

CLOCK DOMAIN CROSSING CHALLENGES IN LATCH BASED DESIGNS

Madan Das, PhD¹ Chris Kwok² Kurt Takara³

¹Mentor- A Siemens Business, Fremont, United States (madan_das@mentor.com)

²Mentor- A Siemens Business, Fremont, United States (chris_kwok@mentor.com)

³Mentor- A Siemens Business, Fremont, United States (kurt_takara@mentor.com)

Introduction: Clock domain crossing (CDC) analysis for registers and memories are well understood problems [1] and there are many software tools to analyze the CDC issues associated with them. However, presence of latches in the designs can complicate CDC analysis as some of the latches may act as pass through combinatorial paths through which the input signal can continuously affect the output of the latch. Still, latches are a necessary part of high performance designs due to their advantages in time borrowing and ability to tolerate on chip variance. This paper describes the challenges in CDC analysis for latch based designs and a systematic approach to handle latches that are not enabled by clock signals. It also presents the results and insights of latch based crossings for several industrial scale designs.

Why do designs have latches?

Latches are bistable storage elements that are transparent when the enable signal is active [5]. Latches and registers are both sequential elements, as their outputs depend on their previous states. However, latches are different from flip flops as they are level sensitive instead of being edge triggered as in the case of registers.

Latches are more difficult to handle in static timing analysis [2] and testability [3] due to time borrowing and transparent operation in the enabled phase. But they are still used in high performance designs due to their faster timing, smaller area and lower power benefits. When latches are involved in CDC paths, it becomes even harder to understand the paths going through such latches and ensure proper functioning. Synthesis tools generally do not create latches because of the aforementioned difficulties. Appearance of latches in synthesized netlists often indicates incorrect RTL coding, such as an incomplete assignment [6]. When performing clock domain analysis, special attention must be paid to these latches to avoid metastability [3] [4].

Clock domain crossings involving latches

The latch enable is typically a clock, so that timing the paths involving the latch is easier. But it is not always the case, since the enable of the latch can simply be a sampling signal composed of other non-clock signals. If CDC analysis will treat the enables of latches as clocks, it could result in spurious clock signals and false CDC paths. For this reason, we cannot always infer that the signal connected to the enable pin of the latch is a clock. Such latches are generally called *unlocked latches*.

This is different from the derived clocks for registers, since the clock distribution network has to be glitch free and is designed for low skew, low slew, and minimal jitter. So it is quite atypical to see arbitrary logic in the clocking network. Typical logic encountered in clock distribution networks are dividers, multipliers (PLLs), buffers, clock muxes and clock gating cells. For example, combining two clocks derived from the same clock can result in unpredictable clocking waveforms, particularly when circuit delays are taken into account. Special structures are necessary for multiplexing different clocks [7]. The clock waveform must also exhibit low slew; i.e. fast rise and fall times, to increase the usable clock period for propagation delays of other signals. But for latches, it is not uncommon to see more complex logic in its enable as jitters or skews are not an issue when the latch is used as an unlocked latch and is not connected directly to another such latch. In fact, most CDC tools check for improper logic gates in the clock tree logic. Those checks should not be applied on control signals that control the latch enables.

To avoid metastability issues at the latch and its destination registers, the enable signal, data input, and data output of the latch must all belong to the same clock domain. If the enable signal of the latch is not a clock, then it leads to few possibilities.

1. *Case 1. All the sequential elements (registers, latches and memories) driving the latch and driven by the latch are in the same clock domain, which is same as the clock domain of signals driving the latch enable input.* Since there is only one clock for the connected sequential elements, they all have a synchronous relationship to each other. Static timing analysis (STA) is able to analyze the timing relationships and flag tim-

ing violation. Hence, there is no CDC path that must be synchronized. Figure 1 shows a simple scenario with just one driver for the latch enable. But for latches, this could be a combinational expression driven by multiple registers in the same clock domain as the TX and RX registers of the latch. This is the most common situation we have observed in the designs analyzed. In case there is a single driver, then it is simply a derived clock (provided other conditions are met, such as a feedback loop) and the results are same with the alternative approach as the derived clock is grouped to the primary clock. In general, the enable could be a combinational expression composed of signals from the same clock domain and can result in a new clock if the clock propagation fails to propagate the driving clocks through the combinational expression.

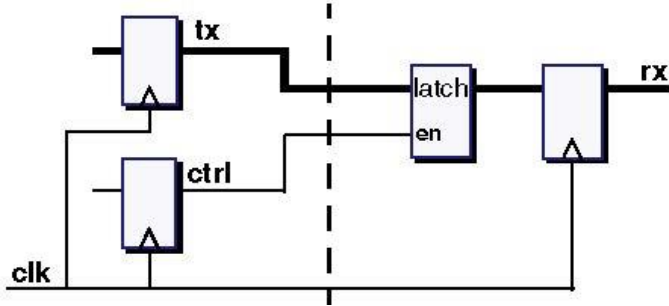


Figure 1 An unlocked latch within a single clock domain

2. *Case 2. The enable signal of the latch is from a different clock domain.* Figure 1 depicts such a situation. This is a CDC path at the latch input. The latch data input can change asynchronously relative to the latching (enable) signal, causing metastability at the latch. The latch input can also trigger a change in output of the latch while the sequential elements driven by the latch are clocking (within its setup and hold window), causing metastability issues at the destination registers. This CDC path should be addressed by the designer. Figure 2 shows this situation.

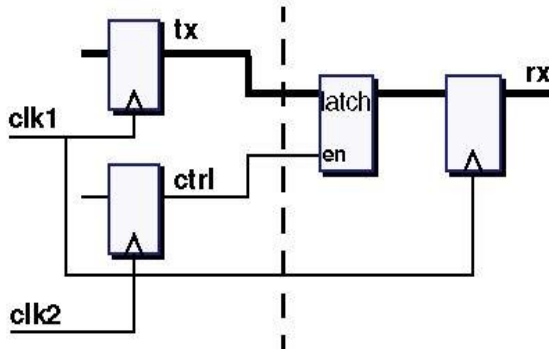


Figure 2. Latch involved in a CDC path

3. *Case 3. The enable of the latch is composed of signals coming from multiple clock domains.* Irrespective of the clock domain of the D input of the latch, its enable signal can asynchronously latch data while the input of the latch is changing, causing metastability issue at the latch. Similarly, a change in the output of the latch can be asynchronously triggered by a change in the enable signal. In Figure 3, this can happen due to a change in the ctrl2 output, thereby causing a change in the input of register rx when it is capturing data, resulting in metastability at register rx. The most likely cause of the issue is an incomplete clock setup or interfering. It could also be due to unspecified clock relationship between clk1 and clk2. Or, it could be a design error, where a signal from the clk2 domain enables a latch operating in the clk1 domain.

Analyzing and Identifying CDC paths through latches: One traditional approach is to treat the enable signals as clocks. However, this can lead to spurious and noisy CDC paths where none exist. As shown in case 1 example, if we treat the enable of the latch as another clock, it creates false CDC paths between the latch and its surrounding logic.

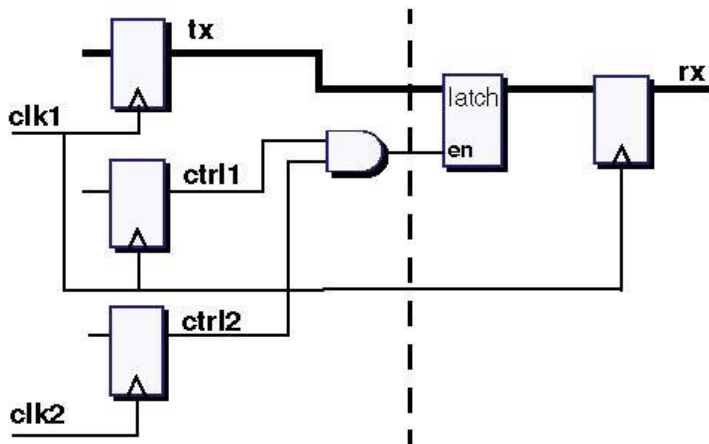


Figure 3. Latch enable is from two different clock domains

But clearly, there is no CDC issue in this case as the enable is dependent on clock *clk*. An alternative approach is to think in terms of clock domains of the latch enables instead of clocks. As mentioned earlier, sometimes the enable signal of a latch is a gating signal which may not be a clock. Using this method, first all the registers are assigned clock domains based on their clock signal. Thereafter, the controlling domains of the latches can be inferred from the domains of the driving registers, latches, primary inputs and constants or stable signals. Since latches can in turn drive the enable of other latches, this inference is applied recursively till all the latches can be assigned to either a single or multiple clock domains based on the logic driving their enable input. Any latch that remains unassigned to a known clock domain is something that the user has to look into, and either define the clock domain of the enable, or declare the enable as a clock.

Results: We applied this method to 129 large and medium scale designs. Many of them did not have any latch CDC paths. Table 1 presents the results from the 8 designs that had the most number of unlocked latches identified by our CDC tool. If we do not resolve the latch domains, CDC results can be very noisy (column 3). The proposed method points to the root causes of the high number of latch crossings that the user can investigate and resolve. Once the latches inferred in multiple domains are resolved, latch CDC path counts for these worst designs dropped by over 90%; for (1) it was a reduction of 98%. Note that this table counts bits of latch registers, so in some cases the numbers are large.

Design	Total latch bits	Latch CDC Paths	Latches inferred in a single domain	Latches inferred in multiple domains
1	455190	2460094	18492	436253
2	5382093	637841	4465438	514991
3	404494	92160	0	98304
4	311255	82741	1516	55549
5	367043	73450	76	10387
6	53792	34662	0	2401
7	52813	18523	133	10280
8	14873	14376	9	14596

Table 1. Eight designs among 82 with most latch CDC paths in our evaluation

We also compared the results for 129 large designs with the alternative approach where we infer clocks for the enables of latches. Overall, this resulted in 88% more identified clocks, some of which are not actually clocks. In two cases, it actually led to a reduction in number of identified clocks, as some of the new clocks helped in resolution of the existing clock groups. Figure 4 demonstrates the number of clocks in 25 of these designs. In addition, we

also compared the number of CDC signals between the two approaches. Overall, total CDC signals increased by 23% when inferring clocks for latch enables vs. using this method of handling unlocked latches.

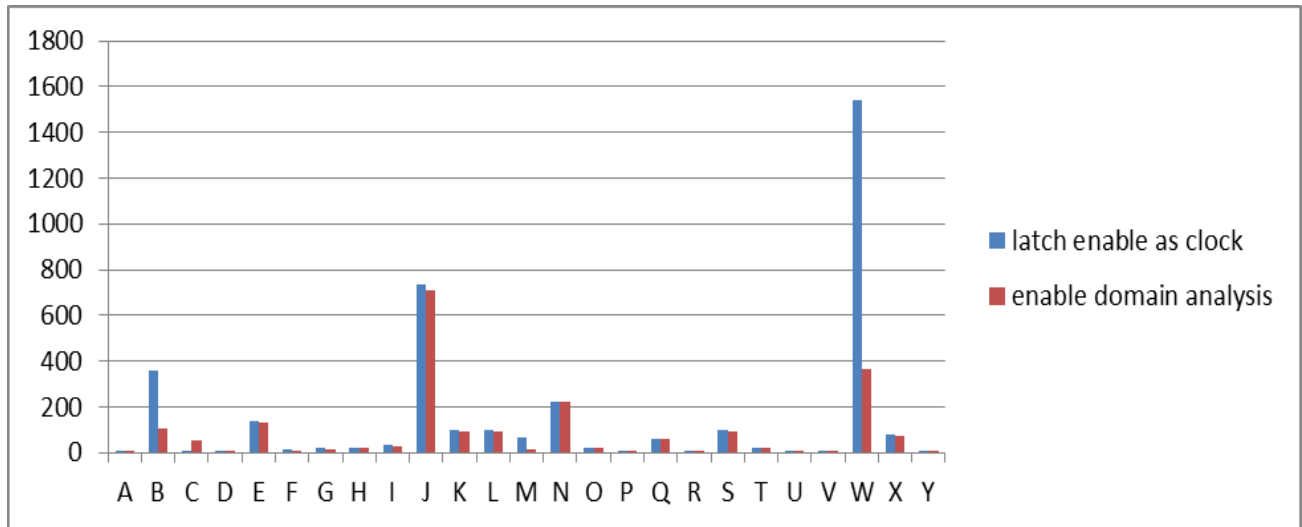


Figure 3 Comparison of number of identified clocks when enables of latches are treated as clocks vs. when the enables are treated as control signals, and their clock domains are used in CDC analysis.

Reviewing CDC Paths for unlocked Latches: The latch CDC paths described earlier can be tool limitations in properly identifying the clock domain of the latch enable, or they may be real design issues of using incorrect signals as latch enables. If a tool limitation, additional constraints to the CDC tool should address the issue. Otherwise, proper design fixes are needed to avoid metastability in the chip.

Once the CDC paths due to the unlocked latches are identified, these paths need to be understood and resolved to avoid potential chip failures. Case 2 scenarios should be reviewed by designers, then either fixed or waived. Based on real designs, case 3 is more challenging. First step is to review and determine if clock domains for such latch enables are correct. Perhaps the different clock domains are synchronous and should be grouped. By adjusting the CDC setup for latch enables, number of latch CDC paths can be reduced significantly.

Conclusion: In this paper, we discussed the added challenges of verifying latch based CDC paths, described an approach to detect the true latch CDC paths and presented our results for a set of industrial designs. We also compared this approach with another method where latch enables are treated as clocks, and showed that the proposed approach is less noisy for majority of the cases. Latches are an integral part of high performance designs; so addressing metastability issues related to latches is critical for proper functioning of the designs in silicon. The approach presented here guarantees that CDC crossings through latches are detected. A latch aware static timing analysis must still be performed to ensure that there are no timing violations.

References:

- [1] Tai Ly, Chris Kwok, L. Curtis Widdoes, "A Methodology for Verifying Sequential Reconvergence of Clock Domain Crossing Signals", DVCon 2005
- [2] Robert Melchiorre, A study of metastability in CMOS latches, MS thesis, Lehigh University, 1992
- [3] C. Tretz et. al, "Metastability of SOI CMOS Latches", International journal of Electronics, Volume 86, 1999- Issue 7, pages 807-813
- [4] Ashish Goel, Ateet Mishra, "Latches and Timing", EETimes 2011, http://www.eetimes.com/document.asp?doc_id=1278980
- [5] Jan M. Rabaey, Anantha Chandrakashan, Borivoje Nikolic, "Digital Integrated Circuits (2nd Edition)", ISBN-10: 0130909963
- [6] "Why should I care about transparent latches", <https://www.doulos.com/knowhow/fpga/latches/>
- [7] B. Jovanović and M. Damnjanović, "Glitch Free Clock Switching Techniques in Modern Microcontrollers," in *Proceedings of the 5th Small Systems Simulation Symposium*, 2014, pp. 119–122.