

# Clock tree synthesis CTS (V1)

Written by: Fatma Ali

## Content:

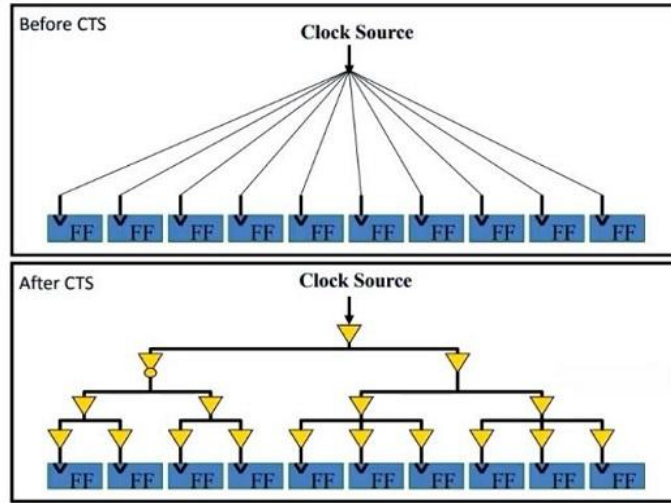
- Introduction
- Launch and capture clock edges
- Clock parameters (Jitter - skew - transition - latency - uncertainty)
- CTS Vs High fanout synthesis (HFS)
- Clock tree DRCs and targets
- Clock tree exceptions (source and sink points)
- Cross talk
- Non default routing rules
- CTS effects and optimizations
- ICC tool
- Important notes

## Introduction:

احنا لحد هنا كده حطينا cells فى اماكنها والدنيا خلاص جاهزة اننا نوصلهم ببعض يعنى ندخل فى routingبقى ... بس فيه حاجة مهمة اوى قبل ال routing لازم نعملها وهى اننا نحط clock network ... احنا لحد دلوقتى بنتعامل مع clock على انها ideal وبمنع ال tool انها تحط اى cells فى طريق ال clock لل flops من خلال set\_dont\_touch command زى ما شوفنا من ال synthesis ... وبالتالى كنا بنضيف شوية constraints عليها بحيث نحاول نعمل modeling لل clock على الواقع زى skew, latency, uncertainty ... فى الخطوة دى بقى احنا هنحط ال clock network يعنى هنشوف ال clock الى واصلة على flops عندى وكنت بخلى ال tool متلمسهاش دلوقتى هخليها تعملها الى اصلا وتحط buffers فى طريقها

طب اول سؤال ليه نحط buffers ومنوصلش على طول بين ال clk port وال clk pin at flops؟؟ ده جاوبنا عليه فى introduction to PnR بس خرينا نقول الاجابة تانى دلوقتى ... مقدرش اوصلها بسلك عادى لل flops لانها critical signal انا محتاجها توصل فى نفس الوقت تقريبا عندهم كلهم ولكن اكيد التوصيل ب wires عادية هيجلى يحصل delay لكل واحد بشكل مختلف حسب بعده عن ال clock port وحتى لو ضمنت انى هوصلهم بشكل يخلى ال delay بينهم واحد فاكد ال signal نفسها هتضعف وهى ماشية فى السلك الطويل ده يعنى transition time بتاعها هيبقى كبير وكده ال clk هتبقى سيئة جدا ودى اهم signal عندنا فالحل ايه؟؟

الحل انى احط فى طريقها buffers & inverters تقويها طول ما هى ماشية وفى نفس الوقت اعمل balance بين طريق ال clk لكل flop بحيث ميكنش فيه delay لوصولها بين كل flop والتانى بقدر الامكان ودى الى كنا فى اعتبارنا فى ال timing analysis بنسميها skew ... فكد من ال clk port هبدأ اطلع network منها بتروح لكل ال flops وفى كل طريق منهم بحط buffers & inverters حسب الى محتاجة الطريق ده عشان اعمل balance بين كل الطرق بتاعة ال clk ... عشان كده اطلقوا عليها clk tree لان من ال clk port كإنى بعمل tree كده وافرع منها كل شوية فرع جديد زى فى الصورة دى بتوضح الفرق بين من غير CTS وب CTS:

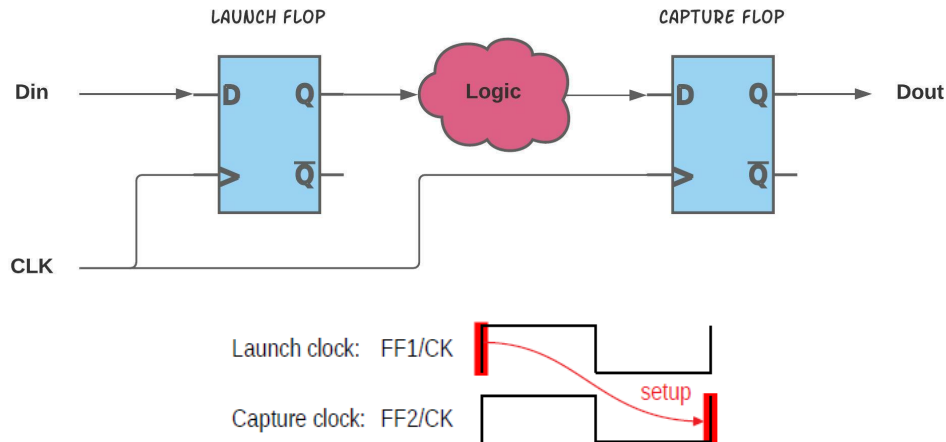


حاجة مهمة هنعملها فى CTS بردوا وهى routing لل clk tree يعنى هنوصل ال network دى ببعضها بالفعل هنا قبل ما نروح لل routing process وده لان ال clk زى ما قلنا هى ال اهم عندنا يبقى اوصل ال network بتاعتها الاول ثم باقى ال routing resources او بمعنى اخر الاماكن المتبقية الى اقدر اعمل فيها routing اخلوها لل cells routing لكن مش هخلى ال clk الى هى اهم signal عندى مع الباقي عادى فى routing ... لا انا هسيبلها الدنيا براحتها تتخط باحسن شكل الاول ثم احط الباقي

كده احنا عرفنا بشكل مبدئى احنا هنعمل ايه فى الخطوة دى وده هيتعمل ب commands من ال tool عادى بس فيه concepts مهمة جدا لازم نكون عارفينها وهى الى هنتكلم عنها قبل ما ندخل فى مرحلة ال tool فأول حاجة خلينا نشوف ال parameters بتاعة ال clock نفسها ونتكلم عن ال clock فى ال design ونفهم كل حاجة خاصة بيها بحيث نبقي عارفين tool commands دى بتعمل ايه وكمان لما نطلع مشكلة نعرف سببها ونعرف نحلها ازاي.

## Launch and Capture clock edges

خلينا نفكر الاول how the system behaves with a clock ... احنا عامة بنبص على ال design على انه عبارة عن flip flops وبينهم combinational logic ودى حاجة اكيد اتعرضتلها قبل كده وانت بتدرس Static Timing Analysis STA فدلوقتى انا بيبقى عندى flop بسميه launch flop يعنى ده الى طالع منه ال data دلوقتى والى بعده بسميه capture flop لإنه ده الى هياخد من الاول ال data الى هيطلعها:



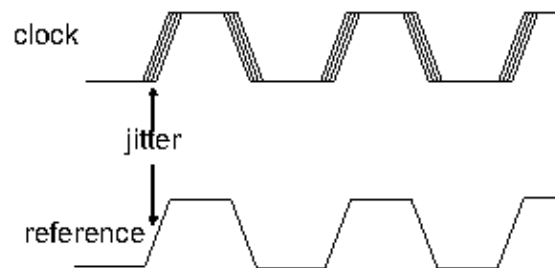
بما ان نفس ال clk رايحة للاتنين فمثلا انا عند positive edge معينة هيت trigger ال two flops بس هتبقى دى launch edge بالنسبة الى دلوقتى على اساس launch flop طلعت data ثم ال positive edge الى بعدها هتبقى capture edge على اساس ال capture flop هو هياخد الناتج من ال data الى طلعت فى ال launch

طبعا الكلام ده بينطبق على كل path بمعنى ان launch flop هنا هو capture flop لل flop الى قبله وال positive edge الى يتمثل launch هنا هي capture لل قبلها وهكذا ... يعنى انت بتبص على two flops وتقول ده launch وده capture لكن اى flop هتلاقيه عند كل positive edge بي launch ال data بتاعته وي capture ال data بتاعة ال flop الى قبله

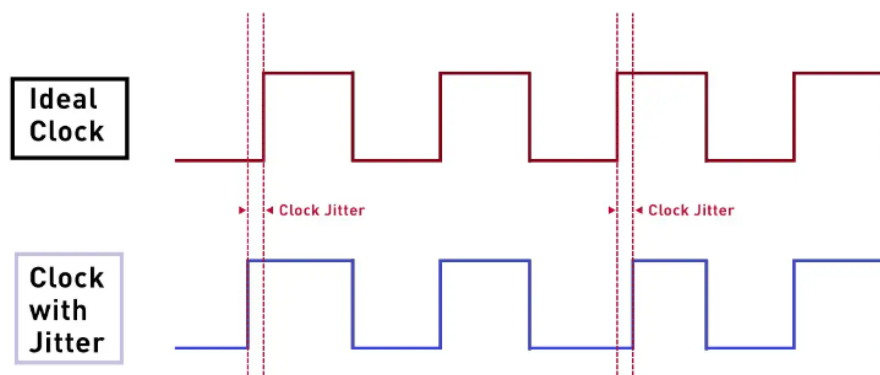
## Clock parameters:

### 1. Clock jitter:

هو عبارة عن الفرق بين ال positive edge of the clk لنفس ال flop يعنى هنا ببص على طريق واحد لل clk الى هو واصل على flop معين عندى وبشوف الفرق بين كل positive edge والى بعدها ... طب ما الفرق ده المفروض Tperiod عادى ... لا ده كلام ideal لان بيكون فيه فرق بسيط بين كل positive edge والى بعدها ففى الواقع ال positive edge مش هتيجى كل Tperiod ممكن بعدها بشوية او قبلها بشوية ومرة تيجى مضبوطة بس المهم ان الفرق ده يكون acceptable فى الاخر وميعملش Timing violations ... وطبعا ده بردوا هيعمل نفس الكلام بالنسبة لل negative edge ... الصورة الى تحت دى بتوضح الفرق بين ideal clk وبين لما اخذ فى اعتبارى jitter

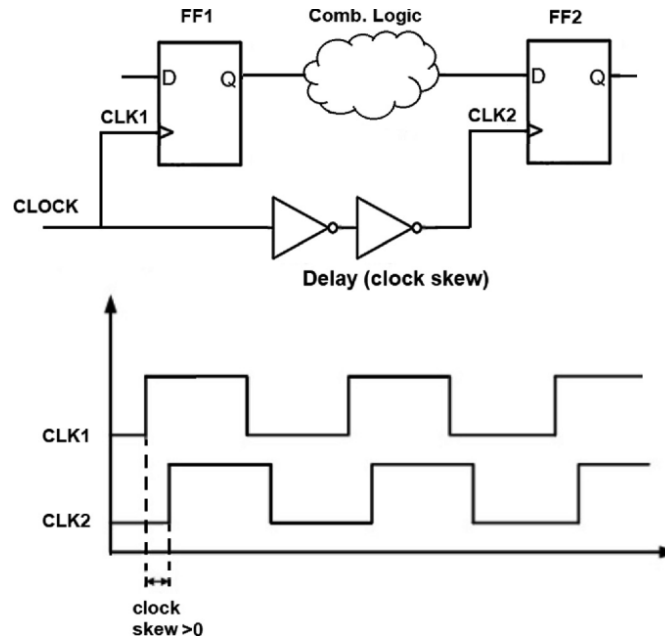


ال clock الى فى صورة فوق الى ب jitter هو مش قصده ان بيكون شكلها كده هو يقصد ان ال positive edge ممكن تبقى فى كذا مكان فى المرة الواحدة يعنى هو ال signal بتبقى زى الصورة الى تحت دى ال edges كل مرة بتبقى مختلفة عن ال ideal بطريقة مختلفة مرة قبلها ومرة بعدها ومرة تبقى مضبوطة:



## 2. Clock skew:

ده الفرق بين positive edge of the clk بس بين two flop مختلفين يعنى بمعنى اصح هو الفرق بين وصول ال clk ل two flops يعني هنا ببص على طريقين لل clk كل طريق بيودي ل flop واحد واشوف ال positive edge وصلت لكل واحد امتى والفرق بين وصولهم هو ده ال skew وده بيكون جاي من ان طريق منهم فيه delay اكبر من التاني زى فى الصورة كده ... الطريق ل FF2 فيه delay جاي من two inverters



$$T_{skew} = \text{Capture clock path delay} - \text{Launch clock path delay}$$

$$T_{skew\_max} = \text{max capture clock path delay} - \text{min launch clock path delay}$$

$$T_{skew\_min} = \text{min capture clock path delay} - \text{max launch clock path delay}$$

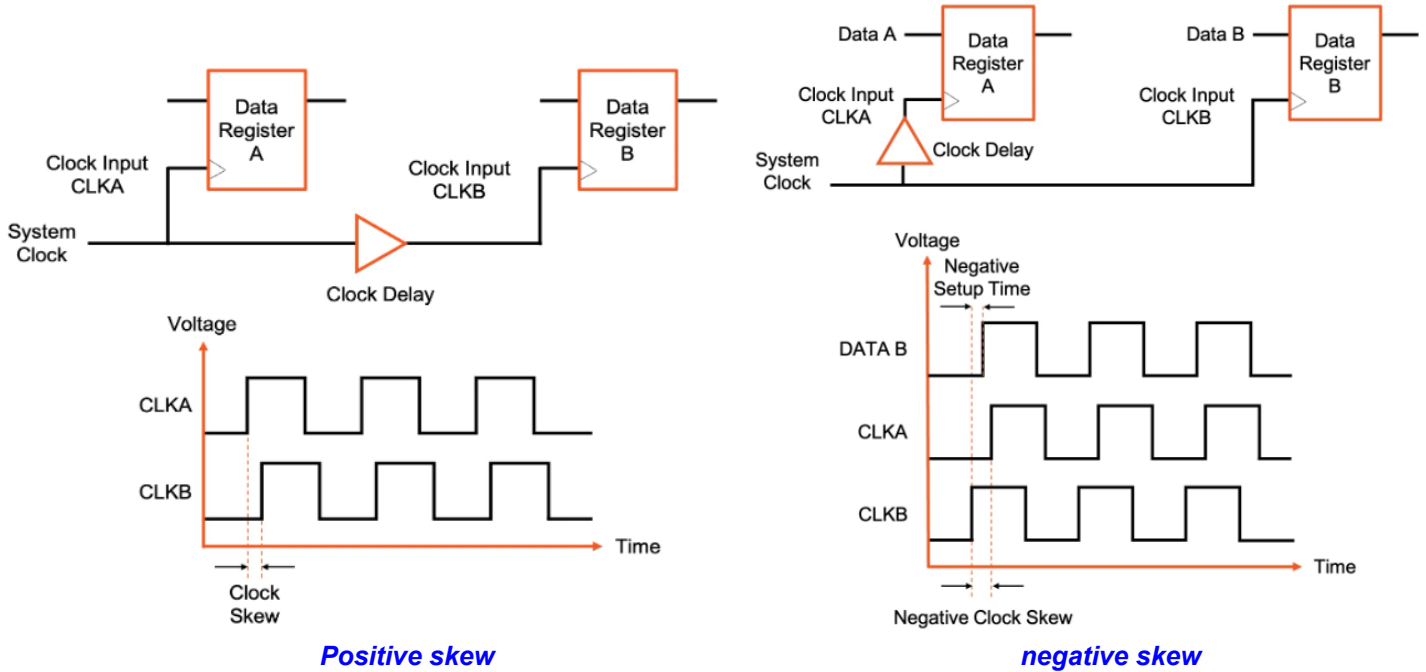
ال clock path هو عبارة عن buffers and inverters فزى اى cell ليهم max and min delays فعلى حسب بقى انا لو عايز اجيب ال max clock path delay هستخدم ال max delays بتاعة ال cells فى ال path لكن لو عايز ال min clock path delay هستخدم ال min delays بتاعة ال cells فى ال path

ال skew بيضاف فى معادلات ال setup & hold وبيكون مؤثر عليهم ... فى ال setup هلاقى انى غيرت فى clock period حسب الصورة فوق مثلا هلاقى ان combinational logic بقى عندها وقت تتغير فيه مش بس clock period لا كمان هيزود عليه delay الى هو ال skew لان ال edge بتاعة ال capture flop بتوصل متأخرة عن launch فمعايها وقت اكبر من ال clock period (وطبعا فى جميع الاحوال سواء كانت Tclk او Tclk+Tskew فبطرح منهم Tsetup وده الوقت ال max ل ال combinational delay logic زى ما احنا عارفين من STA ... بناكد على المعلومة بس) ... فى ال hold بردوا فى نفس الصورة الى فوق يبقى انا بقى عندى hold time اكبر او الاصح يعنى لان ال hold time ثابت بس المقصود ان عندى وقت تانى زاد عليه وهو ال skew ... فى setup أسوء حاجة للمعادلة انه يكون skew اقل قيمة والعكس فى ال hold

- **Setup equation:**  $T_{c2q} + T_{comb\_max} + T_{setup} \leq T_{clk} + T_{skew\_min}$
- **Hold equation:**  $T_{c2q} + T_{comb\_min} \geq T_{hold} + T_{skew\_max}$

اول تصنيف لل skew سيكون:

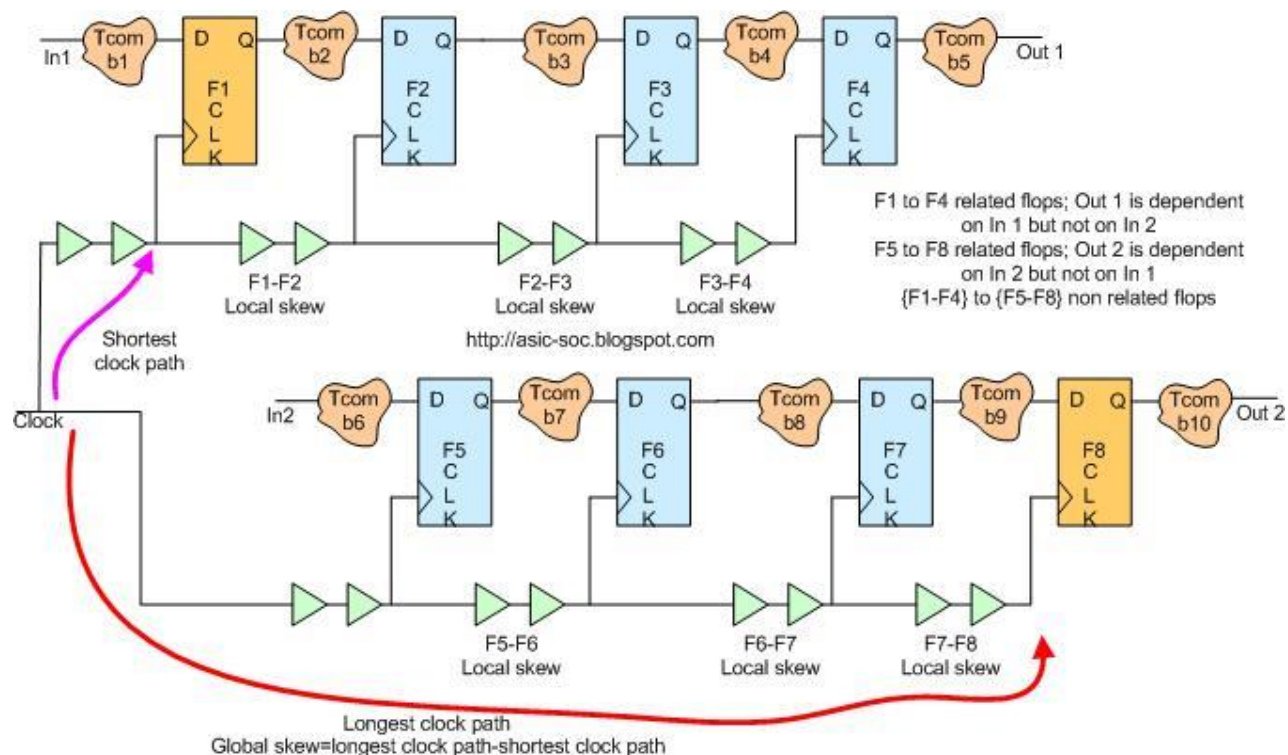
- **Positive skew:** وده لما تكون capture clock هي الى متأخرة عن launch clock وده المثال الى شوفناه فوق وشرحنا عليه ال skew فهنا قيمته positive فهلاقي انه في معادلة setup سيكون احسن وهيزود ال setup slack ولكن العكس في معادلة ال hold سيكون اسوء وهيققل ال hold slack
- **Negative skew:** وده لما تكون launch clock هي الى متأخرة عن capture clock فهنا قيمته negative فهلاقي انه في معادلة setup سيكون اسوء وهيققل ال setup slack ولكن العكس في معادلة ال hold سيكون افضل وهيزود ال hold slack



- **Positive skew improves setup but can lead to hold violations**
- **Negative skew improves hold but can lead to setup violations**

تاني تصنيف لل skew سيكون:

- **Local skew:** ده ال skew بين two flops ليهم علاقة ببعض اوبمعنى اصح بيوصلوا لبعض داتا يعني هما في نفس ال data path وده الى احنا شرحناه عليه كل ال skew الى فات في الامثلة الى في الصور وزى الصورة الى تحت هلاقي اي local skew بين two flops في نفس ال data path
- **Global skew:** ده بيكون اكبر skew في ال design يعني الفرق بين longer clock path و shorter clock path حتى لو بين two flop ملهمش علاقة ببعض يعني كل واحد فيهم في data path مختلف زى في الصورة تحت اقصر clock path هو بتاع F1 واكبر clock path بتاع F8 وكل واحد في data path مختلف ... فالفرق بين ال clock paths دي هي global skew

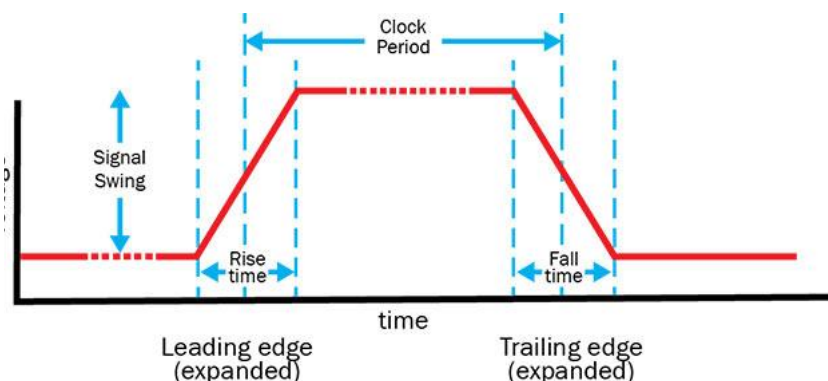


بما ان CTS زى ما قلنا هدفها انها تعمل balance بين clock paths delay لل flops (الى شغالة على نفس ال clock او بمعنى اخر فى نفس ال clock domain) **يبقى ممكن نقول ان CTS بيكون هدفه انه يخلي Global skew = 0 لكل clock domain موجود فى ال system**

=====

### 3. Clock transition (Slew):

ال clock زى اى signal فى الواقع بيكون ليها rise time & fall time ومش بيكون عندها sharp transition ولكن كل ما كان ال transition قريب من ال sharp كان افضل وخاصة ال clock دى اهم signal عندنا عشان كده حتى ال buffers الى بنستخدمها فى ال tree بتكون مخصصة لل clock عندها high rise and fall transition يكون اقرب حاجة لل sharp يعنى transition time صغير بتطلع بسرعة وتنزل بسرعة لان دى الى معتمد عليها ال system كله اصلا ... ال buffers دى بيكون اسمها symmetric buffers عشان ال rise transition & fall transition دول بيكونوا قد بعض ... وبنستخدم ال buffers دى لل clock tree فقط



=====

#### 4. Clock latency:

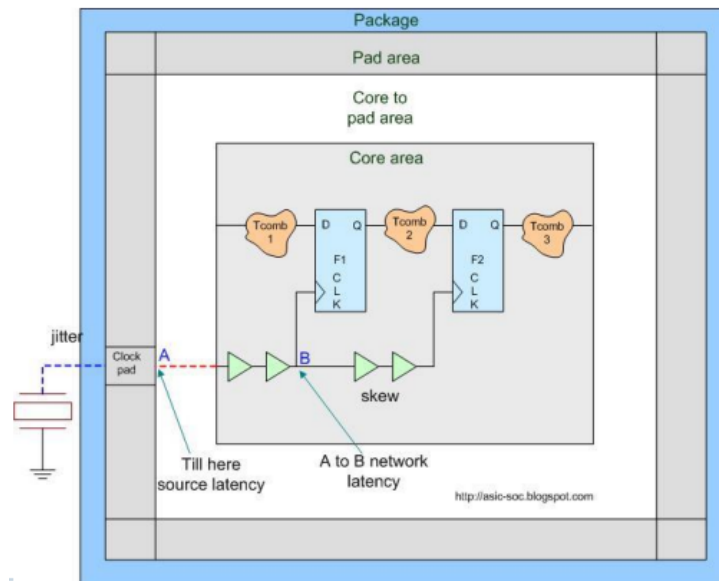
دى عبارة عن مجموع حاجتين:

##### 1. Network latency (delay) or Insertion delay:

ده ال delay من ال clock port فى ال system لحد ال register clock bin يكون مشترك بين كل ال clock paths بمعنى هلاقى delay فى بداية ال clock network مشترك بين اى clock path فيها ثم بعد كده كل path هيبداً ياخذ delay مختلف عن التانى فال delay المشترك ده هو ال latency

##### 2. Source latency (delay):

ده ال delay من ال clock source لحد ال clock port بتاع ال system يعنى انا ال system بتاعى ليه clock definition point الى ال input port بتاع ال clock طب هى ال clock اصلاً هتيجى منين؟؟ اكيد هتيجى من block تانى هى generate ال clock وليكن PLL فده كده ال clock source الى هيطلع ال clock signal اصلاً ... ثم اكيد فيه مسافة هتمشيها ال clock بعد ما تطلع منه لحد ما توصل لل definition point بتاعتها الى انا محددها فى ال design بتاعى فالمسافة دى هى ال source latency



$$\text{Clock latency} = \text{Network latency} + \text{source latency}$$

=====

#### 5. Clock uncertainty

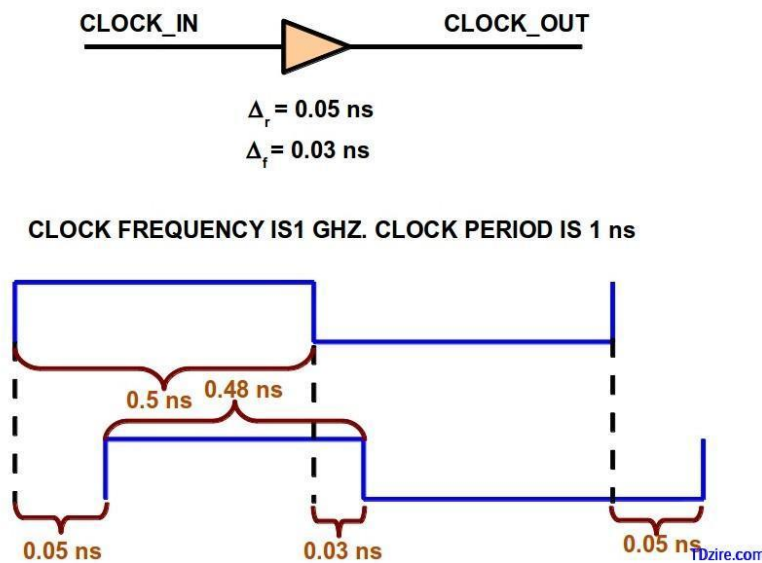
يعنى حاجة مش متأكد منها فدى بتشمل كل الحاجات الى انا مش متأكد منها فى clock فمدام حاجة مش متأكد منها تبقى مش موجودة معاي دلوقتى ومقدرش احدها فلو انا قبل CTS فانا لسه معملتش ال network فمعرفش بعد ما اعملها هلاقى skew قد ايه ولا network delay قد ايه وطبعا source latency دى حاجة برا ال design بتاعى معرفهاش او حتى لو عرفتها (من system engineer مثلاً) فانا لازم احطها لل tool بنفسى هى مش هتعرف تجيبها لوحدها فبردواك tool هى متعرفهاش وطبعا ال jitter بردوا مش معروف ... طيب لو انا بعد CTS هكون خلاص حطيت ال clock network فكده ال tool عرفت skew & network latency وبالتالي ال uncertainty هنا هتكون عبارة عن clock source & jitter فملخص الكلام ده فالمعادلتين دول:

- Pre CTS: Uncertainty = source latency + network latency + skew + jitter
- Post CTS: Uncertainty = source latency + jitter



## CTS Vs HFS

من الحاجات الى ممكن تيجي فى بالنا ان CTS بتشبه شوية فكرة high fanout synthesis الى اتكلمنا عليها فى placement ... هما متشابهين فى فكرة اننا بنحط buffers & inverter فى طريق الاتنين بس الفرق بينهم فى نوع buffers & inverters ... ال HFS بيتعمل ل static signals يعنى ال signals switching بتاعها مش عالى ولكن عندها fanout كبير زى مثلا reset او scan enable فى DFT دول بيوصلوا على كل ال flops الى عندى ولكن لإنهم static فانا اقدر استخدم buffers & inverters عندهم rise and fall time عادين بمعنى انهم مش لازم يكونوا متساوين ممكن يكون فيه اختلاف فى rise & fall time لكن فى CTS دى buffers و inverters هنتحط فى طريق ال clock فلو rise & fall time مش متساويين بالظبط هلاقى duty cycle مبقاش 50% مضبوطة وطبعاً ده هيعمل duty cycle violation وده هياثر على ال timing فى ال design كله



فى الصورة فوق بتورينا لوحطينا buffer عادى مش symmetric يعنى rise & fall time مش متساويين هتلاقى duty cycle مش 50% لأن الوقت الى clock خدته عشان تطلع high مش نفس الوقت الى خدته عشان تنزل low فهتلاقى بدل ما كان high pulse المفروض تبقى 0.5ns بقت 0.48ns

## Clock tree DRC and targets

ال tool وهى بتعمل clock tree بيبقى عندها constraints لازم تحققها وهى دى **Design rule constraints DRC** فلو حاجة منهم متحققتش هيطلع violation ودول هما:

**Max. transition delay, Max.load capacitance, Max. fanout, Max. buffer levels**

زى اى signal هيكون ليها max transition متزدش عنه ... واى node فى ال network ليها max load cap. & max fanout نفس الفكرة الى كنا بنعملها ل design signals عادى يعنى ... وال network كلها ليها max buffer level يعنى اكبر عدد buffers يتحط ورا بعضه فيها ... فكل ده منطقى يكون constraints لازم تتحقق وهنلاحظ ان دى ملهاش علاقة بال balancing فى clock tree هى بس بتحط rules نمشى عليها لكل node ولل signal يعنى متكلمتش عن حاجة تربط بين ال paths وبعضها عشان نعمل بينهم balancing

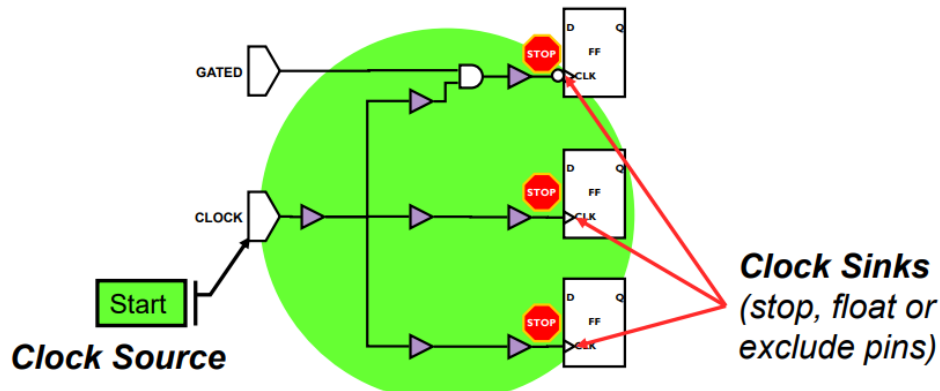


عندنا بقى **targets** ودى بتكون حاجات كويس لو اتحققت بس مش هتعمل violations لو متحققتش وهى:

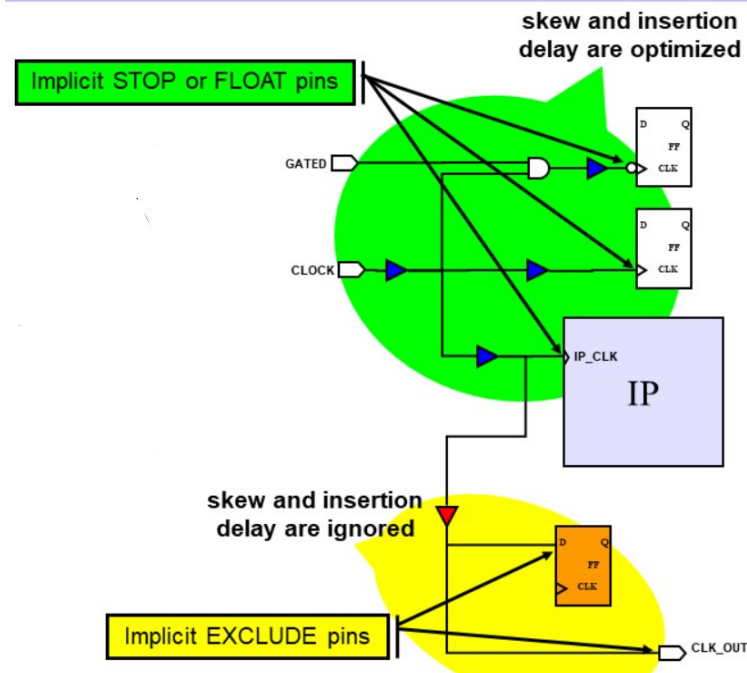
### Max. global skew, Min.&Max. insertion delay

هنا بقى فى ال **targets** هتلاقى ال **balancing** الى بندور عليه لإن ال **tool** فى ال **tree** زى ما قلنا قبل كده هى بعد ما تحط & **buffers** **inverters** هتبقى عملت ال **skew** و ال **insertion delay** الى هو ال **clock network** فهما دول الى المفروض يتحطوا بحيث يعملوا **balancing** بقدر الامكان فى ال **tree** ... ممكن تستغرب انهم المسؤولين عن ال **balancing** بس مش بيطلعوا violations وده منطقى لإن ممكن ميكونش ال **balancing** مضبوط 100% بس ال **timing analysis** تمام يعنى نسبة ال **balancing** كده **acceptable**

### Clock tree exceptions (source and sink points)



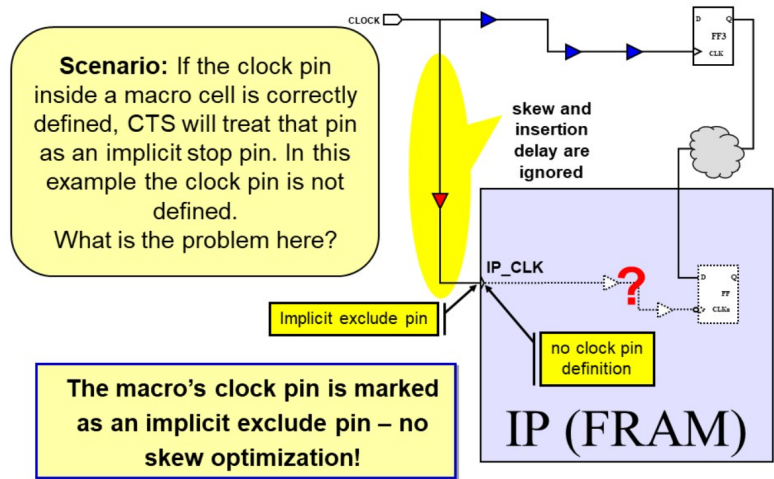
ال **clock tree** ليها بداية ونهاية فى كل **path** وال **tool** بتعمل optimizations فى كل **path** سواء فى **DRC** او ال **targets** اى **path** لازم تهتم فيه ب **DRC** بس ال **targets** ممكن متبقاش مهمة فى بعض ال **paths** وهنشوف ازاى ... البداية هى **source point** ودى إما هتعتبرها **clock signal output from source** الى هى خارجة من PLL مثلا او انك تعتبرها **clock definition point** الى فى ال **design** بتاعك ... ايا كان دى الى هتبدأ من عندها ال **tree** لكن النهاية بنسميها **sink point** يعنى كل **path** فى **tree** هو بدأ بال **source point** بس انتهى فين؟؟ عندى ثلاث انواع من **sink point**:



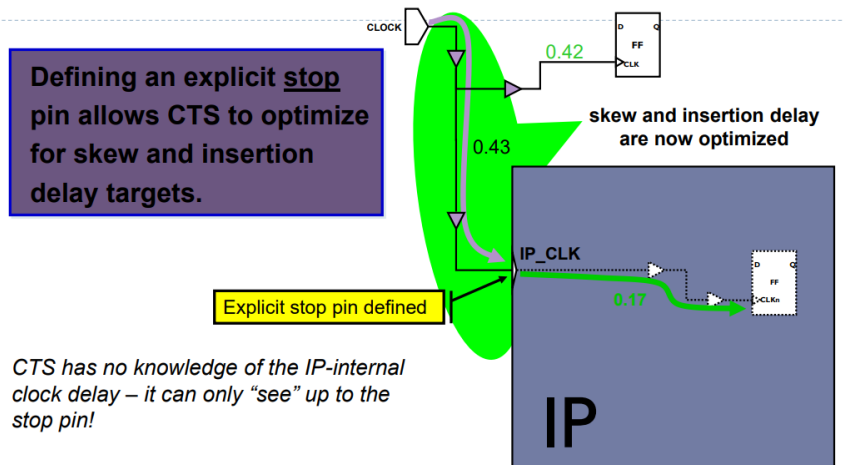
● **Stop pin:** ودى الى بنشرح عليها من بدرى الى هى اول لل clk pin of the flop ودى ال default فى الثلاثة ... وهنا ال tool بتعمل optimization فى ال path من source لل sink pin للحاجتين الى هما DRC & targets ... ال DRC لازم زى ما قلنا وبتعمل لل targets عشان مهم هنا ان ال path ده يكون balanced مع باقى ال paths

● **Float pin:** هنا بردوا ال tool بتعمل optimization فى ال path من source لل float pin للحاجتين DRC & targets ... ال DRC لازم زى ما قلنا وبتعمل لل targets عشان مهم هنا ان ال path ده يكون balanced مع باقى ال paths ... الفكرة هنا ان دى بتكون clock pin of IP يعنى انا موصلتش لحد clock pin of a register بس CTS مش هتتعرف توصل لل delay من بداية ال clk pin of the IP لحد ال register فانا بدهولها فى ال command اما بقولها ان دى float point وهنشوف ال command فى ICC tool section ... بس لو عرفتها ك stop pin فهى هت optimize بس مش هتاخذ فى اعتبارها ال delay الى جوا ال IP بس لو معرفتهاش خالص هيعتبرها exclude pin فزى ما هنشوف فى ال exclude انه مش هيعمل target optimization يعنى مش هيخلي ال path ده balanced مع الباقي وده هيكون غلط

هنا معرفتش ال IP clock pin فشافتها ال tool انها excluded pin عشان كده ما عملتش optimization فى ال targets الى هما ال skew وال insertion delay

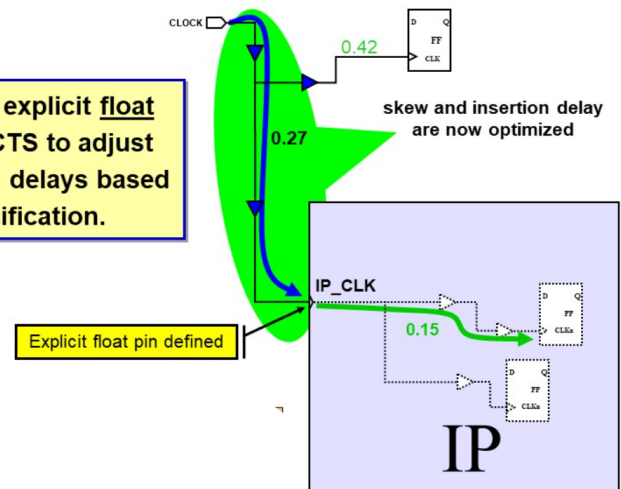


هنا عرفتها انها stop pin فهيعمل optimization لل targets بس مش هتتعرف ال tool حاجة عن ال delay بتاع ال clk الى جوا ال IP هى هتقف لحد ال clk pin وخلص ... فهتلاقى ان ال skew بين ال two paths معمول ومش متاخذ فى الاعتبار ال 0.17 delay الى جوا ال IP

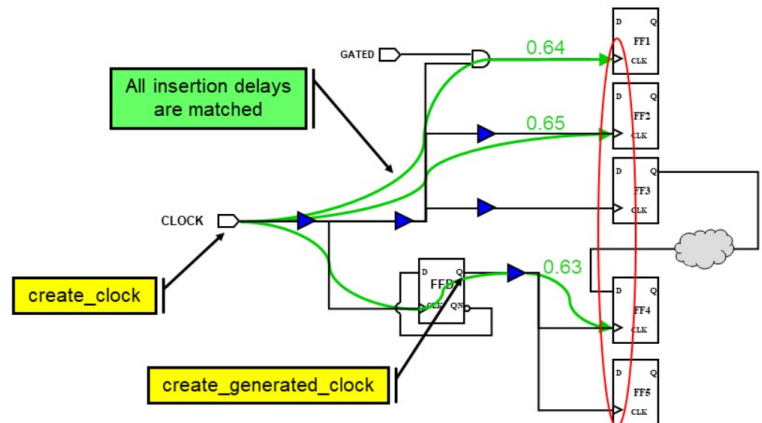
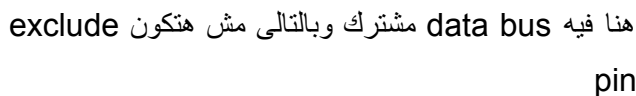


Defining an explicit float pin allows CTS to adjust the insertion delays based on specification.

هنا عرفتھا انها float pin فھيعمل optimization ال  
ل targets وھبطلھ ال delay بتاع ال clk جوا ال  
IP في ال command ... فھتلاقى ان ال skew  
بين ال two paths معمول وھو واخذ في اعتباره ال  
0.15 delay بحيث ان  $0.42 = 0.27 + 0.15$

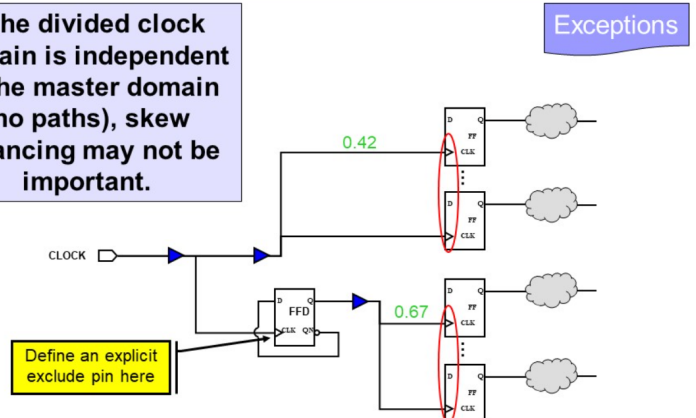


- **Exclude pin:** هنا ال tool بتعمل optimization فى ال path من source ال exclude pin ل DRC فقط مبتعملش لل targets لأن فى ال point دى مش هكون مهتم ان ال path بتاعها يكون balanced مع الباقي وده مثلا ممكن يكون لو ال point دى مش هتدخل ك clk ال flop مثلا هتطلع ك output port او هتدخل على input ال flop ... وفى حالة تانية لكدا وهى انها تبقى هتدخل على clock divider ولكن ال output clock من ال divider هتروح على registers ملهاش علاقة بال registers الى شغالة على ال clock الأساسية يعنى مفيش data path مشترك بينهم ... لكن لو فيه data path مشترك مش هقدر اعمل كده لأن لازم يكون skew ساعتها balanced بينهم



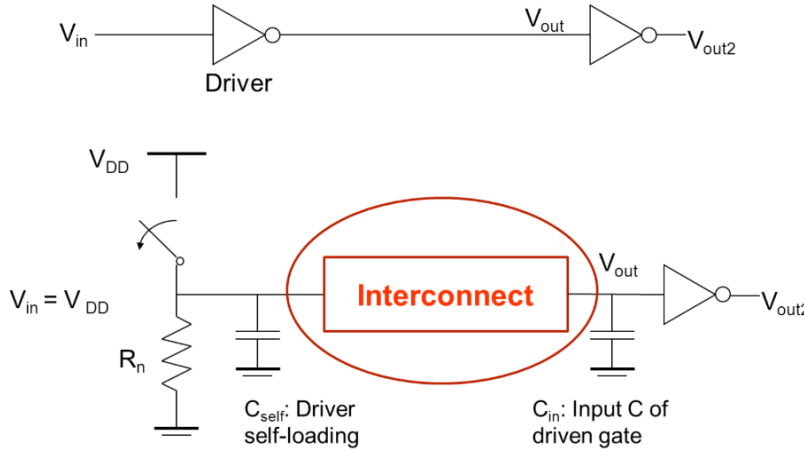
**If the divided clock domain is independent of the master domain (no paths), skew balancing may not be important.**

هنا مفيش data bus مشترك فہعرفھا ك exclude pin

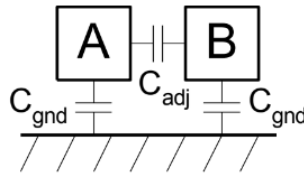


## Crosstalk:

دى بيحصل بين two wires جنب بعض ومعناه ان فيه transition عند wire بيأثر على ال wire التانى يعنى ممكن ال wire التانى ده اصلا عليه stable signal سواء 0 او 1 وتلاقى حصل عليه glitch فجأة او ان هو كمان بيحصل عليه transition فتلاقى ال transition بتاعه بقى اسرع او أبطأ من المتوقع لأنه اتأثر بال transition الى بيحصل على wire تانى جنبه طب ليه بقى wire يأثر على التانى؟؟ خلىنا الأول ندرس حالة ان فيه two wires جنب بعض .. نشوف ايه الى بيربطهم ببعض اصلا عشان نقدر نفهم ازاي واحد ممكن يأثر على التانى:

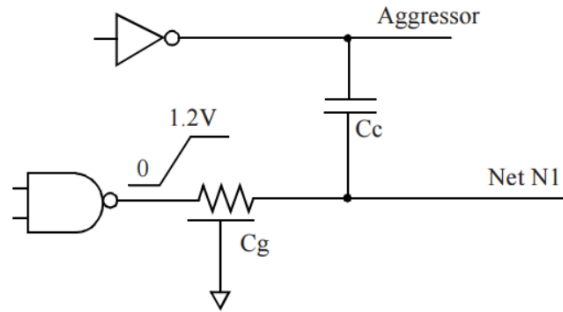


دلوقتى ال wire الواحد اصلا هو اكيد واصل بين two cells على الاقل ... لو دخلنا جوا اى cell هنلاقيه واصل على transistor فى الآخر ... فبداية ال wire هيكون output من ال driver cell الى (هى بت drive ال cell الى بعدها وتديها current) هنلاقيه واصل على output of transistor الى هو هيكون ال drain net بتاع ال transistor فهيكون من الناحية دى شايف drain capacitance ... وفى نهاية ال wire هيكون واصل على input of cell يعنى من جواها هيوصل على input of transistor الى هو gate net يعنى هيشوف gate capacitance ... ففى الآخر الى عايزين نقوله ان ال wire بيكون شايف capacitance من ال cell الى فى بدايته ومن ال cell الى فى نهايته والاتنين بيكونوا capacitance to ground زى ما وضاح فى الصورة فوق ويتضاف ليه ال capacitance بتاعة ال wire نفسه وده كلام المفروض احنا عارفينه من دراستنا لل electronics بس بنفكره دلوقتى ... وكنا درسنا بردوا modules مختلفة نعبر بيها عن total resistance & total capacitance على طول ال wire وكان أشهرهم lumped RC model فلو متعرفش حاجة خالص عن ده ممكن تحاول تسمع عنه معلومات بسيطة بس احنا كل همنا من الذكريات دى اننا نبقى عارفين دلوقتى ان ال wire بيكون ليه capacitance to ground كده ايا كان هتشوفها محطوطة ازاي يعنى كلها متجمعة ولا متوزعة مش مشكلتنا دلوقتى المهم نبقى عارفين المعلومة دى



لما تحط two wires بقى قريبين من بعض بيحصل ايه؟؟ كل واحد منهم له total capacitance to ground وفيه بينهم capacitance بنطلق عليها mutual capacitance وممكن تلاقىها باسم adjacent capacitance بردوا ... دى بتكون موجودة نتيجة ان كل واحد فيهم عليه voltage ففيه electric field بينهم فبقوا عاملين شكل two parallel conductors بينهم مادة عازلة (الفراغات بين ال metal interconnects بتكون مادة عازلة ودى المفروض تكون اتعرضلتها فى مواد زى VLSI design) فاتكون عندى capacitor

خلينا بقى دلوقتى نشرح على مثال ل wire هيحصل عليه transitions ونشوف ال wire الى جمبه هياثر عليه ازاي وامتى ... فاحنا هندرس ال wire الى فى الصورة اسمه Net N1 نشوف ال transitions عليه هتتاثر ولا لا بال wire الى جمبه الى هنسميه aggressor وهنشوف ليه اطلقنا عليه الاسم ده بعد شوية فخلينا ندرس حالات مختلفة ل N1 دلوقتى:



$$V_{cc} = V_{N1} - V_{aggressor}$$

**case1: aggressor net stable:** يعنى مفيش transition عليها ... N1 هى الى عندها transition بس:

Capacitance		Before rising transition at net N1	After rising transition at net N1
Grounded Cap, Cg		$V(Cg) = 0$	$V(Cg) = V_{dd}$
Coupling Cap, Cc	Aggressor net steady LOW	$V(Cc) = 0$	$V(Cc) = V_{dd}$
	Aggressor net steady HIGH	$V(Cc) = V_{dd}$	$V(Cc) = 0$

دلوقتى هيحصل high transition عند N1 فده معناها انها كانت 0 وبالتالي  $Cg = 0$  voltage across وبعد ما ال transition يخلص هيكون خلاص  $Cg$  اتشحن وبقى  $V_{cg} = V_{DD}$  ... اما بالنسبة لل mutual capacitance  $Cc$  فعندى احتمالين:

- ان aggressor كان عليه  $V_{ag}=0$  و  $V_{N1}=0$  وبالتالي  $V_{cc}=0$  فبعد ال transition هتكون  $V_{cc} = V_{dd}-0=V_{dd}$
  - ان aggressor كان عليه  $V_{ag}=V_{dd}$  و  $V_{N1}=0$  بردوا وبالتالي  $V_{cc}=0$  فبعد ال transition هتكون  $V_{cc} = V_{dd}-V_{dd}=0$
- N1 شحن  $Cc$  من 0 الى  $V_{dd}$  او من  $V_{dd}$  الى 0 يعنى فى الحالتين احتاج نفس ال charges بالإضافة لشحن  $Cg$  وهنا ال aggressor متأثرش عليه بحاجة عادى ده المتوقع انه هيتحتاج charges تكفى انه يشحن كل ال capacitors الى عنده الى  $V_{dd}$

=====

**case2: aggressor switching in same direction as N1:** نفترض الاتنين high transition

Capacitance	Before rising transition at net N1 and aggressor net	After rising transition at net N1 and aggressor net
Grounded Cap, Cg	$V(Cg) = 0$	$V(Cg) = V_{dd}$
Coupling Cap, Cc	$V(Cc) = 0$	$V(Cc) = 0$

بالنسبة لل  $Cg$  فهي 0 قبل ال transition وهتوصل ل  $V_{dd}$  بعد ال transition ... بالنسبة ل  $Cc$  فقبل ال transitions الاتنين ب 0 وبالتالي  $V_{cc}=0$  وبعد ال transition بردوا هيفضل 0 لأن هو هيفضل نفس ال voltage على N1 & aggressor دايمافمفيش

voltage drop على  $C_c$  وبالتالي هنا  $N1$  احتاج charges عشان ي شحن  $C_g$  يعني  $Q = C_g * V_{dd}$  ... ولو كان ال aggressor كمان اسرع من  $N1$  يعني transition بتاعه اسرع هيساعد في شحن  $C_g$  فهيجتاج charges اقل من  $C_g * V_{dd}$  ... وفي الحالتين ده اقل من الی احتاجه في case 1 وبالتالي هلاقى  $N1$  transition بقى أسرع من ال normal الی هو في case 1 وده بسبب حركة ال aggressor في نفس اتجاهه

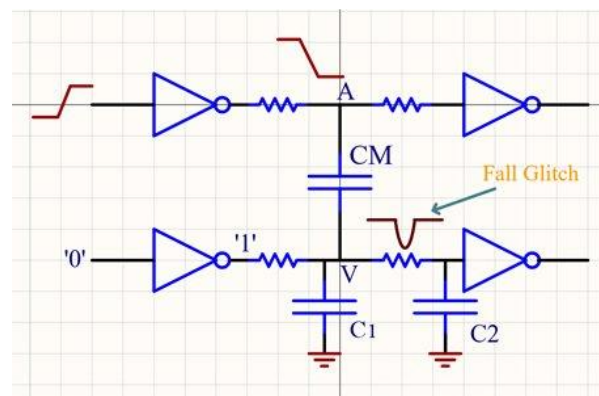
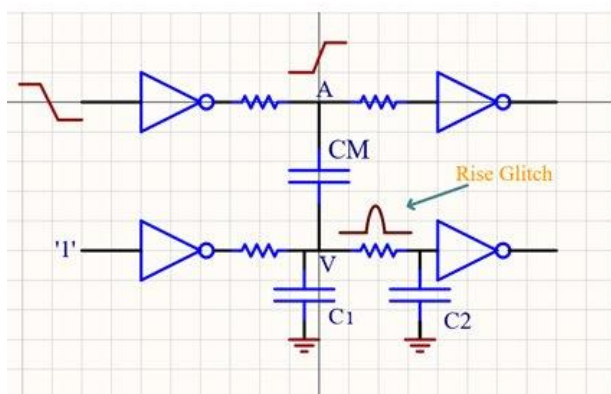
1	2	3	4	5	6
7	8	9	10	11	12

case3: aggressor switching in opposite direction of N1 یعنی ال aggressor عنده low transition

<i>Capacitance</i>	<i>Before transition at net N1 and aggressor net (net N1 is low; aggressor net is high)</i>	<i>After transition (net N1 is high; and aggressor net is low)</i>
Grounded Cap, Cg	$V(C_g) = 0$	$V(C_g) = V_{dd}$
Coupling Cap, Cc	$V(C_c) = -V_{dd}$	$V(C_c) = V_{dd}$

بالنسبة لـ Cg فهي 0 قبل ال transition وهتوصل لـ Vdd بعد ال transition ... بالنسبة لـ Cc فقبل ال transitions هتبقى عندنا VN1=0 و Vag=Vdd وبالتالي Vcc=0-Vdd=-Vdd ثم بعد ال transition هتبقى عندنا VN1=Vdd و Vag=0 وبالتالي Vcc=Vdd-0=Vdd فهنا N1 احتاج charges عشان يشحن Cg كالعادة و Cc احتاج charges عشان يتشحن من Vdd الى Vdd يعني Q=2Cc\*Vdd ... ودى هيووفرها ال aggressor كمان مع N1 بس هتخلي N1 أبطأ من case 1 كإنه بقى شايف capacitance بتساوى 2Cc بدل Cc واحدة وده بسبب حركة ال aggressor في اتجاه عكسه

**case4: N1 is static:** يعنى N1 هتبقى ثابتة وليكن 0 وال aggressor هو الى هيتحرك من 0 الى Vdd فبالنظر ل N1 هو مش بيعمل حاجة ولا هيشارك فى شحن Cc لأنه متغيرش اصلا بس الى هيحصل ان لو aggressor ده عنده transition اسرع من N1 فهلاقي ينتقل charges ل N1 تعمل glitch فيه كده ثم يرجع تاني لقيمتة عادى بس ده ممكن بسبب functional failure فى الدائرة

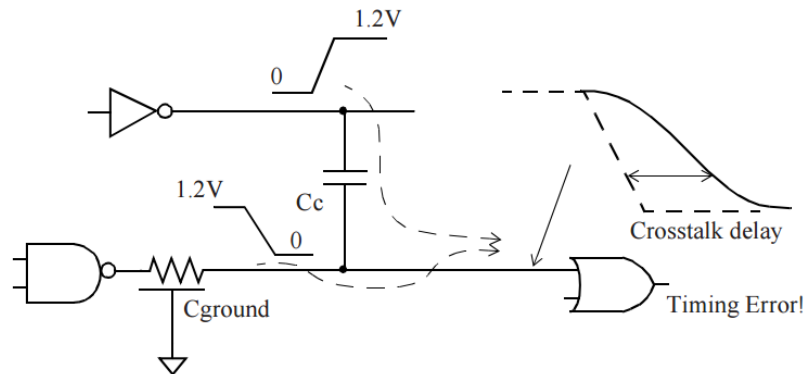


كده احنا شوفنا كل الحاجات الى ممكن تحصل ل N1 فشوفنا فى case1 ال transition فى الحالة الطبيعية ثم case 2 بقى أسرع من الطبيعي ثم case 3 كان أبطأ من الطبيعي ثم فى case 4 كان N1 أصلاً فى حاله مفيهوش transitions وجه ال aggressor عمله glitch



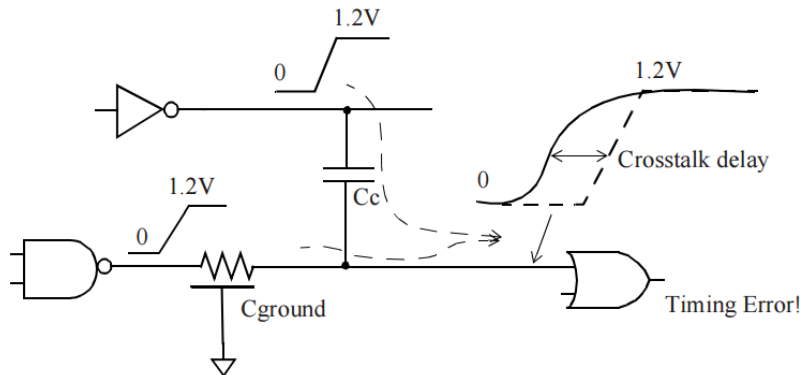
خلينا نكلم بقى بمصطلحات ال crosstalk ... بنسمى N1 ده ال victim على أساس انه بيتأثر بالى بيحصل فى ال aggressor و ال aggressor هو نفس الاسم الى استخدمناه فى شرح ال cases ... ازاي نفرق بين ال aggressor وال victim؟؟ ال transition time بتاعه أسرع من ال victim وبالتالي اما بيحصل transition عند ال aggressor اما انه هيعمل glitch عند ال victim لو ممكنش عنده transition او هيسرع ال victim transition لو كانو الاتنين عندهم transition فى نفس الاتجاه او انه هيبطأ ال transition ال victim لو الاتنين عندهم transitions فى اتجاهين عكس بعض

**Positive crosstalk:** ده لما بيكون ال aggressor وال victim عندهم transitions فى اتجاهين مختلفين لإنه بيبطأ ال victim transition يعنى بيزود ال delay بتاعه عن ال expected ... زى فى الصورة تحت ال dashed هو ال expected transition لكن ال actual خد وقت اكبر يعنى أبطأ



**Positive cross talk**

**Negative crosstalk:** ده لما بيكون ال aggressor وال victim عندهم transitions فى نفس الاتجاه لإنه بيسرع ال victim transition يعنى بيققل ال delay بتاعه عن ال expected ... زى فى الصورة تحت ال dashed هو ال expected transition لكن ال actual خد وقت اقل يعنى أسرع



**Negative cross talk**

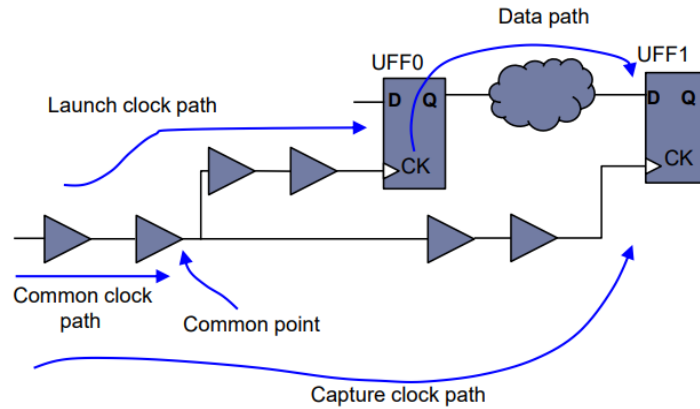
=====

### Crosstalk effects on timing analysis:

ال crosstalk بيغير فى wire delays يعنى هيغير فى ال path delay كله وبالتالي ممكن تحصل timing violations سواء فى setup او ال hold على حسب هل ال crosstalk يسرع ال victim ولا ببطأه ... فأننا من برا مقدرش احدد هل ال wire ال crosstalk هيسرعه ولا هيبطأه ... دى حاجة مش معروفة ممكن يحصل اى حالة من الاتنين عشان كده بنفترض الأسوء فى حساب ال timing



## 1. Setup analysis:



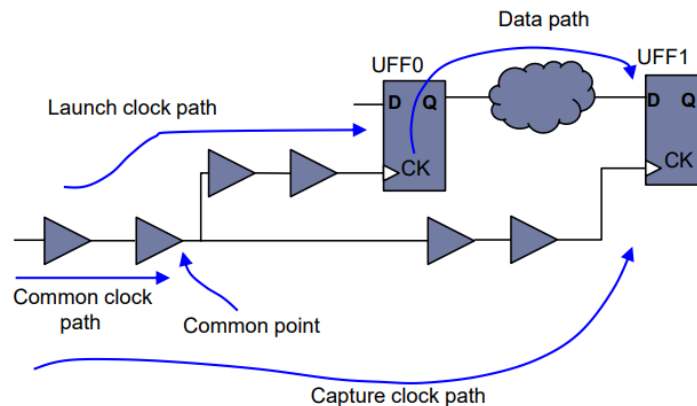
$$T_{c2q} + T_{comb\_max} + T_{setup} \leq T_{clk} + T_{skew\_min}$$

$$T_{skew\_min} = \text{min capture clock path delay} - \text{max launch clock path delay}$$

أسوء حاجة لمعادلة ال setup ان data path يكون عنده max possible delay الى هو بنعبر عنه ب  $T_{comb\_max}$  وان ال skew يكون ال min possible value وبما ان ال skew هو delay of capture clock path - delay of launch clock path يبقى ال min هو ان ال capture path يكون عنده اقل delay ممكن له ولكن ال launch path يكون عنده اكبر delay ممكن له وال clock paths دي من غير ال common path ... يبقى انا عايز max delay لل data path و launch clock path يبقى هنا هيشوف عندهم positive crosstalk ولكن عايز min delay لل capture clock path فهيشوف عنده negative cross talk

=====

## 2. Hold analysis:



$$T_{c2q} + T_{comb\_min} \geq T_{hold} + T_{skew\_max}$$

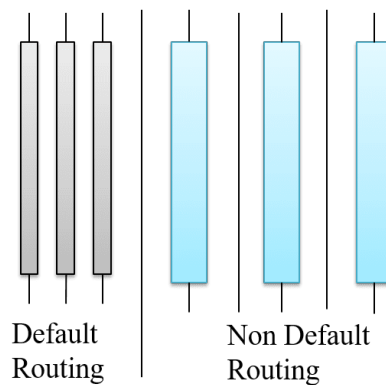
$$T_{skew\_max} = \text{max capture clock path delay} - \text{min launch clock path delay}$$

أسوء حاجة لمعادلة ال hold ان data path يكون عنده min possible delay الى هو بنعبر عنه ب  $T_{comb\_min}$  وان ال skew يكون ال max possible value وبما ان ال skew هو delay of capture clock path - delay of launch clock path يبقى ال max هو ان ال capture path يكون عنده اكبر delay ممكن له ولكن ال launch path يكون عنده اقل delay ممكن له وال clock paths دي من غير ال common path ... يبقى انا عايز min delay لل data path و launch clock path يبقى هنا هيشوف عندهم negative crosstalk ولكن عايز max delay لل capture clock path فهيشوف عنده positive cross talk

## Non-default routing rules

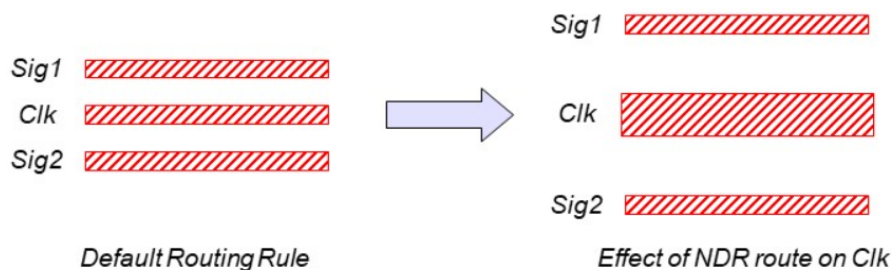
احنا قلنا اننا هنعمل ال clock routing هنا فى خطوة CTS ... فالعادى اننا هنستخدم metal layers من الى كنا مجهزين ال tracks بتاعتها فى ال floorplan وخلص عمل clock routing عليها ... بس فيه concept مهم فى ال routing وهو non default routing rules وده بنستخدمه بالذات فى clock tree ونشوف ليه بنحتاجه بالذات فى clock tree

ال routing rules الى نقصدها فى ال width بتاع ال wire وكمال ال spacing بين كل wire والتانى ... ال default هو ال min يعنى ال tool بتحط min width & min spacing فى العادى طالما مفيش حاجة تستدعى انها تعمل حاجة مختلفة عن كده ... لإن استخدام min هيجلى فيه routing resources اكتر يعنى اماكن اكتر امشى فيها wires لكن كل ما width بتاع ال wire بيزيد فانا كده بقل ال routing resources ... وناخد بالناس ان لو ال width بتاع wire زاد لازم ال spacing بين ال wire ده واى wire حواليه بيزيد بنفس مقدار زيادة ال width

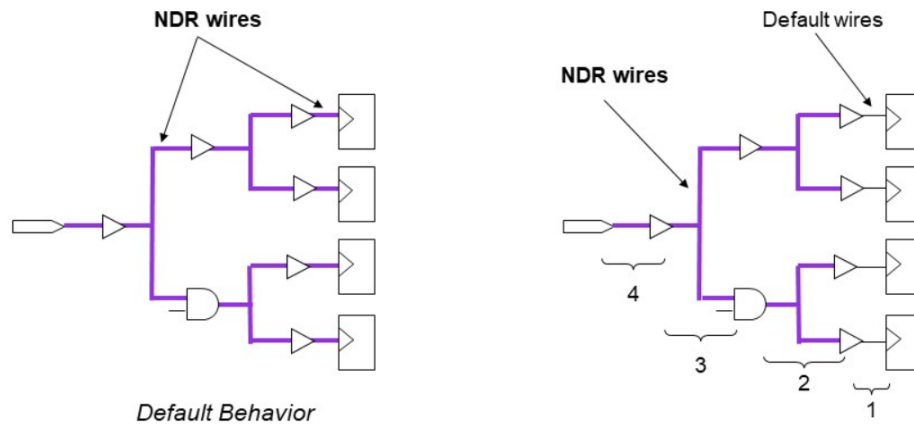


انى اغير فى default routing rules يعنى ازود ال width & spacing عن ال min ده هو non-default routing rules ... ليه بنحتاج نعمل حاجة زى دى؟؟ أنا لما بزود ال wire width فده معناه انى بقل ال resistance بتاعته يعنى بقل ال delay بتاعه وال double spacing هيجليه immune لل crosstalk اكثر لإن كل ما ال wire اتأثر بال cross talk اكثر ال delay هيتأثر أكثر ... وبالتالي NDR هتفيدنى حتى فى design signal routing ... مثلا لو انا عندى design شغال على freq معين ثم قررت انى هشتغل على freq اعلى يعنى period اقل وبالتالي setup slack لو مكنتش كبير كفاية انه يتحمل التغيير ده فهحتاج انى اقل ال delay بتاع ال cells وبتاع ال nets الى هى ال wires فال cells ليها طرقها زى انى استخدم cells ب area اكبر بس احنا حاليا بنتكلم عن ال wires فاقدر اقل ال delay بتاعها باستخدام NDR ... ممكن انا احدد لل tool ال width & spacing فى metal layers معينة او انها من نفسها بتعمل NDR فى critical time paths وهى فى ال optimization عشان تقدر ت meet timing

ال NDR بقى مهمة فى clock tree ليه؟؟ لإن ال clock دى اهم signal عندى هبقى عاوزها متتأثرش بال crosstalk ويكون ال delay wires فيها قليل ... فمثلا يخلى ال wires بتاعة ال clock ليها double width and double spacing (هنا ال double ده relative لل min) ببقى انا كه بخلى ال clock wires او نقول عليها بقى clock routes تبقى اقوى ومتتأثرش بال crosstalk



**Important note:** clock routes بنعملها على layers عالية زي 3,4,5 metal layers مثلا يعني مستخدمتش ال layers الی تحت اوی زي 1 ولكن ال الاجزاء من ال tree الی هتوصل ل clock sink pin دی خلیها default routing عادی والی هی بتكون عادة على 1 metal لأن 1 metal مش المفروض یكون فیها NDR ... ال sink pin نقصد بیها نهاية ال clock یعنی register clock pin



## CTS effects & optimizations

بعد ما هنعط CTS هتأثر على ال design بتاعی ازای؟؟ ال tool ممكن تغير اماكن ال cells الی اتحتت قبل كده عادی وبالتالي ممكن توديه مكان مش كويس لل design زي الی كانو فیه قبل كده ... اضافة clock tree معناها انی حطیت cells الی ال clock buffers وبالتالي congestion هيزيد وممكن یكون مش acceptable ومحتاج یحسن ... وممكن الاقی ظهر violations فی timing او max capacitance او max transition وده متوقع یحصل لإنی بزود cells و routing لل cells دی

طیب ال optimization بقی ممكن یعمل فی ایه ... سواء اثناء ما ال tool بتحتط clock routes & tree او بعد ما تخلص ویقی فیه violations هت optimize فی کذا حاجة مش لازم ال tree نفسها زي ما قلنا انها بتغير اماكن ال cells الاساسیة اصلا فهی ممكن تغير فی اماكن ال cells بتاعة ال design او فی حجمها کمان ونفس الکلام تعمله مع ال clock buffers تغير فی اماكنها او حجمها بحيث تعمل optimization ومیبقاش فیه ای violations وممكن تضیف کمان buffers تانی عادی لو فیه path احتاجت تزود فیه delay اکثر مثلا یبقى نلخص ال optimization الی ال tool تقدر تعمله فی:

**Gate relocation - Gate sizing - Buffer relocation - Buffer sizing - Delay insertion**

## ICC tool

### 1. Checks before CTS:

#### Command:

`check_physical_design -stage pre_clock_opt`

ال command ده بنستخدمه من اول ال placement زي ما شوفنا قبل كده بحيث انه يتأكد من ال design لحد الخطوة دی تمام من خلال checks بیعملها حسب كل stage واقف فیها ... هنا بیعمل checks قبل ما نبدأ CTS وهنا بی check على ان ال placement اتعمل وان ال clocks متعرفة وان ال clock sources جاية من top level port مش hierarchical pin

● **stage:** هنا قبل CTS فکتبنا `pre_clock_opt` وعندنا two possible values غیرها وهما `pre_place_opt` وودی الی شوفناها فی checks قبل ال placement وفیه `pre_route_opt` هنعملها بردوا قبل ما نبدأ ال routing

بعد ما يخلص بيطلع على ال error & warnings ويرد ب 1 على ال command لو مفيش errors زى فى الصورة دى:

```
Total messages: 0 errors, 1 warnings
-----
Other Warning Summary for check_physical_design
-----
ID              Occurrences  Title
-----
MW-349          1           Cell contains tie connections which are not con..
```

=====

### Command:

`check_clock_tree -clocks [clock_list]`

هنا بيعمل check على الحاجات الى ممكن تسبب مشكلة واحنا بنحط ال tree زى مثلا ان يكون فيه generated clock مش متعرف ليها master clock او ان register متوصل عليه اكتر من clock ... لو فيه اى violations هيطلعها ولو مفيش هيكفى بانه يرد على ال command ب 1 فقط بمعنى انه كله تمام

● **clock:** لو انا حظيت ال option ده فهو هيعمل checks على ال clocks الى انا هحطها ليه بس لكن لو محطتهوش خالص وكتبت check\_clock\_tree بس كده فهو هيت check على كل ال clocks الموجودة من نفسه

لو كتبت man check\_clock\_tree فى ال terminal هيطهرلك ال description وفيه كل ال checks الى بتعملها وهتلاقى كل check له code بيبر عنه زى الصورة تحت كده وطبعا اى exceptions او details هتلاقيهم مكتوبين بعد ال description:

```
The check clock tree command is suggested to use before CTS for a
linked design to check for common problems that might adversely impact
clock-tree synthesis. The following classes of checks are currently
performed:
CTS-821 a master clock does not propagate to a generated clock
CTS-822 improperly specified master clock
CTS-823 a master clock terminates at a multi-clock pin
CTS-831 a clock has no synchronous pins
CTS-832 a clock loops to itself
CTS-834 multiple clocks per register
CTS-835 exceptions defined on output pins
CTS-837 dont touch nets/cells not defined by set_clock_tree_exceptions
CTS-838 clock nets are routed prior to CTS
CTS-841 ignored exceptions
CTS-850 no buffer available for CTS
CTS-851 multiple arc between input and out pins
CTS-852 case analysis set on a pin
CTS-853 disable timing set at a pin
CTS-855 no reference can be used
CTS-858 clock tree exception defined on a clock source
CTS-859 max fanout is maller than its default value
CTS-860 nonMCM conflict balancing requirement check
CTS-861 MCM check a master clock terminates at a multi_clock pin
CTS-862 MCM conflict balancing requirement check
CTS-866 different offset values defined for ICDB
CTS-867 different target delay value set for ICDB
CTS-868 conflicting requirement set for clocks for ICDB
CTS-869 cyclic offset values specified for ICDB
CTS-871 phse delay skew of etm/ilm entrance pin larger than threshold
CTS-872 conflict sink group
CTS-875 possible cap/fanout violation on dont touch net
CTS-876 generated clock defined in the output cone of a data pin
CTS-877 cell has output output arcs
CTS-982 a clock root is in fanin cone of another root
```

=====

### Command:

`check_legality -verbose`

ده اتكلمنا عنه بالتفصيل فى placement بس ك check فى نهاية ال placement فهو نفس الحاجة ... احنا بس هنا بنأكد على انه بيتعمل قبل ما ندخل فى CTS يعنى بعد ال placement زى ما شوفنا

## 2. Reports and analysis:

### Command:

`report_clock_tree [-interclock_timing] [-summary] [-settings] [-exceptions]`

هنا هيطلع على report لل clock tree بحيث اعرف فيها cells ايه وال worst slack مثلا وكل التفاصيل لو انا كتبت `report_clock_tree` بس هيديني report زى اللى تحت ده ... بس فى المثال ده مكناش عملنا CTS لسه عشان كده الحاجات اللى هتضاف فى CTS كلها 0 وهلاقي ال longest & shortest path delays محسوبين ب expected values لتوصيلة عادية ب wires لأنه لسه مفيش CTS:

```
===== Global Skew Report =====
Clock Tree Name       : "clk"
Clock Period          : 19.00000
Clock Tree root pin   : "clk"
Number of Levels      : 1
Number of Sinks       : 4240
Number of CT Buffers   : 0
Number of CTS added gates : 0
Number of Preexisting Gates : 0
Number of Preexisting Buf/Inv : 0
Total Number of Clock Cells : 0
Total Area of CT Buffers : 0.00000
Total Area of CT cells : 0.00000
Max Global Skew       : 0.00041
Number of MaxTran Violators : 0
Number of MaxCap Violators : 1
Number of MaxFanout Violators : 1

Operating Condition    worst
Clock global skew      0.000
Longest path delay     0.032
Shortest path delay    0.031

The longest path delay end pin: pc_current_reg_10 /CK
The shortest path delay end pin: datamem/ram_reg[0][6]/CK
```

• `interclock_timing`: هيطلع timing information زى WNS worst negative slack و cumulative negative slack CNS و عدد ال violating paths ... ببينوا فى جدول زى كده:

```
In the following example, the report_clock_tree command generates a
interclock_timing report.

prompt> report_clock_tree -interclock_timing

*****
Report : clock tree
Design : Top Create
Scenario(s): normal_mode
Version: G-2012.06-ALPHA3
Date   : Fri Mar 9 12:18:01 2012
*****

===== Report for scenario (normal_mode)=====

From Clock  To Clock  WNS   CNS   NVP   Local skew(WNS path)
=====
CLK1         CLK2       -1.0   100.1  245    0.1
CLK1         CLK1       -0.5   20.3   100    0.3
```

• `summary`: هيطلع جدول بكل التفاصيل الى فى tree زى ما باين فى الصورة تحت بس دى كانت قبل CTS .. ولو عندى اكثر من tree هيظهر كل واحدة فى row خاص بيها:

```
===== Clock Tree Summary =====
Clock      Sinks   CTBuffers ClkCells  Skew    LongestPath TotalDRC   BufferArea
-----
clk        4240     0         0         0.0004  0.2587    4243      0.0000
l
```

● **settings:** هنا هيطلع reports بكل الحاجات الى حطيتها باستخدام ال `set_clock_tree_options`

● **exceptions:** هيطلع list بكل ال clock tree exceptions الى عملتها

ال command له options تانية كتير بتساعدنى اشوف الى انا عايزه يعنى مثلا قبل ما اعمل CTS ممكن settings دى تخلىنى اشوف هل كل الى حطيتها فى `set_clock_tree_options` ال tool شافته وسجلته عندها ولا لا وهكذا فى ال exceptions وغيرهم كتير ...  
فال command ده مفيد سواء قبل CTS او بعده ... اقدر اعرف منه كل تفاصيل ال clock

=====

### **Command:**

`report_clock_timing -type report_type -clock clock_list [-setup or -hold]`

اعرف منه كل معلومات ال timing paths فى clock tree حسب ال options الى تحتها وفيه options كتير جدا احنا ذكرنا جزء بسيط منها وتقدر من ال manual تشوف الباقي

● **report\_type:** فيه انواع كتير زى:

● **transition:** هيطللك ال transition times فى ال clock tree

● **latency:** هيطللك ال latency فى ال clock tree

● **skew:** هيطلع ال skew بس الى بين two end points يعنى بين two flops بس فى نفس ال clock domain يعنى نفس ال clock source

● **interclock\_skew:** هيطلع ال skew بس الى بين two end points يعنى بين two flops من اثنين clock domains مختلفين

● **summary:** هيطللى معلومات كل الى فات ده فى report واحد من ده

● **clock:** بحت هنا ال clocks الى عايز اطلع لل network بتاعتها ال report ... لو محطتش ال option ده خالص فهو هيطلع لكل ال clock sources الى عندى فى ال design اتعلمهم clock tree

● **setup or hold:** بختار ال data paths الى هستخدمها يكون بتاع ال setup ولا ال hold بمعنى اخر ان skew و latency و transitions زى مالنا عارفين بتفرق انا باخد فيهم ال possible max or min حسب انا هشتغل على setup timing ولا hold timing فهنا بختار ده او ده ... ولو مكتبتش ولا option منهم فال default هو setup

=====

### **3. Clock tree exceptions:**

#### **Command:**

- `set_clock_tree_exceptions -stop_pins [get_pins instance_name/pin_name]`
- `set_clock_tree_exceptions -exclude_pins [get_pins instance_name/pin_name]`
- `set_clock_tree_exceptions -float_pins [get_pins instance_name/pin_name] \`  
`-float_pin_max_delay_rise value -float_pin_min_delay_rise value \`  
`-float_pin_max_delay_fall value -float_pin_min_delay_fall value`

دى ال exceptions الى اتكلما عنها وشرحناها بس فيه exceptions تانية هتلاقىها متوفرة فى options تانية لل command بتاع set\_clock\_tree\_exceptions ... فدى طريقة كتابة ال commands لل exceptions الى شرحناها وبما اننا بنتكلم على pins فهتلاقى بعد get\_pins بتحط اسم ال instance الاول ثم اسم ال pin يعنى مثلا FF1/CLK او IP1/CLK ... بالنسبة لل float pins زى ما قلنا بديها ال delay هنا فى option زيادة عن ال stop & exclude فممكّن تحطه فى صورة:

- **float\_pin\_max\_delay\_rise**: بحسب ال delay من اول ال rising edge بتاعة ال clk ما جت عند ال float pin لحد ما وصلت لل clk pin الى عايزنها جوا ال IP بس هنا بتحط ال max possible delay او بمعنى اخر ال longest path
- **float\_pin\_min\_delay\_rise**: بحسب ال delay من اول ال rising edge بتاعة ال clk ما جت عند ال float pin لحد ما وصلت لل clk pin الى عايزنها جوا ال IP بس هنا بتحط ال min possible delay او بمعنى اخر ال shortest path
- **float\_pin\_max\_delay\_fall**: بحسب ال delay من اول ال falling edge بتاعة ال clk ما جت عند ال float pin لحد ما وصلت لل clk pin الى عايزنها جوا ال IP بس هنا بتحط ال max possible delay او بمعنى اخر ال longest path
- **float\_pin\_min\_delay\_fall**: بحسب ال delay من اول ال falling edge بتاعة ال clk ما جت عند ال float pin لحد ما وصلت لل clk pin الى عايزنها جوا ال IP بس هنا بتحط ال min possible delay او بمعنى اخر ال shortest path

=====

#### 4. Clock tree synthesis, optimization and routing:

##### Command:

set\_driving\_cell OR set\_input\_transition

تفاصيل ال commands موجودة فى synthesis file فى constraints on the input port من اول 7 page ... هنا احنا بنستخدمه عشان نحط constraints على ال clock input port فى ال design لو انا مكنتش حاطط ليها قبل كده او هغيره دلوقتى

=====

##### Command:

```
set_clock_tree_options -clock_trees clk_sources -target_early_delay min_insertion_delay\  
-target_skew max_skew_value -max_capacitance value_in_pf -max_fanout integer_value\  
-max_transition value_in_ns -buffer_relocation true/false -buffer_sizing true/false\  
-gate_relocation true/false -gate_sizing true/false -insert_boundary_cell true/false\  
-ocv_clustering true/false -layers_list routing_layers_name\  
-routing_rule non_default_rule_name -use_default_routing_for_sinks number
```

هنا هجهز ال clock tree يعنى options دى كلها فيها بحدّد ال targets & constraints الى اتكلما عنهم وكمان بقول لل tool ايه الطرق الى تستخدمها فى ال optimization وايه لا ... وكمان فيه options بتكون لل routing بتاع ال clock



- **clock\_trees**: انا هنا بقول ال source بتاع ال clock tree يعنى ال clock input port مثلا بحط اسمها ... ويمكن يبقى عندى اكثر من clock وبالتالي كل واحدة ليها tree فهكتبهم عشان يعمل لكل source منهم tree لو انا محدثش ليه sources فهو هيعمل كل ال options الى هحطها فى ال command ده على كل ال clocks المتعرفة فى ال design
- **target\_early\_delay**: هنا بحدد واحد من ال clock tree targets الى اتكلمنا عنه وهو ال insertion delay بحط ال min بتاعه هنا ب ns وال default بتاعه 0
- **target\_skew**: ده clock tree target بردوا زى ما احنا عارفين بحط ال max الى عايزه وال tool بتحاول ت meet ده الاول ثم بعد كده تركز على ال insertion delay target وتحاول ت meet it ... وبردوا ال default بتاعه 0
- **max\_transition & max\_fanout & max\_capacitance**: دول ال clock tree constraints بحط ال value الى عايزها لكل واحد ... ولو محطتش فكل واحد له default value بس مش 0 وهتلاقيهم فى description فى manual
- **gate\_sizing & gate\_relocation & buffer\_sizing & buffer\_relocation**: دول كلهم طرق optimization اتكلمنا عنها فى الشرح فوق ... فالى عايز ال tool تستخدمها فى ال optimization احطها true والى مش عايزها تستخدمها احطها false
- **insert\_boundary\_cell**: هنا بقول لل tool انها تحط boundary cells عند ال clock input ports لو قولتلها true او محطتش لو قائلها false ولو مكتبتش ال option خالص فهو ال default بتاعه false ... وده **global option** يعنى هيتحط على كل ال clocks الموجودة سواء انا كتبتها فى option clock\_trees او لا
- **ocv\_clustering**: بسمح لل tool انها تعمل OCV clustering بانى اديها true او مسمحش بانى اديها false ... ايه بقى ده اصلا؟؟ OCV اختصار on chip variations ودى الى يتكون بسبب PVT:Process, Voltage, Temperature ... ويمكن انها تأثر على ال timing وبالتالي ال functionality فى real life ففكرة ال clustering انه بيعمل register clustering يعنى يقسم ال registers الى موجودة ل groups والحاجات ال related لبعض يقربها من بعض ويمكن two registers يختصرهم فى واحد بس مثلا ... المهم انه يفضل محافظ على ال functionality وال clustering ده بيساعد انه يعمل optimizations كويسة فى ال placement وفى ال routing كمان
- **layers\_list**: هنا بحط أسماء ال metal layers الى تستخدمها ال tool فى clock tree routing ... ال default انها تقدر تستخدم كل ال layers الموجودة
- **routing\_rule**: هنا بحط اسم ال non\_default routing rule الى عملتها ... فلازم اكون عملتها الاول عشان اعرف احط ال option ده
- **use\_default\_routing\_for\_sinks**: ده الى بخليه يستخدم default routing rules فى ال sink pins الى هما ال clock pins بتوع ال registers يعنى ال stop pins مش ال float ولا ال exclude ... الرقم المكتوب بيمثل ال clock tree level من تحت ناقص واحد فبتلاقيه 1 على اساس يبقى 0 level من تحت يعنى من الاخر الى هيوصل على sink pins ... طبعا عشان تستخدم ال option ده فلازم تكون مستخدم قبله routing\_rule option والا ملوش لازمة لانك كده مش عامل non default rules اصلا فخلاص كله هيبقى default

=====

### **Command:**

```
set_clock_tree_references -clock_trees clk_sources\  
-references [get_lib_cells /*ref_cell_names*] \ -sizing_only OR -delay_insertion_only OR  
-boundary_cell_only
```

هنا يحدد لل tool ايه ال buffers & inverters الى تستخدمهم من ال library فى clock tree لإننى لومحدثش فيسمحتها تستخدم اى حاجة منهم موجودة فى target library ... وكمان اقدر احدد ال gates الى تستخدمها فى ال optimization ل sizing او insertion delay او boundary cell

● **clock\_trees**: نفس الكلام فى ال command الى فات ... بقول ال source بتاع ال clock tree يعنى ال clock input port مثلا بحط اسمها ... لو انا محدثتش ليه sources فهو هيعمل كل ال options الى هحطها فى ال command ده على كل ال clocks المتعرفة فى ال design

● **sizing\_only & delay\_insertion\_only & boundary\_cell\_only**: اى option منهم هيتحط فده معناه ان ال cells الى كتبت أسمائها فى ال references هتستخدمها للغرض ده سواء كان فى cell sizing لو استخدمت sizing\_only او لما يحسن delay insertion لو استخدمت delay\_insertion\_only او لما يحط boundary cell لو كتبت boundary\_cell\_only فعشان يستخدم ال cells الى حطيتها فى عمل ال tree نفسها اصلا ... محطش حاجة فيهم وابقى اعلمهم فى commands منفصلة بعده زى فى EX تحت

**Ex:** `set_clock_tree_references -references [get_lib_cells */CLKBUF*]`

```
set_clock_tree_references -sizing_only -references [get_lib_cells cells_used]
```

```
set_clock_tree_references -delay_insertion_only -references [get_lib_cells cells_used]
```

فى المثال حط ال cells الى هتستخدمهم فى ال tree الاول ثم بتوع ال sizing فى command لوحدهم وكذلك بتوع ال insertion delay وهنا المقصود ب cells\_used انه هتكتب اسماء ال cells الى يعتبره احسن حاجة فى target library تستخدمهم للغرض ده سواء sizing او insertion delay

=====

### **Command:**

```
define_routing_rule rule_name -widths {metal_layer_name width_value .....} \  
-spacings {metal_layer_name spacing_value}
```

بعد ما حطينا ال clock tree وعملنا ال optimizations عايزين نعمل ال routing فاول خطوة انى لو عايز اعمل non-default routing الى اتكلمنا عنه محتاج احددله هيكون فى انى layers واياه ال non-default values لل width & spacing الى هتستخدمهم فى البداية هحط اسم لل rule الى هعملها دلوقتى ثم فى width & spacing بعمل list فيها كل metal layer وجمبها ال value الى هتستخدمها زى فى EX كده

**Ex:** `define_routing_rule route_rule_1 -widths {metal3 0.14 metal4 0.28 metal5 0.28} \`

```
-spacings {metal3 0.14 metal4 0.28 metal5 0.28}
```

**Note:** للتنظيم وعشان يكون ال script بتاعك readable اكثر ممكن اترك في الاول تحط ال constraints & targets في set\_clock\_trees\_options وحط ال reference cells باستخدام set\_clock\_tree\_references ... ثم ابدأ مرحلة تطبيق ال routing فاعمل non default routing rules الاول لو عندك باستخدام define\_routing\_rule بعد كده ارجع استخدم set\_clock\_trees\_options تاني بس ب options الخاصة بال routing بقي المرة دي

=====

### **Command:**

clock\_opt -only\_cts -only\_psyn -no\_clock\_route -continue\_on\_missing\_scandefl  
-fix\_hold\_all\_clocks -congestion

استخدام clock\_opt من غير اى options هيحط ال CTS ويعمل optimization ويعمل routing بس اوقات هتلاقى مكتوب ب options بحيث ان كل خطوة من التالفة تتعمل في command لوحدها .. بعض ال options هي:

● **only\_cts:** بحيث يحط ال cells ويبنى ال tree

● **only\_psyn:** هنا هيعمل optimization فقط

● **no\_clock\_route:** يعنى ميعملش ال clock routing

● **continue\_on\_missing\_scandef:** استخدمناه في placement قبل كده في place\_opt command وشرحناه بالتفصيل وقلنا ده عشان لو معنديش scan chain file يعرف يكمل وميطلعش error message فاحنا هنا بردوا بنعمل placement بس لل clock tree cells فمحتاجينه بردوا

● **fix\_hold\_all\_clocks:** عشان يصلح كل ال hold violations وهو بيعمل optimization لإن ال default انه مش هيصلحه واحنا هننت check بعدين لو فيه ونرجع نحاول نصلحه

● **congestion:** هنا بقول لل tool انها تعمل placement لل clock tree cells بهدف تقليل ال congestion على قد ما تقدر يعنى كده بعمل congestion driven placement الى اتكلمنا عنه في ال placement قبل كده

**Ex:** clock\_opt -only\_cts -no\_clock\_route (this command makes clock tree synthesis (CTS))

clock\_opt -only\_psyn -no\_clock\_route (this command only makes clock tree optimization (CTO))

=====

### **Command:**

route\_group -all\_clock\_nets

لو انا مستخدمتش clock\_opt في انى اعمل clock routing فممكن استخدم ال command ده ... وده اقدر اعمل بيه routing لآى signal على حسب ال options بتاعته ... فاحنا هنتكلم عنه بالتفصيل في routing file بس حالياً يكفى نعرف منه ال option المسؤول انه يخلى ال command يشتغل على routing ال clock tree فقط مش ال signals وهو all\_clock\_nets ... فكتابة ال command بالشكل الى فوق ده هيخليه يعمل clock tree routing لل clock tree الى حطيناها خلاص ب clock\_opt

=====

## 5. Hold time fixing:

### Command:

`set_prefer [-min] {cell_list}`

ده بستخدمه بشكل عام من غير min عشان اقول لل tool تستخدم ال cells الى انا حطاهما في ال cell list وقت ال optimization لكن لما اضيف min يبقى بقولها تستخدمها فقط في hold violations fixing وبالتالي ساعتها ال list هيكون فيها buffers & inverters ... وطبعاً لازم اي cell في ال list تكون في target librar

=====

### Command:

`set_fix_hold [all_clock or put the clock name]`

ده بستخدمه عشان اخلى ال tool تصلح كل ال hold violations الموجودة واما احطها اسماء ال clock sources الى عايز اصلح ال hold violations في ال tree بتاعتهم او اني اكتب all\_clocks وهي هتصلح لكل ال trees الى عندي

=====

### Command:

`set_fix_hold_options -default -prioritize_tns -prioritize_min_delay -preferred_buffer\`  
`-effort medium or high`

ده بستخدمه عشان اخلى ال tool تصلح كل ال hold violations الموجودة بس بحطله اولويات يصلح عليها من خلال ال options دي

- **default:** هنا بتهم انها متأثرش على ال worst negative slack و max transition time دول بيكونوا اهم حاجة هنا بس كده ممكن تلاقي degradation في total negative slack لإنه مهتم بال worst فقط فممكن يحصل degradation في اي negative slack تاني
- **prioritize\_tns:** هنا بتهم انها متأثرش على ال worst negative slack و total negative slack و max transition time دول بيكونوا اهم حاجة هنا
- **prioritize\_min\_delay:** هنا ال tool هتحسن في min delay اكثر من ال max بمعنى اوضح انها هتتم تصلح ال hold violations حتى لو ده كان على حساب ال setup فممكن بعد ما تصلحه الاقي timing violations بردوا من ال setup
- **preferred\_buffer:** بخلى ال tools تستخدم ال buffers الى حطيتها في min option في set\_prefered command غير كده هتستخدم هي اي buffers أثناء تصليح ال hold
- **effort:** بحدد ال effort الى هتبزله ال tool في ال fixing وال default بتاعه medium

=====

## 6. power/ground connection

### Command:

`derive_pg_connection -power_net VDD -power_pin VDD -ground_net VSS -ground_pin VSS`

هنعمل ال connection لل power & ground nets وده شرحناها بالتفصيل فى floorplanning وقلنا هنكررها كل ما نط cells جديدة وده الى حصل هنا اننا عاملناها بعد ال CTS

=====

## 7. Save MW cell:

### Command:

`save_mw_cel -as cell_name`

زى ما احنا عارفين اما اتكلمنا عن MW library وشرحناها بالتفصيل فى floorplanning ... يبقى لازم بردوا بعد ال CTS اعمل save ل MW cell جديدة الى هى ال design بتاعى لحد الخطوة دى واديهها اى اسم وليكن design\_CTS

=====

## Important notes:

فى كل ال stages الى فاتو كنا بنقول انه لازم ميقاش فيه setup violations بس عادى لو فيه hold violations سيبهم هنصلحهم بعدين فى CTS ليه بقى كنا بنقول كده؟؟ لاني فى CTS لسه هضيف clock tree بهدف انى اقلل ال skew اصلا واخلى ال clock paths يكونوا balanced بقدر الامكان فده هيجسن ال hold اصلا ... ولحد هنا احنا حطينا كل حاجة خلاص وبالتالى قبل ما اعمل routing الخطوة الجاية لازم اتأكد من ان مفيش اى timing violations سواء setup او hold

=====

## Contact info:

Linkedin profile: [linkedin.com/in/fatma-ali-57b1a6200](https://www.linkedin.com/in/fatma-ali-57b1a6200)

E-mail: [fatma.ali.2028@gmail.com](mailto:fatma.ali.2028@gmail.com)

All ASIC files are provided on VLSI - ASU Community:

[https://drive.google.com/drive/folders/1aLwCLZj0YG9KJhln1D60\\_nWM7p-L3q9a?hl=ar](https://drive.google.com/drive/folders/1aLwCLZj0YG9KJhln1D60_nWM7p-L3q9a?hl=ar)

=====