Efficient ASIC Implementation of a NB-IoT Security Co-processor

Mohamed A. Sharaf[1], Eslam AbdelBary[1], Hassan Mostafa[12], Ahmed Hussein[1], and Amin M. Nassar[1]
[1]Electronics and Communications Engineering Department, Cairo University, Giza, 12613, Egypt
[2]Nanotechnology Department, Zewail City for Science and Technology, Giza, 12578, Egypt

***Abstract*—This work implements a hardware acceleration of selected confidentiality and integrity algorithms for Long Term Evolution (LTE) based on SNOW3G and ZUC stream ciphers. Structural similarities between both ciphers were utilized to combine a single configurable accelerator. Implementation is primarily optimized to cater for NB-IoT devices where lower resources considerations exist. Different implementation approaches are compared targeting a 130 nm technology resulting in an optimized hardware accelerator for 100 Mbps rate and utilizing cell area of 35.7 kGE. Maximum GDSII energy consumption after PnR is 1.74 pJ/bit. For the sake of comparison with current LTE cipher cores, the co-processor was characterized using a 65 nm technology at 2 Gbps rate achieving 47% area reduction for the cipher core without sacrificing average power consumption.**

I. Introduction

The number of Internet of things (IoT) connections of all types is forecast to reach around 25 billion by 2025 with 5 billion devices having cellular connections [1]. These huge amounts of low-complexity devices do not need to communicate with high frequency. Performance is not needed to be high, and low transmission latency is not a requirement. Many of these devices can be deployed in challenging radio environments and will be relied upon to exchange data for up to 10 year, without battery replacement.

Both 3GPP standardized NB-IoT (Narrowband IoT) and Cat-M are expected to account for over 50% of the devices. Out of the 100 service providers identified as having launched at least one of the NB-IoT or LTE-M technologies, 25% have launched both [2]. While very complementary to each other, they are addressing different types of use cases based on the strength of their capabilities. They communicate sensitive data like sensor reading, payment information, machine-to-machine commands, etc. NB-IoT is a subset of 3GPP's LTE (4G) targeting communication between base station (eNB) and a low-throughput IoT device providing extended coverage.

Generally, IoT applications target lightweight ciphering algorithms for their computational advantage. Implementing ciphering in hardware is more immune to security attacks than software implementations [3]. In addition, optimized hardware implementations consume much less power, with a negligible area overhead [4], [5].

This work presents an ASIC implementation of a security co-processor targeting NB-IoT. Similar implementations are introduced in [6], [7], and [8]. Crypto Processor [6] is mainly optimized for area. HW Accelerator [7] utilizes a top-level design reuse approach. HiPAcc-LTE [8] is targeting ultra high throughput, not for IoT.

This paper is organized as follows: security algorithms are briefly described in Section II. Section III describes the hardware implementation using a bottom-up approach. Also, the stream cipher core's different implementation methodologies are compared leading up to the final core design. Section IV compares back-end results with the related work. Section V concludes the whole work.

II. NB-IoT Security

3GPP technical specification [9] defines LTE (or NB-IoT) user-to-network security features: confidentiality (EEA) and integrity (EIA). Both are defined to use one of the cipher cores: SNOW3G, AES, and ZUC. EEA1 and EIA1 retake UMTS algorithms UEA2 and UIA2, respectively. This work targets the stream cipher-based security algorithms, SNOW3G (EEA1 and EIA1) and ZUC (EEA3 and EIA3).

A. Cipher Cores

SNOW3G and ZUC are stream cipher algorithms, that generate 32-bit output words, called keystream, using a 128-bit secret key and a 128-bit publicly known Initialization Variable (IV). The output word is used to mask plain text input producing the ciphered text. Three main components construct both ciphers, a linear feedback shift register (LFSR), a feedback function feeding the LFSR, and a Finite State Machine (FSM). The LFSR works as the cipher memory preserving its state during operation. LFSR is initialized with a combination of the key and the IV. The feedback function feeds the input of the LFSR with a value calculated from the LFSR and the FSM during initialization. The FSM performs non-linear manipulation on data coming from LFSR to generate the keystream. Ciphers go through an initialization phase for 32 iterations, before going into the working phase generating the keystream every cycle.

B. Confidentiality Algorithms

To protect data from being retrieved by unintended receivers, the sender encrypts the message using secret a key negotiated during communication establishment. Both EEA1 and EEA3 use the stream cipher core output keystream to encrypt the cipher text by bit XORing.

C. Integrity Algorithms

Authentication is to confirm that the message is received from the original sender and is not altered during the transmission. The sender pads the message with a 32-bit Message Authentication Code (MAC). Receiver confirms data origin and integrity by recalculating the MAC for the same message. Authentication passes if calculated MAC is identical to the received one. Integrity algorithms generate MAC using the keystream, the message, and the communication parameters.

1) SNOW3G-based Integrity: EIA1 (or UIA2) waits for SNOW3G cipher initialization, then stores five 32-bit keystreams into the variables z_1 to z_5. The message is manipulated in 64-bit words. MAC is calculated incrementally as a function of the message words and z using MUL function.

2) ZUC-based Integrity: Input message is divided into 32-bits words. EIA3 controls the ZUC cipher core to generate keystream words. MAC is calculated incrementally based on message bits. Every bit is used to decide on XORing the internal variable T with keystream bits or not.

III. Hardware Implementation

A. Stand-alone Cipher Cores

Multiple iterations were done on each cipher as a standalone core to reach the final design. Implementation trials were targeting the computationally expensive operations in each block. The parallel approach can reach high data rates at the expense of a higher area and power consumption. Analyzing both SNOW3G and ZUC internal operations across the feedback function and FSM, showed that the cipher core design can be serialized by a factor of 4 with minimal area and power overhead. Area is reduced by 23% and 24% for the serialized SNOW3G and ZUC cores, respectively. The SNOW3G serialized core consumes 20% more power, while the ZUC serialized core consumes 50% less power than their parallel counterparts at the same throughput of 100 Mbps.

In the SNOW3G feedback function, two implementations for the operators MUL$_\alpha$ and DIV$_\alpha$ were explored. A look-up table (LUT) and a combinational logic approach, with an area reduction of 24% favoring the combinational logic approach.

For ZUC's feedback function, up to 6 modulo 2^{31}-1 additions can be serialized over two adders with the overhead of having registers to store the intermediate addition values. Another option would be using the same adder for modulo 2^{31}-1 and modulo 2^{32}-1 additions executed in feedback and FSM.

Table I states the post synthesis area and power results for the different implementation approaches at constant throughput using Synopsys Design Compiler. SNOW3G design 4 and ZUC design 3 were selected for merge. Area is normalized to the smallest size of the a 2-input NAND gate in the TSMC 130 nm cell library of 4.7068 μ m^2. Power was estimated from feeding RTL activity to the post synthesis netlist.

B. Combined Cipher Core

Figure 1 shows the architecture of the combined cipher core.

1) LFSR: Since both cores use 16 register stages in the LFSR, but with different sizes: 32-bit for SNOW3G and 31-bit for ZUC, single LFSR can serve both cores with 16 32-bit registers while having an extra inactive 16-bit register in case of ZUC. This register is clock-gated to reduce power consumption.

2) Feedback: Since SNOW3G operates on 32-bit arithmetic while ZUC operates on 31-bit, each feedback function is implemented separately as shown in Figure 1. SNOW3G operators, MULα and DIVα, are both implemented using combinational logic but are only sampled every 4 cycles. ZUC feedback function contains 2 modulo 2^{31}-1 serialized adders and 4 31-bit registers holding intermediate addition values. The inputs of each feedback function are gated when its respective core is inactive.

3) FSM: Three 32-bit FSM registers are shared, with only two utilized for ZUC. Two shared 32-bit adders are used in calculating the keystream in both algorithms. The inputs of the FSM registers are based on different S-Boxes according to the target cipher.

SNOW3G S-Boxes are implemented using single instance of the two Rijndael S-Boxes S$_\text{R}$ and S$_\text{Q}$ with 2 sets of 3 8-bit registers holding the intermediate calculation values for each S-Box. Both S-Boxes are implemented using LUTs. Same implementation methodology is applied to ZUC. Internal nodes of the FSM block that are not shared between both ciphers, for example the S-boxes, are operand-isolated to further save power.

TABLE I
Cipher implementations comparison at constant throughput

#	Design	Area (kGE)	Power (mW)
1	SNOW3G (Parallel, LUT)	13.93	0.335
2	SNOW3G (Parallel, Comb.)	11.47	0.131
3	SNOW3G (Serialized, LUT)	11.31	0.138
4	SNOW3G (Serialized, Comb.)	8.77	0.166
1	ZUC (Parallel)	11.76	0.371
2	ZUC (Serialized)	8.99	0.220
3	ZUC (Serialized, Adders Sharing)	8.94	0.174
4	ZUC (Serialized, FB + FSM Sharing)	9.20	0.228

C. Confidentiality Controllers

Both EEA1 and EEA3 are combined in single EEA controller having a simple 32-bit parallel memory (or buffer) interface to read message blocks (words) and write ciphered message. Both input and output data are registered to reduce delays of the memory interface. The serialized cipher core is operating at a higher clock frequency, $clk\_4x = throughput/32 \times 4$. To reduce power consumption, top-level FSM, counters, and registers operate at a slower clock frequency, $clk\_1x = throughput/32$. Top-level registers are enabled by the FSM according to the state. $Operation\_Start$ control signal triggers the FSM to go from $Idle$ through the states $Core\_Initialize$ and $Core\_Keystream$. During $Core\_Keystream$, FSM will enable the core keystream generation and request data from the memory with the generated message block index from 0 to $L-1$ where $L =$

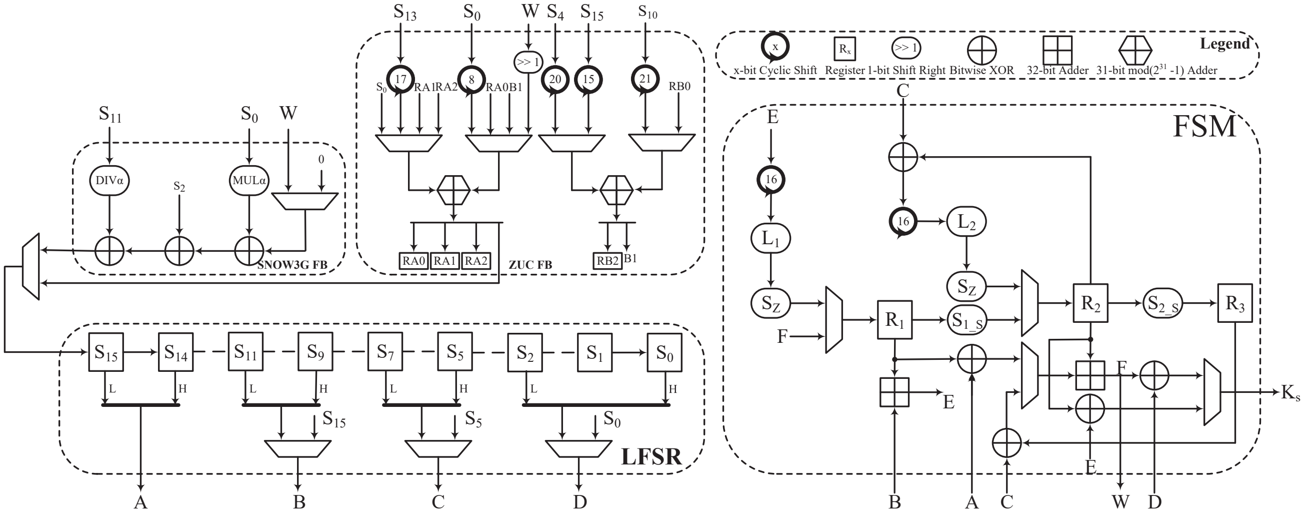

Fig. 1. Hardware architecture of the combined SNOW3G and ZUC cipher core.

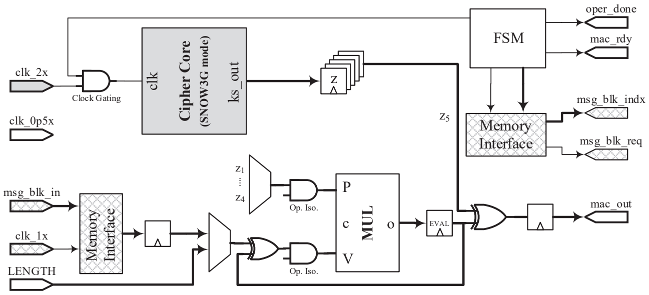

Fig. 2. EIA1 controller implementation architecture.

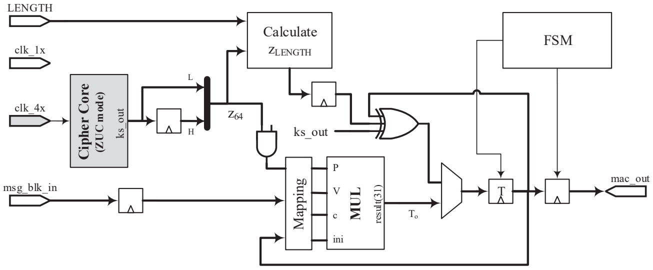

Fig. 3. EIA3 controller implementation architecture.

$\lceil LENGTH/32 \rceil$. If $LENGTH$ is not a multiple of 32, the last keystream LSBs will be replaced by zeros to avoid corrupting irrelevant memory data.

D. Integrity Controllers

1) EIA1: The algorithm uses 64-bit variables. Hence, to operate at the same input message throughput as the confidentiality core, the operating clock of the EIA1 controller is $clk\_0p5x = throughput/64$ and the cipher core operates at $clk\_2x = clk\_0p5x \times 4$, see Figure 2. Memory interface, operating at clk_1x, converts the 64-bit into 32-bit to unify the co-processor interface. FSM goes from $Idle$ through the states $Core\_Initialize$, $Core\_Keystream$, and $Calculate\_MAC$. Cipher core's clock is gated after generating z. Operand isolation is implemented to avoid useless MUL input toggling during $Core\_Initialize$ and $Core\_Keystream$ states.

The combinational MUL function is used such that inputs (V and c) are the same for the 64 MULxPOW function calls. Hence, single MULxPOW is instantiated while the outputs of the internal MULx 64 stages represent MULxPOW for different i values.

2) EIA3: To reduce overall co-processor area, EIA1's 64-bit MUL function is reused to implement EIA3's iterative 32-bit XORing for every message bit with minimal area overhead. MUL is operand-isolated is during $Core\_Initialize$, see Figure 3. During $Calculate\_MAC$, the multiplexer is controlled to select the 3-input XOR's output.

MUL is instantiated with mapping z_{64} to V and $c = 0$. T is mapped to the MSBs of the initial value of $result$. Message block is flipped and mapped to LSBs of P. The output is the MSBs of stage number 31. To reduce power consumption, stage number 32 inputs are operand isolated in EIA3 mode.

E. Toplevel

The 4 security functions are integrated. Interface ports are internally mapped to the relevant function according to a 2-bit FN_SEL. Single clock input clk_4x is used by an integrated Clock Generator to generate other clocks with fixed phase relations to guarantee same clock domain operation and timing in back-end flow. All clocks are gated in Clock Generator to avoid toggling of unused clock tree buffers. VCD activity files are generated from gate-level simulations using test vectors in [10] and [11] for accurate power estimation. Synthesis is performed using a target TSMC 130 nm technology for IoT

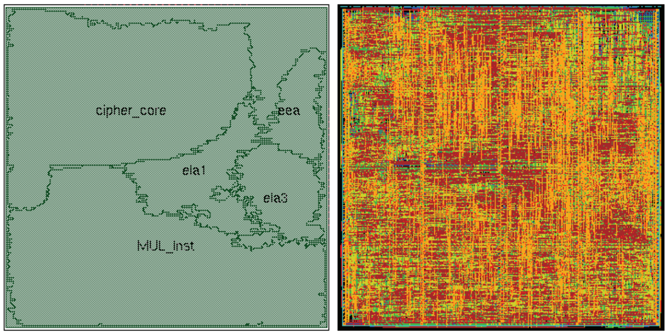

Fig. 4. Toplevel GDS. Amoeba view on the left. Physical on the right.

100 Mbps throughput and a target UMC 65 nm technology for LTE 2 Gbps throughput.

PnR is performed using Cadence Encounter with initial utilization of 75%. Six metal layers are used. Clock and reset ports exist on the left side, memory interface on the right side, and all other ports on the top. Small buffers are used in clock tree synthesis to reduce dynamic power consumption.

IV. Results and Comparison

The shared MUL and Cipher Core utilize 36.6% and 35.3%, respectively. EEA, EIA1, and EIA3 controllers occupy 5.7%, 10.9%, and 9.1%, respectively. Comparison versus related work is shown in Table II. Combined cipher core implementation achieves the lowest area of 12.6 kGE.

TABLE II
Area comparison versus related work

Block	Work	NAND (kGE)	AND (kGE)	Tech. (nm)	Throughput (Gbps)
Top-level	This (100M)	35.7	28.5	130	0.1
	This (2G)	37.1	24.7	65	2
	[6]	46.5	—	65	0.8
	[7]	—	20	90	2
Stream Cipher Core	This (100M)	12.6	10.1	130	0.1
	This (2G)	13.5	9	65	2
	[6]	16.6	—	65	0.8
	[7]	—	17	90	2
	[8]	27.4	—	65	28.8

Power estimation results compared with related work are shown in Table III. The 2 Gbps implementation achieves the lowest energy/bit in SNOW3G modes. ZUC modes consume more power due to the Cipher Core's serialized adders' implementation that is optimal for the IoT 100Mbps throughput.

V. Conclusion

An efficient hardware implementation of a security co-processor for NB-IoT is presented. Combining stream ciphers by utilizing structural similarities significantly reduced area and power. The co-processor is placed and routed using 130 nm technology, with a GDSII area of 47.6 kGE and consumes up to 6.7 pJ/bit at a typical IoT throughput of 100 Mbps. It achieves 47% cipher core cell area reduction targeting LTE throughput of 2 Gbps using 65 nm technology without sacrificing average power consumption.

TABLE III
Power consumption versus related work

Mode	Work	Syn. Power (mW)	Syn. E/B (pJ/b)	PnR Power (mW)	Supply (V)
EEA1	This (100M)	0.19	1.9	0.39	1.2
	This (2G)	1	0.5	2.43	1
	[7]	1.05	0.53	—	1
	[8]	17.32	0.6	—	1.32
EEA3	This (100M)	0.15	1.5	0.31	1.2
	This (2G)	1.2	0.6	3.26	1
	[7]	0.85	0.43	—	1
	[8]	16.83	0.58	—	1.32
EIA1	This (100M)	0.13	1.3	0.32	1.2
	This (2G)	0.46	0.23	1.07	1
	[7]	1.1	0.55	—	1
EIA3	This (100M)	0.29	2.9	0.67	1.2
	This (2G)	1.33	0.67	3.48	1
	[7]	0.9	0.45	—	1

Acknowledgment

This work was supported by the Egyptian Information Technology Industry Development Agency (ITIDA) under ITAC Program PRP2018.R25.23.

References

[1] Ericsson. IoT Connections Outlook. November 2019. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/november-2019/iot-connections-outlook

[2] GSA. NB-IoT and LTE-M Press Release. March 2019. [Online]. Available: https://gsacom.com/press-release/nb-iot-ltem-mar2019/

[3] M. A. Bahnasawi *et al.*, "Asic-oriented comparative review of hardware security algorithms for internet of things applications," in *2016 28th International Conference on Microelectronics (ICM)*. IEEE, 2016, pp. 285–288.

[4] N. Samir *et al.*, "Asic and fpga comparative study for iot lightweight hardware security algorithms," *Journal of Circuits, Systems and Computers*, vol. 28, no. 12, p. 1930009, 2019.

[5] ——, "Energy-adaptive lightweight hardware security module using partial dynamic reconfiguration for energy limited internet of things applications," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–4.

[6] L. Cavo, S. Fuhrmann, and L. Liu, "Implementation of an area efficient crypto processor for a nb-iot soc platform," 10 2018, pp. 1–5.

[7] S. Traboulsi, V. Frascolla, N. Pohl, J. Hausner, and A. Bilgic, "A versatile low-power ciphering and integrity protection unit for lte-advanced mobile devices," in *10th IEEE International NEWCAS Conference*, 2012, pp. 317–320.

[8] S. Sen Gupta, A. Chattopadhyay, and A. Khalid, "Designing integrated accelerator for stream ciphers with structural similarities," *Cryptography and Communications*, vol. 5, no. 1, pp. 19–47, Mar 2013.

[9] 3GPP TS 33.401, "Digital cellular telecommunications system (Phase e 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; 3GPP System Architecture Evolution (SAE); Security architecture. version 13.5.0 Release 13," 2017.

[10] ETSI TC SAGE Specification, "Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2 Document 3: Implementors' Test Data. Version 1.1," pp. 1–20, 2012.

[11] ——, "Specification of the 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3 Document 3: Implementor's Test Data. Version 1.1," pp. 1–21, 2011.