IJCSN - International Journal of Computer Science and Network, Volume 9, Issue 4, August 2020
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

FPGA Based Lightweight Encryption Algorithm for Cyber Security Applications

[1] Bharathi R; [2] Bhagya R; [3]Anjan K Koundinya

[1] BMSIT & M, Bengaluru, India

[2] RVCE. Bengaluru, India

[3] BMSIT & M, Bengaluru, India

Abstract - Security and confidentiality are the prime factors in the field of Cyber security based applications. The Lightweight cryptography gives a solution tailored for the efficient VLSI implementations of resource-constrained devices. A high performance design for the PRESENT block cipher has been proposed. The designed architecture carry out the encryption operation by using key of 80 bit length and an input data of 64 bit. The simulation is carried through Xilinx ISE 14.7 design suite using verilog code and synthesized for Spartan-6 XC65LX45 FPGA device. The performance metrics like throughput, area and power are measured based on the synthesis report. The PRESENT block cipher consumes only 90 slices on total, hence the area consumed is around 0.75% and power consumed is about 36.61mW.

Keywords - *Lightweight,Cryptography,VLSI,Encryption,Spartan-6,FPGA*

1. Introduction

The Cryptography is the field of encryption methods to secure the information and communication techniques where the plaintext is transformed into cipher text using a key generated by cryptographic algorithm. The implementation of conventional ciphers is difficult in the conventional cryptography for resource constrained applications [1]. The standard cryptographic algorithms can be of larger size, very slow or highly energy consuming for the constrained devices.

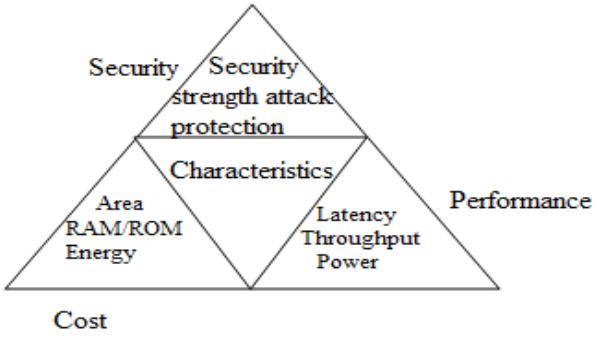

Figure 1

A wide variety of lightweight cryptography primitives are used over resource limited devices. The device spectrum has been divided into two main categories.

- Conventional cryptography: Desktop and servers, cell phones and tablets.
- Lightweight cryptography: CPS, Embedded systems, Sensor Networks and RFID.

The microcontrollers used with the embedded systems resist to adapt with the real time demands for traditional cryptographic strategies. Very less number of gates are present in RFID and sensor network devices for higher security and constrained with power drain on the device [2].

The field of lightweight cryptography studies new algorithms to overcome these problems. The designer has to come up with the trade-off among security, costs and the performance as shown in fig 1. An effective trade-off with security and performance helps to put forward some good solutions to hardware oriented applications [3]. The lightweight cryptography provides security solution for resource-limited devices.

Different lightweight block ciphers include KLEIN, LBlock, PRESENT, HIGHT, Piccolo, SPECK and AES. Among these block ciphers, the PRESENT block cipher has a compact nature for hardware implementation and serves as a benchmark for the new hardware oriented block ciphers and its efficiency higher [4].

Substitution box (S-box) is an only nonlinear part and essential constituent of different lightweight block cipher algorithms .During the process of encryption it creates

confusion in the plaintext [5]. For the improvement of PRESENT algorithm, one S-box is chosen among 16 good S-boxes. It is shown that, PRESENT algorithm provides more security than the fixed present S-box [6].

The proposed design can be implemented using verilog code and is simulated by ISIM simulator and synthesized by Xilinx tool. For calculating the bit-reversal of continuous-flow parallel data minimum latency and memory are measured [7]. The performance metrics are estimated after logical synthesis, map, and place and route compilation by Xilinx ISE 14.7 on the Xilinx Spartan-6 [8]. The designs for state-of-the-art are evaluated and compared, by using area, performance, energy and efficiency as metrics [9]. The hardware descriptions for the PRESENT cipher architecture is created in this work. The throughput is measured for 64 bit data path. The main Block RAM (BRAM) is utilized in FPGAs for storing the internal states, which reduces the number of slices. The S-boxes are realized within the slices [10][11].

Rest of the paper is structured as follows: Section II explains about the PRESENT algorithm. Section III illustrates the system model. Performance analysis and the results of simulation are explained in the Section IV. Section V gives the conclusion and future scope of the proposed work.

2. The Present Algorithm

The architecture is designed based on the algorithm called PRESENT algorithm [12]. Fig 2 shows a flow chart for PRESENT algorithm.

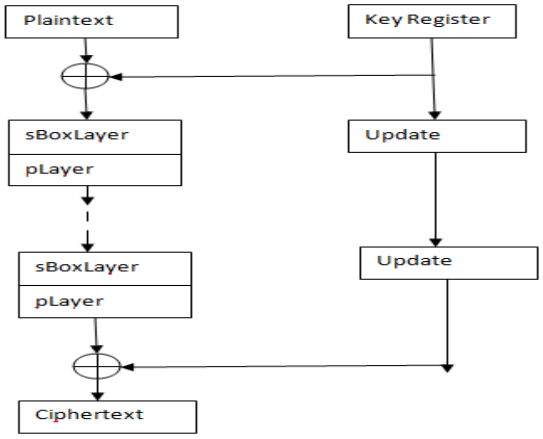

Fig 2 Flow chart for PRESENT algorithm

The algorithm uses 64 bit block size for encryption operation which supports two key lengths i.e., 80-bit and 128-bit. This algorithm uses Substitution-Permutation (SP) concept for encryption which has 31 rounds. XOR operation is performed on each round to generate round key Ki for 0<_i <_31.

There is a linear bitwise permutation layer (p-layer) and non-linear substitution layer (s-layer) based operation. A single 4 bit S-box is applied 16 times in parallel for each round in substitution layer. The four functions included in this algorithm are s-box layer and p-layer, key scheduling, add round key. The S-box is realized by using an area optimized combinational logic network. 80 bit is given to the key scheduling block which generates 31round keys for 31 individual rounds.

3. System Model

An iterative type of architecture is built for PRESENT lightweight cipher for saving the area and computing time. The proposed system is as shown in fig 3.

A 64 bit data path is chosen for the encryption operation. The architecture has three main components- encryption engine or the data path, key scheduling and a controller.

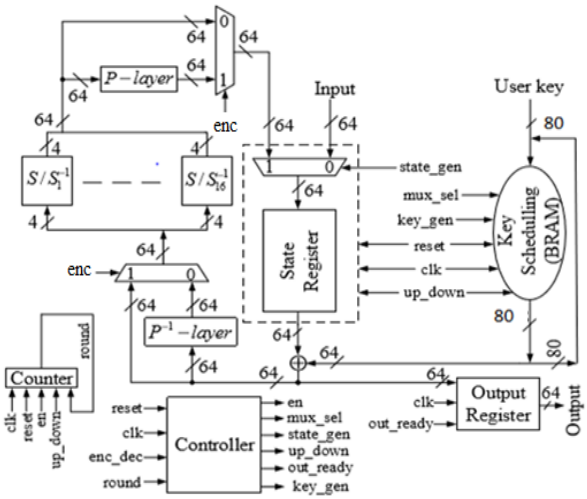

Fig 3 An iterative architecture for PRESENT lightweight cipher for encryption.

3.1 Data path for the Architecture

The data path supports the encryption process with key register. The internal states and the 80 bit register are stored in state register to store the intermediate round key. 64-bit multiplexer switches the input data between load and the round computational phase. The s-box layer (16 S-boxes) and one S-box present in the data path are used for key scheduling. Beside this, the architecture consists of

one 64 bit XOR gate, 5 bit asynchronous up-counter and 5 bit XOR gate.

The outputs and inputs are registered in the proposed architecture. The register is used for synchronizing the output at the last round. The output of the register is available after 33 clock cycles when all rounds are completed. A total of 33 cycles are needed for encrypting a single block of 64-bit input data.

3.1 Key Scheduling

The 64-bit register stores the round key. The state at the intermediate stage is XORed with first left 64-bits of the key register. Next, an 80-bit key is given to key register at the first clock as shown in fig. 2 which performs three steps.

- The output of the key register is rotated to the left by 61 bits

$K_{79}K_{78}.....K_1K_0 \Rightarrow K_{18}K_{19}...K_1K_0K_{79}K_{78}.....K_{20}K_{19}$

- First 4 bit is passed to S-box

$[K_{79}K_{78}K_{77}K_{76}] \Rightarrow S\,[K_{79}K_{78}K_{77}K_{76}]$

- 5-bit of key is XORed with the counter value

$K_{19}K_{18}K_{17}K_{16}K_{15} \Rightarrow K_{19}K_{18}K_{17}K_{16}K_{15} \oplus$ round_counter

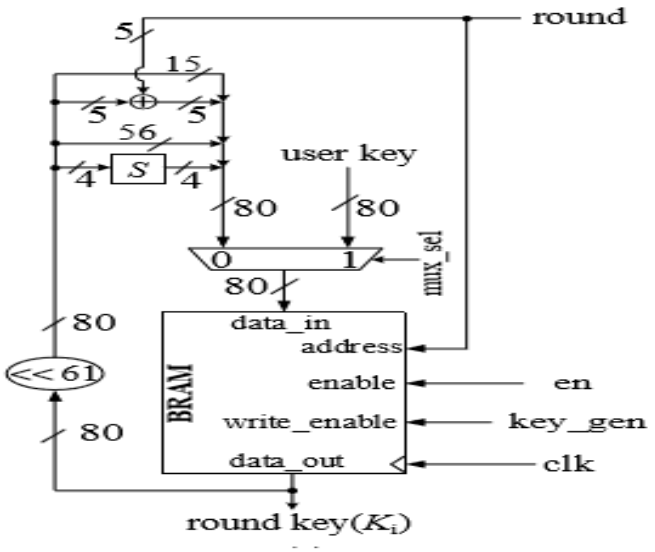

Fig 4 Key scheduling for PRESENT cipher with 80bit key.

3.2 Controller for Encryption

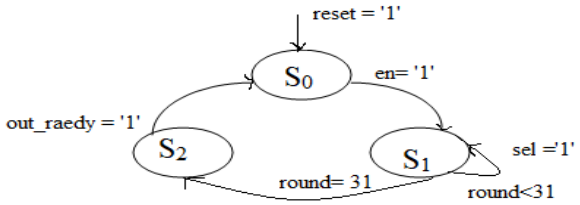

Fig 5 FSM for PRESENT block cipher

Various control signals are generated to control key generating process and data path for encryption operation. Four control signals are generated. They are en, out_ready enc_gen and sel. FSM has 3 states i.e., S0, S1 and S2. The en signal enables the counter in the S0 state and plaintext is given with key at sel='0'. When sel is at logic '1' the multiplexers are switched in state S1. The encryption and key registers enables intermediate operations with the help of enc_gen signal .The state is remained in S1 state till the value up-counter reaches 31 as it has 31 rounds. After that, the state is switched to S2 state. Here counter is disabled with a signal en='0'. The out_ready signal='1'. Finally the cipher text is present through output register in the next cycle.

4. Performance Analysis

The designed architecture is synthesized for the encryption of 64-bit input data with an 80-bit key on Xilinx ISE design suite tool of version 14.7. Fig 6 and fig 7 represents RTL schematic of P-box and S-box respectively.

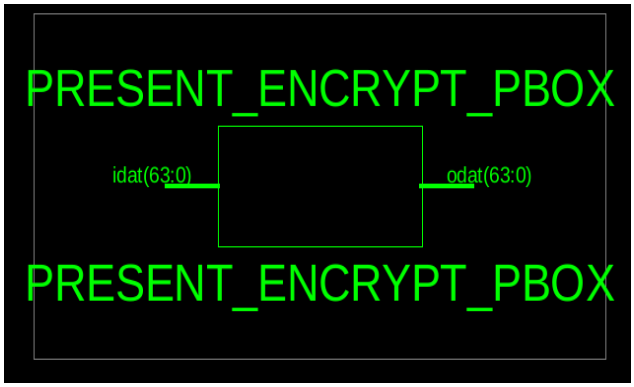

Fig 6 RTL schematic for P-box

Bit 'i' in state is moved to bit position P(i) by the p-layer

IJCSN - International Journal of Computer Science and Network, Volume 9, Issue 4, August 2020
ISSN (Online) : 2277-5420
www.IJCSN.org
Impact Factor: 1.5

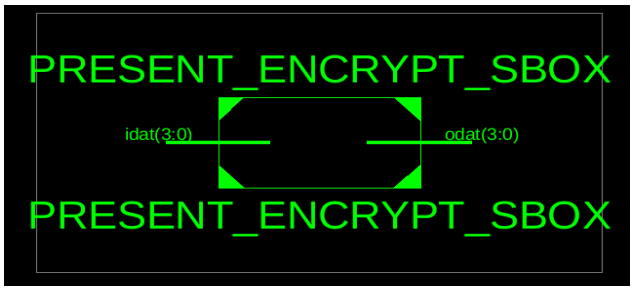

Fig 7 RTL schematic for S-box

4.1 Analysing output waveform through Xilinx ISE design

The 64-bit output data is obtained in the test bench waveform by simulating for different input plaintext data and key. The output is analyzed for different keys i.e. key0, key1 and key2 as shown in fig 8, 9 and 10 respectively.

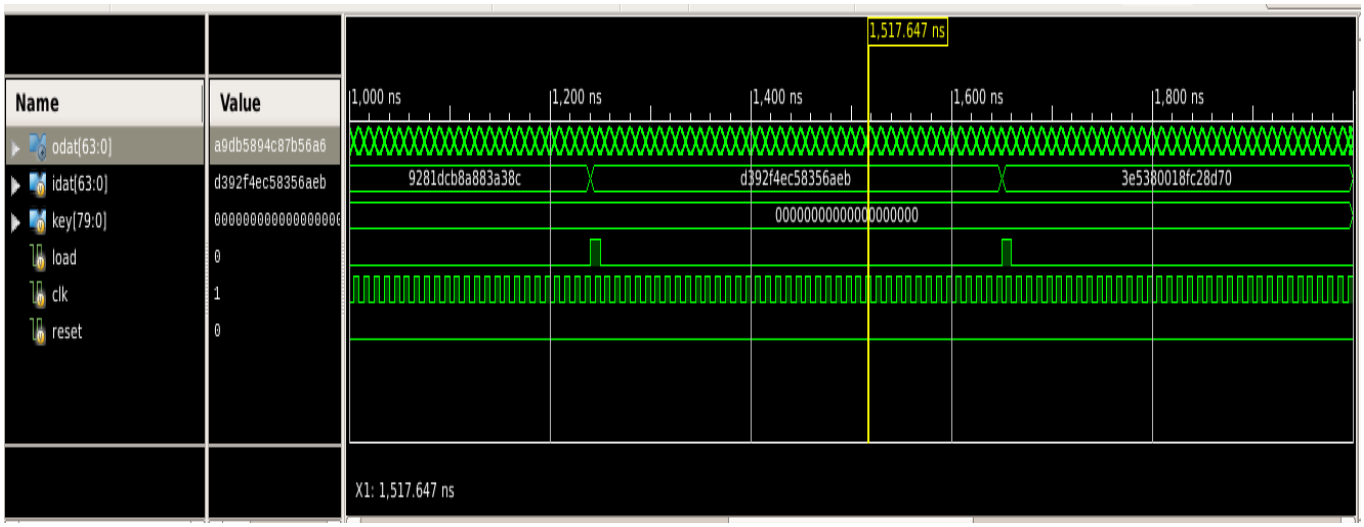

Fig 8: Output snapshot with key 0

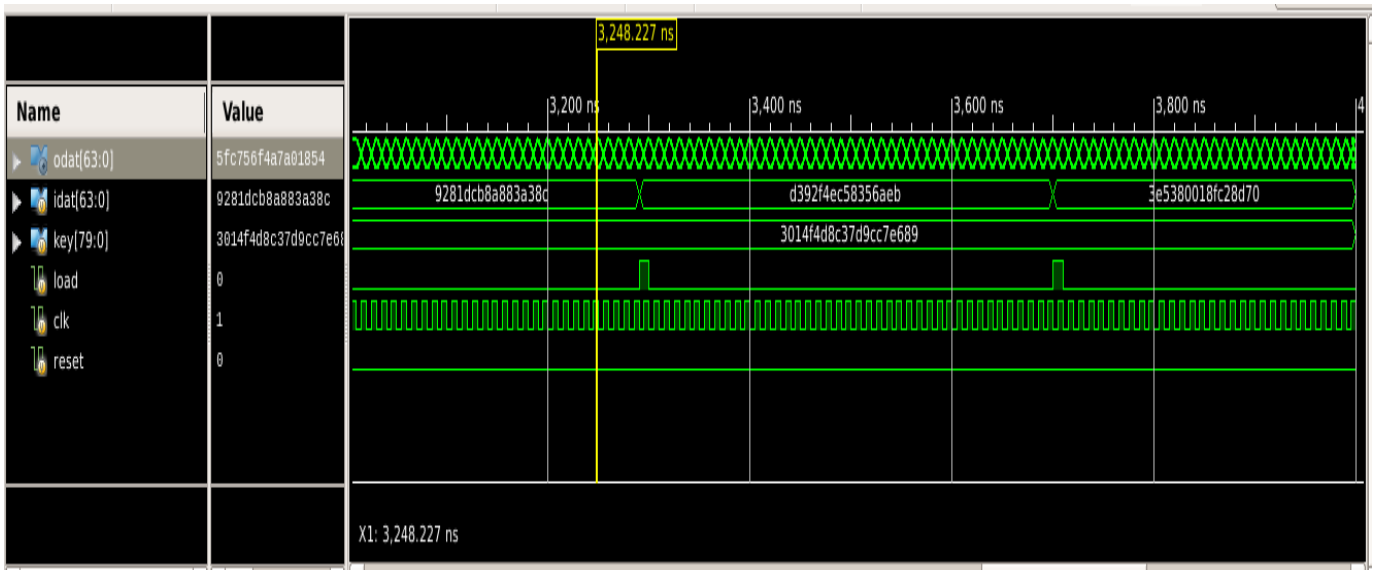

Fig 9: Output snapshot with key 1

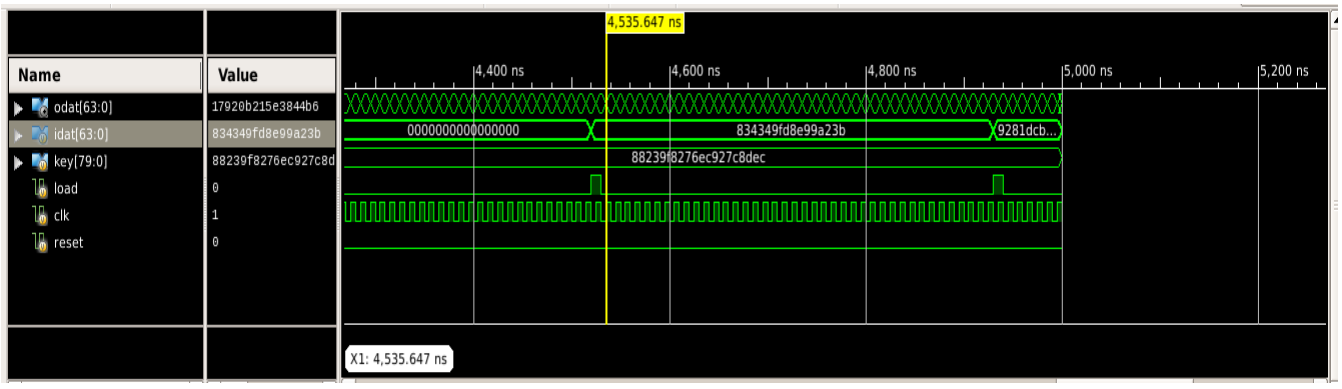

Fig 10: Output snapshot with key 2

4.2 Analysing the output through Chip scope

The program is dumped on the Spartan-6 XC65LX45 FPGA device as shown in fig 11 which has a speed grade equal to -3.

The program is dumped on the Spartan-6 XC65LX45 FPGA device as shown in fig 11 which has a speed grade equal to -3.

Fig 11: Spartan-6 FPGA device

The one-bit load is connected to A10 switch and clock is connected to L15 pin of the kit. The output data is analyzed through ChipScope application available in Xilinx tool.

As the Spartan-6 kit has only 8bit pins, it is difficult to show the 64bit output data. Hence ChipScope application which is available in the Xilinx tool is used to analyse the output waveform. Three steps are carried out through ChipScope as shown in fig.

- Creating and implementing a project in project navigator
- The ChipScope ILA core should be added to the design
- Using ChipScope analyser to debug the design

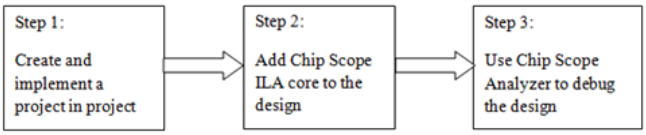

Fig 12: Chip Scope procedure

Fig 13 shows the RTL schematic for the design of complete PRESENT lightweight block cipher architecture for encryption which consists of sub blocks like P–box and S–boxes, multiplexer, ICON, IA, VIO cores etc.

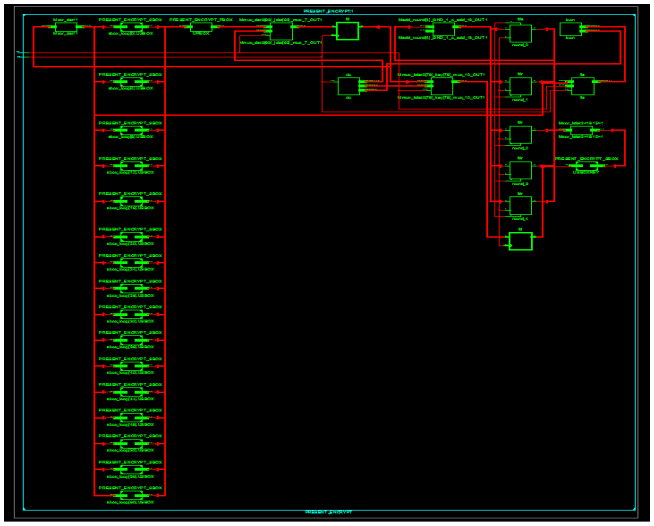

Fig 13: RTL schematic for Encryption

It is an integrated architecture designed with ICON, ILA and VIO cores to analyze the 64-bit output through the Chip Scope analyser.

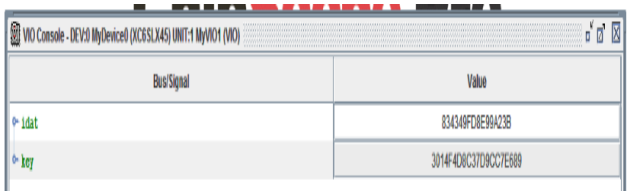

Fig 14: Giving input through ChipScope

The 64-bit input data (idat) and 80-bit key is entered to device through ChipScope as shown in fig 14

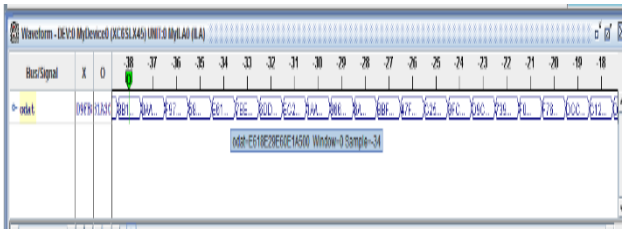

Fig 15: Output data through chip scope at instance 1

When the load is made to trigger through the kit, the waveform is generated as shown in fig 15.

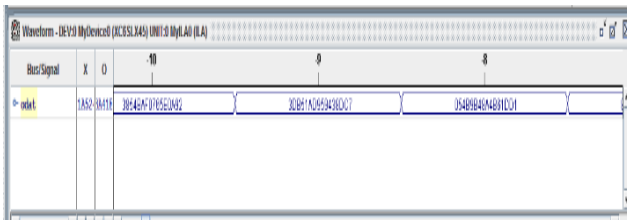

Fig 16: Output data through chip scope at instance 3

The output data (odat) shown in the fig 16 is 64-bit and is shown in the hexadecimal form. The output will be continuously changing as the clock is operating at 100 MHz..

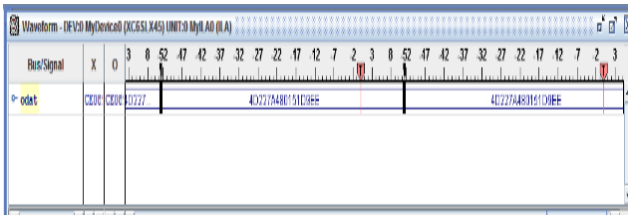

Fig 17: Output data through chip scope at instance 3

The black line in fig 17 indicates the load line which is made high and low for triggering purpose. After some point the odat remains same, even after changing the load to high and low.

Based on the synthesis report, the performance parameters are measured as shown in the table 1.

Table1: Device utilization on Xilinx Spartan-6 XC65LX45 FPGA

Parameters	Resources Available	Utilized Resource
Slice LUTs	27,288	239
Slice Registers	54,576	149
Total Slices	6,822	90
Bonded IOBs	218	210
Latency	--	111
Max.freq.(MHz)	--	296.046
Throughput (Mbps)	--	560.08

Two control signals used are control 0 and control 1. From the table 1 it is concluded that there is 0.75% utilization in the total number of slices consumed which gives the area and 96% utilization of Bonded IOs. The throughput is around 560.08 Mbps and is measured for 64-bit data path by the following expression,

$$\text{Throughput} = \frac{(\text{max. frequency} \times \text{total no. of bits})}{\text{Latency}} \quad (1)$$

The proposed architecture consumes around 36.61mW of power.

5. Conclusion

The PRESENT Lightweight cipher has been designed using verilog code and is synthesized through Xilinx ISE Design suite with the key length of 80 bit. Then the design is implemented trough Spartan-6 XC6SLX45 FPGA kit and the output is analyzed through the chip scope. Finally based on the synthesis report performance parameters are measured. When compared to other existing implementations, the proposed architecture performs better and provides high throughput. Further the design can be implemented with 128-bit key for the same input data for and analyse the performance and make use of different kit versions.

References

[1] Jai Gopal Pandey, Tarun Goel, Abhijit Karmakar, "A High-performance and Area-efficient VLSI Architecture for the PRESENT Lightweight Cipher", International Conference on Embedded Systems, January 2018.

2] William J. Buchanan, Shancang Li & Rameez Asif, "Lightweight cryptography methods", Journal of Cyber Security Technology, VOL. 1, NOS. 3–4, pp. 187–201, 2017.

[3] Chao Pei, Yang Xiao, Wei Liang, Xiaojia Han, "Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks," EURASIP Journal on Wireless Communications and Networking, May 2018.

[4] C.A. Lara-Nino, M. Morales-Sandoval, and A. Diaz-Perez, "Novel FPGA-based low-cost hardware architecture for the PRESENT block cipher", in 2016 Euromicro Conference on Digital System Design, 2016.

[5] 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", Cryptographic Hardware and Embedded Systems (CHES), Lecture Notes in Computer Science, pp. 450–466,2007.

[6] Sufyan Salim Mahmood AlDabbagh, Imad Fakhri Taha Al Shaikhli, "Improving PRESENT Lightweight Algorithm", International Conference on Advanced Computer Science Applications and Technologies, pp. 4799-2758,2013.

[7] Z. Ma, A. Hudic, A. Shaaban and S. Plosz, "Security Viewpoint in a Reference Architecture Model for Cyber-Physical Production Systems", 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW),Paris,2017,pp.153–159.

[8] Tadashi Okabe, "FPGA Implementation and Evaluation of lightweight block cipher – BORON", International Journal of Engineering Development and Research, Volume 5, Issue 1,2017.

[9] Hatzivasilis, George, Konstantinos Fysarakis, Ioannis Papaefstathiou, and Charalampos Manifavas. "A review of lightweight block ciphers." Journal of Cryptographic Engineering 8, no. 2 (2018): 141-184.

[10] Elif Bilge Kavun, Tolga Yalcin, "RAM-Based Ultra-Lightweight FPGA Implementation of PRESENT", International Conference on Reconfigurable Computing and FPGAs, pp. 2325-6532, Dec. 2011.

[11] Anjan K, Abhijith C, Arunraj, Deekshith N, Design and Mathematical Model of Hybrid Cryptographic Algorithm- A3D Algorithm, IJARCCE, Vol 3, Issue 6, Jun 2014.

[12] Bharathi R , N. Parvatham , "LEA-SIoT: Hardware Architecture of Lightweight Encryption Algorithm for Secure IoT on FPGA Platform", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, Issue 1, January 2020.