Clock Domain Crossing

Part 7 – Timing Constraints

Amr Adel Mohammady

in /amradelm

Save The **Palestinian** **Children**

Rami Igra
Former Israeli Intelligence Official

"Children in Gaza over 4 deserve to be starved"

Former Mossad official: 'Children in Gaza over 4 deserve to be starved' – Middle East Monitor

Bezalel Smotrich
Israeli Minister of Finance

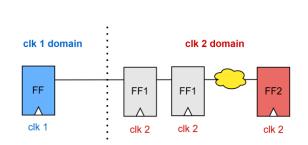"Might be 'justified and moral' to cause 2 million Gazans to die of hunger, but world won't let us"

Israeli minister says it may be 'moral' to starve 2 million Gazans, but 'no one in the world would let us' | CNN
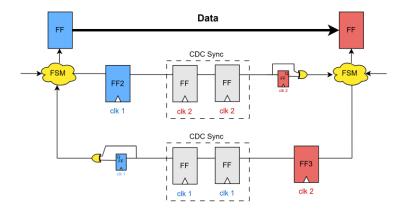
The Human Cost Of The Israeli-Palestinian Conflict
Israeli & Palestinian deaths/injuries documented by the UN

■ Deaths ■ Injuries

Total deaths from 2008 to 2020
🇵🇸 5,590
🇮🇱 251

Palestinian	Year	Israeli
3,202	2008	853
7,460	2009	123
1,659	2010	185
2,260	2011	136
4,936	2012	578
4,031	2013	157
19,860	2014	2,796
14,813	2015	339
3,572	2016	222
8,526	2017	174
31,558	2018	130
15,628	2019	133
2,781	2020	61

Source: United Nations

statista

Israel has been killing Palestinians long before Oct 7
Do Palestinians have the right to defend themselves?
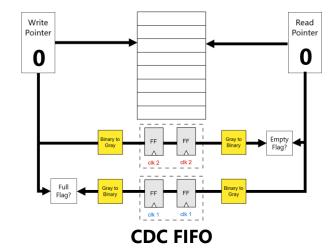
Introduction

- In the previous parts we went through all the CDC solutions and schemes.

- In this part we will discuss the timing constraints associated with these schemes.

- We mentioned that CDC paths are asynchronous and therefore can't be analyzed with static timing analysis. '

- That's why, in the past, the most common approach was applying false paths on CDC paths. We will see how that may lead to major issues.

- It turns out we still need some timing constraints to enforce some assumptions we made when designing the CDC circuit

CDC Synchronizers
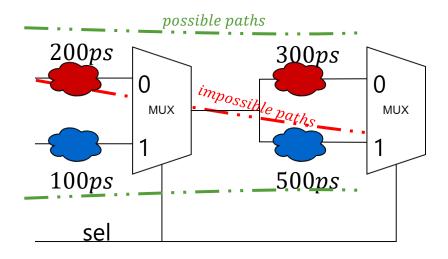
CDC Handshake Protocol

CDC FIFO

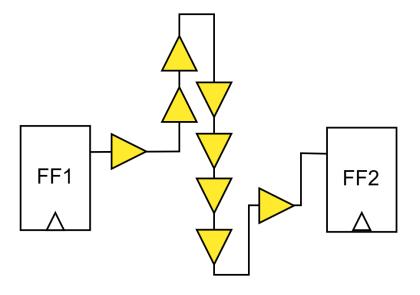# 1st (Trial) Solution : Apply False Path

What is a False Path?

- False paths are timing paths that can't possibly occur due to the logic of the circuit
- **Consider the example below:**
  - Both muxes have the same select signal. This means we have 2 possible timing paths. The one going through both **red** logics ($200 + 300 = 500ps$) and the one going through both **blue** logics ($100 + 500 = 600ps$)
  - The paths going through a **red** logic then a **blue** logic ($200 + 500 = 700ps$) or **blue** logic then **red** logic ($100 + 300 = 400ps$) is impossible to happen.
  - Unless we instruct the tool to ignore these false paths, they will be considered for timing analysis leading to the large T_{comb} of the red to blue path which will violate setup.
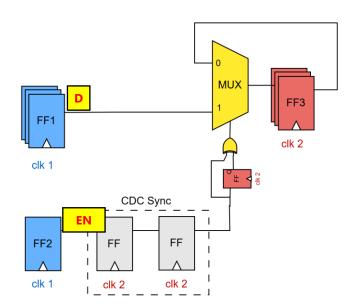
What Would Happen With A False Path Constraint?

- Applying false path will make the tool ignore the timing of the paths and therefore may create unnecessary delay that breaks our CDC circuit.
  - The tool might:
    - Place the launch and capture FF far apart.
    - Create unnecessarily long routes.
    - Add unnecessary buffers in the routes.
    - Use slow cells/FFs to save power.
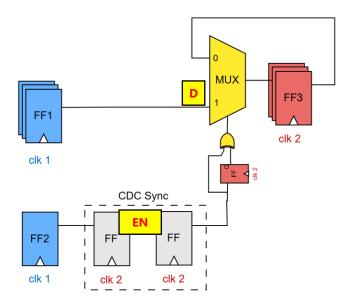 - We will see how this may break our CDC circuits. We will consider 2 examples
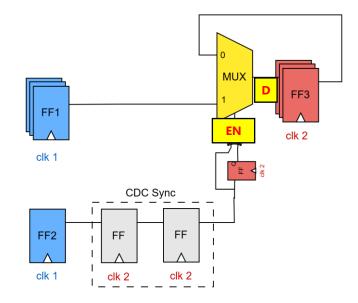   - CDC Mux
    - CDC Gray Coding

Possible Schematic of a very relaxed Path

CDC Mux Scheme And False Path

1 **Let's review the CDC MUX scheme[1]:**
- The data goes directly to the Rx domain MUX
- The enable goes to the Rx domain through FF synchronizers

2
- The data reaches the MUX quickly since no FF exist in the way.
- The enable is still going through the FF syncs

3
- After some time, the enable arrives at the MUX and opens the gate for the data.
- Since the enable is synchronized, it's guaranteed with STA that the gate will open without violating setup or hold time and therefore the data won't cause metastability
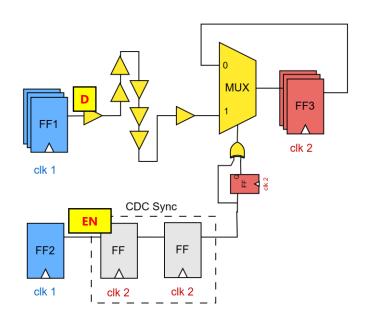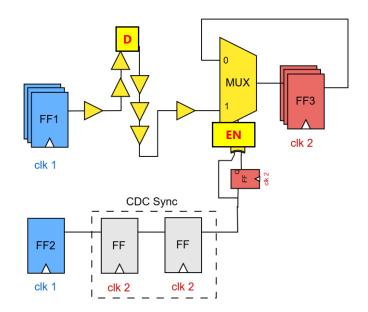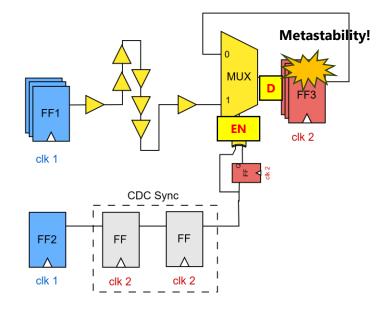
CDC Mux Scheme And False Path

1 **Now let's consider the faulty behavior with a false path constraint applied[1]**
- The data goes to the Rx domain MUX
- The enable goes to the Rx domain through FF synchronizers

2
- The data takes a long time to reach the MUX due to the logic delay.
- The enable reaches the MUX and opens the gate

3
- After some time, the data arrives at the MUX after the enable.
- The data is a domain 1 signal arriving at domain 2 FF. Metastability happens

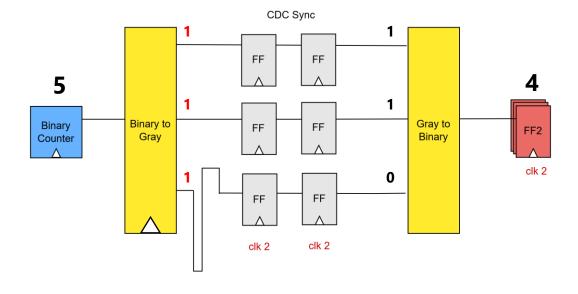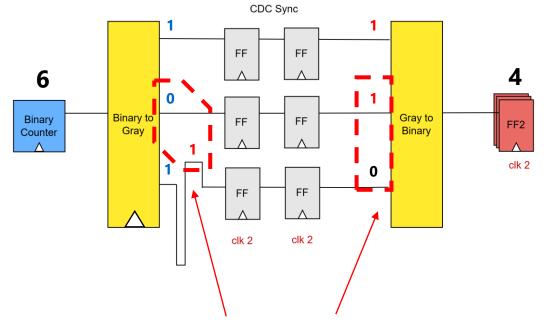Free Palestine   /amradelm   8

CDC Gray Coding And False Path

1 **Now let's see the faulty behavior with CDC Gray coding**
- Initially the Rx sees gray code 110 (decimal 4)
- After that we send gray code 111 (decimal 5)

2
- We then send gray code 101 (decimal 6)
- Due to the long delay on the LSB line, the logic "1" from decimal 5 didn't reach the Rx yet.
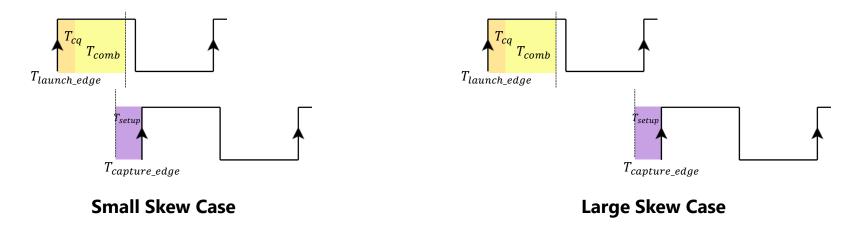- Now the Rx sees two bit changing and may jump to wrong count

Multiple bits are changing at the same time.
The Gray code is violated

Free Palestine /amradelm

2nd (Trial) Solution : Don't Apply False Path

What Could Happen Without a False Path Constraints?

- By default, the tool will assume the 2 CDC clocks are synchronous and will run STA on any path between them.

- **This will lead to one of 2 issues:**

  o <u>If the clock skew between the 2 clocks is small</u>, the path will be very tight and won't meet timing. The synthesis and PnR tools will spend a lot of effort trying to fix the path[1].

  o <u>If the clock skew between the 2 clocks is large</u>, the path will be relaxed and may meet timing with a large setup margin.

    ▪ The tool might add delay (for example, to save power)

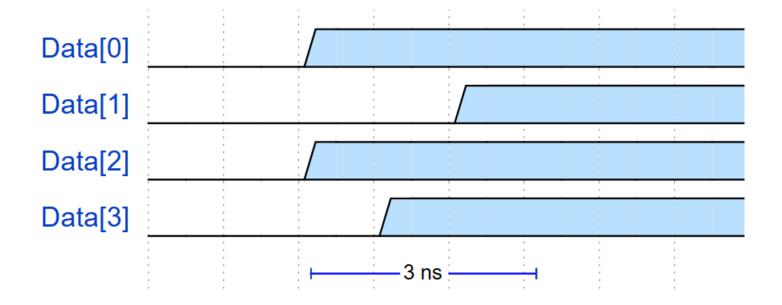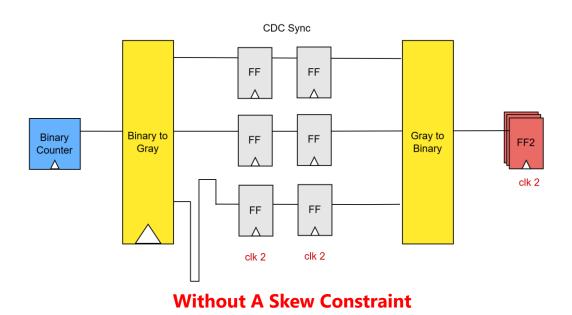   ▪ We get the same issue of applying a false path constraint

Small Skew Case   **Large Skew Case**

[1] : The tools ignore less critical paths and focus on the critical ones. This will lead to real paths being masked by fake CDC violations

Free Palestine   in /amradelm   11

3rd (Correct) Solution : Skew Constraint
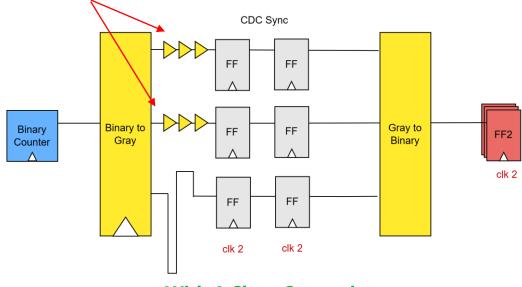
What is A Skew Constraint

- Skew checks constraint the arrival difference between 2 signals or more.

- In the example below we have a data bus of 4 bits. The bits should arrive close to each other with a difference no more than 3ns. This means the difference between the latest bit to arrive and the earliest bit to arrive shouldn't exceed 3ns.

- To fix skew violations we need to speed up slow signals and/or slow down fast ones.

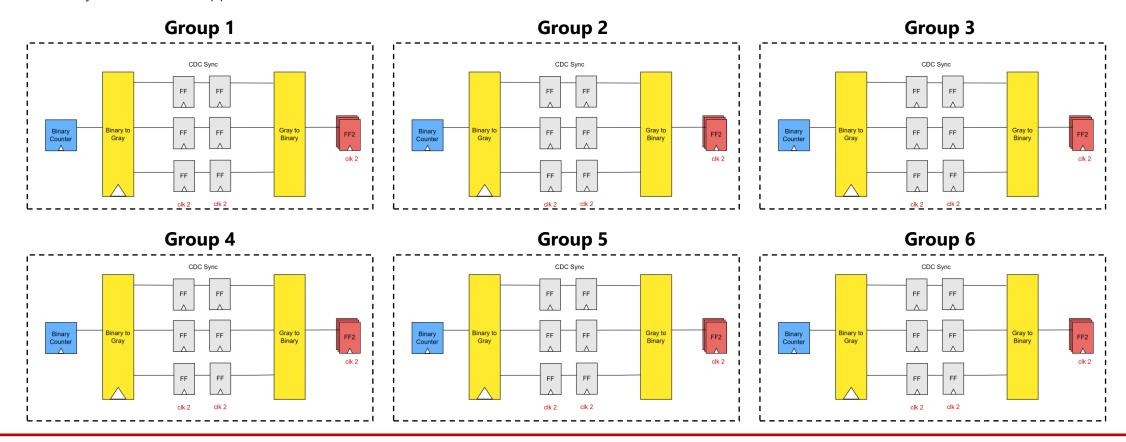# CDC Gray Coding And Skew Constraint

Buffers are added by the tool on the other paths to balance with the LSB route delay.

Without A Skew Constraint

With A Skew Constraint

The Issue With This Approach

- The main issue with this approach is that it needs lots of manual efforts
- If we have multiple CDC paths, we need to identify each group of signals and add skew constraints for them
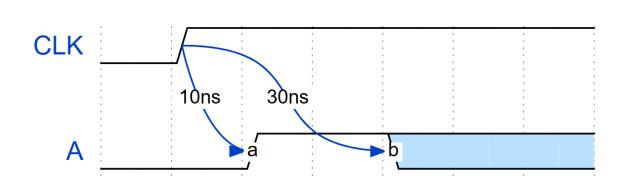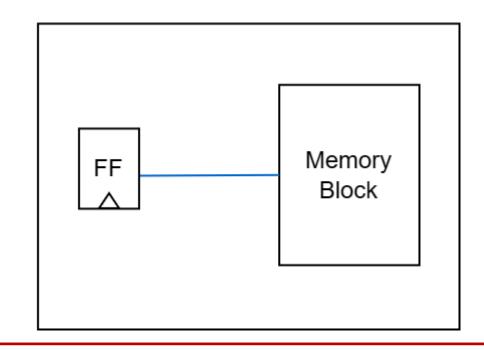- We will try another easier approach

4th (Best) Solution :
Max Delay Constraint

Max and Min Delays

- Sometimes we want to control the arrival time of a signal.
- In the example below, it's required that signal A arrives at the memory block <u>no eariler than</u> 10ns and <u>no later than</u> 30ns after the clock edge.
- To constraint signal A to follow this requirement we need to apply a <u>min delay</u> constraint of 10ns and a <u>max delay</u> of 30ns[1].
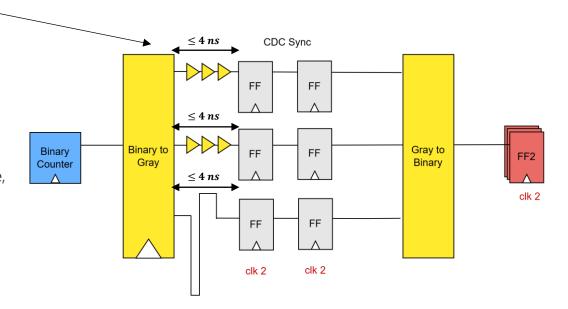
Max Delay Constraint

- The best approach is to add a max delay constraint with an amount small enough that the CDC paths are not broken

- What makes this approach easy is that it can applied to all CDC paths with one line[1] so it doesn't need manual work:

  - `pt_shell> set_max_delay 4.0 -from CLK1 -to CLK2`

- **What remains now is what value to use for the max delay constraint.**

  - In some cases, we need to apply the Tx clock period

  - In other cases, we need to apply the Rx clock period

  - In other cases, we need to apply multiple clock periods.

  - We will use the worst case (smallest) instead of applying a specific max delay value for each

- The value used for max delay might be too tight for some CDC paths. In that case, we can resort to skew constraint

Free Palestine   /amradelm

References

1) https://gist.github.com/brabect1/7695ead3d79be47576890bbcd61fe426

2) Y. Mirsky, O. Tsarfaty, D. Stein, & O. Winner, "Timing Analysis of Unconstrained Clock Domain Crossings – the Need and the Method,"

3) O. Dasa, Y. Mirsky "A New Approach to Easily Resolve the Hidden Timing Dangers of False Path Constraints on Clock Domain Crossings"

Thank You!