KHARKOV NATIONAL UNIVERSITY OF RADIOELECTRONICS

Proceedings of IEEE East-West Design & Test Symposium (EWDTS'08)

SPONSORED BY

IEEE Computer Society Test Technology Technical Council

Lviv, Ukraine, October 9 – 12, 2008

CONTENTS

IEEE EAST-WEST DESIGN AND TEST SYMPOSIUM 2008
ORGANISING COMMITTEE

General Chairs
V. Hahanov – Ukraine
Y. Zorian – USA

General Vice-Chairs
M. Karavay – Russia
R. Ubar – Estonia

Program Chairs
S. Shoukourian – Armenia
D. Speranskiy – Russia

Program Vice-Chairs
M. Renovell - France
Z. Navabi - Iran

Publicity Chairs
C. Landrault - France
S. Mosin – Russia

Program Committee
E. J. Aas – Norway
J. Abraham – USA
A. Barkalov – Poland
R. Bazylevych – Ukraine
A. Drozd – Ukraine
E. Evdokimov – Ukraine
A. Chaterjee – USA
E. Gramatova – Slovakia
S. Hellebrand – Germany
A. Ivanov – Canada
V. Kharchenko – Ukraine
K. Kuchukjan – Armenia
A. Matrosova – Russia
V. Melikyan - Armenia
O. Novak – Czech Republic
A. Orailoglu – USA
Z. Peng – Sweden
A. Petrenko – Ukraine
P. Prinetto – Italy
J. Raik – Estonia
A. Romankevich – Ukraine
A. Ryjov – Russia
R. Seinauskas – Lithuania
S. Sharshunov – Russia
A. Singh – USA
J. Skobtsov – Ukraine
A. Stempkovsky – Russia
V. Tverdokhlebov – Russia
V. Vardanian - Armenia
V. Yarmolik – Byelorussia
A. Yessayan – Armenia

Steering Committee
M. Bondarenko – Ukraine
V. Hahanov – Ukraine
R. Ubar – Estonia
Y. Zorian – USA

Organizing Committee
S. Chumachenko – Ukraine
M. Kaminska – Ukraine
N. Kulbakova – Ukraine
M. Lobur – Ukraine
V. Obrizan – Ukraine
T. Sviridova – Ukraine

EWDTS CONTACT INFORMATION

Prof. Vladimir Hahanov
Design Automation Department
Kharkov National University of Radio Electronics,
14 Lenin ave,
Kharkov, 61166, Ukraine.

Tel.: +380 (57)-702-13-26
E-mail: hahanov@kture.kharkov.ua
Web: www.ewdtest.com/conf/

Verification Challenges of Clock Domain Crossings

D. Melnik, S. Zaychenko, O. Lukashenko
Computer Engineering Faculty, Kharkov National University of Radioelectronics, Lenin Ave. 14, Kharkov, Ukraine, 61166, phone: (057) 70-21-421, (057) 70-21-326
E-mail: hahanov@kture.kharkov.ua

Abstract

This paper discusses typical verification problems occurring within SoC design cycle when multiple clock domains are involved. Critical cases leading to unpredictable SoC behavior during data transfer across clock domains are identified and described. A principle for metastability modeling is suggested.

1. Introduction

Only the most elementary logic circuits use a single clock. Today's system-on-chips (SoC) have dozens of asynchronous clocks. There are a lot of software programs to assist in creating of multimillion-gate ASIC/FPGA circuits, but designer still has to know reliable design techniques to reduce the risk of CDC-related design re-spins. Moreover, the most relevant literature does not cover CDC-related issues and approaches to prevent appropriate costly silicon bugs.

2. SoC Memory Diagnosis and Repair

A clock domain is defined as that part of the design driven by either a single clock or clocks that have constant phase relationships. A clock and its inverted clock or its derived divide-by-2 clocks are considered a clock domain (synchronous). Conversely, domains that have clocks with variable phase and time relationships are considered different clock domains [1].

The sections of logic elements that are driven by different clocks are called different clock domains. The signals that interface between these asynchronous clock domains are called the clock domain crossing paths. The first step in managing multiclock designs is to understand the problem of signal stability: when a signal crosses a clock domain, it appears to the circuitry in the new clock domain as an asynchronous signal. The circuit that receives this signal needs to synchronize it. Synchronization prevents the metastable state of the first storage element (flip-flop) in the new clock domain from propagating through the circuit [2].

Each type of flip-flop in an ASIC or FPGA library has timing requirements to determine «the window of vulnerability». Setup time describes the time an input signal to a flip-flop must be stable before the clock edge. Hold time is the time the signal must remain stable after the clock edge [2]. Metastability happens in silicon when setup or hold constraints are violated. This is unavoidable in designs with asynchronous clocks. When a register input changes within the setup or hold times of its clock edge, the register may become metastable and settles unpredictably to 0 or 1 over time.

3. CDC Signals Synchronization

In order to prevent propagation of metastable state into downstream logic, designers use specific synchronization circuits to connect asynchronous domains. A basic synchronizer comprises two flip-flops in series without any combinational circuitry between them (a synchronized signal is valid in the new clock domain after two clock edges). This design ensures that the first flip-flop exits its metastable state and its output settles before the second flip-flop samples it. For proper work of synchronization, the CDC signal crossing should pass from a flip-flop in the original clock domain to the first flip-flop of the synchronizer without passing through any combinational logic between the domains (Fig 1).

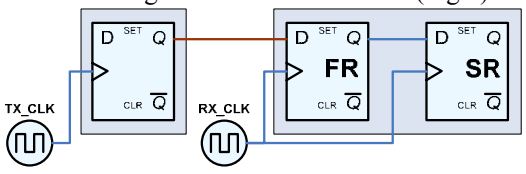

Fig. 1 – Level synchronizer

Combinatorial circuitry should be highly avoided between the domains, because the first stage of a synchronizer is sensitive to its glitches (when glitch meets the setup-and-hold requirements of the first flip-

flop in the synchronizer, it passes a pseudo-valid value to the rest of the logic in the new clock domain.

There are many synchronizer types. Synchronizers fall into one of three basic categories: level, edge-detecting, and pulse [2].

Level synchronizer. In a level synchronizer, the signal crossing a clock domain stays high and stays low for more than two clock cycles in the new clock domain. A requirement of this circuit is that the signal needs to change to its invalid state before it can become valid again. Each time the signal goes valid, the receiving logic considers it a single event, no matter how long the signal remains valid. This circuit is the heart of all other synchronizers.

Edge synchronizer. The edge-detecting synchronizer circuit adds a flip-flop to the output of the level synchronizer (Fig 2).

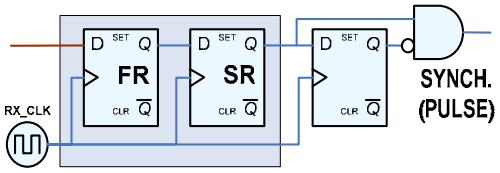

Fig. 2 – Edge synchronizer

This circuit detects the rising edge of the input to the synchronizer and generates a clockwide, active-high pulse (changing the AND gate to a NAND gate results in a circuit that generates an active- low pulse). The edge-detecting synchronizer works well at synchronizing a pulse going to a faster clock domain. This circuit produces a pulse that indicates the rising or falling edge of the input signal. One restriction of this synchronizer is that the width of the input pulse must be greater than the period of the synchronizer clock plus the required hold time of the first synchronizer flip-flop. The safest pulse width is twice the synchronizer clock period. This synchronizer does not work if the input is a single clockwide pulse entering a slower clock domain; however, the pulse synchronizer solves this problem.

Pulse synchronizer. The input signal of a pulse synchronizer is a single clockwide pulse that triggers a toggle circuit in the originating clock domain (Fig 3).

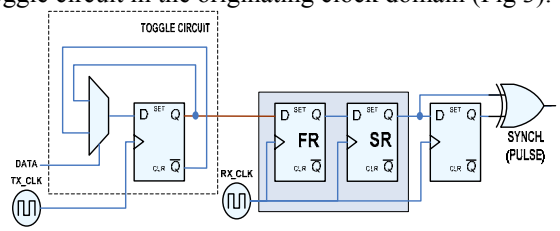

Fig. 3 – Pulse synchronizer

The basic function of a pulse synchronizer is to take a single clockwide pulse from one clock domain and create a single clockwide pulse in the new domain. The restriction of a pulse synchronizer: input pulses must have a minimum spacing between pulses equal to two synchronizer clock periods. If the input pulses are closer, the output pulses in the new clock domain are adjacent to each other, resulting in an output pulse that is wider than one clock cycle. This problem is more severe when the clock period of input pulse is greater than twice the synchronizer clock period. In this case, if the input pulses are too close, the synchronizer does not detect every one.

4. Critical CDC Bugs

Enable assertion. Designers commonly use the 2-DFF technique (level synchronizer) for a control signal's domain crossing path. Typical actions to transfer the data across clock domains:

1. set up the data in the source domain;
2. send a control signal to the destination domain – to enable data capture.

Such data-transfer technique is common and proven, but it relies on data to be stable when an enable is asserted. Therefore, this technique involves pitfalls that require special attention: the data transfer may be corrupted if having too low a margin between the data setting up and asserting the enable. A good way to prevent such problems is to design a full handshake when the data is set up (this approach might add a few cycles of latency, but it avoids functional failures).

Glitches. Glitches are other bug sources. Typically, any combinational logic may be the source of glitches. With synchronous transfers, these issues are generally harmless (they resolve themselves when the next clock edge is activated). But when a destination clock is activated, the design may receive a glitch as a pulse, which will cause a functional failure in a downstream logic. Because of this reason, it is a rule of thumb to avoid using any logic on CDC paths (except recirculation-multiplexer logic, which is the part of the enable flip-flop, – see the Fig 4) [3].

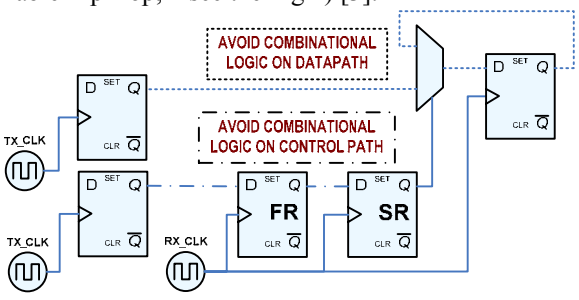

Fig. 4 – Avoid logic at CDC paths

Any computation should be performed just before crossing clock domains or just after the destination domain captures the signals.

Data coherency. Coherency problem occurs due to convergence of independently synchronized signals. When two correlated and separately synchronized signals cross the clock domains, each synchronizer introduces a different latency factor. If one of the signals captures a transition, metastability settles to the correct value in the first cycle, whereas the other signal captures a transition in the next cycle [A]. It makes to wait for the next clock cycle to capture the transition (the incorrect values are observed at the destination domain for at least one cycle – they could cause a functional failure of the design). In order to avoid a data coherency problem, correlated signals should be used so that they change values at different times. Gray encoding is used to correlate the CDC signals (Gray code changes only one bit at a time for each increase or decrease in the count. So, signals are Gray-encoded and registered before crossing clock domains. Still, glitch in the encoder could cause a functional failure.

5. Metastability modeling

Digital simulation is based on behavioral model of circuit – it predicts how the hardware design will behave. A digital simulation is founded on the principle that the hardware design does not violate the setup-hold constraints specified for clocked elements. If setup-hold constraints are adhered with given clock frequency then the simulation results is valid. While a CDC synchronizer circuit prevents metastable values from propagating to the downstream logic (in the receiving domain), synchronizers do not prevent metastability. Therefore, the designer should verify that logic in the receiving domain funtions as intended – even in case when the synchronizer impacts the receiving domain with non-deterministic delays.

When the first register (FR) in the synchronizer goes metastable, it will non-deterministically settle to 0 or 1. Compared with simulation, it non-deterministically exhibits either an extra-cycle delay or a bleed through [4], which the second register (SR) simply propagates to downstream logic:

- *Extra cycle delay*. Signal at FR data input changes to a logic 1 just *before* the clock edge of RX_CLK and violates the setup time for FR. In simulation, therefore, FR captures a 1, while in silicon FR has a high probability of going metastable and, therefore, will unpredictably settle to either 1 or 0. If FR settles to a 1, SR will transition at the same cycle as simulation predicts. However, if FR settles to 0, SR's transition will be delayed by one extra cycle. Hence the simulation predicts the value change will happen one cycle too soon.
- *Bleed through*. Signal at FR data input changes to a logic 1 just *after* the clock edge of RX_CLK and violates the hold time for FR. In RTL simulation, therefore, FR changes to a 1 at the following clock edge, while in silicon FR has a high probability of going metastable, unpredictably settling to 1 or 0. If FR settles to a 0, SR will transition at the same cycle as simulation predicts. However, if FR settles to 1, SR will transition one cycle earlier. Hence the simulation predicts the value change one cycle too late.

The modeling of the extra-cycle and bleed-through delay effects of metastability in simulation is required to ensure that the simulation is a truthful representation of the silicon behavior. The following methods are used to model metastability effects: clock jittering; 3-DFF synchronizer; propagation delay; mixed.

The following are requirements to such methods:

1. the conditions for an extra-cycle delay are present (violation of the setup constraint) => simulation should add a cycle-delay;
2. the conditions for a bleed-through delay are present (violation of the hold constraint) => simulation should subtract a cycle-delay;
3. every CDC signal should be modeled independently (otherwise bugs will be missed).

6. Conclusion

Identifying potential sources of CDC-related hardware failures on early design phases is a critical verification step in the multi-clock SoC desing cycle. Advanced metastability effect modeling is needed within EDA tools in order to detect, prevent, and eliminate CDC defects before SoC manufacturing.

7. References

[1] *Clock domain crossing: closing the loop on clock domain functional verification problems*, Cadence Inc. Technical paper, 2004.

[2] Mike Stein, *Crossing the abyss: asynchronous signals in a synchronous world*, EDN magazine, July 2003.

[3] S. Hamdioui, G. N. Gaydadjiev, A. J. Van de Goor. The State-of-the-art and Future Trends in Testing Embedded Memories, *Records IEEE International Workshop on Memory Technology, Design and Testing*, San Jose, CA, August 2004, pp. 54-59.

[4] Chris Kwok, Roger Sabbagh, *A comparison of metastability modeling methods*, Mentor Graphics Corp., Advanced verification white paper.

Camera-ready was prepared in Kharkov National University of Radio Electronics
by Dr. Svetlana Chumachenko and Volodymyr Obrizan
Lenin ave, 14, KNURE, Kharkov, 61166, Ukraine

Approved for publication: 20.09.2008. Format 60×841/8.
Relative printer's sheets: . Circulation: 150 copies.
Published by SPD FL Stepanov V.V.
Ukraine, 61168, Kharkov, Ak. Pavlova st., 311

Матеріали симпозіуму «Схід-Захід Проектування та Діагностування – 2008»
Макет підготовлено у Харківському національному університеті радіоелектроніки
Редактори: Світлана Чумаченко та Володимир Обрізан
Пр. Леніна, 14, ХНУРЕ, Харків, 61166, Україна

Підписано до публікації: 20.09.2008. Формат 60×84$^{1}/_{8}$.
Умов. друк. арк. . Тираж: 150 прим.
Видано: СПД ФЛ Степанов В.В.
Вул. Ак. Павлова, 311, Харків, 61168, Україна