

Structural Analysis Technique and Bad Synchronization Styles

Melnik D., Lukashenko O.

Abstract — This paper discusses early detection of potentially missing synchronizers on clock domain crossing paths, using structural static analysis.

Index Terms — Clock Domain Crossing, Metastability, Static Verification

I. INTRODUCTION

THE number of independent clock domains found on the typical today's device is continuously growing. According to the latest industry research, the average number of clock domains on a single device is >15—20 and it becomes higher and higher from day to day. The CDC-related design flaws are also growing exponentially, appearing to be very dangerous as the roots of intermittent chip failures (can be found only in the silicon). Static CDC verification is considered as one of the first de-facto steps in today's SoC design methodology; only static techniques can work as soon as the RTL starts taking shape [1].

The sections of logic elements that driven by clocks coming from different sources are called clock domains [2]. The signals that interface between asynchronous clock domains are called the clock domain crossing (CDC) signals (see Figure 1). The DATA_A signal is considered as an asynchronous signal into the receiving clock domain (no constant phase and time relationship exists between CLK_A and CLK_B).

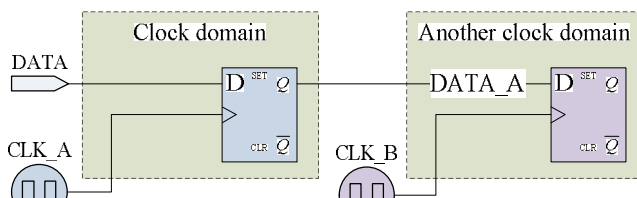


Fig. 1. Clock domains and CDC signal

The nature of CDC bugs is intermittent; it simply means that a test suite can be successfully completed on a chip in

the morning, but the same tests will complete with errors for the same chip in the afternoon [3]. Consider the simplest flip-flop example: such a flip-flop is located anywhere in the chip; the data signal for this flip-flop comes from the domain #A but the clock signal — from the domain #B... so whenever the setup or hold condition is violated, the flip-flop can go to one or to zero and it cannot be predicted (see Figure 2).

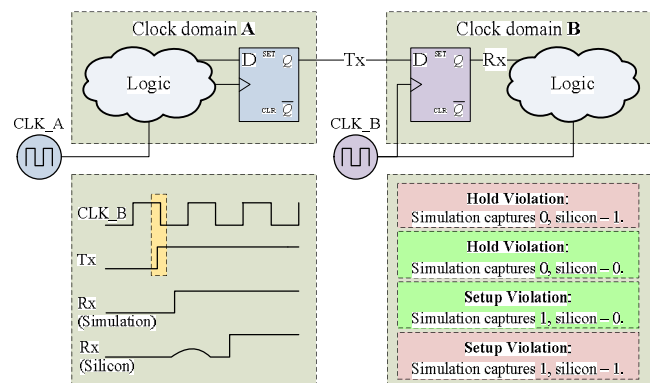


Fig. 2. Possible metastability effects

The metastability term is used to describe what happens in digital circuits when the clock and data inputs of a flip-flop change values at approximately the same time. As shown in the Figure 2, it leads to the flip-flop output oscillating and not settling to a value within the appropriate delay window [4]. Such glitches happen in every design wherein two or more discrete systems communicate (the number of clock domains is greater than two).

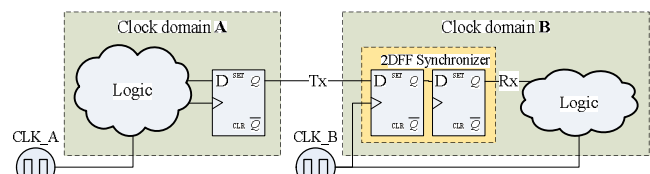


Fig. 3. Simplest synchronizer comprising 2 DFFs in series

Designers have actually found a solution to this and most of them is aware that metastability can be controlled using synchronizers on CDC signals (outputs of metastable registers are isolated so that the metastable value does not propagate to downstream logic) [5, 6]. Whenever there is a

Manuscript received November 9, 2009
 Melnik D. is with Kharkiv National University of Radioelectronic, Kharkiv, Lenin avenue, 14, Ukraine; (e-mail: melnyk.dima@gmail.com).
 Lukashenko O. is with Kharkiv National University of Radioelectronic, Kharkiv, Lenin avenue, 14, Ukraine; (e-mail: olga.lukashenko@gmail.com).

domain crossing signal, two flip-flops are placed one next to the other clocked by the same clock (see Figure 3). Such a synchronization structure decreases the MTBF (Formula 1, where the f_{clk} – clock frequency, f_{in} – input signal frequency, t_d – duration of critical time window) from hours to thousands years [4].

$$MTBF = \frac{1}{f_{clk} \times f_{in} \times t_d} \quad (1)$$

Formula (1) means (average) time between failures. Recent trends have been in favor of using static analysis tools [1]. But the biggest disadvantage of this approach is that it comes pretty late in the game — after the design has been synthesized, and the gate-level netlist is available (finding a CDC at this stage — which needs to be fixed — could set the design schedule totally off). So there is a need in static analysis tool that:

1. Performs lightweight synthesis (netlist synthesis emulation) directly from the RTL description — alongside with Verilog, VHDL or SystemVerilog compilation.
2. Reports domain crossing paths with potentially missing synchronizers, thus providing an obvious advantage in the form of early checking.

II. AUTOMATIC CLOCK DOMAINS EXTRACTION

A. Detect Global Clock in a Design

Clock domains extraction with further synchronizers detection is illustrated by the dataflow that is shown in the Figure 4. It involves several steps, starting with the compilation of the RTL description and creation of the database with netlist elements (lightweight synthesis), proceeding with special attributes assignment and their propagation through a design hierarchy (global clocks detection), and further manipulations with global clocks (clock domains look up).

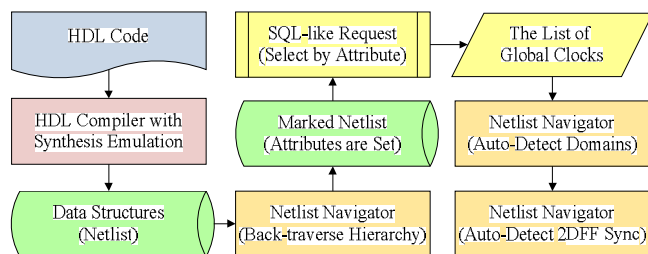


Fig. 4. Clock domains and synchronizers detection dataflow.

Attributes are distributed through the design hierarchy (netlist): “DESIGN_CLOCK” attribute is back-propagated from each flip-flop clock pin. Since all the netlist elements were added to the database, it further can be used for selection by the particular attribute(s) presence (SQL-like request).

- The back-propagation of the attribute is terminated on the storage elements (flip-flops and latches) and tri-states. While back-propagation is stopped, it means that the signal

which feeds the flip-flop clock pin is not an external input signal and thus it cannot be considered as a global clock.

- However, if the attribute reaches an external input pin (passes only through combinatorial logic, buffers and inverters), it is considered as a global clock – added to the list of global clocks (see Figure 5).

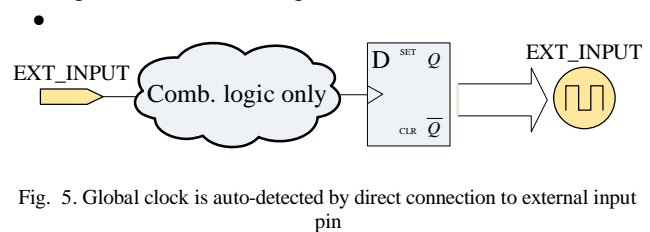


Fig. 5. Global clock is auto-detected by direct connection to external input pin

B. Extract Clock Domains

Clock domains can be detected when the list of global clocks is available; each global clock creates at least one separate clock domain. In order to detect clock domains, global clocks should be propagated through the design hierarchy (external input pins marked with the “DESIGN_CLOCK” attribute):

- Transparent logic. Combinatorial logic, latches and tri-states that happen on the attribute propagation path are considered as transparent objects.
- Flip-flops consideration. Each flip-flop that happens on the propagation path is added to the appropriate clock domain if “DESIGN_CLOCK” attribute reaches its clock input pin; if a flip-flop clock pin is driven by the output of another flip-flop which already belongs to a clock domain, the flip-flop is also added to the same clock domain (Figure 6)

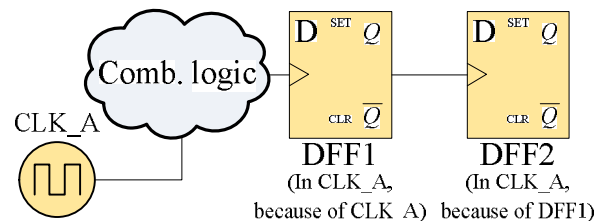


Fig. 6. Flip-flops are added to corresponding domains

- Derived domains. If two or more clock signals are propagated through the same combinatorial logic or multiplexer then the output of this logic or multiplexer derives a new clock signal that correspondingly results in a new clock domain for subsequent connections (see Figure 7). Also if a clock signal is connected to the multiplexer select pin then the output of this multiplexer derives a new clock signal.

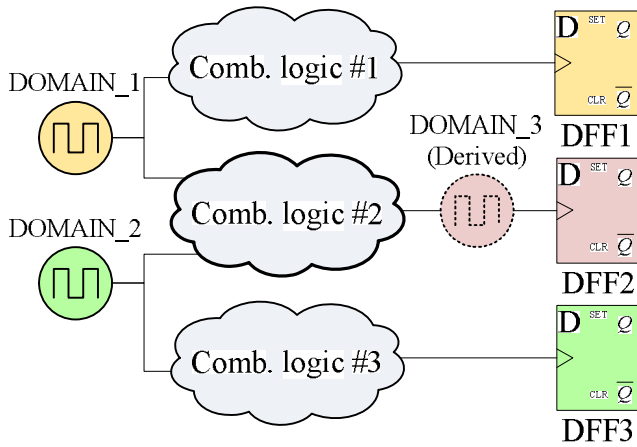


Fig. 7. Derived domains

Design is considered to be in a single clock domain if clock domains were not detected.

III. DETECTING POTENTIALLY NON-SYNCHRONIZED PATHS

While the netlist is marked with clock domain-related attributes, the data about each flip-flop membership is available, it becomes possible to go further and detect synchronized and potentially non-synchronized CDC paths.

A. Detecting missing 2DFF synchronizer

In order to be considered as 2DFF synchronizer, a pair of flip-flops should comply with the following restrictions:

1. Each flip-flop should receive the data only from the same clock domain (correct case – FF#1 receives data from domain A and transmits it to FF#2; incorrect case – FF#1 receives data from domain A, FF#2 receives data from domain B).

2. The outputs of the first and second flip-flops should not be connected to external design output(s) (in each case, the propagation should be blocked by non-clock input of another flip-flop(s) from the same domain).

It should be noted, that for some very high speed designs, the MTBF of a two-flop synchronizer is too short and a third flop is added to increase the MTBF to a satisfactory duration of time [7].

The paths which does not pass through a 2DFF synchronizer upon arrival into the new clock domain can be considered as potentially non-synchronized and reported as the design rule violations (synchronization errors class, see Figure 8).

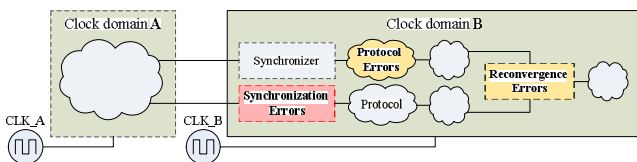


Fig. 8. Synchronization and other potential CDC errors

B. Detect hazardous transfers

But other incorrect design patterns should be detected when transmitting the data between asynchronous clock

domains – except the direct transfers (missing synchronizer).

A simple synchronizer comprises two flip-flops in series without any combinational circuitry between them. This approach 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 such synchronization, the signal crossing a clock domain should pass from flip-flop in the original clock domain to the first flip-flop of the synchronizer without passing through any combinational logic. First flip-flop of a synchronizer is sensitive to glitches that combinational logic produces (glitch that occurs at the correct time could meet the setup-and-hold requirements of the first flip-flop in the synchronizer, causing the synchronizer to pass a pseudo-valid signal to the rest of the logic in the target clock domain). Therefore, combination logic should not be located between asynchronous clock domains, because it significantly increases the risk to propagate pseudo-valid value to downstream logic.

The following pattern is forbidden:

1. Data input of the FF#1 in the receiving domain is fed by combinational logic output
2. Any of the logic inputs is(are) fed by data from FF of the transmitting domain.

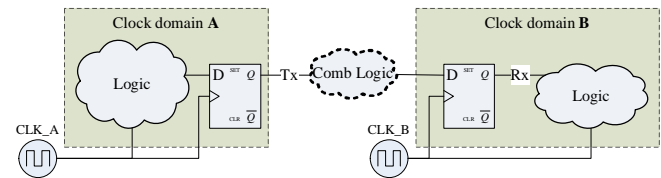


Fig. 9. Combinatorial logic between domains.

If combinational logic is located between two synchronizing flip-flops, the second flip-flop becomes sensitive to glitches produced by combinational logic – a possibility to propagate pseudo-valid value to downstream logic increases significantly and synchronizer could become useless in this case. Following conditions are characterized the situation:

1. FF#1 of the receiving domain feeds some combinational logic
2. Any of the logic outputs feeds receiving domain FF.

Combinational logic placed as described at the Figure 10 may also be treated as synchronizer that consists of one FF only. Above description explains that it is not enough to ensure metastability convergence.

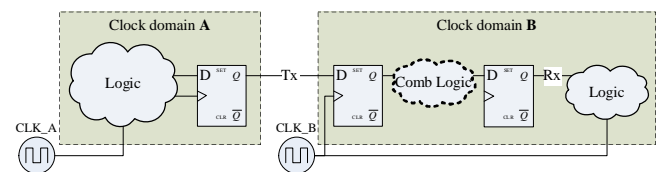


Fig. 10. Combination logic between after FF#1 in receiving domain.

C. Detect hazardous reset lines

Section 2.1 describes global clock signals detection. The same algorithm may be used to detect global reset signals using the netlist database. “DESIGN_RESET” attribute is back-propagated from each flip-flop reset pin.

- The back-propagation of the attribute is terminated on the storage elements (flip-flops and latches) and tri-states. While back-propagation is stopped, it means that the signal which feeds the flip-flop reset pin is not an external input signal and thus it cannot be considered as a global reset.
- However, if the attribute reaches an external input pin (passes only through combinatorial logic, buffers and inverters), it is considered as a global reset.

The global reset leading edge is safe because it set all the circuits to a known starting state. While reset trailing edge is not so harmless [8]. During the global reset all the clocks are started. But when the reset is removed it may happen simultaneously with the sampling edge of one of the clocks. Thus some FFs may enter metastable state. To prevent this situation synchronizer should be used for the global reset trailing edge. The proper synchronization circuit is shown at Figure 11. The leading edge is transferred directly and trailing edge is synchronized properly.

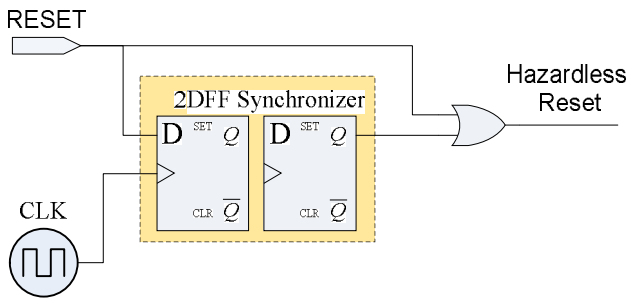


Fig. 11. Global reset synchronization

Internally generated asynchronous reset (set) signal may also be transferred from one clock domain to another. Thus it combines both situations described above. The signal can lead to asynchronous domains related problems as it crosses domain boundaries. On the other hand removal of the reset may coincide with receiving clock sampling edge and so also lead to metastability.

The both methods can be used to solve the problem.

- Asynchronous reset line is synchronized with 2DFF synchronizer
- Asynchronous reset trailing edge is synchronized (Figure 12)

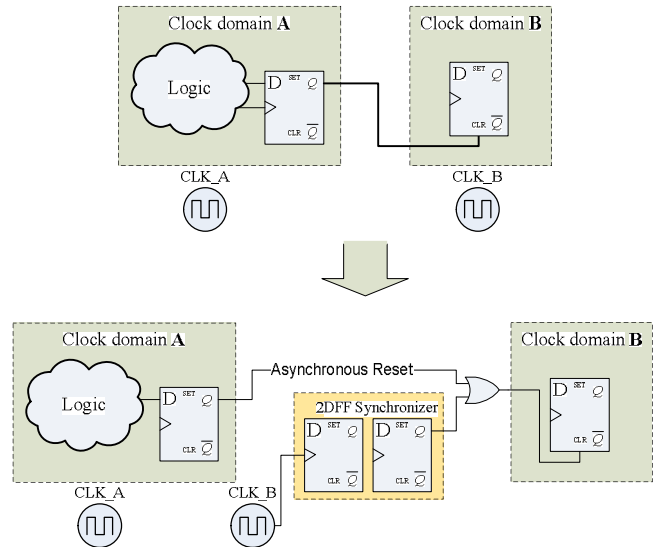


Fig. 12. Synchronized-trail asynchronous clear.

IV. CONCLUSION

Proposed structural analysis technique includes building of a netlist of the target design (lightweight synthesis is performed alongside with compilation) and performing further static analysis on this netlist. The novelty of the approach concerns propagation of various attributes through a design hierarchy: once the database with “netlist element”—“attribute(s)” relations is prepared, it can be used for SQL-like selections by attribute. The result of analysis is a summary of CDC paths in the design where the synchronization is potentially missing or incorrectly implemented (data and reset transfers). Proposed technique deals only with the first of CDC problems list which can be detected with static analysis [8]:

1. Missing and incorrectly implemented synchronizers.
2. Correctly implemented synchronizer.
3. Complex synchronizers that require protocol verification.
4. Potential reconvergence problems.

To perform more complete CDC verification, formal analysis techniques should be used alongside with structural analysis (see Figure 9).

During the structural verification stage, it is possible to generate monitors for CDC transfer protocols. At the stage of formal verification [9], a simple reset sequence is used and cycle-based design analysis is performed (requires knowledge about clock periods of the asynchronous clocks). While a monitor is proven, it means that CDC protocol is followed.

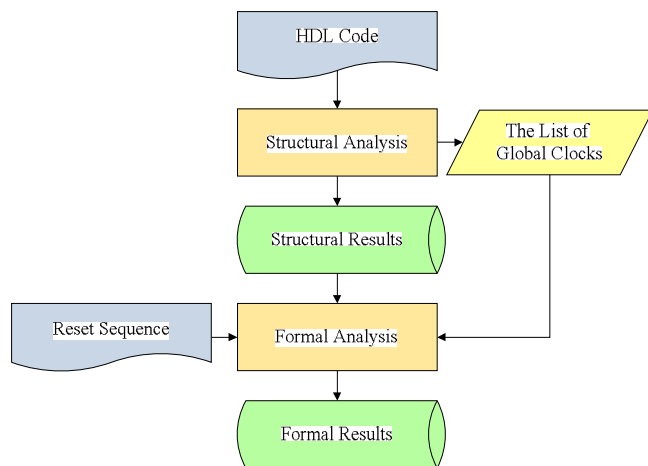


Fig. 13. Structural and formal analysis

REFERENCES

- [1] Sanjay Churiwala and Sapan Garg of Atrenta, and Chirag Gupta and Paresh Joshi of Texas Instruments, "Verification of Clock Domain Crossing in SoCs: Part One — Tools and Needs". Downloadable from www.chipdesignmag.com.
- [2] Cadence technical paper, "Clock Domain Crossing. Closing the Loop on Clock Domain Functional Implementation Problems". Downloadable from w2.cadence.com/whitepapers/cdc_wp.pdf.
- [3] Mentor Graphics technical webinar, "Finding and Eliminating CDC Errors with 0-In CDC Verification". Downloadable from www.mentor.com/player/2007/zero_in_cdc/index.html.

- [4] Michelle Lange of Mentor Graphics, "Automating Clock-Domain Crossing Verification for DO-254 (and other Safety-Critical) Designs". Downloadable from www.do254.com/documents/Papers/Mentor_CDC-for-DO254.pdf.
- [5] Clifford E. Cummings, SNUG-2001, "Synthesis and Scripting Techniques for Designing Multi-Asynchronous Clock Designs". Downloadable from www.sunburst-design.com/papers.
- [6] Tai Ly, "The Need for an Automated Clock Domain Crossing Verification Solution". Downloadable from www.mentor.com/fv.
- [7] Clifford E. Cummings, SNUG-2008, "Clock Domain Crossing (CDC) Design & Verification Techniques Using SystemVerilog". Downloadable from www.sunburst-design.com/papers.
- [8] Ran Ginosar, "Fourteen Ways to Fool Your Synchronizer", Proceedings of the Ninth International Symposium on Asynchronous Circuits and Systems (ASYNC'03)
- [9] Ping Yeung of Mentor Graphics, "Five Steps to Quality CDC Verification". Downloadable from www.mentor.com/products/fv/techpubs/.
- [10] Mentor Graphics, "Formal Verification User Guide V2.5". Feb 2007.

Melnik Dmitry, Ph.D student of Design Automation department, KNURE. Scientific interests: multiple clock domains verification in complex hardware designs, safety-critical systems design and verification. Address: KNURE, Kharkov, 61166 14 Lenin Avenue.

Lukashenko Olga, Ph.D student of Design Automation department, KNURE. Scientific interests: functional verification, multiple clock domains verification, coverage-driven verification. Address: KNURE, Kharkov, 61166 14 Lenin Avenue.