

# Performance of Cryptographic Instructions in RISC Based Architecture

**D. VARA PRASADA RAO**

HOD-ECE

Turbomachinery Institute of Tech.  
& Sci., A. P. , India.,

**RAJESH VEERANKI**

Assoc. Prof, ECE Dept

Turbomachinery Institute of Tech.  
& Sci., A. P. , India.,

**E.V.KRISHNA KISHOR GOWD**

Asst. Prof, ECE Dept

Turbomachinery Institute of Tech.  
& Sci., A. P. , India.,

**Abstract:** On a general purpose processor it results lower throughput and larger power consumption. In security is one of the most important features in data communication. Cryptographic algorithms are mainly used for this purpose to obtain confidentiality and integrity of data in communication. Implementing a cryptographic algorithm this work we propose processor architecture to perform the cryptographic algorithms and also it speed up the encryption and decryption process of data. This processor will perform the cryptographic operations as like general instructions in GPP. The data size of this processor is 32-bit. The processor architecture is designed using Verilog HDL.

**Keywords:** Cryptographic Algorithms, GPP, Verilog.

## I. INTRODUCTION

There are two basic types of processors design philosophies: reduced instruction set computer (RISC) and complex instruction set computer (CISC). As the name suggests CISC systems use complex instructions. For example adding two integers is considered a simple instruction. But an instruction that copies an element from one array to another and automatically updates both array subscripts is considered a complex instruction. RISC systems use only simple instructions. RISC systems assume that the required operands are in the processors internal registers not in the main memory. A CISC design does not impose such restrictions. RISC designs use hardware to directly execute instructions.

Cryptography plays a significantly important role in the security of data transmission. On one hand with developing computing technology implementation of sophisticated cryptographic algorithms has become feasible. The cryptographic algorithms are classified into public key cryptography and private key cryptography. The private key cryptography which usually has a relatively compact architecture and smaller key size than public key cryptography is often used to encrypt/decrypt sensitive information or documents. Some well known examples of public key cryptographic algorithms are RSA (Rivest-Shamir-Adleman) and elliptic curve crypto systems and private key cryptographic algorithms are AES (Advance Encryption Standard), DES (Data Encryption Standard) and TEA (Tinny Encryption Algorithm). Implementation of these cryptographic algorithms on a general purpose processor is complex and also it has the drawback of lower throughput and higher power consumption.

In the present work the design of a 32-bit data width RISC processor is presented based on cryptographic algorithms. It was designed with

simplicity and efficiency in mind. It has a complete instruction set, Hayward architecture memory, general purpose registers and simple Arithmetical Logic Unit (ALU). Here the ALU design performs the cryptographic operations like operations in AES, Blowfish, IDEA algorithms. To design of RISC architecture we used Verilog HDL.

Present work is divided as follows: Section II presents the Processor architecture with cryptographic operations; section III presents the Cryptographic operations are presented; section IV is dedicated functional blocks and results discussions.

## II. PROCESSOR ARCHITECTURE

The proposed processor has 32-bit data size, that its architecture has been designed in a way to be modular.

The ALU unit that uses a minimal instruction set, emphasizing the instructions used most often and optimizing them for the fastest possible execution. In this architecture the execution time of all instructions with the CPU clock cycle. The proposed architecture will perform both basic arithmetic and logical operations and cryptographic operations like rotate word, Swapping, Fixed coefficient multiplication, matrix multiplication.

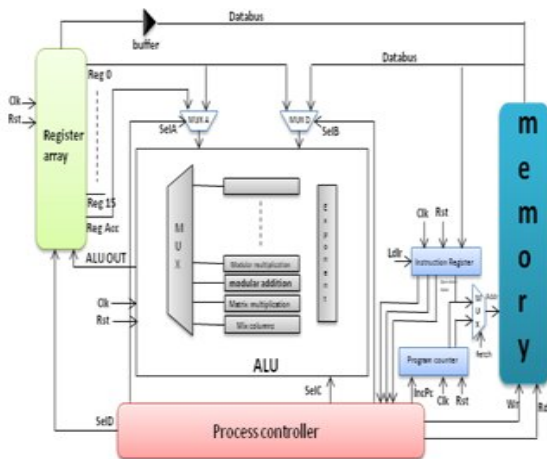


Figure no.1

### III. CRYPTOGRAPHIC OPERATIONS

AES (Advance Encryption Standard) is a block cipher developed in effort to address threatened key size of Data Encryption Standard (DES). It allows the data length of 128 bits and different key lengths 128, 192, 256 bits. The main operations in AES are Shift Rows, Rotate Word, Matrix Multiplication, Mix column.

Blowfish is a symmetric block cipher that encrypts data in 8-byte blocks. The algorithm has two parts;

key expansion and data encryption. Key expansion consists of generating the initial contents of one array namely, eighteen 32-bit sub-keys and four arrays (S-Boxes), each of size 256 by 32 bits from a key of at most 448 bits. The main operations of this algorithm are addition modulo two (XoR) and addition modulo  $2^{32}$ .

IDEA algorithm of the encryption process we provide the original (128 bits) cipher key to the mentioned unit. When the necessary the key generator unit produces different sub-keys by performing circular left shift operation by 25 bits on the current key and provides the sub-keys to other units. The unit named as multiplication modulo  $2^{16}+1$  is used to perform all the multiplication modulo  $2^{16}+1$  operation, when required the same unit is for bit wise Xor.

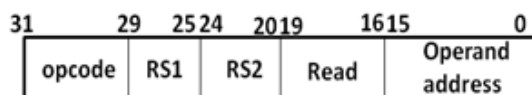
**Instruction Set:** for a complete design it was necessary to create a specific instruction set and its own instruction format. The instructions are classified in to Data manipulation and arithmetic logical operations.

The below table describes the complete instruction set. Each instruction having its own opcode.

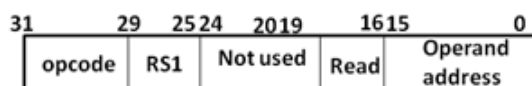
Table no.1

Syntax	Operation	Description
NoP	Nop	No operation
Ld Sr[A]	Sr= Memm[Address]	Move data from memory to register
Addition [A,B]	C=A xor B	GF(2m) addition
ModularMultiplication[A,B]	C=A+Bmod P	GF(2m) modular addition
Modular Multiplication[A,B]	C=A*Bmod P	GF(2m) modular multiplication
MatrixMultiplication[A,B]	Matrix multiplication	Polynomial matrix multiplication
Mix column[A,B]	C=Y*A mod X 4%1	Polynomial mix column transformation
Fixedmultiplier[A,B]	C=(03)*A	Reduction multiplication
AMXModulo [A]	C=A*(2A+1) mod P	Reduction modulo multiplication
Length rotation[A,B]	C=A<<B	Variable length rotation
Rotate word [A]	C=shiftrow(A)	Rotate word
LRShift[A,B]	C=A>>B,C=A<<B	Left, rotate shift operation

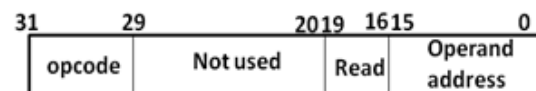
The logical operations like shift left shift right and rotate word which requires only one source register shown in below type.



The operations like addition, modular functions require two source registers and to store result in destination result as shown in below type.



The load instructions and store instructions requires address from different data sources shown in below.



### IV. RESULT DISCUSSION

**Instruction Register:** Instruction registers store the instruction which read from the memory and keep it as an output for the control circuit like operation code, source registers, operand address

and operands these values set to general purpose registers.

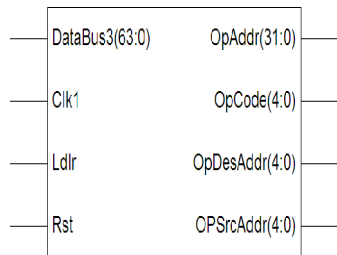


Figure no.2 Block diagram

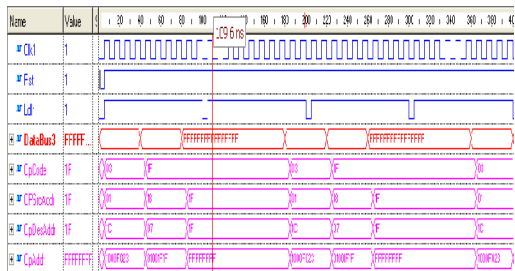


Figure no.3 simulation results

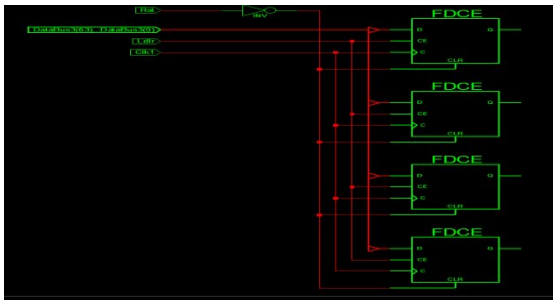


Figure no.4 Technology schematic

Table no.2 implementation results

Logic Utilization	Usage	Availability
Slices	1	768
Flip Flops	47	1536
LUTs	1	1536
IOBs	93	124

**Arithmetic Logical Unit:** The arithmetic logical unit has 16 operations each one of them was created and converted in to a symbol, and then a multiplexer was placed in order to obtain a 4-bit selector.

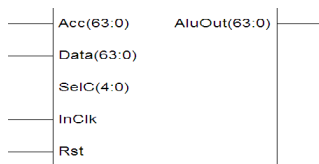


Figure no.6 block diagram

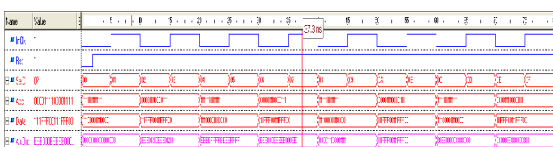


Figure no.7 simulation results

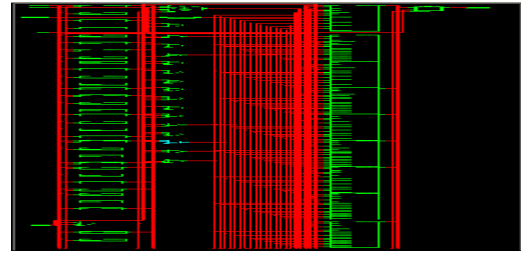


Figure no.8 Technology schematic

Table no.3 implementation results

Logic Utilization	Usage	Availability
Slices	360	768
Flip Flops	64	1536
LUTs	652	1536
IOBs	199	124

**General Purpose Registers:** General purpose registers store and save operands and results during program execution. ALU and memory must be able to write/read those registers so a set of sixteen 32-bit registers were used along with multiplexers and control circuit which are the operands to ALU which perform the operation.

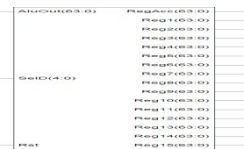


Figure no.10 block diagram



Figure no.11 simulation results

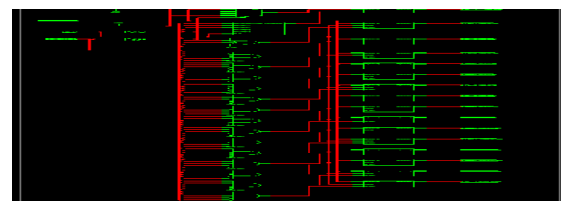


Figure no.12 Technology schematic

Table no.4 implementation results

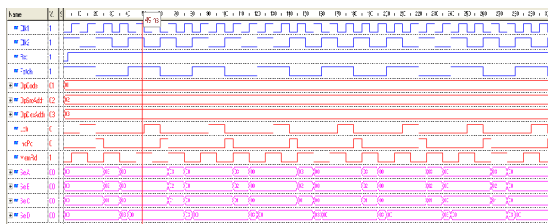
Logic Utilization	Usage	Availability
Slices	48	768
Flip Flops	87	1536
LUTs	1024	1536
IOBs	8	124

**Control Unit:** The control unit is based on using FSM and we designed it in a way that allows each state to run at one clock cycle, the first state is the reset which is initializes the CPU internal registers and variables. The machine goes to the reset state by enabling the reset signal for certain number of clocks. Following the reset state would be the

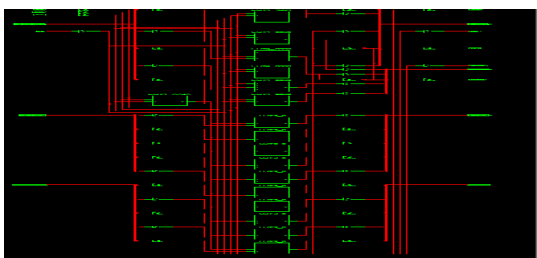
instruction fetching and decoding states which will enable the appropriate signals for reading instruction data from the memory and decoding the parts of the instruction. The decoding state will also select the next state depending on the instruction since every instruction has its own set of states, the control unit will jump to the correct state based on the instruction given.



**Figure no.13 block diagram**



**Figure no.14 simulation results**



**Figure no.15 Technology schematic**

**Table no.5 implementation results**

Logic Utilization	Usage	Availability
Slices	12	768
Flip Flops	44	1536
LUTs	20	1536
IOBs	44	124

**V. CONCLUSION**

32-bit cryptographic processor perform mathematical computations used in symmetric key algorithms has been designed using Verilog HDL the simulations are performed using Active HDL and implementation performed using Xilinx tool.

**REFERENCES**

[1.] Jun-hong chen “A High-Performance Unified Field Reconfigurable Cryptographic Processor”. IEEE-2010  
 [2.] Nima Karimpour Darav “CIARP: Crypto Instruction-aware RISC Processor.IEEE-2012”  
 [3.] Antonio H. Zavala “RISC-Based Architecture for Computer Hardware Instruction” IEEE-2011

[4.] “Data Encryption Standard” 1999 october 25.  
 [5.] “Advance Encryption Standard” November 26 2001