2017 30th International Conference on VLSI Design and 2017 16th International Conference on Embedded Systems

Hardware Software Codesign For A Hybrid Substitution Box

Anuroop K.B, Anu James
MTech Students
Adi Shankara Institute of Engineering and Technology
Kochi, Kerala, India
anuroop.ekm@gmail.com

Neema M.
Assistant Professor
Adi Shankara Institute of Engineering and Technology
Kochi, Kerala, India
neema.ec@adishankara.ac.in

***Abstract*— S-Box (Substitution-Box) is an important part of cryptographic algorithms such as AES, DES, Blowfish etc. This paper focuses on an efficient FPGA-based, reconfigurable AES S-Box design. This is the first S-Box architecture which uses hybrid search method (Z-TCAM), which enables to use the same design for encryption and decryption. The proposed S-Box architectures can be reconfigured using hardware or software based on the user's requirement which makes this unique compared to others. These S-Box designs can be made to work as a normal AES S-Box or a dynamic S-Box which is capable of modifying its contents on the go. Hardware-software communication was enabled by packaging the logic as an IP with AXI4 Lite interface. This IP works equally well for DES, Blowfish, and other similar algorithms. IP parameters can be changed to select the required S-Box type. The design offers low power consumption and high throughput compared to others.**

Keywords*— S-Box, AES, DES, Blowfish, FPGA, Z-TCAM

I. INTRODUCTION

With an increase in the usage of embedded devices that are capable of storing and transmitting data, information security has become a major concern. At present embedded devices are widely used for applications such as banking, aerospace communications, networking etc. which demands information security. In most cases, cryptography is used to secure the data. There are several cryptographic algorithms, which can be classified into Symmetric key, Asymmetric key and Hash functions. Symmetric key algorithms have higher speed and are highly secured. One of the most widely accepted symmetric key cryptographic algorithms is Advanced Encryption Standard (AES). Data Encryption Standard (DES), Blowfish etc. are some of the other symmetric key algorithms. All these algorithms can be implemented in hardware as well as in software. Even though software implementation is cheaper, it cannot be used in high-speed applications. But hardware implementation can be used for this purpose. In hardware implementation of AES, one of the most expensive sections, in terms of area, is Substitution Box (S-Box). It is used to create a non-linear relationship between plain text (raw input data) and cipher text (encrypted output data). This is achieved by replacing input data with another set of data based on some rules. The requirement of high-security application is pushing the designers to use a dynamic S-Box. In the case of a normal S-Box, input-output relations will be predefined but the Dynamic S-Box changes the input-output relations frequently. This can be based on Key or can be based on some other algorithms. This will increase the randomness and will secure the design from differential and linear cryptanalysis. For a dynamic S-Box -design, the designer will have to depend on memory-based S-Box architectures. These architectures suffer from loads of problems. High power consumption, speed limitations and high resource utilization are some of those. To configure a dynamic S-Box extra logic is necessary, which will add to the total resource utilization. So efficient memory based S-Box architecture is necessary to overcome all these problems. Our work targets to create 3 application specific memory based S-Box IP's which can be configured and accessed using hardware or software, which is efficient in terms of power consumption and resource utilization, which can be used as a normal memory if required and can support different algorithms such as AES, DES, Blowfish etc.

II. SUBSTITUTION BOX

S-Boxes are of different types (Fig. 1). The architecture of S-Box changes with respect to the algorithm in which it is used. AES algorithm uses fixed S-Box which takes 8-bit input and produces 8-bit output. A variety of methods can be used to implement AES S-Box. It can be implemented using a 256×8 bits look-up table, equivalent to 2048 bits or 2 Kilobits. Another method is by using block RAM cells. In most of the FPGAs, Block RAMs are available. If we are using it for S-Box design we will have to select 18K Block RAMs for using only 2K memory [1]. Block ROMs also suffer from this kind of problem. ASIC based designs and Xilinx 7 series FPGA based designs are free from these problems due to their constructional features. In [2], author uses Content Addressable Memory (CAM) cells for implementing AES S-Box and is an ASIC design. In the case of standard memories such as RAM, the address of a memory location can be given to read/write data from/to that particular location. In the case of CAM in addition to the basic read-write, it also offers a search option. When we use the search option, it will return the address of the matching data by comparing the input data with the stored data. In all other S-Box implementations, separate S-Box is required for encryption and decryption. In the case of CAM based S-Box (Because of the search option), same S-Box

2380-6923/16 $31.00 © 2016 IEEE
DOI 10.1109/VLSID.2017.46

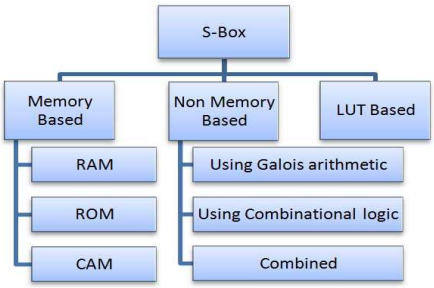

Figure 1: Types of S-Box

can be used for encryption as well as decryption. In our work, we have implemented CAM based S-Box using FPGA and compared it with proposed and existing architectures.

Galois field based S-Box [3][4] and Combinational logic based S-Box [5] are the two basic non-memory based S-Box architectures. Combining these two architectures another S-Box design was proposed in [6]. In [7] modification of the S-Box designed in the reference [5] was proposed. All these are implemented in different platforms. To compare the performance, we have implemented all the specified S-Box architectures based on the specified references in Zynq-7000 All Programmable SoC.

All the above-specified versions of S-Box are enhanced (Throughput/Area reduction) versions of traditional AES S-Box. The input and output relationship are preserved as in basic AES S-Box. There are only architectural differences. But nowadays loads of research works are going on to enhance the security of traditional AES S-Box [8]-[14]. Most of these modifications results in performance bottleneck and to avoid this problem, key dependent S-Box were used [8]-[11]. In addition to the specified advantage, dynamic S-Box or key dependent S-Box is resistant to differential and linear cryptanalysis as its structure is totally hidden from the cryptanalyst. The most well-known dynamic S-Box is in the Blowfish algorithm, which uses the cryptosystem itself to generate the S-Box contents [8]. Performance bottleneck is reduced in the sense that once the S-Box values are written, only after a sufficient number of rounds values will get modified. Setting the frequency of replacing S-Box data depends on the security requirement.

We need an efficient memory-based architecture for creating dynamic S-Box. RAM, CAM, Ternary CAM (TCAM), and Hybrid TCAM (Z-TCAM) are the available options for FPGA-based implementations. The problem of using RAM is, we will have to design both S-Box and inverse S-Box as two individual memory blocks. This will increase the resource utilization. In the other 3 cases, single S-Box can be made to function as S-Box and inverse S-Box. In the case of CAM, the search word should be specified exactly as it is stored in the memory. To increase the flexibility in searching data, TCAM was introduced. In addition to '1' and '0', TCAM enabled the usage of don't care 'X'. That means if we enter '1X11' as a search word, it will return the location of 1011 and 1111. One or more don't care states can be used. This facility made the design slow and high resource/power consuming. To avoid these problems Z-TCAM was introduced in [15], where a hybrid search mechanism was used to speed up the system. TCAM functionality is emulated using SRAM in the case of Z-TCAM. From the comparisons in [15], it is clear that Z-TCAM is better than TCAM in terms of speed and resource utilization. So in this work, we have designed CAM based and Z-TCAM based S-Box architectures. CAM architecture is based on [16] and the Z-TCAM architecture is based on [15].

In this work instead of going for the inbuilt FPGA memory, design is created in such a way that it emulates traditional SRAM cells. This will act as storage elements in our design as specified in [15]. This will help to solve the problem of memory wastage specified in [1]. Details regarding the design of basic Z-TCAM and CAM cells are not under the scope of this work. It is well documented in [15][16].

III. Implementation Details

A. *Galois field based S-Box [6]*

LUT based S-Box consumes large area which will increase the hardware cost. To reduce the hardware cost, Rijmen proposed the use of composite field arithmetic. Factors such as field of mapping, use of irreducible polynomial and choice of isomorphic mapping decides the complexity of arithmetic in a field. Therefore, the use of composite field arithmetic is the effective way to reduce the overall computation cost in the AES S-Box. In this work we have implemented the design in [6] shown in Fig. 2.

Multiplicative inverse module for S-Box is created using composite field arithmetic. The author claims that this work is better in terms of area compared to others. The details regarding the implementation of individual blocks such as Affine transformation, Addition, Multiplication etc. are available in [6]

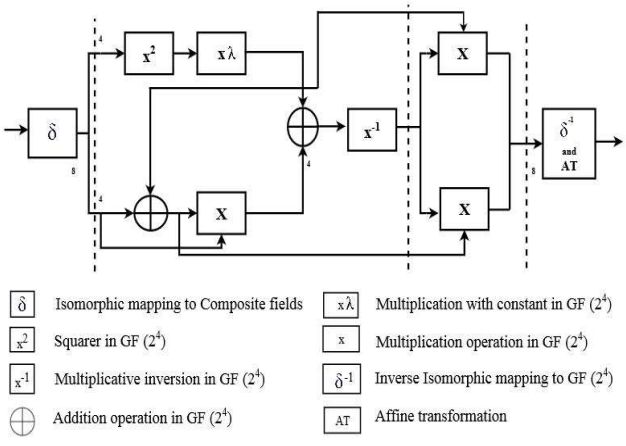

Figure 2: Galois arithmetic based S-Box

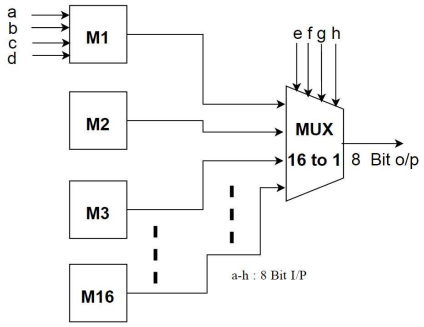

Figure 3: S-Box using 16 X 1 MUX

B. Combinational logic based S-Box (16X1 Mux) [5]

This design employs combinational logic to solve the unbreakable delay incurred by the look-up table, and reduces the critical path delay by using composite field arithmetic. The first 4-bit data input of most significant bit (MSB) will be the input of the sixteen module logic function (M1, M2, M3… M16) derived by Boolean simplification based on Karnaugh map. Remaining bits acts as the selection input of 16 to 1 multiplexer which selects the required output for S-Box. The structure of S-Box is shown in Fig. 3.

C. Combinational logic based S-Box (4X1 Mux) [7]

This design is a modification of the S-Box design in [5]. In this design 16 X 1 MUX used in [5] is replaced using 5, 4:1 MUX blocks. Moreover, pipelining registers are added in the data path to enhance the performance. (Fig. 4)

D. CAM based AES S-Box

Basic building blocks of CAM module are:

1) *SRAM cells emulated using FPGA:* These cells have 2 purposes;

- It will serve as storage element for the CAM
- It can be accessed as SRAM independently.

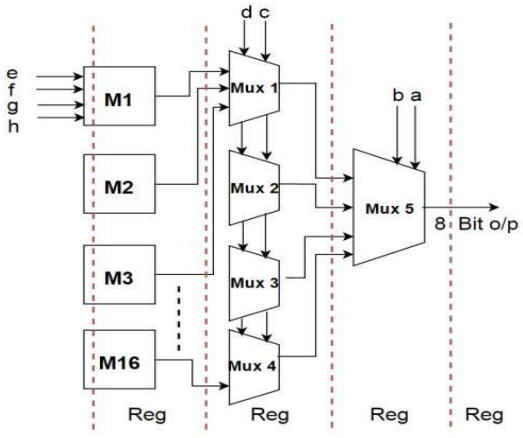

Figure 4: S-Box using 4X1 MUX

This is how both data and address are taken as output, for creating S-Box and inverse S-Box with the same module

2) *Comparators for each row of data:* Search data is compared with each row. If a match occurs it will set match output. For AES S-Box, there will be only one match, so extra circuitries to handle multiple matches can be avoided from this design.

3) *Priority encoder:* When we search for a particular data, there is a possibility that two or more words gives a match at the same time, so priority encoder is needed. S-Boxes are having unique entries so normal encoders are enough. This will further decrease the device utilization. Based on the input pattern to the encoder it will give address location as output.

Here the comparisons are done in parallel to increase operating frequency. Because of this number of comparators will be same as that of the depth of SRAM. Since we are using only the required sections of this structure, wastage of memory can be eliminated. An application specific design is created. Block representation of the design is shown in Fig. 5.

E. Proposed Hybrid AES S-Box

1) Z-CAM based (E:1): Block view of the proposed design is shown in Fig. 6. Major sections in the design are explained as follows:

a) Processing system (Standalone Processor/ Controller or a Soft-core based processing system): This block contains software to control the operation of S-Box. Based on the program loaded into this system, S-Box can be reconfigured. It is capable of accessing the memory through its control, address and data lines. The processing system can be activated if the application demands high security. It can be used to modify the contents of S-Box at scheduled times. This is how dynamic property is achieved for the proposed S-Box. It can be sent to sleep mode if the application is not demanding high security.

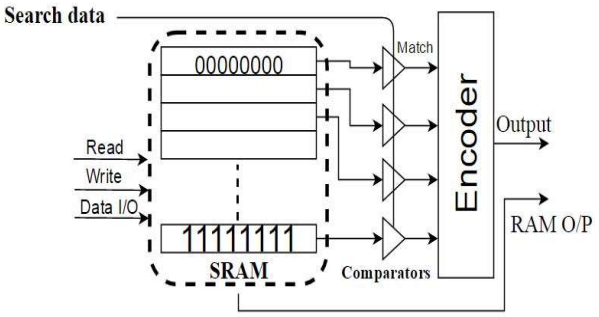

Figure 5: CAM based S-Box

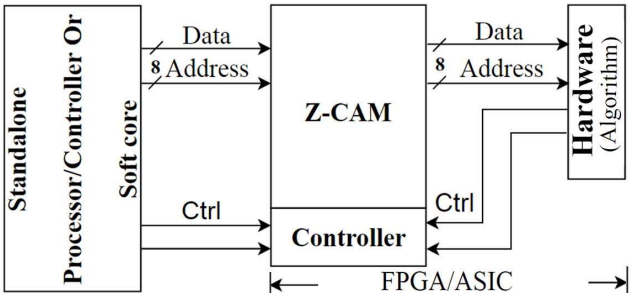

Figure 6: Proposed Hybrid S-Box using Z-CAM (E:1)

b) Z-CAM: This is used as a storage element in our S-Box. The reason behind choosing it in this design is its capability to give both address and data as output. Thereby it eliminates the need for an inverse S-Box. By default, there will be traditional AES S-Box contents in the Z-CAM. The contents can be modified using processing system or default values can be loaded based on the requirement.

In S-Box, we are not dealing with ternary data. Because of this, we can name it as Z-CAM. We are utilizing only the searching procedures used in [15]. Fig. 7 indicates the basic Z-TCAM configured as an 8 bit S-Box. In an AES S-Box, both address and data are 8 bit wide, so the input search data is divided into four 2 bit subwords. All the subwords are applied to 4 layers in parallel to get the potential match address (PMA). PMA is then applied to a CAM encoder to get the match address (MA). Fig. 8 shows the internal architecture of a layer. It contains 4 validation memories (VMs) each subword acts as an address to VM. If the memory location be invoked by a subword is high, it means that the input subword is present, otherwise absent. Thus, VM validates the input subword, if it is present. Next block is 1-bit AND operation then comes 4 original address table address memories (OATAMs), which stores the address at the memory location indexed by a subword and that address is then used to invoke a row from its corresponding OAT. Next section is 4 original address tables (OATs) as the name implies it stores the original addresses. Then comes 8-bit AND operation and a layer priority encoder (LPE). LPE selects PMA among the outputs of 8-bit AND operation.

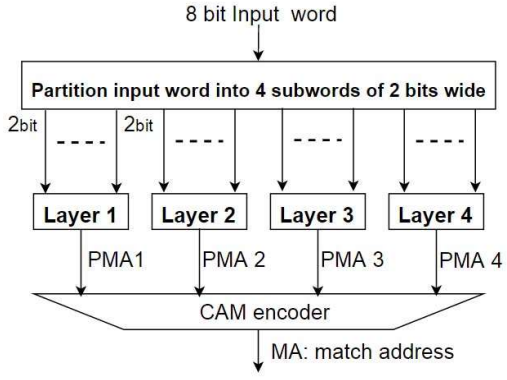

Figure 7: Z-TCAM architecture

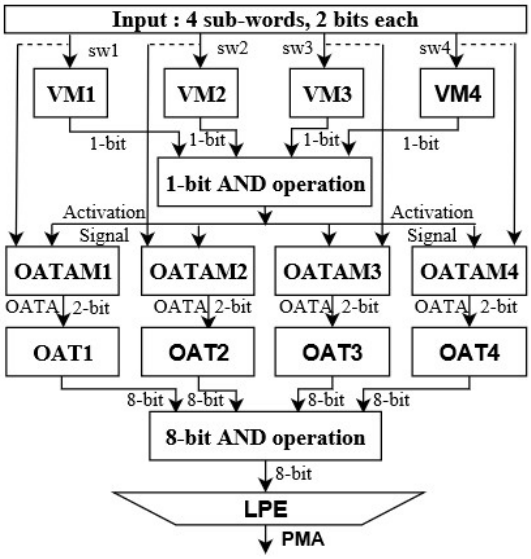

Figure 8: Internal architecture of a layer

A detailed description of the Z-TCAM structure is available in [15]. In this work, we are configuring Z-TCAM as 256*8 memory array to suite AES S-Box architecture.

c) Controller: Control signal input of the Z-CAM is common for both hardware and software. The function of controller section is to make the buses available for hardware or software based on the request. This will help to reduce the number of I/O pins.

The contents of the memory can be used as such, in the case of a static S-Box or it can be altered using processing system in case of dynamic S-Box. The design is highly flexible because the program can be changed to satisfy users need. Different types of algorithms are available in the literature to create S-Box entries. Any of them can be used in the processing system to load the memory with the enhanced S-Box entries. We can program it as a traditional AES S-Box or we can convert it to a key dependent S-Box. All these changes can be made by simply reprogramming the processing system.

2) Dual port Z-CAM based (E:2): In this design, the Z-CAM in the previous design is replaced using dual port Z-CAM. Reading and writing operations can be done from both hardware as well as software, without adding any features such as multiplexing which results in higher access time. The speed of operation will get limited if we are adding extra hardware modules to share address and data buses. The dual port Z-CAM is having no such problems. The block diagram of the design is shown in Fig. 9. When the software enable is pulled high using the processing system, the processing system can configure the Z-CAM. If this signal line is low the hardware algorithms can use the S-Box contents.

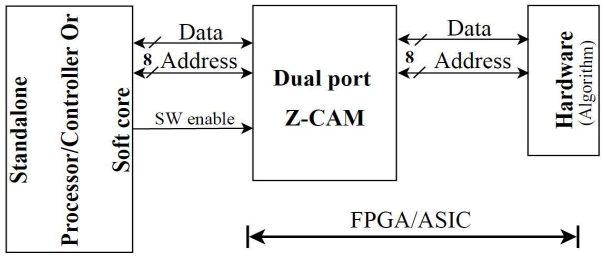

Figure 9: Proposed Hybrid S-Box using dual port Z-CAM (E:2)

In the applications which requires high throughput, the dynamic S-Box algorithm can be ported to hardware else it can be executed using the processing system. So this S-Box design is suitable for reconfigurable crypto applications. In the previous design, to achieve this functionality extra modules were added which limits the performance.

3) Dual port RAM based (E:3): As shown in Fig. 10 the Z-CAM is replaced using the dual port RAM. The operation is similar to the previous design. This is a pure memory based design. The LUT-Flip Flop pairs can be set free to occupy other modules in the design when we implement it using BRAM's. If the designer wishes to add other designs to same FPGA, he will have to create some free space inside the FPGA. In such situations the Z-CAM from the previous design can be replaced using dual port RAM provided the performance is not an issue. In this case to use inverse S-Box the Processing system/Programmable logic should load the values each and every time when required. That means switching from S-Box to inverse S-Box is not that simple as in the case of previous design. But this can be used to free up some slices to add more functionalities to the FPGA.

IV. Results Of Implementation

The existing and proposed architectures were designed using Verilog HDL. For synthesis, Xilinx ISE 14.6 and Vivado 2015.4 were used. For implementation, Xilinx Zynq-7000 All Programmable SoC was used.

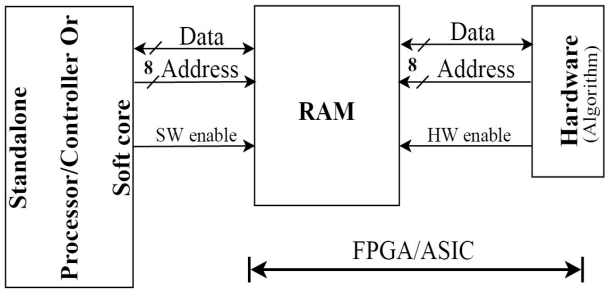

Figure 10: Proposed Hybrid S-Box using dual port RAM (E:3)

TABLE 1. COMPARISON OF S-BOX IMPLEMENTATIONS DONE IN THIS WORK

S-Box Types		Minimum period (ns)	Throughput (Gb/s)	Reconfiguration Capability	Need for separate Encrypt/Decrypt logic
Galois		11.685	.638	No	Yes
16 : 1 MUX		9	.828	No	Yes
4 : 1 MUX		1.891	3.940	No	Yes
CAM		1.891	1.970	No	Yes
Proposed	E:1 - Z-CAM based	4.424	1.684	Yes	No
	E:2 - Dual Port Z-CAM based	2.055	3.625	Yes	No
	E:3 - Dual port RAM based	1.724	4.322	Yes	No

Throughput = Number of bits processed /(No of clk cycles * minimum period)

Table 1 shows the minimum period of the clock in the S-Box designs, Throughput, reconfiguration details and the need for separate encryption and decryption logic. Xilinx ISE 14.6 was used to fetch the minimum period. From Table 1, it is clear that the proposed S-Box designs can be used for high frequency/throughput applications. The designs were implemented on Xilinx Zynq all programmable SOC. The remaining analysis was performed using Vivado 2015.4 on the same device. Table 2 indicates the post placement and route device utilization summary. If the designer is concerned about the area of the design Table 2 can be used to select an application specific S-Box based on area. Table 3 shows the dynamic power consumption of different S-Box designs. Power consumption of the combined encryption and decryption architectures and encryption only architectures are compared in Fig. 11.

Galois arithmetic S-Box is the best in terms of area, proposed Z-CAM based S-Box (E: 1) is the best in terms of power consumption and speed. Combining the results of Table 1, 2 and 3 we can infer that the Galois based S-Box is area efficient, but power consumption is 55% high compared to E: 1 and the speed of operation is 85.58% lower than E: 1.

TABLE 2. DEVICE UTILIZATION SUMMARY

S-Box Types		# of LUT's	#of FF's	# of IO's	# of Block/LUT RAM's
Galois		52	16	17	-
16 : 1 MUX		99	136	17	-
4 : 1 MUX		99	136	17	-
CAM		571	265	36	1
Proposed	E:1 - Z-CAM based	321	16	37	64
	E:2 - Dual Port Z-CAM based	332	26	54	64
	E:3 - Dual port RAM based	-	-	53	50

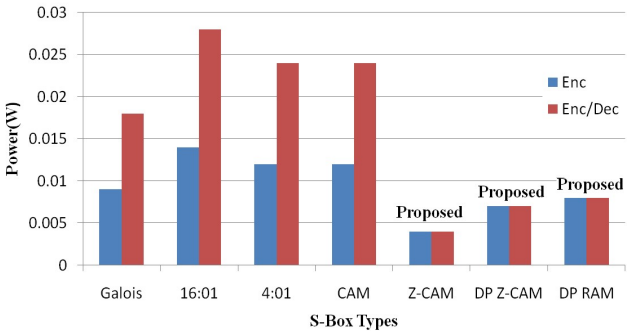

Figure 11: S-Box power comparison (Encryption only, combined encryption and decryption)

E: 2 can be used if both hardware and software requires read and write access to the system, minimum period is almost twice as that of E: 1. Power consumption is 43.8% high and LUT usage is 3% high compared to E: 1. To free up some LUT's, same functionality of E: 2 can be achieved using E: 3, with 16% increase in minimum period, 12.5% increase in power consumption compared to E: 2. This is for encryption only case.

Considering both encryption and decryption (Fig. 11), the power consumption of A, B, C and D is twice as that of the value listed in Table 2. This is because separate logic has to be designed for encryption and decryption. The proposed designs (E: 1-3) are free from this problem. In case of E: 3 there will be a latency to load values when we switch back from encryption to decryption or vice versa. E: 1 and E: 2 are having no such problems.

V. Conclusion

Power, speed and area efficient dynamic S-Box architecture has been introduced. The novelty of our work lies in the fact that, based on the application the contents in the S-Box can be configured using hardware, software or in a mixed mode (High randomness). The proposed E: 1 is a good candidate for (SOC based/hardware only) low power, low complexity and high-performance design compared to other S-Box designs. Device utilization of the Z-CAM based S-Box can be further decreased by the reduction of the number of tables used for the hybrid search. This will be pursued as a future work. Designs were implemented and tested on Zynq-7000 All Programmable SoC ZC702 Evaluation Kit.

References

[1] Nadjia, Anane, and Anane Mohamed, "Efficient Implementation of AES S-Box in LUT-6 FPGAs", *IEEE International Conference on Electrical Engineering (ICEE)*, December 2015

[2] Li, Hua, "A new CAM based S/S-1-box look-up table in AES" *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2005.

[3] Christy, N. Anitha, and P. Karthigaikumar, "FPGA implementation of AES algorithm using Composite Field Arithmetic" *IEEE International Conference on Devices, Circuits and Systems (ICDCS), 2012.*

Table 3. Power Consumption Of S-Box Designs

S-Box Types		Clocks (mW)	Signals (mW)	Logic (mW)	I/O (mW)	Total (mW)
Galois		1	1	1	6	9
16 : 1 MUX		1	2	1	10	14
4 : 1 MUX		2	1	1	8	12
CAM		3	2	2	5	12
Proposed	E:1 - Z-CAM based	1	1	1	1	4
	E:2 - Dual Port Z-CAM based	1	1	1	4	7
	E:3 - Dual port RAM based	1	1	1	5	8

[4] Rachh, Rashmi Ramesh, PV Ananda Mohan, and B. S. Anami, "High speed S-Box architecture for Advanced Encryption Standard", *IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application (IMSAA)*, 2011

[5] Ahmad, Nabihah, Rezaul Hasan, and Warsuzarina Mat Jubadi, "Design of AES S-Box using combinational logic optimization", *IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, 2010

[6] Joshi, Arundhati, P. K. Dakhole, and Ajay Thatere, "Implementation of S-Box for Advanced Encryption Standard", *IEEE International Conference on Engineering and Technology (ICETECH)*, 2015

[7] Soltani, Abolfazl, and Saeed Sharifian, "An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA", *Microprocessors and Microsystems, Elsevier, pp: 480-493* , 2015

[8] Alabaichi, Ashwak, and Adnan Ibrahem Salih, "Enhance security of advance encryption standard algorithm based on key-dependent S-Box", *IEEE International Conference on Digital Information Processing and Communications (ICDIPC)*, 2015

[9] Juremi, Julia, Ramlan Mahmod, and Salasiah Sulaiman, "A proposal for improving AES S-Box with rotation and key-dependent", *IEEE International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, 2012

[10] Niemiec, Marcin, and Lukasz Machowski, "A new symmetric block cipher based on key-dependent S-Boxes" *IEEE International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2012

[11] Nejad, Farshid Hossein, Saman Sabah, and Amid Jamshidi Jam, "Analysis of avalanche effect on advance encryption standard by using dynamic S-Box depends on rounds keys", *IEEE International Conference on Computational Science and Technology (ICCST)*, 2014

[12] Gopi, V., and E. Logashanmugam, "Design and analysis of nonlinear AES S-Box and mix-column transformation with the pipelined architecture", *IEEE International Conference on Current Trends in Engineering and Technology (ICCTET)*, 2013

[13] Wu, Jun, Yiyu Shi, and Minsu Choi, "FPGA-based measurement and evaluation of power analysis attack resistant asynchronous S-Box", *IEEE Conference on Instrumentation and Measurement Technology (I2MTC)*, 2011

[14] Mazumdar, Bodhisatwa, Debdeep Mukhopadhyay, and Indranil Sengupta, "Design for security of block cipher S-Boxes to resist differential power attacks", *IEEE 25th International Conference on VLSI Design (VLSID)*, 2012

[15] Ullah, Zahid, Manish K. Jaiswal, and Ray CC Cheung, "Z-TCAM: an SRAM-based architecture for TCAM", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems, pp: 402-406,* 2015

[16] Ditmar, Johan, Kjell Torkelsson, and Axel Jantsch, "A dynamically reconfigurable FPGA-based content addressable memory for Internet protocol characterization", *Field-Programmable Logic and Applications: The Roadmap to Reconfigurable Computing, Springer, pp: 19-28. Berlin Heidelberg*, 2000