

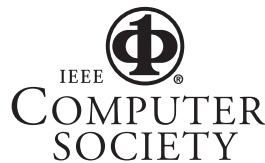
KHARKOV NATIONAL UNIVERSITY OF RADIOELECTRONICS

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

Copyright © 2008 by The Institute of Electrical and Electronics Engineers, Inc.

SPONSORED BY

IEEE Computer Society Test Technology Technical Council



Lviv, Ukraine, October 9 – 12, 2008

CONTENTS

A Systematic Approach for Evaluating Satellite Communications Systems Stefano Di Carlo, Paolo Prinetto, Alessandro Savino, Gabriele Tiotto, Paola Elia	13
Facilitating Testability of TLM FIFO: SystemC Implementations Homa Alemzadeh, Marco Cimei, Paolo Prinetto, Zainalabedin Navabi	17
A Model for Resistive Open Recursivity in CMOS Random Logic M. Renovell, M. Comte, N. Houarche, I. Polian, P. Engelke, B. Becker	21
An Optimized CLP-based Technique for Generating Propagation Sequences F. Fummi, V. Guarnieri, C. Marconcini, G. Pravadelli	25
Validation of a Mixed-Signal Board ATPG Method Val'erie-Anne Nicolas, Bertrand Gilles, Laurent Nana	29
A Low-Cost Optimal Time SICP air Generator I. Voyiatzis, H. Antonopoulou	35
Selected Cost Factors in Modeling and Testing Hardware and Semiconductor Defects by Dynamic Discrete Event Simulation Jack H. Arabian	41
HotSpot : Visualising Dynamic Power Consumption in RTL Designs T. English, K.L. Man, E. Popovici and M.P. Schellekens	45
Characterization of CMOS Sequential Standard Cells for Defect Based Voltage Testing A. Wielgus and W. A. Pleskacz	49
Testing the Control Part of Peripheral Interfaces S. Zielski, J. Sosnowski	55
Concurrent Processes Synchronisation in Statecharts for FPGA implementation Grzegorz Łabiak, Marian Adamski	59
Parallel Fault Simulation on Multi-core Processors Dmitry E. Ivanov	65
Synthesis of control unit with code sharing and chain modifications Alexander Barkalov, Larysa Titarenko, Jacek Bieganowski	68
FSMs Implementation into FPGAs with Multiple Encoding of States Arkadiusz Bukowiec, Alexander Barkalov and Larysa Titarenko	72
Reduction in the number of PAL macrocells for the effective Moore FSM implementation A. Barkalov, L. Titarenko, S. Chmielewski	76
Partial Reconfiguration of Compositional Microprogram Control Units implemented on an FPGA R. Wisniewski, Alexander A. Barkalov, Larysa Titarenko	80

Coverage-Directed Verification of Microprocessor Units Based on Cycle-Accurate Contract Specifications Alexander Kamkin	84
Code-Probability Entities for Constrained-Random Verification Diana Bodyan, Ghennady Bodyan	88
Multidimensional Loop Fusion for Low-Power Dmytro Lazorenko	92
A Synthesis of Common Models of Finite State Machines Using Input and Output Registers of Programmable Logic Devices Adam Klimowicz, Valeri Soloviev	96
An advanced Method for Synthesizing TLM2-based Interfaces Nadereh Hatami, Zainalabedin Navabi	104
Testing Combinational QCA Circuits Mehdi Azimipour	108
Dependability and Complexity Analysis of Inter-channel Connection Schemes for “N out of M” System-on-Chip Vyacheslav Kharchenko, Vladimir Sklyar, Georgiy Chertkov, Yuriy Alexeev, Ladislav Novy	113
Safety-Critical Software Independent Verification Based on Measurement of Invariants during Static Analysis Sergiyenko Volodymyr, Zavolodko Valeriy	117
Designing High Productivity Parallel Algorithms with Algebraic and Heuristic Programming Techniques Anatoliy Doroshenko, Mykola Kotyuk, Sergiy Nikolayev, Olena Yatsenko	121
Multiple Run Memory Testing for PSF Detection I. Mrozek , V.N. Yarmolik, E. Buslowska	125
The analysis of the start up control parameters of the asynchronous electric traction motors Gabriel Popa, Razvan A. Oprea, Sorin Arsene	131
A novel timing-driven placement algorithm using smooth timing analysis Andrey Ayupov, Leonid Kraginskiy	137
Digital Lock Detector for PLL Vazgen Melikyan, Aristakes Hovsepyan, Mkrtych Ishkhanyan, Tigran Hakobyan	141
Diagnosis of SoC Memory Faulty Cells for Embedded Repair Vladimir Hahanov, Eugenia Litvinova, Karina Krasnoyaruzhskaya, Sergey Galagan	143
Testing Challenges of SOC Hardware-Software Components Vladimir Hahanov, Volodimir Obrizan, Sergey Miroshnichenko, Alexander Gorobets	149

SoC Software Components Diagnosis Technology Svetlana Chumachenko, Wajeb Gharibi, Anna Hahanova, Aleksey Sushanov	155
Vector-Logical Diagnosis Method for SOC Functionalities Vladimir Hahanov, Olesya Guz, Natalya Kulbakova, Maxim Davydov	159
Testability analysis method for hardware and software based on assertion libraries Maryna Kaminska, Roman Prihodchenko, Artem Kubirya, Pavel Mocar	163
Different observation time strategies of outputs in diagnostics of sequential digital circuits Yu. A. Skobtsov, V. Yu. Skobtsov	168
Design and Implementation of a Parallel Adaptive Filter Using PBS-LMS Algorithm in a Convex Structure Ali Fathiyan, M. Eshghi	173
An IEEE 1500 Compatible Wrapper Architecture for Testing Cores at Transaction Level Fatemeh Refan, Paolo Prinetto, Zainalabedin Navabi	178
Power-Aware Embedded Software Design Fabian Vargas, Cláudia A. Rocha, Luís Fernando Cristófoli, Luciano Rocha	182
System Level Hardware Design and Simulation with System Ada Negin Mahani, Parnian Mokri, Zainalabedin Navabi	190
Automating Hardware/Software Partitioning Using Dependency Graph Somayyeh Malekshahi, Mahshid Sedghi, Zainalabedin Navabi	196
Reliable NoC Architecture Utilizing a Robust Rerouting Algorithm Armin Alaghi, Mahshid Sedghi, Naghmeh Karimi, Mahmood Fathy, Zainalabedin Navabi	200
Method for Modeling and Fault Simulation using Volterra kernels Pavlenko V., Fomin O.	204
Parity Prediction Method For On-Line Testing a Barrel-Shifter Drozd A., Antoshchuk S., Rucinski A., Martinuk A	208
RTL-TLM Equivalence Checking Based on Simulation Nicola Bombieri, Franco Fummi, Graziano Pravadelli	214
Estimation of the FPAA specification with use of the Artificial Neural Network Damian Grzechca, Tomasz Golonek	219
TUFFAN: A TLM Framework for Fast Architecture Exploration of Digital Systems Sheis Abolmaali, Parisa Razaghi and Zainalabedin Navabi	223
Code Optimization for Enhancing SystemC Simulation Time Homa Alemzadeh, Soheil Aminzadeh, Reihaneh Saberi, Zainalabedin Navabi	227

Automatic Test Pattern Generation Algorithm for Bridging Faults in Sequential Circuits F. Podyablonsky, N. Kascheev	231
Test Suite Consistency Verification Sergiy Boroday, Alexandre Petrenko, Andreas Ulrich	235
A 403-MHz Fully Differential Class-E Amplifier in 0.35 μ m CMOS for ISM Band Applications Ghulam Mehdi, Naveed Ahsan, Amjad Altaf, Amir Eghbali	239
Signal Processing Verification System for the Programmable Digital Matched Filter Kharchenko H.V., Makovetskiy S.O., Tkalich I.O., Tsopa O.I., Vdovychenko Y.I	243
Building a Research Framework for Integrated Circuit Physical Design Andrey Kamaev, Kirill Korniyakov, Iosif Meyerov, Alexey Sidnev, Artem Zhivoderov	251
A High-speed and High Precision IDDQ Measurement for Consumer and Communication SoCs Yoshihiro Hashimoto, Yasuo Furukawa, Nguyen Ngoc Mai Khanh	254
Creating Test Environment for Consumer Video Devices Andrew Johnson, Oleksandr Yegorov	258
An Efficient Inner (De)Interleaver Architecture for DVB-T systems Mojtaba Rezayi, Mohammad Eshghi, and Hamid Reza Tanhaei	259
Redundant tests optimization Dmitriy Speranskiy, Ekaterina Ukolova	263
Sensor Web and Grid Technologies for Flood Applications N. Kussul, A. Shelestov, S. Skakun, Yu. Gripich	267
Persian Digit Recognition by Fourier Coefficients and Neural Networks Nasim Kazemifard, Pedram Azimi, Saeed Mozaffari	271
Deterministic Distinguishing Tests for Given Fault of Discrete Device Synthesis Dmitriy Speranskiy, Ivan Ukolov	276
Digital Implementation of General Regression Neural Network for Function Approximation Applications Saber Moradi, Mahmoud Tabandeh, Nasser Sadati	281
Hardware Implementation of Exponential Function Using a Mathematical approach Saber Moradi, Mahmud Tabandeh, Nasser Sadati	285
Automated Generation of Register Transfer Graph for Processors Victor Belkin	289
One Approach to Fault Dictionary Size Reduction Sergey Mironov, Dmitriy Speranskiy	295
Software engineering for recognition of electronic elements on the circuit board Dmitry Bagayev, Pavel Khrustalev	301

Automatic Identification of Radiotelephone Transmissions in the Maritime Communication Aleksandr V. Shishkin	306
QCA Parallel Prefix Adder Design S. Arab, H. Aghababa, B. Forouzandeh	310
Simple march tests for PSF detection in RAM Ireneusz Mrozek, Eugenia Buslowska	314
Improved Digital Signature Protocols On Elliptic And Hyperelliptic Curves Dolgov V.I., Nelasa G.V.	320
Cascade Structural Encoding of Binary Arrays Vladimir Barannik, Anna Hahanova	322
Mapping DSP Algorithms into FPGA Oleg Maslennikov, Anatolij Sergiyenko, Tatyana Lesyk	325
Precision of FTMPs reliability evaluation based on statistical experiments Romankevych A., Romankevych V., Chernyavskaya K	331
Discrete model for dynamics analysis of the nonlinear oscillating systems with long transient processes and complicated nature Zayats Vasyl	332
Deriving test suites for timed Finite State Machines M. Gromov, D. Popov, N. Yevtushenko	339
Checker Design for Arbitrary Subset of Unordered Code Words A. Matrosova, A. Malgin, N. Butorina	346
Multiple Stuck-at Fault and Path Delay Fault Testable Circuits A. Matrosova, V. Andreeva, A. Melnikov, E. Nikolaeva	356
Minimizing Path Length in Digital Circuits Based on Equation Solving N.Kushik, G.Sapunkov, S.Prokopenko, N.Yevtushenko	365
Utilizing HDL Simulation Engines for Accelerating Design and Test Processes Najmeh Farajipour, S. Behdad Hosseini and Zainalabedin Navabi	371
Performance evaluation of In-Circuit Testing on QCA based circuits Nasim Kazemifard, Maryam Ebrahimpour, Mostafa Rahimi, Mohammad Tehrani, Keivan Navi	375
Partitioning, Floor planning and detailed placement and routing techniques for schematic generation of analog netlist Bikram Garg, Rajeev Sehgal, Ashish Agrawal, Amarpal Singh, Manish Khanna	379
Parallel computer emulator for digital devices modeling Alexander Chemeris, Svetlana Reznikova	383

The Oscillations of an Overhead Contact Line Due to the Pantograph Raising R.A. Oprea, G.C.Popa, S.Arsene	387
Reverse Semantic Quality Control Methods in Software Engineering Vladimir L. Pavlov, Anatoliy Doroshenko, Konstantin Zhereb, Olexii Kuchaev	393
The Interplay of Reliability and Power Consumption in Design of SEU-Tolerant Latches for DSM Technology M. Fazeli, S. G. Miremdi, A. Patooghy	399
Evaluation of a Concurrent Error Detection Technique Using Power Supply Disturbance Fault Injection M. Fazeli, A. Patooghy and S.G. Miremadi	405
Embodying of High Performance Computation in Matlab Parallel Computing Toolbox for Detection of Spread Spectrum Signals Bohdan Yavorsky	411
Implementation of Finite State Machines on the Basis of an Embedded Memory Block V. Chapenko, K. Boule	414
On Macroplaces in Petri Nets Andrei Karatkevich	418
Testing of hardware and software for FPGA-based critical systems Yuliya Prokhorova, Sergey Ostroumov, Vladimir Sklyar	423
Luxury Wallet – new generation of the SoC based consumer products Mikhail Lodygin	427
Descriptor Neural Networks And Singular Implicit Dynamic Systems Rutkas A.A	429
Tools of the Computer Testing of Knowledge in Mathematical Disciplines Shkil A.S., Naprasnsk S.V., Tsimbaluyk E.S., Garkusha E.V.	431
Software for problem components estimation in photometric stereo reconstruction Bohdan Rusyn, Yuriy Lysak, Oleksiy Lutsyk	434
Method of Digital Treatment of the Information Received by Space Diversity Radars Dmitriy Vasiliev	436
Verification Challenges of Clock Domain Crossings D. Melnik, S. Zaychenko, O. Lukashenko.....	438
AUTHORS INDEX	441

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).

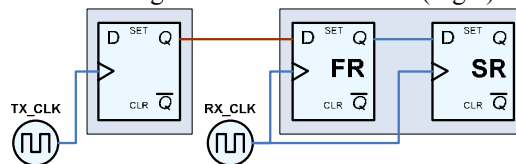


Fig. 1 – Level synchronizer

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).

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).

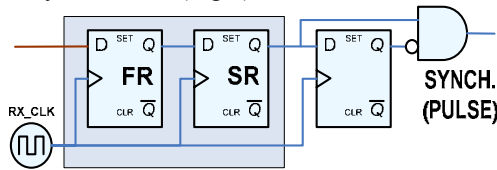


Fig. 2 – Edge synchronizer

This circuit detects the rising edge of the input to the synchronizer and generates a clockwise, 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 clockwise 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 clockwise pulse that triggers a toggle circuit in the originating clock domain (Fig 3).

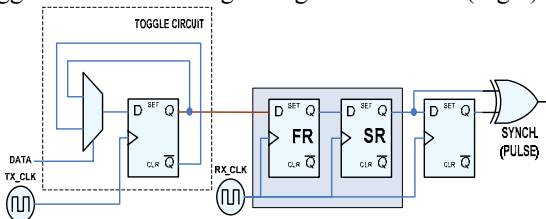


Fig. 3 – Pulse synchronizer

The basic function of a pulse synchronizer is to take a single clockwise pulse from one clock domain and create a single clockwise 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].

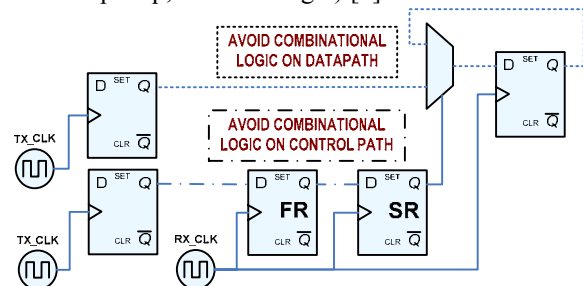


Fig. 4 – Avoid logic at CDC paths

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 functions 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 design 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 симпозиуму «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 60×84¹/₈.

Умов. друк. арк. . Тираж: 150 прим.

Видано: СПД ФЛ Степанов В.В.

Вул. Ак. Павлова, 311, Харків, 61168, Україна