

# Floorplanning Stage (V1)

Written by: Fatma Ali

## Content:

- What is expected to be done?
- Inputs to floorplanning
- Die area and shape / core area
- IO Pads
- Site rows
- Wire tracks
- Placement Blockages
- Macro Placement
- Keep out margin
- Well tap cells
- ICC tool:
  - Open ICC tool
  - Commands used for preparing PNR inputs
  - Floorplanning Commands

---

## What is expected to be done?

خدناه فكرة عنها فى introduction to PNR وعرفنا اننا بنظبط اماكن كل حاجة هنا بس هنبدأ نشوف بالتفصيل احنا بنعمل ايه بالظبط ونتعرف على concepts جديدة كمان واحنا ماشين بس خرينا الاول نشوف احنا ايه الى هنعمله بالظبط هنا حتى لو مش فاهمين معانى كل point دلوقتى فاحنا هنتكلم عن كل واحدة بالتفصيل بس نطهم فى points ورا بعض دلوقتى عشان منقاش تايبين فاحنا هنا هنعمل:

1. Determine die area, shape and core area
  2. IO pads locations
  3. Site rows created
  4. Wire tracks created
  5. Placement Blockages determination
  6. Macros placement
  7. Adding well tap cells (physical only cells)
-

## Inputs to Floorplanning:

نبدأ بقي ندخل في كل واحدة فأول حاجة هي ال inputs اصلا لل floorplanning هتكون ايه؟؟ اكيد هيجي في بالنا اننا هندخل ال netlist الى طلعت معنا من بعد ال synthesis & dft وبما اني هبدأ احدد اماكن ال cells فاكيد هدخل ال libraries والكلامده مضبوط ولكن عندنا كمان شوية inputs زيادة عليهم فخلينا نفس كل input لوحده ونشوف ايه الجديد:

**1. Gate level netlist .v:** ودي اخر netlist وصلناها الى هتطلع من dft

**2. .lef & .lib Standard cell libraries:** هحتاج lib. اكيد عشان اما هعمل timing checks و .lef هي الى هتخليني احدد اماكن ال cells واماكن ال blockages الى فيها زي ما شرحناها بالتفصيل في libraries file

**3. .sdc Synopsys design constraints:** لازم هدخل ال constraints اصلا الى tool بتشتغل عشان تحققة

**4. Technology file:** وده بيكون فيه DRC design rules الى بنمشي عليها واماكن ال wires في كل layer والى بنسميها tracks والمسافات بين كل track والتاني وال width ومعلومات عن ال metal layer نفسها زي height, width, ...., وبيكون ال extension بتاعه tf. يعني باختصار كده ده فيه كل ال parameters بتاعة ال metal layer بيكون شكله زي كده:

```
Technology {
    dielectric      = 3.7
    unitTimeName    = "ns"
    timePrecision   = 1000
    unitLengthName  = "micron"
    lengthPrecision = 1000
    gridResolution  = 5
    unitVoltageName = "v"
}

...
Layer "m1" {
    layerNumber      = 16
    maskName         = "metal1"
    pitch            = 0.56
    defaultWidth     = 0.23
    minWidth         = 0.23
    minSpacing       = 0.23
}
```

**5. TLU+ file:** وده بيكون فيه كل layer عندها R&C بكام per unit length يعني بطلع منه ال parasitics والى منها اعرف ال delays وبلاقي منه two files واحد max وده الى هطلع منه ال max delay وواحد min وده الى هطلع منه ال min delay ... وطبعاً دول محتاجهم عشان ال timing analysis

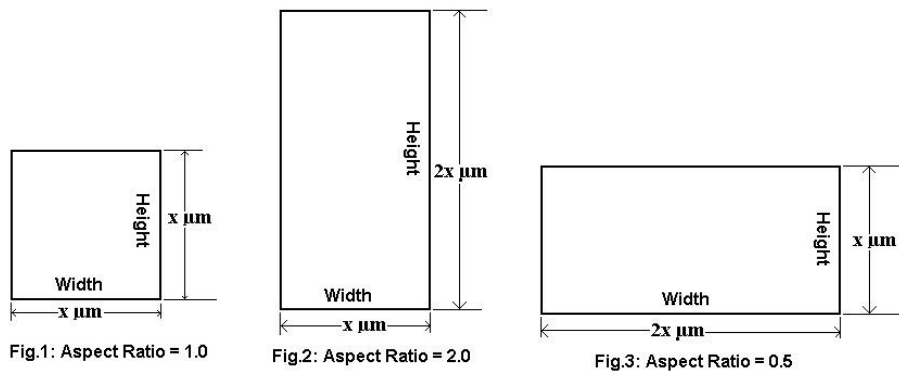
**Note:** ال TLU+ دلوقتي ممكن يكون جاي من ال vendor وممكن لا فعشان تطلعه بتستخدم file تاني بيكون جاي من ال vendor بردوا وهو interconnect technology file ITF وده بيكون فيه معلومات عن ال physical dimensions of the cells زي مثلاً ال dielectric نوعه اسمه x مثلاً وال dimensions بتاعته كذا او ال poly او ال conductor فده كده جواه معلومات عن ال interconnects كلها فبدخلة على tool ويطلع منه TLU+ بس غالباً بقي يتبع ال TLU+ نفسه على طول

**6. Milky way (MW) library:** ودى احنا الى بنعملها ب command وبنحفظ فيها MW cells كل cell هي عبارة عن snapshot للخطوة الى بنعملها مثلا بعد ال floorplanning او بعد ال power planning وهكذا بحيث انى لو حبيت ارجع لاي خطوة منهم وانا مشيت لقدام فى الخطوات فارجع من ال cell بتاعتها فهلاقى انه فى بعد كل خطوة فى PNR مش بس ال floorplanning بعمل save ل cell جديدة تمثل ال design بتاعى لحد ما انا واقف دلوقتى ... فبيكون فيه طبعا كل التفاصيل بتاعة ال design لحد ما وقفت

خلينا بقى ندخل فى الخطوات الى بنعملها ونشوف ازاي هنوصل للحاجات الى قولناها فوق دى وايه معنى كل point اصلا

## **Die area and shape / Core area:**

هنا بحدد شكل ال die بتاعتى هتكون مثلا مربع ولا ايه واحد ال area واحد ال core area بردوا الى هو الجزء من ال die الى هيكون فيه ال logic بتاعى وعشان كده ال die area اكبر من ال core area .. وهنا بحد ال aspect ratio ودى height/width:



بحدد ال core area باستخدام ال area الى طالعة من area report الى معايا لحد دلوقتى قبل ما ادخل فى PNR ودى ال area بتاعت ال design بتاعى وكمان بستخدم ال core utilization عشان احدد ال core area ... طب ايه ال core utilization؟؟ ده عبارة عن ال area الى فيها cells من ال total area of the core يعنى انا لو خليت ال core area هو نفس الى معايا دلوقتى فى ال report ما كدا كل المساحة بقت cells طب هعمل routing فين ولسه قدام هنلاقى اننا بنزود حاجات فى ال design فى CTS طب فين المساحة لكل ده وبالتالي بحدد core utilization معين وبحسب كدا ال core area من المعادلة دى:

$$\text{Utilization} = \frac{\text{Total Standard Cell Area} + \text{Macro Area}}{\text{Total Core Area}} \times 100 \%$$

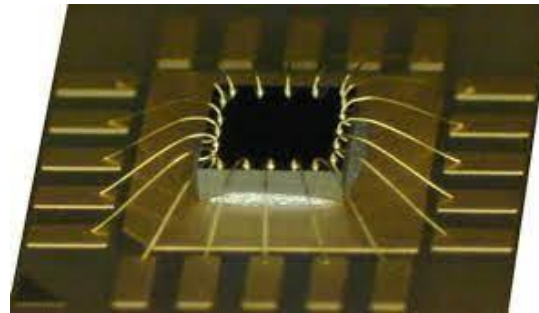
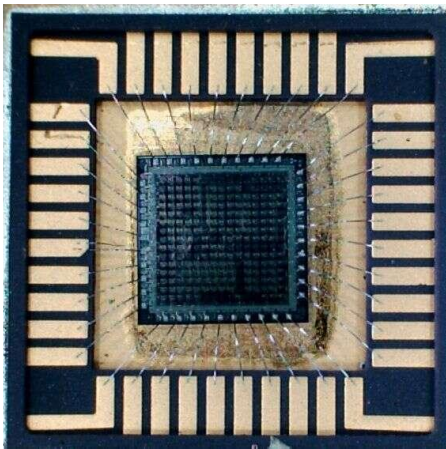
فانا معايا ال design بتاعى قد ايه بال cells & macros ومعايا ال utilization الى عايزه ده انت عارفه فتقدر تعرف ال core area وناخد بالناس ان لو عندى high utilization ده مش كويس لان كده معظم ال core عبارة عن gates وبالتالي بنقرب لمشكلة ان كله ييقى gates من غير utilization اصلا ... فبردوا كده هصعبها عليا قدام فى ال routing & CTS وحتى لو قفلت بالظبط بعد ما اخلص ال PNR ومكنش عندى violations فانا كده خليت ال core زحمة على اخره يعنى لو حبينا قدام نزود حاجة فى ال design ونعمل porting مثلا هيبقى صعب اوى فنحط utilization مناسب مش كبير اوى وكام بالظبط دى حاجة تعرفها اما تشتغل وبالخبرة

## Input/output (IO) Pads:

ال design بتاعنا الى هيكون فى ال core area هو الى فيه ال inputs and outputs ال design ككل من برا والى عاوز اوصلها للعالم الخارجى بحيث اقدر اما اصنع ال chip ادخلها ال inputs واخذ منها ال outputs وزى ما احنا عارفين ويمكن كمان شوفنا chips على الواقع وعارفين ان ال input and outputs بتكون هى ال pins الى طالعة من ال chip طيب ازاي هوصل من ال IO الى فى ال core لحد برا لل pins ... هى دى بقى ال IO pads الى بنحطها ... دى مش مجرد توصيلة عادية ... دى بتبقى دايرة يعنى كل pad هى circuit بتعمل حاجة مش مجرد بتوصل ال IO لبرا ... طب انا ليه احتاج دايرة اصلا بين ال IO بتوعى والعالم الخارجى ... لكذا سبب منهم ان مثلا بالنسبة لل power من برا هتخط مثلا 1V لكن جوا ال transistors هتشتغل على حاجة اقل من كده بكتير فمحتاج تحط level shifters تنزل القيمة دى للقيم الى محتاجها جوا ... حاجة كمان وهى ان IO pads يكون فيها ESD protection circuits طب ايه دى؟؟

ال ESD هي Electronic discharge events ودى بمعنى ان ممكن يحصل من برا حاجات تسبب ان ال voltage & current الى داخلين يكونوا عاليين اوى عن الى ال core يستحمله واحنا عارفين اننا بنشتغل عن voltages & currents وال transistors بتكون sensitive جدا فمجرد ان ايدى تلمس ال pins من برا ده بيمثل charges عالية داخله جوا فلازم protection circuit تفرغ ال charges دى بعيد عن ال core بتاعى وتضمن ان يوصله voltages & current يحافظوا عليه وميتحرقش وهى دى ال ESD protection circuit

يبقى لحد هنا عرفنا ان فيه IO pads بحطها اوصل بيها بين ال IO وال pins الى برا وعرفنا انهم circuits مش مجرد توصيلة عادية ... انا بقى باخد منهم سلكة اوصلها بال pins برا زى فى الصور تحت كدا هتلاقى الاسلاك دى طالعة من تحت معينة حوالين ال core كدا وواصله لل pins فهى مش طالعة من ال core دى IO pads بتكون حوالين ال core كله كإنها border كدا واخذ منها واصل لل pins



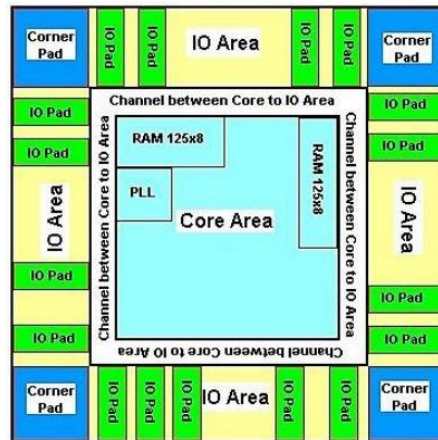
طيب ال IOs دول بتحطوا حوالين ال core كده يعملوا ring حواليه ... فايه انواعهم بقى هل هما كلهم نفس الحاجة؟ لا فى منهم انواع وممكن مش كلهم هيوصلوا بال IOs بتوع الدايرة بتاعنا فيه منهم بيتخط ومش محتاجه اصلا للدايرة بس لازم احافظ على ال continuity بتاعة ال ring دى يعنى كلها بيقى IO pads من غير فراغات بينهم وبالتالي هحط حاجات بس عشان احافظ على ال continuity ولكن مش هستخدمها فايه بقى الانواع بتوع ال IO Pads والى متجمعين فى صورة تحت:

**1. Power IOs:** دول الى بيوصلوا بال power signals الى فى الدايرة عندى الى هما VDD & VSS

**2. Signal IOs:** ودول ال inputs, outputs or inout signals الى فى ال design بتوعى

**3. Filler IOs:** دول الى بحطهم فى الفراغات عشان ال continuity

**4. Corner IOs:** دول بحطهم فى ال corners عشان ال continuity بتاعة ال IO ring بردوا بس دول مخصصين لل corners

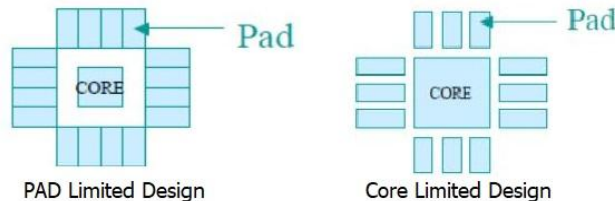


ناخد بالنسبة ان فيه مسافة بين ال core boundaries وبين ال IO ring هنا فى الصورة اسمها channel ودى بردوا هنعدها احنا قد ايه ... طيب بما ان ال die هيكون جواها ال core & IO ring فبكد ال size بتاعها هيتحدد على حسب الاتنين مع بعض ومن هنا هنعرف معلومة جديدة وهى ان عندى two types من ال design وهما core limited and pad limited:

**1. core limited:** ال core size كبير وال IOs قليلة وبالتالي ال dominant الى هيجدد ال die size هيكون هو ال core

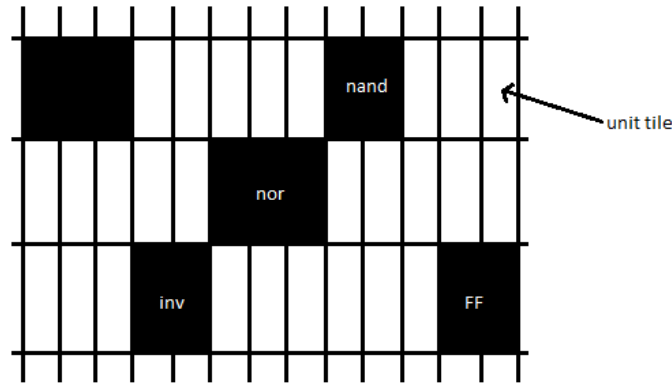
**2. pad limited:** هلاقى فيه ان ال core صغير ولكن ال IOs كتير وبالتالي ال dominant الى هيجدد ال die size هو ال IOs

لان كل ما هعوز IOs اكثر هوسع بال ring اكثر عشان تكفى

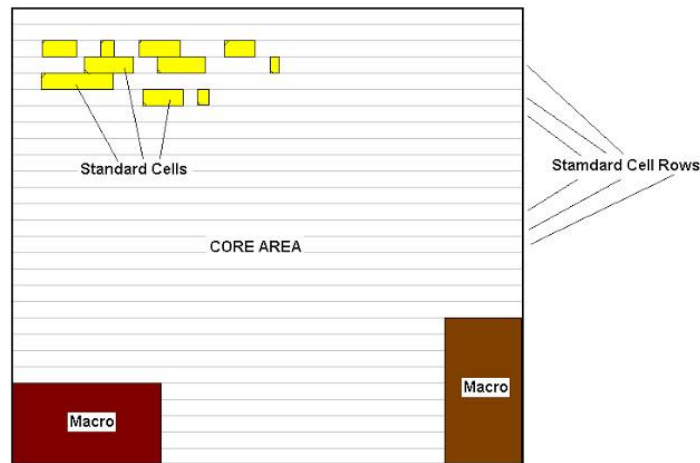


## Site rows:

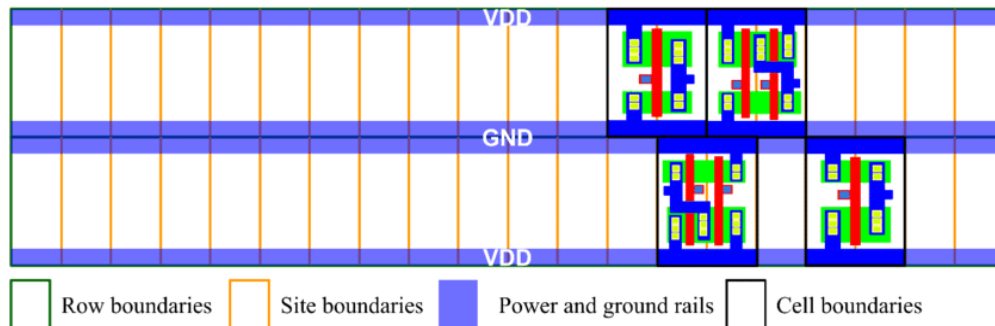
يعنى ايه site rows؟؟ خلىنا الاول نسال كام سؤال تانيين ... هل ال cells الى بستخدمها من library بيكون ال sizes بتاعتها مختلفة بشكل عشوائى ولا بت follow مثلا rule معينة؟؟ هى بت flow rule معينة اول حاجة ان كل ال cells ليهم نفس ال height وتانى حاجة ان ال width هو multiplies من اقل ال width الى هو بيتحدد فى library وال cell ... وهنطلق على ال size الى هو ال height الثابت مع min width اسم ال unit tile يعنى دى اقل ال unit عندنا وى cell sizes تانية هتكون نفس ال height و multiples من ال min width وده بيخلي ال cells تتوزع بشكل منظم واقدر ابقى عارف ايه الفراغات الى هتتسبب بين ال cells ممكن تكون قد ايه لان خلاص ال height معروف وال widths تعتبر range محدد بتحرك فيه ب step معروفة وهى ال min width



ومن هنا بقى بنقسم ال core الى horizontal rows وهما دول ال site rows بحيث ان الفرق بين كل row والثاني هو ال fixed height بناعة ال cell وبكده الماسفة دى هحط فيها ال cells بحيث ال cell من فوقها فيه row ومن تحتها فيه row وهى دى ال rows الى هتوصل لل cell ال VDD & VSS هنفهم اكثر ازاي ال VDD & VSS الى داخلين فى اول ال chip يوصلوا جوا كدا بس حتى الان خلىنا مقتنعين ان ال rows دى هى الى هتوصل VDD & VSS لل cell



وبالتالى لو حطيت cell معدولة عادى يعنى فوق VDD وتحت VSS يبقى لازم ال row الى بعده احطها مقلوبة لان هيكون ساعتها VSS هو الى فوق و VDD الى تحت .... تخيل انه ال row دى محطوطة كده واحد VDD ثم الى بعده VSS ثم VDD وهكذا

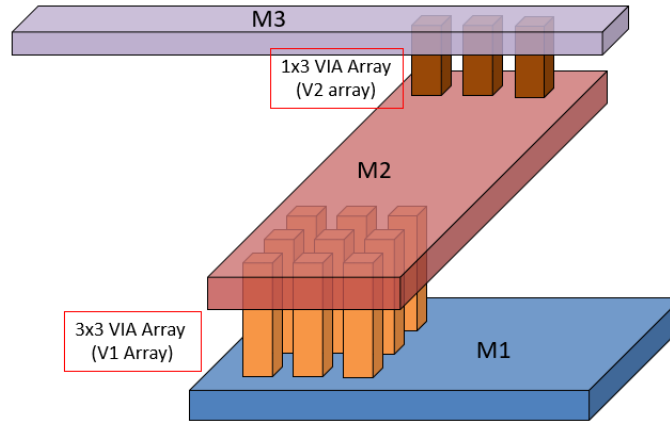


يبقى كده فى الخطوة دى احنا بنحدد ال site rows الى هى معتمدة على unit tile المتحددة من المصنع فى ال library وحتى لو عندى macro هلاقى انه واخد size بكون multiples of unit tile سواء فى ال height او ال width لان جواه طبعا cells كثير ... يبقى انا بالخطوة دى نظمت اماكن ال cells الى هتتحت فيها حتى لو مش هتملاها كلها انا عملتها الحدود المتاحة ليها عشان تتحت

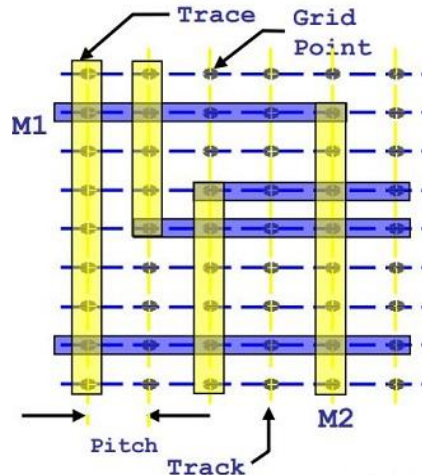
## Wire tracks:

هنا بقي هنظم اماكن ال wires الى هنتحط فيها زى ما عملت لل cells فى ال site rows ... دلوقتى احنا عارفين ان ال wires دى بنكون فى ال metal layers وبتوصل ال metal layers ببعض عن طريق vias كده يعنى ال metal layers طبقات فوق بعض ويوصل بينهم vias تكون vertical كده زى الصورة العمود ال vertical الواحد ده via ممكن احط كذا واحد واعمل array زى ما باين بحيث مثلا لو واحدة متعملتش اثناء التصنيع لأى سبب مفقده ال connection بين ال two metal layers وضمن ان التوصيلة تفضل موجودة بس ده مش موضوعنا حاليا احنا هنركز على ال metal layers

3D View of VIA and Metal connection



انا عندى metal layer يعنى مساحة كاملة لل wires تمشى فيها بس هتمشى فين فى المساحة دى؟؟ هل هحطها بشكل عشوائى كذا اى wire فى اى حته؟؟ اكيد لا طبعا زى ما نظمت اماكن لل cells هنظم بردوا اماكن لل wires بحيث قدام اما ال tool تيجى تحطها ببقى تحطها فى الاماكن المتاحة ليها فى كل metal layer والتنظيم ده هيضمنلى ان المسافة بين كل wire والتانى تكون مناسبة للى مصنع محدده عشان DRC rules الى بتتبع من المصنع فمحصلش short circuit بين wires اثناء التصنيع مثلا



كل metal layer هعمل فيها tracks بحيث ان ال wire بيمشى فى ال tracks دى ومش مسموله يمشى برا ال tracks ... ال track ده زى ما باين فى الصورة هو خط فده centre of the wire لان طبعا ال wire له width زى ما احنا شايفين سواء فى M1 الازرق او M2 الاصفر ولكن width لل wire محدود بانه لازم المسافة بين كل wire والتانى متقلش عن ال min width الى المصنع بيحدده وده بيحصل عليه check طبعا باستخدام DRC rules المبعوطة من المصنع ... توزيع ال wire tracks فى ال metal layer بيكون بحيث بين كل



track والثاني مسافة pitch المتحددة من المصنع بردوا بقيمة مقدرش اقل عنها لو عايز تزود عن min اى حاجة تمام بس متقلش عنه ولكن لو هتزدود فى ال width يبقى تزود ال spacing بين ال wires بنفس المقدار ... فيه تعريف لل pitch بردوا معروف وهو centre to centre distance between two wire وده الى قدامنا فعلا لان ال track فى نص ال wire زى ما باين فى الصورة فوق

فيه حاجة مهمة نلاحظها هنا وهى ان direction كل metal layer بيبكون perpendicular على الى قبله يعنى لو ال wires مشيت vertical فى layer يبقى الى بعدها horizontal وده واضح من اخر صورتين فوق وعشان كده اول metal layer الى هتبقى عند site rows لازم تكون horizontal زى ال site rows عشان ده المكان الى هتوصل لل cells من خلاله VDD & VSS زى ما قلنا فهمشى wires فيه اكيد عشان يوصل وال wires بتمشى فى ال metal layer ... فلو كانت ال site rows محطوة vertical يبقى اول metal layer هتكون زيها بردوا vertical

لو زدونا ال tracks اوى هيبقى عندنا routing resources كتير (يعنى اماكن اوصل فيها wires كتير) ولكن هيحصل congestion ودى حاجة مش كويسة فى اى design فلازم عدد ال tracks يضمنلى routing resources كافية وفى نفس الوقت ميحصلش congestion

## **Placement Blockages:**

Blockages يعنى قفل او منع ... فاما اقول placement blockages بيبقى اماكن انا همنع ان يتخط فيها cells يعنى اما اوصل لخطوة ال placement قدام ال tool هيبقى عندها الاماكن المتاحة لل cells الى حددناها بال site rows وتشيل منها الاماكن الى هعمل فيها هنا فى ال floor planning ال placement blockages ... ولكن ال blockage هنا له انواع ممكن امنع ان اى حاجة تتخط وممكن اسمح بحاجات معينة فخلينا نشوف انواعه:

**1. Partial Blockage:** هنا هسمح ان يتخط cells بنسبة معينة مثلا 10% بس من مساحة جزء معين فى ال core هخط فيه cells واسيب باقى المساحة فاضية

**2. Hard Blockage:** هنا ممنوع خالص فى المساحة الى هحددها دى تتخط اى cells وده بيبكون مثلا حوالين ال macros عشان مثلا ميحصلش noise بين ال cells & macro

**3. Soft Blockage:** ده بيسمح بانى احط buffers and inverters فقط لكن محطش اى cells تانية او macros وده عشان بعدين فى CTS اما اجى احط buffers & inverters زى ما هنشوف ممنعش انهم يتخطوا هنا ... بالنسبالى عادى يتخطوا هنا

## **Macros Placement:**

ال macros هى blocks جاهزة انا بستخدمها فى ال design بتاعى زى ما هى كده من غير ما اعملها انا من الاول مثلا زى memory او ممكن تكون analog block مثلا PLL وبشكل عام اقدر اقسّمهم لنوعين:

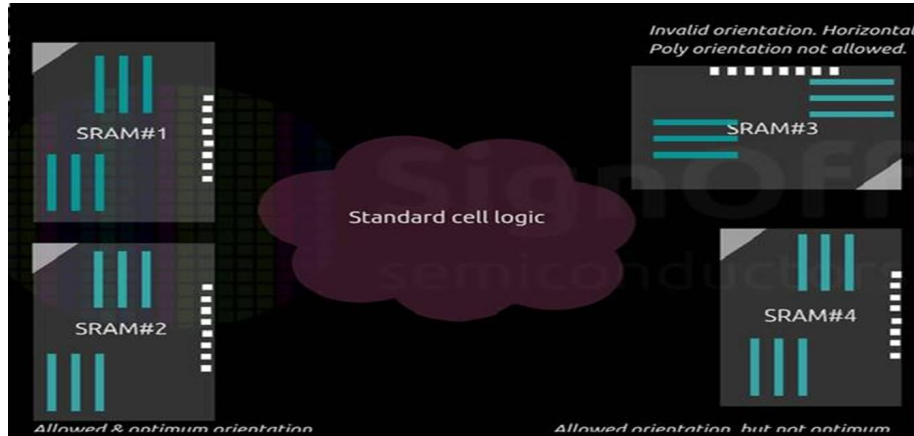
**1. Hard macro:** وده معرفش عنه غير ال timing information بتاعته بس يعنى مقدرش اشوفه من جوا ال gates ازاي ولا اعرف تفاصيل ال functionality بتاعته

**2. Soft Macro:** هنا بردوا بيبكون معايا ال timing information بس اقدر كمان اشوف هو من جوا مستخدم gates ايه بالظبط واشوف تفاصيل ال functionality بتاعته



اقدر اعمل placement لل macros سواء manually اعمله انا بنفسى وده ممكن لو عددهم قليل او اعمله automatically بال tool وده هيكون انسب لو عددهم كبير

هنحط ال macros هنا فى اماكنها فناخد بالنا ان الاحسن نحط ال macros عند ال core boundaries ونخلى ال macro pins ناحية ال cells الى هنتحط جوا ال core عشان توصل بيها متبقاش على ال borders زى فى الصورة ناحية اليمين تحت مكتوب انه مش optimum ولكن الى فوقه بقى not allowed اصلا لانك غيرت ال orientation بتاعه وده مينفعش لان ال cells محطوطة ب orientation معين جوا عشان يكون مضبوط وقت التصنيع



خلى فيه placement blockage حوالين ال macro من الاربع جوانب وال corners كمان عشان ميحصلش congestion لان اصلا ال macro ده من جواه cells كتير ولسه هوصل كل ده ... لو فيه two macros جنب بعض من الناحية الى مفيهاش pins فممكن احطهم abutment لازقين فى بعض يعنى ... لكن لو من ناحية ال bins فلازم اسيب مسافة بينهم:

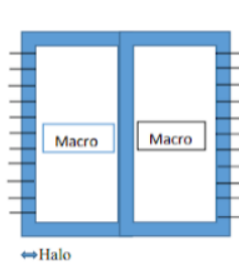


Fig. Abutted non-pin side macros with abutted halo

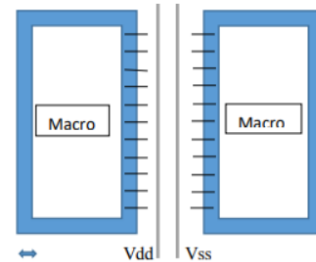


fig: pin side macros with halo

معلومة بردوا انه لو هتخط macro بنفسك manually بتستخدم حاجة اسمها fly lines عشان تحدد انسب مكان لل macro ودى عبارة عن virtual connections يعنى مش حاجة physical حقيقية وتتكون بين ال macros وبعضها او بين ال macros و IOs بحيث توريك انه مثلا المكان ده هيعمل wire طويلة اوى عشان يوصل بال pins الى هناك وهكذا زى الامثلة فى الصور تحت كده:

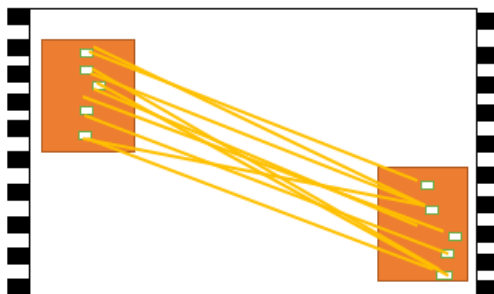


Fig a: macro to macro fly lines

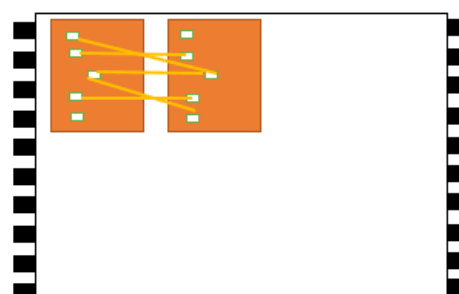


fig b: macros placed near to each other

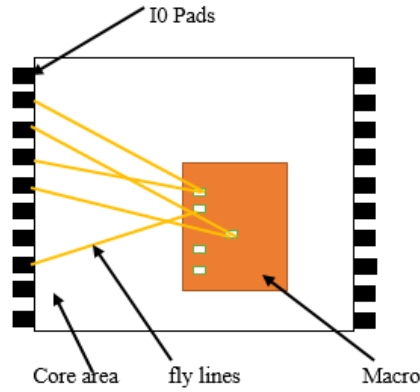


Fig a: macro to IO ports fly lines

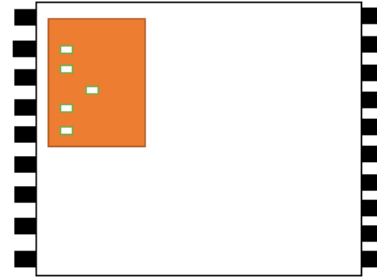


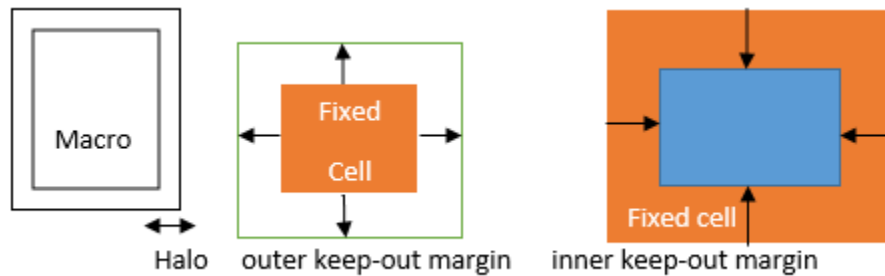
Fig b: macro placed at the core boundary

## Keep out margin (halo)

دى ممكن بدل ما اعمل blockage فى المكان الى حوالين ال macro اعمل keep out margin او بيطلق عليه halo وده معناه انى بعمل مساحة حوالين ال macro ده ميتحطش فيها حاجة زى ما باين فى الصورة تحت ... الفرق بين ده وال blockage انى لو حركت ال macro من مكانه هيتحرك معاه ال keep out margin ولكن ال blockage مش بيتحرك معاه ويبفضل فى مكانه

ال keep out margin عامة بيتعمل على حاجات غير ال macro زى مثلا لو عندى block فى ال design بتاعى معمول من ال team عادى مش جاييه macro من vendor يعنى ... وينطلق عليه hierarchical cell ... ففيه عندى نوعين من keep out margin:

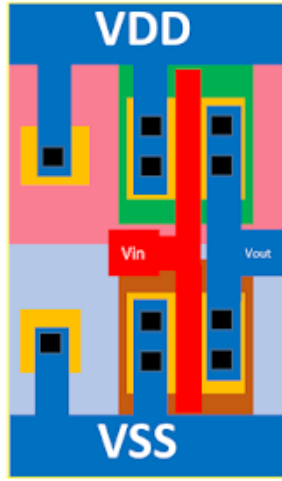
- **Outer keep out margin:** هحط مساحة من برا ال block تبقى فاضية متحطش فيها حاجة وده اقدر اعمله على macro عادى
- **Inner keep out margin:** يعنى هحط مساحة من جوا ال block تبقى فاضية متحطش فيها حاجة وده مش بعمله على macro بس اعمله على hierarchical cell



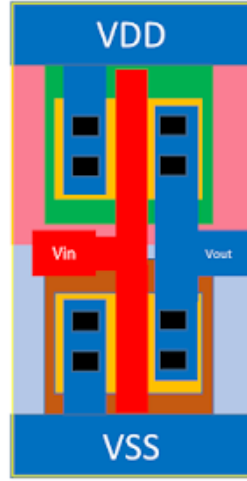
وبما انه فى الاخر معناه انى بعمل مساحة محطش فيها حاجة زى ال blockage ولكنه بيتحرك مع تحرك ال cell or macro الى معمول حواليه ... فبردوا هلاقي منه hard & soft

## Well tap cells

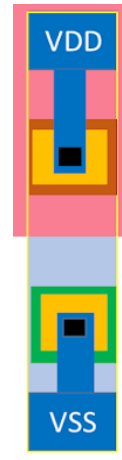
دى عبارة عن cell بيكون فيها nwell واصلته ب VDD و pwell واصلته ب VSS .. طب ايه لازمتها؟؟ ال cells الى احنا عارفينها ال CMOS فيها بيكون nwell & pwell بعمل بيهن nmos & pmos وكل واحد له source & drain & bulk الملاحظ بقى ان كل ال bulk terminal فى ال cell بتكون واصلته ب VDD فى nwell وبال VSS فى pwell فالى حصل انهم خلوا ال cells من غير bulk واطلقوا عليها tapless cells وعملوا بقى ال cell كل دورها هى توصيلة ال bulk دى وهى دى ال well tap cell:



*Traditional std cell layout*

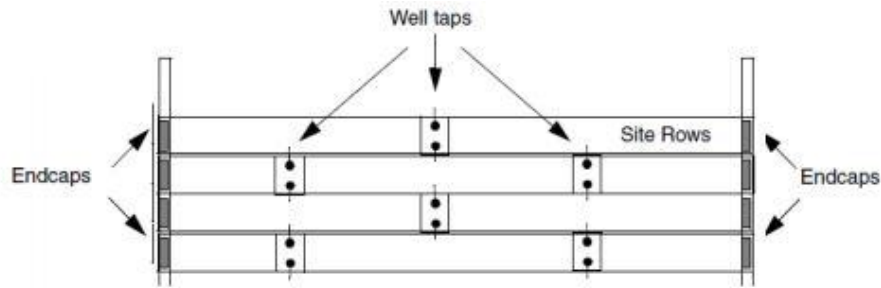


*Tapless std cell layout*



*Well tap cell*

وبنتوزع في ال site rows بحيث توصل بمجموعة من ال cells مع بعض وبنوزعها بالشكل في الصورة تحت كده ... تلاحظ ان فيه مسافات متساوية بين كل واحدة والثانية في نفس ال row ومش بيتحتوا تحت بعض في ال rows يعني فيه مسافة بين كل واحدة والى قبلها في ال row التالى او السابق ليها:



ال well tap cells دي احد انواع ال physical only cells بمعنى ان ملهاش function معينة في ال design بتاعى هي مهمة عشان ال layout بتاع ال cells يكمل صح وتشتغل مطبوط اما تتصنع

## **ICC Tool:**

عايزين بقى نشوف كل ال concepts الى شرحناها دي وقولنا بتتعمل هي فعلا بتتعمل ازاي؟؟ فاحنا عارفين من قبل كده ان دايمنا عندنا طريقتين وهما graphical user interface GUI او اعمل script والاسهل انى اعمل script ودلوقتى هنشوف اغلب ال commands الى بنحتاجها واحنا بنكتب باستخدام TCL scripting وشغالين على ICC tool Synopsys

دلوقتى ال PNR script بتعمله كله في script واحد عادى من اول ال floor planning لحد ال routing احنا مقسمين شرح كل خطوة وهنقول ال commands الخاصة بيها لكن كله في الاخر تكتبه في script واحد وبالتالي انت في بداية ال script وقبل اى حاجة محتاج انك تجهز ال input الى محتاجها في PNR والى هي اكيد بتاعة اول خطوة هعملها فخلينا الاول نشوف اول جزء في ال script الى بنجهز فيه ال inputs لل PNR ثم ندخل في commands بتاعة ال floor planning وقبل كل ده هنشوف ازاي نفتح ال ICC tool الاول اصلا

## 1. Open ICC shell:

عشان نفتح ال ICC tool اصلا هنفتح ال terminal فى الفايل الى هنشتغل فيه لل PNR ثم نكتب `icc_shell` ونسيبه يحمل لحد مايفتحها ويبقى الشكل كده:

```
IC Compiler (TM)
IC Compiler-PC (TM)
IC Compiler-XP (TM)
IC Compiler-DP (TM)

Version G-2012.06-ICC-SP2 for RHEL64 -- Aug 24, 2012

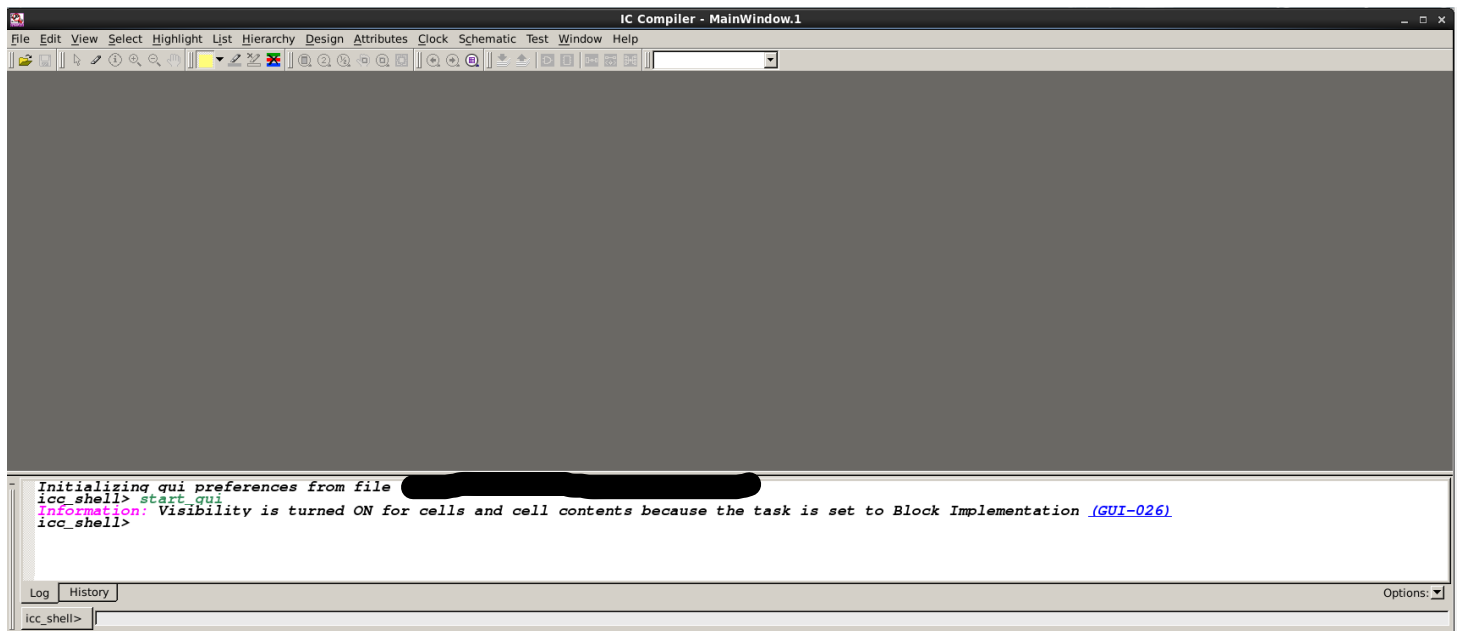
Zroute is the default router for ICC, ICC-PC and ICC-DP in IC Compiler.
Classic router will continue to be fully supported.

Copyright (c) 1988-2012 Synopsys, Inc.

This software and the associated documentation are confidential and
proprietary to Synopsys, Inc. Your use or disclosure of this software
is subject to the terms and conditions of a written license agreement
between you, or your company, and Synopsys, Inc.

Initializing...
Initializing gui preferences from file "C:\Program Files\Synopsys\IC Compiler\gui_preferences.tcl"
icc_shell>
```

احنا كده جوا ال ICC بس عشان اقدر اشوف قدامى تأثير ال commands الى بعملها واشوف ال design بتاعى وانا ماشى عليه ال PNR flow هحتاج افتح ال Graphical user interface GUI يعنى افتح ال tool نفسها اشوفها قدامى فيكتب فى ال terminal ال command الى هيفتحها وهو `start_gui` فهلاقيها فتحت:



وبعد كده إما انى اعمل run لل script بس line by line اكتبه إما فى ال terminal او ال command window الى تحت فى ال GUI هلاقى المربع الابيض ده كانه ال terminal بتاعى وكتب ال command فى المربع الى تحته الى هو مكتوب جمبه `icc_shell`... او انى ارن ال script كله فيكتب `source` ثم اسم ال script ومنسناش نخط ال path لو احنا مش فاتحين ال terminal فى نفس ال folder الى فيه ال script بتاعنا فيكون شكل ال command كده: `source path/script_name.tcl`

## 2. Commands used for preparing PNR inputs:

### - Importing the libraries

اول حاجة انى اجهز ال libraries زى ما شرحنا فى logic synthesis بحط ال search path الى هو المكان الى فيه ال libraries  
ثم ال link library & target library ممكن تفكرهم بالتفصيل من ال logic synthesis file

#### Commands:

```
lappend search_path ".....put the path....."  
set target_library [list lib1.db lib2.db]  
set link_library [list * lib1.db lib2.db Macrolib.db]
```

=====

### - Create MW library

بعد كده هنعمل MW library ولكن لو انا بعمل run كذا مرة فهبقي عملت واحدة مثلا فى اول مرة ولكن انا عايز امسح وابدأ من الاول فعشان  
امسح ال MW library الى موجودة هستخدم command معين وهو:

#### Command: sh rm -rf MW\_lib\_name

هنا rm دى معناها remove ثم rf دى يعنى هيمسح كل حاجات ال lib الى هحط اسمها دلوقتى ثم اسم ال lib الى عملتها ... وجود sh فى  
الاول لان ده command ملوش علاقة ب TCL ده انا عايز امسح حاجة فى مكان معين فى VM يعنى فالمفروض اكتبه برا فى ال shell  
بمعنى انه انا دلوقتى اما بفتح ال terminal عادى ببقى كده بكتب فى ال shell فبكتب commands زى مثلا ls يجلبى list بال files  
الموجودة فى المكان الى واقف فيه او مثلا cd يعنى change the directory وزى rm الى عندنا هنا الى هى remove ... بس انا اول  
ما بفتح icc tool ببقى لازم احط commands الى تفهمها ال tool يعنى ls, cd, rm, .... كل ده مش لل icc فيما ان ال script الى بكتبه  
ده لل icc وانا عايز اعمل حاجة تانية برا ال tool خالص وهى انى امسح file موجود فى مكان معين عندى ب rm command فبكتب sh  
وهو كده بي فهم ان ده command لل shell

=====

#### Command:

```
create_mw_lib ./.name -technology directory/tech_file_name.tf -mw_reference_library directory/  
-hier_separator {/} -bus_naming_style {[%d]} -open
```

ده ال command الى هعمل بيه ال MW library فخلينا نبص على كل option فيه لوحده كده ونفسره:

- **name/.** دى بعد اسم ال command على طول بقول بيها انه هعمل MW library فى ال directory الى انا فيه دلوقتى باستخدام  
name/ وهسميها كذا الى هو ال name
- **technology:** هنا انا عايز احط ال technology file الى كنا اتكلمنا عليه فى البداية وعارفين ان extension بتاعه هو .tf فالاول  
هحط ال directory الى هوموجود فيه ثم هحط اسم الفايل لان ممكن الاقى اكثر من واحد كل واحد لعدد metal layers مختلف مثلا  
الاقى واحد 6m وواحد 10m وهكذا يعنى فبحط اسم الى عايزه بعد ال directory زى ما باين فوق كذا

- **mw\_reference\_library**: دلوقتي ال cells الی هتتحفظ فی ال library والی بتمثل screenshot ل step معينة هقف عندها .. ازای هربطها بال cells الی فی ال library عندی یعنی مثلا MW cell الی هی واحدة صورة دی لخطوة ما وليكن قبل ما ابدأ خالص هتأخذ صورة لل design بتاعی لحد ما قبل PNR فده اصلا فيه cells من ال library هی محتاجة تعرف كل واحدة هی ایه زی ما خدت ال technology file تاخذ كمان ال cells المتوفرة عندی واسمائها وكده فهنا بحط ال path الی هتلاقى فيه الكلام ده والحاجات الی بتحتاجها مش بتكون فی folder واحد ... انت هتلاقى ال path ده عندك فيه files وحاجات بالشكل ده:



واهم حاجة هتميزلك ال path ده هما CEL & FRAM folders لو فتحتهم هتلاقى جواهم نفس ال cells بس الفرق ان كل واحد ب view مختلف یعنی CEL ده جواه ال full layout view ولكن FRAM جواه ال abstract view وشوفنا الفرق بينهم بالتفصيل فی technology libraries file

- **hier\_separator**: ده بس بتحدد ال character الی ال library هتستخدمه عشان تفصل بين ال cell name وهی بتكتب herichary levels

- **bus\_naming\_style**: هنا بتقول ال buses الموجوده هتسميها ازای فمثلا هنا d% معناها انك هتسميها بالارقام مثلا 1,2,3,....

- **open**: بتقوله يفتح بقى ال MW library بعد ما يعملها عشان تبدأ تشتغل وتحفظ فيها ال MW cells

=====

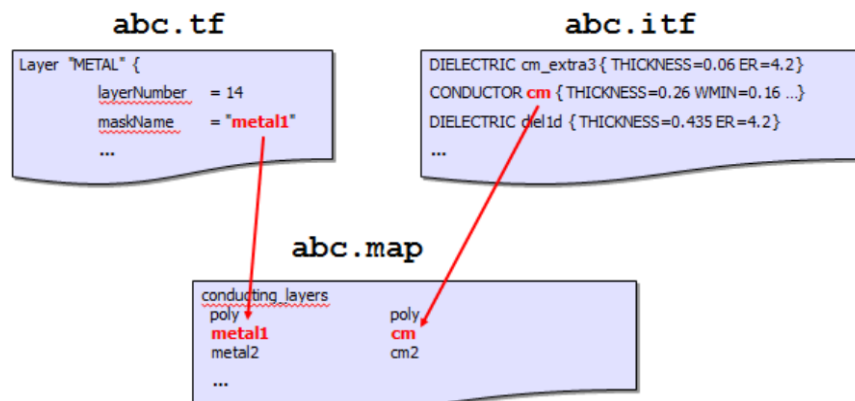
## - TLU+ files:

### Command:

```
set_tlu_plus_files -max_tluplus file_path/file_name.tlup -min_tluplus file_path/file_name.tlup \
-tech2itf_map file_path/file_name.map
```

باستخدام ال command ده هدخل ال +TLU الی اتلكمنا عنه ال min & max بس بحط ال path الاول ثم اسم الفايل ... وادخل ال mapping file بنفس الطريقة ... ایه ال **mapping file**؟؟

ده بييجى من ال vendor وهو عبارة عن انه بياخد كل layers & vias فی technology file يشوف المقابل ليها فی ITF file الی اتلكمنا عنه فوق مع +TLU یعنی زى فی الصورة كده خد 1 metal بال specs الی فيها وشاف الی زيها فی itf لاقها conductor اسمه cm:



### - Import Design File (gate level netlist):

#### Command:

```
import_designs path/gate_level_netlist_name.v -format verilog -top top_module_name\  
-cel design_name
```

اهم input بقى هو ال design نفسه الى خرج من ال synthesis او من DFT لو هتعمله وفى جميع الاحوال هيكون عبارة عن gate level netlist فأول حاجة بعد ال command على طول هتخط ال path الى فيه ال netlist دى ثم اسمها زى ما باين ومننشاش بنكتب اسمها بال extension الى هو .v فى حالة اننا شغالين verilog ... ثم ال format الى هو بتاع ال netlist وزى ما قلنا احنا شغالين verilog بعد كده top مقصود بيها اسم ال top module فى ال design بتاعك ثم cel دى بتخط فيها اسم لل design الى عملته import يعنى بتسميه اى اسم تحبه بحيث تستخدمه قدام اما تحتاج تشير لل design ده

=====

### - Import Design Constraints:

كده احنا ناقصنا input واحد من الى اتكلمنا عليهم فى بداية ال document وهو design constraints الى استخدمناهم فى ال synthesis وهنا ممكن تكتبهم تانى بس توفيراً للمساحة وللتنظيم ممكن تحطهم فى file لوحدهم ال constraints فقط زى input delay, output delay, .... وتعمله run فى PNR script عن طريق استخدام source ثم بعدها تحط ال path الى فيه ال file ثم اسم ال script الى هيعمله run مثلاً cons.tcl:

Command: source path/constraints\_file\_name.tcl

=====

### - Save MW Cell:

كده انا خلصنا كل ال inputs وجهازناها بس ناقص حاجة مهمة وهى انى اعمل save بقى لل design بتاعى بالى واصله لحد دلوقتى فى MW library عشان اما احب ارجعله فكده هو هيحفظ ال design الى عملته import ده وليكن باسم my design فى صورة MW cell اسمها cell\_name وممكن اخليه حاجة واضحة اكثر مثلاً ده كده ال imported design فاسميه مثلاً cell\_imported وتتضاف لل MW library

Command: save\_mw\_cel -as cell\_name



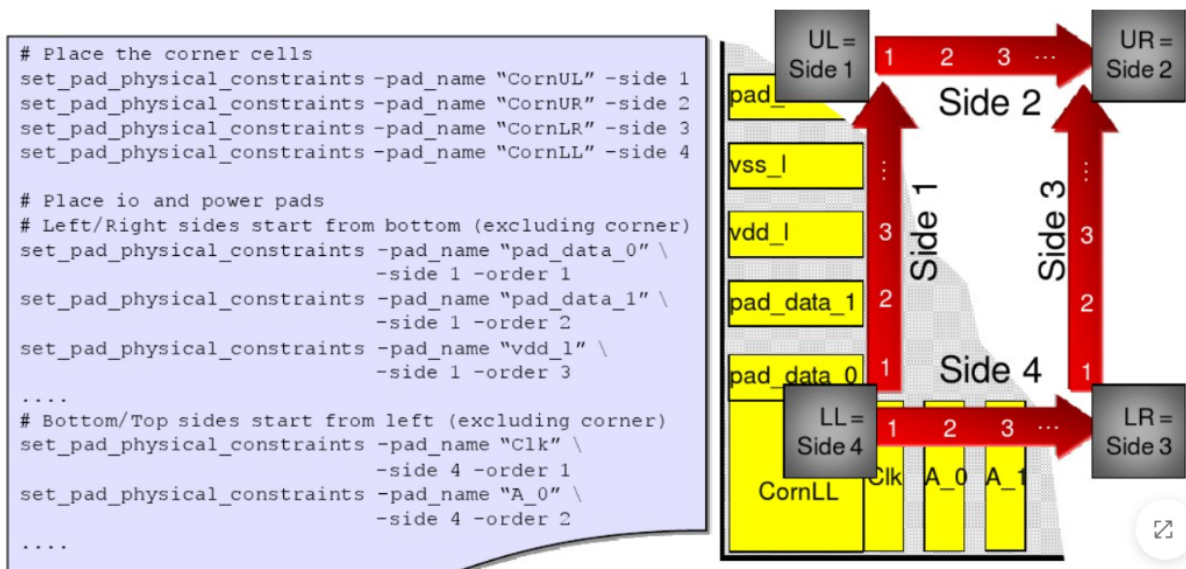
### 3. Floorplanning Commands:

#### - IO Pads:

##### Command:

`set_pad_physical_constraints -pad_name <name> -side <put number> -order <put number>`

ال command ده هو الى بحط بيه ال IO Pads برتبههم حوالين ال chip فدلوقتى بالنسبة لل rectangular chip shape عندى 4 sides مترتين بحيث ان side1>left side2>top side3>right side4>bottom فانا بختار هحطها فى انى side ثم عندنا ال order وده يعنى ترتيب ال pads فى ال side فعندنا ال left & top بيكون الترتيب بيمشى clockwise ولكن right & bottom بيكون اولوية الترتيب بتمشى anticlockwise ... فمثلا فى المثال الى تحت ده هتلاقى التوزيع زى ما شرحنا وهنا فى الاول هو حط ال corner IOs الى اتكلمنا عليها ثم بدأ يوزع الباقي فهتلاقى order1 فى side1 يعنى فى ال left هو اول ال left side من تحت لان زى ما قلنا بيمشى هنا فى الترتيب clockwise على عكس فى side4 الى هو ال bottom هتلاقى ان order1 هو اول ال bottom side من الشمال لان زى ما قلنا ده بيمشى anticlockwise ممكن تسأل نفسك طب فين ال filler IO pads بس استنى هيجوا بعد شوية فى command منفصل ... احنا هنا حطينا الحاجات المعروف اماكنها زى ال corners وحطيت ال IOs بتاعتي الاساسية لكن مش هقعد احط فى كل مكان فاضى filler Pad يعنى اكيد ليها طريقة اسهل فى مرة واحدة املى كل الفراغات الى هتبقى عندى فاستنى شوية وهنشوفها



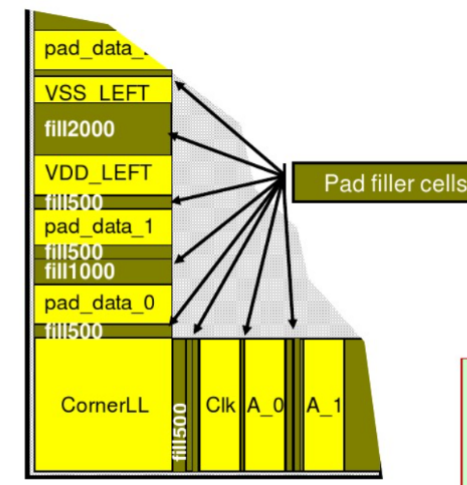
لو انا معملتش order هيبقى ال default هو 0 وده اول حاجة خالص فى الترتيب قبل الواحد عشان كده معمول لل corners لانهم يعتبروا ال 0 من كل ناحية عندى

=====

Command: `insert_pad_filler -cell <cell name in library>`

ده الى هيملى الفراغات ب filler cells لو انا محطتش اسماء cells معينة هو هيستعمل ال default الى فى ال library لل case دى ولو حطيت هيستعملهم ويدور فيهم على المناسب للمكان الفاضى ويحطها ... فاحنا بنحط الاسماء من الكبير فى الحجم للصغير من الشمال لليمين زى ما عمل تحت فى الصورة كده بحيث نقلل فى عدد ال cells الى هiestخدمها ياخد الكبيرة مثلا الى هتكفى المكان بدل ما ياخد اثنين صغيرين فكأنى برتبلة الى هيدور فيه بشكل احسن

```
insert_pad_filler -cell "fill5000 fill2000 fill1000 ... "
```



### - Power/GND Nets:

**Command:** `derive_pg_connection -power_net VDD -power_pin VDD -create_port top`  
`derive_pg_connection -ground_net VSS -ground_pin VSS -create_port top`

ال command بي عمل logical connection بين ال power pins الى في كل cell وبين ال power nets في ال design ...  
 دلوقتى اى cell او اى macro بيكون ليهم power pins ... دى مش موجودة في rtl وانت بتكتبه بس هي موجودة في ال cell فانا محتاج  
 احدد بردوا ال pins دى رايحة على فين بس مش بوصل physically هنا ... انا بس مجرد بقول ان ال pins دى هتوصل على ال nets الى  
 اسمهم كذا دول يعنى بحدد المكان الى هتروح عليه مش اكثر ونفس الاسماء دى هي الى هعمل بيها ال power planning زى ما هنتشوف  
 قدام ... ولو لاقيت ال command متكرر مثلا في script فده ممكن عادى لاني لو حظيت cells جديدة مثلا هعوز ا قوله بردوا اعمل  
 logical connection تاني

### - Metal layers routing directions

**Command:** `set_preferred_routing_direction -layers {Layers name} -direction vertical or horizontal`

هنا بحدد اني layers هتكون vertical واني هتكون horizontal ونفكر اننا بنعملهم عكس بعض يعنى لو عملت واحدة vertical ببقى الى  
 قبلها والى بعدها horizontal فمثلا اخلى كل ال odd layers يبقوا horizontal وكل ال even layers يبقوا vertical:

### EX:

- `set_preferred_routing_direction -layers {M2 M4 M6 ...} -direction vertical`
- `set_preferred_routing_direction -layers {M1 M3 M5 ...} -direction horizontal`

## - Create Floorplan

### Command:

```
create_floorplan -core_utilization 0.3 -flip_first_row -start_first_row -left_io2core 10\
-bottom_io2core 10 -right_io2core 10 -top_io2core 10
```

هنا بقی العمل floorplanning خلاص یعنی هحدد المساحة الی هیط فیها ال cells واضبطها فمثلا اول حاجة core\_utilization وهنا بحدده ال utilization یعنی مساحة ال core قد ایه منها خده وحط فیہ cells زی ما قلنا مش هملاه كله cells وهنا بحط رقم صغیر لانه انا لسه قدام عمل routing & CTS ومحتاج مساحة كافیه لكل الی هضیفه قدام وکمان ابقى قافل ال chip فی النهایة ب utilization کویس برودوا متبقاش کل حاجة لازقة فی بعض ومفیش ای مساحة احط حاجة جدیدة زی ما اتکلمنا فی الشرح ... بعد کده flip\_first\_row وهنا بقوله ی flip ال orintation بتاع اول row ... زی ما احنا عارفين من الشرح فوق ان ال cells الی تحت بعض بتقی واحدة معدولة والتانیة مقلوبة عشان یتشارکوا فی نفس ال VDD ... بعد کده start\_first\_row یبدأ یحطهم من اول row ... وبعدين عندی 4 options کلهم io2core ودول بیعبروا عن المسافة الی هسیبها بین کل side وال IOs الی فی نفس ال side یعنی بین حدود ال core و ال IO ring بیکون فیہ مسافة زی ما قلنا فی شرح ال IO pads بحددها من ال 4 options دول فمثلا left\_iocore هیط المسافة الی فی ال left side وهكذا

=====

## - Keep out margins

### Command:

```
create_keepout_margin -type hard/soft -outer/-inner {left bottom right top} -all_macros or cell
name or macro name
```

هنا بقی هحط keep out margin هختار hard ولا soft ثم بعد ما اختار outer or inner هحدد الابعاد بتاعته ثم اسم ال hierarchical cell او ال macro اولو عایز احط علی کل ال macros اکتبله -all\_macros

### EX:

- create\_keepout\_margin -outer {10 10 10 10} my\_macro
- create\_keepout\_margin -inner {10 10 10 10} my\_hcell
- create\_keepout\_margin -type hard -outer {10 10 10 10} -all\_macros

=====

## - Placement blockages

### Command:

```
create_placement_blockages -boundary {left_lower_x left_lower_y right_upper_x
right_upper_y} -name blockage_name -type hard/soft
```

```
create_placement_blockages -boundary {left_lower_x left_lower_y right_upper_x
right_upper_y} -name blockage_name -type partial -blocked_percentage number
```

هعمل ال placement blockages بس فى ال partial بيبقى عندى option زيادة عشان ال blockage percentage وممكن احط name لل blockage لان ده هيساعدنى قدام لو حببت امسحه مثلا .. بالنسبة لل boundary فانا بحدد مكانه ب two corners من الاربعة بتوعه الى هما left lower يعنى الى تحت على الشمال و right upper الى فوق على اليمين فاول two coordinates هما x,y بتوع ال left lower corner وتانى two coordinates هما x,y بتوع ال right upper corner

### EX:

- `create_placement_blockages -boundary {{10 20} {100 200}} -name PB1 -type hard/soft`
- `create_placement_blockages -boundary {10 20 100 200} -type partial \`  
`-blocked_percentage 40`

=====

**Command:** `remove_placement_blockages -all` or blockage name

ده استخدمه عشان لو عايز امسح blockage اما اكتب اسمه الى حددته وانا بعمله او اديله option ال -all ده فهيمسح كل ال placement blockages الى عملتها لحد دلوقتى

=====

### - Macro Placement:

**Command:** `create_fp_placement`

ده اختصار ل create floor plan placement وهنا بيبدأ يحط ال macros فى امكانها automatically من ال tool

=====

### - Well tap cell insertion:

**Command:**

`add_tap_cell_array -ignore_soft_blockage true/false -master_cell_name tap_cell_name \`  
`-distance tap_pitch -connect_power_name VDD -connect_ground_name VSS`  
`-respect_keepout_pattern stagger_every_other_row \ -tap_cell_idenfifier WELLTAP`

هنحط ال well tap cells باستخدام ال command ده وزى ما شوفنا هي بتتحط ازاى هنلاقى انها بتعمل زى array كل cell محطوة فى مكان محدد وهكذا فكأنى هنا بعمل ال well tap cell array ... فيه options كتير فى ال command ده خلينا نشوفهم:

- `ignore_soft_blockage`: يعنى هل ال cells هتاخد فى اعتبارها ال soft blockages المعمولة فاحط true ولا هنتعامل معاها كأنها مش موجودة فاحط false

- `master_cell_name`: هحط اسم ال cell المستخدمة ك well tap cell

- `distance`: المسافة بين ال centers لاتنين cells جمب بعض وبنطلق عليها pitch: distance from centre to centre الى شوفناها اما اتكلمنا عن شرح ال well tap cells وان فيه مسافة مسافة ثابتة بين كل اتنين جمب بعض

- `connect_power_name`: هنا بحط اسم ال power net الى هتوصل عليها والى اغلب الوقت بنسميها VDD

- **connect\_ground\_name**: هنا يحط اسم ال ground net الى هتوصل عليها والى اغلب الوقت بنسميها VSS

- **respect\_keepout\_pattern**: ده معناه انه ال cells وهى بتتخط بتاخذ بالها انها متتحتش فى الاماكن الى غير مسموح انى احط فيها cells زى ال keep out margin مثلا ... فليها option اسمه stagger\_every\_other\_row وده معناه انه يبدأ يحط ال well tap cells فى ال rows ما عدا ال rows الى فيها keep out patterns مثلا لو عندى فى اول وتالت row فهو يحط ال well tap cells فى تانى و رابع row وهكذا يعنى فى باقى ال rows الى فيه keep out patterns محطش فيه well tap cell ... فيه option تانى هو انى اعمل نفس الكلام بس ك columns لان ال chip فى الاخر هى عبارة عن rows & columns فممكن ابص عليها ك columns هنا واشوف ال columns الى فيها keep out patterns ومحطش فيها welltap cells واحط فلا الباقي باستخدام stagger\_every\_other\_column

- **tap\_cell\_identifier**: اسم مطلقه على ال well tap cell array الى هحطها دى

### EX:

- `add_tap_cell_array -ignore_soft_blockage true -master_cell_name welltap -distance 4\  
-connect_power_name VDD -connect_ground_name VSS\  
-respect_keepout_pattern stagger_every_other_row -tap_cell_identifier WELLTAP`

=====

### 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/1aLwCLZj0YG9KJhIn1D60\\_nWM7p-L3q9a?hl=ar](https://drive.google.com/drive/folders/1aLwCLZj0YG9KJhIn1D60_nWM7p-L3q9a?hl=ar)

=====