and concatenated with the capacity c to form the state t which is mapped to a matrix representation of dimension ($d \times d$) of the state h with s cell size as in (1).

$$h\,[i]\,[j]\,[k] = t[sdi + sj + k] \tag{1}$$

The output is mapped to the size of the output rate r' to form the message digest n by concatenating the segments of output z.

The padding rule of PHOTON hash function is defined by appending the string 10^* where the length of the message is a multiple of the rate r and defined as in (2).

$$pad\ (m) = m \,\|1\|\, 0^k \tag{2}$$

where, m is the message of an arbitrary length of {0,1} bit strings $m \in \mathbb{Z}_2^{\geq 0}$ and $k = (|m|\text{-}1 \bmod r)$.

The state is initialized by a pre-defined initialization vector (IV) based on the variant of PHOTON as in (3).

$$IV = 0^{t-24}||n/4||r||r' \tag{3}$$

where t is the size of the internal state, n is the size of the output hash, r and r' are the input and output rates respectively.

TABLE 1. Variants of PHOTON Hash Function.

PHOTON Variants	State Size t [bit]	Hash Digest n [bit]	Input Rate r [bit]	Output Rate r' [bit]	Capacity c [bit]	Cell Size s [bit]	Matrix Size d [cell]	Rounds N_r
PHOTON-80/20/16	100	80	20	16	80	4	5	12
PHOTON-128/16/16	144	128	16	16	128	4	6	12
PHOTON-160/36/36	196	160	36	36	160	4	7	12
PHOTON-224/32/32	256	224	32	32	224	4	8	12
PHOTON-256/32/32	288	256	32	32	256	8	6	12

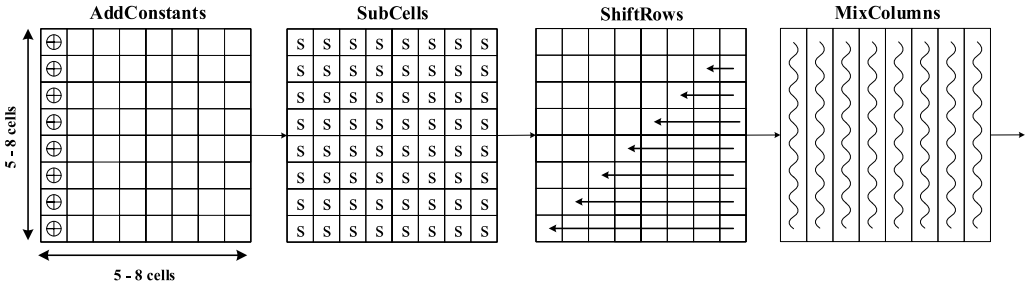

FIGURE 2. PHOTON Permutation.

The permutation and round function of PHOTON are very much like AES and composed of four modules; *AddConstants* (AC), *SubCells* (SC), *ShiftRows* (SR) and *MixColumns* (MC) as shown in Figure 2.

AddConstants (AC): There are two constants in this module: a four-bit round-dependent constant (RC) generated from a Linear Feedback Shift Register (LFSR), and a pre-defined d-dependent internal constant (IC_d). The *RC* is initialized to a specific value and updated every round by the LFSR, whereas the IC_d is initialized based on the value of the dimension d. These two constants are both XORed with the first column of the $d \times d$ internal state. In this operation, only the first column is permuted while other columns are left unchanged. Overall, for N_r round number, the updated state is given as in (4).

$$h'[i,j] = h[i,j] \oplus RC_{N_r}(i) \oplus IC_d(i) \tag{4}$$

where, $h[i,j]$ is the current state, i represents the row number, j represents the column number, and N_r is the round number.

As the AC operation is dealing with the first column only, j is fixed to *0* as shown in (5).

$$h'[i,0] = h[i,0] \oplus RC_{N_r}(i) \oplus IC_d(i) \tag{5}$$

Therefore, the overall *AddConstants* function is as in (6).

$$\text{AC} : h'[i,j] = \begin{cases} h[i,j] \oplus RC_N r(i) \oplus IC - di & for\ J = 0 \\ h[i,j] & for\ 0 < j < d \end{cases} \tag{6}$$

SubCells (SC): The second operation of PHOTON is to perform cell substitution. PHOTON uses two different S-boxes based on the size of the cell s. For the variants where $s = 4$ *bits,* PRESENT [20] S-Box $SB_{PRESENT}$ is used while

TABLE 2. Substitution Box of Present Cipher.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

AES [21] S-Box SB_{AES} is used for $s = 8$ *bits* variant. Each cell of the state is replaced by a corresponding cell from the nonlinear S-Boxes. PRESENT S-Box is shown in Table 2. The overall function of SubCells is given in (7).

$$\text{SC} : h'[i,j] = \begin{cases} SB_{PRESENT}(h[i,j]) & for\ s = 4 \\ SB_{AES}(h[i,j]) & for\ s = 8 \end{cases} \tag{7}$$

ShiftRows(SR): this function is almost identical to that of the AES transformation function. All the rows of the state matrix are rotated to the left by i cells (columns) where i is row index and starts counting from *0.* Within this module, the first row is not permuted, while other rows are updated accordingly. Therefore, the formal notation of SR function is as in (8).

$$\text{SR} : h'[i,j] = h[i, (i+j)\ mod\ d] \quad for\ 0 \le i,j < d \tag{8}$$

MixColumns(MC): The MC is a finite field multiplication based on maximum distance separable (MDS) matrix. The columns of the internal state are independently multiplied with the pre-defined matrix based on the dimension size d. The design of *MC* for PHOTON has some shared similarities with AES but focusing on minimum area consumption. Matrix multiplication for the MC operation uses Galois Field with the irreducible polynomial $x^4 + x + 1$ for GF(2^4) when $s = 4$ and the AES polynomial $x^8 + x^4 + x^3 + x + 1$ for GF(2^8)

when $s = 8$. MC for PHOTON is defined as in (9).

$$(h'[0,j],\ldots,h'[d-1,j])^T = A_t^d \times (h[0,j],\ldots,h[d-1,j])^T \tag{9}$$

where, A is the pre-defined d-dependent matrix, t is the size of the state. A, d and t are different for each PHOTON variant.

III. IMPLEMENTATION OF PHOTON ON FPGA

PHOTON architecture is designed in various flavors with different level of security with the focus on low-area utilization. It can be parallelized in a round-based mode for considerable throughput or it can also be designed in serialized nibble/byte-wise mode for low-area optimization. In this work, the architecture of the most lightweight variant of PHOTON hash function is presented. The design is based on PHOTON-80/20/16 variant with a hash size of 80-bit n, 20-bit input rate r, 16-bit output rate r', 100-bit state size t, (5×5) state dimension d^2 and 4-bit cell size s. Verilog HDL is used to design the internal permutation and round functions of this architecture and implemented on several Altera and Xilinx FPGAs. Altera Quartus II and ModelSim are used as synthesis and simulation tools for Altera FPGAs whereas Xilinx ISE and ModelSim for Xilinx FPGAs. Xilinx ISE was used instead of Xilinx Vivado as our benchmarking works were targeting low-processing conventional FPGA devices which are not supported by Xilinx Vivado. Figure 3 illustrates the block diagram of the proposed PHOTON-80/20/16 architecture. We have optimized the architecture for high throughput while considering the area utilization too.

We have considered only one 20-bit input message where we omitted the padding block for now. Therefore, only one message can be processed at a time. The initialization vector for PHOTON-80/20/16 is as in (10).

$$IV_{100} = \begin{Bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 4 & 1 & 4 & 1 & 0 \end{Bmatrix} \tag{10}$$

Therefore, the input capacity c and rate r are initialized with the *IV,* where r takes the most left bits of IV. For this variant of PHOTON, the width of $IV = t = c + r = 100$ bits. The rate r is XORed with the message, and the result is concatenated with the capacity c and loaded into the *STR* register which then becomes the input to the permutation block. PHOTON permutation processes the message in 12 rounds. In each round, these four functions are executed: *AddConstants*, *SubCells*, *ShiftRows* and *MixColumns*. After the last round, the 16-bit output h is generated.

In the *AddConstants* operation, the round constants are XORed with the internal constants and then XORed again with the first column of the state and the result is passed to the *SubCells* module. The internal constants depend on the d size. For this PHOTON variant, $IC_d = [0, 1, 3, 6, 4]$ or can be generated using LFSR with feedback function $FB(X_r) = x_2\ NOR\ x_1$ for serial implementation. The round constants

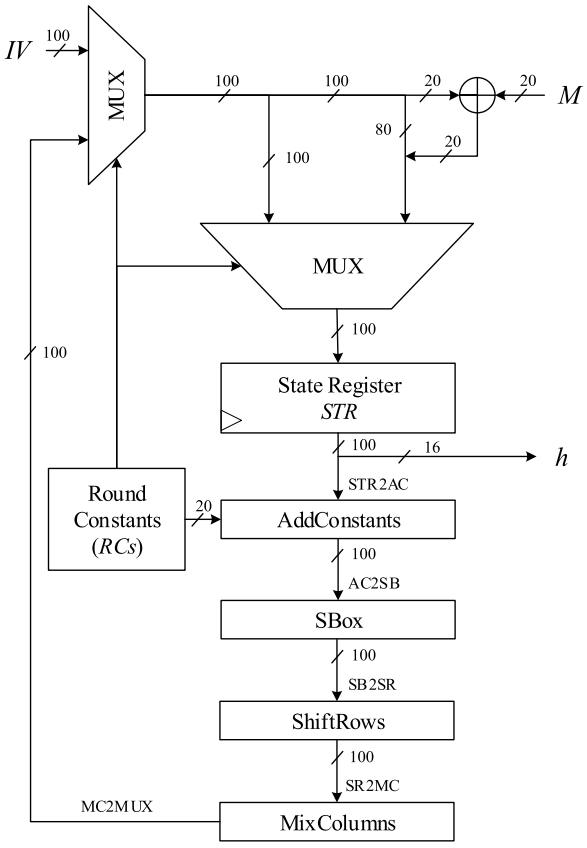

FIGURE 3. Architecture of iterative round-based PHOTON-80/20/16.

TABLE 3. Round Constants for PHOTON-80/20/16 (D = 5).

N_R / ROW	1	2	3	4	5	6	7	8	9	10	11	12
1	1	3	7	E	D	B	6	C	9	2	5	A
2	0	2	6	F	C	A	7	D	8	3	4	B
3	2	0	4	D	E	8	5	F	A	1	6	9
4	7	5	1	8	B	D	0	A	15	4	3	12
5	5	7	3	A	9	F	2	8	D	6	1	E

depend on the round number and the row position, as shown in Table 3.

The *Round Constants* module is supplying the Internal Constants and round constants to the *AddConstants* module every round. Instead of using a round controller, the RC is utilized as a counter to control the number of rounds and the output of both multiplexers to the state register, which reduces the logic resources.

The *SubCells* module depends on the size of the cell s. Since $s = 4$ *bits* for PHOTON-80/20/16, the state is updated from the PRESENT substitution box given in Table 2. Every cell in the state matrix is substituted by its corresponding value from PRESENT S-box and the result becomes the input to the *ShiftRows* module.

The *ShiftRows* module distributes every single column over all columns by rotating the rows to the left by i nibble position for $(0 \leq i < d)$.

The *MixColumns* module is used to enhance the diffusion property . This module uses the most resources among all the PHOTON permutation blocks due to the matrix multiplication. The *MixColumns* finite multiplication can be designed in parallel by applying column-wise single multiplication with matrix A given in (11). It can also be serialized by multiplying matrix A^5 five times with the matrix columns independently. In this work, the design of *MixColumns* module is based on Look-up Tables (LUT) similar to the *SubCells* to mitigate the intensive computation of the matrix multiplication.

$$A_{100} = \begin{Bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 9 & 9 & 2 \end{Bmatrix}^5 = \begin{Bmatrix} 1 & 2 & 9 & 9 & 2 \\ 2 & 5 & 3 & 8 & D \\ D & B & A & C & 1 \\ 1 & F & 2 & 3 & E \\ E & E & 8 & 5 & C \end{Bmatrix} \quad (11)$$

The flow process of the proposed architecture of PHOTON-80/20/16 is illustrated in the ASM chart in Figure 4. It is a round-based architecture where the permutation module is applied in one cycle and the state register *STR* is loaded every round. Therefore, the twelve rounds of PHOTON will be achieved in 12 clock cycles. The round constants of the first row *Row0* are used as a round counter which controls the multiplexers and the loading of the state register *STR* and the output Z. When the RC of *Row0* is initialized to *0*, the *STR* register is loaded with the initialization vector IV where its first 20 bits are XORed with the input message m, and the LFSR is updated. When *Row0* is 10, it indicates the end of the twelve rounds. Therefore, the *STR* is updated directly from the *MixColumns* block, the output z is generated, and the RC is reinitialized. Otherwise, the *STR* takes the output of the *MixColumns* and the LFSR is updated.

The architecture is implemented on various families of Altera and Xilinx FPGA devices including Arria and Cyclone from Altera and Spartan3, Virtex5, Artix7 and Kintex7 from Xilinx. Only 200 logic registers were utilized to implement PHOTON-80/20/16 - 100 bits for the state matrix, 20 bits for the round constants which also used as a counter and 80 bits for concatenating the output. FPGA devices have a different configuration of logic resources resulting in distinct performance and area utilization for each device. Old and low-processing FPGAs have smaller Look-Up Tables (LUT) while the latest and high-processing FPGA can have up to 6-input LUTs. The performance and logic-area utilization for Altera FPGA are illustrated in Table 4 and Xilinx FPGA in Table 5.

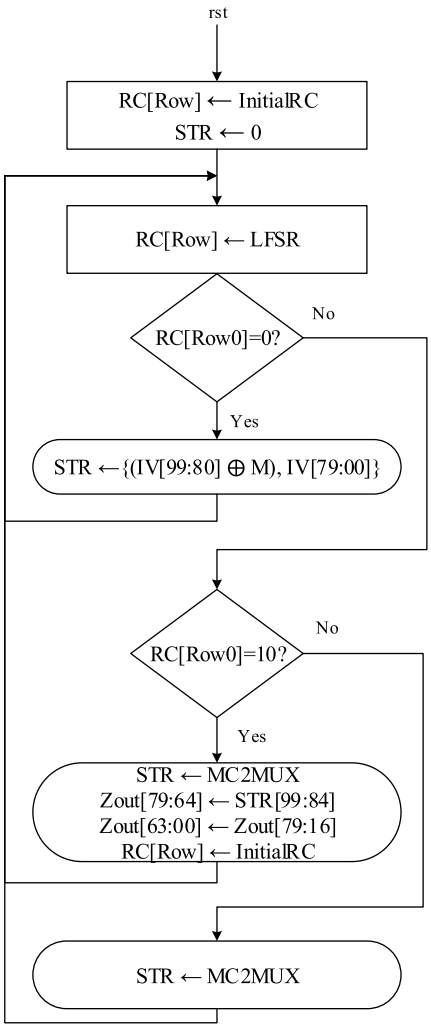

FIGURE 4. ASM chart of round-based PHOTON-80/20/16.

IV. RESULTS AND DISCUSSIONS

Cryptographic modules are the throughput bottlenecks for IoT edge devices and other ultralightweight devices [22]. These modules also contribute much to the logic resources utilization of the low-processing lightweight devices. Therefore, encryption/decryption and authentication engines should be designed with the least possible logic resources and the highest possible throughput to satisfy the constraints of the IoT edge devices. Several PHOTON hash function architectures of different design optimization goals were presented in [5], [16], [23] and implemented on Spartan3, Virtex5, Artix7 and Kintex7 FPGAs from Xilinx. The implementations of these existing designs utilize large amount of logic resources relative to the achieved performance. The trade-off of our proposed architecture outperforms all the current existing works as illustrated in Table 4 and Table 5.

The architecture of PHOTON-80/20/16 variant is implemented on RTL using Verilog HDL. The design is verified on various FPGA devices from Xilinx and Altera using their respective synthesis and simulation tools. The whole permutation operation takes only one cycle to process 100 bits of data. The algorithm of PHOTON-80/20/16 takes 12 rounds to absorb one 20-bit input message and another 48 rounds to squeeze the message and produce the 80-bit output hash.

In this single-round architecture, the hash output is generated after 60 iterations as demonstrated in the simulation waveform in Figure 5 and Figure 6. Every permutation round is processed within one clock cycle. The first 12 cycles are utilized by the absorbing phase of the sponge construction to process a single 20-bit input message *Min*, which generates

TABLE 4. Round-Based Implementation Results of PHOTON-80/20/16 Hash Function on Altera FPGAs.

Design	Datapath (bits)	No. of LEs	No. of FFs	No. of LUTs	No. of Clock Cycles	Max. Freq. (MHz)	T/put (Mbps)	Efficiency (Mbps/LE)	Power mW	FPGA Device
Our Paper Round-based	100	540	200	465	60	245.82	409.7	0.76	75.59	Cyclone II
	100	540	200	465	60	307.41	512.35	0.95	96.86	Cyclone III
	100	539	200	463	60	267.38	445.63	0.83	179.30	Cyclone III LS
	100	536	200	465	60	291.29	485.48	0.90	113.16	Cyclone IV E
	100	539	200	464	60	312.40	520.67	0.97	120.96	Cyclone IV GX
	100	238 (ALMs)	200	384	60	208.07	346.78	-	127.41	Cyclone V
	100	474 (ALMs)	200	385	60	380.66	634.43	-	364.83	Arria II GX

TABLE 5. Round-Based Implementation Results of PHOTON-80/20/16 Hash Function on Xilinx FPGAs.

Design	Datapath (bits)	No. of slices	No. of FFs	No. of LUTs	No. of Clock Cycles	Max. Freq. (MHz)	T/put (Mbps)	Efficiency (Mbps/slices)	Power mW	FPGA Device
Our Paper Round-based	**100**	**265**	**200**	**510**	**60**	**157.24**	**262.07**	**0.99**	**27**	**Spartan-3 XC3S50-5**
Round-based [5]	100	285	127	565	12	78.53	130.88	0.46		
Serialized [5]	4	146	137	256	648	100.43	3.10	0.02		
SRL16 [5]	20	112	68	20.	360	118.19	6.57	0.06		
DLP-PHOTON [16]	100	615	-	-	-	308	1027	1.67		
Our Paper Round-based	**100**	**145**	**188**	**363**	**60**	**376.43**	**627.38**	**4.33**	**82**	**Artix-7 XC7A100T-3**
Round-base [5]	100	142	111	336	12	232.65	387.75	2.73		
Serialized [5]	4	67	134	167	648	329.51	10.17	0.15		
SRL16 [5]	20	58	89	144	360	329.95	18.33	0.32		
DLP-PHOTON [16]	100	402	-	-	-	903	3010	7.48		
Our Paper Round-based	**100**	**188**	**200**	**425**	**60**	**337.27**	**562.12**	**2.99**	**452**	**Virtex-5 XC5VLX50-1**
Serialized [5]	4	82	135	188	648	302.68	9.34	0.11		
SRL16 [5]	20	69	89	759	360	285.2	15.84	0.22		
Iterative [23]	20	302	415	508	12	172.7	287.83	0.95		
Folding [23]	20	251	414	515	24	205.7	171.42	0.68		
Unrolling [23]	20	1066	411	3065	1	25.43	508.6	0.48		
Our Paper Round-based	**100**	**126**	**188**	**366**	**60**	**358.42**	**597.37**	**4.7**	**110**	**Kintex-7 XC7K70T-1**

the first 16-bit output of the hash *Zout[15:0]*. In addition to the first 12 cycles, the squeezing phase takes another 48 cycles to produce the remaining output of the 80-bit hash *Zout*. This is because the algorithm has 12 iterations for the absorbing phase and another 12 $\times$ 4 iterations for the squeezing.

The architecture was implemented on several FPGA families from Altera and Xilinx. Only 200 dedicated logic registers are used to process this variant of PHOTON, where a 100-bit register is to update the state register, a 20-bit register holding the LFSR Round Constants and the counter and an 80-bit register is to hold the concatenated hash output.

For Altera devices, there is no available work in the literature for PHOTON hash function. On these Altera families, our design utilizes around 500 logic elements (LEs) which are approximately 3% of the total FPGA logic resources. The proposed design achieves performance from 208.07 MHz to 380.66 MHz for operating frequency f_{max} and 346.78 Mbps to 634.43 Mbps for throughput. The power dissipation ranges from 75.59 mW for low-processing FPGA devices to 364.83 mW for high-processing FPGA devices. The power consumption was measured using the PowerPlay Power Analyzer Tool provided in Quartus II software. We used the value change dump file which is generated from the design,

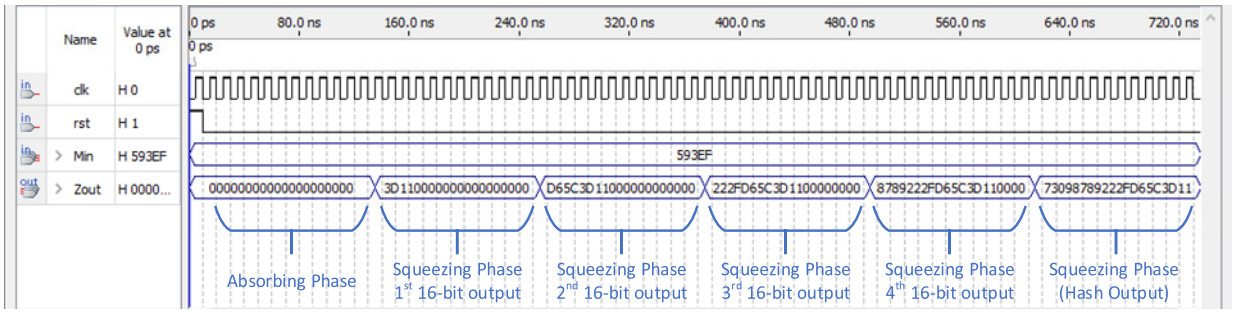

FIGURE 5. Simulation results of round-based PHOTON-80/20/16 with message (593EF) by Quartus II.

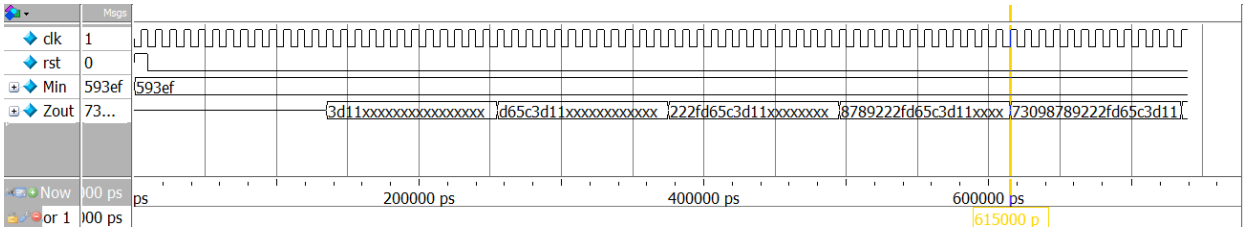

FIGURE 6. Simulation results of round-based PHOTON-80/20/16 with message (593EF) by ModelSim.

to provide sufficient toggle rate for more accurate power measurement including thermal dynamic, static and I/O power consumption. Table 4 summarizes the results of PHOTON-80/20/16 implemented on Altera FPGAs

For Xilinx devices, PHOTON-80/20/16 was implemented on Spartan-3, Virtex-5, Artix-7 and Kintex-7. Performance results and logic utilization are improved compared to the available implementations in literature as shown in Table 5 with our results highlighted in bold. In this architecture, Xilinx FPGA devices utilize 126 to 265 slices and 363 to 510 Look-Up-Tables to implement this variant of PHOTON depending on the processing capability of the device. They achieved a performance of 157.24 MHz to 376.43 MHz for operating frequency and 262.07 Mbps to 627.38 Mbps for throughput.

The efficiency is measured as the ratio of the achieved throughput to the number of the utilized slices (Mbps/slices). This architecture achieves better trade-offs between performance and utilization of the logic area than other existing works. Therefore, the proposed architecture outperforms the existing works and achieves higher efficiency except for the case of [16] as they achieved higher efficiency at the cost of a huge area because their implementation of PHOTON was in double-length Sponge construction. For Spartan-3, an efficiency of 0.99 was achieved and it is much better than all the existing implementations except for the design proposed by [16] with a slightly higher efficiency because their implementation is for high performance at the cost of 3x higher resources utilization. The implementations on Artix-7 and Kintex-7 achieve almost double the efficiency of the benchmarking architectures, with the same exception mentioned earlier for the design proposed in [16].

The work in [5] presented three different architectures of PHOTON hash function – round-based, serialized and SRL16 synthesized in Xilinx Spartan-3, Artix-7 and Virtex-5 FPGA devices. For Spartan-3 FPGA, our design utilizes a slightly smaller number of logic slices (265 slices) than their round-based architecture design (285 slices) with higher operating frequency (157.24 MHz), throughput (262.07 Mbps) and efficiency (0.99) compared to their operating frequency (78.53 MHz), throughput (130.88 Mbps) and efficiency (0.46) of the same design. For the other two architecture designs in [5] (serialized and SRL16), they used a smaller width of datapath to reduce logic footprints. Their designs utilize a smaller area but at the cost of much higher latency and lower throughput.

For Artix-7 FPGA, the round-based design proposed in [5] utilizes a slightly smaller number of logic slices (142 slices) than our proposed design (145 slices), but our design achieved higher operating frequency (376.43 MHz), throughput (627.38 Mbps) and efficiency (4.33) compared to their operating frequency (232.65 MHz), throughput (387.75 Mbps) and efficiency (2.73) of the same design. For the serialized and SRL16 designs in [5], they reduced the datapath width resulting in smaller footprints and considerable operating frequency but at the cost of much higher latency and much lower throughput.

For Virtex-5, Anandakumar *et al.* [5], [23] presented five architecture designs of PHOTON hash function. The two designs in [5] were optimized for area, whereas the three designs in [23] were exploring the performance trade-offs and side-channel attack. The serialized and SRL16 architectures in [5] achieved a much smaller area with a very low throughput compared to the proposed design. In contrast, the architectures (iterative, folding, unrolling) in [23] consume a larger

amount of logic resources with considerable throughput compared to the proposed architecture. The Unrolling architecture in [23] achieved competitive throughput but utilized almost six times the logic resources of the proposed design. The detailed results of PHOTON-80/20/16 implementation on Xilinx FPGAs and their comparisons with existing results are shown in Table 5.

IoT edge devices are adopted in many different environments, where some of them operate on batteries. These battery-powered devices have low-processing capabilities. Therefore, the security part of their design should utilize small footprint and consume low power. Logic area and data processing contribute to the consumption of power which in turn shortens the battery lifetime. In literature, there is no specific power consumption rating for IoT devices; but the lower power consumption the longer battery life. Our design has very reasonable power consumption compared to the other cryptographic primitive targeting IoT edge devices such as AES, SIMON, SPECK, PRESENT, LED and TWINE [24] with our design achieving much higher throughput. The total power consumption for Xilinx FPGA ranges from 24 mW for low-processing devices to 452 mW for high-processing FPGA devices. Xilinx XPower Analyzer was used to measure the power consumption of Xilinx FPGA devices with 12.5% toggle rate for flip-flops and I/O for better power measurements. The current industry demand is for high bandwidth with the same or lower footprints, power consumption and cost. The variation of the power consumption of different FPGA devices for the same design is due to many factors including process technology and architecture design. Smaller process technology node provides greater performance with lower power than the previous nodes which achieves certain level of the current industry demand. The power consumption of PHOTON-80/20/16 implementation on Xilinx FPGAs is reported in Table 5.

V. CONCLUSION

An iterative architecture of PHOTON-80/20/16 lightweight hash function is implemented on several Altera and Xilinx FPGA devices. It is a round-based architecture where all the permutation operations are executed in one round. The absorbing phase of the sponge construction takes 12 rounds to process a single 20-bit input message. The squeezing phase takes 48 rounds to produce the output of the 80-bit hash. The proposed design achieves better area-performance trade-offs than the existing designs as the architecture of the *MixColumns* module is designed using look-up tables to avoid the intensive computations of the multipliers. The round constants are also utilized as rounds counter to reduce the logic resources. It consumes less logic resource and achieves higher performance resulting in higher efficiency. The overall performance was improved with a rate of 10.26% to 51.04% for throughput and operating frequency and 7.55% to 60.64% for logic area utilization based on the targeted FPGA device. For future work, it is recommended to run the design on live hardware for better timing analysis, explore the design space for serialized and pipelined architectures as well as authenticating the existing lightweight block ciphers which have similar internal architecture.

ACKNOWLEDGMENT

The authors would like to thank the support from Universiti Teknologi PETRONAS for providing the Graduate Assistantship (GA) and Graduate Research Assistantship (GRA).

REFERENCES

[1] A. Bogdanov, M. Kneževi, G. Leander, D. Toz, K. Varc, and I. Verbauwhede, "SPONGENT: A lightweight hash function," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer-Verlag, 2011, pp. 312–325.

[2] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "Keccak sponge function family main document," *Submission NIST*, vol. 3, no. 30, pp. 320–327, 2009.

[3] J.-P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, "Quark: A lightweight hash," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2010, pp. 1–15.

[4] J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON family of lightweight hash functions," in *Proc. Annu. Cryptol. Conf.*, 2011, pp. 222–239.

[5] N. N. Anandakumar, T. Peyrin, and A. Poschmann, "A very compact FPGA implementation of LED and PHOTON," in *Proc. Int. Conf. Cryptol.*, 2014, pp. 304–321.

[6] B. Jungk, L. R. Lima, and M. Hiller, "A systematic study of lightweight hash functions on FPGAs," in *Proc. Int. Conf. ReConFigurable Comput. FPGAs*, 2014, pp. 1–6.

[7] F. Kahri, H. Mestiri, B. Bouallegue, and M. Machhout, "High speed FPGA implementation of cryptographic KECCAK hash function cryptoprocessor," *J. Circuits, Syst. Comput.*, vol. 25, no. 4, Apr. 2016, Art. no. 1650026.

[8] E. Aerabi, M. Bohlouli, M. H. A. Livany, M. Fazeli, A. Papadimitriou, and D. Hely, "Design space exploration for Ultra-Low-Energy and secure IoT MCUs," *ACM Trans. Embedded Comput. Syst.*, vol. 19, no. 3, pp. 1–34, Jul. 2020.

[9] R. Martino and A. Cilardo, "SHA-2 acceleration meeting the needs of emerging applications: A comparative survey," *IEEE Access*, vol. 8, pp. 28415–28436, 2020.

[10] A. Alzahrani and F. Gebali, "Multi-core dataflow design and implementation of secure hash Algorithm-3," *IEEE Access*, vol. 6, pp. 6092–6102, 2018.

[11] K. Bussi, D. Dey, P. R. Mishra, and B. K. Dass, "MGR hash functions," *Cryptologia*, vol. 43, no. 5, pp. 372–390, Sep. 2019.

[12] C.-Y. Lu, Y.-W. Lin, S.-M. Jen, and J.-F. Yang, "Cryptanalysis on PHOTON hash function using cube attack," in *Proc. Int. Conf. Inf. Secur. Intell. Control*, Aug. 2012, pp. 278–281.

[13] C. Dobraunig and B. Mennink, "Key Recovery Attack on PHOTON-Beetle," NIST, Gaithersburg, MD, USA, Tech. Rep.

[14] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2011, pp. 326–341.

[15] M. Al-Shatari, F. A. Hussin, A. A. Aziz, G. Witjaksono, M. S. Rohmad, and X. T. Tran, "An efficient implementation of LED block cipher on FPGA," *Proc. 1st Int. Conf. Intell. Comput. Eng. (ICOICE)*, 2019, pp. 1–5.

[16] B. T. Hammad, Y. A. Abbas, N. Jamil, M. E. Rusli, and M. R. Zaba, "FPGA Implementation of DLP-PHOTON Hash Function," *Int. J. Future Gener. Commun. Netw.*, vol. 10, no. 12, pp. 71–78, 2017.

[17] B. Jungk, "FPGA-based evaluation of cryptographic algorithms," Goethe University Frankfurt am Main, Frankfurt, Germany, Tech. Rep., 2016.

[18] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "On the indifferentiability of the sponge construction," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2008, pp. 181–197.

[19] N. Kishore and P. Raina, "Parallel cryptographic hashing: Developments in the last 25 years," *Cryptologia*, vol. 43, no. 6, pp. 504–535, Nov. 2019.

[20] A. Bogdanov, "PRESENT: An ultra-lightweight block cipher," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.*, 2007, pp. 450–466.

[21] J. Daemen and V. Rijmen, *The design of Rijndael: AES-The Advanced Encryption Standard*. Berlin, Germany: Springer-Verlag, 2013.

[22] A. Singh, N. Chawla, J. H. Ko, M. Kar, and S. Mukhopadhyay, "Energy efficient and side-channel secure cryptographic hardware for IoT-edge nodes," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 421–434, Feb. 2019.
[23] N. N. Anandakumar, "SCA resistance analysis on FPGA implementations of sponge based MAC-PHOTON," in *Proc. Int. Conf. for Inf. Technol. Commun.*, 2015, pp. 69–86.
[24] W. Diehl, F. Farahmand, P. Yalla, J.-P. Kaps, and K. Gaj, "Comparison of hardware and software implementations of selected lightweight block ciphers," in *Proc. 27th Int. Conf. Field Program. Log. Appl. (FPL)*, Sep. 2017, pp. 1–4.

MOHAMMED AL-SHATARI (Graduate Student Member, IEEE) received the bachelor's degree in computer engineering and the master's degree in electronic and telecommunication from Universiti Teknologi Malaysia (UTM), Skudai, Johor Bahru, Malaysia, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS (UTP), Seri Iskandar, Perak, Malaysia. He is also a Graduate Assistant with UTP. His current research interests include hardware security and cryptography.

FAWNIZU AZMADI HUSSIN (Senior Member, IEEE) received the bachelor's degree in electrical engineering from the University of Minnesota, Twin Cities, Minneapolis, MN, USA, in 1999, under PETRONAS Scholarship, the M.Eng.Sc. degree in systems and control from the University of New South Wales, Sydney, NSW, Australia, in 2001, under UTP Scholarship, and the Ph.D. degree in core-based testing of system-on-a-chip (SoCs) from the Nara Institute of Science and Technology, Ikoma, Japan, in 2008, under the scholarship from the Japanese Government (Monbukagakusho). He was the Program Manager of Master by coursework program from 2009 to 2013, the Deputy Head of the Electrical and Electronic Engineering Department from 2013 to 2014, and the Director of Strategic Alliance Office from 2014 to 2018 with UTP. He spent one year as a Visiting Professor with Intel Microelectronics (Malaysia)'s SOC DFx Department, from 2012 to 2013. He is currently an Associate Professor of electrical and electronic engineering with Universiti Teknologi PETRONAS. He is also a Visiting Professor with the Malaysia-Japan International Institute of Technology (MJIIT-UTM). He has been actively involved with the IEEE Malaysia Section as a Volunteer, since 2009. He was the 2013 and 2014 Chair of the IEEE Circuits and Systems Society Malaysia Chapter. He is currently serving as the Chair for the IEEE Malaysia Section, in 2019 and 2020.

AZRINA ABD AZIZ (Senior Member, IEEE) received the bachelor's degree (Hons.) in electrical and electronic engineering from The University of Queensland, Australia, in 1997, the M.Sc. degree in system level integration from the Institute for System Level Integration (ISLI), Scotland, in 2003, and the Ph.D. degree in computer systems engineering from Monash University, Melbourne, Australia, in 2013. She was a Process Engineer with Unisem, Sdn Bhd. She is currently a Senior Lecturer with the Department of Electrical and Electronic Engineering, Universiti Teknologi PETRONAS (UTP), Malaysia. Her current research interests include energy-efficient techniques for topology control in wireless sensor networks (WSNs), wireless body area networks (WBANs) for biomedical applications and medical imaging, and machine learning techniques. She is a member of the Board of Engineers Malaysia. She is also the Journal Reviewer of *Ad Hoc Networks* under the ScienceDirect publisher. She is a Treasurer of the IEEE RAS Malaysia and an Executive Committee of the IEEE WIE Malaysia from 2020 to 2021.

GUNAWAN WITJAKSONO (Senior Member, IEEE) received the B.S. *(magna cum laude)* and M.S. degrees in electrical engineering from Michigan Technological University, Houghton, MI, USA, in 1992 and 1994, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Wisconsin–Madison, in 2002. From 1994 to 1996, he was with the National Aeronautics and Space Agency, Indonesia. In 2002, he joined Denselight Semiconductors Pte Ltd., Singapore, where he developed high-speed, long wavelength, and distributed feedback lasers. He was with Finisar Malaysia to develop uncooled and high-speed optical transceiver. From 2005 to 2007, he was with the Department of Electrical Engineering, University of Indonesia, before joining MIMOS when he held various key positions such as a Principal Researcher and the Director of Research and Sensor System Architect, until 2016. He is currently a Professor with the BRI Institute of Technology and Business, Jakarta, Indonesia, where he is also an Indonesia-Chapter Professional Engineer.

XUAN-TU TRAN (Senior Member, IEEE) received the Ph.D. degree in micro nano electronics from Grenoble INP (at the CEA-LETI), France, in 2008. He was an Invited Professor with the University Paris-Sud 11, France, in 2009, 2010, and 2015, also with The University of Electro-Communication, Tokyo, in 2019, also with Grenoble INP, in 2011 and 2020, and also an Adjunct Professor with the University of Technology Sydney, from 2017 to 2020. He is currently an Associate Professor with the VNU University of Engineering and Technology, Vietnam National University (VNU), Hanoi. He is also the Director of the VNU Key Laboratory for Smart Integrated Systems (SISLAB). His research interests include design and test of systems-on-chips, networks-on-chips, design-for-testability, asynchronous/synchronous VLSI design, low-power techniques, and hardware architectures for multimedia applications. He has published more than 80 journal articles and conference papers in these areas and given invited talks and courses at several universities. He is a Senior Member of the IEEE Circuits and Systems (CAS), the IEEE Solid-State Circuits and Systems (SSCS), a member of IEICE, and the Executive Board of the Radio Electronics Association of Vietnam (REV). He serves as the Chairman for the IEICE Vietnam Section and the IEEE SSCS Vietnam Chapter.