Research on PLD Software Design of Communication Devices

WANG Jiasi ZHANG Ying Xiang Liang Du Songyang

Keywords: FPGA Software, Clock Design, Synchronous Design, Metastable State
Abstract: The PLD software design of communication device is taken as application background, under which this paper analyzes four typical issues of software design such as clock design, synchronous design, burr elimination and metastable state, gives solutions based on actual conditions and finally describes characteristics and methods of FPGA software testing. The research can provide references for design and test personnel of the FPGA software.

Introduction

With development of programmable technology, the field programmable gate array (hereafter referred to as the "FPGA") has been widely used for various electronic design applications. More and more complex systems adopt FPGA for design of core circuits and the scale of FPGA generally exceeds one million pcs. A universal FPGA chip can be immediately configured by means of programming to a hardware digital circuit based on the user's requirements, thus greatly accelerating development cycle of electronic products and reducing development costs.

Compared with traditional DSP design, FPGA has a great advantage in the communication system design subject to large data volume and high processing rate. The FPGA technology has been used in more and more communication devices.

Definition of FPGA

FPGA (Field-Programmable Gate Array) is a product developed on the basis of programmable devices such as PAL, GAL and CPLD. It is a semi- customized circuit of the application-specific integrated circuit (ASIC) field, which can solve the shortcomings of customized circuit and overcome the deficiency of limited gate circuits in original programmable devices. A universal FPGA chip can be immediately configured by means of programming to a hardware digital circuit based on the user's requirements, thus greatly accelerating development cycle of electronic products and reducing development costs. Just because of the above characteristics, more and more communication device has begun to adopt a number of FPGA devices.

Design Analysis and Research of Communication Devices

Clock design

Clock design is the key to FPGA software design and it is also applicable to the design of communication device software system. The entire system is designed generally by synchronous sequential circuit and the clock plays an important role in the design of synchronous circuit. So, we need to complete the clock design first.

Common clock types include global clock, internal logic clock and gated clock.

Global clock

The global clock, namely synchronous clock, is driven by global clock cabling network or

© 2016. The authors - Published by Atlantis Press

local clock network in the FPGA chip. The global clock shows high fanout, high precision, low Jitter and low Skew. It has the minimum delay for access to each register of the chip and the delay could be deemed as a fixed value. Therefore, it is recommended that global design should be adopted in all design processes. The global clock is designed by the following methods:

(1) The global clock is generated by PLL phase-locked loop.

(2) The clock generated by internal logic of FPGA chip is distributed to global clock cabling network.

(3) External clock is introduced to the FPGA via special global clock input pin.

In our design, it is generally recommended that all clocks of the circuit should be generated by PLL phase-locked loop. The PLL phase-locked loop can achieve frequency multiplication and phase shifting to help us easily obtain the clock under the required frequency and phase. In addition, it distributes the driven clock to global clock network or local clock network by default; and Jitter and Skew are very low.

The following figure shows a PLL phase-locked loop design taken from our project. The PLL is used to drive DDR interface module. To meet function requirements, the DDR interface needs three 133MHz clocks respectively with a phase of -90, 0 and -180. The figure shows generation module of the clock. We adopt Megawizard of Quartus II to generate IP core of PLL phase-locked loop. The inclk_66 is an input clock of PLL phase-locked loop. It is provided by external 66MHz crystal oscillator and can generate three global clocks by PLL frequency multiplication and phase shifting.

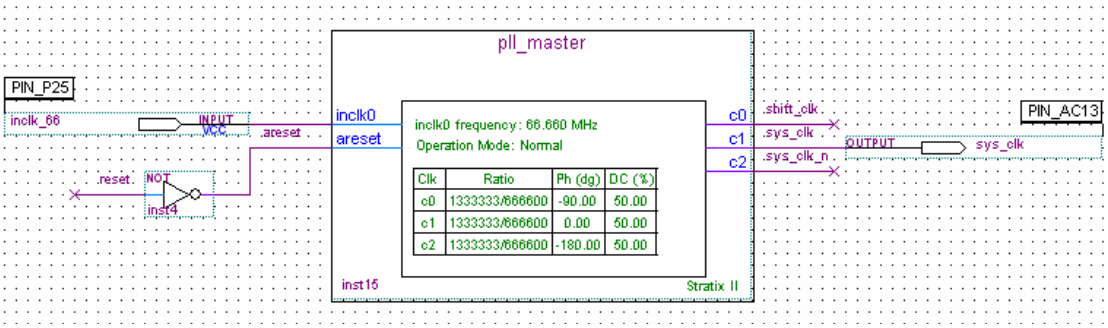

Figure 1 Phase-locked Loop Design

Internal Logic Clock

Internal logic clock refers to the clock generated by combinational logic or counter frequency division in a chip.

The combinational logic clock, especially the clock generated by multi-level combinational logic, is prohibited, for the combinational logic is prone to burrs; Jitter and Skew of the combinational logic circuit are relatively high, which can lower clock quality. Therefore, internal clock of the combinational logic apply only in case of low frequency and precision requirements.

The clock generated by counter frequency division also shall be less frequently used, because the clock could result in a larger delay so as to reduce the reliability of design and make static timing analysis complex. The logic function required by the counter frequency division can be replaced completely with PLL phase-locked loop or clock enable circuit.

In addition, ripple clock is generated by trigger, that is, output of a trigger is used as clock input of the other trigger. Clock frequency division circuit described in Section 1.1.2 is just a ripple clock. Triggers have a larger time offset between clocks and can easily violate the requirements for

time establishment and maintenance, thus causing a metastable state. So, the ripple clock shall be prohibited.

Gated clock

The gated clock shall not be used under normal circumstances, because the gated clock generated by the combinational logic is quite likely to generate burrs so as to cause great damage to the system. But some high-power systems need to have their power consumption reduced by using gated clocks.

The gated clock design described in the following figure is recommended and it will not generate burrs or metastable state generally, as the trigger can avoid burrs and the metastable state only occurs on the falling edge of source clock. The metastable state could encounter low phase of the source clock, but no impact will be generated finally.

It is advised to set the gated clock only on the top module and separate it into an independent top module. Thus, each module on the bottom has a single clock and its clock is not gated.

Synchronous design

Synchronous design means that all circuits are triggered by the rising or falling edge of the same common clock to work synchronously. But multi-clock domain is usually available in actual systems. In such a case, the concept of synchronization will be extended. It will not specially indicate the whole design is synchronized with the same clock edge but indicate the design shall reach local synchronization and circuits in each clock domain shall be synchronized on the same clock edge.

Advantages:

1. The synchronous design can effectively avoid the impact of burrs and make the design more reliable.

2. The synchronous design could easily add asynchronous reset, thus making the whole circuit have a definite initial state.

3. The synchronous design can reduce environmental impact on the chip and protect components from impacts of temperature, voltage and process.

4. The synchronous design can make static timing analysis simple and reliable.

5. The synchronous design can easily organize a flow line and improve chip running speed.

Design criteria:

1. The design shall, as practically as possible, adopt the same clock in the global clock network. The clock in the global clock network is the most simple and predictable, with high driving capacity. It can drive all triggers in the FPGA and ensure the clock skew is minimized for negligence.

2. Mixed clock edge sampling data shall not be used, that is, rising edge and falling edge of the clock shall not be used for the design simultaneously.

3. The clock generated by counter frequency division shall be less frequently used in the module. The logic function required by the counter frequency division can be replaced completely with PLL phase-locked loop or clock enable circuit. The shortcomings of the counter frequency division clock are making system clock uncontrollable, generating a higher clock skew and making the static timing analysis complex.

4. The gated clock shall not be used, as the gated clock generated by the combinational logic is quite likely to generate burrs so as to cause D trigger mal-operation.

5. When the entire circuit needs multiple clocks, it can be divided into several local synchronous circuits (taking the same clock as a module as far as possible). The interface between

local synchronous circuits is taken as an asynchronous interface and clock skew (ΔT) of each clock signal shall be strictly controlled.

6. Maximum actual operating frequency of the circuit shall not exceed theoretical maximum operating frequency. It shall have a margin to ensure reliable operation of the chip.

7. All registers and state machines in the circuit shall remain a known state when the system is reset.

Burr generation and elimination

(1) Race and hazard

When the input of a logic gate has two or more variables to change, the variables will change the state at different time because they are generated via different routes. The phenomenon caused by time difference is called race. Race results are quite likely to cause a hazard (such as the burr) and error and affect system operation.

The hazard of the combinational logic circuit causes the burr only when the state signal changes, which is transitional and will not make the steady state value deviated from normal value. However, the hazard is intrinsic in the sequence circuit and can cause the circuit output value permanently deviated from normal value or oscillated.

It is the simplest way for avoiding the hazard to allow the change of only one input variable or take register sampling method.

(2) Generation and hazard of burrs

The signal must be delayed when it is connected via logical unit in the FPGA device. The delay is related to connecting line length and logic unit quantity as well as manufacture process and operating condition of the device. Therefore, signal transmission time of the device cannot be accurately estimated; the race and hazard will be generated when multi-way signals are changed at the same time. Usually, some wrong peak signals will occur at the time. The peak signal is just the burr. Let us see how to generate the burr. In the following figure is a AND gate circuit:

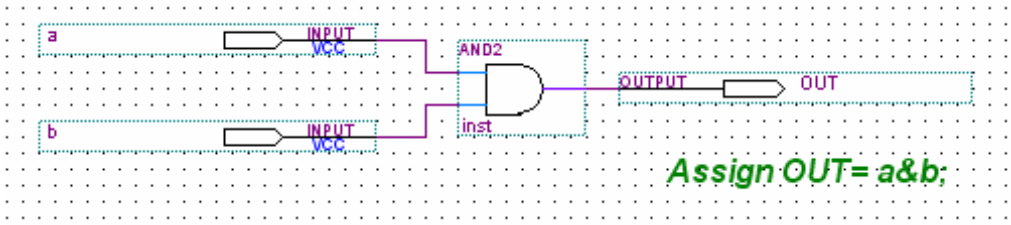

Figure 2 AND Gate Circuit Diagram

Our expected design is that signals a and b change at the same time. In the design, the output will always remain at 0. But the output generates the burr under actual conditions and its simulation waveform is shown as follows:

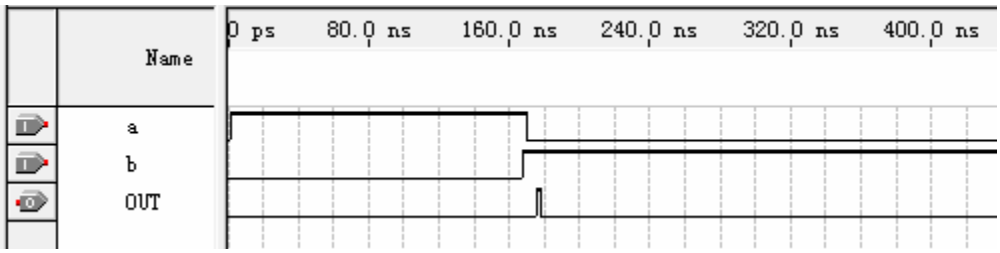

Figure 3 Simulation Waveform Diagram

Even if multi-way signals change in the simplest logical operations at the same time, the burr will certainly occur after internal routing. Signals of existing digital circuit design are generally under the control of clock. If the output signal with a burr is directly connected to clock input end, reset or setting port, it may cause serious consequences; for complex operation systems of multi-data input, the data is comprised of considerable bits. In such a case, the burr at each stage will seriously affect the results. If multilevel design is adopted, the burr will affect reliability and accuracy of the whole design after accumulation.

For determining whether a logic circuit will generate burrs when some input signals change, first, determine whether signals will change at the same time, and then determine whether a burr will be generated when signals change at the same time. It can adopt Karnaugh map of logic function or logic function expression for judgment.

(3) Elimination of burrs

The burr is a difficult problem of digital circuit design. It can affect stability and reliability of circuit operation and even cause mal-operation and logical breakdown of the total digital system.

The burr can be eliminated by the following methods:

First: output signals to D trigger. It is a traditional burr elimination method. The principle is that a D trigger reads the signal with burr to eliminate the burr (the D trigger is not sensitive to the input signal). Under actual conditions, it is very significant for the simple logic circuit, especially for the burr signal generated on non-clock clock jump edge. If the burr signal is generated on the clock signal jump edge, D trigger effect will not so significant (future output q of D trigger still has burrs). In addition, D trigger operation also will cause a certain delay. Especially when system levels are more, the delay will increase. Therefore, burr elimination depends on specific conditions. Not all burrs can be eliminated by the D trigger.

Second, signal synchronization method. For design of digital circuits, the synchronous circuit can greatly reduce burrs. Most burrs are shorter (about several nanoseconds). Burr signals will not harm the system only if no burr occurs on the clock jump edge. Therefore, the system synchronization can be realized only if the total system adopts the same clock. But the clock signal transmission in the FPGA device has a delay. For the reason, accurate position of the clock judge edge cannot be predicted. In other words, we cannot guarantee that the data read on the judge edge of a clock is stable. It is more prominent especially in the multilevel design. Therefore, it is critical for burr elimination to realize real synchronization. The key of synchronization is to ensure the data read on the clock judge edge is stable without burr. In the following are two specific signal synchronization methods: ① signal delay synchronization method. A delay link is added in the two-level signal transmission to ensure the data read in the next module is stable without burr. The abovementioned signal delay can be data signal delay or clock signal delay. ② State machine control. The state machine also can perform signal synchronization and burr elimination. In the multi-module system subject to complex data transmission, the state machine can send a clock signal or module enable signal at the specific time to control a specific module. State machine cycle control can enable coordinated operation of the entire system and reduce burr signals. Thus, we can avoid the race and hazard by processing trigger time of the state machine in order to inhibit burr generation.

Metastable state

In the synchronous circuit or asynchronous circuit, the metastable state may occur if failing in meeting setup time or hold time of the trigger. In such a case, trigger output end Q remains in an

uncertain state in a long time after effective clock edge. During the time, the Q end will generate burrs, be continuously oscillated and finally fixed at some voltage value, and the voltage value is not always equal to original data input end D value. The time is called resolution time. Q end will be stabilized at 0 or 1 after the resolution time. But 0 or 1 is random, without necessary relation with the input.

Metastable state harm is mainly the damage to system stability. The output may have burrs, oscillate or be fixed at some voltage value before stabilization, so the metastable state can cause logic misjudgment and even make the next level generate the metastable state (that is, the metastable state is spread) by output of medium voltage value between 0 and 1 under serious conditions. The logic misjudgment will result in function errors while the metastable state spread can expand the fault and even cause system crash.

The asynchronous sequential circuit is more prone to the metastable state, so the asynchronous circuit generally has multiple clock domains. Data transmission between two clock domains easily results in the metastable state due to failure in meeting the setup time or hold time. In the synchronous sequential circuit, excessive combinational logic delay between two triggers may result in the metastable state due to failure in meeting the setup time.

Test Technology of Communication Device Software

Like ordinary software testing, software testing plays a role of verification and confirmation at the design and development stage of FPGA software in the communication device. But the difference is that, output result of each integrated link in the FPGA is a form expression method of design, it can be verified by static method and each integrated link has dynamic simulation by means of simulation tools at the development stage, thus facilitating early identification of defects and reduction of defect elimination cost. In the entire system test stage of communication device, gate-level, unit-level, element-level and system-level tests are conducted successively from bottom to top under real conditions to respectively verify the gate-level model, RTL model, and behavioral model and system requirements.

Different from traditional software tests, black box test, white box test and grey box test are used widely for the FPGA software under the support of embedded logic analysis software. Major FPGA software development tools provide an embedded logic analyzer to enable the grey box test. The embedded logic analyzer can output prefabricated internal system signals via the interface in real time by using redundant chip resources under real operating conditions of FPGA. Thus, it can monitor internal signals, variables and state of the software under real operating conditions and avoid code defect exposure failure caused by input & output domain information loss and fault - tolerant link in order to improve FPGA code testability.

Summary

The FPGA software can perform a wide range of functions. It can almost realize all digital circuit functions. Application and development of the FPGA software simplify circuit design, reduce costs, improve system reliability and greatly change design of digital circuit systems. Many industries in China still need to import FPGA chips, which seriously restricts development of the digital circuit technology in China. Through analysis and research on design method and test technology of FPGA software, the design and testing methods suitable for different FPGA software

are explored and a reliable test platform is built to provide supports for improvement of China's localized FPGA products.

References

[1] Embedded Communication Software Design, Wang Ansheng, Beijing University of Aeronautics and Astronautics Press, 2004;

[2] FPGA Realization of Digital Signal Processing, Liu Ling, Tsinghua University Press, 2012;

[3] FPGA Design, He Chun, China Machine Press, 2009.