



Conception d'un bloc numérique du code au layout



Par J. NANNI le 16 Juillet 2014

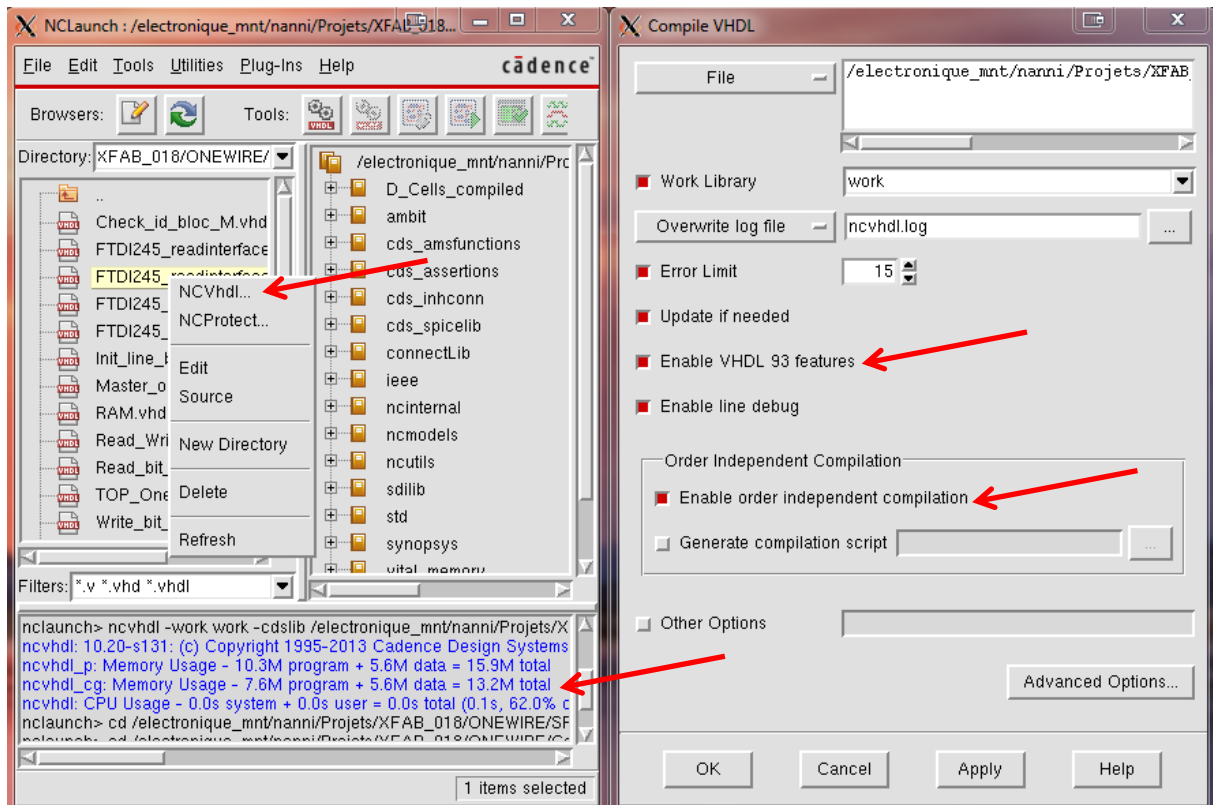
1 Simulation comportementale du bloc: (outils Cadence NCLaunch)

Pour valider le bon fonctionnement d'un bloc, il faut passer par 3 étapes :

- ✓ Compilation (vérification de la syntaxe),
- ✓ Elaboration (pour le testbench)
- ✓ Simulation

1.1 Compilation :

Sélectionnez le répertoire des sources /path/Projet/SRC/hdl/, puis sélectionnez les sources à compiler, clic droit puis NCVhdl.



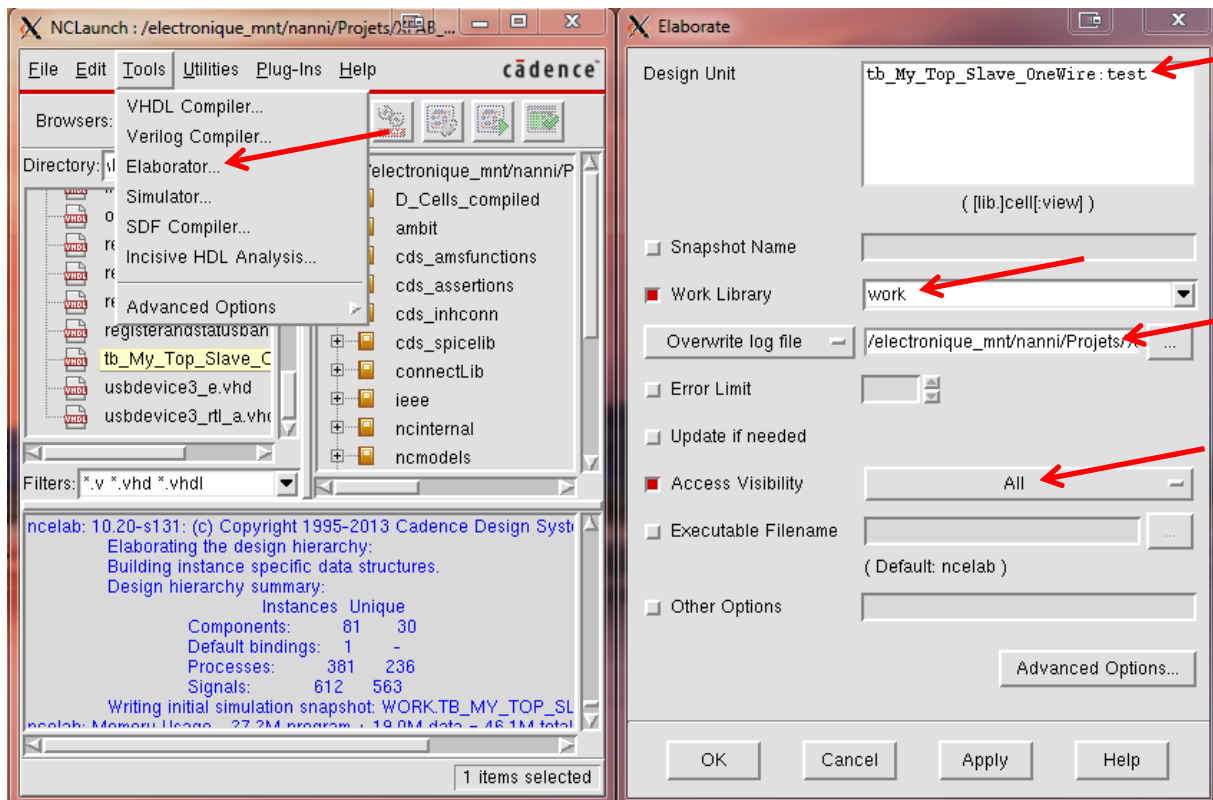
Il est préférable d'activer le mode VHDL 93 et la compilation par ordre de dépendance des sources.

Si tout se passe bien, il devrait apparaître **en bleu** dans le terminal de NCLaunch, l'utilisation de la mémoire et le temps de compilation. Dans le cas contraire, il apparaîtra **en rouge** les lignes dans les sources qui comportent des erreurs de syntaxes.

1.2 Elaboration :

L'objectif de l'élaboration est de créer des modules exécutables à partir des modules compilés précédemment.

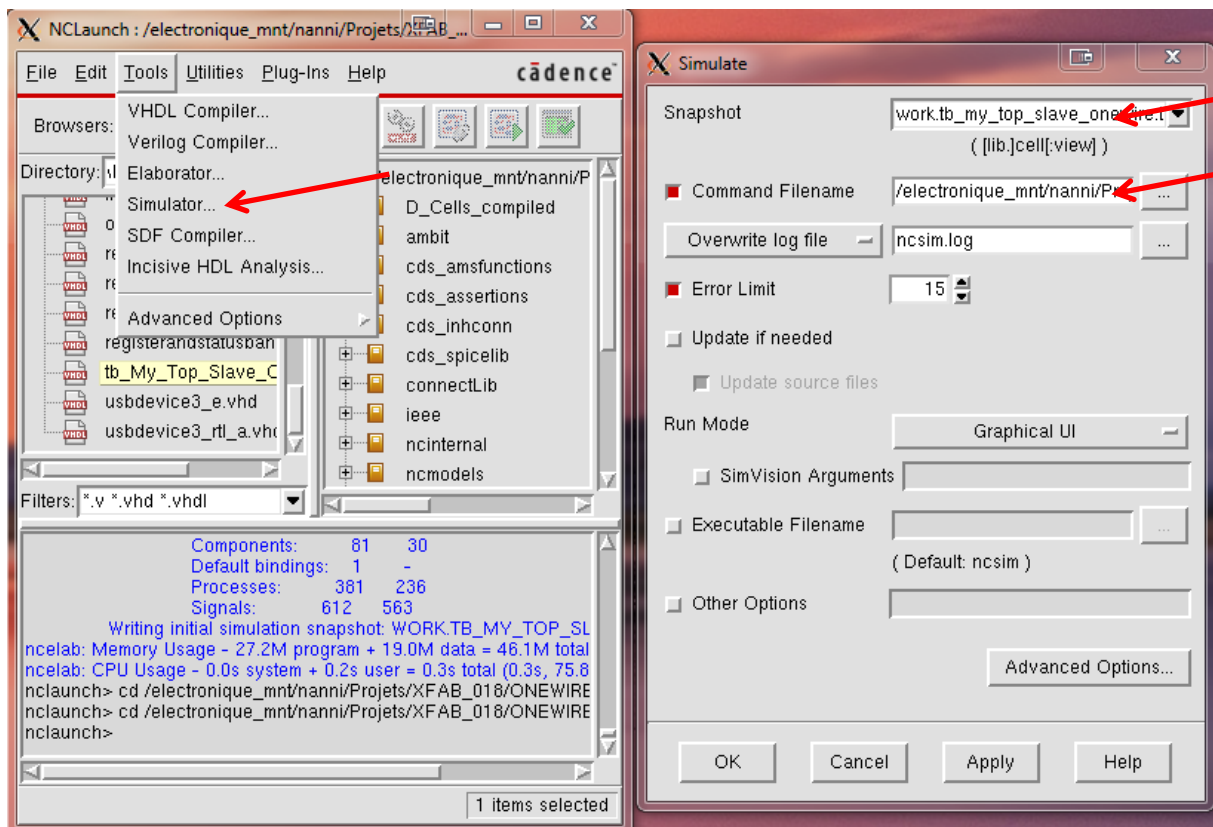
Sélectionnez le répertoire du testbench, cliquez sur **Tools** puis **Elaborator**. La fenêtre Elaborate s'ouvre, remplissez le champ Design Unit par : **library.tb_name:view**. Sélectionnez la librairie de travail (work par défaut), sélectionnez le fichier log (afin de ne pas avoir de warning) et ajoutez l'option **All** dans l'onglet **Access Visibility**.



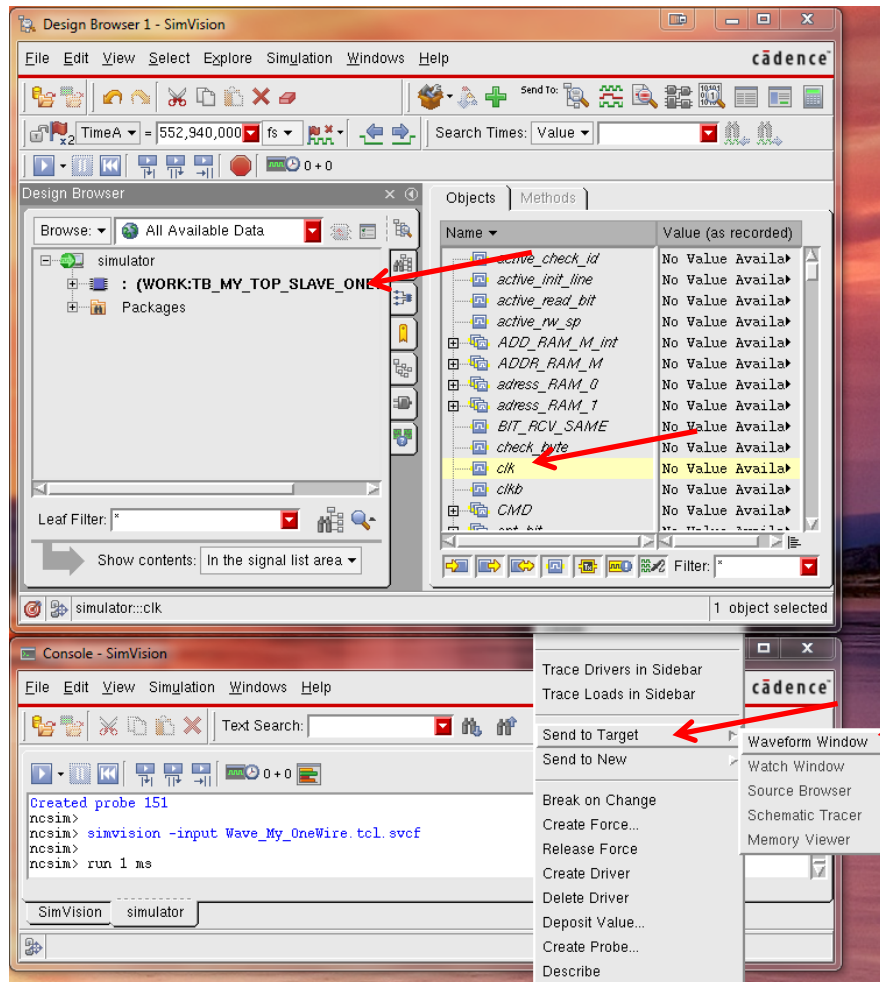
Si tout se passe correctement, il devrait apparaître **en bleu** un résumé du nombre de composants, de process et de signaux.

1.3 Simulation :

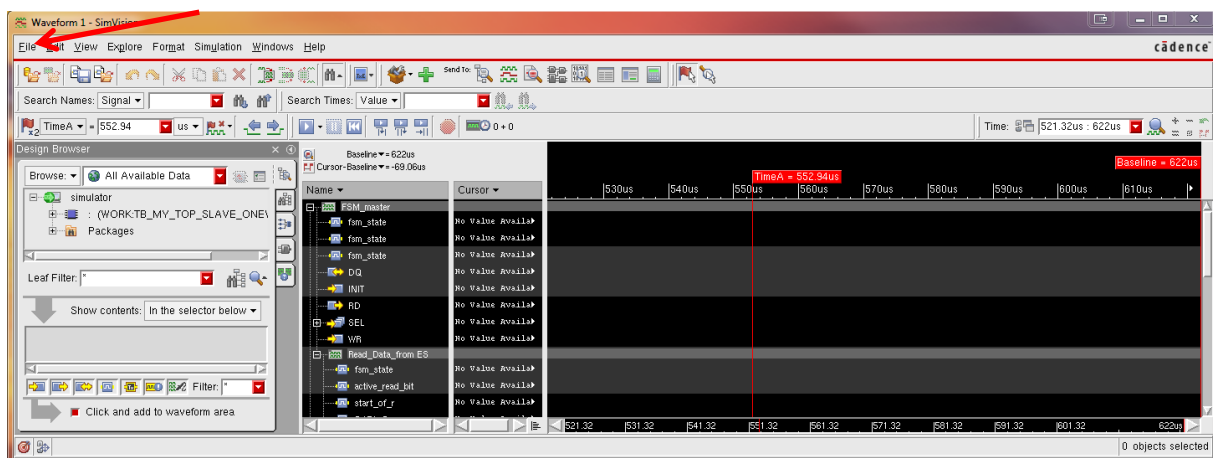
Pour cette dernière étape, sélectionnez **Tools**, puis **Simulator**. Sur la seconde fenêtre sélectionnez le snapshot (la vue à simuler soit **library.tb_name:view**).



La fenêtre Design Browser s'ouvre, cliquez sur votre Snapshot puis fait un clic droit sur un signal, puis **Send to Target** et **Waveform Window**.



La fenêtre ci-dessous apparaît, pensez à sauvegarder vos signaux au format **.tcl** en cliquant sur **File, Save Command Script**. Ainsi, à chaque simulation, en remplissant le champ **Command Filename** (dans la fenêtre Simulate) avec le fichier .tcl précédent, l'ensemble de vos signaux se chargera automatiquement.



Une fois la phase de simulation comportement validée, il est temps de passer à la phase de synthèse.

2. Synthèse du code source : (outils Cadence RC)

Le principe de la synthèse est de convertir le code source vers une description basé sur la techno utilisée. Pour ce faire, il faut 3 scripts :

- ✓ ScriptSetup : décrit le path des sources, les paths de la techno (.lib, .lef) et nom du design
- ✓ ScriptConstraints : pour définir les contraintes du design
- ✓ ScriptTop : permet la lecture du code, l'élaboration, puis le type de synthèse.

2.1 ScriptSetup :

Les champs à modifier sont les suivants :

set SOURCES_PATH : le path des sources,
set LIBRARY : le nom des librairies,
set LEF_LIBRARY : le nom des fichiers .lef,
set TECH_LIBRARY_PATH : le path des librairies et des .lef,
set WL_SELECT
set DESIGN : le nom du design

2.2 ScriptConstraints:

C'est à l'utilisateur d'indiquer ses propres contraintes.

2.3 ScriptTop:

Les champs à modifier sont les suivants :

Read_sdc: indiquez le nom du script (par défaut ScriptSetup.sdc),
Synthesize: choix entre to_mapped, to_generic, option (incr / no_incr)

2.4 Execution du script:

Dans le terminal, tapez : **rc -file ScriptTop.tcl**

Le script prend 2-3 minutes à s'exécuter et génère 2 fichiers :

DESIGN .v
DESIGN .sdc

2.5 Vérifications:

Il est important de vérifier différents points tel que:

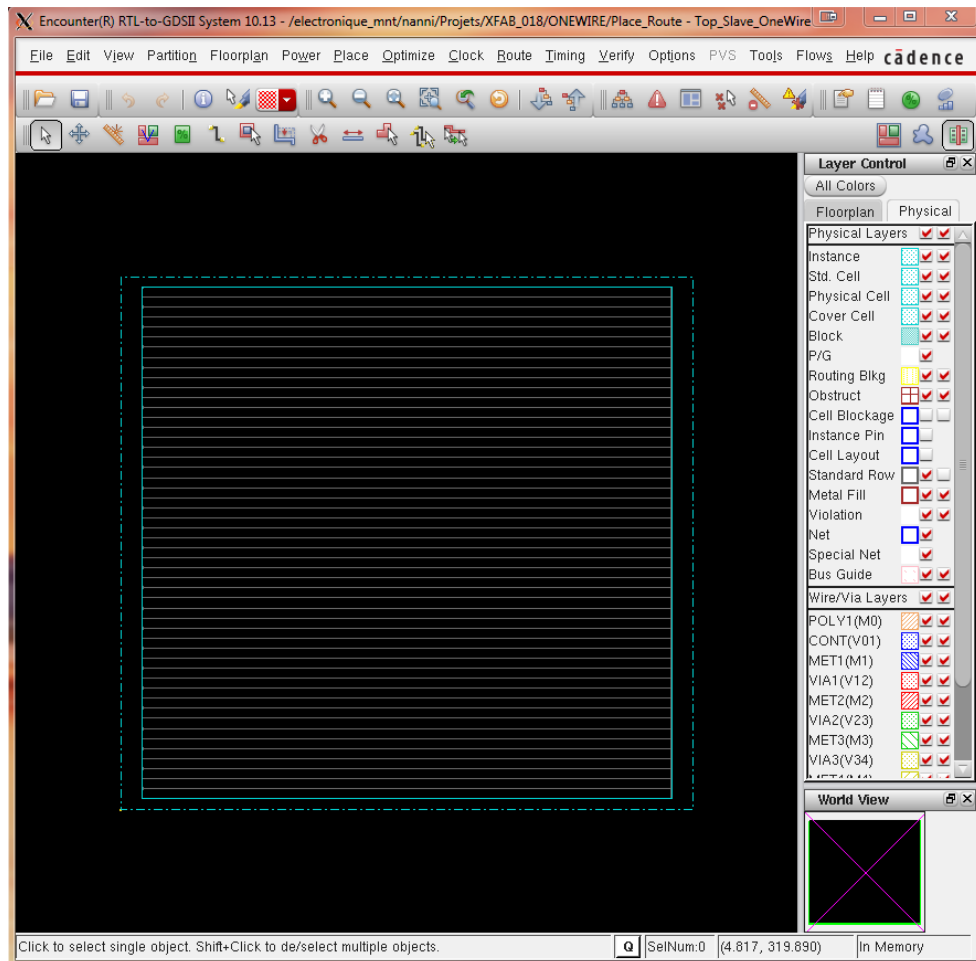
report message -all: pour voir tous les warnings ou erreurs à la synthèse
report timing: il faut que le slack soit positif (estimation des timings),
report area.

3. Placement Routage (outils Cadence Encounter)

Dans le terminal, tapez : **encounter -config config_encounter.config**, la fenêtre ci-dessous s'ouvre sous cette forme.

Le fichier de configuration permet de renseigner les champs suivants :

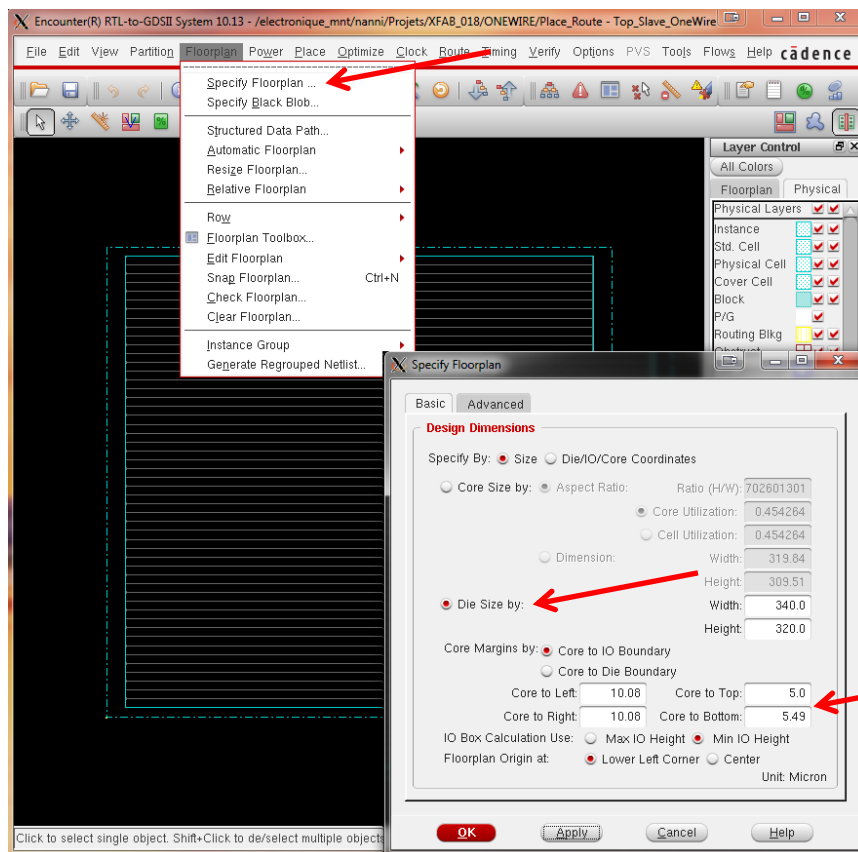
```
set cwd : répertoire de l'exécution d'Encounter
set rda_Input(ui_netlist): path + nom du fichier Verilog (DESIGN_jourmois_heure_hdl.v)
set rda_Input(ui_topcell): nom du DESIGN
set rda_Input(ui_timelib): path+nom du fichier .lib qui décrit les timing de la techno
set rda_Input(ui_timingcon_file): path+nom des contraintes de timings
set rda_Input(ui_oa_reflib): nom des bibliothèques où chercher les instances numériques
set rda_Input(ui_oa_abstractname): définir le nom de la vue des instances numériques (abstract)
set rda_Input(ui_oa_layoutname): définir le nom de la vue des instances numériques (layout)
set rda_Input(ui_pwrnet): pour définir le nom de l'alimentation positive (vdd!)
set rda_Input(ui_gndnet): pour définir le nom de la masse (gnd!)
```



Les étapes de conceptions suivantes sont :

- ✓ Définir la taille du floorplan,
- ✓ Fixer la position des I/O,
- ✓ Placer l'alimentation,
- ✓ Placer les cellules du design,
- ✓ Générer un arbre d'horloge,
- ✓ Faire une optimisation,
- ✓ Insérer les cellules de remplissages,
- ✓ Faire le routage,
- ✓ Faire une optimisation.

3.1 Floorplan :



Pour régler la taille totale du bloc, il y a 2 options :

En définissant la taille du cœur → **Core Size By**

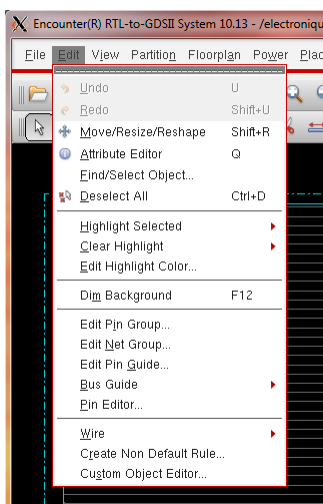
En définissant l'extérieur du bloc (alim. + core) → **Die Size By**

Il faut définir la place nécessaire pour faire passer les alimentations : **Core Margins**

L'idéal étant d'avoir une utilisation de l'espace 90%, pour que l'outil puisse router. Sinon, il faut augmenter la taille du cœur, pour avoir un cœur plus aéré et avoir une complexité de routage plus faible.

Par exemple, un circuit de taille 340x320 μm^2 avec des marges de 10 μm et 5 μm , permet d'obtenir un circuit de 320x310 μm^2 .

3.2 Pin editor :

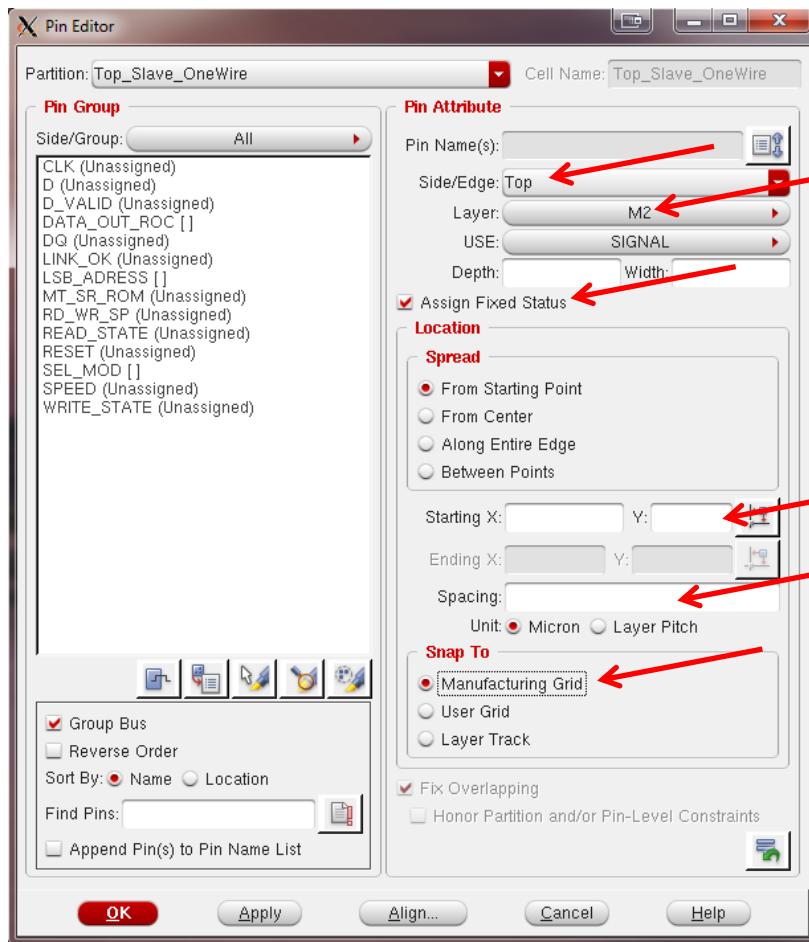


Cliquez sur **Edit**, puis **Pin Editor**.

La fenêtre ci-dessous d'ouvre, il faut ensuite configurer toutes les pins.

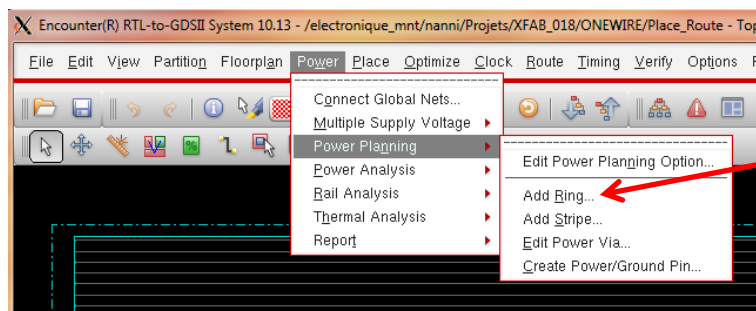
Sélectionnez :

- ✓ Le niveau de métal (M1 gauche/droite et M2 haut/bas),
- ✓ **Toujours cocher Assign Fixes Status pour tous les pins** (cela évitera que les pins bougent de place à chaque routage),
- ✓ Indiquez la position en X, Y et le spacing pour un vecteur
- ✓ **Toujours cocher Manufacturing Grid** (cela évitera qu'à insertion du bloc dans le top, il y a un décalage lors du routage).



3.3 Power Planning :

Cliquez sur **Power**, puis **Power Planning**, puis **Add Ring**(privilegié) ou Add Strip.

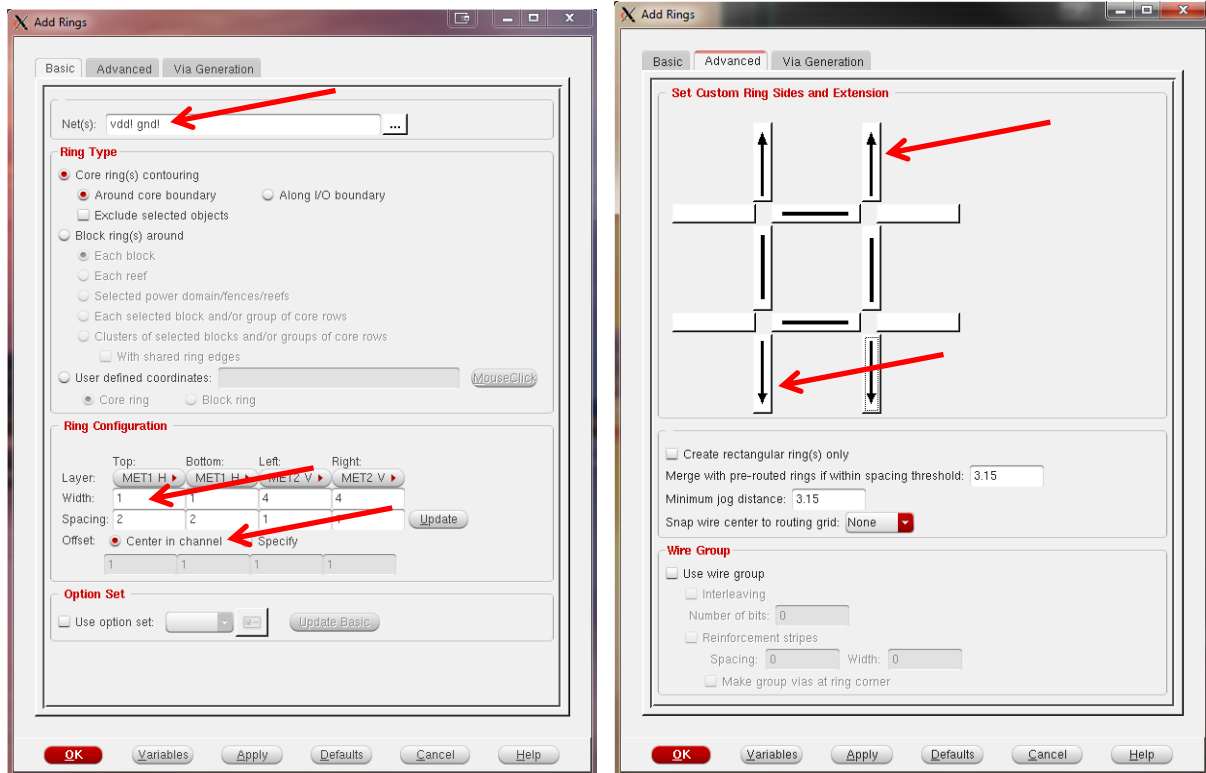


Prenons l'exemple de la création d'un anneau d'alimentation (**Add Ring**).

Il vous faut :

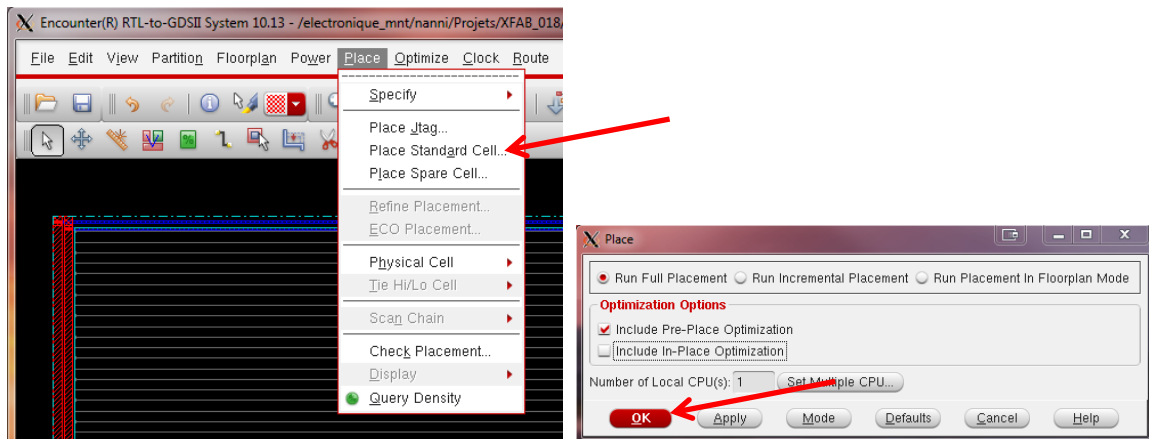
- ✓ Renseigner les alimentations (par défaut **vdd!** et **gnd!**)
- ✓ Indiquez la taille et l'espacement des rails d'alimentation (dépendant des champs **Core to left/right ...**) renseignés à l'étape **Floorplan**.
- ✓ Pour l'**Offset**, sélectionnez **Center in Channel** (pour plus de simplicité)

Pour placer des pins sur les rails d'alimentations, cliquez sur l'onglet **Advanced** et cliquez sur les rectangle extérieur afin d'obtenir des flèches qui correspondent aux pins.



3.4 Place Standard Cell:

Cliquez sur **Place**, puis **Place Standard Cell**, puis OK sur la fenêtre suivante.

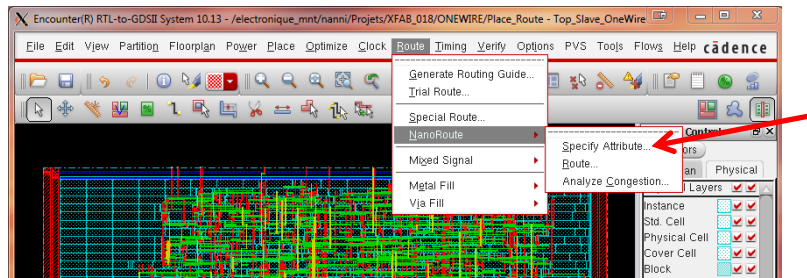


Le nombre d'itération et le temps de placement routage seront fonction du champ **Cell Utilization** dans l'étape du **Floorplan**.

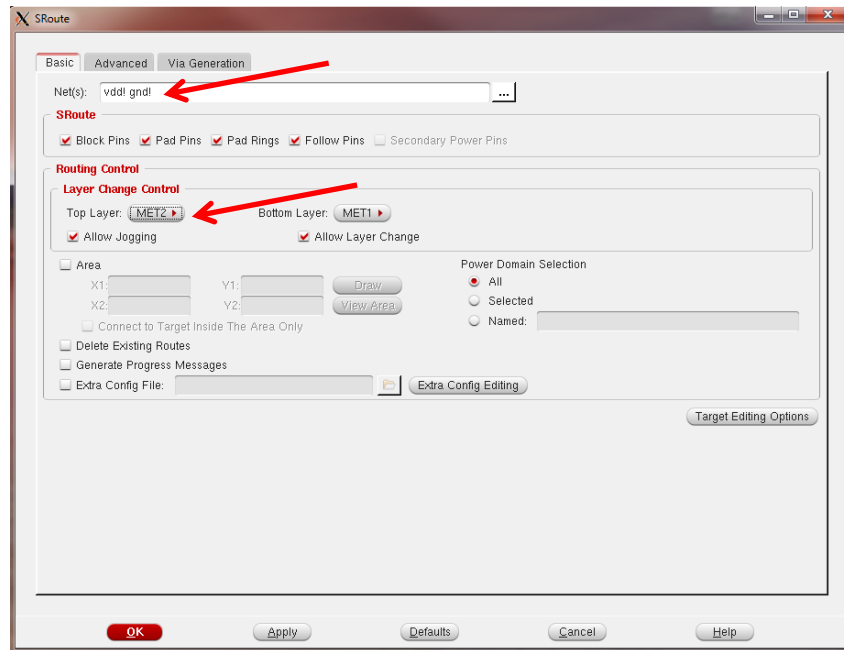
3.5 Special Route:

La première chose à faire est placer les rails d'alimentations qui traversent les cellules et relient les anneaux d'alimentations.

Cliquez sur **Route** → **Special Route**



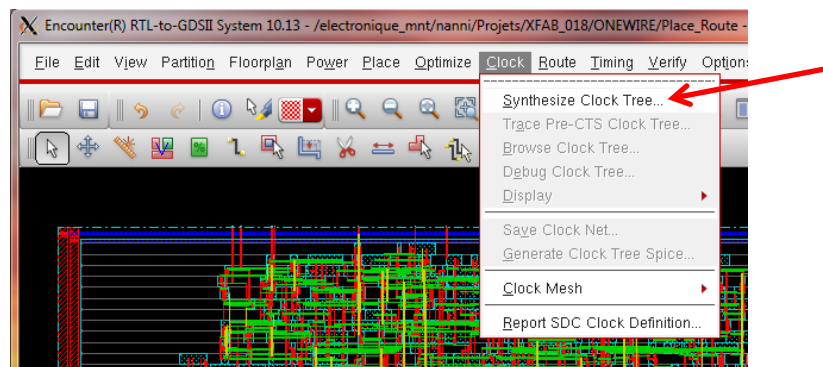
Renseigner les alimentations (par défaut **vdd!** et **gnd!**) et fixez le **Top Layer** au **MET2**.



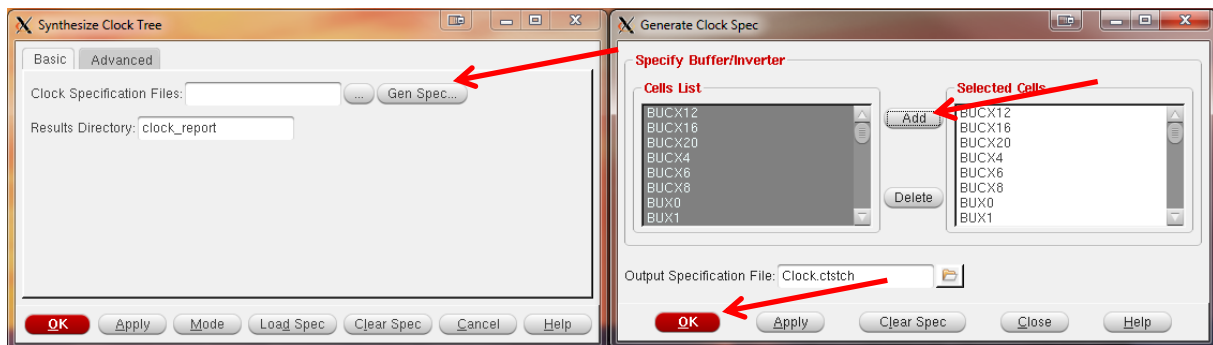
3.6 Synthesize Clock Tree :

Cette étape est très importante dans le cas d'un circuit synchrone évidemment. En effet, cela permet d'avoir un délais ou retard égal entre toutes les bascules.

Cliquez sur Clock → Synthesize Clock Tree.



Il faut maintenant générer le fichier de spécification de l'horloge, afin d'indiquer les délais max, les temps de Setup et de Hold.

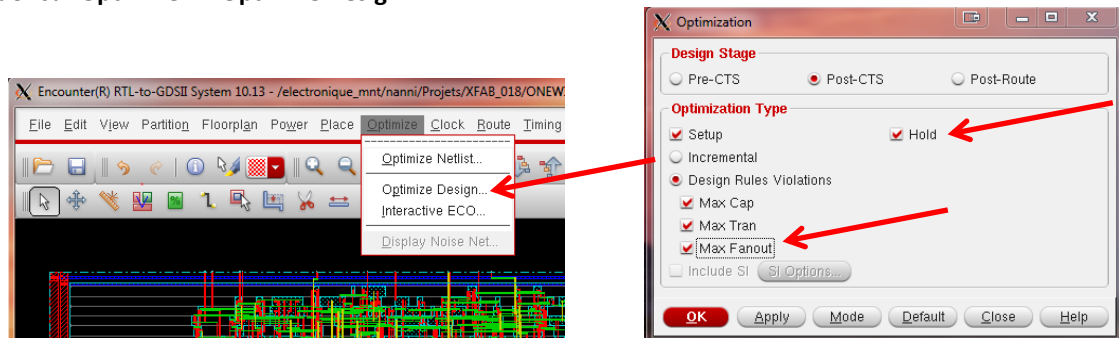


Cliquez sur **Gen Spec ...**

Sélectionnez l'ensemble de la Cells list, cliquez sur **Add** puis **OK** dans la fenêtre **Generate Clock Spec**.
Puis cliquez OK, dans la fenêtre **Synthesize Clock Tree**.

3.7 Optimization :

Cliquez sur **Optimize → Optimize Design ...**



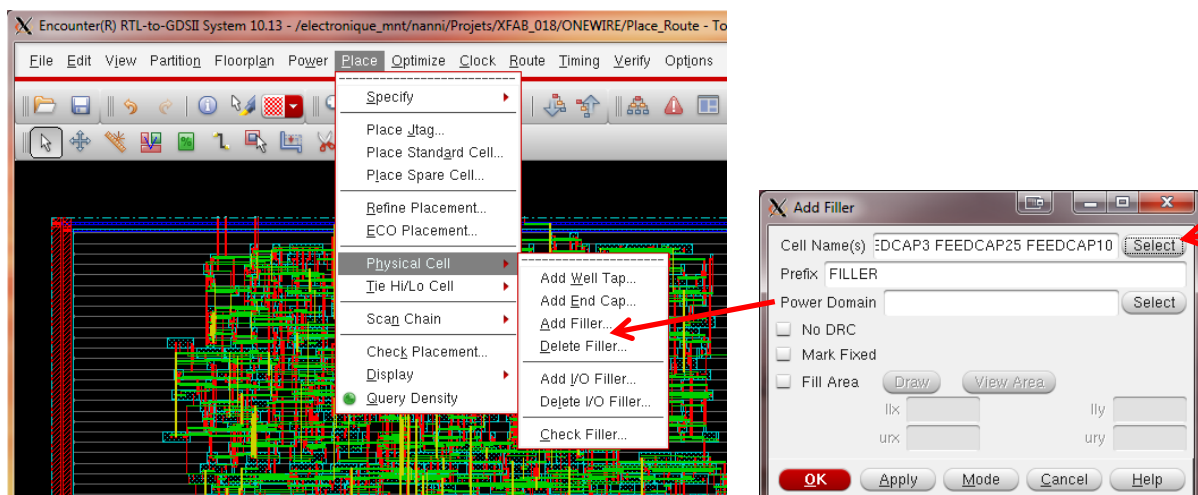
Sélectionnez **Post-CTS**, ajoutez **Hold** et **Max Fanout** et cliquez sur **OK** de la fenêtre Optimization.

3.8 Filler (Cellules de remplissage) :

Dans l'étape de **Floorplan**, nous avons décidé d'avoir un taux de remplissage le plus grand possible mais pas de 100%. Il faut donc remplir les espaces vides avec cellules appelés **Filler** (capacitive ou non).

En fonction de la technologie utilisée, il est possible (voir nécessaire) de placer également des **End Cap**, cellules qui entourent le cœur.

Cliquez sur **Place → Physical Cell → Add Filler**

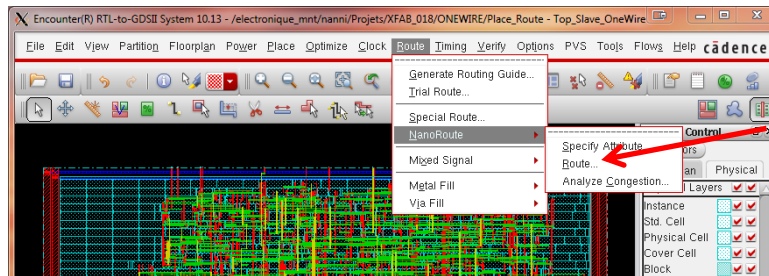


Cliquez sur **Select**, puis choisissez les cellules adéquates (**FILLER** ou FEEDCAP ou FEED), puis OK.

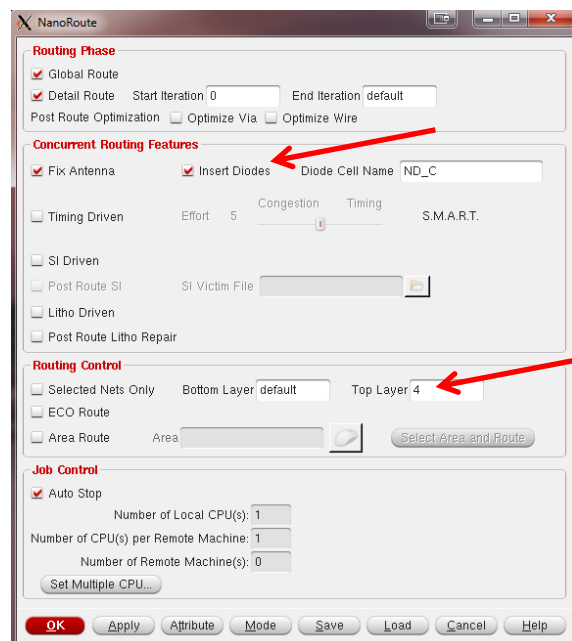
3.9 Nano Route :

Cette étape permet de faire le routage complet du cœur du circuit.

Cliquez sur **Route** → **NanoRoute** → **Route ...**



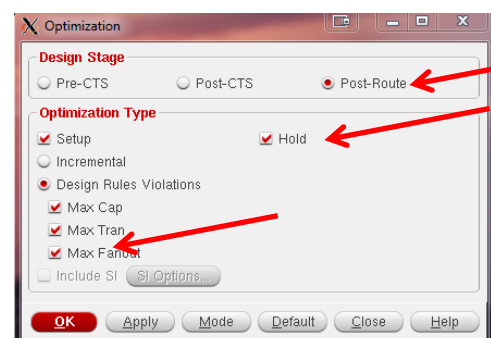
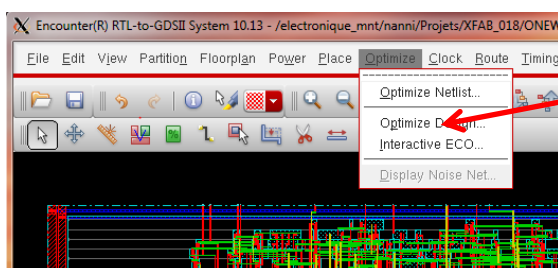
Renseigner le niveau de métal le plus élevé, dans le cas de la technologie XFAB 018 : MET 4 → 4



Vous pouvez également insérer des diodes pour éviter des problèmes de règles de dessins suites à des fils trop long.

3.10 Optimization :

Cliquez sur **Optimize** → **Optimize Design ...**



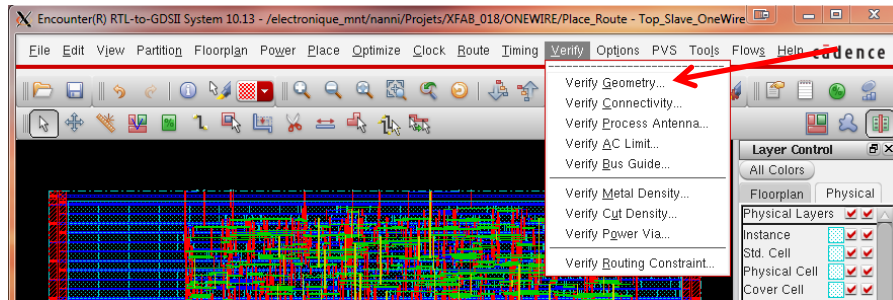
Sélectionnez **Post-Route**, ajoutez **Hold** et **Max Fanout** et cliquez sur **OK** de la fenêtre Optimization.

3.11 Vérifications :

Cliquez sur **Verify** → **Verify Geometry ...**

Il est important de vérifier différents points tel que:

Geometry
Connectivity
Process Antenna



3.12 : Sauvegarde :

La dernière étape consiste à sauvegarder 5 vues :

- ✓ La vue encounter (.enc) pour les possibles modifications : File → Save Design, sélectionnez **Encounter**,
- ✓ La position des pins : File → Save → I/O File,
- ✓ Le floorplan : File → Save → Floorplan ,
- ✓ La vue OpenAccess (.oa) pour utiliser le layout dans Cadence IC : File → Save Design, sélectionnez OA,
- ✓ La netlist (.v) pour faire un import design dans Cadence IC et ainsi générer un schematic pour faire un LVS.

BRAVO, vous avez généréz votre premier layout avec Encounter.