Ultra Low-Power and Low-Energy 32-bit Datapath AES Architecture for IoT Applications

Duy-Hieu Bui, Diego Puschini, Simone Bacles-Min, Edith Beigné
Université Grenoble Alpes, CEA, LETI, MINATEC
Campus, 38054, Grenoble, FRANCE
firstname.lastname@cea.fr

Xuan-Tu Tran
Laboratory for Smart Integrated Systems (SISLAB)
VNU University of Engineering and Technology
Hanoi, Vietnam
tutx@vnu.edu.vn

***Abstract*—In this paper, we propose a novel AES microarchitecture with 32-bit datapath optimized for low-power and low-energy consumption targeting IoT applications. The proposed design uses simple shift registers for key/data storage and permutation to minimize the area, and the power/energy consumption. These shift registers also minimize the control logics in the key expansion and the encryption path. The proposed architecture is further optimized for area and/or power/energy consumption by selecting a suitable implementation of S-boxes and applying the clock gating technique. The implementation results in TSMC 65nm technology show that our design can save 20% of area or 20% of energy per bit at the same area when compared with the current 32-bit datapath designs. Our design also occupies smaller core area with lower energy per bit and at least 4 times higher in throughput in comparison with other 8-bit designs in the same technology node.**

Keywords—Advanced Encryption Standards; AES; Low-Power; Low-Energy; IoT

I. INTRODUCTION

The development of the Internet-of-Things (IoT) raises the concerns about the security. It used to be an additional feature for integrated systems but it is, nowadays, crucial in many applications. Security functions implemented in software reduce the overall throughput but increase power/energy consumption [1]. One way to optimize the throughput and power consumption is to implement the security algorithms in hardware with the trade-offs among security, area, throughput, power consumption and energy consumption [2]. In general, those security functions such as data encryption, authentication and identification are normally based on cryptographic algorithms and we will focus, in this paper, on the widely-used algorithm called Advanced Encryption Standard [3].

The Advanced Encryption Standard (AES) is one of the main algorithms used in the current Internet-of-Things proposals such as IEEE 802.15.4 [4], LoraWan [5], Sigfox [6]. AES is used not only for IoT but also for other security applications such as security storage, data transmission, data verification, etc. There exist many hardware designs of AES for different applications but overall performances still need to be seriously improved [7].

For high-throughput applications, AES can be designed with 128-bit datapath, with unrolled architectures or pipeline implementations. These designs require large area and high power consumption which are not suitable for constrained devices as IoT applications. They often use AES in authenticated encryption with authenticated data (AEAD) mode [8], which means that the throughput of AES has to double the data rate because it takes two encryptions for a block of data [8] in the current IoT proposals. To reduce the area, designers may choose a serial architecture with 8-bit datapath and one or two substitution boxes (S-boxes) because they occupy a large area in AES hardware. However, this leads to a reduction in throughput and an increase of the encryption time because of the serialization. Some power reduction techniques such as back biasing and supply voltage reduction are also widely used [9].

In this paper, we present a novel AES 32-bit datapath encryption architecture targeting a wide range of IoT applications from low-power low-speed network to medium- and high-speed network with 44 cycles per 128-bit encryption. We propose a structure for key and data storage and permutations by using simple shift registers to minimize area and power consumption, and the power optimization for substitution boxes (S-boxes) in the key expansion. Clock gating techniques are applied to further reduce internal switching activities, area and power consumption. Our proposed architecture provides medium throughput (about 28Mbps at 10MHz) with the core area equivalent to 8-bit AES designs for RFID and IoT applications in TSMC 65nm technology and 15-30% gain in energy consumption.

The rest of the paper is organized as follows. In Section II, we present our microarchitecture and power reduction technique with a small core area. Section III shows our method to evaluate the power consumption and the energy consumption as well as the obtained results in power/energy simulations.

II. PROPOSED AES MICROARCHITECTURE

AES is a round-based block cipher with the block size of 128 bits supporting the key size of 128 bits, 192 bits, and 256 bits with 10 rounds, 12 rounds and 14 rounds respectively. It has been standardized in 2001 under the name FIP-197 by US National Institute of Standard and Technology (NIST) and then included in ISO/IEC 18033-3. In this paper, we exclusively present the encryption architecture for AES with 128-bit key; however, this architecture can be extended to other key sizes and to the decryption architecture. In our framework, we choose to design AES encryption only with 128-bit key

because it is sufficient for long-term security and there is a mechanism to use AES with encryption only for constrained devices for data encryption and decryption [5].

There are four basic operations in a round of AES encryption datapath including *AddRoundKey*, *SubBytes*, *ShiftRows*, and *MixColumns*. The key expansion is composed of three operations: *RotateWords*, *SubWords* and the *XOR*s. *AddRoundKey* is the *XOR*s of data and the key. *SubBytes* and *SubWords* are similar because they both implement S-box operations. The following subsections describe in detail our architecture in Fig. 1.

A. Encryption path

The encryption path includes four parts: a 128-bit state register, 4 S-boxes, a *MixColumns*, and a 3×32-bit output register which also acts as a 96-bit temporary register to store the intermediate results. Our design is a 32-bit datapath architecture which means the input data and the input key are divided into 32-bit chunks. Each pair of 32-bit data and 32-bit key is loaded together. This takes 4 cycles to load the 128-bit key and 128-bit data and *XOR* them into the state register.

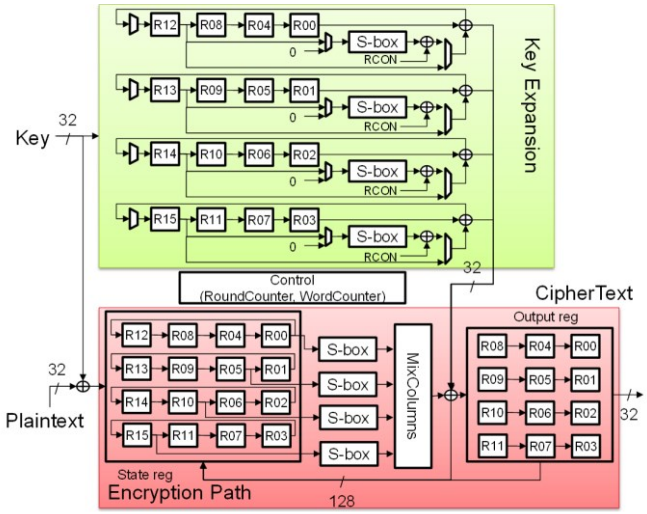

Fig. 1. AES 32-bit datapath architecture.

The state register is organized so that, after loading the input data and the input key, the encryption is done by shifting the data 32 bits in each clock cycle. The state register consists of 16 8-bit registers (so called "state matrix") which are further divided into 4 4-stage shift registers. AES standard specifies that *ShiftRows* are permutation operations on the rows of the state matrix while *MixColums* are operations on the columns. However, in our design, based on *ShiftRows* specification, we completely eliminate *ShiftRows* by selecting the diagonal of the state matrix (from lower-left corner to upper-right corner) as its outputs after each shift operation; then we only use operations by columns to save area and power consumption. Therefore, the 32-bit output of the state register is a column of 4 bytes after *ShiftRows* according to AES standard. This reduces the control logic for the state register and also completely removes the logic for *ShiftRows* steps. Thanks to this structure, the state register's content will be swapped with the content of the output register concatenated with 4 last bytes of the round operation every 4 cycles (or after each round finishes). Consequently, we save a 32-bit register because we need to store only 3×32-bit temporary data from the encryption path in the output register, while the last 32-bit data are written back directly into the state register.

After the state register, there are four S-boxes followed by the *MixColums* to enable processing four bytes in each clock cycle. The temporary results are stored in the output registers. When the encryption finished, the results are read out from the output register. In the 128-bit key configuration, AES encryption module needs 10 rounds, which leads to 40 cycles to finish the encryption for a 128-bit block of data. The total number of cycles to encryption a block in our architecture is 44 cycles.

The dynamic power is saved by applying clock gating techniques on the shift registers. When in the inactive state, the shift registers are not changed, which means that there is no activity in the encryption path. The power simulation shows that even in the highest throughput mode (44 cycles/ encryption) the applied clock gating technique can save more than 13% of power. Certainly, with smaller throughput the clock gating technique can even save much more power consumption.

B. Key expansion

The key expansion deploys the same mechanism as in the encryption path. The expanded key is calculated on-the-fly to save the area. Key expansion module consists of a 4×32-bit shift register, 4 S-boxes, and 32 2-input *XOR*s. The *RotateWords* step is not necessary because it exchanges the positions of bytes in a 32-bit register.

Based on the key expansion specification, 4 S-boxes are used one cycle during 4 cycles of a round. The inputs to these S-boxes are gated to save the dynamic power. These S-boxes are enabled for the first cycle of a round. After that, they remain inactive. This leads to 30-60% reduction in power consumption of the S-boxes in the key expansion depending on the type of S-boxes used.

The 32-bit output of the key expansion is sent directly to the encryption path to be *XOR*ed in the *AddRoundKey* step. The clock gating technique is also applied in the key expansion to save power consumption. During the idle state, the key register and the S-boxes will not create any activities.

C. AES Substitution box (S-box)

The substitution box (S-box) has a big impact on area and power consumption of the AES design. In our architecture, the S-boxes occupy from 40% to 55% of the total cell area, while they consume about 10%- 20% of the total power consumption. The smallest implementation of S-boxes until now is from Canright [10]. Canright S-box demonstrates optimized area (292 gates/S-box) but needs more power/energy consumption because it creates more activities especially in an architecture with 8 S-boxes. The most popular and straight-forward S-box implementation is the LUT-based S-box. LUT-based S-box is bigger in terms of area (434 gates/S-box) but smaller in power/energy consumption than Canright S-box. The most efficient S-box in terms of power consumption is Decode-Switch-Encode (DSE) S-box [11]; however, it occupies a larger area (466 gates/S-box).

In our design, to achieve the trade-offs between area and power/energy consumption, we chose to use a mixed design style for S-boxes. The encryption path uses S-boxes more often; therefore, DSE S-boxes can be deployed to save the power consumption. In the key expansion, S-boxes are used during the first cycle of each round. In this case, Canright S-boxes are used to save the area, and the power is saved by gating the inputs of these S-boxes. By doing this, we can implement the whole AES module with 32-bit datapath in the same core area with the designs claimed for RFID and IoT with 8-bit data path. For example, we save 16% in area when using DSE S-boxes in the encryption datapath and Canright S-boxes in the key expansion, while the power consumption increases 13% compared with the case where only DSE S-boxes are used.

III. Power Simulation

A. Methodology

Our design is synthesized using Synposys Design Compiler J-2014.09, implemented using Cadence Innovus 15.2. PrimeTime J-2014.12 is used for power estimation. All the power estimation results are post placement-and-route simulations with full timing and parasitic parameters. We use the best case corner for the simulations to evaluate the worst case in terms of power consumption. The extraction condition is set to 1.32V, at 0ºC.

Our verification model for both behavior simulation and power simulation is shown in Fig. 2. The input data and input key to the AES core are randomly generated in SystemVerilog and passed to the core through the 32-bit interfaces. The reference model used to verify the results is from Libgcrypt [16], an open source library distributed with Linux Operating system. The verification results show that our design is conformed to the AES standard.

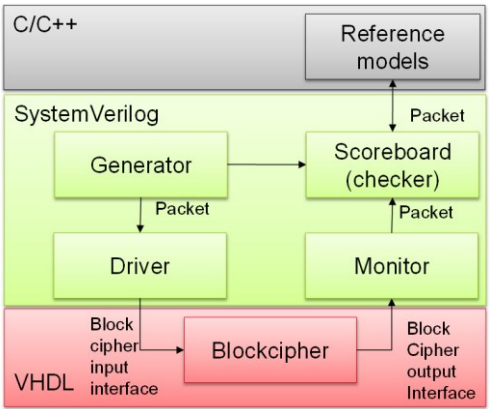

Fig. 2. Verification model.

B. Results and comparison

Fig. 3 shows the occupied area and the estimated energy of our AES cores with different styles of S-boxes (LUT S-boxes, Canright S-Boxes, DSE S-boxes and some mixtures of them in the encryption path and the key expansion) after placement and route. In comparison with the closest architecture from Banik *et al* [7], our design can save up-to 30% energy consumption with DSE S-boxes, 20% energy consumption with the same gate counts in case of the deployment of the mixed style of S-boxes, or 20% smaller in area with Canright S-boxes with nearly the same energy per bit. However, Banik's results are from the post-synthesis estimation. Therefore, it is less accurate than our results. In comparison with other 8-bit designs in Table I, depending on the selection of S-boxes, our design can have the best occupied core area with Canright S-boxes and the second best energy per bit after the design from [9] with DSE S-boxes in 65nm technology. However, the design in [9] uses also the back-biasing technique and the results are measured at very low supply voltage.

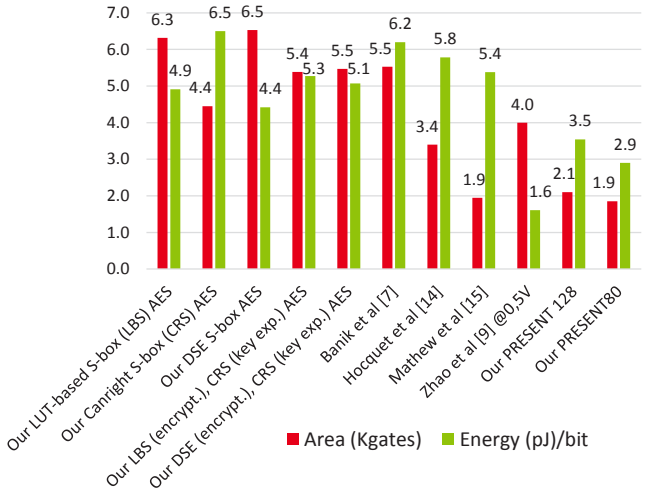

Fig. 3. Area and Energy/bit comparison.

We also compared our AES cores with a lighter cryptography block called PRESENT [17] that we implemented in the same technology and simulated after back-end. Our DSE S-box AES core presents 20% and 30% more energy per bit than our PRESENT cores with 128-bit key and 80-bit key respectively. However, our PRESENT cores have nearly 40% fewer throughputs than the ones in our AES cores, because our PRESENT cores take at least 72 cycles to complete the encryption of 128-bit data while our AES cores process the same number of bits in 44 cycles.

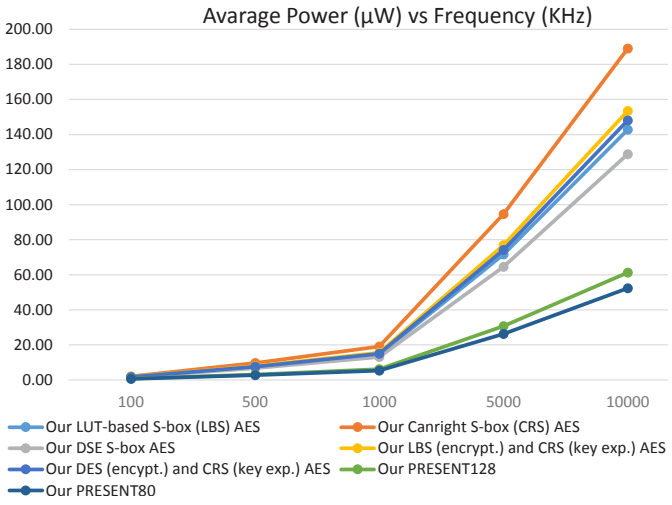

Fig. 4. Power consumption at each frequency.

The relation between the average power consumption and frequency is shown in Fig. 4. Our AES cores with different styles of S-boxes consume over two times more power than our PRESENT cores with 80-bit key and 128-bit key. It is clear from Fig. 4 that the power consumption is proportional to the operating frequency. Our AES cores consume less than 20µW at 1MHz and approximately 150µW at 10MHz with the throughput of 2.8Mbps and 28Mbps respectively. The power consumption is quite similar in the proposals of the mixed S-boxes.

IV. Conclusion and Future Works

In this paper, we proposed an optimized microarchitecture of AES with 32-bit datapath. Our proposals are modeled in VHDL, fully implemented as an IP core for IoT applications. Area and power consumption are saved by employing a well-organized structure of simple shift registers for data and key storage and permutations. The power consumption is further optimized by using clock gating technique and the gating of the inputs to the S-boxes in the key expansion. Our implementation results show that it can be used for IoT applications with different requirements in terms of area, throughput, and power/energy consumption. The implementation results with the clock gating demonstrate less core area and energy consumption than the ones in other designs for RFID and IoT applications. The power consumption of our core is proportional to the operating frequency and the throughput which may be adapted to applications. In the future, we would like to apply more advanced low-power techniques such as back-biasing with subthreshold voltage to fully evaluate the capabilities of the proposed architecture.

Acknowledgment

This work is partly supported by Vietnam National University, Hanoi (VNU) through research project No. QG.16.73 (ADEN4IOT).

References

[1] F. Zhang, R. Dojen, and T. Coffey, "Comparative performance and energy consumption analysis of different AES implementations on a wireless sensor network node," International Journal of Sensor Network vol. 10, no. 4, pp.192-201, October 2011.

[2] L. Batina, A. Das, B. Ege, E. B. Kavun, N. Mentens, C. Paar, I. Verbauwhede, T. Yalçın, " "Dietary recommendations for lightweight block ciphers: power, energy and area analysis of recently developed architectures," in Proceedings of RFIDsec'13, LNCS, vol. 8262, pp. 103–112, Springer, 2013.

[3] National Institute of Standards and Technology. "Advanced encryption standard," http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[4] IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs), 2011.

[5] Lora Alliance, "LoraWan Specification", 2015.

[6] Sigfox Technology, http://sigfox.com

[7] S. Banik, A. Bogdanov, and F. Regazzoni, "Exploring Energy Efficiency of Lightweight Block Ciphers," Cryptology ePrint Archive, 2015.

[8] P. Rogaway, "Authenticated-encryption with associated-data", In Proceedings of the ACM 9th Conference on Computer and Communications Security (CCS02):, pp. 98-107, November 2002.

[9] W. Zhao, Y. Ha, and M. Alioto, "Novel Self-Body-Biasing and Statistical Design for Near-Threshold Circuits With Ultra Energy-Efficient AES as Case Study," IEEE Transaction on Very Large Scale Integration (VLSI) Systems, pp. 1390 – 1401, vol. 23, no. 8, Aug. 2015.

[10] D. Canright, "A Very Compact S-Box for AES," in Proceedings of CHES 2005, pp. 441-455, August 2005.

[11] G. Bertoni, M. Macchetti, L. Negri, and P. Fragneto, "Power-efficient ASIC synthesis of cryptographic S-boxes," in Proceedings of GLSVLSI'04, pp. 277-281, NY, USA, 2004.

[12] T. Good, and M. Benaissa, "692-nW Advanced Encryption Standard (AES) on a 0.13- m CMOS," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 18, pp. 1753-1757, December 2010.

[13] A. Satoh, S. Morioka, K. Takano, S. Munetoh, and C. Boyd, "A Compact Rijndael Hardware Architecture with S-Box Optimization," Advances in Cryptology - ASIACRYPT 2001, pp. 239-254, Dec. 2001.

[14] C. Hocquet, D. Kamel, F. Regazzoni, J. Legat, et al., "Harvesting the potential of nano-CMOS for lightweight cryptography: an ultra-low-voltage 65 nm AES coprocessor for passive RFID tags," Journal of Cryptographic Engineering, vol. 1, pp. 79-86, February 2011.

[15] S. Mathew, S. Satpathy, V. Suresh, M. Anders, H. Kaul, A. Agarwal, S. Hsu, G. chen, and R. Krishnamurthy, "340mV-1.1V, 289 Gbps/W, 2090-Gate NanoAES Hardware Accelerator with Area-Optimized Encrypt/Decrypt GF(24)2 Polynomials in 22 nm Tri-Gate CMOS," IEEE JSSC, vol. 50, no. 4, pp. 1048-1058, April 2015.

[16] GNU Project, "Libgcrypt", https://www.gnu.org/software/libgcrypt/.

[17] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT - An Ultra-Lightweight Block Cipher," in CHES 2007, vol. 4727, Springer-Verlag, 2007, pp. 450-466.

TABLE I. Comparison with Other Designs

Design name	Block size (bits)	Key size (bits)	Archi	Tech (nm)	#Cycles	Area (gates)	Power (µW) (@10MHz)	Throughput (Mbps) (@10MHz)	Energy /bit (pJ/bit)	Core Area (mm^2)
LUT-based S-box (LBS)	128	128	32-bit	65	44	6,315	142.80	**27.74**	4.91	0.0180
Canright S-box (CRS)	128	128	32-bit	65	44	**4,449**	189.00	**27.74**	6.50	**0.0074**
LBS (Encrypt.); CRS (Key Exp.)	128	128	32-bit	65	44	5,388	153.20	**27.74**	5.27	0.0094
Decode-Switch-Encode (DSE)	128	128	32-bit	65	44	6,531	**128.50**	**27.74**	**4.42**	0.0113
DSE (Encrypt.); Canright (Key Exp.)	128	128	32-bit	65	44	5,469	147.50	**27.74**	5.07	0.0096
Banik *et al*. [7]	128	128	32-bit	90	44	5,528	-	**27.74**	6.20	-
Good *et al*. [12]	128	128	8-bit	130	365	5,500	**100.00**	4.31@12MHz	22.00	0.0246
Satoh *et al*. [13]	128	128	32-bit	110	44	6,292	-	**400@137MHz**	-	-
Hocquet *et al*. [14]	128	128	8-bit	65	1142	3,400	0.85@890KHz@0.4V	0.1@890KHz@0.4V	5.78	0.018
Zhao *et al*. [9]	128	128	8-bit	65	160	4,000	-	7.40	**1.61**	0.008
Mathew *et al*. [15]	128	128	8-bit	22	216	**1,947**	170@0.34V	5.65	5.38	**0.0022**
PRESENT128 (our design)	64	128	64-bit (32b IO)	65	37	2,106	61.2	16.5	3.54	0.0037
PRESENT80 (our design)	64	80	64-bit (32b IO)	65	36	**1,850**	**50.2**	16.9	**2.90**	**0.0031**