



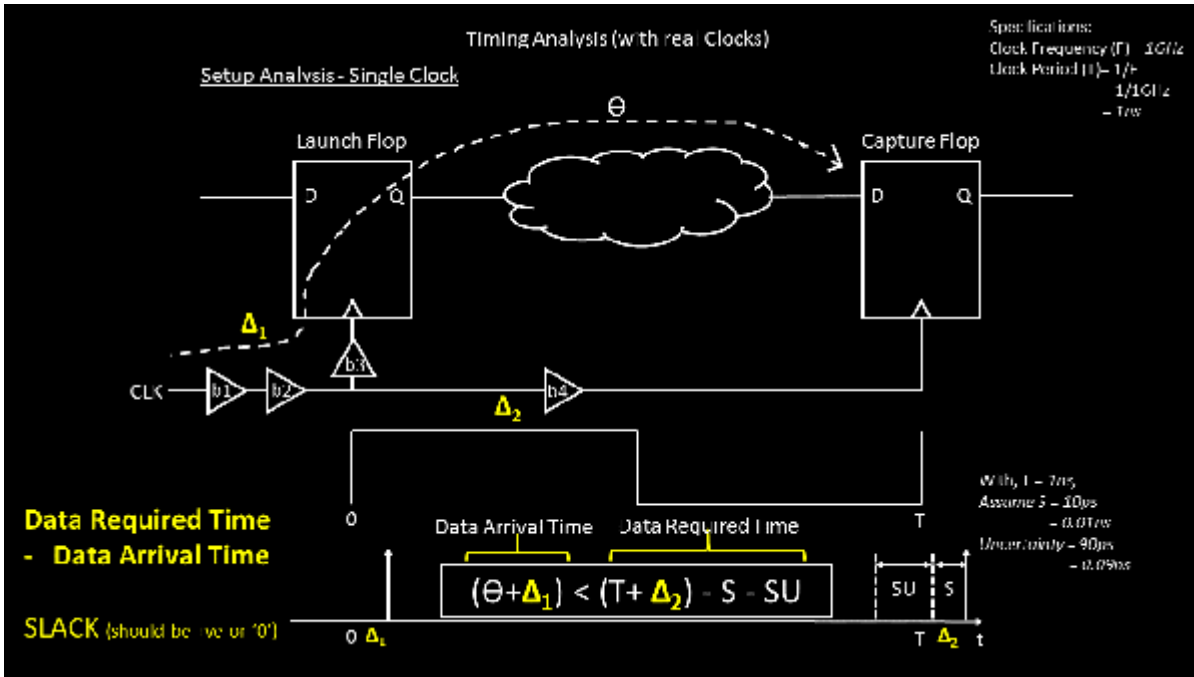
Common clock path pessimism removal (CPPR)

Kunal Ghosh

With **On-Chip variation**, we might introduce extra pessimism in clock path, common to launch and capture flop clock pins. How? I will get back to this in below post (or may be next one).

Our job, is to **remove this pessimism and make a timing path analysis**, close to a real one. How? I will get back to this, as well, in follow-up post

Let's consider below image, to **visualize how a real timing path looks like, what is data arrival, data required and slack**. (for setup timing analysis)



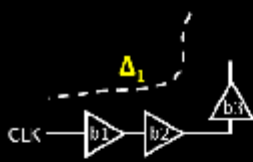
$S =$ library setup time, $SU =$ setup uncertainty

And below images are the textual conversion of above image. **This is what you see in a standard timing report** (we will focus on clock path for now, as that's our point of concern)

Timing Analysis (with real Clocks)

Setup Analysis - Single Clock - Textual Representation

Specifications:
Clock Frequency (F) = 100MHz
Clock Period (T) = 1/F = 10ns
= 1/100MHz
= 10ns

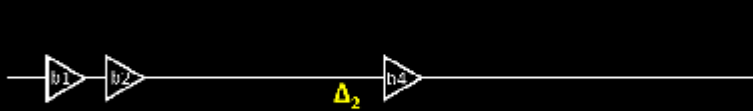


$$\Delta_1 = b1/a$$
$$b1/y$$
$$b2/a$$
$$b2/y$$
$$b3/a$$
$$b3/y$$

Timing Analysis (with real Clocks)

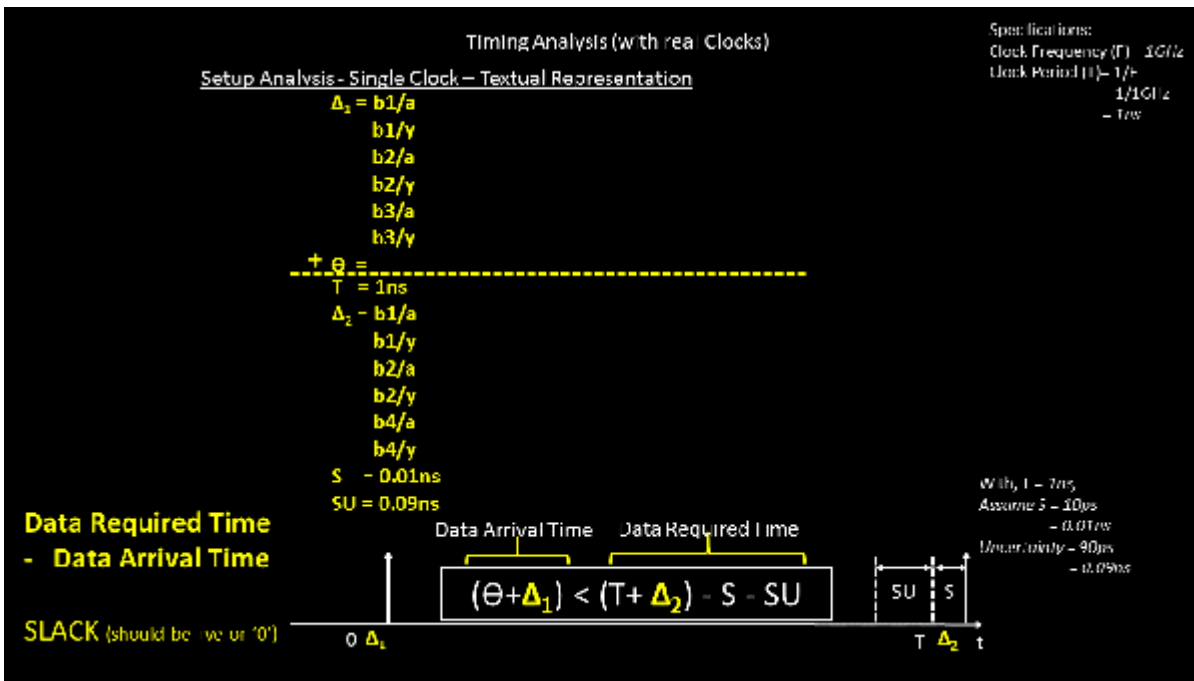
Setup Analysis - Single Clock - Textual Representation

Specifications:
Clock Frequency (F) = 100MHz
Clock Period (T) = 1/F = 10ns
= 1/100MHz
= 10ns

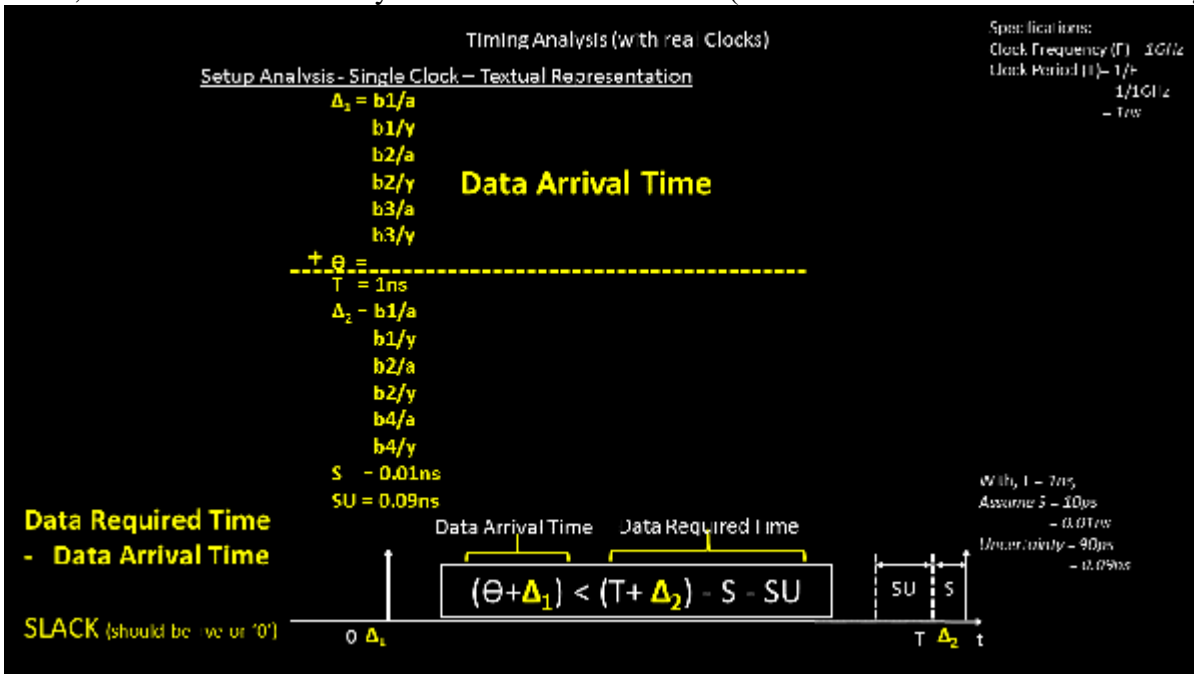


$$\Delta_2 = b1/a$$
$$b1/y$$
$$b2/a$$
$$b2/y$$
$$b4/a$$
$$b4/y$$

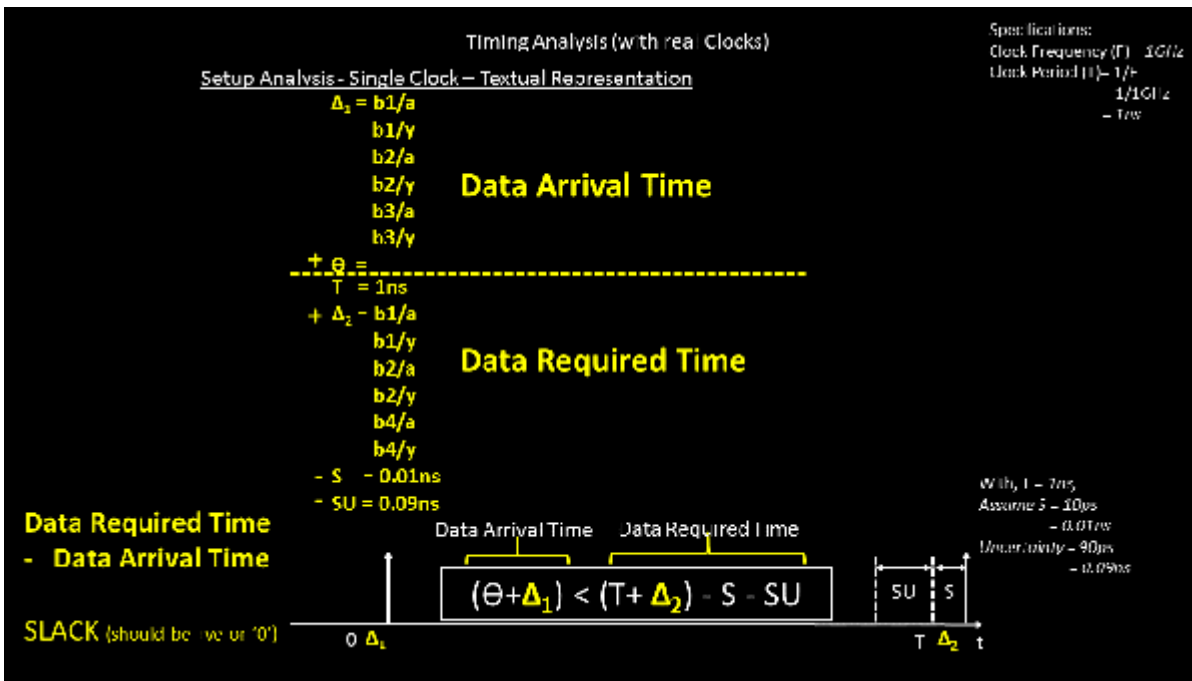
Let's structure the timing report in an understandable format as below



Here, the first half becomes your **data arrival time** (sentence continued after below image)



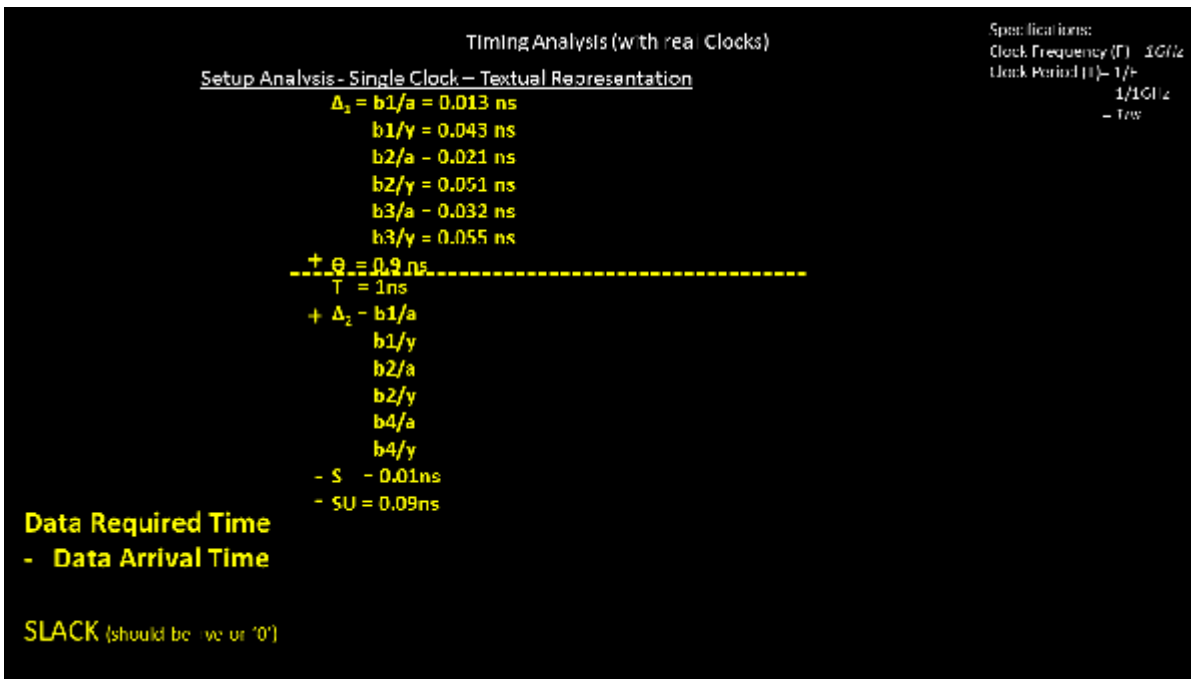
.... (sentence continued from above) and the second half becomes your **data required time**



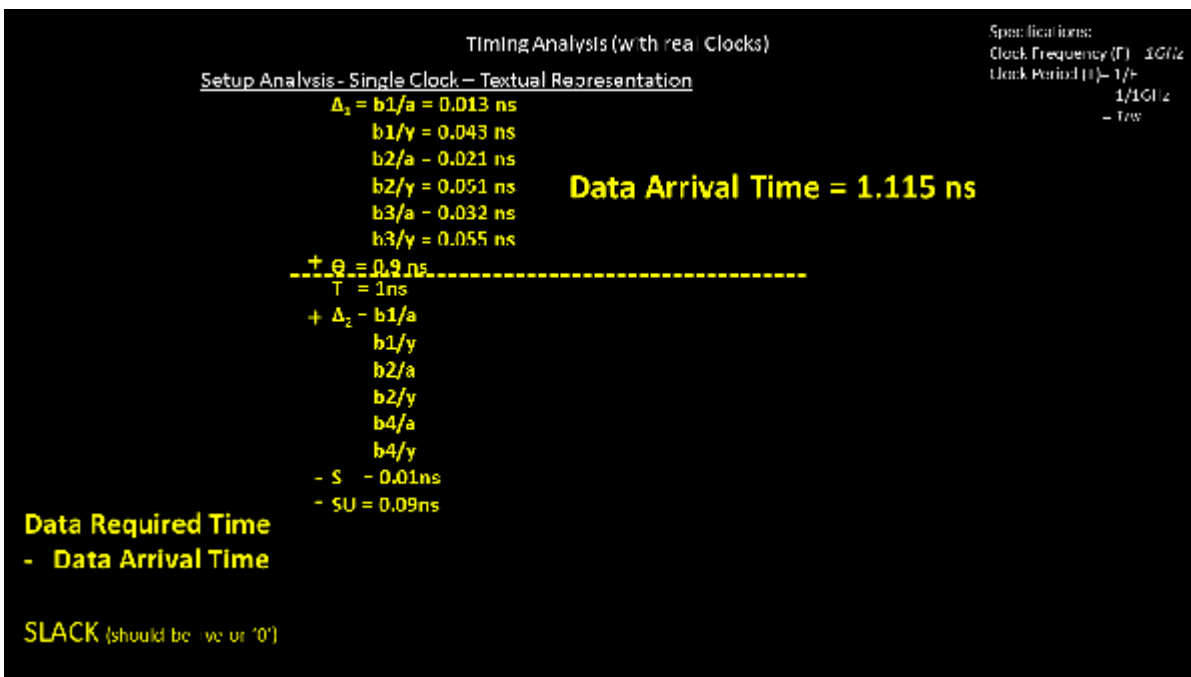
PS: may for homework, take a timing path in your real design, and see if the above makes sense

A timing report without real numbers, is like “A body without skeleton”

Assuming below values for the cell and net delays (over here, **net delay** is the value on xyzyz/a and **cell delay** is the value on xyzyz/z)

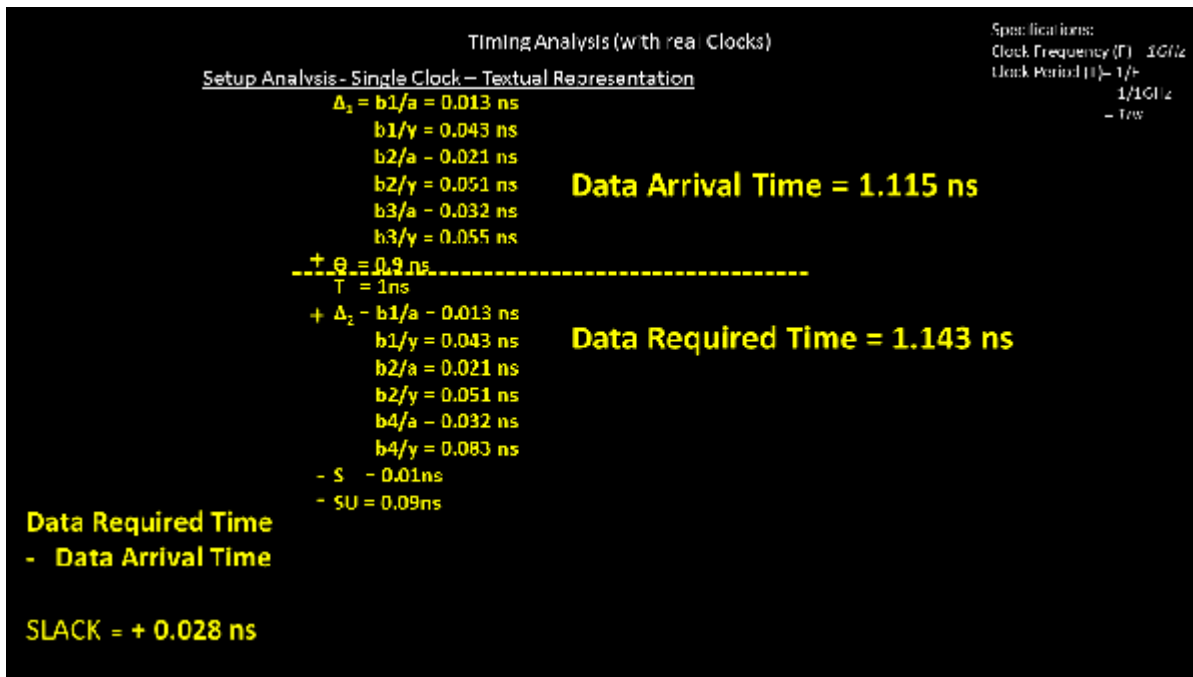
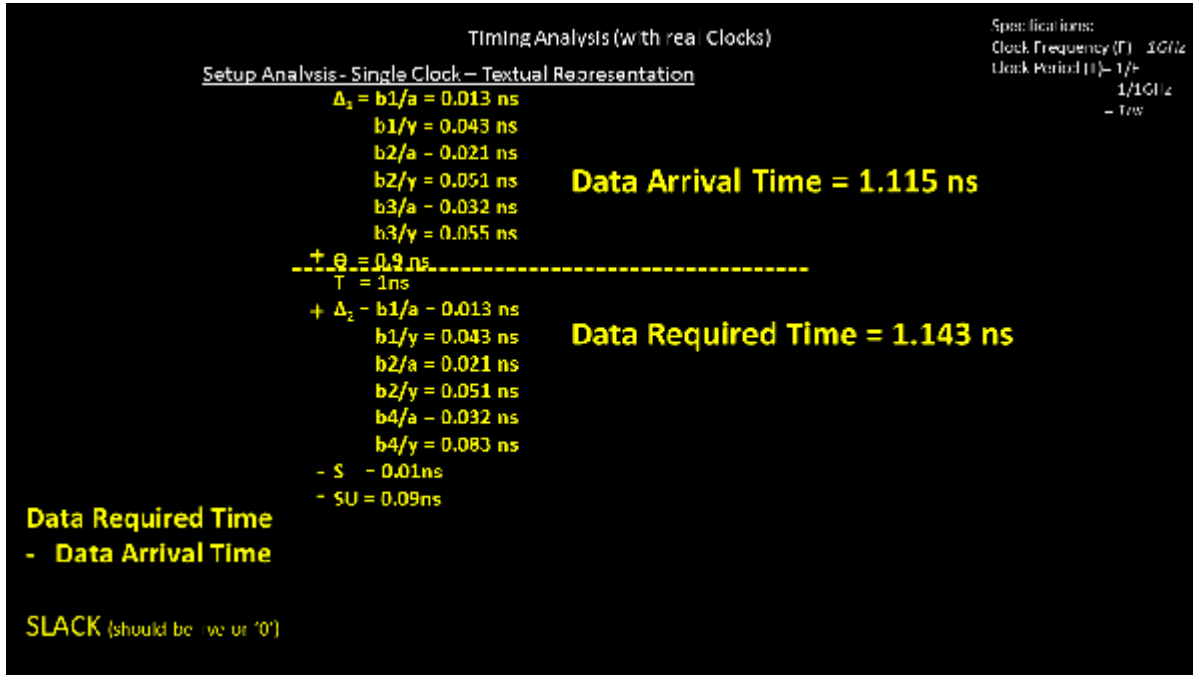


Data arrival time in this case is 1.115ns (nothing complex, just used a hand calculator)



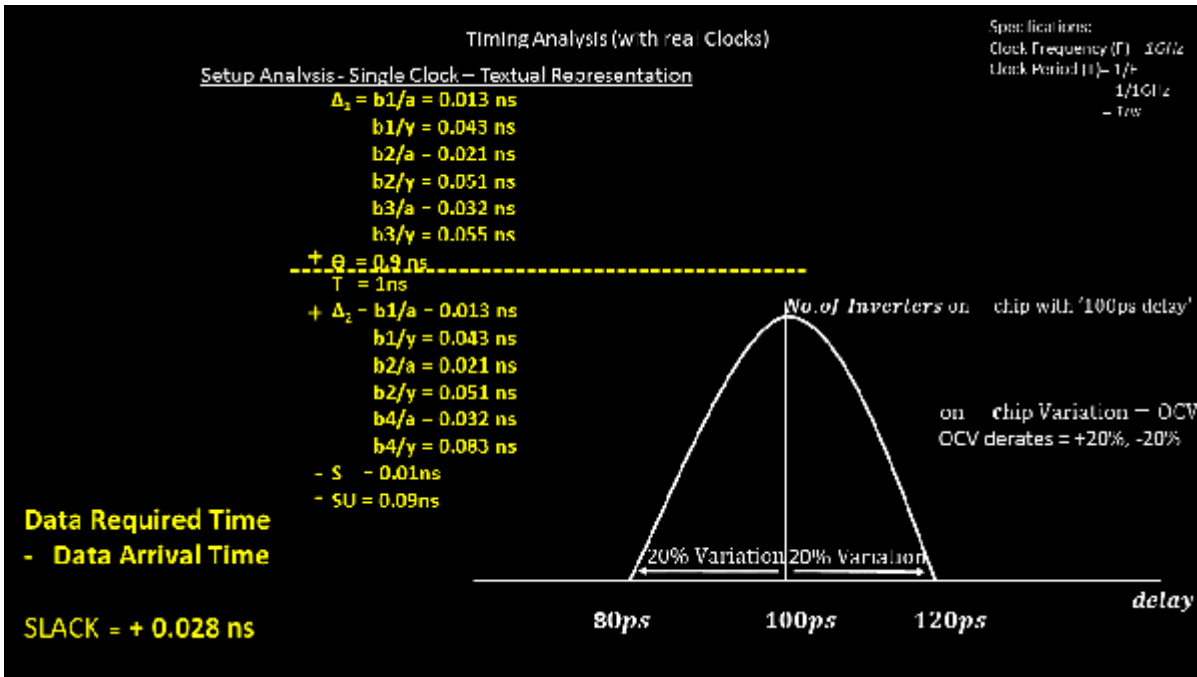
Notice, **b1 and b2 are common cells in launch and capture path**. So, while assuming numbers for capture clock path, the delay values for these cells will remain same as shown below.

With that said, data required time will be 1.143ns slack becomes +0.028ns



And I know, what it means for an STA engineer to see that positive slack.

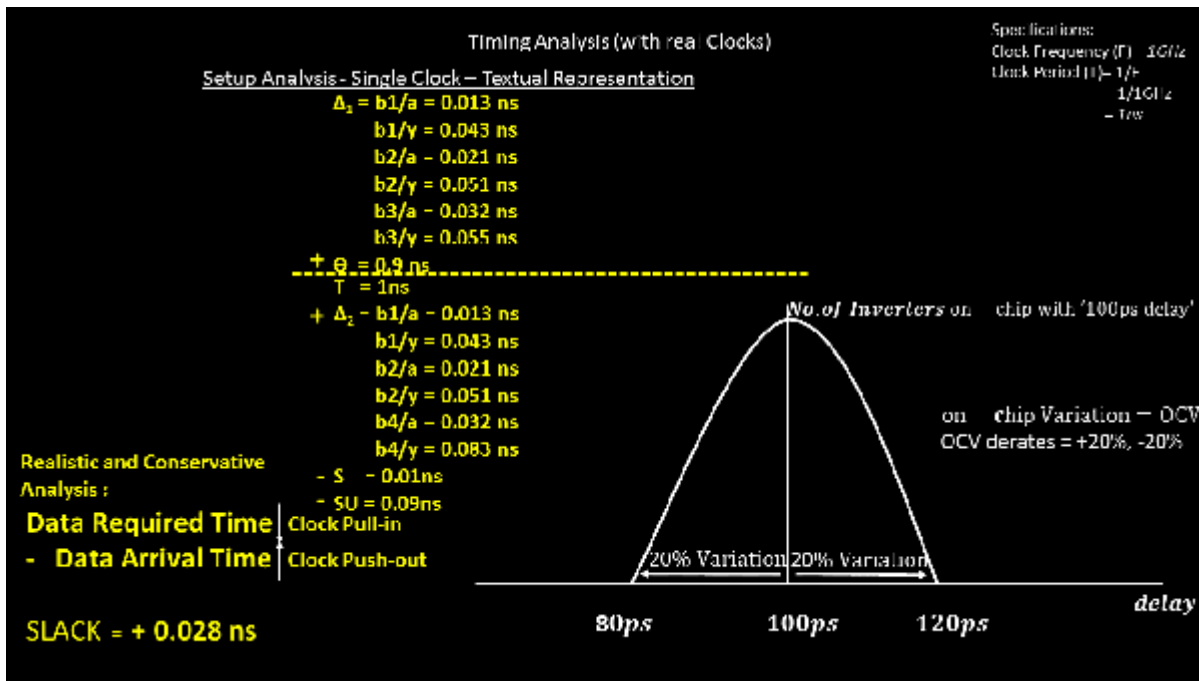
We will take up an OCV graph with +20% and -20% as derates (just to keep the calculations simple over here)



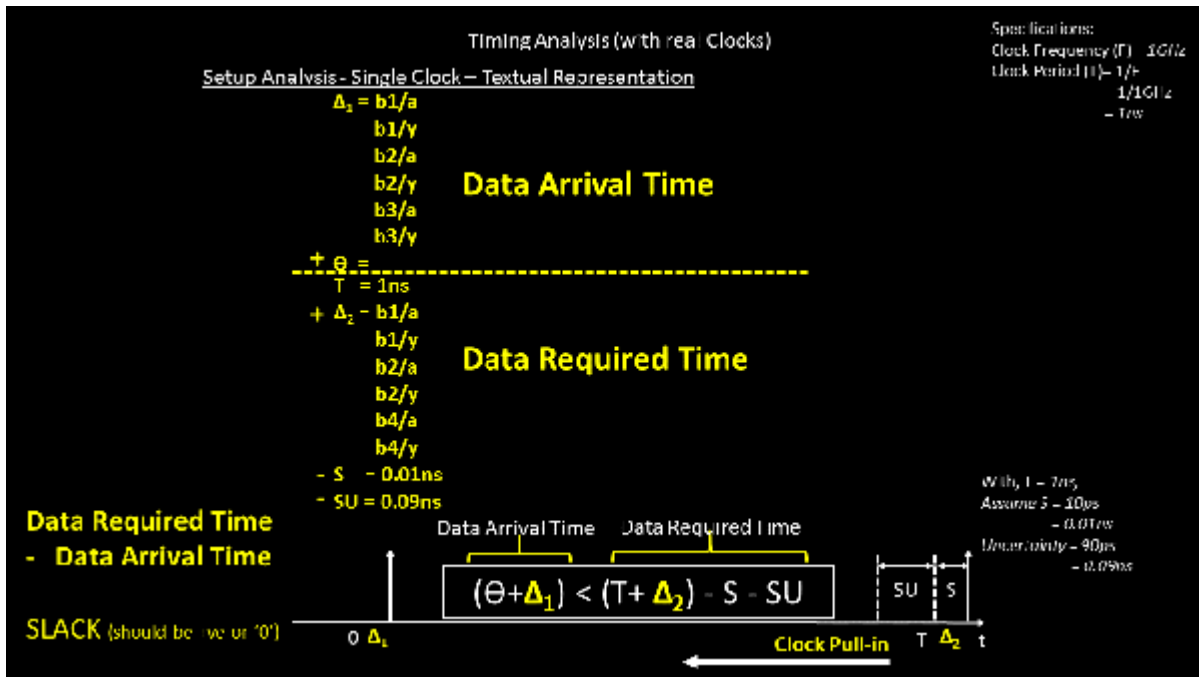
And we will use these OCV values (for now and usually it's the case) on clock path only.

Now, for a moment, look back to the last post for SLACK calculation (for setup analysis). To make these OCV values helpful for us, we need to pull-in the "data required time" and/or push-out the "data arrival time".

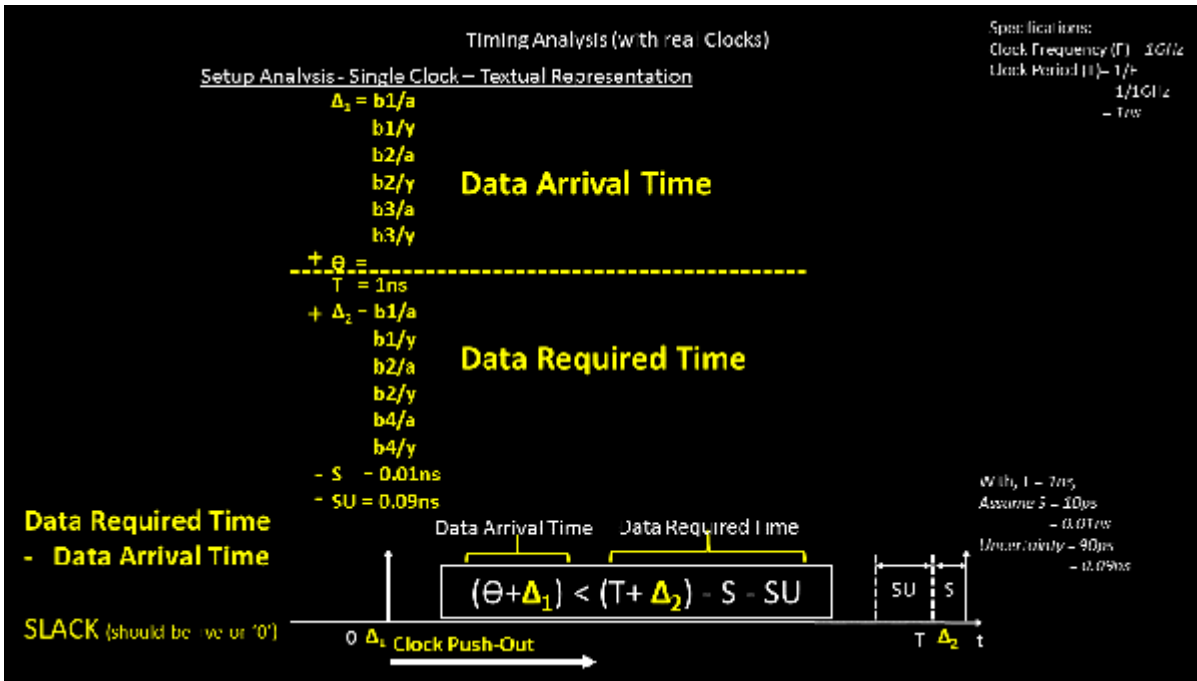
This will make a real worst-case analysis – meaning, if the SLACK meets the above criteria, we can guarantee you, the chip will function, no matter what.



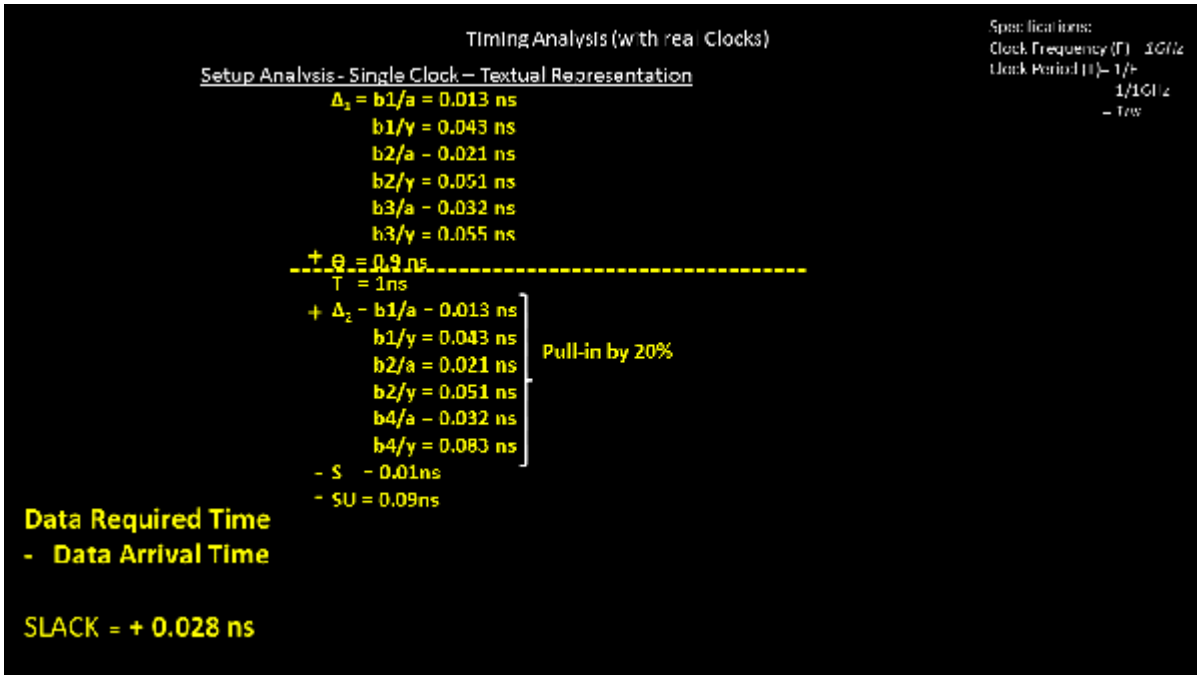
By “pull-in”, we mean, we will bring the capture clock edge more towards the left-hand side, as shown below (in the bottom-right of image), and (sentence continued after below image)



.... (sentence continued from above), by the term “push-out”, we mean to push the launch clock, to the right side (as shown in bottom left of below image)



For now, let's do only one thing i.e. pull-in the capture clock by 20%, i.e. every cell and net delay will be reduced by 20%. Below 2 images show the same



Timing Analysis (with real Clocks)

Specifications:
 Clock Frequency (f) = 100MHz
 Clock Period (T) = 1/f = 1/100MHz = 10ns

Setup Analysis - Single Clock - Textual Representation

$\Delta_1 = b1/a = 0.013$ ns
 $b1/y = 0.043$ ns
 $b2/a = 0.021$ ns
 $b2/y = 0.051$ ns
 $b3/a = 0.032$ ns
 $b3/y = 0.055$ ns

 $\uparrow \theta = 0.9$ ns
 $T = 10$ ns

+ $\Delta_2 = b1/a = 0.013$ ns	- 0.0104 ns
$b1/y = 0.043$ ns	= 0.0344 ns
$b2/a = 0.021$ ns	= 0.0168 ns
$b2/y = 0.051$ ns	= 0.0408 ns
$b4/a = 0.032$ ns	= -0.0256 ns
$b4/y = 0.083$ ns	= 0.0664 ns

- S = -0.01ns
 - SU = 0.09ns

Data Required Time
 - **Data Arrival Time**

SLACK = + 0.028 ns

So, if (for e.g.), delay of cell b1/y was 0.043ns, after applying OCV derates of 20%, the new delay of this cell will be 0.0344ns (no magic, I have used hand calculator :)), i.e., reduced delay of this cell from its original value by 20%

And how does this affect the “data required time” and “slack”

Timing Analysis (with real Clocks)

Specifications:
 Clock Frequency (f) = 100MHz
 Clock Period (T) = 1/f = 1/100MHz = 10ns

Setup Analysis - Single Clock - Textual Representation

$\Delta_1 = b1/a = 0.013$ ns
 $b1/y = 0.043$ ns
 $b2/a = 0.021$ ns
 $b2/y = 0.051$ ns
 $b3/a = 0.032$ ns
 $b3/y = 0.055$ ns

Data Arrival Time = 1.115 ns

 $\uparrow \theta = 0.9$ ns
 $T = 10$ ns

+ $\Delta_2 = b1/a = 0.013$ ns	- 0.0104 ns
$b1/y = 0.043$ ns	= 0.0344 ns
$b2/a = 0.021$ ns	= 0.0168 ns
$b2/y = 0.051$ ns	= 0.0408 ns
$b4/a = 0.032$ ns	= -0.0256 ns
$b4/y = 0.083$ ns	= 0.0664 ns

- S = -0.01ns
 - SU = 0.09ns

Data Required Time
 - **Data Arrival Time**

Data Required Time = 1.0944 ns

SLACK = + 0.028 ns **SLACK = - 0.0206 ns**

OH MY GOD!!! (This is the real expression of an STA engineer, when he applies derates and sees the negative slack)

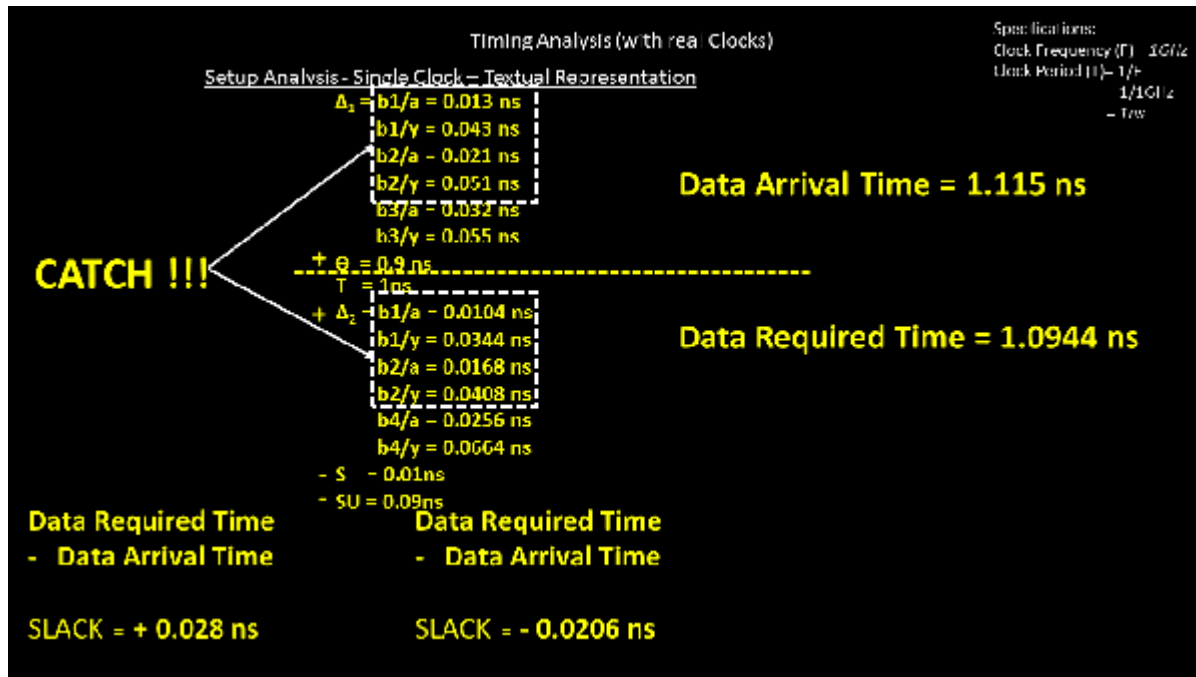
Yes, the slack is -20ps. This chip will fail... We must run it with reduced frequency. We are not meeting specs... blah...blah...blah.... which “by-the-way” are true words of an STA engineer and his/her manager

And here enters an optimistic engineer, who follows exactly what “Joseph Sugarman” believes in that “Each problem has hidden in it an opportunity so powerful that it literally dwarfs the problem. The greatest success stories were created by people who recognized a problem a turned it into an opportunity”

Even the above problem has a “CATCH”, and the engineer who identifies this catch (in his team) will, probably, have his back ‘pat’ and receive ‘congratulations awards’ :).

as this is an **important and critical one for reducing a lot of pessimism during a chip tape-out.**

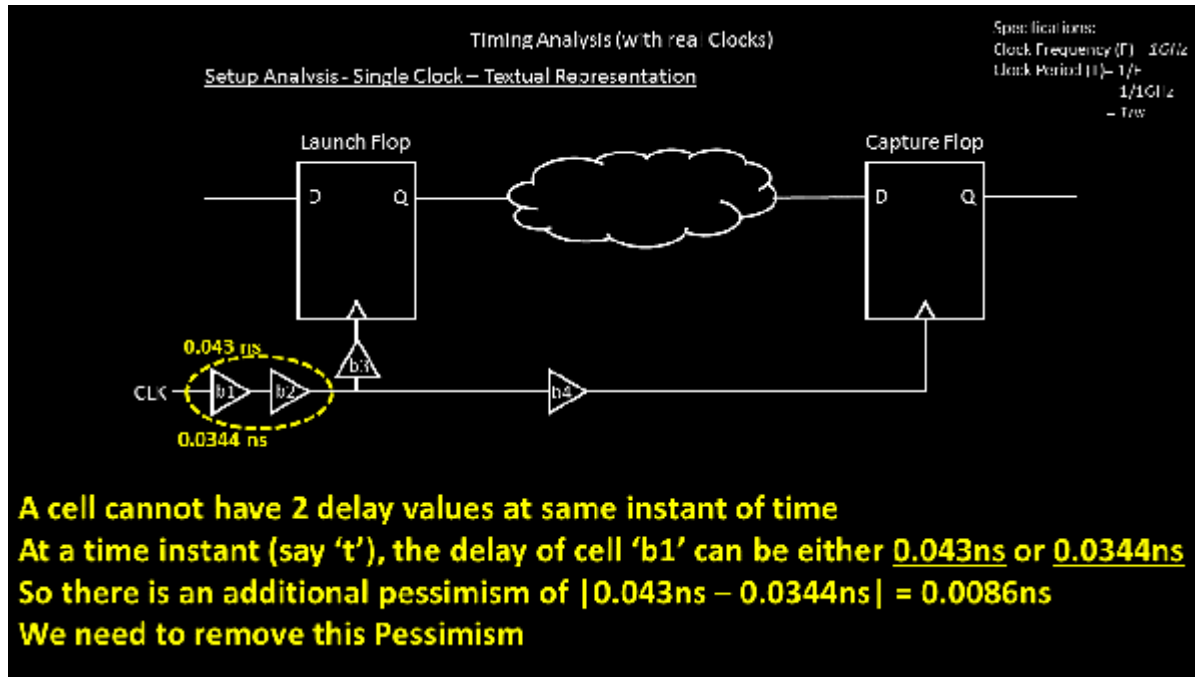
We saw **negative slack being created due to OCV derates.** But, below is the catch



And let me give you a hint. Can you run at **2 different speed at the same time instant?** Think Think

If you are a real person with 2 eyes, 2 ears, 1 nose, 2 hands, and 2 legs, you can't :). It doesn't mean that animals can ... They can't either No living being on this earth can run at 2 different speeds at same time instant 't'

Circuits (non-living being) also behave in the same way ... fortunately. Look at the below image, and with the above calculations, we are trying to say, that cell 'b1', **which is in common path of launch and capture clock, has 2 delays at same instant of time 't', i.e. 43ps and 34ps**

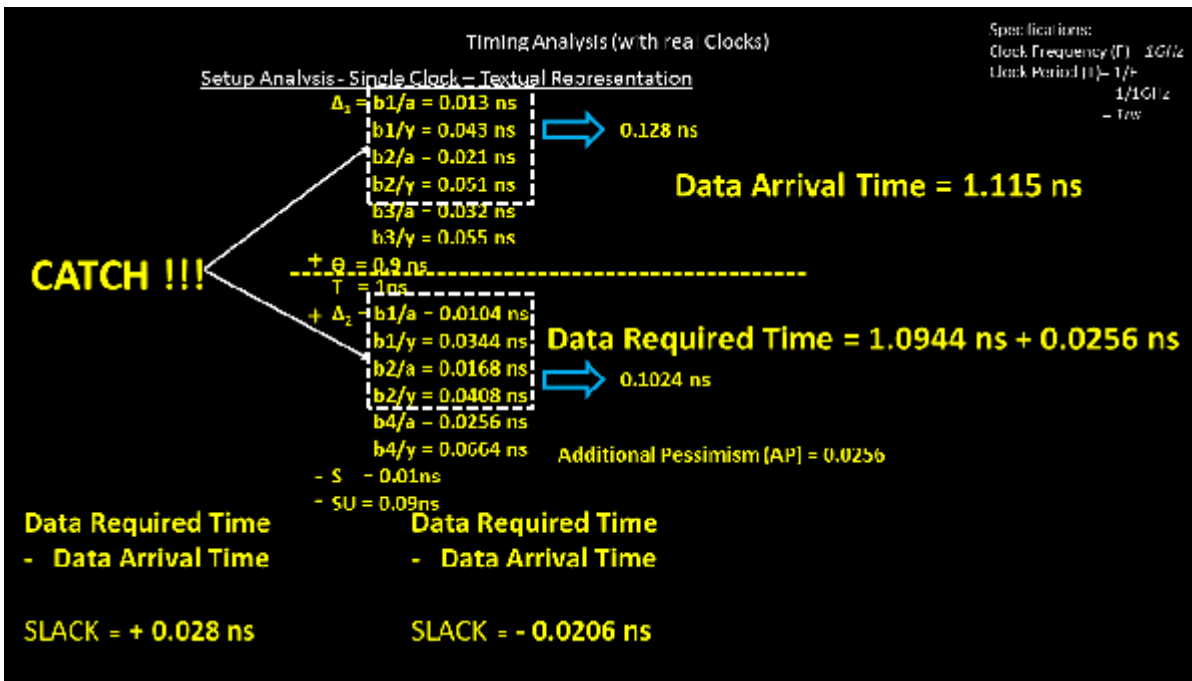


It can either have 34ps or 43ps, but not both. So, for our calculations, either we take 43ps for both OR 34ps for both, in the common clock path.

Now since, **the algorithm has already done the calculations, smart engineers came up with a simple solution, without changing the algorithm.**

What they did, they allowed the **slack calculation to happen in a traditional way**, as we have shown in above image and last post, AND, introduced a new term “**Clock Path Pessimism Removal**“, which says, remove the additional pessimism from **final Slack calculation**.

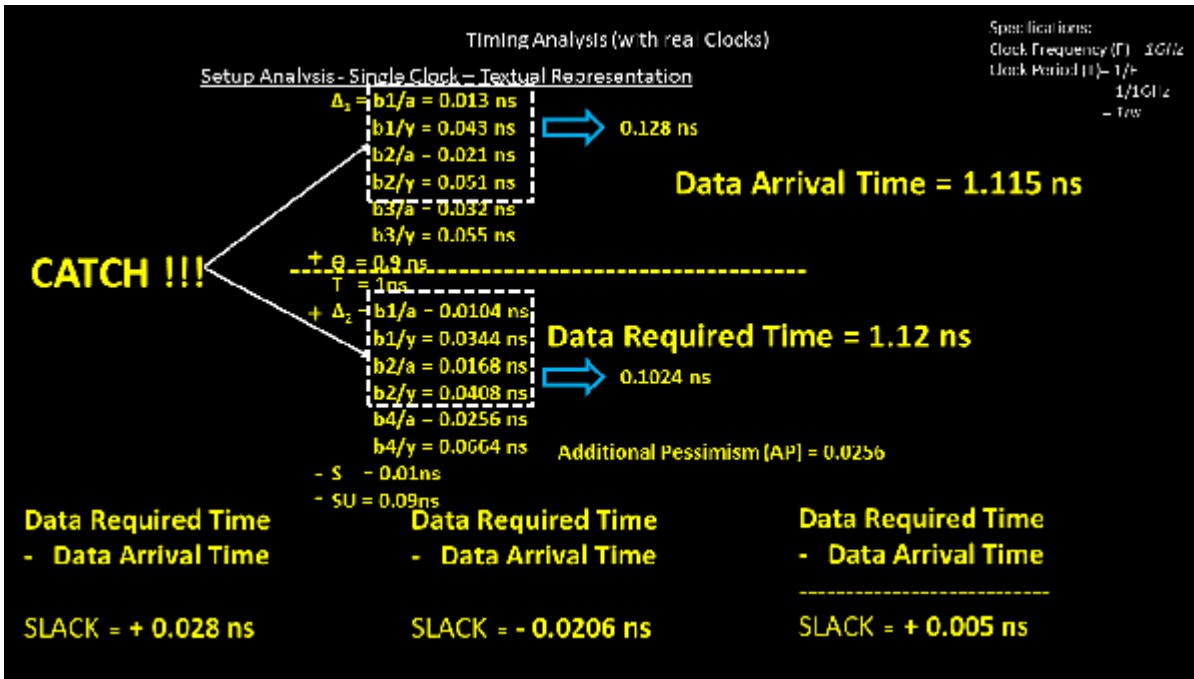
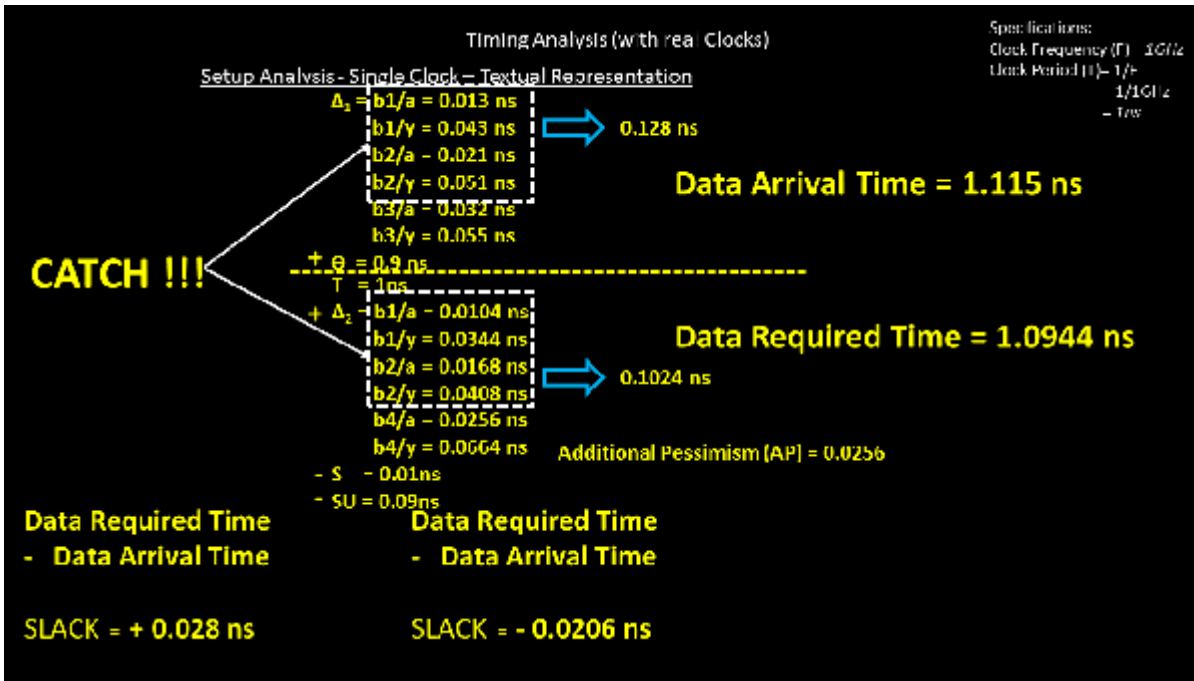
Let's see how



Above **launch clock common path** has a delay of 128ps and **capture clock common path** has a delay of 102.4ps. So, the additional pessimism is 25.6ps (again.... used a hand calculator ... I think i am pissing you off now, by saying that every time :))

The additional “25.6ps”, we can either 1) add it in “Data required time” OR 2) subtract from “Data arrival time” Why? Because, we want the “delays” in common path to be same

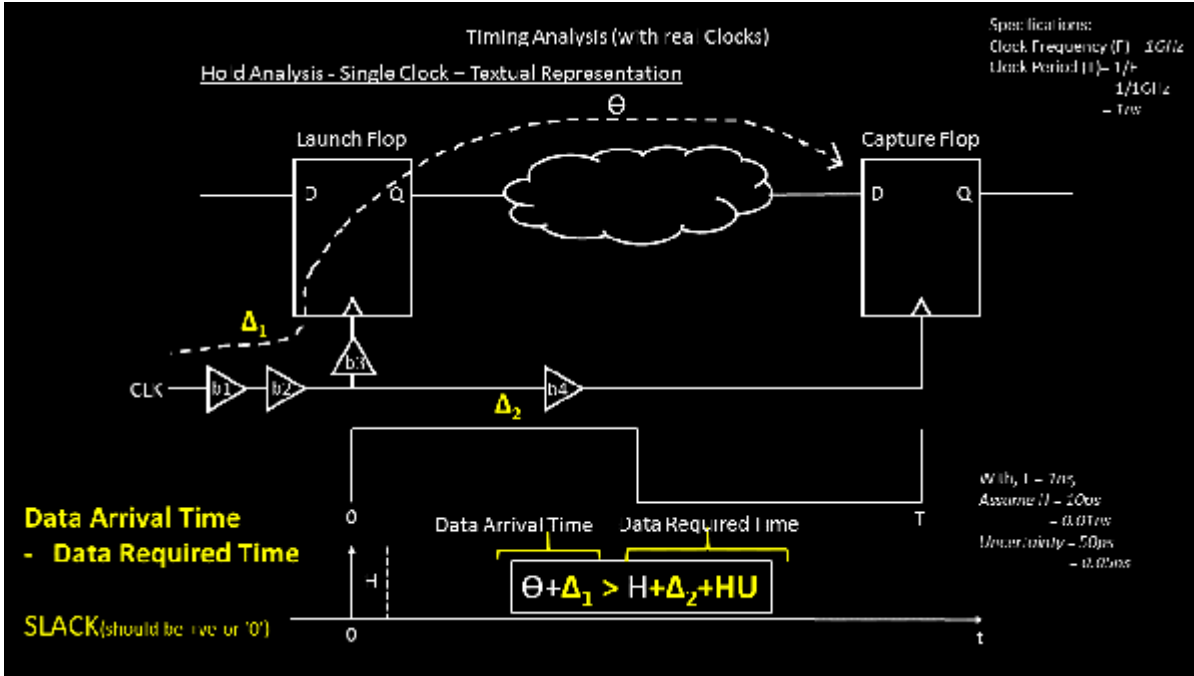
Let’s do 1) for this example and see what we get



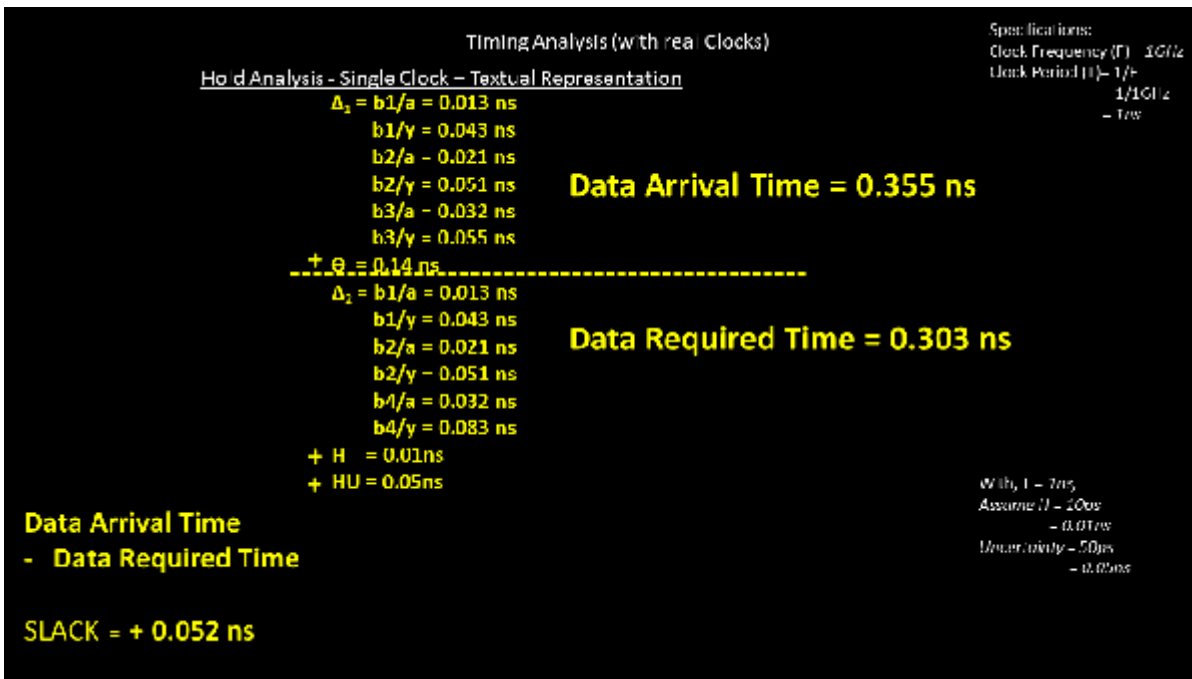
There you go without doing even a **single ECO (Engineering Change Order)**: For now, google this one, I will plan a separate post on this one) and using pure concepts, we were able to meet this path and attain the required frequency.

And you thought we are done with CPPR... No ... not yet ... **We haven't done the "Hold" analysis yet.** It's simple, but it's tricky. Why it's simple? It's because of the amazing images that I use to describe things :).

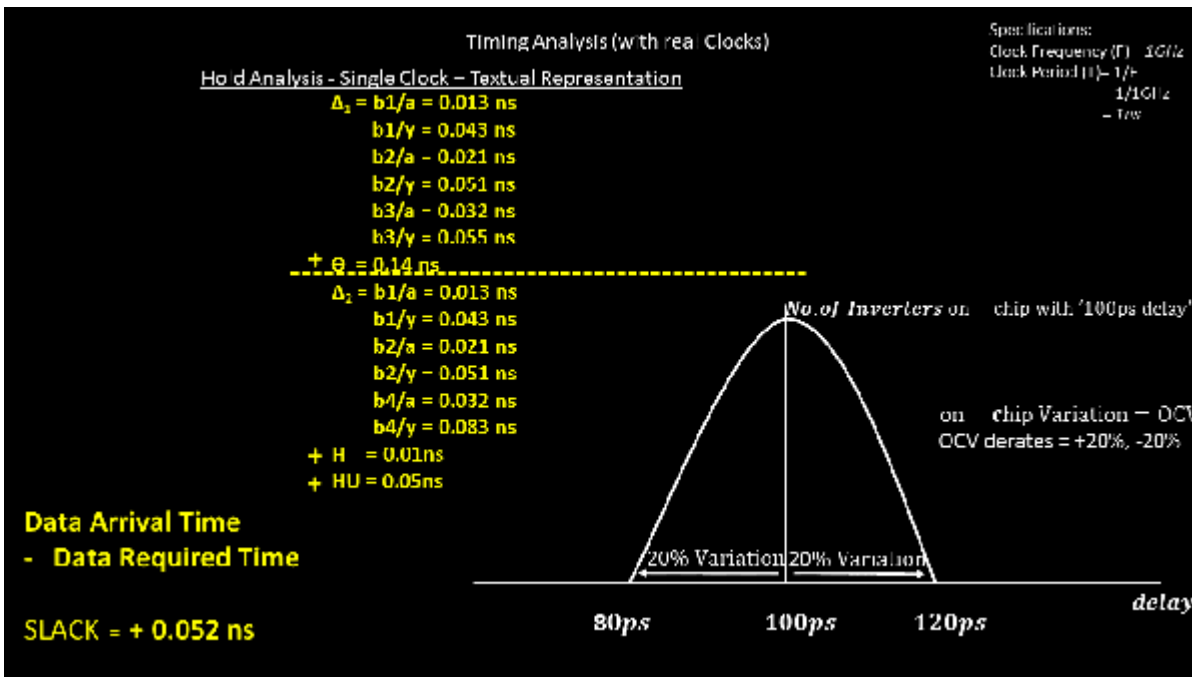
Why it's tricky? Let's watch below. **It's now 'data arrival' – 'data required' that needs to be positive, in contrast to 'setup analysis'**



With the below values assumed for the cell and net delays, we **get a positive slack**. Note: **we haven't accounted for OCV derates yet**

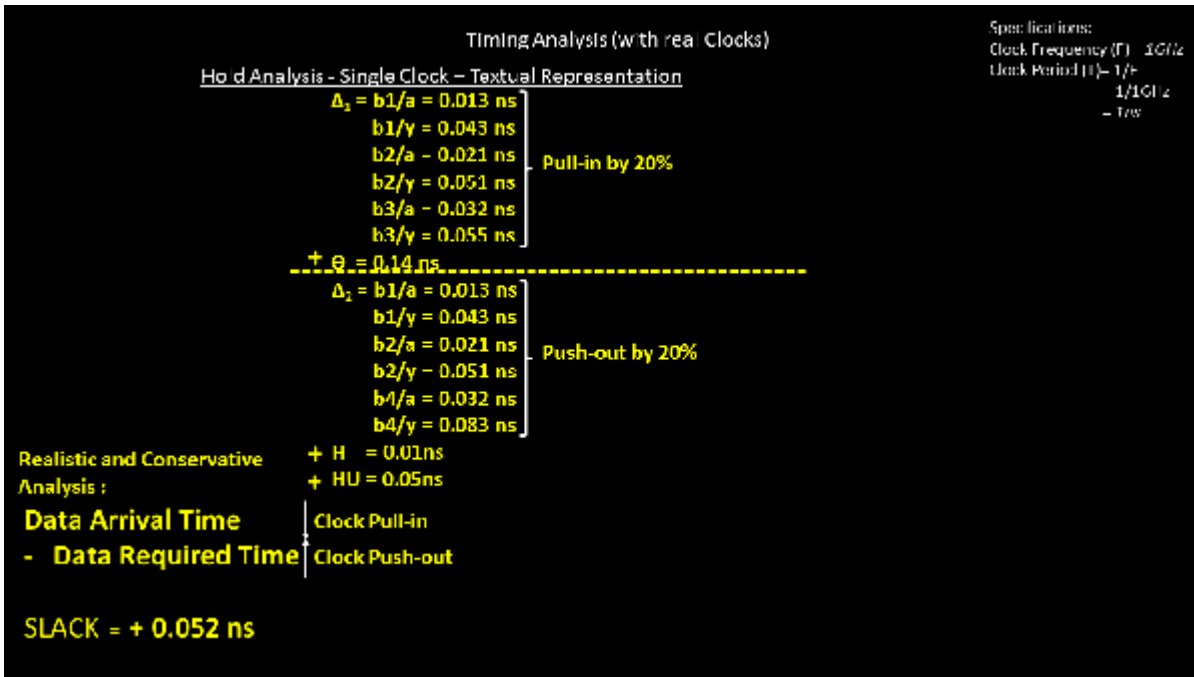


We will, again, assume a 20% variation for OCV,



and do a more conservative hold analysis again, to observe how the positive slack becomes negative. This time we will 'Pull-in' the launch clock by 20% and 'Push-out' the capture clock by 20%.

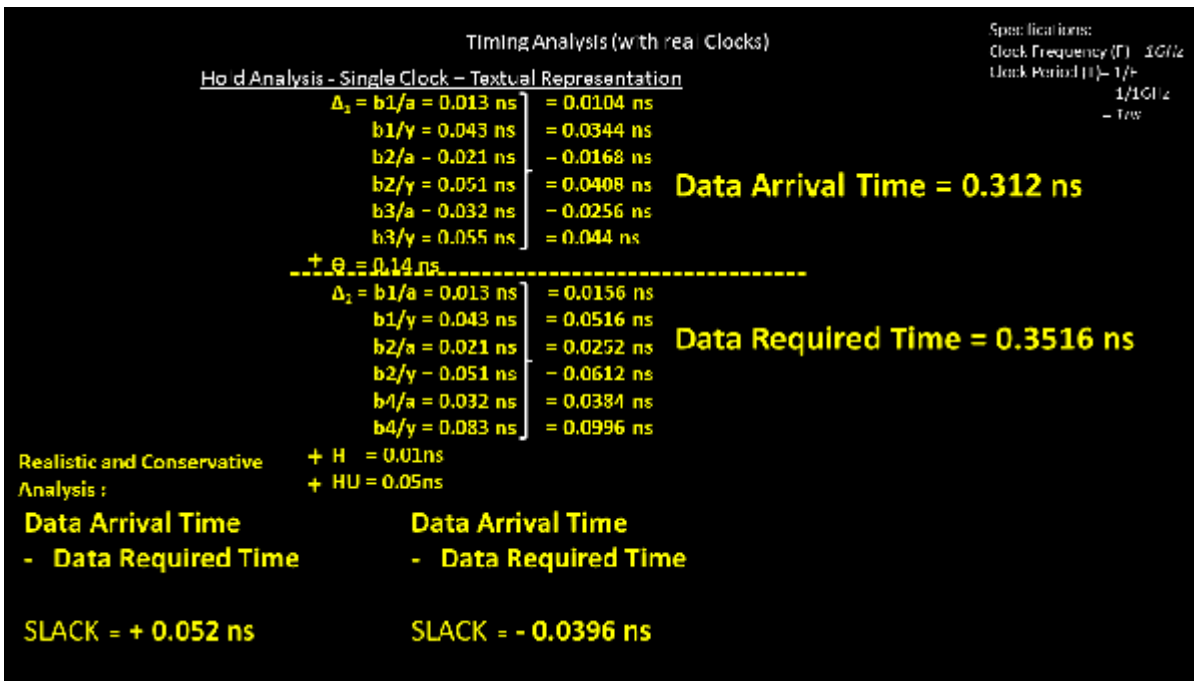
Why we do this? We just need to be extra careful about hold analysis, and it's just a single edge check. I will get back to this in my upcoming post



Here we get a negative slack, and a negative slack in hold is like your LIFE. Needs to be taken seriously :).

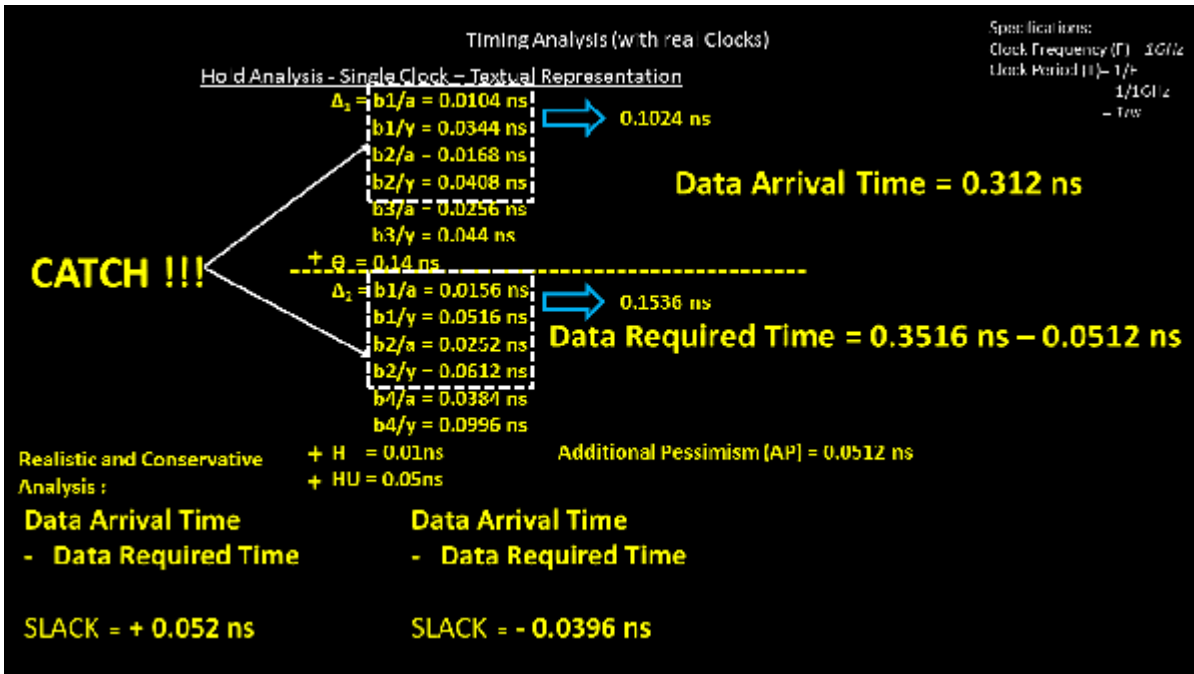
There are still ways to recover from a setup violation, but there are **no ways to recover from a hold violation** (in a specific PVT corner). I will talk more about this in my upcoming posts.

Below shows original and 20% derated delay side-by-side

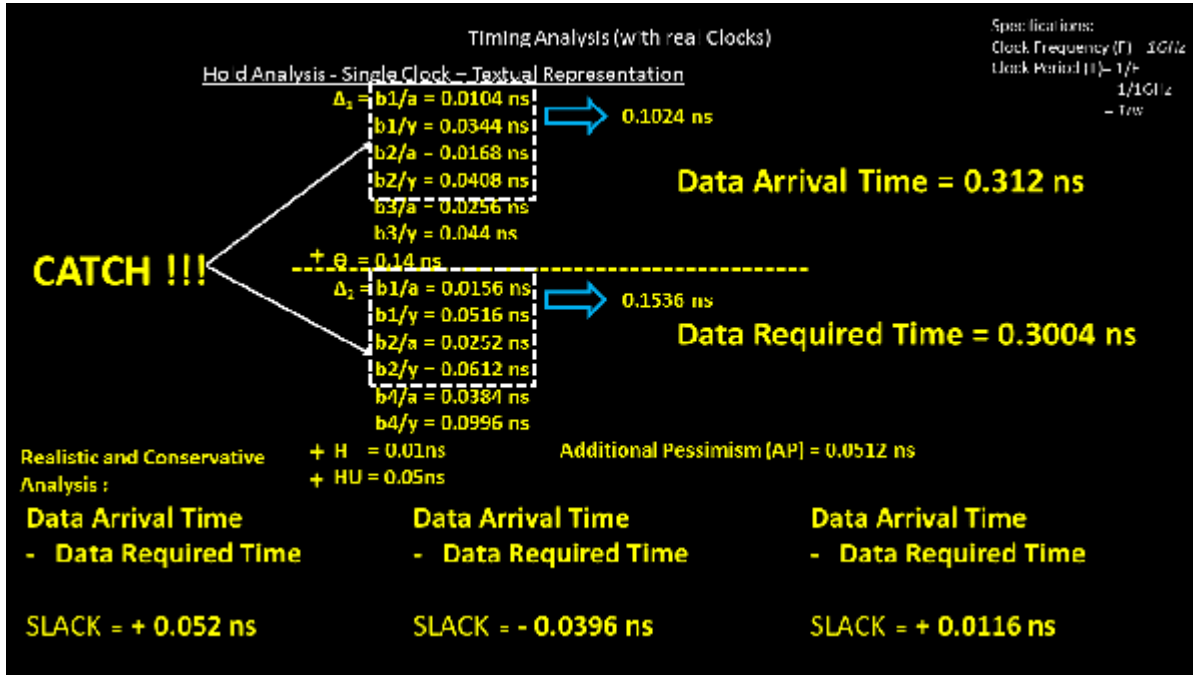


Ahh... STA engineers just hate this part of -ve hold violations. And it's annoying, if this is seen towards end of release.

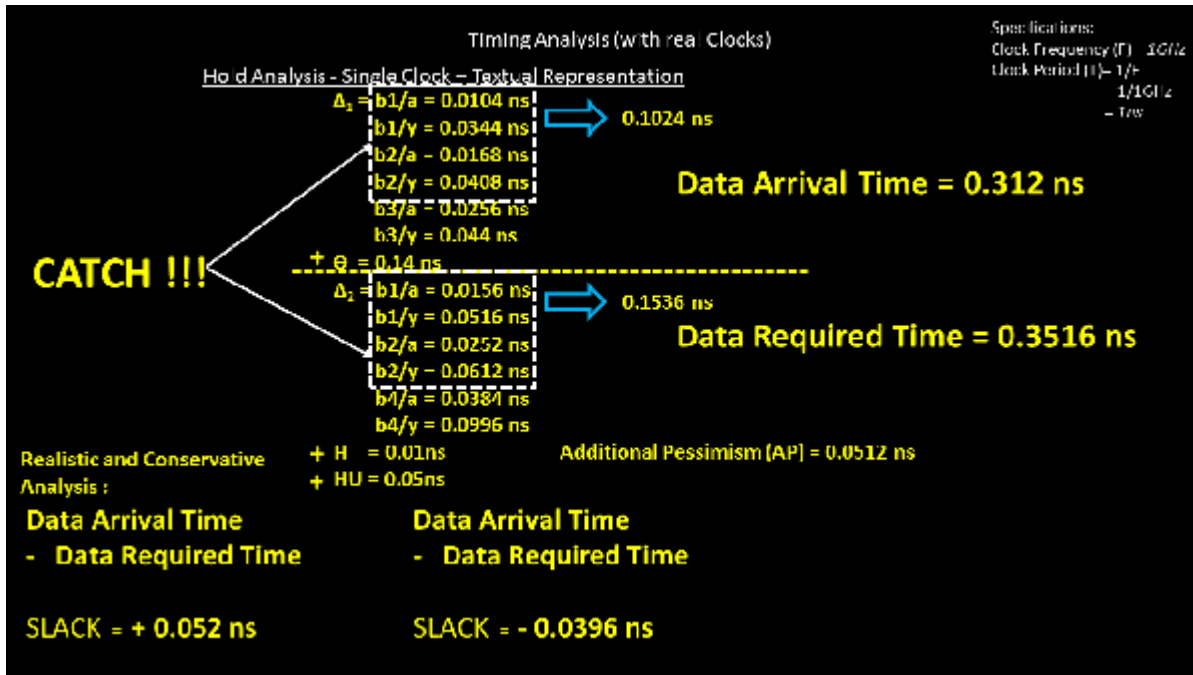
But Hello Catch Common Clock Path 2 different delays not possible blinks a light previous post



The happy part, we will remove the additional pessimism,



And.... Bang There you go ... you just got rid of those nasty hold violations, smartly



This **completes the basic CPPR**, and should be good enough to get you started with your critical timing analysis. And if these concepts on CPPR help you, don't forget to send a note to me

And if this is not enough, go through my **CPPR videos on YouTube**....

<https://www.youtube.com/playlist?list=PLUSK3BZWA60uEbKzWP6SST8MwkQ-0GOrs>