Routing (V1)

Written by: **Fatma Ali**

Content:

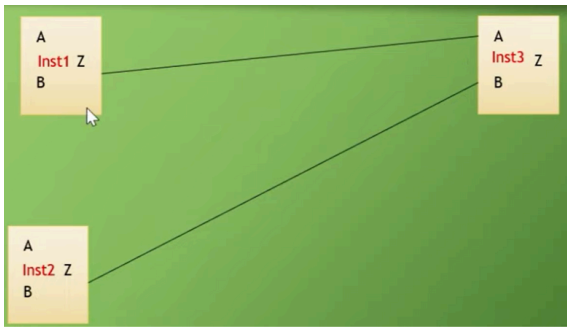
==========

Introduction:

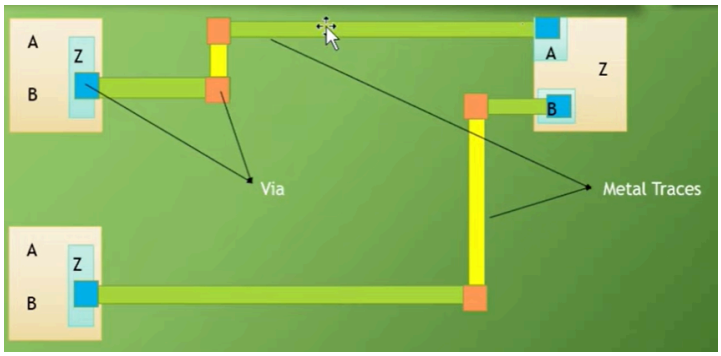
لحد هنا نكون حطينا ال cells وعملنا CTS بال routing بتاعها زى ما شوفنا فدلوقتى ناقص اننا نوصل ال design cells ببعض ودى هى خطوة ال routing الى هنشوفها فى ال file ده بالتفصيل ... يبقى قبل ما ادخل ال routing المفروض يكون:

- Placement خلصان يعنى ال logic cells محطوطة فى اماكنها
- CTS خلصان سواء حط clock tree cells و كمان ال routing بتاعهم زى ما شوفنا فى CTS file
- Estimated congestion يكون مقبول يعنى متكونش الدنيا زحمة اوى زى ما اتكلمنا عنه فى CTS بالتفصيل
- Estimated timing: يكون من غير violations طبعا
- Max capacitance and max transitions: يكونوا من غير violations بردوا ... دول بيكونوا ضمن ال constraints ومهمين جدا فلازم ميكونش فيه violations عليهم قبل ما ادخل ال routing

طول الوقت الى فات كنا بنتعامل مع ال connections بين ال cells انها logical connections (يعنى physically لسه مش موجودة) ودى بنعمل بيها estimations بس عشان نعرف نشوف ال timing تمام ولا لا او اى حاجة تانية محتاجة ال wires الى احتاج فيهم اطلع ال R&C بتوع ال wire ... فى ال routing بقى الهدف اننا نحول ال logical connections دى الى physical connections حقيقة يعنى نوصل بين ال cells فعلا باستخدام ال metal layers والفرق ده واضح من الصورة انه ال logical ده مجرد توصيل بين ال pins بتعمله ال tool عشان تعرف تشتغل على ال design ولكن الحقيقة بيكون توصيل horizontal & vertical باستخدام metal layers & Vias



Logical connections

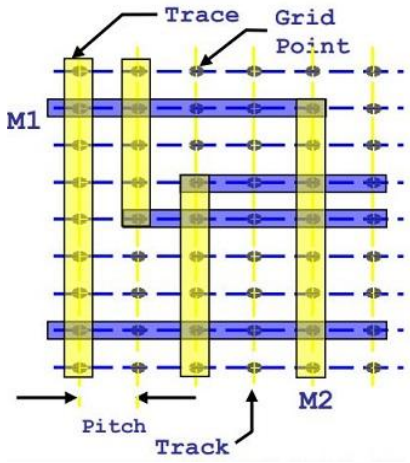


Physical connections

فبما اننا بنعمل routing زى ما عملنا فى clock network بس هنا بنعمل لباقى ال design كله ... بالتالى فى concepts مهمة جدا اتكلمنا عنها فى CTS هنحتاجها هنا زى ال crosstalk, non-default routing, congestion ... خلينا بقى نبدأ ونشوف ايه هى ال routing steps ثم فى الاخر نشوف ال tool commands

==========

Grid routing system:



Metal layers grids

نفتكر فى floorplanning كنا بنحدد حاجة اسمها wire tracks واتكلمنا عنها بالتفصيل وقتها ... فكل metal layer ليها routing tracks محطوطة جمب بعض وزى ما عارفين ان ال tracks direction بيختلف من layer للى بعدها بحيث يكونوا perpendicular يعنى واحدة vertical يبقى الى بعدها horizontal وهكذا فكإن بقى عندى Grid فى كل metal layer ... وكل track بيمثل مكان متاح ل wire (او نطلق عليه بقى route) انه يمشى فيه ... ولكن ال track ده عبارة عن خط واكيد ال route له width فهو بيكون centered

عند ال track يعنى ال track فى نص ال route ثم حواليه ال route وبالتالى المكان الى خده ال route كله بنطلق عليه trace زى ما باين فى الصورة فوق عندى M1 ال tracks فيها horizontal يبقى M2 ال tracks فيها vertical زى ما باين كده ... وعادة دى ال preferred directions ... فكده احنا كنا مجهزين ال grid فى كل layer من قبل كده وهنبدأ نستخدم ال traces بالفعل عشان نوصل بين ال cells ونعمل ال routing فعشان كده بنطلق عليه grid based routing system

Note: لو مش فاكر ال wire tracks كويس ممكن ترجع لشرحهم فى floorplanning file

==========

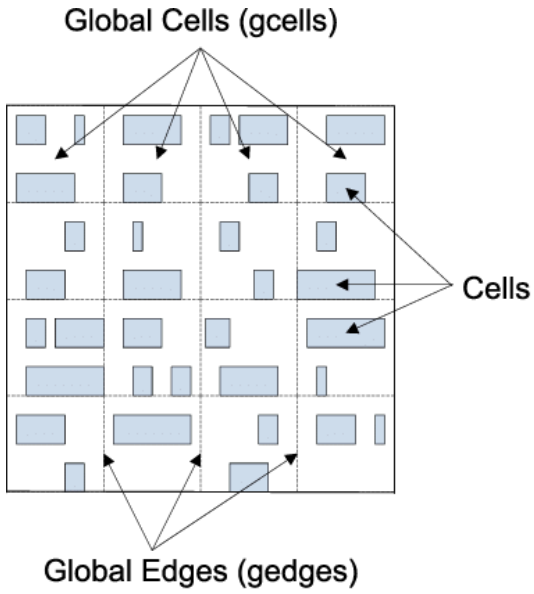
Routing Stages:

هنلاحظ ان اى process اتكلمنا عنها قبل كده كنا بنعملها على خطوات زى مثلا ال placement كان عندى فال coarse & legalized بيعمل ال IC compiler stage الواحدة على كذا خطوة وهنا نفس الكلام ... هيقسم ال routing على خطوات وهنمثلهم فى اربع خطوات:

- Global routing
- Track assignment
- Detailed routing
- Search and repair

1. Global Routing:

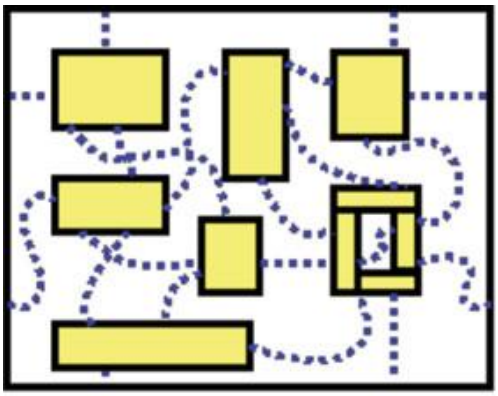
خلينا نشوف اول حاجة وهى تقسيم ال design قبل ما اعمل فيه حاجة ... ال compiler بيقسم ال design الى cells بيطلق عليها Global routing cells او اختصارا **Gcells** وهى مساحة معينة من ال design فكإنى قسمت ال design الكبير الى اجزاء صغيرة وده بيساعده زى ما هنشوف انه يبص على كل جزء لوحده ويتعامل معاه ويصلح ال violations الى فيه بطريقة منظمة كإنى بشوف حتة حتة فى ال design بس عامل standard size للحتة دى



Dividing the design into Gcells

ال design عندى بعد ال placement عبارة عن cells فعايز اوصل بينهم ب nets الى هى يعنى ال wire الى هيوصل بينهم جوا ال design ... فانا محتاج مبدأيا احدد ال path الى هتمشى فيه كل net وهيكون فى انى metal layer ... وده الى بيتعمل فى الخطوة دى انه

مجرد بتحدد ال path الى هتاخده ال net دى بس مش بتعمله بالفعل عشان كده بيكون شكله عشوائى زى الى فى الصورة تحت مش vertical او horizontal ... وكمان بتحدد هتستخدم انى metal layers لل net وهل هتحتاج اكتر من واحدة مثلا وهتوصل بينهم ب vias ولا لا ومدام هيحدد ال metal layers الى هتمشى فيها ال net يبقى كده هو حدد هل ال routing بتاع ال net دى هيكون vertical ولا horizontal لان ال routing direction لكل layer متحدد من ال floorplanning ...وبما اننا قلنا انه هيتعامل مع ال design على انه Gcells فهو هيحدد ال nets الموجودة فى كل Gcell وياخد فى اعتباره انه ميحطش nets اكتر من ال available tracks الى فى Gcell دى عشان ميحصلش ال overflow الى اتكلمنا عنه فى ال floor planning ... ثم يحدد ال path الى هتاخده كل net فى Gcell دى وال metal layer الى هتستخدمها زى ما قلنا



Global Routing Operation

يبقى فى global routing هو هيحدد ال nets الى عنده ويشوف دى بتنتمى لأنى Gcell ويحدد ال path الى هتاخده وال metal layers الى هتستخدمها لكن هو معملش اى حاجة physical ده مجرد تحديدات ... ناخد بالنا ان ال nets بتكون من Gcell للتانية عادى مش المقصود انه Gcell الى فيها مش هيوصل على Gcell تانية ... دى عملية تنظيمة فقط

فى ال global routing بيكون ال compiler وهو بيحدد ال path لكل net واخد فى اعتباره كذا حاجة زى:

- انه ميقربش من power mesh routing يعنى ال routes بتاعة ال ring & straps & rails كل دول مش هيستخدمهم
- ال routing blockages بحيث ميعملش routing فيها
- يقلل ال congestion ولكن بردوا يقلل ال detour لانه بيزود ال delay زى ما شرحنا بالتفصيل فى CTS

على الجانب الاخر هلاقى انه مش بيهتم ب Design rule checks DRCs خالص هنا يعنى مثلا overlapped paths ممكن تكون موجودة عادى واى حاجة تعمل DRC violation ممكن انك تلاقيها لانه مش بيبصلها فى الخطوة دى

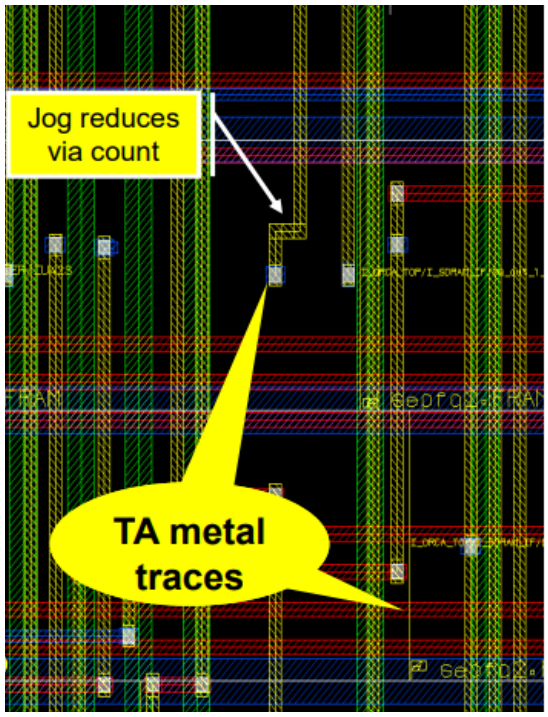
==========

2. Track Assignment:

بعد ما حددت ال paths بتاعة ال nets وحددت كل path هيكون فى انى metal layer ... هاجى فى الخطوة دى احدد ال tracks بتاعة كل path يعنى فى ال global قالى ال net دى هتاخد path فى metal layer 1 مثلا ففى ال track assignment هبدأ احدد هى هتمشى فى انهى track فى metal layer 1 لإن زى ما احنا عارفين من ال floorplanning ان كل metal layer فيها tracks ... فأبدا احدد ال path الحقيقى بقى هيكون شكله ازاى يعنى ال physical path بال horizontal & vertical directions ... فكده هو حدد ال tracks وحدد ال metal trace بالفعل على ال tracks دى ... بس لسه ال metal نفسه متحطش ده تحديد دقيق للاماكن الى هيتحط فيها

بيكون مهتم انه يطول ال path على انه يخليه قصير وينزل ب via ل metal layer تانية ... يعنى بمعنى اخر بيفضل انه يمشى طريق طويل على نفس ال metal layer على انه يقطع الطريق الطويل ده بانه ينزل ب via ل metal layer تانية مثلا ويكمل طريقه ... لإن ال via بتكون ال resistance بتاعتها عالية نظرا لصغر مساحتها وبالتالى كده بيزود ال delay اكتر من لو كمل فى straight path فى نفس ال metal layer فال tool بتحاول تقلل عدد ال vias

حتى انه مثلا لو ماشى vertical فى track معين فى metal layer وعايز يمشى حتة horizontal ثم يرجع vertical تانى فاول حاجة هتفكر فيها انه ينزل من ال vertical direction metal layer ب via ويوصل للى تحتها الى هتكون horizontal direction زى ما احنا عارفين ثم يمشى الجزء ال horizontal الى عايزيه ويرجع يطلع لل vertical metal layer تانى ب via بس لإنه بيحاول يقلل ال vias فممكن تلاقيه عمل حاجة تانية اسمها **jog** وهى انه فى نفس ال vertical metal layer دى هيمشى horizontal الحتة الى عايزها ثم يرجع vertical تانى زى الصورة تحت كده ... هتلاقى انه مشى vertical فى two tracks والحتة ال horizontal كانت على نفس ال layer كإنها وصلت بين ال two tracks دول



TA Metal Traces & Jog

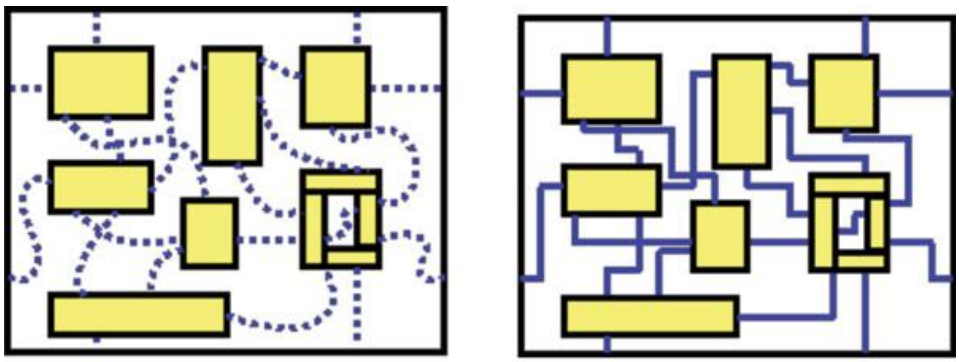
فى الخطوة دى بردوا ال tool مش بتهتم بال DRCs خالص يعنى هتلاقى violations عادى بعدها ... يبقى فى track assignment هياخد كل global route ويحطه بالفعل فى metal layer بتاعته فى track معين فيه

==========

3. Detailed Routing:

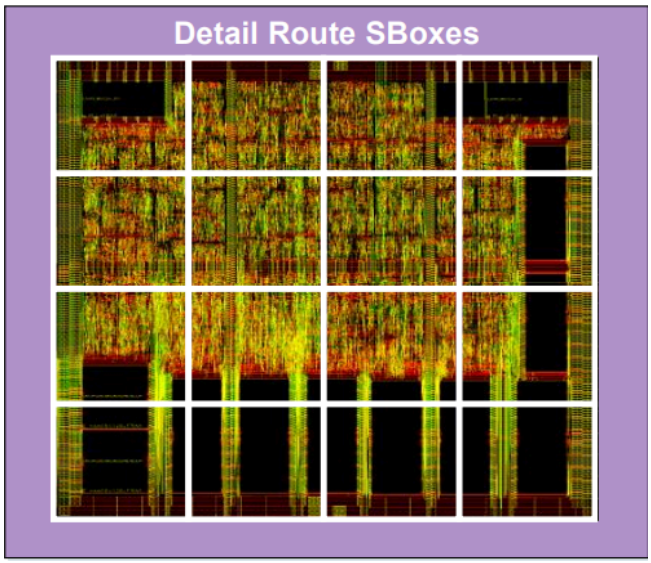
هنا بقى هنحط ال metal بالفعل فى الاماكن المتحددة لكل route ونحل ال DRCs الى ال tool مكنتش مهتمة بيها قبل كده ... يبقى ال tool هتاخد ال routing plan الى اتعملت فى global route ثم تاخد ال tracks الى اتحددت لكل route فى ال plan دى فى track assignment وتبدأ تحط ال metals بالفعل وتوصل بين ال nets & pins يعنى هتعمل ال routing فعلا خلاص

وبالتالى لو جيت اقارن بين ال routes الى عملتها ال global routing والى اتعملت فى detailed routing هلاقى زى الصورة تحت كده ... لإن ال detailed استخدمت ال tracks المتحددة فى كل metal layer ب horizontal or vertical directions ... يعنى عملت شكل ال routing الحقيقى المطلوب على عكس ال global كان مجرد توصيل عشوائى يحدد ال routing plan فقط



(a) Global routing (b) Detailed routing

ثم زى ما قلنا هتحتاج بقى تصلح ال DRCs بعد ما تحط ال metal routes ... فهتصلحه ازاى؟؟ احنا قلنا ان ال tool مقسمة ال design الى gcells فالى هيحصل انها هتبص على ال design فى صورة اجزاء كل جزء فيه عدد من gcells وتصلح اى DRC violation فيه ثم تروح للجزء الى جمبه وهكذا لحد ما تخلص ال design ... كإنها حددت box ب size معين وبما ان هى بتشوف ال design على هيئة عدد من gcells فال box هيكون ال size بتاعه هو multiple of gcell ويطلق عليه switch box or Sbox يعنى مثلا لوعندى 4*4 Sbox يبقى ده فيه 16 gcell بحيث ان طول ال box هو 4gcells وعرضه هو 4gcells يبقى هو بيحتوى على 16gcells ... فال tool تبص على الى جوا ال box ده تشوف اى DRC violation وتحله لحد ما ال box كله يكون clear فتروح لل box الى بعده الى جواه منطقة تانية من ال design وهكذا لحد ما تخلص

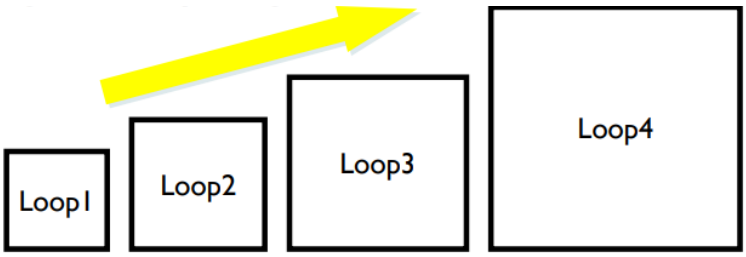


المشكلة هنا ان ال design كله اتقسم ل Sboxes ليهم نفس ال size جمب بعض كده زى ما باين من الصورة فوق ... طب ما ممكن يبقى فيه DRCs فى جزء ال design الى عند ال boundary بين اى two boxes ... يعنى ال tool بتصلح ال violations الى جوا ال Sbox مش الى ال box boundary واقف عليها ... فهنعمل ايه؟؟ هنعمل اخر خطوة وهى ال search and repair ودى الى هنصلح فيها اى DRCs violations لسه موجودة

==========

4. Search and repair:

هنا هحل اى DRC violations لسه موجودة زى ما قلنا وهنستخدم نفس concept ال Sbox بس الفرق انه هعمل كذا iteration وكل واحدة ب size مختلف فمثلا فى الاول هستخدم sbox size يكون 2*2 وامشى على ال design كله بنفس الطريقة الى شرحناها فى ال detailed routing ثم ال iteration التانية هستخدم size اكبر مثلا 3*3 وامشى على ال design كله تانى وبالتالى اى DRC كان موجود عند ال boundary وقت استخدام sbox 2*2 هيكون جوا ال boundary وقت استخدام sbox 3*3 وهكذا كل ما اكبر ال size هيكون boundary ال sizes الاصغر جواه ... وهكذا هعقد اكبر فى ال size لحد ما اخلص ال design كله كدا



==========

Non Default Routing Rules & Crosstalk

اتكلمنا عنهم بالتفصيل فى Clock Tree Synthesis ممكن تراجع عليهم من هنا CTS

==========

Signoff DRC checker

بعد اخر خطوة فى routing فالمفروض هعمل ال checks على ال design والى منهم DRC check الى اتصلح كله بعد search and repair بس DRC check ده باستخدام ICC tool مقدرش اعتمد عليه كأخر DRC check على ال design قبل ما يروح المصنع ... ليه؟؟ عشان routing DRC rules هى جزء من Technology DRC rules مش كلها ... ال design بتاعى ليه كذا view او بمعنى اخر ممكن اشوفه بكذا شكل ... view يخليني اشوف كل التفاصيل الى فيه و view تانى اشوف فيه بعض التفاصيل مش كلها ... ال ICC مش بيشوف كل تفاصيل ال design هو بيشوف FRAM view الى هو زى lef الى اتكلمنا عنه فى libraries وده زى ما احنا عارفين مش بيكون فيه كل التفاصيل يعنى مش بيتكلم عن poly مثلا ولا بينزل لل transistor level واكيد كل الحاجات دى ليها بردوا DRC rules لازم ميكنش فيها اى violations وبردوا ICC بيتعامل مع اى Macro على انه black box مش بيعمل check على الى جواه ... فكده ناقص حاجات كتير اوى فى DRC rules متعملش عليها check ... ال tools المستخدمة لعمل signoff DRC check هى

Hercules / ICV / Calibre

==========

ICC tool

1. Checks before Routing:

Command:

check_routeability -error_cell cell_name

ال command ده هو الاكثر استخداما وكافى لل prerouting check … بيعمل check ان ال design جاهز لل routing او بمعنى اخر انه routable design فبيبص مثلا على ال pins يتأكد انهم محطوطين فى مكانهم مظبوط تبع ال constraints المحطوطة ليهم من قبل كده وان مفيش اى violations على اماكنهم ويتأكد ان ال wire tracks مظبوطة وبيبص على ال blockages فى ال design وعلى ال optimizations الى تقدر ال tool انها تعملها بعدين سواء هتصلح بيها errors او تعمل design optimizations فقط فيعنى بال command ده تقدر تعرف هل ال design جاهز لل routing ولا فيه اى violations لازم تتحل الاول وفى نفس الوقت ال tool بتجمع شوية معلومات عن ال design وال area الى تقدر ت optimize فيها وفى الاخر تقولك هل تمام ادخل على ال routing ولا لا … لو تمام ال command هيرجع 1 بعد ال checks زى كده:

```
icc_shell> check_routeability

Cell mips_16.err existed already. Replace it ...
Warning: Cell contains tie connections which are not connected to real PG. (MW-349)

============================================
==        Check Pin-Spot Min-Grid         ==
============================================


============================================
==         Check Pin out of bound         ==
============================================


No out-of-bound error found

============================================
==             Check Min-Grid             ==
============================================

No min-grid error found

============================================
==        Check for blocked ports         ==
============================================

No blocked port was detected

[        CHECK DESIGN] CPU = 0:00:05, Elapsed = 0:00:05

[        CHECK DESIGN] Peak Memory =    364M Data =     53M

Update error cell ...
1
```

ال errors الى بتطلع بيطلعها فى .err ده file بي get generated ويتحط فيه ال errors … وبالتالى هو مش هيطلع الا لو فيه errors ال default لاسمه انه يكون top_cell_name.err وحسب اسم ال top cell عندك يعنى بمعنى اخر ال top module name بتاعك … وال file ده بنطلق عليه error cell … ولكن تقدر تغير اسمه عن طريق ال option:

- error_cell: بحطله اسم ال error cell بدل ال default بتاعها فمثلا لو حطيت mydesign هيطلع اسمه mydesign.err

======

Command:

check_physical_design -stage pre_route_opt

ال command ده استخدمناه فى ال placement وال CTS زى ما شوفنا قبل كده بحيث انه يتأكد من ال design لحد كل خطوة فيهم انه تمام من خلال checks بيعملها حسب كل stage واقف فيها ... هنا بيعمل checks قبل ما نبدأ routing

- stage: هنا قبل CTS فكتبنا **pre_route_opt** زى ما شوفنا قبل كده **pre_place_opt** قبل ال placement و **pre_clock_opt** قبل ال CTS

بعد ما يخلص بيطلعلى ال errors & warnings ويرد ب 1 على ال command لو مفيش errors زى فى الصورة تحت وكل ده بيحطوا فى html file زى ما هو كاتب:

```
************************************************
Report : check_physical_design
Stage  : pre_route_opt
Design : mips_16
Version: G-2012.06-ICC-SP2
Date   : Thu Aug  6 23:20:19 2015
************************************************
Total messages: 0 errors, 6 warnings

----------------------------------------
Warning Summary for check_physical_design
----------------------------------------

  ------------------------------------------------------------------------------
  ID            Occurrences     Title
  ------------------------------------------------------------------------------
  PSYN-523      2               Geometries are not integer multiple of width or.
..
  ------------------------------------------------------------------------------

------------------------------------------------
Other Warning Summary for check_physical_design
------------------------------------------------

  ------------------------------------------------------------------------------
  ID            Occurrences     Title
  ------------------------------------------------------------------------------
  MW-349        2               Cell contains tie connections which are not con.
..
  ZRT-026       1               Layer %s pitch %.3f may be too small: wire/via-.
..
  ZRT-517       1               The %s.err error view already exists; the curre.
..
  ------------------------------------------------------------------------------
dump check_physical_design result to file ./cpd_pre_route_opt_2015Aug06232015_21
523/index.html
1
icc_shell>
```

======

Command:

all_high_fanout -nets [-threshold value] [input_coll] [-through_buf_inv]

Note: زى ما ذكرنا قبل كده ان ال fanout هو عدد ال cells الى واصل على ال output cell واتكلمنا عنه بالتفصيل فى design constraints فى ال synthesis واتكلمنا عن high fanout synthesis فى ال placement

ال command ده بقدر اعرف منه ال nets الى عندها high fanout فى ال design او فى جزء معين من ال design بطلق عليه collection ... ودى معلومة مهمة ابص عليها قبل ما اكمل بحيث اعرف ايه ال critical nets من ناحية ال fanout الى عندى لحد دلوقتى:

- nets: عشان يرجعلى ال high fanout nets عشان كده ده مش optional (مش محطوط بين [ ] يبقى لازم نحطه مش optional)
- threshold: هنا بحطله ال threshold الى لو ال fanout net اعلى منها يبقى ال command هيرجعها ك high fanout net وهو optional لإنه ممكن تحط قيمة ال threshold بطريقة تانية وهى انك تعمل set ل variable اسمه high_fanout_net_threshold وهو ده المستخدم ك default لو محطتش ال threshold option ... كمعلومة بردوا ان ال high_fanout_net_threshold ال default بتاعه 1000 لو انت مغيرتش قيمته وبياخد integers فقط
- input_coll: ده بحط مكانه اسم ال collection الى عايزه يدور على ال high fanout nets فيها ولو محطتهاش هيدور فى ال design كله فالغالب بيبقى تعويض عن اسم variable زى كده:

```
The following example shows the collection of all high-fanout nets from
an existing collection stored in $COLL:

  prompt> all_high_fanout -nets $COLL
  {ii2/hin iv3/in iv3/out tin ii2/hout tout}
```

- through_buf_inv: ال clock tree الى حطينها فى CTS ال tool بتعتبرها هنا حاجة واحدة يعنى مش هتبص على ال nets الى جواها كل واحدة لوحدها هى هتبص على ال network على بعضها كده فلما تحط ال option ده هتبدأ هى تبص على ال clock tree من جوا وتشوف كل net فيها هل هى high fanout ولا لا فمثلا فى الصورة لما حطيت all_high_fanout -nets مرجعش حاجة لانه مفيش high fanout nets وهو كده مش واخد فى اعتباره ال clock tree nets لكن لما زودت ال option وكتبت all_high_fanout -nets -through_buf_inv رجع net من ال clock tree ... ممكن ميلقيش بردوا فى ال clock tree وميرجعش حاجة بس يعنى المثال وضح الفرق بين انى اضيف ال option او لا:

```
icc_shell> all_high_fanout -nets
icc_shell> all_high_fanout -nets -through_buf_inv
{n39 clk n155}
icc_shell>
```

======

Command:

all_ideal_nets [input_coll]

ال command ده بقدر اعرف منه لو فى ideal net عندى يعنى nets عندها zero capacitance and zero resistance فالمفروض انى لو استخدمته فى المرحلة دى قبل ال routing ميطلعش حاجة لو كله تمام لان خلاص مبقاش فيه حاجة ideal دلوقتى:

- input_coll: ده بحط مكانه اسم ال collection الى عايزه يدور فيه زى ما ذكرنا قبل كده فى all_high_fanout command ولو محطتوش هيدور على ال ideal nets فى ال design كله

لو استخدمته سواء قبل ال routing او فى اى stage تانية وكان فيه ideal nets هيرجعوا بالشكل زى الامثلة دى كده:

```
The  following  example returns a collection of all ideal nets from the
design:

  prompt> all_ideal_nets
  {ii2/hin iv3/in iv3/out tin ii2/hout tout}

The following example returns a collection of all ideal  nets  from  an
existing collection stored in $COLL:

  prompt> all_ideal_nets $COLL
  {ii2/hin iv3/in iv3/out tin ii2/hout tout}
```

==========

2. Pre-Route Settings:

Command:

set_delay_calculation_options [-preroute elmore | awe] [-routed_clock elmore | arnoldi] \
[-postroute elmore | arnoldi] [-awe_effort low | medium | high] [-arnoldi_effort low | medium | high]

ال tool بتستخدم algorithms عشان تقدر تحسب ال delay لل cells & nets ففيه كذا algorithm متاحين وهما elmore و awe و arnoldi ممكن تبحث عن كل واحد لو عايز تعرف تفاصيل عنه بس المهم بالنسبالنا حاليا انه فيه three algorithms وال default هو elmore وبالتالى هتلاقى ال options عبارة عن انى بحدد ل stages معينة انه يستخدم model من التلاتة فى حساب ال delay او انى بحدد ال model effort:

- preroute: هنا بحددله ال delay model الى يستخدمه لحساب ال delays لل design قبل ال routing والمتاح هو elmore و awe وال default هو elmore
- routed_clock: هنا بحدد ال model المستخدم لحساب ال delay ل clock tree nets سواء بعد ال CTS او بعد ال routing والمتاح هو elmore و arnoldi وال default هو elmore ... وبالتالى ال model المستخدم لحساب ال delay لل design عادى يختلف عن المستخدم لحساب ال clock tree delays
- postroute: بنفس الفكرة بختار ال delay model الى هيستخدمه فى حساب ال delay لل design بس بعد ال routing
- awe_effort: بحدد لل tool ال effort الى تبذله وقت استخدام ال awe model ممكن low, medium or high وال default هو ال medium
- arnoldi_effort: بحدد لل tool ال effort الى تبذله وقت استخدام ال arnoldi model ممكن low, medium or high وال default هو ال medium

امثلة لل command ده:

```
EXAMPLES
       The following command specifies that the Arnoldi delay  model  is  used
       with high effort for postroute designs:

         prompt> set_delay_calculation_options \
            -postroute arnoldi -arnoldi_effort high

       The following command specifies that the Elmore delay model is used for
       preroute designs:

         prompt> set_delay_calculation_options -preroute elmore

       The following command specifies that the AWE delay model is  used  with
       high effort in preroute designs:

         prompt> set_delay_calculation_options \
             -preroute awe -awe_effort high

       The following command specifies that the AWE delay model is used on the
       data nets and and the Arnoldi delay model is used  on  the  clock  nets
       after clock tree synthesis:

         prompt> set_delay_calculation_options \
            -preroute awe -routed_clock arnoldi
```

======

Command:

set_route_options [-default] [-groute_timing_driven true | false]\

[-groute_timing_driven_weight number] [-groute_congestion_weight number]\

[-groute_incremental true | false] [-track_assign_timing_driven true | false]\

[-track_assign_timing_driven_weight number] [-droute_connect_tie_off true | false]\

[-droute_connect_open_nets true | false] [-droute_reroute_user_wires true | false]\

[-droute_CTS_nets normal | minor_change_only] [-same_net_notch ignore | check_and_fix]

ال command ده بيظبط ال settings الى هتستخدمها ال tool فى ال routing steps عشان كده فيه options لل global routing هتلاقى بدايتها groute وفيه لل track assignment هتلاقيى بدايتها track_assign وفيه لل detailed وهتلاقى بدايتها droute فخلينا نبص على بعض ال options:

- default: كده هيستخدم ال default settings لكل ال steps
- groute_timing_driven: هنا بقوله هل يخلى ال global routing مهتم بال timing ولا لا ... لونفتكر من ال placement ذكرنا جزء ال timing driven وده معناه انى بخلى ال step الى بعملها دلوقتى تكون مهتمة بال timing يعنى ت meet ال timing حتى لو كان ده على حساب حاجات تانية زى ال congestion مثلا فلو خليت ال option ده ture ابقى بخلى ال global route يهتم بال timing وال default هو false
- groute_timing_driven_weight: بحط رقم من 1 الى 7 بيعبر عن درجة الاهتمام بال timing بمعنى انه 7 مثلا معناها انه اهم حاجة ال timing حتى لوعلى حساب ال congestion او اى حاجة تانية فال default هو 4 ودى قيمة متوسطة تخليه يهتم بال timing وفى نفس الوقت ميجيش على حساب باقى ال aspects

- groute_congestion_weight: هو بيهتم بال congestion فهنا احنا بنحطله ال weight نفس فكرة ال timing driven weight وبيكون من 1 الى 12 فكلما كان الرقم عالى كل ما هيهتم بانه يقلل ال congestion اكتر من اى حاجة تانية وال default بتاعه 4

- groute_incremental: لما اخليه true بيعمل كذا iteration فبضمن انه global routing results احسن كإنه كل مرة بيعمل optimization عن الى قبلها وال default هو false

- track_assign_timing_driven: نفس فكرة ال timing driven فى ال global بردوا هنا بخليه ياخد ال timing constraints فى اعتباره وال default هو false

- track_assign_timing_driven_weight: بحط رقم من 1 الى 10 بيعبر عن درجة الاهتمام بال timing نفس الى كان موجود فى global routing عادى بس ال default هنا هو 1

- droute_connect_tie_off: فى ال detailed routing بقوله هل يوصل ال tie-high and tie-low cells الى حطيناهم فى ال placement ولا لا (شرح ال tie high and tie low cells موجود فى ال placement file) وال default انه true

- droute_connect_open_nets: فى ال detailed routing بقوله هل يوصل اى unconnected pin ولا لا يعنى ممكن يكون فيه بعض ال pins فى ال design مش متوصلة بحاجة ولكن ممكن تعمل مشاكل فى ال design لو اتسابت فال default هنا true انه بيحاول ي connect اى open pins بس ممكن تخليه false لأسباب كتير زى انك عايز تعملها manually connected او انك قاصد تسيبها unconnected

- droute_reroute_user_wires: ممكن يكون ال user حط بعض ال routes بنفسه manually ومش هيعوز ان ال tool تغير فيهم فهنا بيقول لل detailed routes هل تغير فيهم عادى لو احتاجت لده فى ال optimization ولا لا فال default هنا false انها متغيرش فيهم

- droute_CTS_nets: ال detailed route ممكن يغير فى ال clock tree nets عشان ال optimization ولكن ال clock tree زى ما احنا عارفين انها critical جدا عشان كده بنعملها حتى قبل routing باقى ال design عشان كده ال default هنا هو minor_change_only عشان ميغيرش فيها بقدر الامكان ولكن لو خليتها normal هيغير فى ال clock nets بشكل كبير عشان ال optimization زى ما هيعمل فى ال design nets عادى

- same_net_notch: دى مشكلة ممكن تحصل وهى ان لو عندى wires جمب بعض وكلهم واصلين على نفس ال net لكنهم قريبين جدا من بعض يعنى المسافة بينهم اقل من ال min spacing فى ال DRC rules فدى مشكلة بقول له انه يحلها لما احط check_and_fix ولكن ال default هو ignore يعنى مش بيحلها by default فانا لازم اقول لل detailed route انه يحلها بالطريقة دى لو انا عايزه يحلها

مثال لل command:

```
icc_shell> set_route_options -groute_timing_driven true -groute_incremental true -track_assign_timing_driven true -same_net_notch check_and_fix
1
icc_shell>
```

======

Command:

set_si_options [-delta_delay true | false] [-min_delta_delay true | false]\

[-static_noise true | false] [-timing_window true | false] [-route_xtalk_prevention true | false]

ال command ده بيحط options تخص ال signal integrity بحيث ال tool تعرف هتاخد ايه من ال non ideality فى اعتبارها وايه لا وبعض options ال command هى:

- delta_delay: ده بيعبر عن ال delay الى بيحصل لل signal بسبب ال cross talk الى اتكلمنا عنه بالتفصيل فى CTS فهنا اما بخليه true ال tool بتاخد فى اعتبارها تأثير ال cross talk على ال signal delay وبالتالى ال calculations بتكون accurate اكتر ولكن ال default بتاعه انه false … لما بتخليه true فال min_delta_delay option هو automatically بيكون true
- min_delta_delay: ده بيعبر عن انه فى timing analysis and optimization هياخد فى اعتباره ال cross talk عشان كده بيكون automatically set لما بتخلى delta_delay option = true وده بيساعد فى ال setup analysis عن طريق انه بيحط min noise propagation لل capture clock وبالتالى min delay to clock capture path ودى worst case فى ال setup زى ما شرحنا بالتفصيل فى crosstalk effects فى ال CTS … وبردوا بيحسب ال min timing paths فى حالة وجود crosstalk وبالتالى بي check ال hold كمان
- static_noise: بقول لل tool انه تاخد فى اعتبارها تأثير ال static noise على ال signals فى ال analysis وال noise دى الى هى جاية مثلا من supply noise او من تغيير ال environment زى ال temperature او من ال crosstalk … وال default ليها هو false
- timing_window: ال timing window هى قبل وبعد ال transition بتاع ال signal يعنى بمعنى اخر بتمثل الوقت ال critical الى بتتغير فيه ال signal … لو فيه اكتر من signal بيتغيروا فى نفس الوقت بالتالى هيحصل overlapping بين ال timing windows بتاعتهم وده بيأثر بشكل كبير على ال crosstalk زى ما شرحنا بالتفصيل فى CTS فلما اخلى ال option ده ture معناه انه بخليه ياخد فى اعتباره تغير ال signals فى نفس الوقت (او بمعنى اخر ال timing window overlapping) فى crosstalk analysis وهو بيكون false by default ولازم عشان تعمله true يكون delta_delay=true او static_noise=true
- route_xtalk_prevention: هنا لو خليته true ال tool بتبدأ تستخدم techniques بحيث انها تقلل ال crosstalk على قد ما تقدر ولكن ال default بتاعه هو false

مثال عليه:

```
EXAMPLES
       The following example removes all annotated delays, specifies crosstalk
       delta delay and static noise calculation and defines the threshold.

         prompt> set_si_options -delta_delay true -static_noise true -static_noise_threshold_below_high 0.3
```

==========

3. Hold Time Fixing:

Command:

set_prefer [-min] {cell_list}

زى ما اتكلمنا عنه فى CTS ... ده بستخدمه بشكل عام من غير min عشان اقول للـ tool تستخدم الـ cells الى انا حطاها فى الـ cell list وقت الـ optimization لكن لما اضيف min يبقى بقولها تستخدمها فقط فى hold violations fixing وبالتالى ساعتها الـ list هيكون فيها buffers & inverters ... وطبعا لازم اى cell فى الـ list تكون فى target library

======

Command:

set_fix_hold [all_clock or put the clock name]

زى ما اتكلمنا عنه فى CTS: ده بستخدمه عشان اخلى الـ tool تصلح كل الـ hold violations الموجودة واما احطلها اسماء الـ clock sources الى عايز اصلح الـ hold violations فى الـ tree بتاعتهم او انى اكتب all_clocks وهى هتصلح لكل الـ trees الى عندى

======

Command:

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

زى ما اتكلمنا عنه فى CTS: ده بستخدمه عشان اخلى الـ tool تصلح كل الـ hold violations الموجودة بس بحطله اولويات يصلح عليها من خلال الـ options دى

- default: هنا بتهتم انها متأثرش على الـ worst negative slack و max transition time دول بيكونوا اهم حاجة هنا بس كده ممكن تلاقى degradation فى total negative slack لإنه مهتم بال worst فقط فممكن يحصل degradation فى اى negative slack تانى
- prioritize_tns: هنا بتهتم انها متأثرش على الـ worst negative slackو total negative slack و max transition time دول بيكونوا اهم حاجة هنا
- prioritize_min_delay: هنا الـ tool هتحسن فى min delay اكتر من الـ max بمعنى اوضح انها هتهتم تصلح الـ hold violations حتى لو ده كان على حساب الـ setup فممكن بعد ما تصلحه الاقى timing violations بردوا من الـ setup
- preferred_buffer: بخلى الـ tools تستخدم الـ buffers الى حطيتلها فى min option فى set_prefered command غير كده فهتستخدم هى اى buffers أثناء تصليح الـ hold
- effort: بحدد الـ effrot الى هتبذله الـ tool فى الـ fixing والـ default بتاعه medium

==========

4.Routing Operation:

Command:

route_opt [-stage global | track | detail] [-xtalk_reduction] [-only_xtalk_reduction]\

[-effort low | medium | high] [-power] [-size_only] [-optimize_wire_via] [-area_recovery]\

[-wire_size] [-initial_route_only] [-skip_initial_route] [-incremental] [-only_wire_size]\

[only_hold_time] [only_design_rule] [only_power_recovery] [only_area_recovery]

ال command بيعمل كل steps ال routing ثم بيعمل postrouting optimization فال output من ال command ده هيكون routed and optimized design حسب كل ال constraints & settings المحطوطة وبالتالى ال options بتاعته هيكون فيها انى بحدد ال routing stage يعنى هيعمل global ولا track ولا detailed ولا كلهم وفيها الطرق الى يستخدمها فى ال optimization يعنى مثلا يغير فى cells size ولا ال wire size ثم بقوله ال optimization target يعنى هل يعمل optimization بهدف تحسين ال power ولا ال area مثلا وبعد ما يخلص هيكون قيه violations سواء timing او DRCs فبحددله بردوا يصلح ايه فيهم ... بعض ال options:

- stage: هنا بحدد ال stage الى هوصلها ثم ي run ال optimization بعدها يعنى مثلا لو track يبقى هيعمل global routing ثم track assignment ثم ال optimization ويقف على كده ... لو محددتش ال stage فال default هو detail يعنى هيعمل global routing ثم track assignment ثم detailed routing ثم بعد كده ي run ال optimization
- effort: المجهود الى هتبذله ال tool فال post-route optimization وال default هو medium وكل ما تزود هتعمل optimization اكتر فمثلا ال high بيخليها تعمل three optimization loops
- initial_route_only: بقول لل tool تعمل routing من غير optimizations وبالتالى مينفعش استخدمه مع options زى skip_initial_route و incremental
- skip_initial_route: بقول لل tool متعملش routing وتعمل optimizations فقط وبالتالى لازم يكون ال design بتاعى fully routed يعنى كل ال routing steps خلصانة ... لو ال signal integrity كانت enabled فهتعمل ECO routing من غير signal integrity ثم بعدها تعمله مع signal integrity وده مختلف عن الى بيحصل فى ال incremental option زى ما هنشوف .. مينفعش استخدمه مع options زى initial_route_only و incremental و stage
- incremental: هنا بردوا مش هتعمل initial routing وهتعمل بس optimization و ECO routing وهتكون incremental process ولكن لو كان signal integrity موجود فهتعمل ECO routing with signal integrity ولكن مش هتعمل ECO routing من غير ال signal integrity ... مينفعش استخدمه مع options زى initial_route_only و skip_initial_route و stage
- xtalk_reduction: يعمل optimization بهدف تقليل ال crosstalk بس عشان احط ال option ده لازم اكون جهزتله فى ال pre-route settings وده نعمله عن طريق اننا نخلى route_xtalk_prevention option ب true فى ال set_si_options

command والا ال tool هتطلع error … ال default ان ال option ده disabled يعنى لو كتبت ال command منغيره فهو مش هيعتبره موجود

- only_xtalk_reduction: يعنى يعمل optimization بهدف تقليل ال crosstalk فقط وعشان استخدمه لازم اكون جهزتله فى pre-route settings بنفس الطريقة الى قولناها فى xtalk_reduction option
- power: بيبدأ يستخدم topologies يقلل بيها ال leakage power فى ال optimization loops الى بيعملها يعنى يعمل optimization بهدف تحسين ال power
- optimize_wire_via: بقول لل tool تعمل extra optimization لل wire length وعدد ال vias
- area_recovery: هنا بقول لل tool انها تعمل optimization فى ال area يعنى تقلل ال area اكتر بمعنى انها تستبدل ال cells ب cells اصغر … بس كده ال delay بتاع ال path الى فيه ال cells دى هيزيد لانه زى ما شرحنا فى files تانية قبل كده انه اما cell area بتقل يبقى delay بتاعها زاد وبالتالى ممكن يحصل timing violations فى ال path ده فمحتاج ان تكون ال paths ليها positive setup slack كويس
- only_power_recovery: بقوله ان ال optimization الى هتعمله هيكون عشان تحسن ال leakage power فقط
- only_area_recovery: بقوله ان ال optimization الى هتعمله هيكون عشان تحسن ال area فقط
- wire_size: بسمح لل tool انها تستخدم rules مختلفة لل wire sizing الى هى non-default routing rules الى اتكلمنا عنها فى CTS وشوفنا ازاى نحددها … وبالتالى لازم اكون محدد non-default routing rules الاول
- size_only: يعنى فى ال optimization استخدم فقط طريقة انه تغير فى cell sizing وده مناسب لل designs الى بتكون critical فى postroute optimization يعنى مش هتغير فى ال routes … هتعمل cell sizing بس
- only_wire_size: يعنى بقول انه استخدم فقط ال wire sizing المختلفة فى ال optimization وبالتالى محتاج اكون معرف ال non-default routing rules قبلها نفس الى احتاجنا نعمله فى wire_size option
- only_hold_time: بقوله ان ال violations الى اتصلحها هى فقط ال hold time violations وبالتالى محتاج ابقى عامل enable لل hold fixing وده عن طريق set_fix_hold command تبقى كاتبه قبل ال route_opt
- only_design_rule: بقوله ان ال violations الى اتصلحها هى فقط ال Design rules violations (DRCs)

ال **ECO** هى Engineering Change Order ودى هنتكلم عنها بالتفصيل فى chip finishing ولكن ناخد فكرة بسيطة عنها دلوقتى وهى انها عبارة عن تغيير فى ال design بعد ما يكون طلع ال layout خلاص واحنا بنعمل checks عليه زى STA, DRCs وغيرهم هنلاقى violations فهيحتاج نرجع نغير فى PnR

أمثلة لل command:

```
EXAMPLES
       The following example runs initial routing and  stops  without  running
       optimization:

         prompt> route_opt -initial_route_only

       The following example performs only global routing:

         prompt> route_opt -initial_route_only -stage global

       The  following  example  performs only global routing and track assign-
       ment:

         prompt> route_opt -initial_route_only -stage track

       The following example runs only detail routing (if you  are  using  the
       classic router, it also runs search and repair):

         prompt> route_opt -initial_route_only -stage detail

       The  following  example  runs  only  optimization and skips the initial
       routing steps:

         prompt> route_opt -skip_initial_route

       The following example runs only incremental optimization:

         prompt> route_opt -incremental

       The following example runs routing-based crosstalk reduction  in  addi-
       tion to the default route_opt signal integrity flow:

         prompt> route_opt -xtalk_reduction
```

==========

5. Post Routing Optimization:

Command:

psynopt [-area_recovery] [-power] [-congestion] [no_design_rule | -only_design_rule]\
[-size_only | -in_palce_only] [-only_area_recovery] [-only_power] [-continue_on_missing_scandef]\
[-only_hold_time]

ده بيعمل optimizations واتكلمنا عنه فى ال placement وقولنا انه اقدر بردوا استخدمه بعد ال routing اخليه يعمل final optimizations فهو بي optimize ال design بشكل عام .... واقدر اتحكم فى ال optimizations الى تتعمل من ال options .. بعض ال options هى:

- area_recovery: هنا بقول لل tool انها تقلل ال area اكتر بمعنى انها هتسبدل ال cells ب cells اصغر ... بس كده ال delay بتاع ال path الى فيه ال cells دى هيزيد لانه زى ما شرحنا فى files تانية قبل كده انه اما cell area بتقل يبقى delay بتاعها زاد وبالتالى ممكن يحصل timing violations فى ال path ده فمحتاج ان تكون ال paths ليها positive setup slack كويس
- congestion: هنا بقول لل tool انها تعمل optimizations لتحسين ال congestion
- power: هنا بقول لل tool انها تعمل optimizations لتحسين ال power

- no_design_rule | only_design_rule: بختار واحدة منهم وهنا ببقى عايز اقول هل تعمل fixing لل design rules violations بس يعنى متعملش fixing لل timing violations فاستخدم only_design_rule او انى اقول لل tool متعملش fixing لل design rules violations اصلا فاستخدم no_design_rule ... لو مستخدمتش ولا واحدة فهتمشى على ال default وهى انها تعمل fixing للاتنين ال design rules violations & timing violations
- size_only | in_palce_only: الاتنين دول بيخلوا ال optimizations تكون بتغيير حجم ال cells وبس يعنى ميحصل remove او add ل cells ... الفرق بينهم ان in_place_only بتحط constraints on ECO placement change ... طيب ال ECO دى اختصار ل Engineering change order فهنا المقصود هو التغير الى هيحصل فى ال physical placement الموجود دلوقتى فيعنى دى مش هتغير فى اماكنهم الموجودة ... لكن size_only ممكن تغير عادى مش بتحط constraints on ECO فاحنا بنستخدم واحد من ال options دى بس ... وبيكون recommended انك تستخدم in_place_only بعد ما تخلص routing يعنى فى تقفيل ال design خلاص ويكون legalized design بحيث انك تعمل optimizations من غير تغييرات فى ال design بشكل كبير فمحتاج option يحط constraints on ECO
- only_area_recovery: يعنى تعمل optimization لل area recovery لكن متعملش لل design rules او ال timing وبالتالى ده مقدرش استخدمه مع وجود only_design_rule & no_design_rule options
- only_power: يعنى تعمل optimization لل power فقط
- only_hold_time: يعنى هنصلح ال hold timing violations فقط وبالتالى بحتاج set_fix_hold command عشان اعمل hold fixing
- continue_on_missing_scandef: بقول لل tool تكمل حتى لو ملقتش scan chain file ودى شرحناها فى place_opt

==========

6. Violations Check:

Command:

```
verify_zrt_route [-nets {collection_of_nets}] [-open_net  true | false]\
[-report_all_open_nets true | false] [-drc true | false] [-antenna true | false]\
[-voltage_area true | false]
```

ده بيعمل check على DRCs violations و open nets و antenna violations وهنتكلم عنها بالتفصيل فى chip finishing وكمان check على ال voltage area violations ... بعض ال options:

- nets: بحدد nets معينة فى ال design يعملها verification ولو محطتوش فال default انه ي verify كل ال nets
- open_net: لو true يعنى يعمل check على ال open nets ولو false يبقى مش هيعمل check على ال open nets وال default هو true
- report_all_open_nets: لو true يعنى يعملى reporting لل open nets الموجودة وال max الى يعمله هما 200 open nets ولكن ال default هو false

- drc: لو true هيعمل check على ال DRCs وال default هو true
- antenna: لو true هيعمل check على ال antenna violations وال default هو true
- voltage_area: لو true هيعمل check على ال voltage-area rule violations وال default هو true

أمثلة لل command:

```
EXAMPLES
       The following example reports DRC violations, open nets, antenna viola-
       tions, and voltage-area violations for all nets in the design.

         prompt> verify_zrt_route

       The following example checks for open  nets,  antenna  violations,  and
       voltage-area violations, but not for DRC violations.

         prompt> verify_zrt_route -drc false

       The  following  example  checks  for open nets, antenna violations, and
       voltage-area violations for nets N1, N2, and N3.  Note that these rules
       are checked by default, and DRC violations are not checked.

         prompt> verify_zrt_route -nets {N1 N2 N3}

       The  following example reports only antenna and voltage-area violations
       for nets N1 and N2.

         prompt> verify_zrt_route -nets {N1 N2} -open_net false
```

==========

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 routing اعمل save ل MW cell جديدة الى هى ال design بتاعى لحد الخطوة دى واديها اى اسم وليكن design_routing

==========

طبعا لازم قبل ما اقول انى خلصت ال routing اشوف ال reports واتاكد ان timing clean و مفيش high congestion الى اتكلمنا عنه وعرفنا نشوفه ازاى فى placement file ثم بعد كده ننتقل لشوية حاجات هنضيفها ونطلق على المرحلة دى chip finishing ثم نعمل verification قبل signoff الى قولنا انه يختلف عن الى بنعمله فى ICC Tool وده هيكون topic منفصل وهو signoff checking

==========