ASIC IMPLEMENTATION OF 128 BIT KEY RC5 CIPHER

[1]Raghunatha & [2]Sowmya Sunkara

Department of Electronics & Communication Engineering, BMSCE, Bangalore
Email: [1]raghunathmd@gmail.com, [2]sowmi.ece@bmsce.ac.in

Abstract -This paper presents the design of RC5 algorithm and its physical design implementation. The RC5 cryptographic algorithm is widely used in wireless communication networks like WSAN and WBSN. RC5 is a fast block cipher, developed by RSA Security, which exploits data rotation to achieve a high level of nonlinearity. 128 bit key and 12 rounds of operation in the design will provide great security to the user's data. The RC5 hardware implementation will make the computation faster. The optimization of area, timing and power is done during physical design process. The design works at maximum frequency of 100MHz and power dissipation is 0.976mW.

Index Terms— cipher, low power, net-list, RC5.

I. INTRODUCTION

With the wireless communication being the emerging technology, the need to have secure data transmission is of great importance. Today, it is important that data is sent confidentially over the network without fear of hackers or unauthorized access to it. This makes implementing security in networks a vital demand. There are a numerous encryption algorithms that are now commonly used in computation. Here RC5 cipher patented by RSA Security is presented and implemented. A hardware system design is proposed to improve its performance. RC5, a fast block cipher, was proposed in 1994, which exploits data rotation to achieve high level of nonlinearity. RC5 symmetric cipher provides data protection via the use of a secret key only known to the encryption and decryption ends of the communication path. Communication applications such as Wireless Transport Layer Security (WTLS), which is the security layer of Wireless Application Protocol (WAP) and WSAN (Wireless Sensor Area Network), uses RC5 algorithm to protect the user's privacy.

Generally, implementing ciphers in software based on its speed in terms of computation is not efficient and hence the use of hardware devices is an alternative. The market for consumer and wireless devices is changing rapidly with the convergence of applications, standards and usage. The complexity challenges are forcing the design of devices at the sub-micron technology. The VLSI industry made this feasible using ASIC and FPGA technology with the help of on-chip memories. This not only has the advantage of added security but also increases speed of operation as it decreases the data transaction latency between the core and the memory.

The RC5 algorithm written in verilog is simulated using Synopsys VCS compiler. After generating gate level netlist using Synopsys DC compiler, physical design implementation is carried out with Synopsys IC Compiler. Physical design involves floor planing, power plan, placement, clock tree synthesis and routing.

II. BRIEF OVERVIEW OF RC5 ALGORITHM

RC5 is a symmetric block cipher with data dependent cyclic rotation which provides high degree of nonlinearity. Notation of particular RC5 algorithm is RC5-w/r/b, in this design RC5-32/12/16 is employed which indicates 32-bit words, 12 rounds and 16-byte secret key. The input and output blocks will be 2w=64bits. Algorithm involves three processes- key expansion, encryption and decryption. These routines consist of three primitive operations (and their inverse operation): words addition, bitwise XOR and data-dependent cyclic left / right rotation of x by y denoted as x <<< y / x>>>y

A. Key expansion

The b-byte secret key is stored into an array L [0…c-1] of c=b/u words, where u=w/8 is the number of bytes per word. The two magic constants P_w and Q_w are used to generate an array S [0…t-1] where t=2r+2.

P_w=odd $((e-2)2^W)$

Q_w=odd $((\phi-2)2^W)$

where e= 2.718281828459(base of natural logarithms)

ϕ=1.618033988749(golden ratio)

odd (x) is odd integer nearest to x.

Algorithm for key expansion is given as

$S[0]=P_w$

for i = 1 to t- 1 do

$S[i] = S[i-1] + Q_w$

i = j = 0;

A = B = 0;

do 3 * max(t, c) times:

A = S[i] = (S[i] + A + B) <<< 3;

B = L[j] =(L[j] +A+B) <<< (A + B);

i = (i + 1) mod (t);

j = (j + 1) mod(c);

B. Encryption

We assume that the input block is given in two w-bit registers A and B. The key-expansion should be performed before the encryption, so that the array S[0…t-1] has been computed. The encryption algorithm in pseudo-code is given by,

A = A + S[0]

B = B + S[1]

for i = 1 to r do

A = ((A xor B) <<< B) + S[2* i]

B = ((B xor A) <<< A) + S[2* i + 1]

The output is in the registers A and B.

C. Decryption

The cipher text input is stored in A and B registers. The decryption algorithm is given by

for i = r downto 1 do

B=((B- S[2 * i + 1]) >>> A) xor A

A = ((A- S[2*i]) >>> B) xor B

B = B- S[1]

A = A- S[0]

The registers A and B contain plain text output at the end.

III. HARDWARE IMPLEMENTATION

The verilog HDL is used to implement the proposed design. The design has two modules- one for encryption and another for decryption, which will operate in parallel. Key expansion process is common for both encryption and decryption. The hardware for key expansion is included in both the modules to make the operation faster. Figure.1 shows the architecture for RC5 Crypto processor.

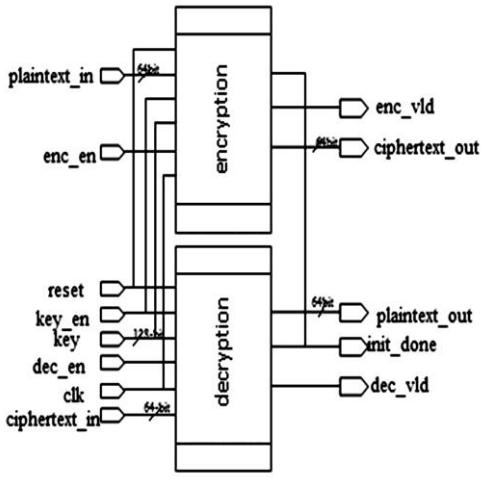

Fig.1: Architecture of RC5 Crypto processor

The key expansion unit converts the secret key bytes to words, it initializes sub key array. The secret key words copied in L-array are mixed with subkey array to generate final S-array S [0…t-1].The S-array values are stored in the RAM and processed with incoming data in encryption and decryption block. The architecture is shown in figure 2 and figure 3.

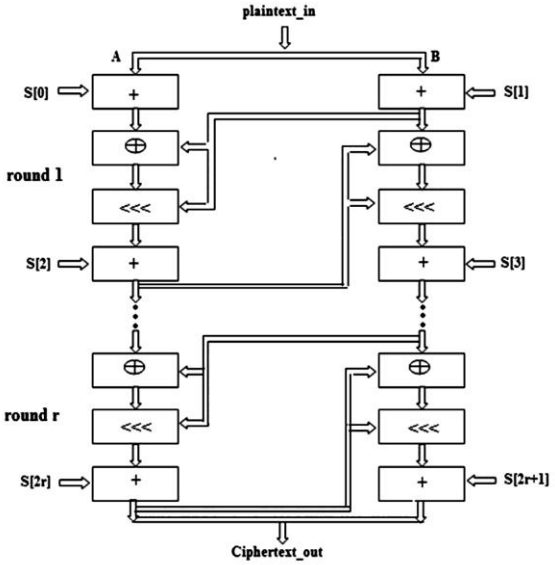

Fig. 2 Architecture for Encryption

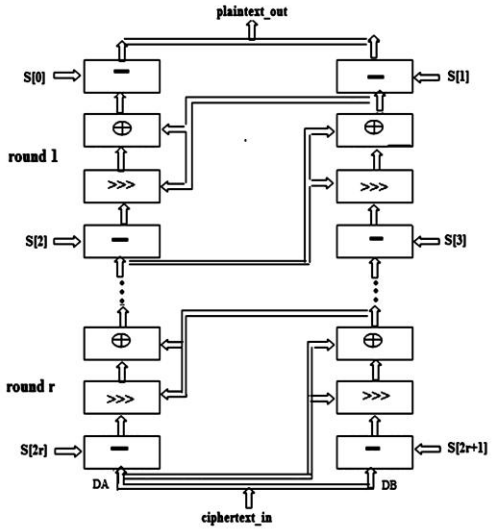

Fig. 3 Architecture for Decryption

IV. PHYSICAL DESIGN IMPLEMENTATION

The RTL code simulated using Synopsys VCS compiler and its functionality is verified. The gate level synthesis is done with Synopsys Design Compiler and netlist is generated. This netlist contains all the information about the cells used, interconnections between the cells, area used, and other details. During the synthesis, constraints are applied to make sure that the design meets the required functionality and speed (specifications). After the netlist is verified for design functionality and timing is sent for the Physical Design flow. Synopsys IC Compiler is used for Physical design. The general IC Compiler flow is shown in figure 4.

Physical design steps are briefly discussed below.

A. Floor planning:

Floor plan determines the size of the design cell (or die), creates the boundary and core area, and it creates line tracks for placement of standard cells. Floor planning identifies structures that should be placed closer and allocates space for them in such an area that it is utilized to the best. Floor planning controls parameters like aspect ratio & core utilization It is defined as follows:

a] Aspect Ratio= Horizontal Routing Resources / Vertical Routing Resources

b] Core Utilization= Standard Cell Area / (Row Area + Channel Area)

For floor planning we should provide logical and physical library. Logical library contains timing and functionality information for all standard cells (and, or, flip flop…etc), timing information for hard macros (IP, ROM, RAM,… etc), maximum fan-out, transition, maximum/minimum capacitance. Physical library contains physical information of standard and macro cells necessary for placement and it defines placement unit tile.

B. Power planning

Each cell must be connected to power and ground along its ends. To protect the chip wiring, the current must be limited below some threshold through any particular wire. Based on our design's speed, toggling activity and layout, power rails must be disseminated across the design so that this limit is not violated.

C. Placement

Placement is the step where all the standard cells of the design are placed in the tracks laid during floor planning. Before the start of placement optimization all Wire Load Models (WLM) are removed. Placement uses RC (resistance/capacitance). Figure 6 shows the layout after the placement stage. Logic is moved closer for shorter nets and cells are upsized for optimal drive strength and speed.

D. Clock Tree Synthesis

The main idea here is balancing of the skew between endpoints. The clock tree is built with the following constraints.

a] Clock Skew: Difference between the clock arrival times.

b] Clock Latency: Max delay between the clock root and clock leaf.

c] Transition Time: Clock buffers are usually bigger in size and have a shorter transition time as well as a more even rise and fall times.

The clock_opt command in ICC builds the clock trees, performs incremental logic and placement optimizations, running the clock tree optimizations, routing the clock nets. Optionally, clock_opt will also fix hold time violations, does inter-clock balancing.

E. Routing

Routing creates physical connections to all clock and signal pins through metal interconnects. Routed paths must meet setup time and hold time, maximum capacitance/ transition, and clock skew requirements
The four steps in routing operations are given by

a] Global routing

b] Track assignment

c] Detail routing

d] search and repair

Global Route assigns nets to specific metal layers and global routing cells. Global route tries to remove congested global cells while minimizing detours. Global route also removes pre-routed P/G nets, placement blockages and routing blockages.

Track Assignment (TA) will assign every net to a specific track and actual metal traces are laid down by it. It also tries to make long, straight traces to overcome the number of vias. DRC is not checked after TA stage. TA operates on the complete design at once.

Detail Routing tries to fix all DRC violations after track assignment using a fixed size small area known as "SBox". Detail route traverses the whole design box by box until entire routing pass is complete.

Search and Repair fixes remaining DRC violations through multiple iterative loops using progressively larger SBox sizes.

IC Compiler does static timing analysis based on the operating conditions specified in setup library by the designer. Once the design has been physically verified, block and chip level sig off are done. The layout is represented in the GDSII stream format (Binary) & is sent to the foundry.

Results

The functionality of the design is verified using Synopsys VCS simulator. Output waveform is shown in figure 5.

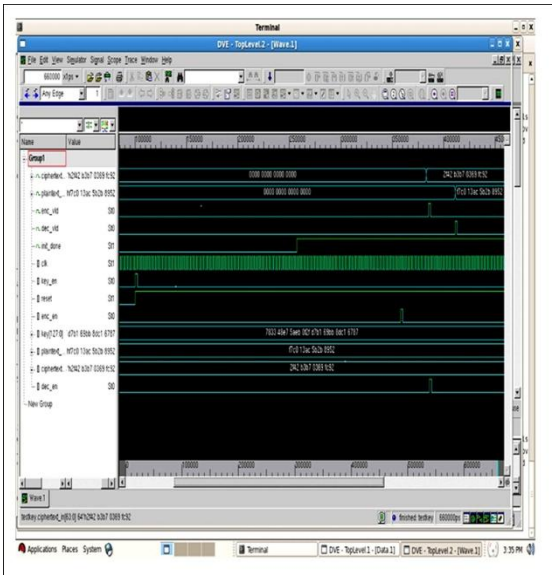

Fig.5: Simulation result of RC5 algorithm.

The layout after physical design is shown in figure. 6

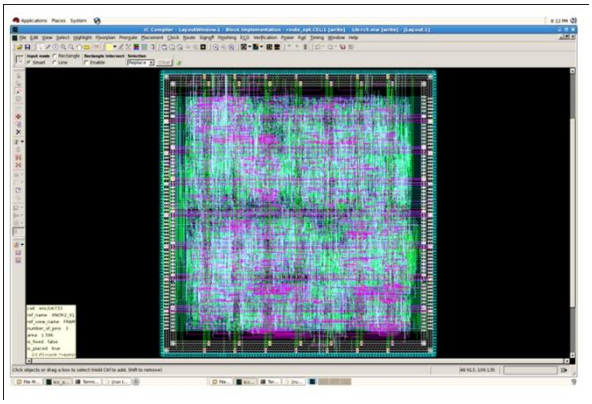

Fig. 6.Post route Layout.

The power consumption and chip area details are given in table I and II respectively. Performance comparison with previous works is given in table III.

Table I. Power measurement

Cell Internal Power	687.7801 uW (71%)
Net Switching Power	284.9116 uW (29%)
Total Dynamic Power	972.6917 uW (100%)
Cell Leakage Power	262.4366 uW

Table II. Area measurement

Area (sq. um)	
Combinational area:	16187.164216
Non combinational area:	12241.053784
Net Interconnect area:	11111.098624
Total cell area:	28428.218000
Total area:	39539.316624

Table III Performance comparison with prior works

works	Process	Max. Frequency (MHz)	Power (mW)	Year
2	FPGA	71	N/A	2003
3	FPGA	50	N/A	2005
4	FPGA	42	138.3	2008
5	180nm	50	5.87	2010
proposed	40nm	100	0.97	2013

VI. CONCLUSION

The architecture for RC5 in this project is power-efficient and area-efficient design. Nonlinearity produced by cyclic data dependent rotation and mixing of key arrays provides good security against the four main attacks. RC5 is a better cipher solution for devices with limited resources. RC5 can be considered as one of the best ciphers in terms of overall performance, when used in system with limited memory and processing capabilities. The leakage power can be further reduced using power gating technique. As an industry standard, RC5 can also be utilized in other types of wireless networks such as WBSN.

REFERENCES

[1] William Stallings' Cryptography and Network Security: Principles and Practice, 5e

[2] N. Sklavos, C. Machs, and O. Koufopavlou, "Area optimized architecture and VLSI implementation of RC5 encryption algorithm," in Proc. 10th IEEE International Conference on Electronics, Circuits and Systems, vol. 1. December 2003.

[3] L. Hua, L. Jianzhou, and Y. Jing, "An efficient and reconfigurable architecture for RC5," Canadian Conference on Electrical and Computer Engineering, May 2005.

[4. O. Elkeelany and S. Nimmagadda, "Effect of loop-unrolling in hardware reconfigurable implementations of RC5-192 encryption algorithm," IEEE Region 5 Conference, April 2008.

[5] Yain-Reu Lin, Chia-Hao Hsu, Student Member, IEEE, R. Rieger, and Chua-Chin Wang, "Low Power RC5 Cipher for ZigBee Portable Biomedical Systems", 2011 IEEE International Conference on Consumer Electronics (ICCE)

[6] K. Hyejung, K. Yongsang, and Y. Hoi-Jun, "A low energy bio sensor node processor for continuous healthcare monitoring system," IEEE Asian Solid-State Circuits Conference,November 2008.

[7] RC5 algorithm: Potential cipher solution for security in wireless body sensor networks (WBSN), International Journal Of Advanced Smart Sensor Network Systems (IJASSN), Vol 2, No.3, July 2012

[8] Sin-Yu Chen, Rung-Bin Lin, Hui-Hsiang Tung, and Kuen-Wey Lin"Power Gating Design for Standard-Cell-Like Structured ASICs" IEEE 2010.

❑❖❑