An Implementation method of Encrypting Partial Bitstream Based on FPGA

Abhishek Kumar, Kumar Gaurav, Anshuman kumar Gautam

Abstract— Partial reconfiguration of FPGAs requires loading partial bitstream. But loading a faulty or corrupted partial bitstream might cause some errors when undergoing reprogramming; what's more, it may damage the FPGA device. A encrypt method based on cyclic redundancy check (CRC) before loading them into the device is introduced. The encrypt method of partial bitstream can realize the data integrity and security of FPGA Partial Reconfiguration.

Index Terms— AES, CRC, encrypting partial bitstream, FPGA, partial reconfiguration.

———————————— ◆ ————————————

1 Introduction

The technique of Reconfigurable is a kind of brand-new mode in digital circuit design, which keeps high calculate speed with hardware, high efficiency and it has advantages of flexible software with short development cycle and easy maintenance characteristic[1]. Partial reconfiguration is a kind of dynamically reconfiguration, which refers to the selected area FPGA device for reconfiguration and the rest remain the current function. From the beginning to the end, the circuit keeps the continuity of the system. Partial reconfiguration could improve the utilization of the resources, shorten the cycle of the reconfiguration and improve the reliability of the system. But loading a faulty or corrupted partial bitstream might cause some errors when undergoing reprogramming or it will damage the FPGA device, thus, in this application, a method has been presented to realize the data integrity and security of Partial Reconfiguration through software and hardware.

2 The Technique of Encryption in the Reconfiguration

2.1 Cyclic redundancy check error

CRC checking is the most common error check code in the data communication system. The basic principle is to add a R-bit check code behind a K-bit information code to make the length of the whole coding is N-bit. There is a highest power($N-K=R$)polynomial: $G(x)$,according to $G(x)$,it could generate a check code with K-bit. In virtex architectures, the configuration logic continuously calculates a CRC value as it loads a full bitstream into the configuration memory. The configuration logic cannot facilitate this type of CRC checking for partial bitstreams because it lacks a buffering mechanism as it reads the partial bitstreams. Thus, the PRC system implements CRC checking of partial bitstreams in FPGA logic before loading them through the ICAP, to make buffering of partial bitstreams feasible.***********

.

2.2 Encryption support

Encryption Standard(AES) is a kind of encryption standard which reco[illegible]merican National Standards Institute. T[illegible]e synchronization of the external [illegible]dules: Interface control unit, Encrypt and decrypt unit and Key expansion unit[2]. Fig.1 shows the architecture of the AES module.

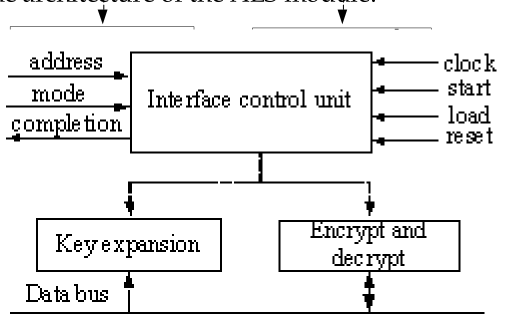


Figure1. Architecture of the AES module

The algorithm of AES is based on finite field GF (2^8), and the algorithm contains two mathematical operations: Multiplication and exclusive or operation, the primary formula refer to formula (1):

$$\begin{cases} S'_{0,c} = (\{02\} * S_{0,c}) \oplus (\{03\} * S_{1,c}) \oplus (\{01\} * S_{2,c}) \oplus (\{01\} * S_{3,c}) \\ S'_{1,c} = (\{01\} * S_{0,c}) \oplus (\{02\} * S_{1,c}) \oplus (\{03\} * S_{2,c}) \oplus (\{01\} * S_{3,c}) \\ S'_{2,c} = (\{01\} * S_{0,c}) \oplus (\{01\} * S_{1,c}) \oplus (\{02\} * S_{2,c}) \oplus (\{03\} * S_{3,c}) \\ S'_{3,c} = (\{03\} * S_{0,c}) \oplus (\{01\} * S_{1,c}) \oplus (\{01\} * S_{2,c}) \oplus (\{02\} * S_{3,c}) \end{cases} \quad (1)$$

AES is a iteration group code with the Key, the process of encryption contain a initial key-addition that be called AddRoundKey, then begin to process $N_r - 1$ times conversion(Round), and finally it use a round conversion(Final Round).The both of initial key-addition and every round conversion make input with state and a Round Key, each cycle carry out four different operates for 128bit Data flow, the order is SubBytes, ShiftRows, MixColumns and AddRoundKey,to accomplish a process of the encryption with ten times round iteration.Fig.2 shows the Flow chart of AES Encryption algorithm.

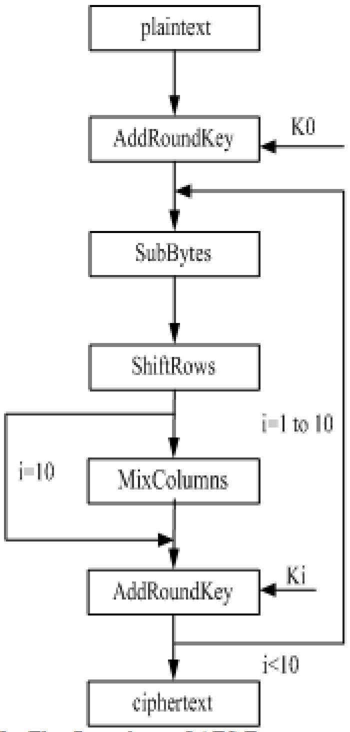


Figure2. The flow chart of AES Encryption algorithm

AES Encryption has been the Core technology in data encryption by its security and the others superiority. The algorithm of AES could resist any Password attack that all we knew, no matter it uses the Feedback mode or No feedback mode, AES Encryption is very excellent in implementation of hardware and software, thus it has the strong practicability in aspect of reliability of Data encryption.

Virtex-5 and Virtex-6 devices have on-chip advanced encryption standard (AES) decryption Logic to make the data integrity and security of Partial Reconfiguration. It provide a 256-bit key. After load the key to the configuration memory, the configuration logic will decrypt the encrypted bitstream.

3 Implementation of the Software and the Whole Design of the System

3.1 Implementation of the software

The design of software could splits a partial bitstream into packets. Each individual packet can be up to 2048 bytes or 512 32-bit words. The first 32-bit word of the packet is a packet header and the last word of the packet contains a CRC32 value calculated across the entire packet including the packet header. The internal configuration memory of a Virtex device is subdivided into many frames. In the design of the software, bitstreams contain the write commands which in internal configuration logic are deposited into the Frame Data Register In(FDRI) and the Frame Address Register (FAR). The FDRI is a pipeline input stage for configuration data frames to be stored in the configuration memory. The FAR specifies the frame address, A non-encrypted partial bitstream generated by BitGen contains write blocks to the FDRI register. Each FDRI block consists of an address write to the FAR register and a multi-word write to the FDRI register [5]. BitGen protects the full bitstreams by encrypting the FDRI writes with AES-256.The system of the encrypted Partial Reconfiguration can encrypts the content of partial bitstreams before the system packet zing and inserting CRC values.

Controller and ICAP should be initialized before the partial bitstreams encrypted and then start Perl script through the command:

Xilperl eprBitgen.pl -fb <fullbitfile[.bit]> -pb <partialbitfile1[.bit]> -pb <partialbitfileN[.bit]> -key <keyfile[.nky]> --encrypt

Encrypted information input:

-fb <fullbitfile[.bit]>	input full bitstream file
-pb <partialbitfile[.bit]>	input partial bitstream file
-key <keyfile[.bit]>	input key file
-encrypt	enables encryption

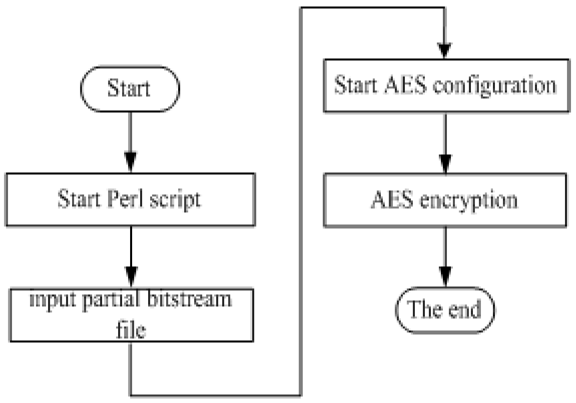


Figure3. The flow chart of encrypting partial bitstream

The Perl script also scrambles the IDCODE in the partial bitstream. The EPRC core in hardware descrambles the IDCODE into its original value. BitGen-generated .bit files contain proprietary header information that does not need to be downloaded to the FPGA, The script-generated .prc files preserve the header information and update the total byte size with the added number of bytes for all packet headers and CRC words. the following code illustrates the structure of a sample partial bitstream after running the script:

```
-Bitstream Header          (if included)
-00 00 01 FF               (Packet 0, length: 512)
```

-Configuration Data (510 words)
-95 BD EA 34 (Packet 0 CRC)
-00 01 01 FF (Packet 1, length: 512)

-Configuration Data (510 words)
-26 A1 C6 19 (Packet 1 CRC)
-00 1D 00 11 (Packet 29, length: 4)

-Configuration Data (Two words)
-8E D7 C8 47 (Last Packet CRC)

When the script be started, it will splits the encrypted partial bitstream into packets, encrypting and adding CRC values into each packet and then input the partial bitstream that to be encrypted, starting the encrypted configuration and then input Key(AES)to encrypt partial bitstream. At this time, the function of the task is over. Fig.3 shows the flow chart of encrypting partial bitstream.

3.2 The whole design of the system

The design include five modules: Packet FIFO, Config, ICAP outputs, Reset and a ERROR. Fig.4 shows a top-level of the Encryption Partial reconfiguration(EPRC).

Packet FIFO module ensures the incoming packets pass error checking prior to being forwarded onto other modules, the Packet FIFO module receives the packets through a FIFO interface and uses the packet information (length, sequence number, and CRC value) to perform error checking; The Config module monitors configuration data forwarded from the Packet FIFO module for configuration commands. It also descrambles the IDCODE into its original value; The ICAP outputs module is responsible for reading the ICAP FIFO and output. It also monitors the PR_ERROR signals for status. The ICAP outputs module generates an abort sequence and stops reading the ICAP FIFO if it detects an error condition; The Reset module consists of a 3-bit down counter which in charge of download and reset; The ERROR module encodes the various error signals inside the core into PR_ERROR and PR_ERROR_CODE signals for external usage.Table1 shows the encoding of the PR_ERROR_CODE signal.

Table1. PR_ERROR_CODE signal

Inputs				Outputs
EXT_PR_ ERROR	RE-SERVE D	CONFIG _ERROR	PACKET _ERROR	PR_ERROR _CODE
1	x	x	x	1and EXT_PR _ERROR _CODE
0	0	1	x	0010
0	0	0	1	0001
0	0	0	0	0000

4 Implementation of the Hardware and System Verification

Bitstream core provide a netlist file and a implemented design, make sure that the content of each packet in hardware by calculating a new CRC value and matching it to the one contained in the packet[5]. The core that in the Virtex-5 device also checks the ordering of the received packets by monitoring the contained sequence number in each packet. The operation is happened before Configuration Data of the encryption in each packet that be send to ICAP of the partial bitstream. Internal encryption logic be encrypted when it load the configuration data of the encryption to the Configuration Data Memory.

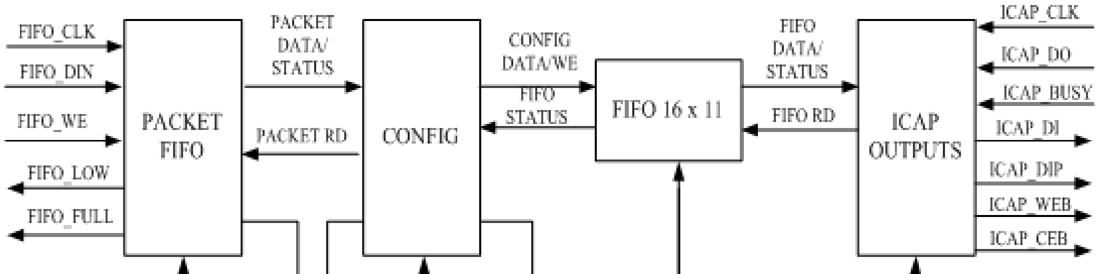
In order to save the resource of FPGA in this design, the PlanAhead development platform is used for Floor planning, Table2 shows the utilization of FPGA in this design.

Table2. The utilization of FPGA in this design

Resource	Utilization	Available	Utilization
Register	3189	28800	11%
LUT	3300	28800	11%
Slice	1780	7200	24%
IO	55	480	11%
BSCAN	2	4	50%
DSP48E	3	48	6%
ICAP	1	2	50%
PLL_ADV	1	6	16%
Global Clock Buffer	6	32	18%

This system is implemented in Virtex-5 device. Copy the generated scripts to the SystemACE™ interface which is used as a delivery method for partial bitstreams. The task of this system is to send the partial bitstreams from a storage device (such as a CompactFlash card) to the EPRC core.

SystemACE™ interface provides a simple UART console to select the partial bitstreams, check the design status after partial bitstreams, and identify the error conditions. Connect the UART cable to the host PC. Launch a terminal program with baud rate 115200 and then launch the iMPACT to encrypt



the design of the encrypting partial bitstream by the encryption Key. The partial bitstream have been protected before being loaded into FPGA and realizing the data integrity and security of Partial Reconfiguration.

To observe the EPRC interface signals as the

design operates on the board,and use the ChipScope™ analyzer core to analyse the condition of the encrypted partial bitstream. Fig.5 and fig.6 shows the assertion of PR_DONE.

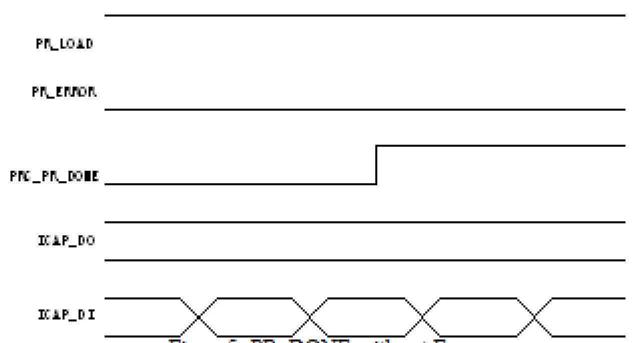


Figure5. PR_DONE without Error

The two figures show the assertion of PR_DONE after the FIFO write of the last word of the last packet. For a successful PR, the core only asserts ICAP_DI as the Fig.5 shows.

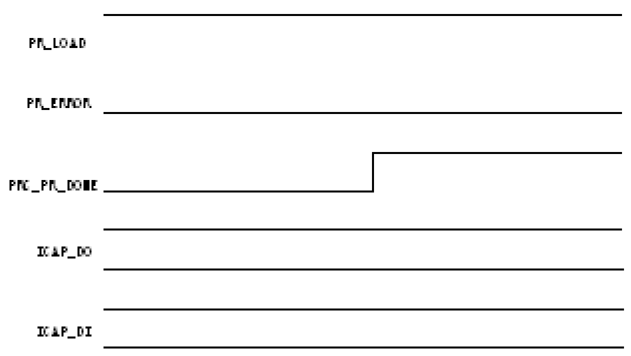


Figure6. PR_DONE with Error

In case of an error, the core also asserts ICAP_DI shows no data like fig.6. From the analysis, the result of the encrypted partial bitstream will be determined; the system is implemented in Virtex-5 device by the verification.

5 Conclusion

The technique of Partial reconfiguration optimizes the efficiency of resource utilization of the FPGA and improves the reliability of it. Partial bitstream is the most important data file in Partial reconfiguration. Through the Advanced Encryption Standard (AES) that is provided by Virtex-5 FPGA device, data integrity and security for Partial Reconfiguration are implemented using ISE and PlanAhead development platform and verified in a Virtex-5 Development Board. By this method, FPGA devices can effectively avoid damage by loading a faulty or corrupted partial bitstream and protecting data integrity and security for Partial Reconfiguration.

References

[1] Wu Feng-Yan. "The dynamic partial reconfigurable system based on FPGA," Master's degree thesis,Guang Xi province: Guang Xi university,2010.

[2] Li Wan-Cai,Wang Ying,Xu Jun,Peng Cheng-Lian. "Design of data encryption in reconfigurable system,"in Computer Engineering and Design, 20rd ed,vol.29,2008,pp.5148-5149.

[3] Liu Zhen-Zhen. "Implementation of AES Encryption based on FPGA",in Modern electronic technology, rd ed 23, 2007,pp:103-104.

[4] reconfigurablesystem[EB/OL].http://hi.baidu.com/simonyuee/blog/item/b6975409197b22ae2eddd4a3.html,2005.

[5] Xilinx,XAPP887(v1.0) PRC/EPRC:Data Integrity and Security Controller for PartialReconfiguration January 12, 2011.

[6] QIN Xiang-Ju, ZHU Ming-Cheng, ZHANG Tai-Yi, WEI Zhong-Yi. "Analysis of the Fundamental and Implementation Method about Dynamic Recof igurable FPGA". In Chinese Journal of Electron Devices. 27rd ed, vol.2,2004, pp:277-278.

[7] Tian Yun,Xu-Wen-Bo,Hu Bin. Xilinx ISE Design Suite 10.x Guide.Posts&Telecom Press,Bei jing,2008

[8] Xilinx,UG191(V3.9.1), Virtex-5 FPGA Configuration User Guide,August,20,2010.http://www.xilinx.com/support/documentation/user_guides/ug191.pdf