Design and Implementation of DES IP Based on LEON3 SOPC

Jing-Jiao Li, Chao-Qun Rong, Dan-Yang Peng, and Dong An
Institute of Electronic Science and Technology, Northeastern University
Shenyang, China
qqrcqq@163.com

Abstract—**This paper proposes a novel method to design and implement DES algorithm IP based on LEON3 SOPC platform. Since this DES IP core is a standard AMBA APB slave device, it can be easily embedded to SoC designs where AMBA bus is used as the interconnect interface, making it much more effective to implement DES algorithm in SoC designs. So comparing with common hardware implementation of DES algorithm, this DES IP core has a very large application prospects in SoC designs. Also the method this paper presents to design an AMBA APB slave device in LEON3 architecture can be referred to. The DES IP is simulated by Modelsim and tested within the LEON3 SOPC platform. Results indicate that this DES IP core has a fine performance and the method this paper demonstrates to design and implement an APB slave device is reliable and referable.**

Keywords— **DES; IP; LEON3; AMBA; SoC; SOPC**

I. INTRODUCTION

DES (Data Encryption Standard) [1], which has been widely used in many fields owing to its excellent performance, is one of the most popular encryption algorithms. DES algorithm has been widely used in satellite communications, gateway server and other important facilities [2]. The main way to implement DES algorithm usually comes with either software or hardware. Considering hardware implementation has more physical security and it is more effective than software implementation, several work about hardware implementation has been done. Xie Shuangjian proposed a way to build DES algorithm using Hardware Description language [3]. Also Ji Yao put out a hardware design with the dynamic key management based on the conventional DES [4].

However, such work just implement DES algorithm as a hardware module which doesn't own too much portability and compatibility. Users or system designers are required to know its operations and timing characteristics quite well before applying the module in their own designs. Meanwhile, because the module is not designed as an IP (Intellectual Property) core, it is not very suitable for SoC (System on Chip) designs since IP is an inevitable choice for SoC designs [5]. Taking those problems in to account, this paper proposes a novel method to design and implement a DES algorithm IP core based on LEON3 SOPC platform. DES algorithm is firstly implemented using VHDL (Very-High-Speed Integrated Circuit Hardware Description Language). After that the DES IP core is built abiding by the protocol and timing characteristics of AMBA APB interface. The DES IP core is thereafter embedded into the LEON3 SOPC platform and finally it is tested to encrypt and decrypt a 400×240 24-bit RGB image. Results indicate this IP core functions properly. Since this DES IP core is a standard AMBA APB slave device, it can be easily embedded to SoC designs where AMBA bus is used as the interconnect interface. Thus, it has much more application prospects in the SoC designs than common hardware implementation of DES algorithm. At the same time the method this paper proposes to design and implements an APB slave device is reliable and referable.

II. SOC AND LEON3 PROCESSOR

SoC is a rapidly growing field in VLSI (Very Large Scale Integrated circuits) design [6]. As an important type of SoC, SOPC (System on a Programmable Chip) is more flexible and appropriate for small batch production by combining the advantages of SoC and PLD (Programmable Logic Device) [7]. SOPC is tailorable, scalable, reprogrammable, and upgradeable with low cost. Thus, it becomes a good solution for high speed system application.

The LEON3 processor is a synthesizable VHDL model of a 32-bit processor compliant with the SPARC V8 architecture [8]. LEON3 processor is of high performance, low complexity and low power consumption. It is more important that all the source code of LEON3 processor and other IP cores belonged to GRLIB is free to use for research and development under the GNU GPL license. LEON3 processor is particularly suitable for SoC designs.

Due to its market dominance, fine documentation and open source, AMBA2.0 AHB/APB bus has been selected as the standard interconnect interface with additional "sideband" signals for automatic address decoding, interrupt steering and device identification in the LEON3 SOPC platform[9]. SoC system built by the open source LEON3 processor and AMBA bus owns a high openness. Designers can add IP cores into the SoC systems flexibly.

III. LEON3 SOPC PLATFORM

The hardware-level architecture of LEON3 SOPC platform is shown in Fig. 1. LEON3 processor is the core of this platform, and a 64MBytes SDRAM (Synchronous Dynamic random access memory) and 8MBytes Flash works as the memory devices. The dashed box is the DES IP core this paper will design and implement. Some other peripherals such as Timer, UART (Universal Asynchronous Receiver/Transmitter) and GPIO (General Purpose Input Output) are also integrated in this platform for their common use. The platform runs on Altera DE2-70 development board.

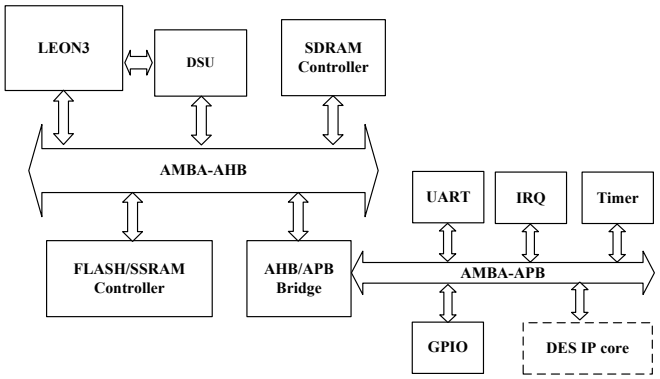


Figure 1. Architecture of LEON3 SOPC Platform

IV. DESIGN OF THE DES IP

Fig.2 shows the block graph of the DES IP core. It is an AMBA APB slave device, mainly consisting of following three components: DES logic module, Register control unit and APB interface. DES logic module completes the logic task of DES algorithm using VHDL hardware description language. Register control unit accomplishes the address mapping, making AMBA APB address space map to the desired registers correspondingly. In this way CPU can control the logic module by using these registers. APB interface finishes the work to support AMBA APB protocol involving writing and reading bus data, timing characteristics and clock supplying.

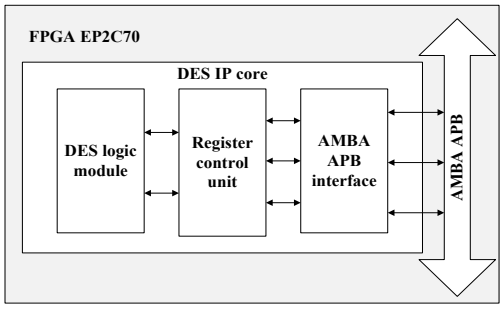


Figure 2. Block graph of DES IP

A. DES logic module

DES operates on 64-bit blocks of plaintext utilizing a 64-bit key. The original 64-bit plaint text is converted with the initial permutation and 16 times of iterative computation with 16 different 48-bit sub-keys. Eventually with inverse permutation, 64-bit ciphertext can be gotten. The hardware implement of DES algorithm module on FPGA is shown in Fig.3. This module mainly consists of sub-keys generating unit, P-Boxes generating units and the top module of this logic module. Signals and interface pins are defined as TABLE I declare.

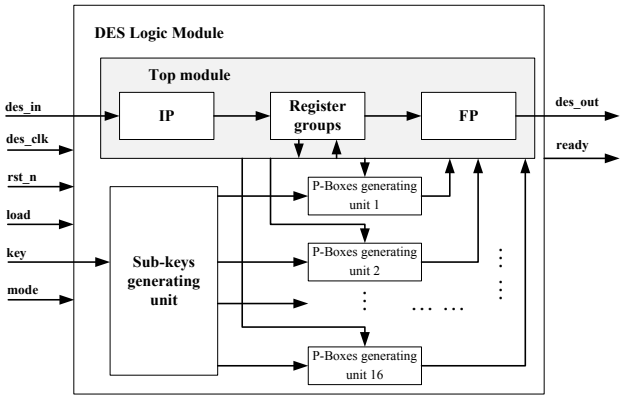


Figure 3. Block graph of the Des logic module.

TABLE I. SIGNALS AND INTERFACE PINS DESCRIPTION

Signal	*Function*
des_clk	Clock
rst_n	Asynchronous reset signal, active low
mode	Decryption/encryption mode selection, encryption when "1", decryption when "0"
load	Data_in and key will be loaded at des_clk's rising edge when "1"
data_in[63:0]	Input data port
data_out[63:0]	Output data port
key[63:0]	64-bit Key
ready	When the module accomplishes decryption or encryption, ready turns to "1"

1)Sub-keys generating unit: During the 16 times of iterative computation, the sub-key generating unit generates sub-keys for those iterative computation. The 56-bit key goes through a 15 registers, forming a pipeline. By using the permutation table and the signal "mode", 16 sub-keys can be gotten, as K1, K2, …, K16.

2)*P-Boxes generating unit:* The core part of the DES logic module is the 16 rounds of iterative computation. Every round of iterative computation is produced by P-Boxes generating unit, shown in Fig. 4. P-Boxes mainly contains E-Boxes, S-Boxes and expand operations of P-Boxes. The 32-bit input data goes through E-boxes and expands to 48-bit, and then do the XOR operation with the 48-bit sub-key, and the 48-bit result data is again

shrinks back to be 32-bit through 8 eight substitution boxes (S-boxes).

3) *Top module*: Top module consists of initial permutation (IP), inverse initial permutation (also called final permutation, FP), replacement selection of keys and 16 rounds of iterative computation. Initial permutation rearranges and disturbs the original sequence of the original 64-bit plaint text. This is the same way that final permutation used. A new register is inserted during the every round of iterative computations. So pipeline architecture is formed from the 16 rounds of iterative computations.

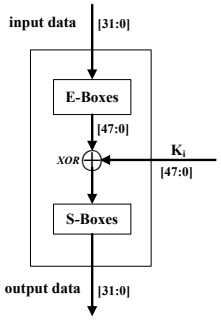


Figure 4. Block graph of P-Boxes generating unit

B. Register control unit

Registers realize the communication between LEON3 processor and DES logic module. The DES IP core mainly consists of some registers listed in the table II. All the registers are 32 bits. And three address lines (the 5th, 4th and 3rd address line) are used to code for these registers. Table II shows the registers this IP core used. Since LEON3 processor is a 32-bit processor, and the bus width of AMBA APB is also 32-bit in this SOPC platform, all the registers are designed in 32-bit for easier control. Thus two registers "datain_L" and "datain_H" are used for 64-bit data input. This is the same with key and data output. Ready signal can be leaded out and arouse an interrupt.

The connection between registers and DES logic module is shown in Fig. 5. A 64-bit shift registers are used to combine the "datain_L" and "datain_H" to make a 64-bit data input. This is also the same with data output and key.

TABLE II. REGISTERS

Register	*offset*	*R/W*	*Function*
mode	0x00	W	Mode selection register. Encryption when "1", decryption when '0'.
load	0x04	W	Data and key stowage register. Write 1 will load data_in and key to start encryption or decryption.
datain_L	0x08	W	Data input register. Lower 32 bits for data input
datain_H	0x0C	W	Data input register. Higher 32 bits for data input
dataout_L	0x10	R	Data output register. Lower 32 bits for data output
dataout_H	0x14	R	Data output register. Higher 32 bits for data output
key_L	0x18	W	Key input register. Lower 32 bits for key input
key_H	0x2C	W	Key input register. Higher 32 bits for key input

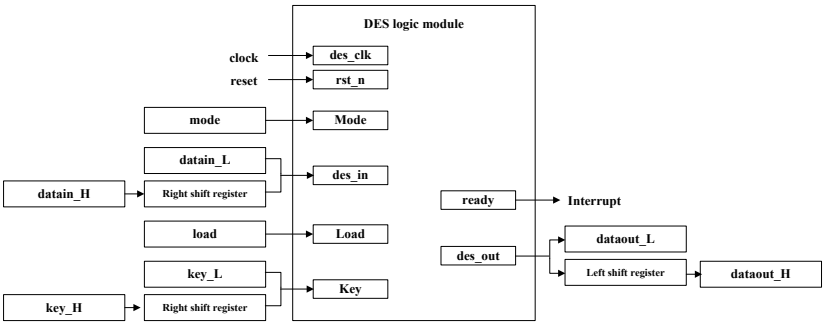


Figure 5. The connection between registers and DES logic module

C. AMBA APB interface

In APB interface unit, APB bus signals control the clocks and registers, and then indirectly control the runtime and operating mode of DES logic module. APB interface unit accomplishes the read operation and write operation, as well as ensures the timing control. APB operation mode is shown in Fig.7 [10]. IDLE status is the default status of peripheral bus. Status changes to SETUP status if there is a need for transmission. The corresponding selecting PSELx signal should be set high. SETUP status will be held for one bus clock and at the next clock the status will always be the ENABLE status. At ENABLE status, PENABLE signal will be set high at The PENABLE status will also be held for one bus clock. If there is no further transmission requires, bus will return to IDLE status. Besides, if a transmission is required, bus will turn to SETUP status.

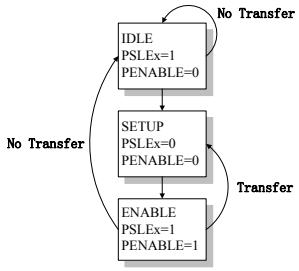
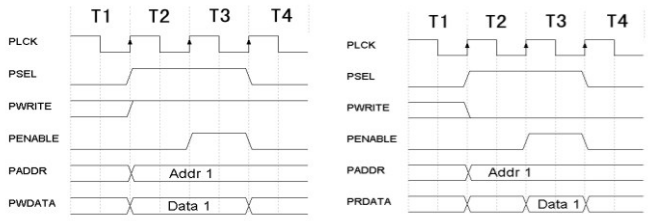


Figure 6. APB operation mode

Fig. 9 (a) shows the timing characteristics of APB write operation. T1 period refers to IDLE status. T2 is SETUP status, during which PSELx signal is sent from APB Bridge to choose this DES IP core, and registers are selected according to the address. T3 refers to ENABLE status. PENABLE signal is created by APB Bridge to tell the DES IP core that data is available. During this period, any writing operation to the bus will change the data in corresponding registers. For example, when the address bus is "000", the mode register is selected. All writing to this address will write mode register. In T4 bus returns to IDLE status. In order to save power, signal PWRITE and address keep unchanging until next transmission start. Read operation is similar to write operation, with the main difference of PWRITE signal. As is shown in Fig. 9 (b), PWRITE signal must be high is write operation while low in read operation.



(a) Write operation (b) Read operation

Figure 7. APB timing characteristics

V. EXPERIMENT RESULTS

Firstly the DES IP core is simulated in RTL level with testbench under Modelsim environment and the simulation waveform is shown Fig. 8. The key data is 64'H 1234 1234 1234 1234, and the input plaintext is 64'H 1234 56789 ABC DEF0. The resulting ciphertext is 64'H 386A 976C 5361 10BC when the DES IP core works in encryption mode. The key data is also 64'H 1234 1234 1234 1234, and the input plaintext is 64'H 386A 976C 5361 10BC. The resulting ciphertext is 64'H 1234 5678 9ABC DEF0 when the DES IP core works in decryption mode. It is clear that both encryption and decryption function well.

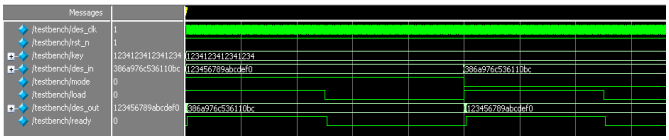
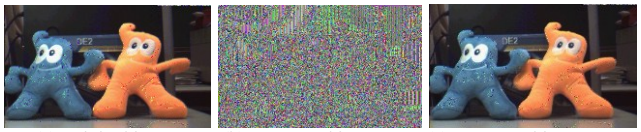


Figure 8. Simulation waveform

The DES IP core is thereafter embedded into the LEON3 SOPC platform as Fig.1 shows. After synthesizing, translating, mapping, place & routing, and downloading by Altera Quartus II, the LEON3 SOPC platform runs well at 50MHz on Altera DE2-70 development board. Fig. 9 shows the RTL top of the DES IP.

Figure 9. RTL top of DES IP

Then a 400×240 24-bit RGB color image is sent to the DES IP core from LEON3 processor to operate encryption and encryption. Fig. 10 shows results, proving that the encryption and decryption execute correctly.



(a) Original image (b) Encrypted image (c) Decrypted image

Figure 10. Test of the DES IP core

Also a simple encrypting system is built to test the IP after adding a Camera module and a LCD module in to this SOPC, as showed in Fig. 11.

Figure 11. Encrypting System on DE2-70

Table III shows the flow summary and occupancy of this IP core, and the power analysis is as well put out by Altera Powerplay Power Analyzer. The Quartus II version is VERSION 11.1 Build 173 01/11/2011 SJ Full Version, and the FPGA is a Cyclone II EP2C70F896C6 FPGA. From the

tables we can see this DES IP core doesn't occupy too much resource and consumes only 219mW.

TABLE III. FLOW SUMMARY

Title	*Occupancy*
Total logic elements	4,381 / 68,416 (6 %)
Total registers	1694
Total pins	72 / 622 (12 %)
Total combinational functions	3,693 / 68,416 (5 %)
Dedicated logic registers	1,694 / 68,416 (2 %)

TABLE IV. POWERPLAY POWER ANALYZER SUMMARY

Title	*Power analysis*
Total Thermal Power Dissipation	219.33 mW
Core Dynamic Thermal Power Dissipation	2.65 mW
Core Static Thermal Power Dissipation	155.04 mW
I/O Thermal Power Dissipation	61.65 mW

VI. CONCLUSION

This paper presents a novel method to design and implement a DES algorithm IP core based on LEON3 SOPC platform. The DES IP tested within the LEON3 SOPC platform and results show that it functions properly. Comparing with common hardware implementation of DES algorithm, this DES IP core has much more application prospects in the SoC designs since it is a common APB IP core. At the same time the method this paper puts out to design and implement an AMBA APB slave device in LEON3 architecture can also be referred to.

REFERENCES

[1] NBS, Data Encryption Standard, FIPS Pub. 46, US, National Bureau of Standards,Washington DC, January 1977.

[2] Zodpe H. D., Wani P. W., Mehta R. R.. "Design and implementation of algorithm for DES cryptanalysis", 12th IEEE International Conference on Hybrid Intelligent Systems (HIS), pp 278-282, 2012.

[3] X. Shuang-jian, Y. Liang, X. Fang-fang. "The Principle of DES Algorithm and Realization on FPGA". Computer Technology and Development, vol. 21, no. 7, pp 158-160,164, 2011.

[4] J. Yao, H. Kang. "FPGA implementation of dynamic key management for des encryption algorithm". 2011 International Conference on Electronic & Mechanical Engineering and Information Technology (EMEIT 2011), pp 4795-4798, 2011.

[5] M. Cheng-hai, L. Zhi-jun; M. Xiaoyue "Design and implementation of APB bridge based on AMBA 4.0". 2011 International Conference on Consumer Electronics, Communications and Networks, CECNet 2011 - Proceedings, pp 193-196, 2011.

[6] H. Chun-Ming, W. Chien-Ming, Y. Chih-Chyau, L. Kuen-Jong, W. Chin-Long. "Programmable System-on-Chip (SoC) for Silicon Prototyping". IEEE Transactions on Industrial Electronics, vol. 58, no. 3, pp 830-838, 2011.

[7] S. Shi, G. Tong. "Design Method of Data Gathering and Displaying System Based on SOPC". Microcomputer Information, no. 1, pp104-106, 2011.

[8] P. Somerlinck, S. Habinc, J. Gaisler, S. Redant, B. Glass. "LEONDARE - Sparc V8 processor with high-speed FPU and MMU". Proceedings of DASIA 2008 - Data Systems in Aerospace, 2008.

[9] GRLIB IP Library User 's Manual, Version 1.1.0 B4100, October 1, 2010.

[10] Advanced RISC Machines Ltd, AMBA Specification Rev2. 0. 1999.