ASIC IMPLEMENTATION OF MULTI-THREADED PIPELINED AES CRYPTO PROCESSOR

Subramanya. N

Electronics Department B.M.S. College of Engineering
Bangalore, INDIA
n.subrahmanya.nag@gmail.com

Prapulla .G

Electronics Department B.M.S. College of Engineering
Bangalore, INDIA
prapulla.ece@bmsce.ac.in

Abstract - this paper presents AES algorithm implementation using multi-threaded and pipelining concept for faster data encryption and its physical implementation using 40nm technology. AES crypto-algorithm is widely used in Secured wireless communications, zigbee, Electronic financial transactions, Content protection, digital rights, set-top boxes etc. The AES algorithm is symmetric block cipher 128/192/256 bit key which takes 10/12/14 rounds for encryption respectively. The design provides highest throughput of 6.22 Gbps and dynamic power dissipation of 1.3209 mw.

Index Terms - AES, pipelined, cipher, parallelism

I. INTRODUCTION

Information security is a hot topic where in security of data is important in modern days. Data is protected from unauthorized person resulting in crypting of data using some key so that data is not read by unauthorized person which led to crypto analysis. In cryptography there are two types one symmetric key where a same key is used for encryption and decryption. Whereas In asymmetric different key is used at encryption and decryption. Early before 1997 Data encryption standard was used. Which was finally broken in 1998. It also suffered slow in implementing using software. Triple DES takes 3 times as many rounds of DES hence slower. Also DES uses 64 bit block size so less security. On 26th November 2001 National Institute of Standards and Technology (NIST) of the United States finally choose Rijndael out of fifteen algorithm which is called Advanced Encryption Standard (standard version) . The AES encryption is considered to be efficient to implement in both hardware and software. Compared to software, hardware implementation is more reliable with respect of speed. Some works have been presented on hardware implementations of the AES algorithm using ASIC [5], [6].

II. AES ALGORITHM

AES is a symmetric key block cipher which means same key is used for both encryption and decryption. key size can be 128, 192 or 256 bits. if key size is 128bit it requires 10 rounds of data encryption . Whereas it is 12 & 14 for 192 & encryption process starts with add round key stage followed by nine rounds of four stages followed by tenth round of three steps. The four stages of encryption are

a] Substitute bytes
b] Shift rows
c] Mix columns
d] Add round key

The tenth round leaves out mix columns stages. The first nine round of decryption are

a] Inverse shift rows
b] Inverse substitute rows
c] Inverse mix columns
d] Add round key

Again the tenth round leaves out inverse mix columns stage. Each of these is considered in detail as follows[3].

A] Sub bytes / Inv Sub Byte transformations: Sub byte transformation is a non-linear byte of substitution. This can be done by using two methods either by using LUT approach or by using combinational logic. The LUT based approach is used in this design. In the Sub Bytes step, each byte in the matrix is replaced using an 8-bit substitution box. In the inverse Sub Bytes step, each byte in the matrix is replaced by using an inverse 8-bit substitution box

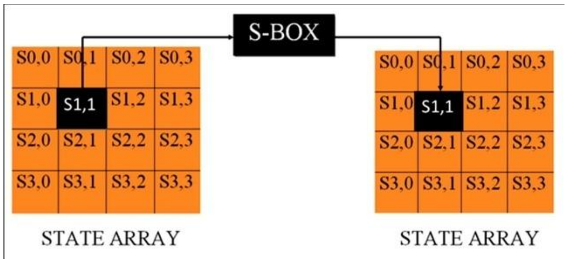

Figure 1 sub byte transformation

B] Shift Rows/Inv Shift Rows: Shift Rows is a simple shifting transformation of thirty two bits. First row of the does not change while the second, third and fourth rows cyclically shifted by one byte, two bytes and three bytes to the left, respectively. In the Inv Shift Rows, the first row of the State is kept as it is, while the rest of the rows are cyclically shifted to the right by the same offset as that in the Shift Rows.

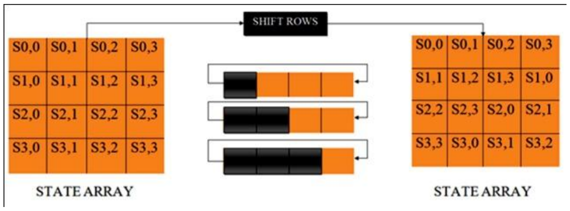

Figure 2 shift row transformation

2.3 Mix Column / Inv Mix Column transformation: This transformation operates on the columns of the State, treating each columns as a four term polynomial the finite field GF(28). Each columns is multiplied modulo x4+1 with a fixed four-term polynomial a(x) = {03}x3 + {01}x2 + {01}x + {02} over the GF(28) in mix column

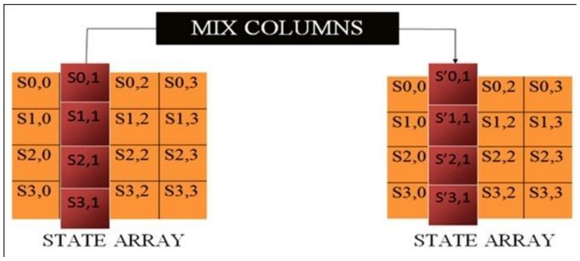

Figure 3 mix column transformation

2.4Add Round key: During the Add Round Key transformation, the round key values are added to the State by means of a simple exclusive or (xor) operation.

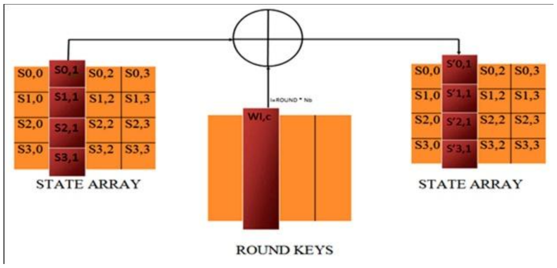

Figure 4 add round key transformation

All the above modules are implemented to give output after one clock cycle.

2.5. Key expansion: In the AES algorithm, the key expansion module is used for generating round keys for every round. There are two approaches to provide round keys. One is to pre-compute and store all the round keys, and the other one is to produce them on-the-fly. In this design first approach is used where a group of registers/memories are used and stored.

III. MULTITHREADING / PARALLELISM APPROACH

Traditional implementation of AES computation of data blocks is done serially .hence efficiency and speed is not good as output is dependent on previous input from work [1].

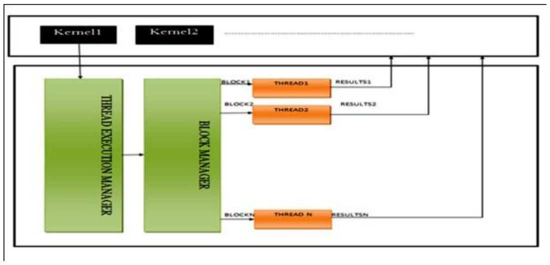

Figure 5 principle of parallelism

In graphical processing unit, it has three components: the thread execution manager, the block manager and multiple threads. when CPU calls the kernel function executed. It will enable the block manager active through the thread execution manager. The block manager will then divide plaintext into multiple blocks Then, each block will be computed in individual thread. Finally, the encrypted block will be outputt ed to CPU, which will be assembled into the cipher text. Since GPU allows the number of thread in parallel to be the magnitude of one hundred thousand. Therefore, the AES encryption has high efficiency using parallel computing.

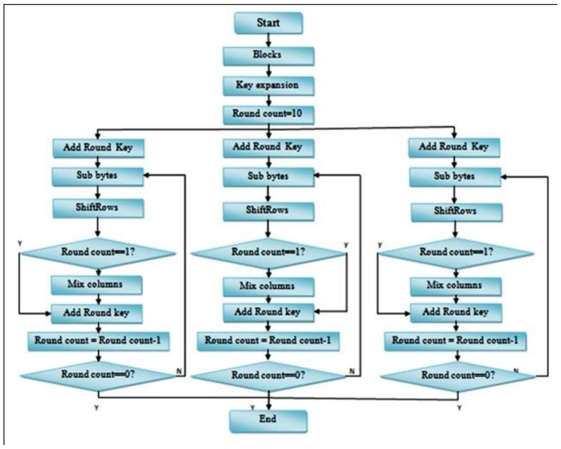

Figure 6 parallel AES algorithm

According to the principle of AES parallelism, the parallelized AES algorithm illustrated as follows. In this design by using parallelism concept four blocks of data is encrypted in parallel resulting in increase in throughput of four times the speed of regular AES. Instead of using four key expander to encrypt four blocks of data a single key expander is shared by all by all the modules resulting in reduction in area and power consumption but increase in throughput based on threading concept of operating system.

IV. PIPELINING

To further speed up data encryption pipelining is used. Basically pipelining means to process the data that is given as input in a continuous manner without having to wait for the current process to get over. This pipelining concept is seen in many latest processors. Now instead of waiting for encryption of current block is completed. Another data to be encrypted is also pipelined. Since each round of encryption consist of maximum 4 stages. Four blocks of 128 bit data can be encrypted at a time resulting in increase in throughput by nearly 4 times the speed of conventional AES. Additional three cycles are required to collect the processed data.

By combining both parallelism and pipelining it is possible to increase the throughput nearly to sixteen times the speed of conventional AES for both encryption/decryption is what is been proposed in this design.

V. IMPLEMENTATION

The Verilog HDL is used to implement the design. The design has eight modules four for encryption and four for decryption, which will operate in parallel. Key expansion is shared by encryption and decryption modules. Figure shows the architecture for multithreaded pipelined AES crypto processor.

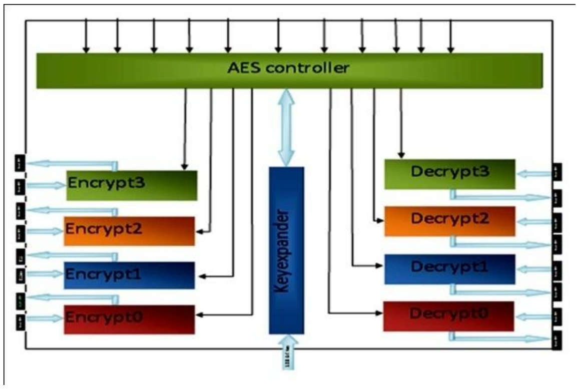

Figure 7 Architecture of multithreaded pipelined AES crypto processor

The key expander processes the data whenever a start signal is given & produces different keys according to input 128 bit key. It stores the values in a multi-dimensional array which can be accessed by encrypt and decrypt modules. Each encrypt/decrypt modules operates on 512 bit input and gives out 512 bit output. Each encrypt modules has 10 rounds of encryption of which each round has four stage which are pipelined to process four blocks of 128 bit data. a 4:1 multiplexer is used to select a four block of 128 bit data at input and similar 4:1 demultiplexer is used to collect the data as shown below in figure . Add round key module is implemented using xor operation. Each of the modules in encrypt as shown below is made give output after one clock cycle. Hence to encrypt input data of 128 bit a minimum of 40 clock cycles are required. A controller will control all the modules in parallel.

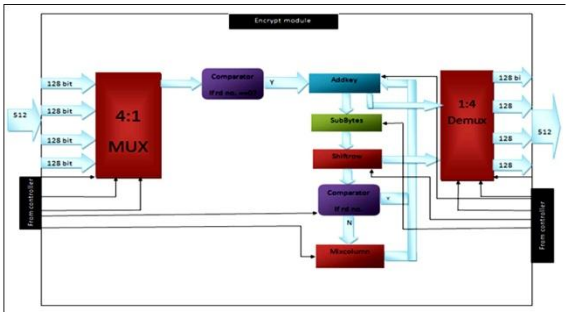

Figure 8 Architecture of pipelined encrypt module

VI. ASIC IMPLEMENTATION

The RTL code is simulated using cadence nc-verilog Simulator and functionally verified. The gate level synthesis is done using synopsys DC compiler by providing timing constraints in sdc files and gate netlist is generated using standard libraries. Once netlist is verified it is sent for physical implementation. Synopsys ICC compiler is used for physical design.

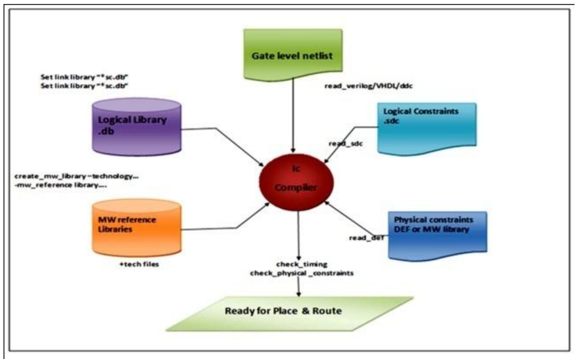

Figure 9 ICC_compiler data flow

ASIC implementation has 5 major steps

a] Floor planning: The main objectives of floor planning are to minimize area, timing. During floor planning size of chip is estimated, various blocks of the design are arranged on the chip.

b] Power planning: Each cell must be connected to power and ground along its edges. power rails must be distributed across the design. Routing of power and ground nets is often given first priority, because the power and ground wires are usually laid out entirely on a metal layer(s) due to its low resistivity

c] Placement: Placement is a step in the Physical implementation process where the position of standard cells are defined. The main objective is to make the chip as dense as possible, reduce total wire length. Above steps are carried out by having some conditions such as timing constraint is met and placement is routable.

d] Clock tree synthesis: The goal of clock tree synthesis (CTS) is to minimize skew and insertion delay. Also to fix some set up & hold violations.

e] Routing: Routing uses metal interconnects to create physical connections for all clock and signal pins. Routed paths must be able to meet some of the constraints such as setup and hold. Detailed routing is done to fix DRC's. At the end layout versus schematic is done to check errors.

VII. RESULTS

The functionality of the design is verified using cadence nc-verilog simulator. Output waveform is shown in figure

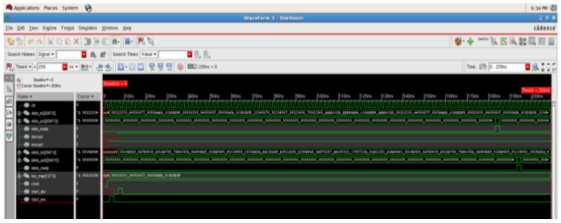

Figure 10 Simulation result of multi threaded pipelined AES crypto processor

The layout after ASIC implementation is shown in figure

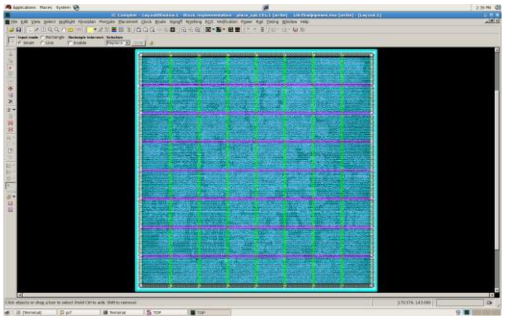

Figure 11 Final layout after routing stage

Power consumption at 0.95v

Cell Leakage Power	1.1254 mW
Cell Internal Power	769.4731 uW (58%)
Net Switching Power	551.4118 uW (42%)
Total Dynamic Power	1.3209 mW (100%)

Cell count

Hierarchical Cell Count	76
Hierarchical Port Count	12803
Leaf Cell Count	111008
Buf/Inv Cell Count	22918

Area details

Combinational Area	106407.714801
Noncombinational Area	38767.105437
Net Area	61133.239001

Comparison with earlier works

Works	Technology	Frequency (max) MHz	Throughput	year
5	350nm	132	2.41 Gbps	2004
6	130nm	12	N/A	2010
Proposed	40nm	142.85	6.22Gbps	2013

VIII. CONCLUSION

In conclusion, a hardware implementation for the AES multithreaded pipelined AES crypto processor was designed and verified using the Verilog hardware description language. The implementation of both parallel and pipelining concept have been implemented and verified successfully. The hardware model was then synthesized using the Synopsys Design Compiler tool. It has been observed that the Key expander takes 45 cycles to generate all the keys for 128bit key. After 45 cycles of clock the encryption throughput is 6.65Gbps and decryption throughput is 6.22Gbps. if key expander time is also taken into account output throughput is 3.21Gbps. It works at a maximum frequency of 142.8 MHz it consumes a power of leakage power of 1.1254mw and dynamic power of 1.3209 mw. It can be extended for low power implementation in order to reduce consumption of power. Throughput of above model can be further increased by using sub-pipelining concept.

IX. REFERENCES

[1] Deguang Le, Jinyi Chang, Xingdou Gou, Ankang Zhang, Conglan Lu-"Parallel AES algorithm for fast Data Encryption on GPU" Computer Engineering and Technology (ICCET), 2010 2nd International Conference IEEE April 2010 pp. V6-1 toV6-6, 16-18.

[2] J. Daemen and V. Rijmen, -AES Proposal: Rijndael, AES algorithm submission, September 3, 1999, available: http://www.nist.gov/ CryptoToolkit

[3] Draft FIPS for the AES available from: http://csrc.nist.gov/encryption.aes , February 2001.

[4] A. Hodjat and I. Verbauwhede,- Minimumarea cost for a 30 to 70 Gb/s AES processor, in Proc. IEEE Comput. Soc. Annu. Symp, Lafayette, LA, Feb. 2004, pp. 83–88.

[5] Refik Sever, A. Neslin, IsmailoGu, Yusuf C. Tekmen, Murat askar Tubitak, bilten " A high speed ASIC implementation of the Rijndael algorithm" 2004 IEEE vol II- pp. 541-544.

[6] Tim Good and Mohammed Benaissa 692-nW Advanced Encryption Standard (AES) on a 0.13- µ CMOS IEEE transactions on very large scale integration (VLSI) systems, vol. 18, no. 12, December 2010 IEEE Dec 2010 pp. 1753-1757

[7] A. Hodjat and I. Verbauwhede,- Minimum area cost for a 30 to 70 Gb/s AES processor, in Proc. IEEE Comput. Soc. Annu. Symp, Lafayette, LA, Feb. 2004, pp. 83–88.