SINGLE CORE HARDWARE MODULE TO IMPLEMENT ENCRYPTION IN TECB MODE

M. B. I. Reaz[1], M. I. Ibrahimy[1], F. Mohd-Yasin[2], C. S. Wei[2], M. Kamada[3]

[1]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[2]Faculty of Engineering, Multimedia University, Selangor, Malaysia
[3]Department of Computer and Information Sciences, Ibaraki University, Hitachi, Japan

Key words: Encryption, DES, 3DES, FPGA, Synthesis, Hardware

Abstract: The growth of the Internet as a vehicle for secure communication has resulted in Data Encryption Standard (DES) no longer capable of providing high-level security for data protection. Triple Data Encryption Standard (3DES) is a symmetric block cipher with 192 bits key proposed to further enhance DES. Many applications crave for the speed of a hardware encryption implementation while trying to preserve the flexibility and low cost of a software implementation. This project used single core module to implement encryption in Triple DES Electronic Code Book (TECB) mode, which was modeled using hardware description language VHDL. The architecture was mapped in Altera EPF10K100EFC484-1 and EP20K200EFC672-1X for performance investigations and resulted in achieving encryption rate of 102.56 Mbps, area utilization of 2111 logic cells (25%) and a higher maximum operating frequency of 78.59 MHz by implementing on the larger FPGA device EP20K200EFC672-1X. It also suggested that 3DES hardware was 2.4 times faster than its software counterpart.

Elektronski modul za izvedbo šifriranja v TECB načinu

Kjučne besede: šifriranje, DES, 3DES, FPGA, sinteza, strojna oprema

Izvleček: Porast zahtev po uporabi varnih internetnih storitev je privedel do spoznanja, da DES standard (Data Encryption Standard) ne omogoča več zelo visoke zaščite podatkov. Predlagani trojni DES (3DES), ki je simetrična šifra s 192-bitnim ključem, naj bi dodatno izboljšal DES. Izvedba 3DES standarda v strojni opremi omogoča visoke hitrosti šifriranja in poskuša obdržati fleksibilnost in nizko ceno programskih rešitev. V delu opišemo uporabo elektronskega modula za izvedbo 3DES TECB (3DES Electronic Code Block), ki smo ga modelirali z uporabo VHDL jezika. Arhitekturo smo preslikali v Alterini FPGA vezji in dosegli šifrirne hitrosti 102.56 Mbps, izkoristek površine 2111 logičnih celic (25%), in višjo delovno frekvenco 78.59MHz pri uporabi večjega vezja EP20K200EFC672-1X. Ocenili smo, da je elektronska izvedba 3DES do 2.4-krat hitrejša od programske rešitve.

1. Introduction

In the wake of advancement in computer technology and increasingly volatile information flow, we are faced with challenges of safeguarding information that is not meant for public knowledge /1/. It is common to see all sorts of electronic inventions such as the cellular phone, various devices in the military system and smart cards today /2/.

The growth of the Internet has contributed to the increase in the amount of data transferred daily across regions. These data transmissions may contain funds amounting to millions of dollars or government records. However, these applications require high data security. The ease in obtaining and duplicating these data through resourceful parties /e.g. hackers/ has resulted in a decline in confidence amongst Internet users towards online transaction. As such, it is essential to ensure the privacy and authenticity of these data. One of the existing methods that can be used to guard the security and authenticity of data through the Internet is through cryptography.

Data Encryption Standard, DES has been the world wide standard for more than 20 years /3/. DES is used in IPSec protocols, secure socket layer (SSL) protocol and ATM cell encryption. During those years, bundles of software and hardware had been developed to implement this algorithm. However due to the need of higher security, 3DES had been chosen based on its close relationship to DES /4/. Triple DES is an improved version of DES and provides better security compared to DES. This is due to its longer key length and more rounds of DES encryptions. DES only has an effective key length of 56 bits, which is insufficient to resist any brute force attack today /3/. Research has shown that a key-breaking machine that costs less than $1 million can find a key in an average of 3.5 hours and the cost is estimated to drop by a factor of 5 every 10 years /3/. Even though 3DES is three times slower than DES, if used properly, it can be as strong as the 2304-bit public key algorithm because it has longer key length. With an increase in its security standards and compatibility to the DES software and hardware, 3DES is clearly a better choice compared to other algorithms such as RSA and ECC /5/.

Due to its symmetric nature, 3DES is a better choice in encrypting bulk data and is therefore less expensive /1, 6/.

3DES uses only 128 or 196 bits symmetric keys and has simpler algorithm. It is less complicated, less computationally intensive and does not introduce much overhead. Thus, it requires relatively inexpensive hardware /1, 3/. 3DES is faster than RSA. Due to its much longer key length, RSA causes high-level resource utilization and is not suitable to be used in mobile or wireless devices as these devices have underpowered processors /6/.

In comparison to AES, 3DES is faster /7/. The limitation of AES exists because the cipher and its inverse use different codes and/or tables. As such, it does not have bidirectional architecture for encryption and decryption as that of the 3DES. The inverse cipher can only partially reuse the circuitry that implements the cipher, resulting in a larger hardware presumably.

Hardware realization of cryptography has the advantages of being more secured and faster in speed /8/. It gives a higher performance as desired /9/. Even though software implementation of cryptography uses general-purpose processors that offer enough power to satisfy the needs of individuals, hardware realization is the only way to achieve speeds that is more significant than the general-purpose microprocessor /10/. This feature is important for commercial and communication purposes as this is shown in /9/ that security related processing can consume up to 95% of a server's processing capacity. By using a dedicated hardware to run encryption application, more computing can be done within a stipulated period due to parallel processing. Field Programmable Gate Array (FPGA) offers a potential alternative to speed up the hardware realization. From the perspective of computer-aided design, FPGA comes with the merits of lower cost, higher density, and shorter design cycle. The programmability and simplicity of FPGA made it favorable for prototyping digital system.

In this paper, the framework of FPGA-based hardware realization of cryptography using 3DES is proposed. With this approach, both the speed and performance are preserved without the need to trade-off between these two important criteria in encryption and decryption. In this method, VHDL (Very High Speed Integrated Circuit Hardware Description Language) is selected as the hardware description language to realize the system.

2. Triple DES Algorithm

Triple DES encrypts a block of 64-bit data using two or three unrelated 64 bits keys /5/. The internal operation done on these data is similar to that of DES where the only difference is that DES consists of 16 iterations whereas 3DES consists of 48 iterations. In other word, 3DES contains three successive DES operations. Out of the 64-bit key used in DES, the effective key size is only 56 bits. The eighth bit in each byte is used for odd parity checking and is thus ignored. As such the total effective key size for 3DES is 168 bits.

A DES encryption operation is divided into two stages involving the key and the data. In the first stage, 16 subkeys are created from the key whereas the encryption of data message is occurred in the second stage.

During the first stage, permutation is initially done on the 64-bit key and resulting in a 56-bit permutated key. After key permutation, this 56-bit permutated key is divided into left and right halves C_0 and D_0, where each half has 28 bits. With C_0 and D_0 defined, sixteen blocks C_n and D_n, where n = 1, 2, 3,..., 16 are formed by left shifting C_{n-1} and D_{n-1} (once or twice). C_n and D_n are then concatenated to form a 56-bit data, C_nD_n. This 56-bit data is then permutated and resulted in 48-bit subkeys formed. After 16 iterations, 16 sets of subkeys are created. These subkeys are used for data encryption during the second stage.

During the second stage, permutation is done on the 64-bit message data block, M. As this is the first permutation process being done on the data, it is called Initial Permutation. The permutated data are then divided into left half L_0 and right half R_0, each having 32 bits. It is followed by 16 iterations of operations, using function f, which operates on two blocks: data block of 32 bits and subkey block of 48 bits to produce an output block of 32 bits.

$$L_n = R_{n-1} \quad (1)$$

$$R_n = L_{n-1} + f(R_{n-1}, K_n) \text{ where } n = 1, 2, 3,..., 16 \quad (2)$$

As shown in (1) and (2), during each of the 16 iterations, the right 32 bits of the previous iteration, R_{n-1} is used as the left 32 bits of the current iteration, L_n. The right 32 bits in the current iteration, R_n is obtained by implementing XOR to the left 32 bits of the previous step with f function.

To calculate f function, each block R_{n-1} is expanded from 32 bits to 48 bits and the expanded R_{n-1}, $E(R_{n-1})$ is then XORed with the block of subkey K_n, i.e.,

$$K_n + E(R_{n-1}) = B_1B_2B_3B_4B_5B_6B_7B_8 \quad (3)$$

where n = 1, 2, 3,..., 16 and B_i is a group of 6 bits. This results in a 48-bit block, which is then divided into $B_1B_2B_3B_4B_5B_6B_7B_8$. Each B_i gives an address in a different S box, S_i. The 4-bit blocks for the entire eight S boxes are combined to form a 32 -bit block.

Function f is obtained by implementing permutation on the group output such as,

$$f = P(S_1(B_1)S_2(B_2)... S_8(B_8)) \quad (4)$$

At the end of the sixteenth iteration, the order of the two blocks $L_{16}R_{16}$ is reversed to $R_{16}L_{16}$ before applying the permutation on the reversed block. This is the last permutation to be done on the data, thus being called the Final Permutation.

Decryption in DES uses the same process as the encryption operation. The only difference lies in the order in which

the subkeys are used. In the decryption process, the subkeys are used in reverse order, meaning that K_{16} is applied first with K_1 being applied last.

Triple DES shows a high level of similarity in operation to that of DES. Encryption and decryption in 3DES are done by compounding the operation of DES encryption $E_k(I)$ and decryption $D_k(I)$ operations. Encryption operation in 3DES is defined by,

$$Encryption = E_{K3}(D_{K2}(E_{K1}(I))) \quad (5)$$

whereas the decryption operation is defined by,

$$Decryption = D_{K1}(E_{K2}(D_{K3}(I))) \quad (6)$$

From equation (5), it shows that the plaintext is first encrypted by $K1$ using DES. The encrypted data is then decrypted by $K2$ before being encrypted by $K3$. In contrast to that, equation (6) indicates that the 3DES cipher text is initially decrypted by $K3$ using DES, whereby the result is then being encrypted by $K2$. The plaintext is recovered by decrypting the output from second DES operation by $K1$.

Final permutation is actually the inverse operation of initial permutation. As such in a 3DES operation, the initial permutation of the second DES round cancels the final permutation of the first DES round. This is the same in the third DES round where its initial permutation cancels off the final permutation of the second DES round, leaving only an initial permutation and a final permutation during the whole 3DES operation.

3. Design Flow of 3DES Single Core Module

The specification of the 3DES core is set prior to the start of the design process. Different 3DES operation mode could result in different design complexity and different level of security. As such, a trade off between these two conditions must be taken into consideration during the design stage. As to avoid complicated design, 3DES Electronic Code Book (TECB) had been chosen as the mode of operation in this project. This resulted in reduced area utilization and compromised security level in the core design.

Due to varying number of bits being shifted during the different iteration rounds, normal shift register could not be used. A counter had to be added in the design so as to determine the current iteration round. The input signals to the shifting module were shifted appropriately depending on the output of the counter. The output of counter must be passed correctly to the shifting module. Error in connections such as MSB of the counter output being connected to the LSB of the shifting module input could result in error in bit shifting.

During DES encryption operation, the subkeys were transmitted in the sequence of 1 to 16 whereas during DES decryption operation, the subkeys were transmitted in the sequence of 16 to 1. The initial design in mind was to have a multiplexer and a demultiplexer. The 16 subkeys were multiplexed. This was then followed by demultiplexing these multiplexed subkeys either in the sequence of 1 to 16 or from 16 to 1. The sequence in which the subkeys were sent out was determined by the select signal of the demultiplexer. However, this design was difficult to implement as complicated control signals were needed to obtain the 16 subkeys in the correct sequence. These subkeys had to be stored in registers before being demultiplexed.

Based on this, the design of the full implementation of 3DES encryption engine was produced as shown in Figure 1. The multiplexers and demultiplexers in Figure 1 played the role of realising the 16 iterations in a DES operation and the 48 iterations in the 3DES operation.

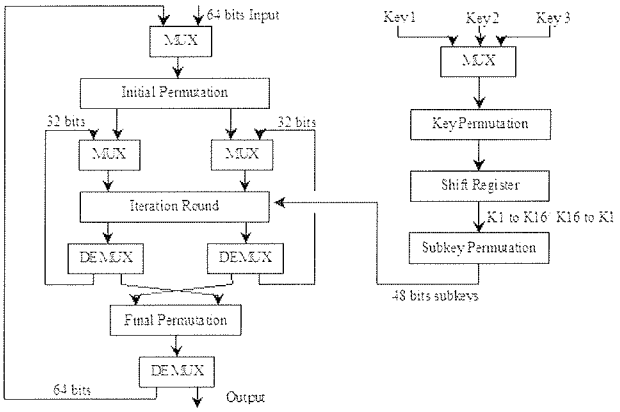

Fig. 1: Full implementation of 3DES

4. Simulation and Synthesis

4.1 Functional Simulation

Functional simulation was done to verify that the design behaved as expected on the VHDL coding using MAX PLUS II software. Since the delay of the combinational logic and wires were not known, the signal suffered only a constant signal change delay of 0.1ns. This delay must be taken into consideration. As the design operated on positive clock edge, this delay could cause the response of the circuit to be delayed by 1 clock cycle. As such, the processes of the other modules were also delayed 1 clock cycle so as to synchronize the operation of the whole design.

The validation of the 3DES operation was done by referring to /11/, where the data message was encrypted based on DES. Validation of the above design after 16 iterations showed the same result as that of DES in /11/. To further verify the design, a simulation done for the whole 3DES operation showed that the encrypted input data could be decrypted to recover the original data message. The two aforementioned verification methods indicated that this design implemented 3DES correctly. The timing diagrams obtained from the simulation that was done during the verification process are shown in Figure 2 and Figure 3 for encryption and decryption respectively.

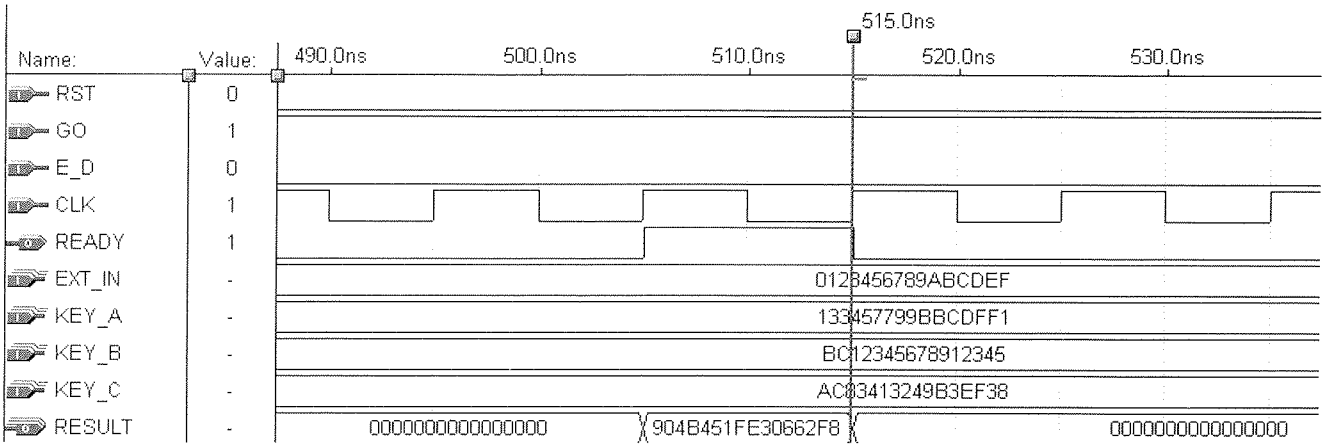

Fig. 2: Functional simulation of encryption operation

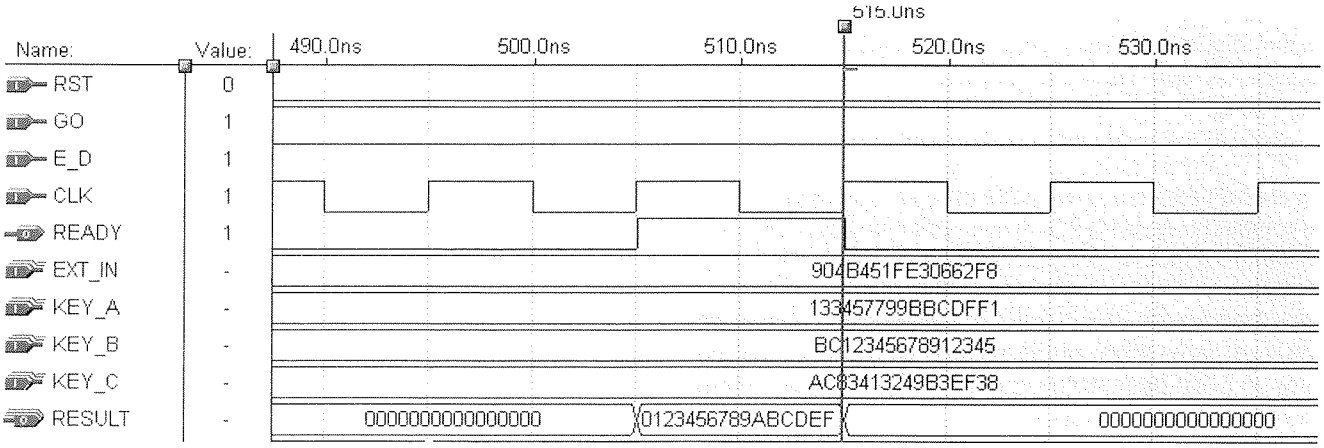

Fig. 3: Function simulation of decryption operation

From Figure 2, the keys used in the encryption of data message 0123456789ABCDEF were 133457799BBCDFF1, BC12345678912345 and AC83413249B3EF38. The cipher resulted text was 904B451FE30662FB. During the decryption operation, the cipher text was decrypted to obtain the original data message of 0123456789ABCDEF as shown in Figure 3.

4.2 Synthesis and Optimization

With the functional simulation showing the correct behavioural result, synthesis was done using Altera Quartus II 4.0 software on the core design implemented into FPGA. Device family that could fit the design into it was chosen and the timing requirements were set.

Different FPGA could result in different maximum frequency obtained. A larger FPGA device family such as APEX20KE gave a higher maximum clock frequency than FLEX10KE. This was an important criterion to be considered while deciding on the FPGA to be used even though the design could be fitted into both. The smaller device family had a higher resource utilization percentage. By deciding to use a larger device family, speed optimization had been given priority in view of excessive amount of resource in the FPGA selected.

There was a trade off between area and speed. A higher number of logic elements used that resulted in higher maximum operating frequency. Initial synthesis of the design on APEX20KE family gave a maximum frequency of 72.77 MHz. However, after switching off the 'Remove Duplicate Registers' and 'Remove Duplicate Logic' setting, the maximum operating frequency achieved approximately 77MHz. The logic cells used that summed up to be 25%, which was 2% more than the previous setting. By setting the maximum frequency requirement to 80MHz, a higher value of 78.59 MHz was achieved. This was the highest maximum clock frequency value with APEX20KE family that could be obtained from the optimization process.

4.3 Timing Simulation

Timing simulation was performed to verify that the module functioned correctly and there were no timing violations in the implemented design. The functional simulation was done using MAX PLUS II software but the timing simulation was done using Quartus II 4.0 software.

During timing simulation, the total delay of the wires and combinational logic was taken into account. Initial testing using the clock signal having frequency that was higher than that of maximum operating frequency resulted in er-

roneous output. The result obtained was not the encrypted data message. This was because the encrypted data cannot be decrypted to recover the initial data message. The total delay had exceeded one clock cycle period.

The clock signal period was then set to 13ns. This clocking period was larger than the total wire and combinational logic delay. Different sets of keys and input data blocks were used during the simulation. It was found that the encrypted data could be decrypted to recover the original data. Besides that, the reset pin had also been tested. Reset signal was set to 'high' to reset the design.

4.4 Synthesis Results

Table 1: Synthesis results

Family	APEX20KE
Device	EP20K200EFC672-1X
Name	Core
Total logic elements	2111 / 8320 (25%)
Total I/O pins	325 / 376 (86%)
Total memory bits	2048 / 106496 (1%)
Total PLLs	0 / 2 (0%)
Total combinational functions	2110
Total registers	408
Performance, f_{max}	78.59 MHz
Clock period	12.724 ns

Table 1 shows the synthesis results of the 3DES encryption engine. The FPGA family that had been selected for the realization of 3DES encryption engine was APEX20KE (more precisely, EP20K200EFC672-1X).

Out of the 8320 logic elements contained in the device, a total of 2111 logic cells were used. A total of 325 I/O pins were utilized, which is equivalent to 86 percent of the total pins in the device. Out of these 325 pins, 65 pins were output pins while the remaining 260 pins were input pins. Out of a total of 106496 memory bits in the device, 2048 of them were utilized. This is equivalent to 1 percent of the total memory bits resource. Besides that, the total number of registers used in the EP20K200EFC672-1X device summed up to be 408. A maximum clock frequency of 78.59 MHz was obtained. The clock signal that was used in the device must have a period of at least 12.724 ns. Any period below this value gave a faulty result.

4.5 Timing and Area Analysis

The results for timing and area analysis of the main modules are presented in terms of maximum operating frequency and logic cell (LC). The analysis was done using Quartus II software. The devices chosen for the implementation were EP20K200EFC672-1X of APEX20KE family and EPF10K100EFC484-1 of FLEX10KE family. Comparison was done between the two devices.

Tables 2 and 3 show the effect of registers and logic cells duplication in EP20K200EFC672-1X and EPF10K100EFC484-1 respectively when the full 3DES architecture was mapped into them. To implement these features, the 'Remove Duplicate Registers' and 'Remove Duplicate Logic' settings were selected or deselected.

When the 'Remove Duplicate Registers' and 'Remove Duplicate Logic' settings were selected during the hardware implementation of the encryption module in EP20K200EFC672-1X, this resulted in lower area utilization of 1984 logic cells and lower maximum operating frequency of 72.77 MHz. When these settings were deselected, higher area utilization of 2111 logic cells and higher maximum operating frequency of 78.59 MHz was obtained.

However, that is not the case when EPF10K100EFC484-1 was used. Selecting the 'Remove Duplicate Registers' and 'Remove Duplicate Logic' setting resulted in lower area utilization but higher maximum operating frequency.

Table 4 shows the synthesis results for the final design of the project. Two devices were used, namely EP20K200EFC672-1X and EPF10K100EFC484-1. EP20K200EFC672-1X is a larger device compared to EPF10K100EFC484-1.

Table 2: Effect of registers and logic cells duplication in EP20K200EFC672-1X

'Remove Duplicate Registers' and 'Remove Duplicate Logic'	Area (LC)	Clock Period (ns)	Maximum Operating Frequency (MHz)
On	1984 / 8320 (23%)	13.742	72.77
Off	2111 / 8320 (25%)	12.724	78.59

Table 3: Effect of registers and logic cells duplication in EPF10K100EFC484-1

'Remove Duplicate Registers' and 'Remove Duplicate Logic'	Area (LC)	Clock Period (ns)	Maximum Operating Frequency (MHz)
On	1924 / 4992 (38%)	17.5	57.14
Off	2080 / 4992 (42%)	17.6	56.82

Table 4: Synthesis results

Device	Area (LC)	Clock Period (ns)	Maximum Operating Frequency (MHz)
EP20K200EFC672-1X	2111/ 8320 (25%)	12.724	78.59
EPF10K100EFC484-1	1924 / 4992 (38%)	17.5	57.14

When the larger device was used, it was found that the final design had a higher maximum operating frequency of 78.59 MHz. It utilized more logic cells. However, when the smaller device from FLEX10KE family was used, it only had a maximum operating frequency of 57.14 MHz. Besides that, the design used only 1924 logic cells of the resource, which was lesser than the 2111 logic cells used in EP20K200EFC672-1X.

With this, it can be concluded that the mapping of the design architecture on different devices can result in different maximum operating frequency and area utilization. A larger device results in higher maximum operating frequency and larger area utilization. As such, considerable decision must be taken on whether a faster operation is needed or a smaller device is required.

Figure 4 demonstrates the RTL view of the core entity. It is shown that core was formed by three smaller entities, namely s_mac, key_block and inp_block. Each of these entities had its own unique function. S_mac controlled and synchronized the operations of the other two entities while key_block processed the three keys, producing the subkeys needed before sending them to inp_block. Inp_block was the entity where the actual encryption and decryption of the plaintext occurred.

Table 5: Comparisons between hardware and software implementation

	3DES (FPGA)	3DES (software)
Key size (bits)	192	192
Data rate (Mbps)	102.56	42.9

Table 5 shows the comparisons done on the performances of the hardware and software implementation of 3DES. Triple DES was implemented into FPGA and as well as MATLAB using an Intel Pentium III 866 MHz machine. It shows that 3DES hardware was significantly (2.4 times) faster than its software counterpart. The 3DES software could only manage a data rate of 42.9 Mbps compared to 102.56 Mbps of 3DES hardware.

5. Conclusions

The hardware implementation of 3DES encryption engine on FPGA chip was realized. The chip selected was EP20K200EFC672-1X of APEX20KE family. It could encrypt data at a rate of 102.56 Mbps, with a maximum operating frequency of 78.59 MHz and area utilization of 2111 logic cells.

The throughput of 102.56 Mbps in the current full implementation of 3DES core can be considered as low by industry standard. As such, to improve the throughput of the design, pipelining of the iterations process can be implemented. Registers can be added to store data during the pipelining process. This will invariably reduce the maximum clock frequency; however the number of clock cycles being used for one complete 3DES operation can be greatly reduced, thus reducing the latency.

To allow more secured encryption process, additional 3DES operation modes can be added to the core module. Currently, the encryption hardware only operates under TECB mode. By including more modes of operation, users can choose to operate under certain mode, depending on their preference.

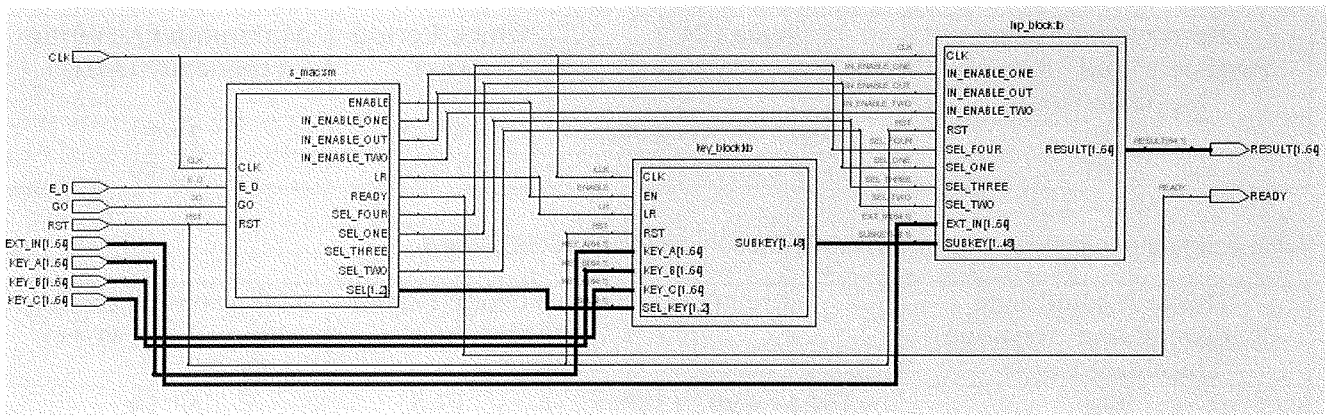

Fig. 4: RTL view of core entity

References

/1/ Aladdin Knowledge System, "The enduring Value of Symmetric Encryption", White Paper, pp: 5-8, August 2000.

/2/ Harper, S. and Athanas P., "A Security Policy Based Upon Hardware Encryption", System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference, pp: 190 - 197, Virginia, 5-8 Jan. 2004.

/3/ Davor Runje, Mario Kovac, "Universal Strong Encryption FPGA Core Implementation", Design, Automation and Test in Europe, 1998, Proc., pp: 923-924, France, 23-26 Feb 1998.

/4/ O.Y.H.Cheung, P.H.W.Leong, "Implementation of an FPGA Based Accelerator for Virtual Private Networks", Proceedings of IEEE International Conference on Field-Programmable Technology (ICFPT), pp: 34-41, Hong Kong, 2002.

/5/ "Data Encryption Standard", Federal Information Processing Standards (FIPS) Publication 46-7, National Institute of Standards and Technology (NIST), USA, 1999.

/6/ Young Sae Kim, Woo Seok Kang and Jun Rim Choi, "Implementation of 1024-bit Modular Processor for RSA Cryptosystem", AP-ASIC 2000, Proceedings of the Second IEEE Asia Pacific Conference, pp: 187-190, Korea, 28-30 Aug. 2000.

/7/ C. Sanchez-Avilla & R. Sanchez-Reillo, "The Rijndael Block Cipher (AES Proposal): A Comparison with DES", Security Technology, 2001 IEEE 35th International Carnahan Conference, pp: 229-234, London, 16-19 Oct 2001.

/8/ Raghuram, S.S. and Chakrabarti, C, "A Programmable Processor for Cryptography", Proceedings. The 2000 IEEE International Symposium on Circuits and Systems, Volume: 5, pp: 985-688, Geneva Switzerland, 28-31 May 2000.

/9/ Lisa Wu, Chris Weaver, and Todd Austin, "Crypto Maniac: A Fast Flexible Architecture for Secure Communication", Computer Architecture, 2001. Proceeding. 28th Annual International Symposium, pp: 110-119, Goteborg, Sweden, 30 June-4 July 2001.

/10/ Pawel R. Chodowiec, "Comparison of the Hardware Performance of the AES Candidates Using Reconfigurable Hardware", Master Thesis, 150 pages, 2002, George Mason University.

/11/ J. Orlin Grabbe, www.aci.net/kalliste/des.htm, 5 Jan 2004.

M. B. I. Reaz[1], M. I. Ibrahimy[1],
F. Mohd-Yasin[2], C. S. Wei[2], M. Kamada[3]
[1]Department of Electrical and Computer Engineering,
International Islamic University Malaysia,
53100 Kuala Lumpur, Malaysia
[2]Faculty of Engineering, Multimedia University,
63100 Cyberjaya, Selangor, Malaysia
[3]Department of Computer and Information Sciences,
Ibaraki University, Hitachi, Ibaraki 316-8511, Japan
ibrahimy@iiu.edu.my

Prispelo (Arrived): 17.04.2007 *Sprejeto (Accepted): 15.09.2007*