Design of High Performance MIPS Cryptography Processor

Kirat Pal Singh[1,*], Shivani Parmar[2], and Dilip Kumar[3]

[1] Department of Electronics and Communication Engineering,
SSET, Surya World University, Bapror, Rajpura, Punjab, India
[2] Department of Electronics and Communication Engineering,
Sachdeva Engineering College for Girls, Gharuan, Punjab, India
[3] ACS Division, Centre for Development of Advanced Computing,
Mohali, Punjab, India
kirat_addiwal@yahoo.com,
{shivaniparmar03,dilip.k78}@gmail.com

Abstract. This paper presents the design and implementation of low power 32-bit encrypted and decrypted MIPS processor for Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES) based on MIPS pipeline architecture. The organization of pipeline stages has been done in such a way that pipeline can be clocked at high frequency. Encryption and Decryption blocks of DES, TDES and AES cryptography algorithms on MIPS processor and dependency among themselves are explained in detail with the help of architecture. Clock gating technique is used to reduce the power consumption in MIPS crypto processor. This approach results in processor that meets power consumption and performance specification for security applications. Proposed design Implementation concludes higher system performance and reducing gate propagation delay while reducing operating power consumption. The purpose this processor is to find the maximum clock frequency and adjusted setup and hold time based on propagation delay for circuits with combinational and sequential gates. Testing results shows that the MIPS crypto processor operates successfully at a working frequency of DES, TDES & AES crypto processor at 218MHz, 209MHz, & 210MHz and a operating bandwidth of 664Mbits/s, 636Mbits/s, and 560Mbits/s.

Keywords: Cryptography, Delay, Datapath, Throughput, MIPS.

1 Introduction

Security attacks against network are increasing significantly with time. Our communication media should also be secured are confidential. Cryptanalysis is the study used to describe the methods of code-breaking or cracking the code without using the security information, usually used by hackers. For this purpose, these three suggestions arrive in everyone's mind: (i) one can transmit the message secretly, so

* Corresponding author.

K. Singh, A.K. Awasthi, and R. Mishra (Eds.): QSHINE 2013, LNICST 115, pp. 778–793, 2013.

that it can be saved from hackers, (ii) the sender ensures that the message arrives to the desired destination, and (iii) the receiver ensure that the received message is in its original form and coming from the authenticate person. In order to achieve the same one can use the two techniques, (i) one can use invisible link for writing the message through the confidential person, and (ii) use of scientific approach called "Cryptography". Cryptography is the technique used to avoid unauthorized access of data. Data can be encrypted using a cryptographic algorithm by various keys. It will be transmitted in an encrypted state, and later decrypted by the intended party. If a third party intercepts the encrypted data, it will be difficult to decipher. The security of modern cryptosystem is not based on the secrecy of the algorithm, but on the secrecy of a relatively small amount of information, called a secret key. The fundamental and classical task of cryptography is to provide confidentiality by encryption methods. It is used in applications present in technologically advanced societies; it includes the security of ATM cards, computer passwords, and electronic commerce.

An encryption algorithm provides Confidentiality, Authentication, Integrity and Non-repudiation. Confidentiality ensures that the information is accessible to only authorized set of people. Authentication is the act of establishing that the algorithms are genuine. Integrity in the general means completeness but in encryption it is adhering to some mathematical proof. Non-repudiation in cryptology means that it can be verified that the sender and the recipient were, in fact, two parties who claimed to send or receive the message, respectively.

The MIPS is simply known as Millions of instructions per second and is one of the best RISC (Reduced Instruction Set Computer) processor ever designed. MIPS architecture is employed in a wide range of applications. The architecture remains the same for all MIPS based processors while the implementations may differ [1]. There is a 16- bit RSA cryptography MIPS cryptosystem have been previously designed [2]. Some adjustments and minor improvements in the MIPS pipelined architecture design are made using authenticating devices [3] such as Data Encryption Standard [DES], Triple-DES and Advanced Encryption Standard [AES] to protect data transmission over insecure medium. High speed MIPS processor possesses Pipeline architecture to speed up the processing as well as increase the frequency and performance. A MIPS based RISC processor was described in [4]. It consists of basic five stages of pipelining that are Instruction Fetch, Instruction Decode, Instruction Execution, Memory Access and Write Back. These five pipeline stages generate 5 clock cycles processing delay and several Hazards during the operation [2]. These pipelining Hazard are eliminates by inserting NOP (No Operation Performed) instruction which generate some delays for the proper execution of instruction [4]. The pipelining Hazards are of three types: data, structural and control hazard. These hazards are handled in the MIPS processor by the implementation of Forwarding Unit, Pre-fetching or Hazard detection unit, Branch and Jump Prediction Unit [2]. The Forwarding unit is used for preventing data hazards which detects the dependencies and forward the required data from the running instruction to the dependent instructions [5]. Stall occurs in the pipelined architecture when the consecutive instruction uses the same operand as that of the instruction and requires more clock cycles for execution. This reduces the performance. To overcome this situation, Instruction Pre-fetching Unit is used which reduces the Stalls and improves

performance. The control hazard occurs when a branch prediction is mistaken or in general, when the system has no mechanism for handling the control hazards [5]. The control hazard is handled by two mechanisms: Flush mechanism and Delayed jump mechanism. The branch and jump prediction unit uses these two mechanisms for preventing control hazards. The flush mechanism runs instruction after a branch and flushes the pipe after the misprediction [5]. Frequent flushing may increase the clock cycles and reduce performance. In the delayed jump mechanism, Specific numbers of NOP's are pipelined after the Jump instruction to handle the control hazard. The branch and jump prediction unit placement in the pipelining architecture may affect the critical or the longest path. The standard method of increasing performance of the processor is to detect the longest path and design hardware that results in minimum clock period.

2 MIPS Crypto Processor Architecture

The single chip MIPS crypto processor (shown in Fig. 1) consists of various components like Datapath, Data I/O unit, Control Unit, Memory unit, Crypto Specific Unit, Dependency Resolver and Arithmetic Logic Unit. The dedicated data processing block consist of Datapath and Crypto IP core (coprocessor) that performs the128-bit AES cipher operation and a 64-bit DES/TDES cipher or decipher operation. Advanced Encryption Standard (AES) algorithm operates on 128bits block size by using cipher keys with lengths 128, 192 and 256 bits for encryption process respectively. The incoming data and key are stored in a matrix called state matrix and all the operations are performed over the state matrix [6]. Data Encryption Standard (DES) and Triple DES is a Symmetric crypto algorithm, which operates on 64-bit block size with 16 rounds. The input plaintext, cipher keys and output cipher text are of 64-bit. The main operation in DES and TDES is bit permutation and substitution in

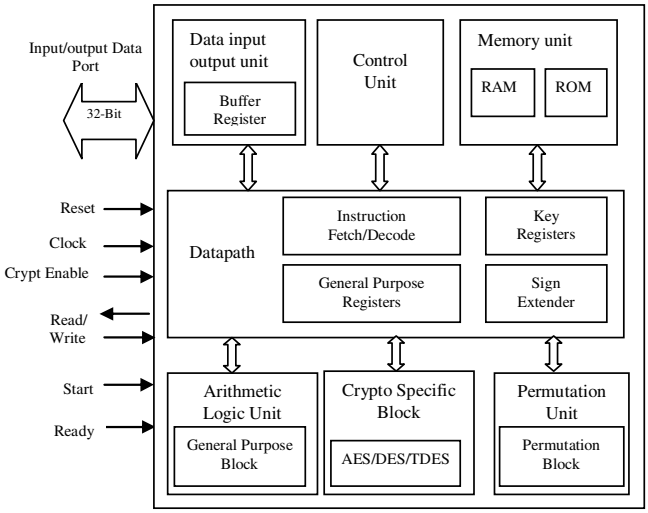


Fig. 1. MIPS crypto processor architecture

one round which is performed by the permutation unit. Datapath processing unit performs the 5 stages pipelining process inside the processor. It consists of Program Counter, 32-bit General Purpose Registers, Key Register and Sign Extender Unit. The program counter unit updates the values available at its input bus at every positive edge clock cycle and also fetches the next instruction from the instruction ROM memory. The registers are read from the General purpose register and the opcode is passed to the control unit which asserts the required control signals. Sign extension is used for calculating the effective address. The data and instruction memory have capability of storing 256 bytes and each byte is referred by the address in between 0 to 256. The address is represented by 8-bits.

The MIPS controller is the main core of the architecture which consists of control unit and ALU control signal unit. The function of controller is to controls the dedicated crypto block and performs the interface and specific operation with the external devices such as Memory, I/O bus interface controller. Single control unit controls the activities of other modules according to the instruction stored inside memory. The crypto specific block executes various other private and public key algorithms such as RSA, DSA, elliptic curve and IDEA with other application programs such as user authentication programs.

The arithmetic logic unit (ALU) performs the NOP (no operation), addition, subtraction, OR, NOR, set less than, shift left logic operation. The data and address calculations for load and store instruction are performed by ALU. The Load and Store instructions write to and read from the RAM memory in the memory unit while the ALU results and the data read from RAM are written in to the register file by the register type and Load instruction respectively. Data I/O has two different external interfaces which stored data initially at buffer registers or move data to output. The bit permutation operation has a big process part in DES and TDES algorithms as it improves diffusion properties. The incoming data is subjected to some bit position according to the permutation type. The dependency resolver block has a function to avoid stall by rearranging the instruction sequence and checking the successive instruction for their stall possibility by comparing their operands. This module handles both stalling as well as data forwarding of previous stage. In case of data dependency between two consecutive instructions the receiving instruction waits for one clock cycle. Thus dependency resolver controls the data forwarding in pipeline stages.

2.1 Data Encryption Standard

Data Encryption Standard (DES) algorithm uses the complicated logical function such as non-linear permutation and substitution. In this algorithm, there are 16 rounds of identical operation and in each round, 48-bit sub keys are generated, and substitution using S-box, bitwise shift, and XOR (exclusive –OR) operation are performed. The algorithm is designed to encrypt and decrypt blocks of data consisting of 64-bit using 56-bit key. Sometimes the key is considered as 64-bits in length for computational purpose (but only 56bits are used for conversion purpose and rest bits are used for parity checking). DES acts on 64-bit block of the plaintext, involving 16 rounds of permutations, swap, and substitutes as shown in Fig. 2. The standard includes labels

describing all of the selection, permutation and expansion operations mentioned below; these aspects of the algorithm are not secrets. The basic DES steps are:

(1) The 64-bit block to be encrypted undergoes an initial permutation (IP), where each bit is moved to a new bit position; e.g., the 1st, 2nd and 3rd bits are moved to the 58th, 50th and 42nd position, respectively.
(2) The 64-bit permuted input is divided into two 32-bit blocks, called left and right, respectively. The initial values of the left and right blocks are denoted L0 and R0.
(3) These are then 16 rounds of operation on L and R blocking. During each iteration (where n ranges from 1 to 16).

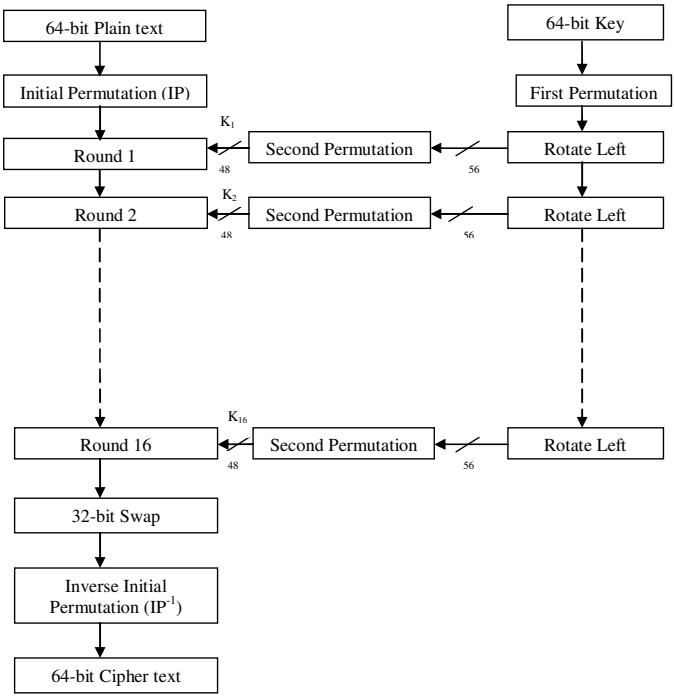


Fig. 2. DES Algorithm

At any given step in the process, the new L block value is merely taken from the prior R block value. The new R block is calculated by taking the bit-by-bit exclusive-OR (XOR) of the prior L block with the results of applying the DES cipher function f, to the prior L block and K_n. (K_n is a 48 bit value derived from the 64 bits DES key. Each round uses a different 48 bits according to the standards key schedule algorithm).

The cipher function, f, combines the 32-bit R block value and the 48-bit sub key in the following way. First, the 32-bits in the R-block and expanded to 48 bits by an expansion function (E); the extra 16 bits are found by repeating the bits in 16 predefined positions. The 48-bit expanded R block is then XORed with the 48-bit value that is then divided into eight 6-bit blocks.

There are fed as input into 8 sections (S) boxes, denoted S1,…, S8. Each 6bit input yields a 4-bit output using a lookup table (LUT) based on the 64 possible inputs; this results in a 32-bit output from the S-box. The 32-bits are then arranged by a permutation function (P), producing the results from the cipher function.

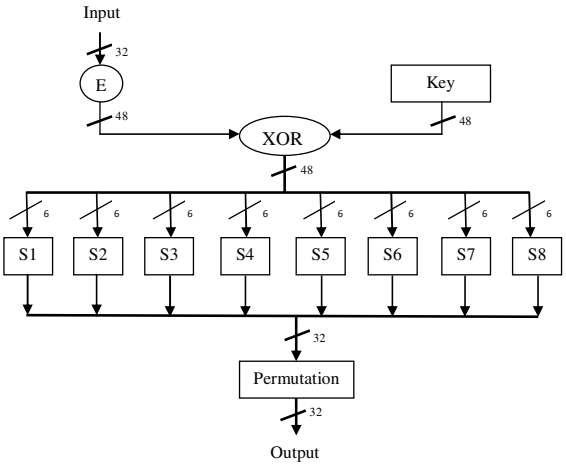


Fig. 3. DES Round (F[R,K] box) Detail

(4) The results from the final DES round- i.e., L16 and R16 are recombined into a 64-bit value and fed into an inverse initial permutation (IP^{-1}). At this step, the bits are rearranged into their original positions, so that 58^{th}, 50^{th}, and 42^{nd} bits, for example, one moved back into the 1^{st}, 2^{nd} and 3^{rd} positions, respectively. The output from IP^{-1} is the 64 bit cipher text block.

2.2 Triple Data Encryption Standard (TDES)

A DES algorithm is no longer considered to be a secure algorithm for many applications by the NIST (National Institute of Standard and Technology). A more secure algorithm based on DES is called Triple Data Encryption Algorithm (triple DES, 3DES, or TDEA) which is still supported by NIST. Fig. 4 shows the Triple Data Encryption Algorithm. This involves applying DES, then DES^{-1}, followed by a final DES to the plain text using three different options [7]. This results in a cipher text that is much harder to break. TDEA uses the same set of operations as DES.

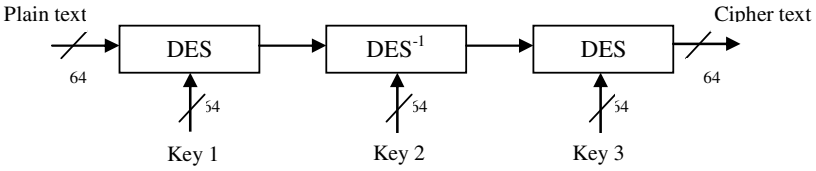


Fig. 4. TDES Block Representation

2.3 Advanced Encryption Standard (AES)

There are numerous encryption algorithms that are now commonly used in computation, but U.S government has adopted the Advanced Encryption Standard (AES) to be used by Federal departments, and agencies for protecting sensitive information. The AES algorithm is a symmetric cipher and used a single secret key for both the encryption and decryption. In addition, the AES algorithm is a block cipher as it operates on fixed-length groups of bits (blocks), whereas in stream ciphers, the plaintext bits are encrypted one at a time, and the set of transformation applied to successive bits may very during the encryption process. The AES algorithm operates on block length [N_b] of 128-bits, by using cipher keys with key length [N_k] of 128, 192 or 256 bits or the encryption process.

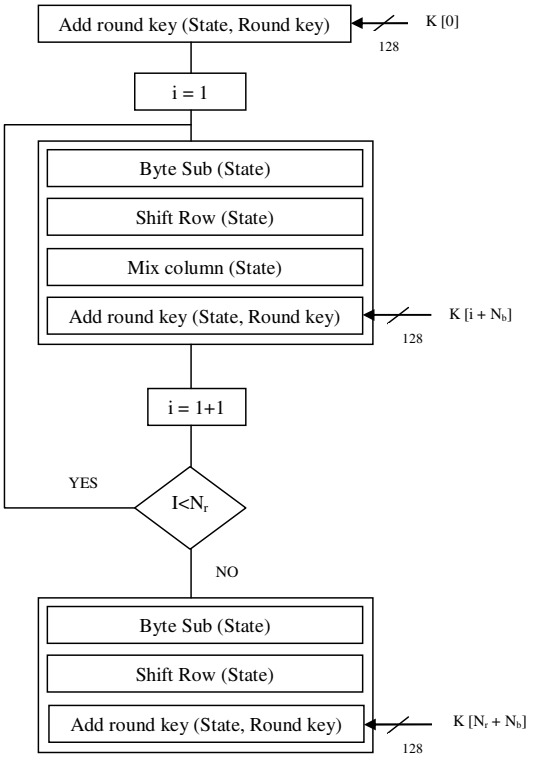


Fig. 5. AES Block Diagram for Key Length of 128bits and the number of Iterations required are 10(Nr = 10)

The encryption and decryption process of AES block consists of number of different transformation applied consecutively over the data block bits, considered as a 4x4 array of 8 bit bytes (also called "state" in the algorithm). The state undergoes four different transformations in each round having fixed number of iterations. These transformations are *"Sub Byte", "Shift Row", "Mix Column",* and *"Add Round Key"* transformations. *"Sub Byte"* can be implemented by non–linear substitution of bytes that operates independently on each byte of the state using a substitution LUT

(S-box). In this S-box; each byte in the state matrix is an element of a Galois Field GF (2^8), with irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x+1$. In simple terms, GF ($2^n$) is a set of 2^n elements each represented by an n-bit string of 0's and 1's and affine transformation is applied (over GF (2)). The *"Shift Row"* can be implemented using a cyclically shift the rows of the state over different offsets. *"Mix Column"* are considered as most complicated operation in the algorithm and need GF (2^8) fields and multiply by modulo x^4+1 with a fixed polynomial $a(x)=\{03\}x^3 + \{01\}x^2 + \{02\}x$. *"Add Round Key"* is added to the state by a logical XOR operation. Each round key consists of N_b words from the key expansion. These N_b words are added into the state columns. Each round key is a 4-word (128bit) array generated as a product of previous round key, and a sense of substitution LUT for each 32-bit word of the key. The key expansion generated a total of $N_b(N_r+1)$.

3 Design and Implementation Methodology

Current applications demand high speed processor for large amount of data transmission in real time. As compared to software alternatives, hardware implementation provides highly secure algorithms and fast solutions approaches for high performance applications. Software approaches could be a good choice but it has some limitations like low performance and speed. Main advantages of software are low cost and short time to market. But they are unacceptable in terms of high speed and performance specification. So that, Hardware alternatives could be selected for implementing MIPS crypto processor architecture.

Table 1. Hardware v/s software alternatives for crypto processor

Parameters	Software	Hardware	
		FPGA	ASIC
Performance	Low	Medium-High	Very High
Power consumption	Depends	Very high	Low
Logic integration	Low	Low	High
Tool cost	Low	Low	Low
Test development complexity	Very low	Very low	High
Density	High	Very low	High
Design efforts	Low-medium	Low-medium	High
Time consumed	Short	Short	High
Size	Small-medium	Small	Large
Memory	Fine	Fine	Fine
Flexibility	High	High	-
Time to market	Short	Short	High
Run time configuration	-	high	-

Hardware implementation supports both Field Programmable Gate Arrays (FPGAs) and Application Specific Integrated Circuits (ASIC) at high data rates. Such design has high performance but more time consuming and expensive as compared to software alternatives. The detailed comparison of hardware v/s software solutions for implementing the MIPS crypto processor architecture is shown in Table 1. Based on

the comparison, hardware solution is a better choice in most of the cases because they have high performance. The main advantage of FPGA in hardware alternative, FPGA are low density and low area consumption. Logic integration, size and density are the major drawbacks in ASIC but have higher performance than FPGA.

3.1 Hardware Implementation of Cryptographic Engine

The global architecture of encrypted and decrypted MIPS pipeline processor is modified in a way that it executes encrypted instruction. Fig. 6 shows the block diagram of encrypted MIPS processor. To modify MIPS processor for encryption, we insert the cryptography module such as Data Encryption Standard (DES), Triple Data Encryption Standard (T-DES), Advanced Encryption Standard (AES) etc. to the pipeline stage. Only single cryptographic module is used in same hardware implementation. The instruction fetch unit of encrypted MIPS contains Program Counter (PC), Instruction Memory, Decryption core and MUX. The Instruction memory reads address from the PC and stores instruction value at the particular address that is pointed by the PC. Instruction Memory sends encrypted instruction to MUX and decryption core. The decryption core gives decrypted instructions which are further sent to the MUX. The output of MUX is fed to the IF register. The MUX control signal comes from control unit. The instruction decode unit contains Register file and Key register. Key register stores the key data of encryption/decryption core. Key address and Key data comes from write back stage. The key data to be stored into the register file and remains same for all program instruction execution. The control unit provides various control signals to other stages. This acts as select line for two multiplexers. When the control unit detects a store/branch/jump it asserts the control signal high and keep it asserted till a load instruction is detected. During that period,

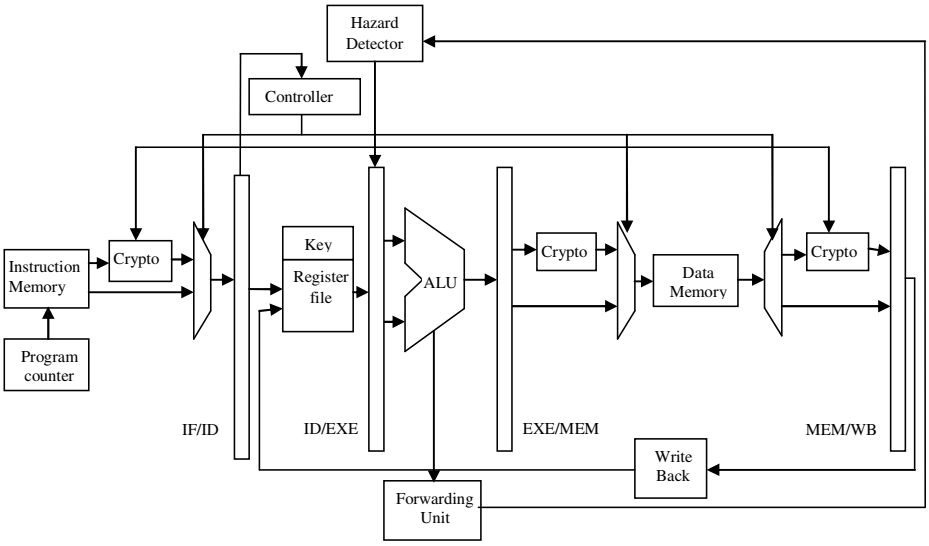


Fig. 6. Detailed MIPS crypto processor architecture

the write back stage gets the forwarded data and the memory stage gets a constant zero value thus preventing only further transitions. When the control signal is de-asserted, then the data pass through the standard pipeline structure. The execute unit executes the register file output data and performs the particular operation determined by the ALU. The ALU output data is sent to EXE register which temporarily store address value. The Memory Access Unit contains Encryption core, Decryption core, Data Memory, MUX and DEMUX. The second register data from register file is fed to the encryption core and the MUX. Here the crypt signal enable/disable encryption operation when occurs. The read/write signal of data memory describes whether reading/writing operation is done. Output of data memory pass through DEMUX whose one output goes to decryption core and other to MEM register. Here the unencrypted memory data and decrypted data are temporarily stored to the MEM register. The MEM output is fed to the write back data MUX and according to the control signal, the output of MUX goes to register file.

3.2 Initialization

The operational mode of the MIPS crypto processor is controlled by a RESET signal. When the RESET signal is at logic “0”, the crypto processor is in the reset mode and the processing unit writes the memory and register contents using the 32-bit bidirectional data bus, 10-bit address bus, and four control signals. When the reset signal is at logic “1”, the crypto processor is in the running mode and acts as an FPGA, implementing one of the three algorithms based on the preloaded contents of the memory blocks. The keys are kept in the key registers of the register file of crypto processor that are available to other stages of processor.

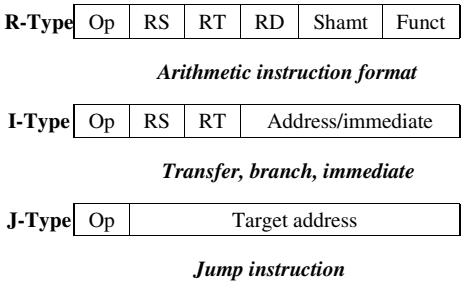
3.3 Microinstruction Set

The MIPS instruction set is straightforward like any other RISC designs. MIPS are a load/store architecture, which means that only load and store instructions access memory. Other instructions can only operate on values in the registers [8]. Generally, the MIPS instructions can be broken into three classes: the memory-reference instructions, the arithmetic- logical instructions, and the branch instructions. Also, there are three different instructions formats (as shown in Fig.7) in MIPS architecture: R-Type instructions, I-Type instructions, and J-Type instructions. A subset of the instruction has been implemented in our design, the list of which is given in Table 2.

Instruction Type	Instruction
R-Type	AND, OR, NOR, ADD, SUB, SLT
I-Type	ADDI, SUBI, NORI, ANDI, SLTI,SLL, SRL
	LW,SW, LKLW, LKUW
	BEQ, BNE
J-Type	J, JR, JAL, CRYPT

Fig. 7. Implemented MIPS Instruction Types

Table 2. MIPS Instruction Format



Field	Description
Op[31-26]	6-bit operation code
RS[25-21]	5-bit source register
RT[20-16]	5-bit target register
Immediate[15-0]	16-bit immediate address
Target[25-0]	26-bit jump target
RD[15-11]	5-bit destination register
Shamt[10-6]	5-bit shift amount
Funct[5-0]	6-bit function field

The MIPS instruction Format is used to minimize the number of bits in each instruction, note that the 6-bit operation code field in the instruction format is used to have the word length of the memory as 32-bit and used standard memory blocks for the program memory. Only 32-bit instruction set is required for the current implementation as shown in Table 3. There are three more new instructions that support encrypted and decrypted operation. These instructions are load key upper word (LKUW), load key lower word (LKLW) and encryption mode (CRYPT). These instructions randomly use opcodes in the hardware implementation. LKLW and LKUW come under I-type instruction and variant of load word (LW). These two instructions do not need to specify a destination address in the assembly code. CRYPT instruction comes under J-type instruction and instead of address, only single argument i.e., Boolean value is assigned. This indicates enable/disable encryption and decryption process. Any non zero value enables the encryption/decryption process and zero value disables the encryption process.

4 Implementation Results

The complete pipeline processor stages are modeled in VHDL. The syntax of the RTL design is checked using Xilinx tool. For functional verification of the design the MIPS processor is modeled in Hardware Descriptive Language. The design is verified both at the block level and top level. The complete design along with all timing constraints, area utilization and optimization options are described using Synthesis Report. The design has been synthesized targeting 40nm triple oxide process technology using Xilinx FPGA Virtex-6 (xc6vlx240t-3ff1156) device. The Virtex family is the latest and fastest FPGA which aims to provide up to 15% lower dynamic and static power and 15% improved performance than the previous generation [18]. It is obvious that there is a trade-off between maximum clock frequency and area utilization (number of slices LUT's) because the basic programmable part of FPGA is the slice that contains four LUTs (look up table) and eight Flip flops. Some of the slice can use their LUT's as distributed RAM.

Table 3. ISA overview of Implemented MIPS Crypto Processor

Instruction	Operation	Syntax	Opcode	Clock cycle
ADD	Arithmetic Addition operation	*Add $rd, $rs, $rt*	0(0x000000)	4
SUB	Arithmetic Subtraction operation	*Sub $rd, $rs, $rt*	0(0x000000)	4
AND	Logical AND operation	*AND $rd, $rs, $rt*	0(0x000000)	4
OR	Logical OR operation	*OR $rd, $rs, $rt*	0(0x000000)	4
NOR	Logical NOR operation	*NOR $rd, $rs, $rt*	0(0x000000)	4
SLT	Set less than manipulation operation	*SLT $rd, $rs, $rt*	0(0x000000)	4
ADDI	Immediate arithmetic addition operation	*ADDI $rd, $rs, constant*	8(0x001000)	4
SUBI	Immediate Arithmetic Subtraction operation	*SUBI $rd, $rs, constant*	8(0x001000)	4
SLTI	Immediate Set less than manipulation operation	*SLTI $rd, $rs, constant*	8(0x001000)	4
ORI	Immediate Logical OR operation	*ORI $rd, $rs, constant*	8(0x001000)	4
ANDI	Immediate Logical AND operation	*ANDI $rd, $rs, constant*	8(0x001000)	4
NORI	Immediate Logical NOR operation	*NORI $rd, $rs, constant*	8(0x001000)	4
SLL	Shift left logic operation	*SLL $rd, $rs, shamt*	0(0x000000)	4
SRL	Shift right logic operation	*SRL $rd, $rs, shamt*	0(0x000000)	4
BEQ	Branch equal operation	*Beq $rd, $rs, label*	4(0x000100)	3
BNE	Branch not equal operation	*Bne $rd, $rs, label*	4(0x000100)	3
JR	Conditionally jump to register	*Jr $rd*	2(0x000010)	3
JAL	Unconditionally jump to program	*Jal $rd*	2(0x000010)	3
J	Conditionally jump to program	*J $rd*	2(0x000010)	3
CRYPT	Encryption/decryption enable	*Crypt $rd*	65(0x111111)	3
LW	Load data word to CPU	*Lw $rd, offset($rs)*	35(0x100011)	5
SW	Store data to memory	*sw $rd, offset($rs)*	43(0x101011)	4
LKUW	Load key upper word to target register	*LKUW $rd, offset($rs)*	64(0x111110)	5
LKLW	Load key load word to target register	*LKLW $rd, offset($rs)*	60(0x111100)	5

4.1 Design Performance, Area and Power Requirement

The performance of MIPS crypto processor based on three different crypto modules such as DES, TDES, and AES algorithms. For DES and TDES, 16 clock cycles are used for DES/TDES crypto specific block to execute data, 20 clock cycles are needed to execute the R-type instruction, 21 clock cycle are needed for I-type instruction and 19 clock cycle for J-type instruction data.

The power consumption is estimated by the Xilinx XPOWER Analyser tool, using the post layout netlist of the crypto processor along with the node activity data for each algorithm. The power consumption can be further reduced by running the processor at lower voltages than the normal voltage of 1.5v (as long as the speed and throughput requirements are satisfied). Power analysis was done for the portion between the EXE/MEM and MEM/WB stage. This is performed for both the encryption and decryption process. Clock gating technique is used to minimize energy reduction during pipeline stall stages. This technique identifies low processing

requirement periods and reduces operating voltage with clock frequency (voltage-frequency scaling), resulting in reduced average operating power consumption. This may or may not occur frequently depending upon compiler efficiency. The power analysis result is carried out on the same clock frequency. in our design, a symbol is processed every clock cycle, the throughput is calculated on the basis of number of instruction execution per second. The formula for calculating throughput is:

$$\text{Throughput} = f * \text{symbol width/total clock frequency}$$

Where f is the operation frequency and symbol width is one of our parameterized values. Table 4 and Table 5 show the performance throughput, area and the estimated power consumption of DES and TDES MIPS crypto processor. Maximum throughput of MIPS DES based crypto processor is of 664Mbits/s at 4.58ns and for TDES based crypto processor is 636Mbits/s at 4.78ns.

Table 4. Throughput estimates for the MIPS crypto processor based on DES

Features	Processor
Crypto processor	DES
Data length	64-bits
Speed	218MHz (clock rate)
Throughput	664 Mbits/s (Data Bandwidth)
Area	66072 Slice LUT's(look up tables)
Latency	21 clock cycles(both for encryption and decryption)
Power consumption	1.746W(quiescent-1.303 and dynamic-0.444)

Table 5. Throughput estimates for the MIPS crypto processor based on TDES

Features	Processor
Crypto processor	TDES
Data length	64-bits
Speed	209MHz(clock rate)
Throughput	636 Mbits/s (Data Bandwidth)
Area	64673 Slice LUT's(look up tables)
Latency	21 clock cycles(both for encryption and decryption)
Power consumption	1.981W(quiescent-1.131 and dynamic-0.851)

Table 6. Throughput estimates for the MIPS crypto processor based on AES

Features	Processor
Crypto processor	AES
Data length	128-bits
Speed	210MHz (clock rate)
Throughput	560Mbits/s (Data Bandwidth)
Area	109738 Slice LUT's(look up tables)
Latency	48 clock cycles(for encryption)
Power consumption	1.313W(quiescent-1.008 and dynamic-0.396)

Table 7. Some Recent Cryptography Algorithm Implementation Specification

	Algorithm	HW/SW	ASIC/FPGA	Clock (MHz)	Area/LUTs	Throughput	Power	Technology
[12] 2006	DES AES	HW	ASIC	13.56	2.25mm^2	3.5Mbps 1.83Mbps (Enc), 0.85 (Dec)	15.9mW 16.3mW @1.8V	Synopsys DC (0.18µm TSMC Library)
[14] 2006	AES	HW	FPGA	100	4584LEs (Logic Element)	297Mbits/s	-	Altera Stratrix
[13] 2007	DES TDES AES	HW	FPGA	400 400 400	- - -	732Mbits/s 244Mbits/s 731Mbits/s	- - -	Xilinx Spartan 3 (90nm)
[1] 2009	Normal MIPS	HW	FPGA	205.7	1890 (4-Input LUT's)	-	1.139W	Xilinx Spartan 3E (90nm)
[3] 2010	Normal MIPS	HW	FPGA	140.39	133893(4-Input LUT's)	-	-	Xilinx Virtex-6 (40nm)
[5] 2011	Normal MIPS	HW	ASIC	50.76	200215 µm^2	-	475.78mW	Synopsys DC (130nm TSMC Library)
[19] 2011	AES	HW	ASIC	333	3.78mm^2 3.25mm^2	10.656Gbps	4.1mW 3.9mW	Synopsys DC (180nm TSMC Liberary)
Present work	Normal MIPS	HW	FPGA	181.29	34040 (Slice LUTs)	1160.3Mbits/s	1.293W	Xilinx Virtex-6 (40nm)
	DES			218	66072(Slice LUTs)	664Mbit/s	1.746W	
	TDES			209	64673(Slice LUTs)	636Mbits/s	1.981W	
	AES			210	109738(Slice LUTs)	560Mbits/s	1.313W @1.2V	

In case of AES crypto processor, 43 clock cycles are used for crypto specific block to execute data, 47 clock cycles are needed to execute the R-type instruction, 48 clock cycles are needed for I-type instruction and 46 clock cycles for J-type instruction data. Table 6 shows the performance throughput; area and the estimated power consumption of AES based MIPS crypto processor. Maximum throughput of AES based MIPS Crypto processor is 560Mbits/s at 4.76ns. Moreover, it is possible to trade performance with area and power in the implementation. For example, higher performance can be obtained by running processor at higher frequency up to 300MHz for the current design (increasing power consumption) and/or using pipeline (increasing area) for more performance demanding applications. Some recent cryptography algorithms specification are shown in table 7.

5 Conclusion

In this paper, we have presented the design of high performance 32-bit cryptography MIPS processor that executes encrypted/decrypted instructions. Initially it read encrypted data from instruction memory and decrypts the same data and sent it to the next pipeline stages. The processor uses the symmetric block viz., DES, TDES and AES plain/cipher that can process data length of 64 bits, 64bits and 128bits respectively. The crypto algorithm block inside the MIPS processor performs data encryption and decryption algorithm. The design has been modeled in VHDL and synthesize using Xilinx tool and functional verification and optimization policies are adopted for it. Optimization and synthesis of design is carried out at latest and fastest FPGA Viretx-6 device that improves performance. Each program instructions are tested with some of vectors provided by MIPS. We conclude that the performance of MIPS crypto processor using DES and AES is High 664Mbits/s and 560Mbits/s respectively.

References

[1] Gautham, P., Parthasarathy, R., Balasubramanian, K.: Low-power pipelined MIPS processor design. In: International Symposium on Integrated Circuit (ISIC 2009), pp. 462–465 (2009)

[2] Zulkifli, Yudhanto, Soetharyo, Adinono: Reduced Stall MIPS architecture using Pre-fetching accelerator. In: IEEE International Conference on Electrical Engineering and Informatics, pp. 611–616. IEEE (August 2009) ISBN: 978-1-4244-4913-2

[3] Ghewari, P.B., Patil, J.K., Chougule, A.B.: Efficient hardware design and implementation of AES cryptosystem. International Journal of Engineering Science and Technology 2(3), 213–219 (2010)

[4] Patterson, D.A., Hennessy, J.L.: Computer Organization and Design, The hardware/Software Interface. Morgan Kaufmann (2005)

[5] Lotfi, P., Salehpour, A.-A., Rahmani, A.-M., Afzali-kusha, A., Navabi, Z.: Dynamic power reduction of stalls in pipelined architecture processors. International Journal of Design, Analysis and Tools for Circuits and Systems 1(1), 9–15 (2011)

[6] Sever, R., Neslin Ismailoglu, A., Tekmen, Y.C., Askar, M.: A high speed ASIC Implementation of the rijndael Algorithm. In: IEEE International Symposium on Circuits and Systems (2004)

[7] Advanced Encryption Standard (AES). Fed. Inf. Process. Syandards Pub. (November 2001)
[8] Balpande, R.S., Keote, R.S.: Design of FPGA based Instruction fetch & decode Module of 32-bit RISC (MIPS) processor. In: International Conference on communication Systems and Network Technologies, pp. 409–413. IEEE (2011) ISBN: 978-0-7695-4437-3
[9] Taherkhani, S., Ever, E., Gemikonakli, O.: Implementation of Non-pipelined and pipelined data encryption standard (DES) using Xilinx Virtex-6 technology. In: 10th IEEE International Conference on Computer and Information Technology (CIT 2010), pp. 1257–1262 (2010)
[10] Navabi, Z.: VHDL: Modular design and synthesis of cores and systems, pp. 283–291. McGrew-Hills (2007) ISBN: 978-0-07-147545-7
[11] Floyd, L.: Digital Fundamental with VHDL, pp. 362–368. Pearson Education (2003) ISBN: 0-13-099527-4
[12] Eslami, Y., Sheikholeslami, A., Glenn Gulak, P., Masui, S., Mukaida, K.: An Area-Efficient Universal Cryptography Processor for Smart Cards. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 14(1), 43–56 (2006), doi:10.1109/TVLSI.2005.863188, ISBN: 1063-8210
[13] Askar, M., Egemen, T.: Design and SystemC Implementation of a Crypto Processor for AES and DES Algorithms. In: Information Security & Cryptology Conference with International Participation (ISC Turkey). Bildiriler kitabi Proceedings, pp. 145–149 (December 2007)
[14] Hani, M.K., Wen, H.Y., Paniandi, A.: Design and Implementation of a Private and Public key Crypto Processor for Next-Generation IT Security Applications. Malaysian Journal of Computer Science 19(1), 29–45 (2006)
[15] Sklavos, N.: On the Hardware Implementation Cost of Crypto-Processors Architectures. Information Security Journal: A Global Perspective, 53–60 (June 2010), doi:10.1080/19393551003649016, ISSN: 1939-3555 print/ 1939-3547
[16] Xilinx, ISE Simulator, http://www.xilinx.com/tools/isim.htm
[17] Xilinx, XST Synthesis, http://www.xilinx.com/tools/xst.htm
[18] Xilinx, ISE In-Depth tutorial, pp. 95–120 (June 2009), http://www.xilinx.com/support/documentation/sw_manuals/xilinx11/ise11tut.pdf
[19] Saravanan, P., RenukaDevi, N., Swathi, G.: A High-Throughput ASIC implementation of Configurable Advanced Encryption Standard (AES) Processor. International Journal of Computer Applications Special Issue on “Network Security and Cryptography” 3, 1–6 (2011)