

Chip Finishing (V1)

Written by: Fatma Ali

Content:

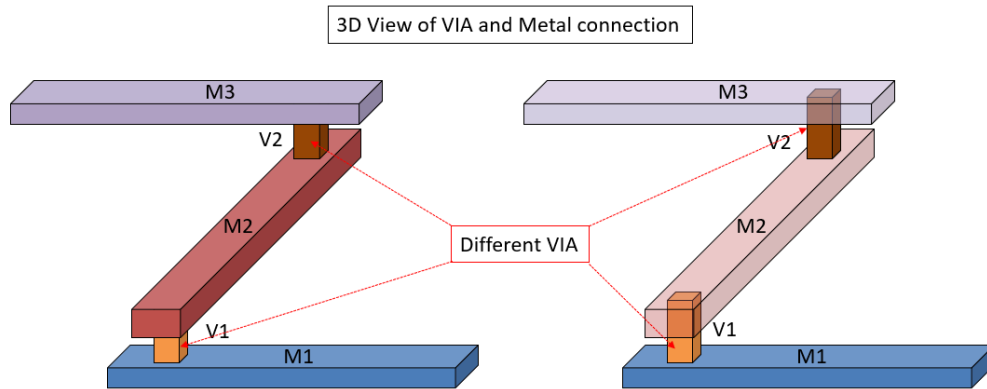
- Introduction
- Redundant Vias
- Physical only Cells
 - Filler Cells
 - Decap Cells
 - End Cap/Boundary Cells
 - Well Tap Cells
 - Tie high and Tie low Cells
- Engineering Change Order
 - Non-Freeze ECO
 - Freeze ECO
 - Spare Cells
- Antenna Effect
- Metal Fill
- ICC Tool Commands

Introduction:

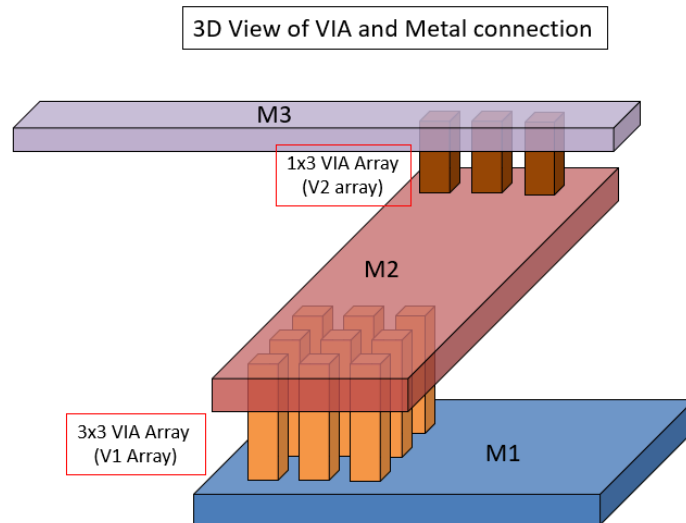
حتى الآن احنا خلصنا كل ال main stages فى ال ASIC Flow من اول ال logic synthesis لحد اخر مرحلة فى PnR وهى ال routing ولكن لسه ناقص شوية حاجات محتاج اضيفها لل design بتاعى تعتبر finishing اخير لل design وده بيكون بهدف حماية ال circuit من اى unexpected conditions تحصل فى التصنيع او حتى بعد كده وقت استخدام ال chip بعد التصنيع ... فهتلاحظ ان ال file ده هو عبارة عن مجموعة من المشاكل الى ممكن تحصل لل circuit بتاعى اثناء التصنيع او بعده واحنا بنستخدمها ... وبنقول ايه الى نضيفه لل circuit قبل ما نودبها نتصنع بحيث نحميها من كل المشاكل دى.

هنلاقى هنا بعض المشاكل الحلول بتاعتها كنا بنعملها فى stages قبل كده عادى ... احنا هنجمع كل المشاكل الى ممكن تحصل من اى عوامل خارجية سواء فى التصنيع او بعده ونشوف حلها نعمل ايه من دلوقتى عشان نتجنب حدوثها ونشوف بقى الحل ده كنا بنعمله فى انى خطوة فى ال ASIC flow

Redundant Vias:

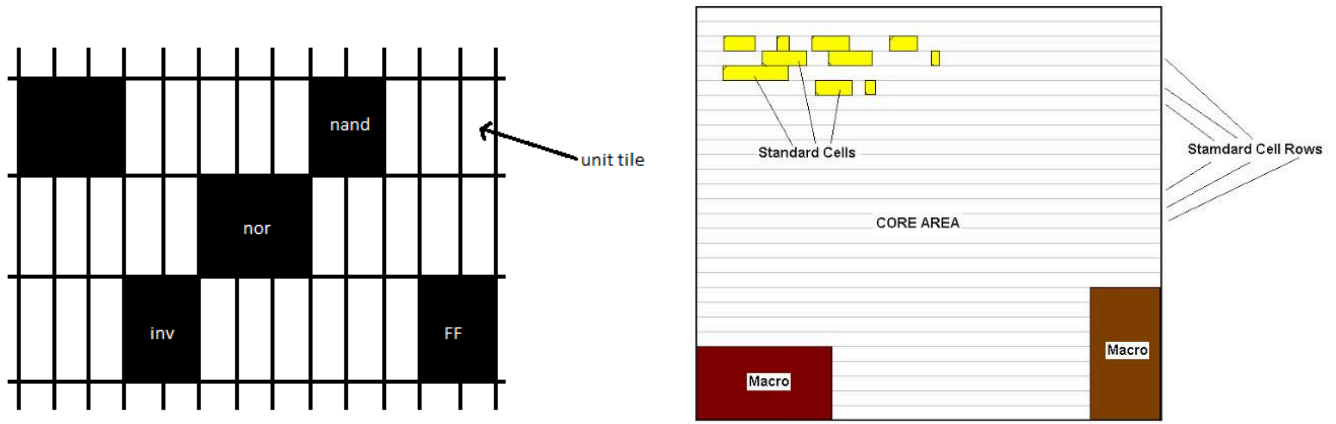


ال via زى ما احنا عارفين من قبل كده انها ال metal الى بيوصل بين two metal layers زى ما واضح من الصورة فوق ... فأحد المشاكل الى بتحصل اثناء التصنيع ممكن تسبب circuit damage هي ان ال via متصنعش بالغلط فلو انا معتمد على via واحدة فكده ال connection كلها باظت يعنى ال circuit باظت وبالتالي عشان نتجنب حاجة زى دى بنحط اكثر من via عشان كده بنسميهم redundant vias لأن هما زيادة وهتظهر أهميتهم لو واحدة منهم متصنعش ... بس ناخذ بالنا ان ال redundant vias دى هتأثر على ال timing لأن زيادة عددهم هنكبر المساحة الى بيمشى فيها ال current وبالتالي delay اقل فناخد بالنا من تأثير ده على ال STA ... لما نحط اكثر من via جمب بعض هنلاحظ انهم بقوا زى array فنطلق عليهم array of vias زى ما احنا شايفين فى الصورة تحت:



Physical only Cells

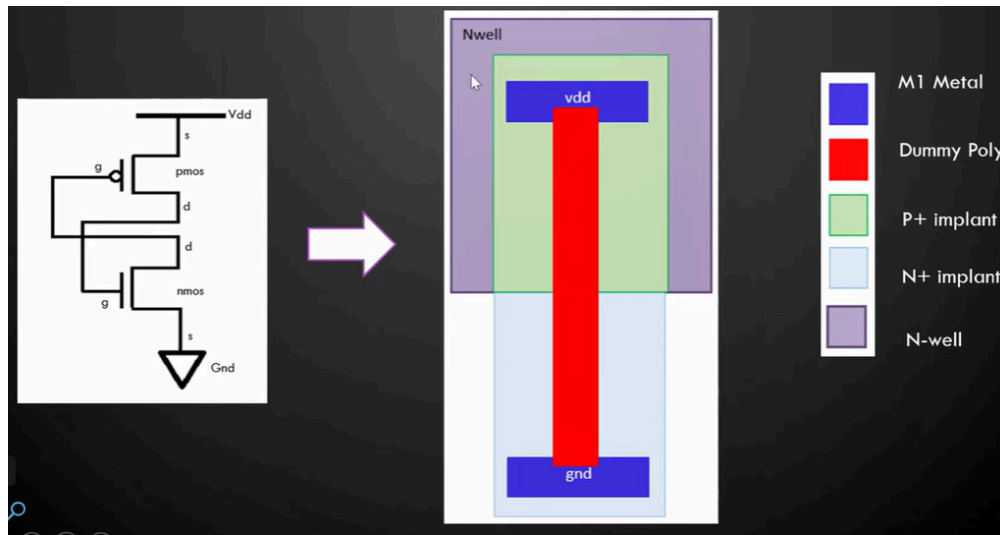
دول عبارة عن cells ملهاش علاقة بال design يعنى مش بتتخط عشان تحقق ال functionality ... دى هدفها physically بمعنى اننا نعمل layout كامل مطبوع مفيهوش اى physical violations و cells تانية تحمى ال circuit بتاعتى من المشاكل الى ممكن تحصلها بسبب ال environment الى هتشتغل فيها ... اى cell هتتخطها (سواء ال functionality cells او ال physical only cells) لازم هتكون فى الاماكن المتاحة ليها فى ال core area فخلينا نفكر الكلام الى قلناه عن الاماكن دى فى ال floorplanning



نفكر من ال floor planning ان ال core area المتاح نحط فيها ال cells قسمنها الى مسافات متساوية بال site rows بحيث ان المسافة بين كل row والثاني تكون ال cell height المتحدد من المصنع وقلنا ان ال row بيتكون من unit tiles بحيث ان ال unit tile دى اقل size ممكن لل cell وى sizes اكبر بتكون multiples of unit tile وشرحنا الجزء ده بالتفصيل فى ال floorplanning ثم فى ال placement حطينا ال design cells فى الاماكن المتاحة ليها فى ال site rows

احنا هنا هنجمع كل ال physical only cells بغض النظر انت المفروض تحطها فى مرحلة ال chip finishing ولا فى مرحلة سابقة وهنقول كل واحدة بنتحط امتى وهتلاقى cells اتكلما عنها قبل كده بس الهدف هو تجميعهم تحت title واحد دلوقتى

1. Filler Cells:



بعد ما بنحط كل ال design cells ... لو بصينا على ال core area دلوقتى هنلاقى ان فيه مسافات فاضية (سواء صغيرة قد ال unit tile او multiples of unit tiles) متحطش فيها cells وده مينفعش فلزام نحط cells هنا وهنشوف ليه مينفعش بس هى كدا مش محطوطة عشان ال functionality وبالتالي هنلاقى انها مجرد ال two transistors من غير input او output

أول أهمية لل filler cells هنتكلم عنها هى ان ال cells زى ما احنا عارفين فيها pwell & nwell ودول بيتعملوا لل chip كلها ب masks فمحتاج يكون فيه continuity لل nwell & pwell يعنى متجيش تحط cell ثم جمبها فراغ ثم cell تانية لان كده بين nwell الاولى والثانية فيه فراغ وهكذا لل pwell وده غلط لأن هيكوّن فيه صعوبة فى التصنيع وممكن تحتاج كذا mask وتكلفة ال mask عالية جدا وبالتالي محتاج تحافظ على ال nwell & pwell continuity وده عن طريق ال filler cells

الفراغ بين ال cells بيعمل مشكلة تانية وهى discontinuity of the cells power rails ... بمعنى انه ال power rails الى عملناها فى ال power planning هتبقى واصلة بايه فى الفراغ ده؟؟ مش واصلة بحاجة وده ممكن يسبب مشاكل زى voltage drop على او مشاكل تانية فى ال power network الى احنا اصلا عاملينها عشان نوصل ال power بشكل مضبوط على كل ال chip

مشكلة كمان هتسببها الفراغات بين ال cells لو سببناها هى انها هتقلل ال yield ... ال yield يعنى عدد ال chips الى اتصنعت مضبوط من غير manufacturing faults يكون كبير بالنسبة لعدد كل ال chips الى اتصنعت فكل مكان ال yield على يبقى احسن وعشان ده يحصل محتاج يكون فيه cells continuity يعنى مفيش فراغات بين ال cells وده الى بتعمله ال filler cells

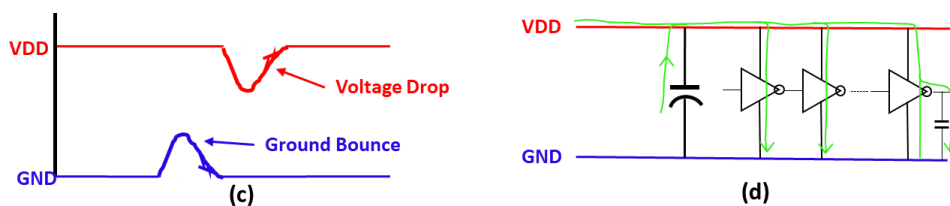
وبما انها cell بردوا فليها sizes مختلفة ال min هو ال unit tile ثم ال sizes الاكبر multiples of unit tiles زى اى cell بحيث انها تملئ اى فراغ الى هيكون لو نفس ال expected values of the sizes لان core chip كله معمول كده ... وكده نكون عرفنا ايه هى ال filler cells ودورها الرئيسى الى ظهر فى 3 حاجات ذكرناهم هما nwell & pwell continuity و power rails continuity و better yield

هتلاحظ ان منطقى منحطش filler cells الا دلوقتى فى الاخر بحيث تكون حطيت ال design cells وكمان buffers & inverters فى ال CTS stage عشان تسبب المكان متاح لل cells المهمة ثم تقفل الاماكن الفاضية مش العكس اكيد

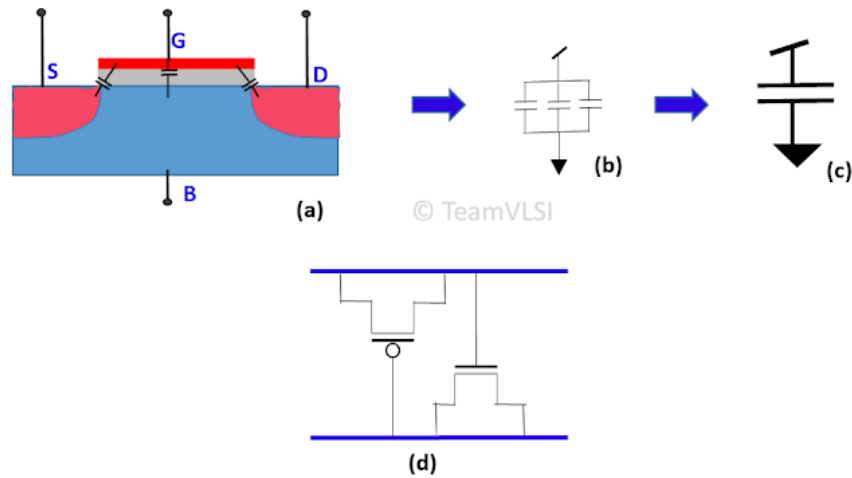
=====

2. Decap Cells:

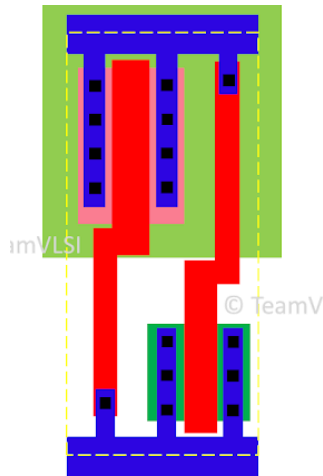
بنحطها بردوا فى الفراغات بين ال cells ولكن دى بتضيف حاجة مهمة مميزة عن ال filler cells وهى انها عبارة عن capacitor يعنى charge storage device وبالتالي لو حصل فى اى وقت voltage drop فهى تقدر توفر ال voltage ده لل cells وحتى لل Vss لو حصل اى bounce يعنى زى peak فجأة بردوا هتوفر ال Vss المضبوط ليهم فتساعد فى انها تخلص ال power delivery network تكون more robust



ازاى بقى بتتعمل ال cell دى باستخدام ال transistor ... هنلاقى من الصورة تحت فى figure a انه ال transistor عامة فيه ال three capacitors الأساسيين دول الى احنا درسناهم اكيد فى ال basics of the electronics ... هنلاحظ اننا لو وصلنا source & drain و body مع بعض هيبقى عندى ثلاثة capacitors بس in parallel بحيث ناحية فيها source or drain or body والتانية ال gate زى فى figure b كده وبالتالي محصلة الثلاثة هتكون capacitor واحد كبير الى فى figure c ... فال cell عامة بتكون زى figure d كده مكونة من pmos & nmos بحيث ان pmos موصل ال source & drain & body وبعض موصلهم على Vdd وموصل ال gate ب Vss وفى ال nmos موصل بردوا ال source & drain & body ببعض بس موصلهم ب Vss وال gate هى واصلة ب Vdd ... فهتلاقى ان nmos عبارة عن capacitor دلوقتى وال pmos هى كمان capacitor يعنى two parallel capacitors يعملوا مع بعض capacitor واحد بس اكبر



من التوصيلة الى شوفناها دي هنلاقى ان ال layout بتاعها هو ده:



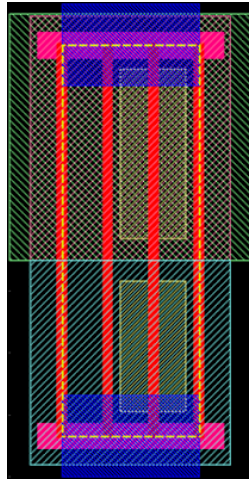
:Notes

- عيب ال decap cells وهى ان leakage current فيهم عالى وبالتالي هيزودوا ال leakage power فتاخذ بالنا
- احنا بنضيف ال Decap cells بعد ما بنضيف ال filler cells ونعمل dynamic voltage drop analysis يعنى بنشوف ال voltage drop فى كل حته فى ال chip قد ايه ... ال tool بتحقق target voltage drop يعنى ال acceptable voltage drop الى تفضل عنده ال cells شغالة كويس عن طريق انها بعض ال filler cells بال decap cells ولكن فى نفس الوقت تحاول تقلل ال cost of inserting decoupling capacitors (ال cost مقصود بيه ال area بتاعت ال Decap cells وال Leakage power بتاعتهم)

=====

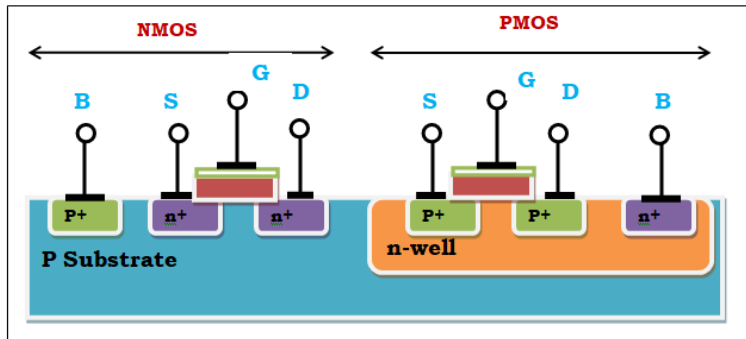
3. Endcap/boundary Cells:

بنطلق عليها endcap cells او boundary cells ومن اسمها باين انها بتتخط عند ال boundaries ... بردوا ملهائش اى function هى فقط فهتلاقى ال layout بتاعها بيتكون من Nwell, implant, poly layer, metal rails فقط ... بيوصل عليها VDD و VSS:



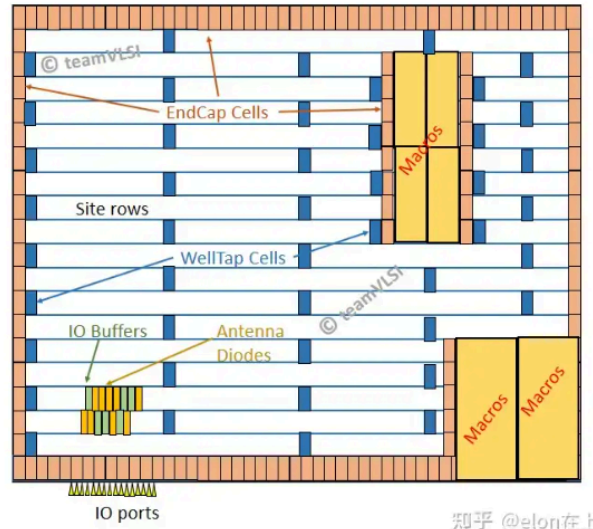
احنا عارفين ان فيه اكثر من chip بيتصنعوا على نفس ال silicon wafer ثم عشان تطلع كل chip لوحدها اكيد بتحصل عملية cutting للأطراف بتاعة ال chip وغالبا باستخدام laser بس ما علينا باستخدام ايه المهم انه واضح لينا ان ال chip boundaries بتتعرض لحاجات اصعب وقت التصنيع من باقى ال chip يعنى لو فيه standard cells من ال design عند ال boundaries فهى ممكن تتعرض ل damaging بنسبة اكبر من اى cell جوا ال chip فوجود ال encap cell هو حماية لإنها ملهاش اى functionality خالص يعنى هى physical فقط فحتى لو حصلها damage مش هتأثر على ال chip functionality بالعكس هكون حميت ال design cells بتاعتى ان يحصلها هى ال damage ده فخلينا دلوقتى نتكلم عن بعض ال damaging الى ممكن يحصل واحنا بنحمى ال standard cell منه:

- أثناء التصنيع ال cells الى عند ال boundaries بتكون معرضة ان يحصل damaging لل poly بتاع ال gate وبالتالي وجود ال encap cells بيضمن ان حتى لو حصل damaging لل poly هيكون لل endcap cell مش لل standard cell الأساسية وبالتالي مش هتأثر فى ال circuit functionality
- تانى حاجة هى ان فى DRC rules لل nwell layer (ودى الى بنعمل منها ال doping فى جزء من p-type substrate بحيث نعمل n-type substrate لل pmos) او ال implant layer (الى هى بتستخدم عشان نعمل source & drain سواء فى nmos او pmos الفكرة انها doping عكس الموجود يعنى فى nmos عندنا p type substrate فنعمل source & drain ب n doping وعلى العكس فى pmos عندنا n type substrate فنعمل source & drain ب p doping):



من ضمن ال DRC rules فيه rule بتحط min spacing بينهم وبين ال edge وده بردوا لحمايتهم من ال damaging أثناء التصنيع فوجود ال encap cell بيضمن انك سبيت ال min space المطلوبة دى فكداهى damage هيجصل لل endcap مش لل standard cells الأساسية

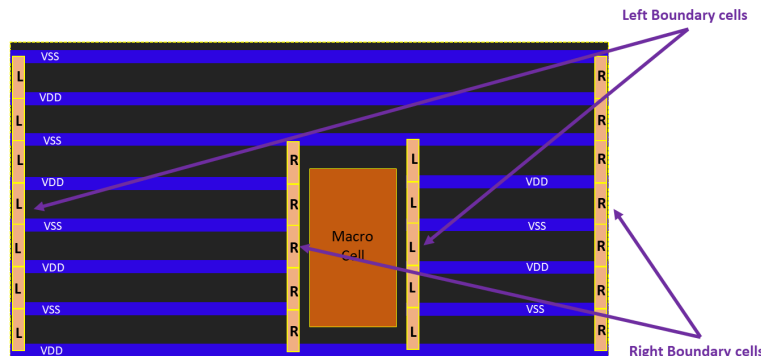
- على scale ال macros بقى يعنى لو انا حاطط macro عندى فى ال design بحط حواليه بردوا endcap cells عشان بتساعد فى ال integration بينه وبين macro block بحيث انها بتعمل well continuity يعنى زى ال filler cells احنا حطينها عشان نعمل well continuity بين ال standard cells وبعضها فففس الفكرة بتعمل well continuity بين ال blocks بقى ... لو محطينهاش بتلاقى ال engineer بيرسم well ring او احيانا metal حوالين ال block عشان يعمل well continuity فتسهيلا للموضوع بنحط ال endcap cells



知乎 @elon在上

- ال endcap cells فيه منها **types** حسب المكان الى هتخطه يعنى ال **types** بتختلف فى ال **orientation** بتاعها لكن كلهم بهدف واحد:
- **Horizontal endcap cells**: بتتخط عند ال edges بتاعة ال site rows يعنى فى left & right بتوع ال chip
 - **Vertical endcap cells**: بتتخط فى bottom & upper بتوع ال chip
 - **Corner endcap cells**: بتتخط فى corners ال chip ودى احيانا بيستخدموها يعنى ممكن يحط vertical & horizontal فقط زى الصورة المبسطة فوق كده بدل ال corner حط vertical & horizontal

حاجة مهمة كمان لوبصينا على horizontal endcap cells هتلاقى فيه اختلاف بين right & left فى مكان ال poly ... يعنى ال layout الى شوفناه فوق هتلاحظ ان فيه اتنين dummy poly على الشمال ووحد dummy كمان على اليمين وتانى واحد من على اليمين ده كده توصيلة ال gate (هى كمان توصيلة ملهاش functional goal بس دى الى موصلة ال gate فى النهاية يعنى) المهم بقى ده ال layout ال left endcap cell بحيث ان فيه اتنين dummy poly عند ال left side فلما يحصل damage عند ال left side of the chip هتلاقى دول الى حصلهم damage فمش مشكلة لسه الباقي موجود وبنفس الفكرة فى ال right بس هلاقى ان ال two dummy poly بقوا على اليمين بقى واكيد هتلاقى نفس الفكرة فى ال upper & bottom



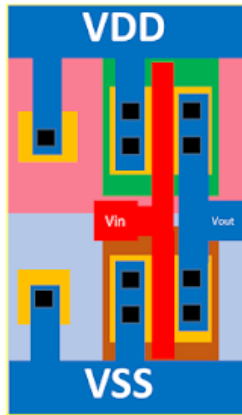
:Notes

- ال endcap cells بتتخط بعد ال macro placement يعنى من بدرى هنكون حاطينها قبل ما نخط ال design cells اصلا وده منطقى لإن امكانها معروفة ومهمة فعشان تتخط هى الاول عند ال boundaries وحوالين ال blocks ثم بقى تخط ال design cells باقى الاماكن
- في ال tool احنا بنختار اي standard cell بحيث نستعملها ك end cap cell ... بس طبعا مينفعش نستعمل اي حاجه وخلاص ... فيه شركات بتستعمل decap cell ك end cap cell بحيث تستفاد من مميزات ال decap cell وفي نفس الوقت تكون محسوبة ك end cap.

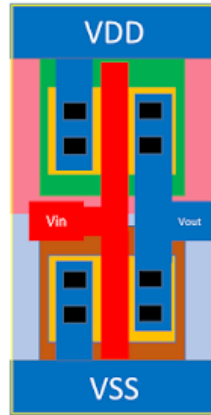
=====

4. Well tap Cells:

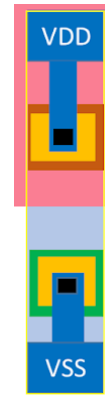
دى اتكلمنا عنها بالتفصيل فى floorplanning وقولنا اننا بنحطها فى ال floor planning وده الصح بس هنا زى ما قلنا هنجمع كل ال physical only cells فهنقول نفس الكلام الى قولناه فى ال floorplanning عنها ... دى عبارة عن cell بيكون فيها nwell واصله ب VDD و pwell واصله ب VSS .. طب ايه لازمتها؟؟ ال cells الى احنا عارفينها ال CMOS فيها بيكون nwell & pwell بعمل بيهم nmos & pmos وكل واحد له source & drain & bulk الملاحظ بقى ان كل ال bulk terminal فى ال cell بتكون واصله ب VDD فى nwell وبال VSS فى pwell فالى حصل انهم خلوا ال cells من غير bulk واطلقوا عليها tapless cells وعملوا بقى cell كل دورها هى توصيلة ال bulk دى وهى دى ال well tap cell:



Traditional std cell layout

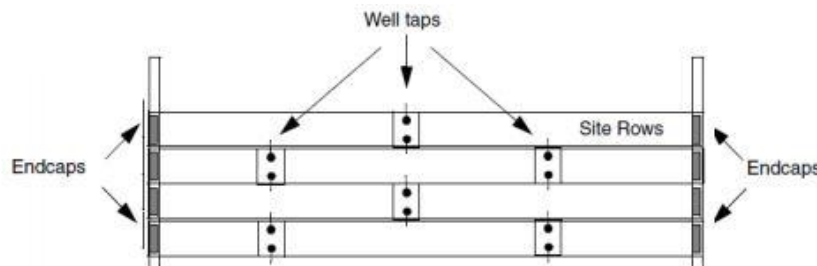


Tapless std cell layout



Well tap cell

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

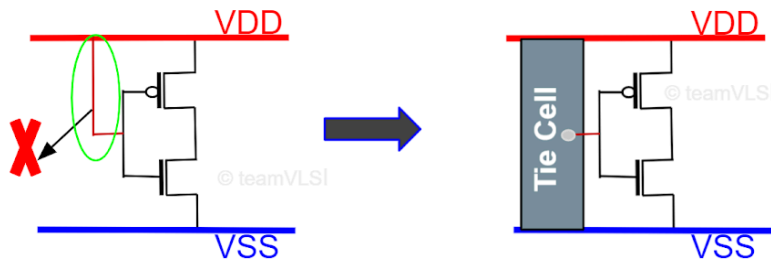


Note: مننشاش ان دى بنحطها فى ال floorplanning زى ما قلنا يعنى قبل ما نخط ال standard cells

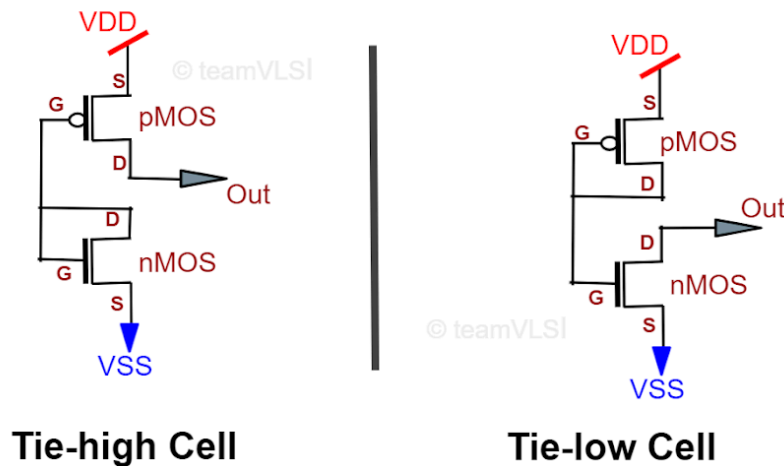
=====

5. Tie high and Tie low cells:

اتکلمنا عنها فی ال placement ... بتتحت عند ال input بتاع ال cells وهی tie cells طب ایه دی ولیه لازم احطها؟؟ دلوقتی احنا عارفین ان فیہ cells بیكون ال input بتاعها high یعنی VDD او low یعنی VSS ... لان مش لازم ال input بیقی جای من cell قبلها ممکن یكون input ثابت لیها VDD or VSS وفی الحالة دی بنجیبه من ال supply ... فحتی الان الی فی خیالنا انه ال wire الی جای من power mesh وهیوصل لل cell خلاص هاخذه اوصله علی input pin علی طول ... بس ده مش صح لان مهما حصل وحاولت انك تقلل supply noise هیفضل فیہ noise فیہ ومش هتضمن انه pure VDD or pure VSS واحنا عارفین ان ال input بتاع ای cell هو فی الاخر واصل علی gate of a transistor الی هی polysilicon تحتها oxide مع تقدم ال technology بیكون more thinner و sensitive اکثر فممكن یبوظ بسبب ال noise دی ... فینحط cell ملهاش ای input هی لیها VDD & VSS & output pins وبنسمیها tie cell



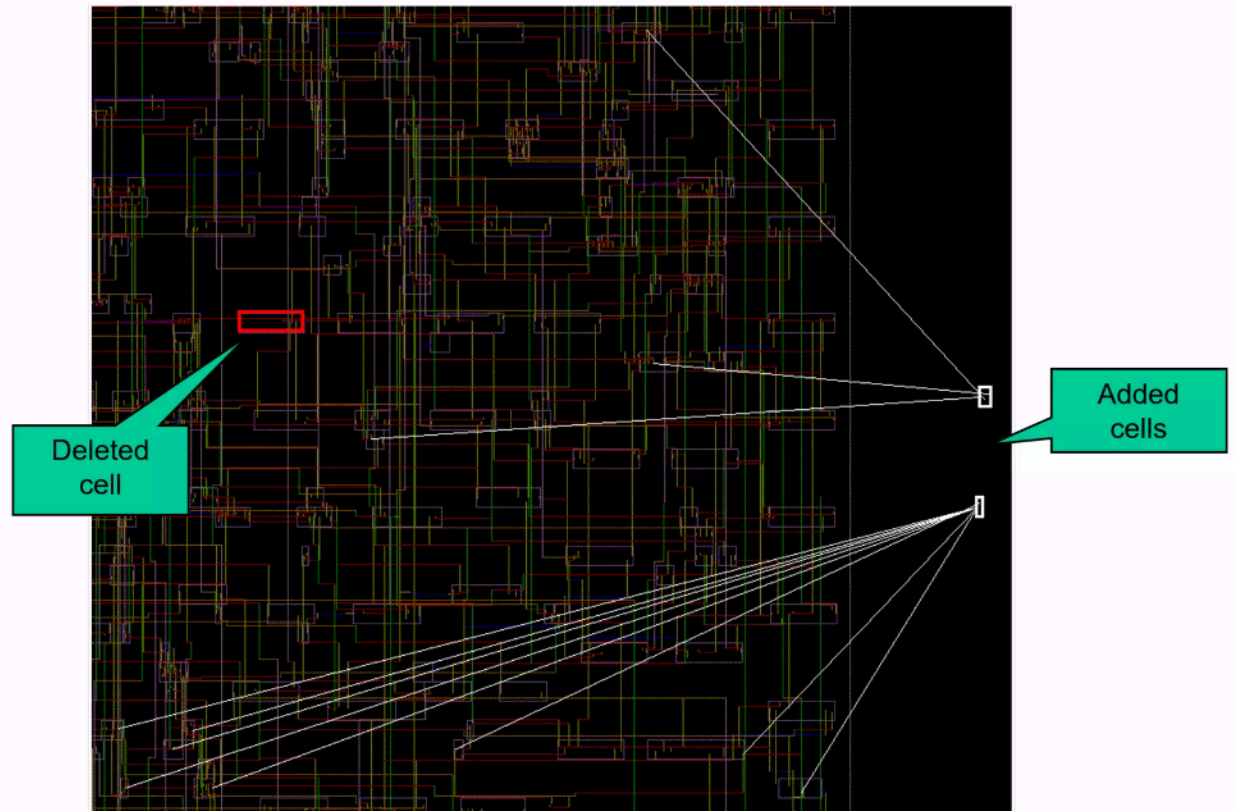
واما ان ال output بتاعها دایما high وبنسمیها tie high cell وتوصل علی gate input الی محتاج یوصل ل VDD او ان ال output بتاعها بیقی low دایما وبنسمیها tie low cell وتوصل علی gate input الی محتاج یوصل ل VSS



Note: مننشاش ان دی بنحطها بعد ما نحط standard cells یعنی بعد ال placement ومنطقی عشان تتحت عند ال input of the cells فلازم ال cells دی تكون موجودة اصلا

=====

Engineering change order (ECO):



ال ECO هو تغييرات بتعملها لل design عشان تصلح ال functionality بس بعد ما اول layout يطلع ... يعني انت بعد ما بتطلع layout بتعمل عليه verification وتتأكد من ان physically تمام وكمان ال functionality تمام وال timing تمام وهنتكلم عن كل ده بالتفصيل فى ال file القادم بس حاليا احنا عارفين ان بيحصل physically & functionaly & timing verification على ال layout الى هيطلع ... فلو كان فيه مشكلة ومحتاجين نرجع نعدل ونضيف cells او نشيل cells حسب ال functionality هصلحها ازاي من غير ما ارجع اعمل redesign من الاول

فهنرجع نعدل من اول step معينة مثلا هضيف cells فهرجع لل placement واكمل من عنده فيعني مش لازم ابدأ from scratch تاني وابدأ ال flow من الاول ... فالتعديلات دي بقى بنطلق عليها Engineering change order ECO .. هنلاقى بقى فيه نوعين من ECO او بمعنى اصح ممكن تعمله فى مرحلتين مختلفتين وبالتالي كل مرحلة هيكون ليك limitations مختلفة ... اول واحد هو الى اتكلمنا عنه دلوقتى يعني التعديلات الى بعملها بس قبل ما اصنع لسه يادوب مجرد layout وده بنسميه Non-Freeze silicon ECO والثاني هو انى اعمل تعديلات بس بعد ما تكون ال chip اتصنعت ورجعت اصلا وده بنطلق عليه Freeze silicon ECO واكيد ده الى هيكون فيه limitations كثير:

1. Non-Freeze silicon ECO:

زى ما قلنا دي قبل ال fabrication يعني ال layout طلع بس انا لاقيت مشاكل وانا بعمل verification عليه وبالتالي انا هرجع اعدل بس معايا سماحية عادى انى اعدل فى ال placement وال routing بأحسن optimization ممكن ... يعني هضيف cells مثلا فى ال placement واخلى ال tool تمشى عليها تاني وتعمل routing تاني فهتعمل احسن حاجة تقدر عليها بردوا ... عشان كده بنطلق عليها Non-Freeze Silicon لأنها متصنعتش لسه على ال silicon وبالتالي ال design بتاعى مش freezed يعني مش متصنع وثابت

خلاص على حاجة معينة مقدرش اغير فيها ... لا هو لسه متصنعش فاقدر اغير واعدل اى حاجة عايزها ... ال cells بقى الى بضيفها دى بتطلق عليها ECO cells بما انك ضيفتهم وانت بتعمل ECO

والمنطقى ان انت هنا تشيل اى filler cells كنت حاططها لو هترجع تضيف cells تانى عشان تقضى المكان لل logic cells بتاعتك الاول ثم بعد كده تبقى تشوف ايه الفراغات الى اتبقت وتحط ال filler cells تانى

=====

2. Freeze silicon ECO:

زى ما قلنا بردوا ان ده بيكون بعد ما ال chip تتصنع ... ازاي؟؟!! مش المقصود انى هعمل تغييرات فى ال chip الى قدامى دى واخلوها تشتغل صح وانا قاعد كده ... لا هو اكيد انا محتاج اصنعها تانى بس هل هبدأ من الاول او بمعنى اخر هغير فى ال layout بتاعى كله ولا الاحسن انى اعمل تعديلات بسيطة وهنا التعديلات بتكون فى التوصيلات بين ال cells يعنى بغير فى metal layers فقط واحنا عارفين ان عملية التصنيع بيتم على خطوات وكل خطوة ب maks الى هو تكلفته عالية جدا فانا كده مش هغير غير فى metal mask بس ووفرت وقت كبير كمان انى مش همشى كل ال flow من الاول ... لا انا هعدل فى توصيلات ال cells يعنى ال routing بس الى هيتغير

طب ازاي بقى نهجهز ال circuit لحاجة زى دى يعنى لو انا عايز اضيف cell مثلا هعملها ازاي وانا هغير فى ال routing بس؟؟ الاجابة انك بتكون حاطت cells زيادة فى ال design بتاعك هما ليهم function بس انت مش بتستخدمهم فى ال design دلوقتى ... انت حاطتهم احتياطي عشان لو بعد ما ال chip اتصنعت وجت وعملتلتها testing ولاقيت فيه مشكلة ومثلا وصلت ان الحل انك تحط buffer فى path ما او تزود and gate and مثلا فتبقى اصلا انت حاطط buffer و and بس مش مستخدمين وترجع بقى تعيد توصيل ال metal routes بحيثت تدخلهم معاك فى ال path الى انت عايزه وبكده هتلاقى ان اماكن ال cells متغيرتش ولا حاجة هى كل حاجة زى ما هى فى مكانها بس انت كل الى عملته انك غيرت فى التوصيلات بينهم

ال cells الى ضيفتها بردوا هى ال ECO cells بس ال cells الى بتكون موجودة اصلا وانت بتستخدم بعضها (ويبقوا ECO cells بتوعك) هما اصلا ليهم اسم وهو Spare cells وهنتكلم عنها بالتفصيل اكثر دلوقتى

Spare cells:

ال cells الى بتكون حطينها فى ال design احتياطي زى ما قلنا بنطلق عليها spare cells فخلينا نتكلم عنها اكثر ... دول بيكونوا مجموعة من ال standard cells المتعارف عليهم او الى بتحتاجهم اغلب الوقت زى MUXs , AND , OR , buffers , ... وغيرهم يعنى بتحط الحاجات الأساسية لان التغيير الى هتعمله اكيد مش انك كنت ناسى تحط macro مثلا لا دى تغييرات بسيطة فى ال path وال standard cells الاساسية تقدر تكون بيها functionalities مختلفة زى مثلا تعمل AOI (AND OR Invert) Circuit عن طريق انك تستخدم AND ثم OR ثم Inverter من ال spare cells الموجودة وهكذا

وانت بتطلع اول layout وحاطط ال spare cells الى مش مستخدمهم دلوقتى هتكون ال inputs بتاعة ال spare cells متوصلة على Vdd or Vss (مش بشكل مباشر لانك بتوصله على Tie cells الى اتكلمنا عنها فوق) وال outputs هتسييه floating عادى مش هتوصله بحاجة ... ال ouput مش واصل على حاجة فلو جاله noise مش هيبوظ حاجة فسيبناه floating لكن ال input واصل على ال transistor gate فلو جه عليه noise ممكن يحرقها او هيعمل switching عالى يستهلك power على الفاضى فعشان كده خليناها واصله ب tie cells

فكده وجود ال spare cells خلانا منغيرش فى ال placement الى اتعمل بيه ال initial layout ونغير فى ال routing بس تغييرات بسيطة يعنى الى الحاجة الى هتتغير فى التصنيع هى ال metal mask بس مش كل ال masks وكمان انت هتوفر وقت كبير انك مش هتعيد الشغل من الاول لا انت هتكمل من بعد ال placement

بكده ال ECO cells فى حالة ال freeze silicon انت مضيفتهمش لا انت استخدمت من spare cells الموجودة كإنك عملت mapping لل ECO cells بتاعتك على ال spare cells الموجودة ... وبالتالي لو كان فيه cells فى ال design بعد التعديل مبقتش مستخدمة فى ال design فكده بقت هى موجودة بس مش مشاركة فى ال design بتاعى يعنى بقت ضمن ال spare cells دلوقتى

فيه أسئلة ممكن تيجى فى بالنأ زى انه طب هحط قد ايه spare cells وهما بيكونوا من ال basic standard cells بس اختارهم بالطبط ازاي واوزعهم فى ال design ازاي دى تعرفها بالخبرة بس احنا بنفهم ال concept هنا

Note:

- المفروض تتحط قبل ال standard cells placement بحيث اوزعها براحتى فى ال chip قبل ما احط ال cells والاقى مثلاً congestion فى مكان فمعرفش احط فيه
- ممكن تلاقيها تحت عنوان physical only cells ... عادى لإن طول ما هى spare cells فهى احتياطى مش مستخدمة فى ال design تبقى physical only حتى لو ال cell نفسها ليها function فهى مش مستخدمة دلوقتى

=====

3. ECO File:

التغييرات الى هتحتاج تعملها ممكن تبقى بسيطة وتكتب command لكل واحدة وتعملها لو انت هتستخدم ال ICC shell او تعملها manually من ال Graphical user interface GUI ولكن اوقات بتبقى تغييرات كتير عايز تعملها فى script وت run بردوا فممكن بقى انك تحطها فى file وهو ده الى بنسميه ECO file فهو script عادى انت حاطط فيه PnR tool commands لإن اى تعديلات هتعملها اكيد هتكون بواسطة ال commands الى احنا شرحناها قبل كده سواء انت هتعمل تغييرات فى placement ثم routing فى حالة ال non-freeze او فى ال routing فقط فى حالة ال freeze فانت عادى هتستخدم ال commands الى انت عارفها قبل كده فاكيد هتكتب PnR tool commands

=====

4. Non-Freeze Vs. Freeze silicon ECO:

نعمل مقارنة سريعة لكل الى شرحناه فوق ... وبعد ما تعمل ال ECO زى ما قلنا هتكمل routing فى الحالتين انت هتكمل خطوة ال routing عادى

Non-Freeze Silicon ECO	Freeze Silicon ECO
<ul style="list-style-type: none"> Placement is not fixed Allows to add new cells Spare cells are not required 	<ul style="list-style-type: none"> Placement is fixed No cells are added or moved Spare cells are required

=====

Antenna Effect:

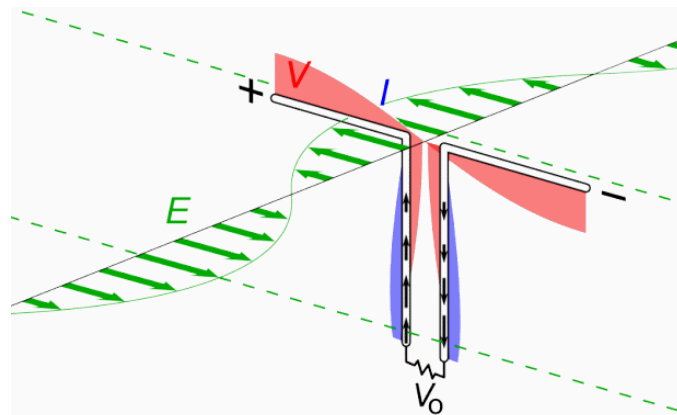
دى مشكلة بتحصل اثناء التصنيع ممكن تسبب circuit damage فخلينا نقولها باختصار ثم هنتكلم بالتفصيل لو حابب تفهم اكثر ثم نقول الحلول ... اولاً ال antenna هنا مقصود بيها ال wire connected to the gate .. المشكلة هنا ان وقت التصنيع بيحصل تراكم للشحنات يعنى accumulation of charges على ال metal wire الى واصل على ال gate بتاعت ال Transistor يعنى على ال gate نفسها وده ال antenna effect ... ال gate عندها very thin oxide فلو كان فيه charges كثير ممكن تعمل damage لل oxide وبالتالي ال transistor كله باظ ... خلينا بقى نفهم ال effect ده بيحى منين اصلاً وبيعمل ايه فى ال gate بالتفصيل

ليه بقى اصلاً ال wire اطلقوا عليه antenna؟؟؟ ال antenna احنا عارفين انها conductor وبتنقل وتستقبل ال electromagnetic waves ... بتبقى transmitter عند طريق انها بتحول ال electrical signal الى بتمشى فيها ال EM wave والعكس ك receiver بتحول ال EM wave الى electrical signal ... احنا مهتمين بحالة ال receiver ... ليه؟؟؟

لأن فى مرحلة من التصنيع وهى ال plasma etching ودى بنشيل فيها زيادات ال metal بحيث يبقى ال wires فى ال tracks المطلوبة مذبذبة ... ال plasma دى هى ionized gas يعنى غاز تم تأينه او بمعنى اخر فقد electrons والى بيخليه يتأين هو ال EM field لأنه بيوفر طاقة عالية جدا تخرج ال electron من مكانه ... سؤال ممكن يجى فى بالنا وهو انا ليه اصلاً استخدم ionized gas عشان اشيل الزيادات وده اجابته ان ال atoms بتاعت الغاز زى ما هى كده مش هتتفاعل مع ال wire بحيث يحصل etching واشيل الزيادات من ال wire ... فانا محتاج اخليها ionized بحيث انها تتفاعل مع المادة الى هتقع عليها وتسهل انها تتشال ... وعلى حسب انا عايز اعمل etching لايه بيحدده الغاز ده هيكون عبارة عن ايه ... وبالتالي ال wire اتعرض ل EM wave زى ال antenna فى ال receiving ... تعالى بقى نشوف هو ايه الى بيحصل فى ال antenna وقت ال receiving عشان نفهم ايه الى بخلى تحول ال wire الى antenna مشكلة عندنا

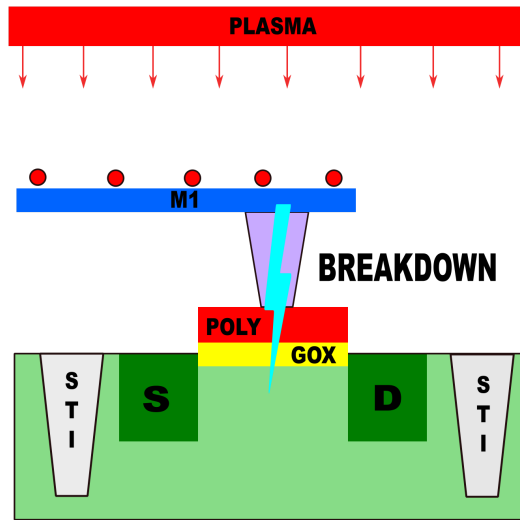
مش عايزين ندخل فى تفاصيل اوى فممكن نفهم الموضوع ببساطة انه ال conductor هيعدى عليه EM wave فهنلاقى ال electric field الى معدى ده بيحرك ال charges الموجودة فى ال metal وزى ما احنا عارفين انه بيحرك ال positive فى اتجاه وال negative فى الاتجاه معاكس وبالتالي هلاقى حصل charge accumulation عند اطراف ال antenna ادى الى وجود voltage variation كبير وبما ان ده alternating electric field فلما يغير اتجاهه هيتغير مكان ال positive & negative charges هلاقهم عكسوا اماكنهم زى ما باين فى الصورة هنلاقى اما بيكون اتجاه ال electric field شمال فبتتجمع ال positive charges ناحية الشمال وال negative فى اليمين فيكون فيه voltage variation على ال antenna واما يعكس ال electric field اتجاهه ويبقى ناحية اليمين هنلاقى ال positive charges اتجمعت ناحية اليمين وال negative ناحية الشمال وهكذا ... الصورة دى ليها animation ممكن تشوفه

[من هنا](#)



فبنفس الفكرة فى ال wire عندنا هو conductor زى ما ال antenna هى كمان conductor فلما يتعرض ل EM filed أثناء عملية ال plasma etching هتتحرك ال charges بنفس الطريقة بردوا ويحصل accumulation عند اطراف ال wire يعنى voltage variation ... المشكلة بقى فى ال wire الى واصل على ال transistor gate لأن كده هيجصل charge accumulation على طرف ال wire الواصل على ال gate يعنى على ال transistor gate نفسها ... طب ليه ده مشكلة؟؟

عند ال gate احنا عارفين ان فيه oxide بيكون بين ال silicon wafer و ال polysilicon الى بيوصل على ال terminal وبيكون very thin oxide يعنى ال thickness صغير ... كل ما ننتقل الى technology احدث ال oxide thickness بيصغر اكثر وبالتالي بيتأثر أكثر بال charges الى على ال gate بمعنى لو كانت كثير ممكن يحصل tunneling يعنى تعدى خلال ال oxide ده بسبب ان سمكه صغير جدا وتدخل ل substrate فهتتحرق ال transistor وبالتالي ال circuit كلها damaged لأن ال functionality باظت



يبقى كده ال EM field الى بيتعرض له ال metal أثناء ال plasma etching فى ال manufacturing بيسبب charge accumulation على ال wire الواصل بال transistor gate فممكن انه يحرق ال transistor

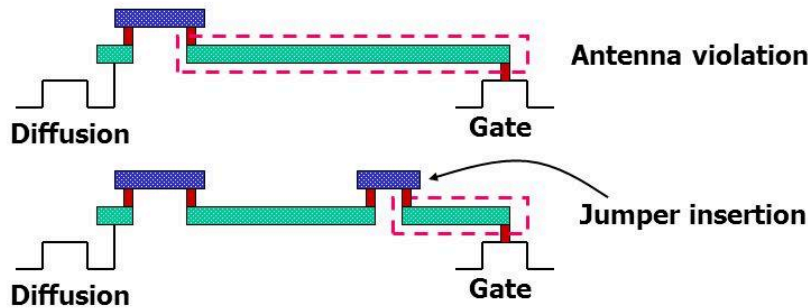
تجنباً للمشكلة دى ففيه rule بيحطها المصنع وهى ال antenna rule ودى بتحدد ال wire area الى واصل على ال transistor gate يعنى بتحط upper limit لل wire area بحيث ميزدش عن كده لأن زيادة ال area معناها انى هيجصل charge accumulation اكثر فبزود امكانية حدوث gate damage لكن لو ال wire area صغير فحتى لو حصل charge accumulation هيكون عدد ال charges مناسب انه يحصل damage لل oxide فالى هيقول مناسب ولا لا هى rule من المصنع الى هيصنع بيعرف parameter اسمه antenna ratio (وهنا ال antenna الى هى ال wire connected to the gate) بحيث ان:

$$Max\ Antenna\ Ratio > \frac{Antenna\ area}{Gate\ area}$$

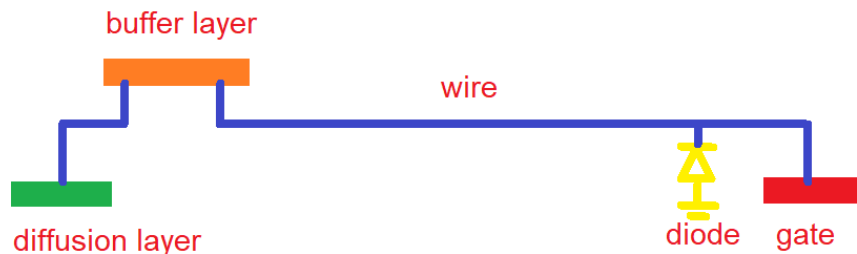
- Antenna area: Area of the metal connected to the gate
- Gate area: total area of the gate

طيب لو انا لاقيت بقى antenna violation عندى بعد ما خلصت ال design ممكن اعمل ايه؟؟

اول حل انى اعمل **layer jumping** ... يعنى ال wire الطويل الى واصل على ال gate عايز اقلل ال area بناعته فبذل ما واصل كل ال wire ده انا هاجى قبل ال gate بشوية واطلع ب via ل metal layer ثانية ثم ارجع انزل ب via لنفس ال metal layer الى كنت فيها واصل على ال gate فكد هلاقى ان بقى جزء صغير من ال wire هو الى واصل على ال gate فقللت ال wire area بشكل كبير زى ما باين فى الصورة ... اثناء ال fabrication التصنيع بيتم layer by layer من تحت ل فوق ... يعنى لما انا كنت بعمل تنصيع لل metal wires كانت ال vias لسه متحطيتش ... فكد ال wire الكبير مش واصل لسه على ال wire الصغير اللي واصل على ال gate فكد ال gate مش شايقة ال accumulation of charges ... عيوب الحل ده هو انه زود عدد ال vias

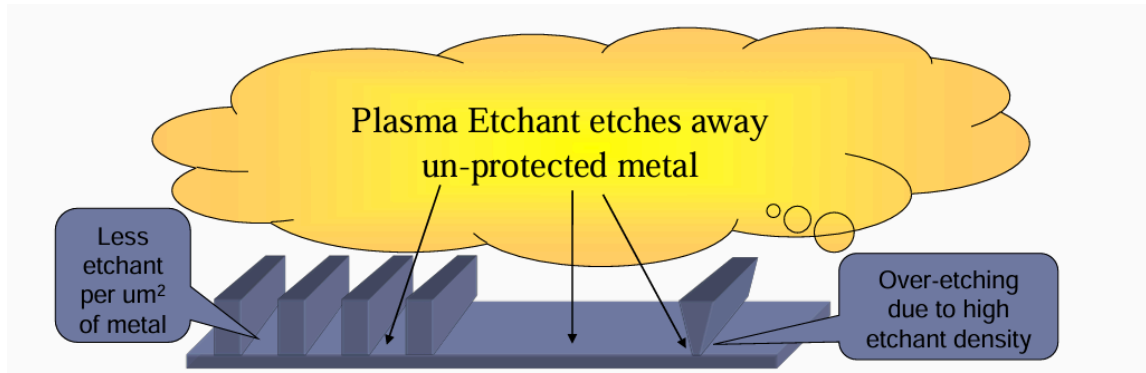


الحل التانى هو **diode insertion** عند ال gate بحيث افرغ فيه ال charges الزيادة ... بس انا مش عايز يكون ال diode ده path أساسى لل current فى ال circuit ويستهلك power على ... فعشان كده بيكون reverse biased بحيث انه فى ال reverse بيسمح بمرور current بسيط زى ما احنا عارفين وده كافى انه يفرغ ال charges الزيادة ... طبعا هو فى الصورة هنا symbol لكن اكيد شكله مش كدا وهو بيتصنع بس الصورة عشان نفهم الفكرة ... عيوب الحل ده هو ان وجود diode هيكون زود ال capacitance فى ال path ده وبالتالي ال delay هيزيد فممكن يحصل timing violation اما نعمل STA check

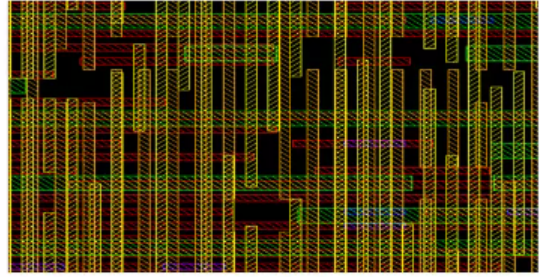
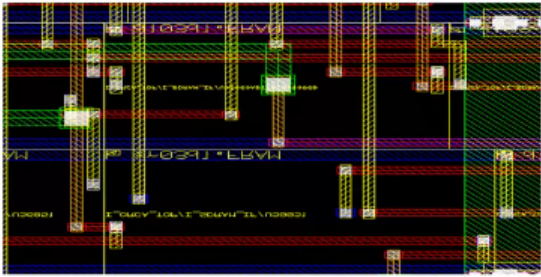


Metal Fill

ده حل لمشكلة ال metal over etching بس ايه المشكلة دى الاول؟؟ دى ممكن تحصل فى التصنيع لو انا عندى مكان فيه low metal density يعنى عندى عدد metal routes قليل .. احنا محددين ال اماكن ال metal tracks (الى تمثل ال routing resources عندك دلوقتى) من ال floorplanning stage ثم فى ال routing حطينا ال metal routes بالفعل فيهم فاكيدهم مستخدمتش كل ال tracks لإنك مش هتخلص كل ال resources فانت عندك minimum metal density rules وى بتقول انه فى مساحة وليكن x المفروض متحطش اقل من مثلا 5 routes فممكن فى المساحة دى تكون انت حاطط route واحد وسايب باقى ال tracks مش مستخدمهم وبالتالي ال route ده وبالتالي هيتأثر بشكل اكبر فى عملية ال etching لدرجة انه ممكن يحصل over etching يعنى ينشال من ال metal نفسه مش مجرد الزيادات وبالتالي يحصل failure



فحل المشكلة دى انك تعمل metal fill يعنى تحط metal فى ال tracks الى انت مش مستخدمهم حتى لو مش هتستخدم ال metal route ده ... انت بس بتحطه عشان متعملش violation لل metal density rule ... يعنى الصورة تحت هنلاقى ال layout على الشمال كان فيه routes قليلة فعمل metal fill يعنى زود عدد ال tracks المستخدمة فبقى الشكل زى ال layout على اليمين وهلاحظ زيادة عددهم



Note: ناخد بالننا من حاجة مهمة وهى ان لو عندى violation فى minimum metal density rule فقبل ما اعمل metal fill لازم ابقى خلاص مش محتاج ال routing resources تانى لانى هستخدم اغلب الموجود وفاضى دلوقتى ك dummy routes فمثلا مش هقدر بعد كده اعمل fixing لل routing او لل antenna violation لانى بحتاج routing resources عشان اصلحهم فخلص كل حاجة الاول ثم اعمل metal fill



ICC tool:

- Redundant vias insertion:

Command:

```
define_zrt_redundant_vias [-from_via {list_of_from_vias}]\
[-from_via_x_size {list_of_contact_numbers}] [-from_via_y_size {list_of_contact_numbers}]\
[-to_via {list_of_to_vias}] [-to_via_x_size {list_of_contact_numbers}]\
[-to_via_y_size {list_of_contact_numbers}]
```

يحدد بيه شوية options هستخدمها اما احط ال redundant vias بال command الجاي .. بعض الى بحدده بيكون:

- **from_via**: يحدد أسماء ال vias الى عايز ابدلها وال default انها تكون empty list يعنى هيغير كل ال vias الموجودة
- **from_via_x_size**: يحدد size بتاع ال via array او بمعنى اخر عدد ال contacts الى فيها هل فيها via واحدة ولا اتنين ولا كام بالظبط في ال x direction بتاع ال array على اساس اى array ليها x & y directions فكل via موجودة في from_via المقابل ليها هنا هو عدد ال contacts فيها في x direction
- **from_via_y_size**: يحدد size بتاع ال via array او بمعنى اخر عدد ال contacts الى فيها هل فيها via واحدة ولا اتنين ولا كام بالظبط في ال y direction بتاع ال array فكل via موجودة في from_via المقابل ليها هنا هو عدد ال contacts فيها في y direction
- **to_via**: يحدد أسماء ال vias الى هبدل بيها وال default انها تكون empty list
- **to_via_x_size**: يحدد size بتاع ال via array الى هتتحط بقى او بمعنى اخر عدد ال contacts الى فيها هل فيها via واحدة ولا اتنين ولا كام بالظبط في ال x direction بتاع ال array فكل via موجودة في to_via المقابل ليها هنا هو عدد ال contacts فيها في x direction
- **to_via_y_size**: يحدد size بتاع ال via array الى هتتحط او بمعنى اخر عدد ال contacts الى فيها هل فيها via واحدة ولا اتنين ولا كام بالظبط في ال y direction بتاع ال array فكل via موجودة في to_via المقابل ليها هنا هو عدد ال contacts فيها في y direction

فمثلا لو انا عايز ابدل ال via ب 1*2 via array هيبقى ال command زى ال example تحت .. حدد اسماء ال vias الى هيسبدلهم وقال انه هيسبدل كل واحدة ب via array في x واحد و y اتين يعنى كإن عند one column فيه two rows ... أسماء ال vias الى هيثم التبدل بيها مش متحدد ببقى ال tool هتتحط هي المناسب من عندها بس بال dimension الى اتحدت:

```
prompt> define_zrt_redundant_vias -from_via {via1 via2} \
-to_via_x_size {1 1} -to_via_y_size {2 2}
```

=====

Command:

```
insert_zrt_redundant_vias [-effort low | medium | high] [-nets {collection_of_nets}]\
[-timing_preserve_nets {collection_of_nets}]\
[-timing_preserve_setup_slack_threshold slack_value]\
[-timing_preserve_hold_slack_threshold slack_value]
```

الطبيعي انه يبديل كل via ب two via array الا لو احنا كنا محددين حاجة تانية باستخدام define_zrt_redundant_vias ... بعض ال options هنا:

- **effort**: ال effort الى هتبذله ال tool وكل ما زاد كان accuracy احسن بس وقت اطول وال default هو medium
- **nets**: بحط لل tool ال nets الى انا عايز انها تستبدل ال via فيها ب array of vias ... الطبيعي انها بتستبدل كل الى في ال design الا لو انا حددت بال option ده nets معينة فقط
- **timing_preserve_nets**: ال nets الى بحددها هنا بتعتبرها ال tool انها timing critical nets فمش بتعمل فيها via insertion
- **timing_preserve_setup_slack_threshold**: هنا بحدد قيمة لل setup slack فإى net عندها setup slack قد القيمة دى او أسوء (يعنى أقل) ال tool مش هتخط فيها redundant vias
- **timing_preserve_hold_slack_threshold**: هنا بحدد قيمة لل hold slack فإى net عندها hold slack قد القيمة دى او أسوء (يعنى أقل) ال tool مش هتخط فيها redundant vias

هنلاحظ انه فى ال examples تحت اما بيحتاج يعدل على default الى هو يبديل ال via ب two via array بيستخدم الاول define_zrt_redundant_vias ثم insert_zrt_redundant_vias

```
EXAMPLES
The following example replaces all single cut vias by 2-cut via arrays.

prompt> insert_zrt_redundant_vias

The following example replaces vias named VIA23 and VIA34 with vias
named VIA23 and VIA34 that have corresponding x-sizes of 1 and 5 and y-
sizes of 3 and 1.

prompt> define_zrt_redundant_vias \
  -from_via {VIA23 VIA34} -to_via {VIA23 VIA34} \
  -to_via_x_size {1 5} -to_via_y_size {3 1}
prompt> insert_zrt_redundant_vias

The following example replaces a via named VIA23 with a via named VIA23
that has a corresponding x-size of 1 and a y-size of 3.

prompt> define_zrt_redundant_vias -from_via VIA23 -to_via VIA23 \
  -to_via_x_size 1 -to_via_y_size 3
prompt> insert_zrt_redundant_vias

The following example replaces a via named VIA23 with a via named VIA23
that has a corresponding x-size of 1 and a y-size of 2 on net net1.

prompt> define_zrt_redundant_vias -from_via VIA23 -to_via VIA23 \
  -to_via_x_size 1 -to_via_y_size 2
prompt> insert_zrt_redundant_vias -nets net1
```

=====

- Filler cell insertion:

Command:

```
insert_stdcell_filler [-cell_without_metal {cellx8 cellx4 ...}][connect_to_power power_net_name]\  
[connect_to_ground ground_net_name] [-pin_net pin_net_list]\  
[-ignore_hard_placement_blockage] [-ignore_soft_placement_blockage]\  
[-check_only] [-bounding_box {llx lly urx ury}]
```

ده بستخدمه عشان احط ال filler cells ... بعض ال options بتاعته:

- **cell_without_metal:** بتكتب list باسماء ال cells الى عايز ال tool تستخدمهم فى ال filling بس ناخد بالنا ان ال tool بتستخدمهم بالترتيب الى هتكتبه بمعنى بتشوف المكان الى هى عاوز تملاه ب filler cell وتبص على اول cell عندك فى ال list دى لو مناسبة ليه سواء قده بالظبط او اصغر هتخطه ولو اصغر هتبقى ملت جزء من الفراغ ده واتبقى جزء هترجع تدورله تانى فى ال list من الاول بردوا يعنى هتاخذ نفس ال cell عادى لو قد الفراغ او اصغر وهكذا فانت المفروض تحطهم من الاكبر للصغر بحيث هى تدور فى ال sizes الكبيرة الاول عشان لو واحدة كبيرة تكفى يبقى احسن من انك تحط اتنين صغيرين مثلا ... ال cell دى بتكون زى ال layout الى شوفناه فيها supply & ground rails يعنى مش عشان without metal يبقى مفياش ... بس المقصود ان مفياش metal تانية وعشان كده ICC مش هتعمل DRC check بعد ما تحطها لان DRCs بتاعة ال poly وال transistor level عامة بتكون فى physical verification ب tools تانية غير ICC فمفياش metals فهنت check على ايه خلاص
- **cell_with_metal:** بتكتب list باسماء ال cells بردوا وال tool بتستخدمهم بنفس الفكرة وهى بتدور فلانم تحطهم من الكبير فى الحجم للصغير بس الفرق بينها وبين ال option الى فات هى ان دى بيكون فيها metal routes ... طب هستفاد منها ايه؟؟ هتبقى بتحل مشكلة ال metal over etching كمان لانك زودت metal بس ملوش functionality فمثلا لو فيه مكان محتاج metal fill فممكن اصلا تخلى ال filler cells تكون with metal
- **connect_to_power:** بكتب اسم ال power net الى هتوصل بيها ال cell الى هى اغلب الوقت بنطلق عليها VDD ... مينفعش تستخدمه مع option pin_net وهنشوف ليه اما نتكلم عنه
- **connect_to_ground:** بكتب اسم ال ground net الى هتوصل بيها ال cell الى هى اغلب الوقت بنطلق عليها VSS ... ده كمان مينفعش تستخدمه مع option pin_net
- **pin_net:** ده بحدد بيه بردوا اسماء ال power & ground nets الى هيوصلوا على ال cell عشان كده مينفعش استخدمه مع connect_to_power او connect_to_ground بس الفرق ان ده استخدمه لو عندى design فيه كذا power domain او بمعنى اخر ان كل area بتشتغل على power مختلف فال list بتكون على الشكل ده:

{pin_name net_name voltage_area_name} — Ex: {{VDDA PWR2 VA1}}

ال pin_name ده اسم ال power pin فى ال filler cell نفسها ... ال net_name ده اسم ال power net بتاعى الى هوصل عليه ال pin دى ثم ال voltage_area_name ده بحد اسم ال voltage area فى حالة ان عندى multiple voltage domains فيقول ده تبع انى domain يعنى وبالتالي مش هيكون موجود لو انت اصلا عندك single voltage domain هتكون بس ال pin_name & net_name الى موجودين

في ال example هلاقي حاطط one list فمممكن تزود عادى حسب لو محتاج اكثر ... ناخذ بالننا ان اى power pin في ال filler cell انت مش هتذكر هي متوصلة بايه في ال power فهي هتتوصل by default بال main power source فانت مش محتاج تكتب في list غير ال pins الى هتوصل على ال power domains الثانية مش الاساسى بتاعك ... والكلام ده ينطبق على ال power & ground بردوا عادى

- **ignore_hard_placement_blockage**: يعنى مش هياخد في اعتباره ال hard placement blockage الى شرحناه في ال placement بالتفصيل يعنى هيحط filler cells فيه عادى بس by default ال tool مش هتخط filler cells فيه الا اما انت تقولها تحط عن طريق ال option ده
- **ignore_soft_placement_blockage**: يعنى مش هياخد في اعتباره ال soft placement blockage الى شرحناه في ال placement بالتفصيل يعنى هيحط filler cells فيه عادى بس by default ال tool مش هتخط filler cells فيه الا اما انت تقولها تحط عن طريق ال option ده
- **respect_keepout**: ده معناه انه ال filler cells مش هتتخط في اى soft or hard keep out margin الى بنعمله على ال hard macros ... ال keep out margin شرحناه بالتفصيل في ال placement
- **bounding_box**: بحدد coordinates معينة ل rectangular box هو ده الى هحط فيه filler cells بدل ما احط في كل ال chip بس ال default ان ال tool بتخط في كل ال chip
- **check_only**: ب run بال option ده معناه اني في checking mode يعنى مش هحط filler cells دلوقتي انا بشوف اصلا ايه الاماكن الفاضية فده هيرجعلى الاماكن الفاضية وايه ال filler cell المناسبة للمكان ده من ال std library الى مستخدمها بس من غير ما يحط ال cells هو بس مجرد report

```
EXAMPLES
The following example fills empty spaces with nonmetal filler cells
named FILL_4X, FILL_2X, and FILL_1X in the specified order. If the
sizes of the filler cells are such that FILL_4X > FILL_2X > FILL_1X,
this ordering minimizes the number of filler cells added by using the
larger filler cells first.

prompt> insert_stdcell filler \
-cell_without_metal {FILL_4X FILL_2X FILL_1X}

The following example fills empty spaces with metal filler cells named
FILL_4XM, FILL_2XM, and FILL_1XM in the specified order. The filler
cells are inserted at places where the design rules are satisfied. If
design rule errors occur, the command uses the nonmetal filler cells
named FILL_4X, FILL_2X, and FILL_1X.

prompt> insert_stdcell filler \
-cell_without_metal {FILL_4X FILL_2X FILL_1X} \
-cell_with_metal {FILL_4XM FILL_2XM FILL_1XM}
```

=====

- Well tap cell insertion:

Command:

```
add_tap_cell_array [-ignore_soft_blockage true/false] -master_cell_name tap_cell_name_in_lib\  
-distance tap_pitch [-connect_power_name VDD] [-connect_ground_name VSS\  
[-respect_keepout_pattern stagger_every_other_row] [-tap_cell_identifier cell_name]
```

نحط ال well tap cells باستخدام ال command ده وزى ما شوفنا هى بتتحط ازاي 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 خرينا نشوفهم:

- **ignore_soft_blockage**: يعنى هل ال cells هتأخذ فى اعتبارها ال soft blockages المعمولة فاحط true ولا هتتعامل معاها كأنها مش موجودة فاحط false
- **master_cell_name**: هحط اسم ال cell المستخدمة ك well tap cell ودى لازم تكون موجودة
- **distance**: المسافة بين ال centers لاتنين cells جنب بعض وبنطلق عليها pitch: distance from centre to centre ودى الى شوفناها اما اتكلمنا عن شرح ال well tap cells وان فيه مسافة ثابتة بين كل اتنين جنب بعض ... بيكون رقم لازم يكون موجود
- **connect_power_name**: هنا بحط اسم ال power net الى هتوصل عليها والى اغلب الوقت بنسميها VDD
- **connect_ground_name**: هنا بحط اسم ال ground net الى هتوصل عليها والى اغلب الوقت بنسميها VSS
- **respect_keepout_pattern**: ده معناه انه ال cells وهى بتتحط بتأخذ بالها انها منتحطش فى الامكن الى غير مسموح انى احط فيها cells زى ال keep out margin مثلا ... فليها option اسمه **stagger_every_other_row** وده معناه انه بيدأ 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 وابع row وهكذا يعنى فى باقى ال rows الى فيها 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 فمممكن ابص عليها ك columns هنا واشوف ال columns الى فيها keep out patterns ومحطش فيها welltap cells واحط فى الباقي باستخدام **stagger_every_other_column**
- **tap_cell_identifier**: اسم هطلقه على ال well tap cell array الى هحطها دى

=====

- Tie cells insertion

Command:

```
connect_tie_cells -objects {object_coll} -obj_type port_inst | cell_inst | lib_cell\  
[-tie_high_lib_cell lib_cell] [-tie_low_lib_cell lib_cell] [-tie_highlow_lib_cell lib_cell\  
[-tie_high_port_name port] [-tie_low_port_name port] [-max_fanout number]  
[-max_wirelength number] [-incremental true | false]
```

هحط ال tie high & tie low cells الى اتكلمنا عنهم:

- **objects**: ده option لازم بيقى موجود ... هحط فيه الحاجة الى عايز اوصلها على ال tie high & tie low cells مثلا ممكن احط اسم cells عندى فيبقى اى input ports لل cells دى واصلة على VDD & VSS هتوصل على tie high & tie low cells ... ويمكن احط port معين يعنى هحدد input port لل cell واقول ده الى هيوصل عليهم على حسب هو واصل على VDD ولا VSS هيوصل على tie high او tie low
- **obj_type**: ده option لازم بيقى موجود ... هحط فيه نوع ال elements الى انا حطيتها فى object هل حطيت ports فاستخدم هنا port_inst ولا حطيت اسماء instant of cells فاستخدم cell_inst ولا حطيت اسم cell بشكل عام من ال library فاستخدم lib_cell ... ناخذ بالناسم ال اسم ال cell فى library واحد بس انا بستخدمها كتير فى ال design فكل مرة استخدمها فيها بتبقى instant عندى وله اسم يميزه عن باقى ال instances الى جاين من نفس ال library cell
- **tie_high_lib_cell**: هحط فيه اسم ال cell الى هستخدمها ك tie high cell فلانم استخدم معاه tie_low_lib_cell option
- **tie_low_lib_cell**: هحط فيه اسم ال cell الى هستخدمها ك tie low cell فلانم استخدم معاه tie_high_lib_cell option
- **tie_highlow_lib_cell**: هحط فيه اسم ال cell الى هستخدمها ك tie high cell & tie low cell يعنى هى cell واحدة بيقى فيه port فيها يمثل logic high و port تانى يمثل logic low يعنى مش output واحد وبالتالي لازم استخدم معاه options تانى وهما tie_high_port_name & tie_low_port_name
- **tie_high_port_name**: هكتب فيه اسم ال port الى بيمثل logic high فى ال cell الى اختارتها فى tie_highlow_lib_cell
- **tie_low_port_name**: هكتب فيه اسم ال port الى بيمثل logic low فى ال cell الى اختارتها فى tie_highlow_lib_cell
- **max_fanout**: هحط اكبر عدد من ال ports يتوصل على ال tie high \ tie low cell يعنى ال cell الواحدة ال output بتاعها يوصل على كام port بمعنى اخر ت drive كام cell فى ال design
- **max_wirelength**: هحط ال max length لطول ال wire الى هيخرج من tie high \ tie low cell يوصل لل input port الى هى المفروض توصل عليه ... لو محطتش هياخد قيمة default
- **incremental**: فى ال mode ده بخلى ال tool تزود على الموجود بمعنى انه لو موجود tie cells اصلا قبل كده محققة ال max_fanout & max_wirelength constraints فهتسيبهم ... لكن ال default انه يكون false بحيث ان ال tool بتتشيل اى tie cells موجودة وتحط من جديد

```

EXAMPLE
In the following example, all input ports of the BONUS_SET_*/ cells
that are tied off to power or ground are connected to new tie-high and
tie-low cells.

prompt> connect tie_cells -objects [get_cells "BONUS_SET_*/"] \
-obj_type cell_inst \
-tie_high_lib_cell TIEH -tie_low_lib_cell TIEL

In the following example, all input ports named A of the BONUS_SET_*/
cells that are tied off to power or ground are connected to new or
existing tie-high and tie-low cells that are at most 64 microns of Man-
hattan distance away.

prompt> connect tie_cells -objects {"BONUS_SET_*/A"} \
-obj_type port_inst \
-tie_high_lib_cell TIEH -tie_low_lib_cell TIEL \
-max_wirelength 64 -incremental true

```

```
In the following example, all input ports of the BONUS_SET */* cells
that are tied off to power or ground are connected to new tie-highlow
cells. Ports that are tied off to power are connected to the OUT1 port
on the TIEHL cell. Ports that are connected to ground are connected to
the OUT0 port on the TIEHL cell.
```

```
prompt> connect tie_cells -objects [get_cells "BONUS_SET */*"] \
        -obj_type cell inst \
        -tie_highlow_lib_cell TIEHL \
        -tie_high_port_name OUT1 \
        -tie_low_port_name OUT0
```

- لما كتب في `get_cells` اسم `cell` حط بعدها `*/*` فدى معناها انه يقصد اى `cell` ببدا اسمها ب `BONUS_SET` يعنى ممكن مثلا في واحدة اسمها `BONUS_SET_A` وواحدة تانية `BONUS_SET_B` وهكذا يعنى مقصود بيها انه بي `match` اى `pattern` اوله `BONUS_SET`

- لما بدل `cell` حب يتعامل مع `port` فراح كتب نفس الشكل بتاع ال `cell` وزود `"/"` ثم اسم ال `port` وهنا `A` ... ال/ عامة بتعبر عن `path` فانا بقوله اى `cells` عندك اولها `BONUS_SET` انا عايز ال `port` الى اسمه `A` فيها

=====

- Decap cell insertion:

Command:

```
set_rail_options [-analysis_mode static | dynamic] [-decap_lib_cells lib_cell_list] \
[-filler_lib_cells lib_cell_list] [-vd_threshold value]
```

- زى ما قلنا احنا هنبدل بعض ال `filler cells` ب `decap cells` فانا محتاج اعمل `analysis` الاول واشوف ايه الاماكن الى المفروض ابدل فيها فالاول اقول لل `tool` شوية `options` كده ك `guide` ليها قبل ما اقولها تعمل `analysis`:
- `analysis_mode`: بحدلها نوع ال `analysis` فانا عايز اشوف `voltage drop` يبقى هعمل `dynamic analysis` ... فهنحط ال `value` لل `option` ده `dynamic`
- `filler_lib_cells`: زى ما قولنا ال `tool` لما تحط ال `Decap cells` فهي هتحطهم مكان شوية `filler cells` موجودين ... فلو عايز احدد `filler cells` معينة الى تتشال بحط اسمائها الموجود فى `std library` هنا بحيث يبقى دول الى تقدر تبدلهم ب `decap cells`
- `decap_lib_cells`: لو عايز ابدل ب `decap cells` معينة بحط اسمائها من ال `std library` هنا
- `vd_threshold`: نحدد ال `target voltage drop threshold` وال `unit` هي المستخدمة فى `library` وممكن تعرفها من `report_unit` command

=====

Command:

```
analyze_rail nets [-decap]
```

- في الخطوة دي ال `tool` بتختار شوية `filler cells` موجودة وبتبدلها بشكل `virtual` ب `decap cells` وبتحاول تقلل ال `area` وال `leakage current` وفي نفس الوقت تحقق ال `voltage drop reduction target` بس هي مش هتحطهم فعليا هي بس هتجرب كذا `iteration` ... بعد كده هتلاقيها عملت `generation` لكذا `ECO file` اسمهم `decap_ude_n` وال `n` دي هي رقم ال `analysis`

iteration ... الفايل ده بيكون مكتوب فيه اسماء ال filler cells اللي هتتشال واسماء ال Decap cells اللي هتتحت مكانها ... ال options المهمين بيها هي:

- **nets** : اسماء ال nets الى هعمل عليها analysis الى هي دلوقتى بالنسبالنا ال power nets ... بدل كلمة nets بحط أسمائهم على طول مش بعد nets يعنى ... لو هحط كذا net بحطهم فى list يعنى بين { }

- **decap** : ده option عشان ال tool تفهم انى بعمل dynamic analysis عشان احط decap cells فتطلعلى ال ECO files ال قولنا عليها

مثال بقى لل **set_rail_options & analyze_rail** مع بعض:

```
For example, set the following rail options if you want to perform decoupling capacitor insertion on the VDD net of a cell with filler cells named FillerCell1 and FillerCell2 and decoupling capacitor cells named DecapCell1 and DecapCell2 with a voltage threshold target above 1.5V (for a 1.8V power supply):
set_rail_options
  -filler_lib_cells FillerCell1 FillerCell2
  -decap_lib_cells decapCell1 DecapCell2
  -vd_threshold 1.5

Then, run the analyze_rail command with the -decap option on the VDD net:
analyze_rail VDD -decap
```

=====

Command:

source path_to_ECO_file/.decap_ude_n

ده مش command مخصوص لل ICC ده source command عادى فى TCL بخليه ي source واحد من ECO files الى طلعته عشان بقى فعليا ابدل ال filler cells ب decap cells

=====

- End cap cell insertion:

Command:

add_end_cap -lib_cell lib_cell_name [-vertical_cells lib_cell_names] [-fill_corner] \ [-mode both | bottom_left | upper_right] [-mirror] [-respect_blockage] [-respect_keepout] \ [-ignore_soft_blockage]

ال command ده بيضيف endcap cells بس horizontally ده ال default فانت عشان تحطها vertically او يعنى فى ال upper وال bottom بتاع ال chip محتاج تستخدم options فيه تقوله كده زى ما هنشوف:

- **lib_cell** : ال option ده عشان نحدد ال std cell اللي هنستعملها ك horizontal end cap cell ... ونقدر نختار cell واحدة بس وده option لازم تحطه لان مفيش cell فى std library محددة ك endcap cell يعنى اى cell ممكن تحطها فلازم تقوله هيحط انى واحدة من الى عنده فى library

- **vertical_cells**: ال option ده علشان نحدد ال std cell اللي هنستعملها ك vertical end cap cell وناخد بالنا ان by default ال tool مش هتخط vertically الا اما اقولها
- **fill_corner**: ال option ده بيخلينا نخط end cap cells في ال corners بإستعمال ال horizontal وال vertical end cap cells وبالتالي لازم نكون محددين ال horizontal وال vertical end cap cells بإستعمال ال two commands اللي فاتوا
- **mode**: بحدد هخط ال endcap cells فين ... لو **both** فده ال default وبيخط end cap cells علي ال both ends بتوع ال row ... ال left cap بيكون علي normal orientation الى هو شرحناه فوق وكان left endcap cell كان two dummy poly على الشمال وال right بيكون flipped orientation يعنى عكس ال left ... ال **bottom_left** بيخط end cap عند ال bottom بتاع vertical row او علي ال left بتاع horizontal row ... ال **upper_right** بيخط end cap عند ال top بتاع vertical row او علي ال right بتاع horizontal row
- **mirror**: ده بيخلي ال orientation بتاع ال end cap cell يكون flipped ... ال normal orientation زى ما قلنا الى هو بتاع ال left endcap cell ... ال option ده ملهوش تأثير لو انا حاطط value ب both لل option mode لانه بيعمل by default ال left يكون normal orientation وال right يكون flipped orientation
- **respect_blockage**: ده بيخلي ال tool متحطش end cap cells في ال placement blockages ... ال default هو ان ال tool بتخط ال end cap cells في ال placement blockages الا لو انت حطيت ال option ده
- **respect_keepout**: ده بيخلي ال tool متحطش end cap cells في ال keepout margins حوالين ال macros
- **ignore_soft_blockage**: ده بيخلي ال tool تحط endcap cells عادى فى ال soft blockages بس متحطش فى ال hard blockage وبالتالي لازم تكون مستخدم معاه **respect_blockage**

EXAMPLES

The following example adds horizontal end cap cells named MY_END_CAP to the current design.

```
prompt> add_end_cap -lib_cell MY_END_CAP
```

=====

- Antenna Fixing

Command:

verify_zrt_route [-antenna true | false]

ده اتكلمنا عنه فى ال routing انه بيعمل check على DRCs violations و open nets و antenna violations ... وكان فيه options تانية ذكرناها بس هنا مركزين نفكر بتاع ال antenna:

- **antenna**: لو true هيعمل check على ال antenna violations وال default هو true

=====

Command:

```
set_route_zrt_detail_options [-antenna true | false]\
[-antenna_fixing_preference hop_layers | use_diodes] [-check_antenna_on_pg true | false]\
[-diode_insertion_mode new_and_spare | spare | new] [-diode_libcell_names lib_cells]\
[-insert_diodes_during_routing true | false] [-hop_layers_to_fix_antenna true | false]
```

ال command ده ممكن تستخدمه عشان تحط options لل detailed routing عامة الى اتكلمنا عنه قبل كده فى ال routing بس احنا استخدمنا commands تانية ... ال command ده فيه options كثير جدا بس احنا هنهتم نستخدمه فى اننا نحط ال options الى ليها علاقة بال antenna violation عشان نصلحه بعد كده فهنذكر فقط ال options دى:

- **antenna**: يعمل antenna analysis اصلا ولا لا ... فلو عايز هخليها true وال default هو true
 - **antenna_fixing_preference**: بحدله هستخدم انى technique عشان يحل ال violation ... لو حطيت hop_layer هيعمل ال layer jump الى شرحناه ولو حطيت use diodes هيحط ال diodes زى ما شرحنا ... وال default هو hop_layer
 - **check_antenna_on_pg**: بقوله هل يعمل check لل antenna violation على ال power & ground nets كمان ولا لا فلو عايز بخليها true بس ال default هو false
 - **diode_insertion_mode**: لو هستخدم diodes فهل يحط diodes جديدة من عنده فهختار new ولا انا عندى diodes حاططهم ك spare cells فيستخدمهم وساعتها اختار spare او اخليه يستخدم الاتنين فاختار new_and_spare وال default هو new_and_spare يعنى يستخدم الاتنين
 - **diode_libcell_names**: بحدد أسماء ال diodes فى ال standard cell library الى يستخدمهم فى ال antenna fixing وال default هى empty list يعنى ال tool بتحدد هى المناسب وتحطه الا لو انا حددتلها diodes معينة من ال option ده
 - **insert_diodes_during_routing**: بقوله يقدر يستخدم diodes فى تصليح ال antenna violations ولا لا وال default هو false يعنى ميسخدمش
 - **hop_layers_to_fix_antenna**: بقوله يقدر يستخدم layer jump technique فى تصليح ال antenna violations ولا لا وال default هو true يعنى يستخدمها
- لو استخدمت **antenna_fixing_preference** هتحدد هتستخدم layer jump ولا diode insertion هتحدد واحد فيهم لكن تقدر تستخدم الاتنين من خلال انك متستخدمش ال option ده وتخلي **insert_diodes_during_routing & hop_layers_to_fix_antenna** الاتنين ب true

=====

Command:

```
route_zrt_detail
```

ال command ده بيعمل detailed routing حسب ال option الى حددتها فوق فاحنا هنا هنستخدمه كده زى ما هو بحيث هيرجع يصلح بقى اى antenna violation بانه يغير ال detailed routing ويعمل layer jump مثلا او يحط diode حسب ال options الى حددناها فوق ... ال options بتاعته مش مهمة بالنسبالنا هنا

=====

- 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 المرحلة دى اعمل save ل MW cell جديدة الى هى ال design بتاعى لحد الخطوة دى واديهها اى اسم.

=====

بكده نكون خلصنا شرح الجزء ده والمفروض انه قبل ما نعمل signoff محتاجين نعمل checks كتير على ال chip ولكن مش باستخدام ICC لإنها زى ما اتكلمنا قبل كده بتشوف ال FRAM view الى هو ال abstract يعنى تفاصيل ال poly وال transistors نفسها مش بتأخذها فى اعتبارها وبالتالي بنعمل signoff checks على ال functionality و timing و physically وده الى هنتكلم عنه بعد كده

=====