CDC design techniques

By – Vishal Bhatt

In today's era of complex SoCs various design sub-blocks running at different frequency are common occurrence. Even with today's advance functional verification solutions, CDC signals pose unique and challenging issues. While STA is an integral part of the timing closure solutions, it does not address the issue of proper CDC implementation. To say the least, CDC issues if not detected in late design stages or worst during post-silicon validation can lead to heavy financial penalties.

Various issues caused by CDC can be bracketed into two basic problems,

1. Issue of meta-stability propagation.
2. Functional issues due to signal crossing clock domains.

Following article describes design methods to avoid above two problems.

1. Issue of meta-stability propagation:

Meta-stability refers to condition when signals do not assume logic 1 or 0 for some duration of time. In multi-clock environment meta-stability cannot be avoided but its effects can be neutralized.

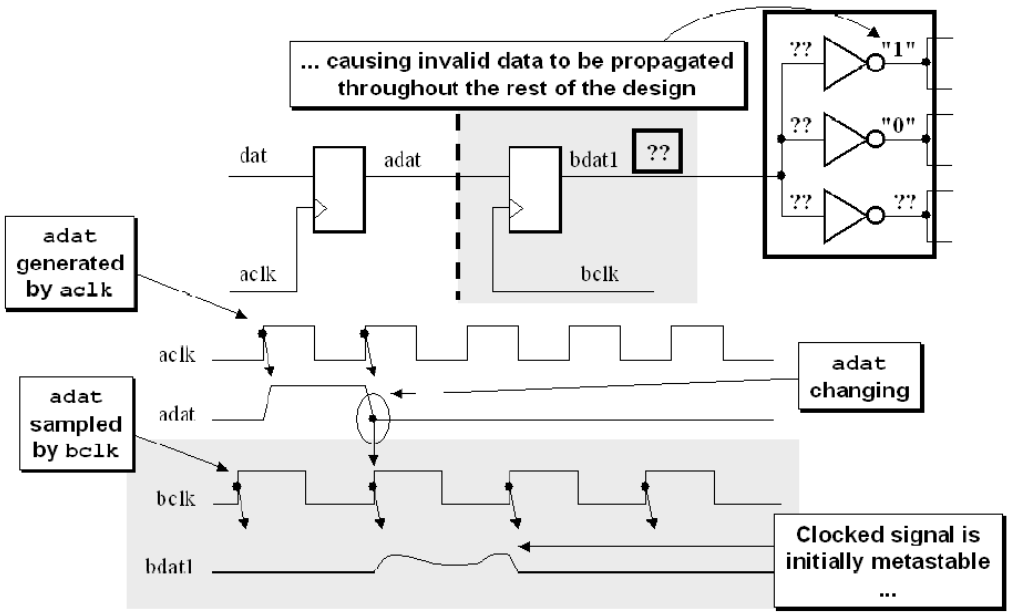

Figure 1

Fig 1 shows how the pulse in aclk domain fails to propagate to bclk domain as it is sampled very close to the capture clock of the second clock domain thus violating its setup/hold time. As the signal 'bda1' in bclk domain fluctuates for 1 clock period

before settling down it can propagate erroneous 1s and 0s in the following logic as shown in figure 1 by the outputs of the inverter connected to the bclk domain flop.

In the example shown above the 'bdat1' signal only fluctuates for 1 clock period and then finally settles down to logic 0, but it is very well possible that the signal settles after 2 period of receiving clock domain. Theoretically one cannot ascertain exactly after how many clocks the metastable signal will settle down, however one can have reasonable judgment of probability of synchronization failure using MTBF (Mean time before failure) analysis for CDC signals.

When calculating MTBF, failure means that signal is metastable when it is sampled in the second stage of synchronizer flop. For MTBF larger numbers are better, if the MTBF value is large there is less probability of failure while smaller numbers indicate there is high chances of metastability propagating in subsequent logic.

Dally and Poulton give good equation with very thorough analysis of the how to calculate the MTBF of the synchronizing circuit. Looking at figure 2 the two most important factors which impact the MTBF are clock frequency of the sampling domain and rate at which synchronizing data is changing. For higher values of this two factors there are high chances that it will result synch failure.

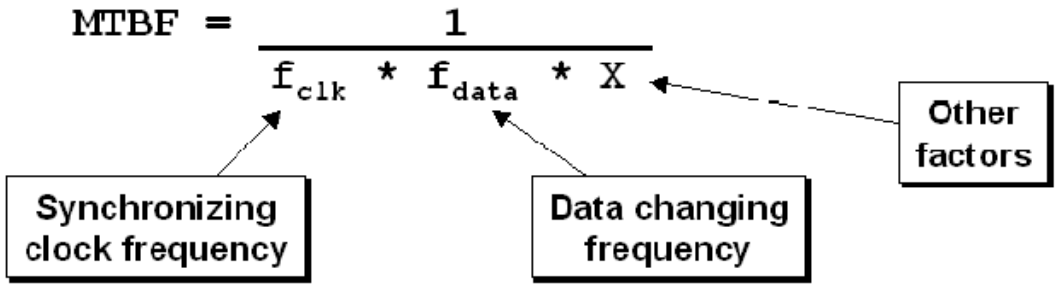

Figure 2

It is seen that for most of the designs as 'thumb rule' two flop synchronizers are used and for certain high speed applications three flop synchronizers are used. However it for the system architect to judge as to how many synchronizing flops are required in the given design based on the MTBF calculation.

2. Functional issues due to signal crossing clock domains.

Following points are the main concern in terms of functionality of signal crossing clock domain,

1. Does the signal correctly propagate to the other domain?
2. Is there a convergence between two propagating signals in the other clock domain?

Signals crossing from one clock domain to another can be of various types,

1. Single clock pulse
2. Level signals (remains active high or active low for multiple clock signals in the sending domain)
3. Vectors (Multi-bit signals)

3. Single-clock pulses and level signals.

Also, the CDC can be from Fast to slow domain or from slow to fast. Let us first discuss the case when CDC is from Fast clock to slower clock domain. A very important fact is that how the synchronizing logic handles fast changing CDC signal. When passing signal from fast clock domain to slow it is possible that signal changes even before it is captured. There are generally two approaches to handle this problem

1. Open loop solution which ensures that CDC signals are captured without any acknowledgement returning back to the sending domain.

   This solution needs very thorough knowledge of CDC signal as there is no acknowledgement involved. It is not used for single clock pulse signals. According to Mark Litterick, when passing CDC signal using double flop sync, the signal must be 1-1/2 times the cycle width of the capturing domain clock period. This requirement is known as requirement of 'Three edges'. Generally as a 'thumb' the open loop solution like the double flop sync is preferred only when the CDC signal is a level signal and remains stable for not less than 3 clocks in the receiving domain.

   If one can ascertain that the CDC signal is 'reasonably' (satisfying the 3edge requirement) stable for the capturing clk domain this is the fastest method for clk sync.

   There are other solutions available where if the CDC single is a pulse which does not satisfy the 3edge requirement but if the frequency of pulse is such

that between two consecutive pulses there is a time gap of 3edges of receiving clock. Following figure shows one such example called toggle flop mechanism can work.

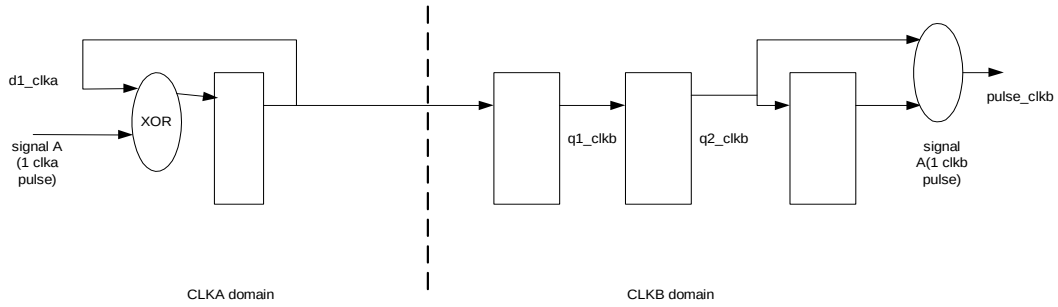

Figure 3

2. Closed loop solution which requires acknowledgement of the CDC signal safely reaching the other clock domain.

   As the name suggest close loop solution always ensures the 3edge requirement by controlling the desertion of the CDC signal in the sending domain using acknowledgement. Figure 4 shows the closed loops solution for passing a clock pulse from fast clock domain to slow.

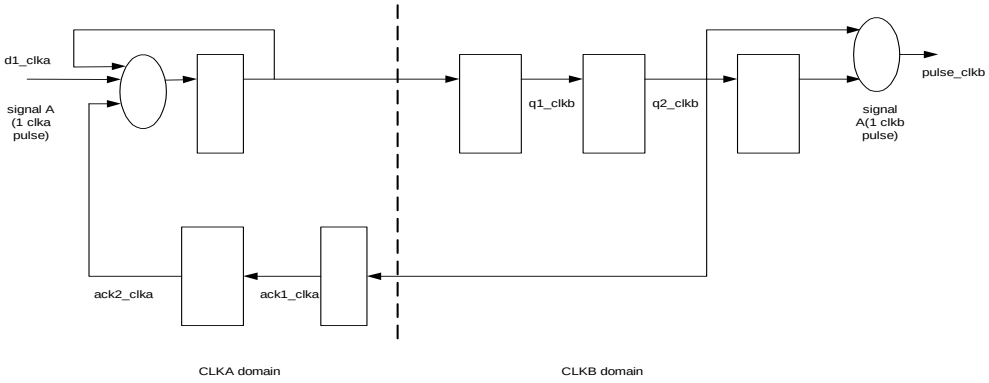

Figure 4

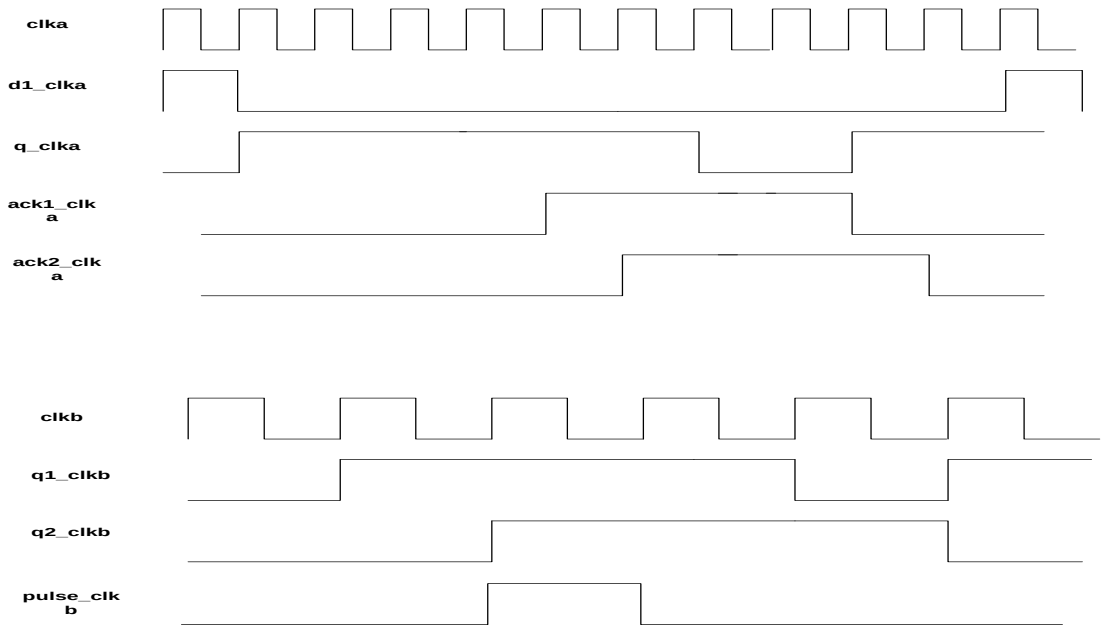

Figure 5

The disadvantage of using the closed loop solution can be seen from the figure 5. , the acknowledgement signals ack1_clka and ack2_clka introduces delay between two back to back pulses. Here a system in the clka domain can use the ack2_clka signal as a 'ready' signal which indicates that the synchronizer is able to accept new pulses when the ready signal is at logic low.

4. Vectors (Multi-bit signals)

Vectors cannot do CDC using 2D or 3D Flops as explained earlier because of issues with convergence. It is possible that in-case the vector is of 2 bits then both the bit do not reach the destination logic at same time thus causing functional error.

There are two basic ways in which Vectors can be passed

1. Using open-loop solution where a control signal validating the 'vector' is synchronized into other another clock domain.

   Here a control signal is used along with Vector which validates a new vector at the CDC boundary. This valid signal is then CDCed using standard 2D or 3D flop logic and used in the receiving domain as a pulse to register the Vector. There are two assumptions in this case,

   a. The input vector will be stable by the time Valid signal reaches the receiving logic

b. The input vector does not change during the valid signal is crossing clock domain.

Figure 6 shows one such implementation.

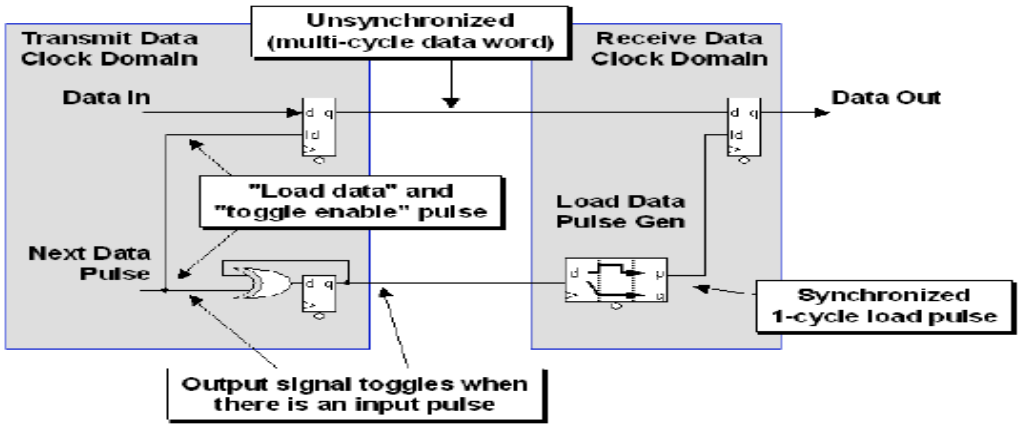

Figure 6

2. Using closed-loop solution where successful data received generates 'ack' in the sending domain.

   To remove the limitation b mentioned above there are closed loop solution which are available which indicate the sending logic if the synchronizer is ready to take another input. Fig 7 shows one such closed solution,

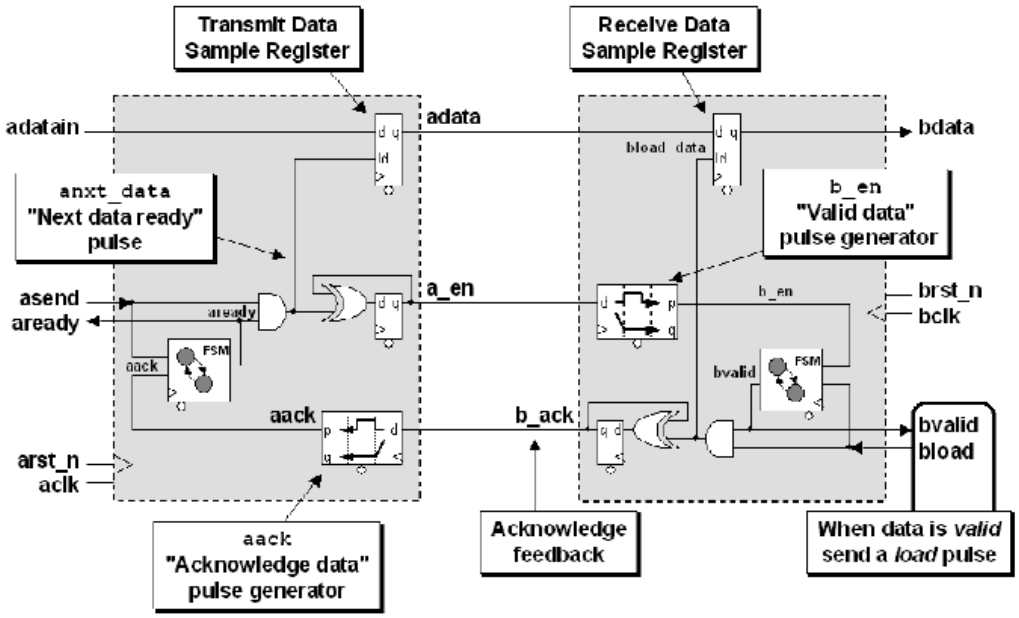

Figure 7

Here the sending logic checks the status of signal 'aready' before pulsing in new data using 'asend' control signal. Again this type of solution has penalty with respect to delay in sending back the 'ack'.

5. Using Asynchronous FIFOs

In case of bursty traffic which needs to be send from one clock domain to another this type of solution can be of real help, as in such scenario applying back pressure in middle of burst is not possible. When going from fast to slow clock domain the depth of the FIFO depends on the burst length of incoming data.

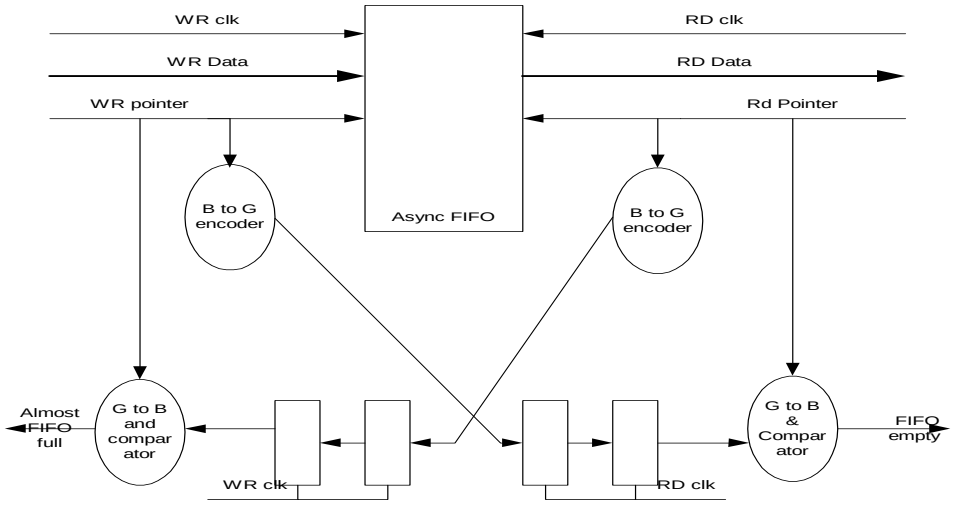

Figure 8

Here the Binary to Gray counters are used to cross WR and RD counters from WR clock domain to RD and vice versa. It is possible that all the counts are not always passed and some counts are missed but the greater consideration is that the counters don't overrun the boundaries and miss the FIFO full or empty condition. Even though the sampled gray count is missed overall the system is robust.

References

1. CDC Design and Verification techniques using System Verilog by Clifford E. Cummings
2. Clock Domain Crossing : Closing the loop on clock domain functional implementation problems
3. Technical paper by Cadence design systems.