A novel Implementation of Encryption Algorithms based on FPGA Gates

Huabo Zhang[1, a], Hongmin Gao[2,b], Xuetian Wang[3,c] and Xiangzhi Yu[4,d]

[1,2,3]Department of Information and Electronics, Beijing Institute of Technology, Beijing, China

[4]Department of Engineering, Beijing Science Technology Management College, Beijing, China

Email: [b]gaohm@bit.edu.cn (communication author)

[a]13501168815@163.com, [c]wangxuetian@bit.edu.cn, [d]yuxiangzhi129@126.com

Keywords: Time delay, Encryption Algorithm, FPGA Gates, IC Design, Embedded Software

Abstract. A novel implementation of encryption algorithms based on FPGA gates is described in this paper. The encryption algorithm is focused on different FPGA devices with the same number of cascaded NOT gates but have different time delays to signals. The cascaded NOT gates are achieved by programming. These different time delays are the theoretical basis of the encryption algorithm. In this paper, we will firstly introduce the core idea of the algorithm. And then, the implementation of the encryption algorithm, IC design and Embedded Software will be explained in detail. At the end of the paper, the factors which affect the encryption will be analyzed.

Introduction

Field Programmable Gate Array (FPGA), as a kind of Programmable Logic Devices (PLD), is initially developed in the late 1970s. FPGA have gained widespread acceptance as substitutes for ASICs in many applications. Such as electronic engineering, communication, aerospace, industrial automation, electronic information, home appliances, bio-engineering ect.. The basic motivation is the elimination of the lengthy manufacture cycle and the reduction of manufacturing cost [1]. In fact, the characteristic of repeatable programming has made FPGA very attractive in the embedded market, where software and functionality updates are usual and desirable by customers.

At the same time, in order to protect the core technologies from being copied, the FPGA developers and FPGA vendors have proposed different strategies of security [2,3]. This paper proposes a novel encryption algorithm to FPGA devices based on its logic gates for the protection of core technologies.

Background and Theoretical Basis

The encryption algorithm of was originally proposed in the research for “Multi-Channel Precision Time Interval Measurement Instrument (MTIM)”. The main function of MTIM is to measure the time interval between measured signals and the reference signal [4]. The precision of MTIM can reach 100ps at room temperature. We have done the following experiment: To achieve a NOT gate, 100 cascaded NOT gates, 500 cascaded NOT gates and 1000 cascaded NOT gates separately by programming . And then use MTIM to measure the time delays of these cascaded NOT gates for 200s respectively. The horizontal axis represents the measuring time and the vertical axis represents the time interval between measured signal and reference signal in real time.The results are shown in Fig. 1 to Fig. 4.

Seen from Fig. 1 to Fig. 4, the MTIM shows the maximum values, minimum values, average values and RMS values automatically in real time. In this experiment, we take the average time delay during the 200s as the standard time delay. According to the measured results, the average time delays are 0.38160ns, 39.19218ns, 189.25615ns and 371.26392ns separately. Considering the precision of MTIM is 100ps, the average time delays are corrected as Table 1. From Table 1, signal is delayed about 0.4ns when passes through a NOT gate inside FPGA. However, the time delays of cascaded NOT gates do not have a linear additive. For example, when signal flows through 1000 cascaded gates inside FPGA, the time delay is less than 400ns.

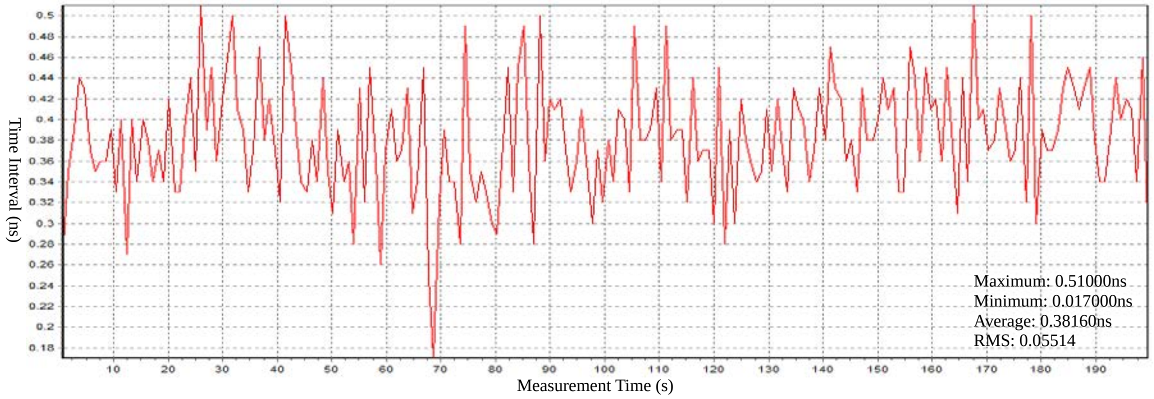

Fig. 1 a NOT gate

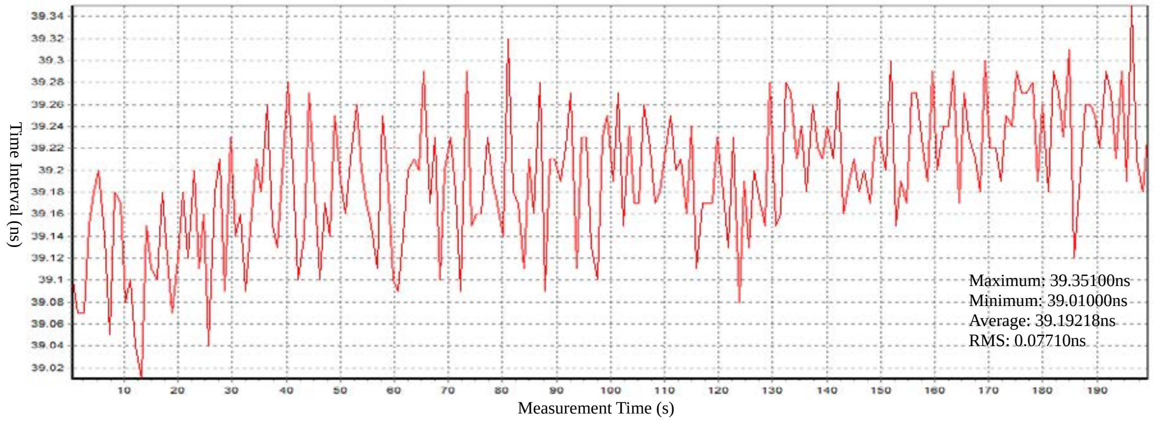

Fig. 2 100 cascaded NOT gates

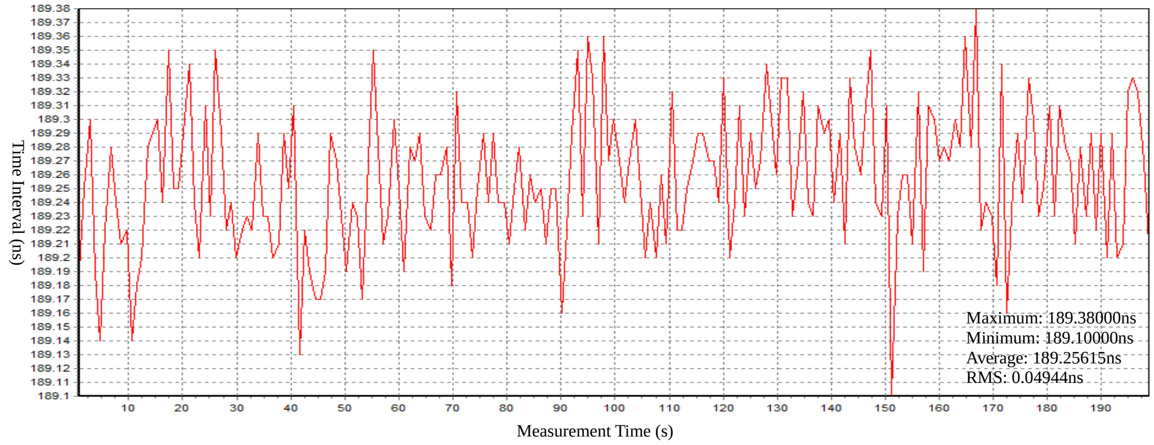

Fig. 3 500 cascaded NOT gates

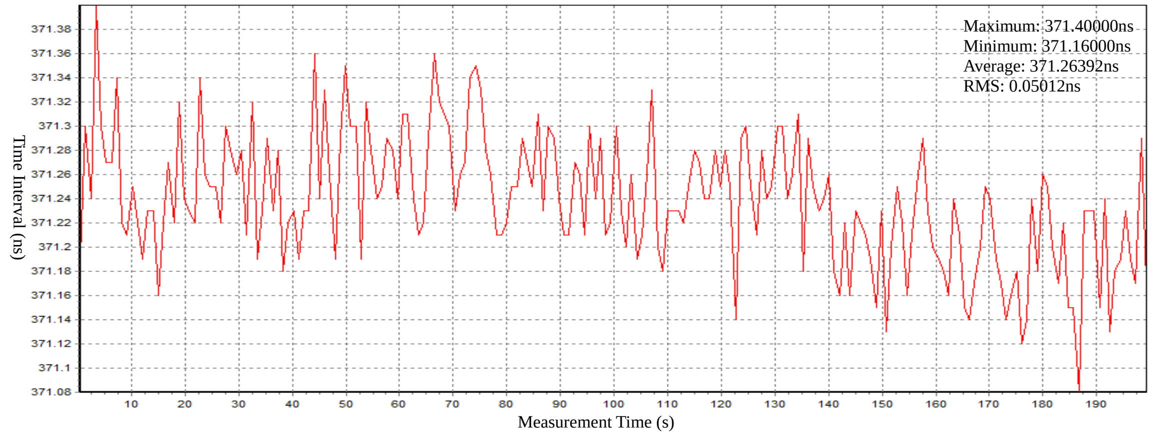

Fig.4 1000 cascaded NOT gates

Found after doing a lot of tests, the same number of cascaded NOT gates inside different FPGA devices have different time delays. With the increasing number of cascaded NOT gates, the differences between time delays are bigger. Table 2 shows the time delays of 1000 cascade NOT gates inside four different FPGA devices.

Table 1 The corrected average time delays

Number of cascaded NOT Gates	Time delay（ns）
1	0.4
100	39.2
500	189.3
1000	371.3

Table 2 Different time delays

FPGA No.	Time delay（ns）
1	365.6
2	357.8
3	371.3
4	362.1

Seen from Table 2, the three different FPGA devices with the same number of cascaded NOT gates have different time delays. This is the theoretical basis for the encryption algorithm.

The Implementation of Encryption Algorithm

Hardware Structure. The hardware structure to realize the encryption algorithm is presented in Fig. 5. The hardware structure mainly includes three parts: several FPGA development boards, a “Multi-Channel Precision Time Interval Measurement Instrument (MTIM)” and a computer.

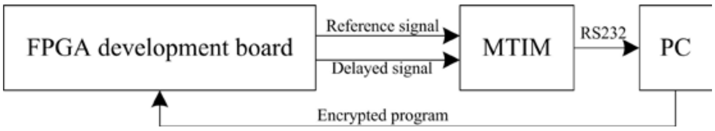

Fig. 5 Hardware structure

The FPGA devices used in this experiment are produced by Altera Corporation. The type is EP25T144I8 [5]. The precision of the MTIM is 100ps at room temperature. Moreover, the MTIM comes with a set of software to display the time delays between input signals and the reference signal in real time. The computer must be required with a serial port, by which the measured data are transmitted to computer and then stored in the form of files.

Time delays Measurement of cascaded NOT Gates. Time delays of cascaded NOT gates inside FPGA is measured by the following steps:

(1) Programming to achieve *X* cascaded NOT gates (where *X* can be 1, 100, 500 and so on);

(2) Providing an input signal to the FPGA, which can be a sine wave or square wave;

(3) Dividing the above input signal into two(CH1 and CH2), one is exported directly to the MTIM as the reference signal, the other passes through the *X* cascaded NOT gates and then exported as the measured signal [6].

(4) Measurement. When MTIM starts to work, the display software will display the time delays between the measured signal and the reference signal in real time. However, the time delays are not fixed, floating in ±100ps.

Structure flowchart of time delays measurement is as follows in Fig. 6.

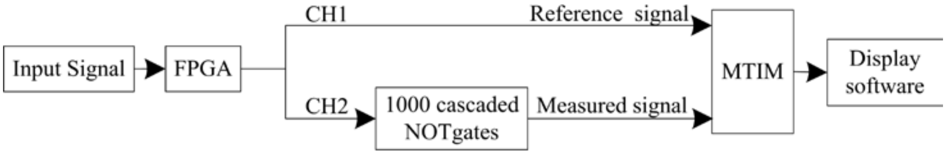

Fig. 6 Measurement of time delays

After these steps above, we will obtain the time delays of *X* cascaded NOT gates of a FPGA device. The time delay is the only “identity” to a FPGA device.

Correction of Time delays. The precision of MTIM determines the precision of time delays. That means the precision of the time delays is ±100ps. But these time delays are not the real “identity” programmed into program to identify different FPGA devices. In fact, the real “identity” are corrected values based on the measured values according to sampling clock.

The main reason to correct these time delays is related to the frequency of sampling clock. FPGA judges the time by clock count. If we want FPGA to identify this precision of time delays accurately, the frequency of sampling clock of FPGA must reach more than 10GHz. Such a high frequency of sampling clock is almost impossible to accomplish for most FPGA devices. Therefore, the correction of the time delay is very necessary.

The frequency of sampling clock of FPGA devices used in this experiment can only reach 500MHz [5]. That means the clock cycle is 2ns. When FPGA distinguishes time delays by 500MHz sampling clock, the differences between these time delays can not be less than 2ns.Otherwise, it will not be able to distinguish them correctly.

Supposing two measured time delays are 263.2ns and 264.5ns respectively. It is impossible to distinguish the two values if using the sampling clock of 500MHz. The two values must be processed as follows: 263.2ns is corrected to 262ns and 264.5ns is corrected to 264ns. After processing, the difference between them can reach 2ns. Then the sampling clock of 500MHz can distinguish the two values correctly by clock count.

Supposing T_0 is the time delay measured by MTIM. T_1 is the time delay after being processed. T_{err} is the error for correction. So the relationship betweenT_0, T_1 and T_{err} is:

$$T_1 = T_0 \pm T_{err} \quad (1)$$

The maximum value of T_{err} is T_{clk}, where T_{clk} is the cycle of sampling clock.

Implementation of Encryption Algorithm. The structure diagram to achieve the encryption algorithm is shown in Fig 7.The external reference clock is multiplied to a high frequency by internal phase-locked loop of FPGA as the sampling clock. Assuming the cycle of the sampling clock is T_{clk}.

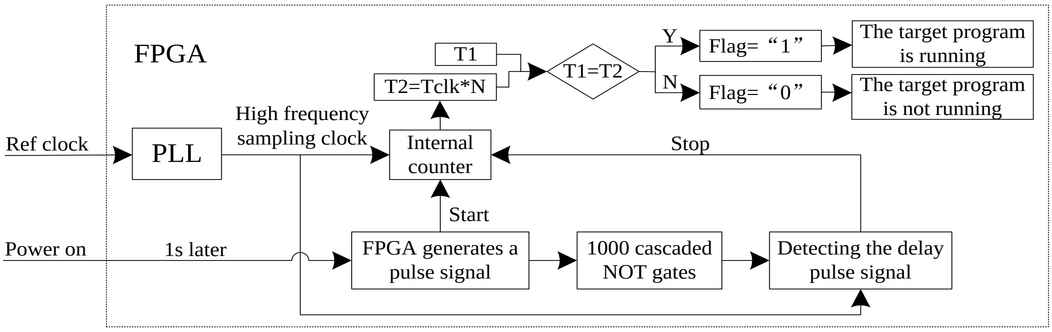
FPGA
T1
T2=Tclk*N
T1=T2
Y
N
Flag= "1"
Flag= "0"
The target program is running
The target program is not running
High frequency sampling clock
Ref clock
PLL
Internal counter
Stop
Start
Power on
1s later
FPGA generates a pulse signal
1000 cascaded NOT gates
Detecting the delay pulse signal

Fig. 7 The structure diagram to achieve the encryption algorithm

When the system is powered on for 1s (In order to make the system reach a steady state), FPGA generates a pulse signal. At the same time, FPGA initiates the internal counter and begins to count with the sampling clock. In addition, the sampling clock continuously detects the delayed pulse signal which passes through the *X* cascaded NOT gates inside FPGA. Once detecting the rising edge or falling edge of the delayed pulse signal, the counter stop to count immediately.

Supposing the value of the counter is N, which is obtained from the time starting the counter to stopping it. Therefore, the time delay of cascaded NOT gates is:

$$T_2 = T_{clk} \cdot N \quad (2)$$

Comparing the relationship between the value T_1 measured by MTIM and the value T_2 measured by sampling clock of FPGA whether they are equal. If T_1=T_2, the identification signal is effective, i.e "*Flag=1*". Conversely, the signal is invalid, i.e "*Flag=1*".

The identification signal "*Flag*" is a key variable, which communicate with the encryption program and the encrypted program. The identification signal "*Flag*" can be seen as the result of the encryption program. And it is used as a control variable programmed into the encrypted program. Before running the encrypted program, it is essential to determine the validity of the identification signal. If it is effective, then the encrypted program will be executed. Otherwise, the encrypted program will not be executed.

Analysis of Influence Factors

The above introduces the basic theory and implementation of encryption algorithm in detail. In practical application, the encryption algorithm is affected by some factors.

The precision of the measurement instrument. The precision of the measurement instrument used in this experiment is 100ps. So the measurement instrument is completely enough to determine the time interval greater than 100ps. Supposing the precision of the measurement instrument is 10ns, and then the measurement error will be ±10ns. For a certain percentage of the FPGA devices, the time delays difference between them may be less than 10ns. For these FPGA devices, the measurement instrument can not distinguish these time delays.

The frequency of sampling clock. As the time delays measured by FPGA are obtained by the value of counter multiplying by the cycle of sampling clock. So the frequency of sampling clock is a key factor to encryption. Supposing the sampling clock is 100MHz, so the maximum error will reach 10ns. Thus, the difference of time delays less than 10ns can not be distinguished by sampling clock. For lower frequency sampling clock, it will lead to vast majority of FPGA devices can not be distinguished from each other.

The aging of FPGA devices. With the extension of working years, the internal structure of FPGA devices may get older to some extent [7]. For the aging FPGA devices, the time delays may have already varied. So the time delays initially set to the encryption program is no longer applicable, and that will lead to a failed encryption.

Conclusions

The encryption algorithm introduced in this paper has its own advantages. For example, it do not rely on the third party encryption tool; high integration in hardware; encryption procedure is simple and efficient; low costs. But also it has its own shortcomings. Limited to the precision of measurement instrument and the frequency of sampling clock, it is difficult to distinguish each FPGA device accurately. To be fair, as long as there is a certain percentage of the FPGA devices can be distinguished from others. That breaks the generality of program. To any developers, a non-universality program is of little value. Although the encryption algorithm has its limitations, but it also has huge market value in practical application.

Acknowledgment

At the point of finishing this paper, my deepest gratitude goes first and foremost to professor Gao, my supervisor, for his constant encouragement and guidance. Second, I would like to express my heartfelt gratitude to professor Wang, who gives lots of guidance about the thesis. Without their consistent and illuminating instruction, this thesis could not have reach its present form.

References

[1] H. Yang, A. G. Li, Newest Operative Technology Guide of FPGA/CPLD[M]. Tsinghua University Press, Beijing, China, 2005.

[2] Z. D. Wang. The research of implementation of generic algorithms Based on FPGA[J]. Instruments & Meters User, 2006.

[3] A. Lesea, "IP Security in FPGAs"[A], Xilinx, 2007.

[4] Wei Tao, Liangbo Ming. The principle and Function Realization of Multi-Channel Time Interval Measurement Instrument. The 4th China Satellite Navigation Conference[C], 2013.(In Chinese)

[5] EP2C5T144I8. Altera Corporation, 2008.

[6] Y. W. Xia. Digital system design tutorial (Second Edition)[M]. Beijing University of Aeronautics & Astronautics Press, 2008.(In Chinese)

[7] Zhenhua Yu, Weiliang Zhu. Research on dynamic aging technology of FPGA circuit[J]. Electronics & Packaging, 2010.(In Chinese)