

## PLACEMENT ET ROUTAGE CIBLES ASIC AVEC L'OUTIL INNOVUS DE CADENCE :

Innovus® est un outil de placement routage automatique des circuits numériques qui appartient à la suite d'outils Cadence. Il s'agit d'un outil puissant flexible et paramétrable mais également gourmand en ressource machine. Son utilisation est assez complexe et demande une certaine expertise pour tirer parti de toutes les possibilités. Il offre en effet des fonctionnalités et des capacités avancées.

Le format d'entrée d'Innovus est une liste de portes (netlist) au format Verilog. Un circuit complet est constitué d'un cœur (core), et d'une couronne de plots constitués d'amplificateurs (buffers) pour les connexions vers l'extérieur et de pastilles de connexion (pads). Les amplificateurs de connexion servent à adapter les niveaux de tension et de puissance entre le cœur et la carte sur laquelle sera soudé le circuit. L'ensemble cœur + amplificateurs de connexion + pastilles de connexion constitue la puce (chip = core + buffers + pads).

A l'aide de l'outil de synthèse Design Vision (cf.VI), vous effectuerez une synthèse du cœur du circuit de filtre pour des conditions de fonctionnement typiques et une fréquence d'horloge à déterminer. Il vous sera demandé de générer liste de portes (netlist) au format Verilog. Cette netlist sera importée dans l'outil Innovus afin de procéder au placement et routage de votre circuit. Afin de simplifier la rédaction de ce document, nous supposons que cette netlist est nommée **filtre\_synth\_plat.v**. Vous devrez ajuster adapter à vous de traduire ce nom par le nom que vous avez effectivement employé.

Pour des raisons de temps imparti, la description des plots d'entrée et de sortie et de leurs connexions avec le cœur a déjà été faite pour vous, car cette étape peut s'avérer assez chronophage. Ces fichiers seront étudiés dans la suite du document.

Un assez grand nombre de commandes est nécessaire pour placer et router totalement un circuit numérique complexe.

Le graphe de la figure donne une idée de la succession des opérations à effectuer. L'outil Innovus gère toutes ces étapes, offrant trois possibilités pour placer et router un circuit :

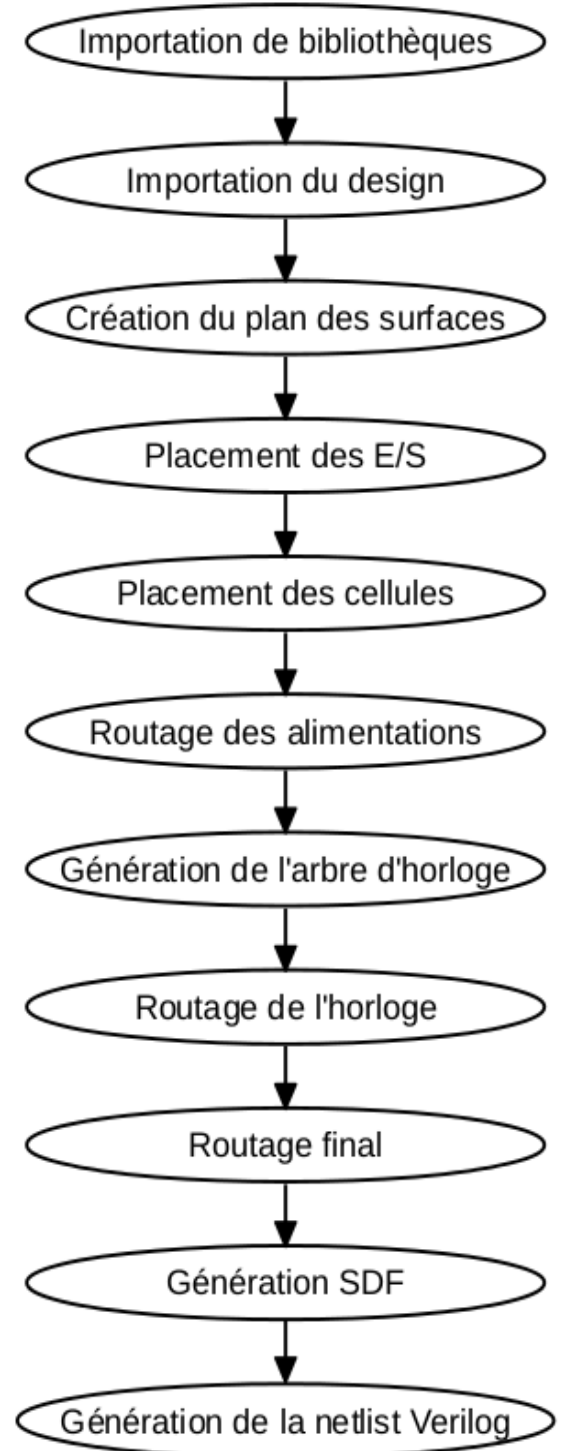


Figure 1 Etapes de placement et routage

- de façon interactive par les menus de l'outil.
- en entrant directement des commandes « innovus » dans la console de l'outil.
- en enchainant ces commandes dans un script.

Dans ce TP, vous utiliserez un mélange de ces possibilités, afin de rendre le TP faisable dans le temps imparti tout en concervant les aspects pédagogiques.

Pour cette partie du TP vous devez sourcer le fichier de configuration **config\_ASIC**. Ce script de configuration définit notamment les variables d'environnement **INNOVUS** et **AMS\_DIR**.

**Trouver de l'aide sur Innovus :**

- Manuel de l'interface graphique avec détails des menus  
\${INNOVUS}/doc/innovusMR/innovusMRTOC.html

### Travail préliminaire :

Avant de commencer votre projet Innovus pensez à appeler votre fichier synthétisé dans le fichier **filter\_io\_etudiants.v** décrivant la puce complète( coeur avec amplificateurs et plots de connexion) comme montré sur la figure suivante:

```
|`include "../filtre_synth_plat.v"

module filter_io ( CLK, RESET, Filter_In, Filter_Out, ADC_Eocb,
ADC_Convstb, ADC_Rdb, ADC_csb, DAC_WRb, DAC_csb, LDACb, CLRb );

    input [7:0] Filter_In;
    output [7:0] Filter_Out;
```

*Figure 2 Fichier Verilog du design*

### **ATTENTION :**

Il a été supposé lors de la création du fichier {TP\_PATH}/asic/VERILOG/filter\_io\_etudiants.v que le nom de l'entité du module englobant dans la description du coeur était FILTER (dans le fichier filtre\_synth\_plat.v). Si vous aviez choisi un autre nom vous devez modifier le fichier filter\_io\_etudiants.v pour assurer la concordance des noms.

Visualiser et comprendre le document filtre\_io\_etudiants.v sans le modifier !

## **1. Configuration d' Innovus :**

L'outil de placement et routage doit être réglé de façon à utiliser le kit technologique du fondeur (design kit) et prendre en compte les spécificités du projet. Ceci est fait au moyen d'un fichier script. Ce script garantit que la configuration initiale de l'outil est toujours la même à chaque fois qu'un travail est effectué sur le même circuit. En effet, le nombre de commandes à exécuter avant de commencer à travailler peut être conséquent ! D'où l'intérêt du script : **init.tcl**  
les commandes lancées depuis le script servent à :

- Spécifier les bibliothèques requises pour le placement et routage dans la technologie visée.
- Pour indiquer le nom du module englobant (*top cell*).
- Renseigner le nom du fichier verilog du design.

## 2. Démarrage de l'outil Innovus :

Dans un premier temps il faut démarrer l'outil Innovus :

Placez-vous dans le répertoire de travail `${TP_PATH}/asic/par/Innovus` et tapez la commande (sans '&'): **innovus**

## 3. Chargement des scripts du fondeur :

Le fondeur AMS mets gracieusement à disposition des scripts qui permettent de réaliser des opérations complexes dans Innovus. Certains de ces scripts seront utilisés au cours de ce TP. Ceux-ci doivent d'abord être chargés dans Innovus, ce qui peut se faire en exécutant la commande :

**source amsSetup.tcl**

dans la console depuis laquelle Innovus (celle dont le prompt a été remplacé par `innovus {n°}>`).

Cette opération n'est nécessaire que lors de l'importation d'un design (= création du projet de placement routage, voir 4). Lors de la restauration d'un projet commencé, ces scripts sont rechargés automatiquement s'ils avaient été chargé avant la sauvegarde.

Pour connaître la liste des scripts qui viennent d'être chargés, il suffit de taper dans cette même console, la commande : **amsHelp**

L'aide sur ces scripts peut être trouvée à l'URL suivante :

<http://asic.ams.com/hitkit/hk410/edi/flow.html>

## 4. Importation du Design:

Dans ce chapitre nous proposons deux méthodes d'importation de design ; graphique et par script.

### i. Méthode graphique

Pour pouvoir importer correctement le circuit dans Innovus, l'outil utilise le type de fichiers suivants :

- Library Exchange Format (LEF) : ce sont les bibliothèques contenant les différentes cellules utilisées dans le circuit ainsi que le fichier technologique.

`/softslin/AMS_410_CDS/cds/HK_C35/LEF/c35b4/c35b4.lef`

`/softslin/AMS_410_CDS/cds/HK_C35/LEF/c35b4/CORELIB.lef`

`/softslin/AMS_410_CDS/cds/HK_C35/LEF/c35b4/IOLIB_4M.lef`

- Timing Library (.lib) : ce sont les fichiers contenant les informations de timing pour chaque bibliothèque :

`/softslin/AMS_410_CDS/liberty/c35_3.3V/c35_CORELIB.lib`

`/softslin/AMS_410_CDS/liberty/c35_3.3V/c35_IOLIB_4M.lib`

- Netlists Verilog :
  - `netlist.v` : netlist du circuit obtenue après synthèse
  - `top_io.v` : netlist verilog qui instancie le top et les plots d'entrée/sortie
- Fichier Captable : `c35b4.capTable` (optionnel)
- Fichier de contrainte (SDC). (exemple : `filter.sdc`) (optionnel)
- Pines d'alimentations (voir la figure 3)

### Remarque

Les fichiers .lib, .captable, .sdc sont répertoriés dans le fichier view\_definition.tcl que vous devez charger dans l'onglet <<Analysis Configuration>> dans la fenêtre <<Design Import>>.

Pour importer les bibliothèques nécessaires, leurs fichiers de timing respectifs ainsi que le Design, aller dans le menu:

### File > Import Design ...

Ajouter les fichiers comme montré sur la figure suivante (fig3).



Figure 3 Importation graphique du design

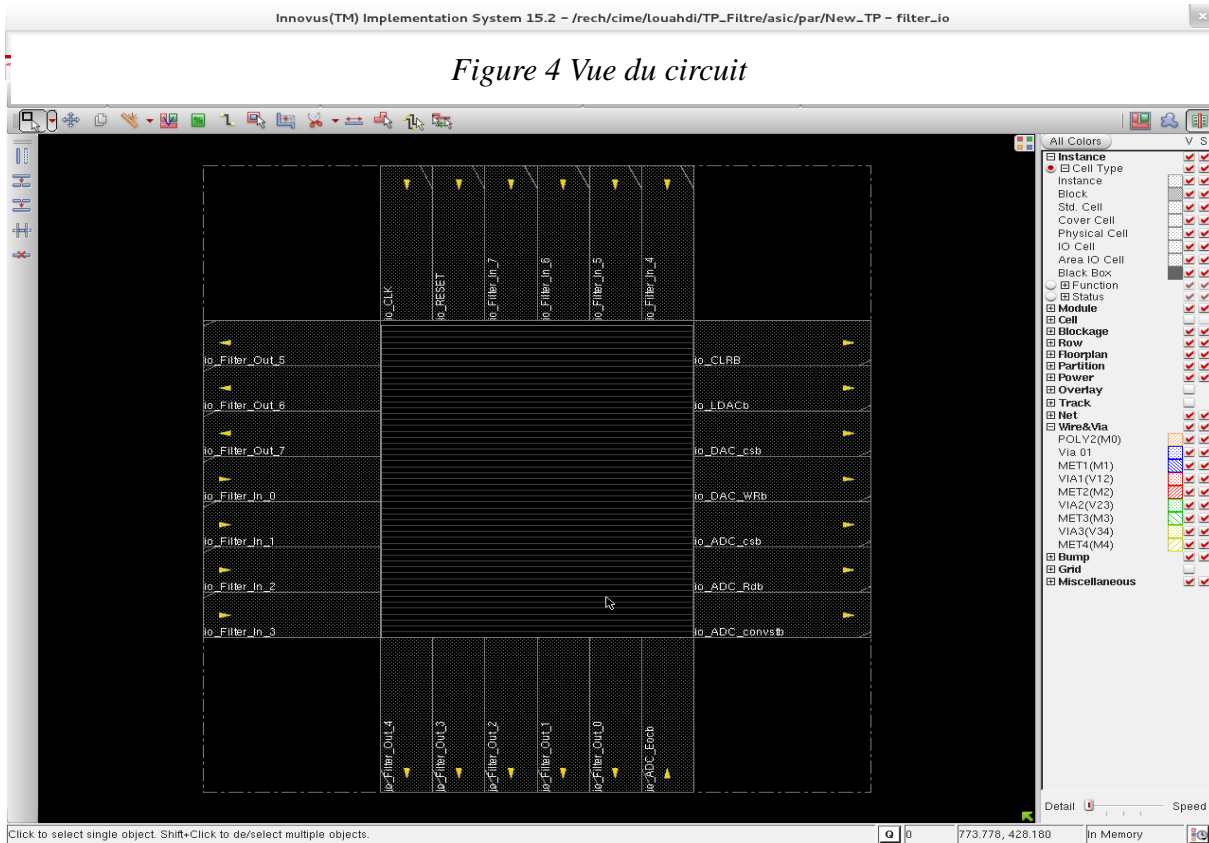


Figure 4 Vue du circuit

## ii. Méthode script

La procédure décrite dans ce paragraphe n'est à faire qu'une seule fois dans la vie du projet, après le premier démarrage d'Innovus.

Le script de configuration doit d'abord être chargé dans Innovus, ce qui se fait en exécutant la commande, dans la console depuis laquelle Innovus (celle dont le prompt a été remplacé par innovus {n°}>) : **source init.tcl** puis **init\_design**

Permettant d'initialiser le flot à l'aides des variables répertoriées dans le script, peut être appelée une fois et uniquement si les données de configuration de design ne sont pas chargées.

### ATTENTION :

Dès que vous puissiez sauvegarder la 1er étape de votre projet, vous devez restaurer votre design on démarrart seulement Innovus; les deux commandes ne seront plus utiles.

Vous obtenez une vue du circuit avec la couronne de plots placée et une zone centrale réservée pour le cœur.

Le rafraichissement de la zone de vue ne se fait pas toujours automatiquement. Pour forcer le programme à redessiner la zone de vue, tapez **Ctrl+R** ou cliquez sur l'icône Pour zoomer et dézoomer, tournez la molette de la souris.

Un clic sur le bouton central de la souris place la zone pointée par la flèche de la souris au centre de la zone de vue.

La composition des touches **Ctrl et Maj (Shift)** avec la rotation de la molette permet de faire défiler la vue latéralement ou verticalement.

Pour replacer la vue au centre du circuit et la réajuster de façon à voir le circuit dans son ensemble, appuyer sur la touche 'f'.

Pour sauvegarder un design, menu

**File | Save Design ...**

Les fichiers de sauvegarde sont regroupés dans un dossier .dat. Il est possible de donner un nom différent à chaque sauvegarde.

Pour charger une sauvegarde et revenir à un état antérieur, menu

**File | Restore Design ... ou taper la commande :**

restoreDesign (votre projet Innovus.dat) (topcell)

**ATTENTION** : certaines actions sont irréversibles ! Donc, sauvegardez votre travail à chaque étape et utilisez des noms de sauvegarde différents.

## 5. Définition des surfaces de placement routage (Floorplan)

### i. Couronne de plots

L'agencement de la couronne et le positionnement des plots est défini dans le fichier

`${TP_PATH}/asic/par/CONSTRAINTS/top_FILTER_etudiants.io`

qui a été préparé pour vous dans ce TP. Ce fichier doit être chargé en sélectionnant le menu

**File | load | IO Assignment File...**

Vous pouvez toutefois le visualiser dans un éditeur de texte à titre d'information.

La spécification d'un nouveau plot commence par le mot clé `Pad`: (premier champ).

Le deuxième champ est le nom de l'instance du plot. Ce nom s'affiche sur le dessin du plot (zoomer si le nom ne s'affiche pas) dans la zone de vue de la fenêtre principale de Innovus. Pour les signaux d'entrée ou de sortie du cœur qui doivent être connectés avec l'extérieur, ce nom d'instance renvoie au nom de l'instance de l'amplificateur de connexion associé. Les amplificateurs de connexion sont instanciés dans le fichier décrivant la puce complète `VERILOG/filter_io_etudiants.v`.

Les coins (*corner*) sont des plots spéciaux qui n'interviennent pas dans l'interconnexion avec le cœur. Il n'apparaît donc pas dans le fichier `VERILOG/filter_io_etudiants.v` et leur nom d'instance est conventionnel.

Le troisième champ est la zone de placement du plot par rapport au cœur, selon une désignation dans le style des points cardinaux. Le nord (N) est en haut de la zone de vue, l'est (E) à droite, etc. L'ordre d'écriture des lignes détermine l'ordre de placement dans une zone donnée, le n° 1 est à gauche ou en bas.

Le quatrième champ n'est renseigné que sur les plots spéciaux : les coins (`CORNERP`) et les plots d'alimentation (`VDD3ALLP` et `GND3ALLP`).

Des champs supplémentaires, non-enseignés ici, pourraient être ajoutés pour spécifier des contraintes de dimension du plot.



## ii. Alimentation du circuit

Pendant la conception du cœur du circuit, vous ne vous êtes pas préoccupés de l'alimentation du circuit :

Aucun de vos modules ne comporte explicitement les entrées de puissance (Vdd ou Gnd). Pourtant les circuits électroniques fonctionnent à l'électricité ! Il faut donc bien apporter l'énergie jusqu'aux cellules du cœur ! En fait, les bornes de puissance sont sous-entendues dans la description des portes logiques. Dans le cas général, il peut y avoir plusieurs alimentations (par exemple +5V, +3V et -5V) et même plusieurs masses (par exemple un GND numérique et un GND analogique). Dans ce TP, nous restons dans le cas simple d'une seule alimentation qui sera désignée par vdd! et d'une seule masse appelée gnd! (le point d'exclamation fait partie du nom)

La mise en place de l'alimentation du cœur se fait en plusieurs étapes :

- Aménagement d'un espace entre le cœur et la couronne de plots
- Placement d'anneaux d'alimentation (respectivement vdd! et gnd!) dans l'espace aménagé entre le cœur et la couronne de plots
- Placement de bandes de conduction destinées à réduire l'effet de l'impédance des rails d'alimentation du cœur
- Connexion des rails d'alimentation du cœur aux bandes de conductions

### a) Aménagement d'un espace entre le cœur et la couronne de plots:

Menu

**Floorplan | Specify Floorplan...**

Choisissez les valeurs suivantes :

- Ratio : 1.0  
(impose une forme carrée au cœur)
- Core Utilization : 0.70
- Core Margins by :  
Core to IO Boundary : 80 (sur les quatre côtés)

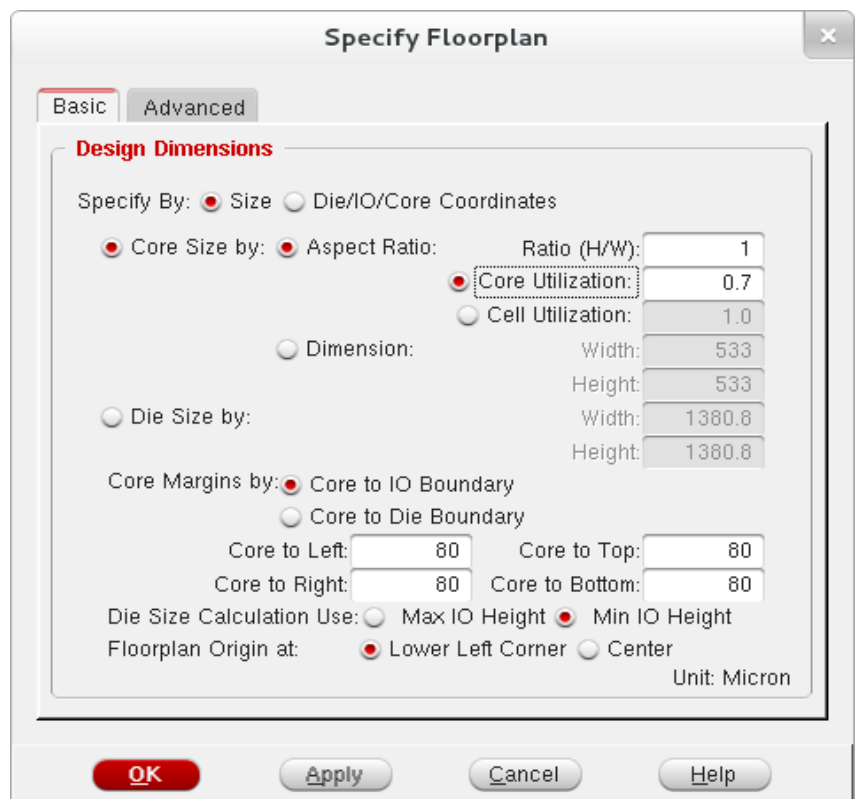


Figure 5 paramètres pour aménager l'espace entre le cœur et la couronne de plots

Cliquez sur « OK » et vous obtenez une nouvelle vue du circuit comme cela est montré sur la figure 5.

**Sauvegardez régulièrement votre travail!**

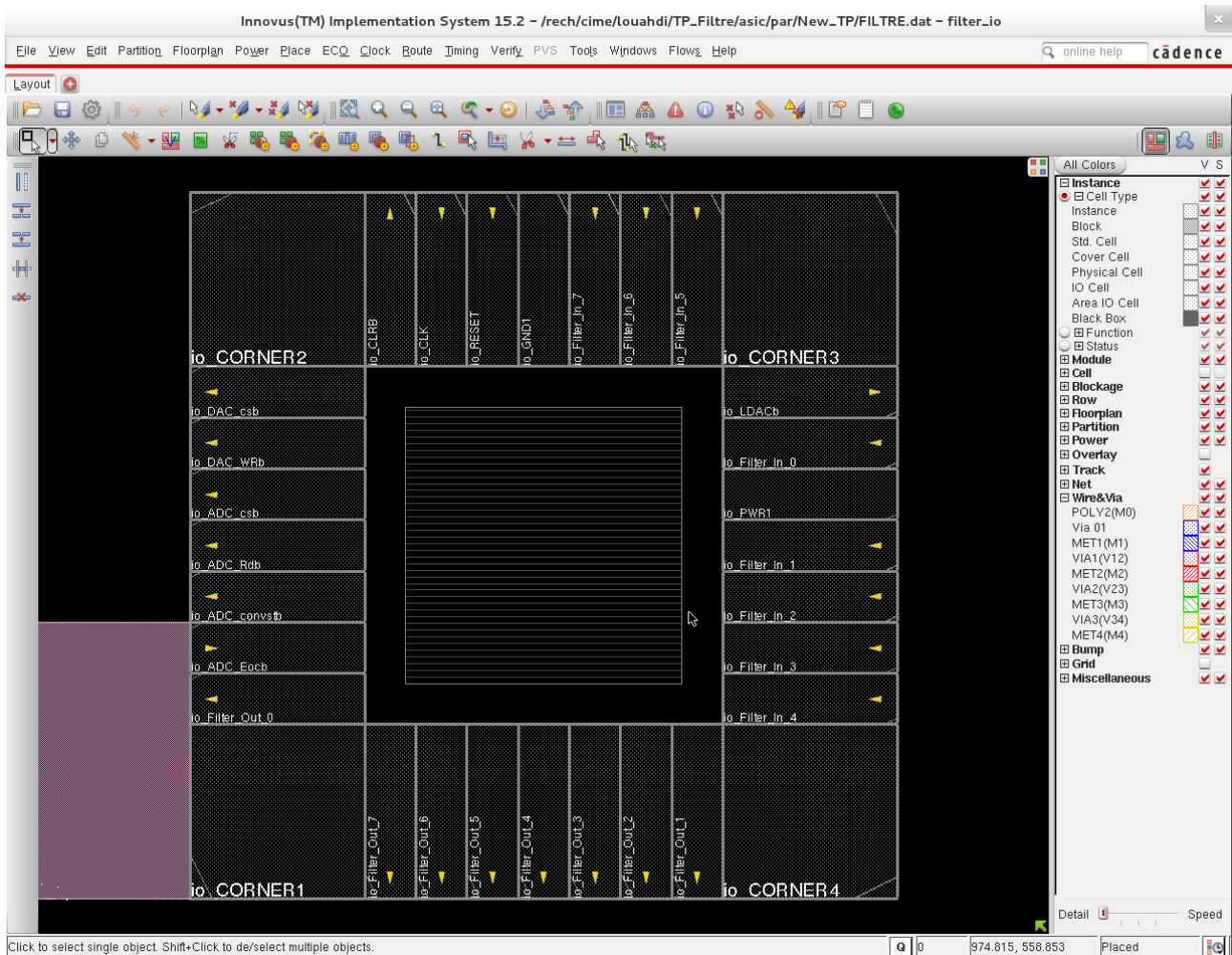


Figure 6 Vue du circuit après la spécification du FloorPlan

### b) Placement des anneaux d'alimentation:

La largeur des anneaux d'alimentation dépend de la consommation de votre circuit et des caractéristiques de la technologie utilisée. Il faut compter une largeur de piste d'environ 1  $\mu\text{m}$  pour 1 mA. Ici, des anneaux constitués de pistes de 20  $\mu\text{m}$  suffiront.

Menu

**Power | Power Planning | Add Ring...**

Dans l'onglet « Basic », remplir la zone de saisie « Net(s) » avec le nom des alimentations, soit ici vdd! et gnd!

Définir la largeur et l'espacement des anneaux d'alimentation pour chaque côté du cœur : width = 20 et spacing = 10.



Le paramètre offset indique l'espacement entre le cœur du circuit et le premier anneau d'alimentation.

En sélectionnant « Center in channel », les anneaux d'alimentation sont placés à égale distance des plots et du cœur. Voir la figure 6.

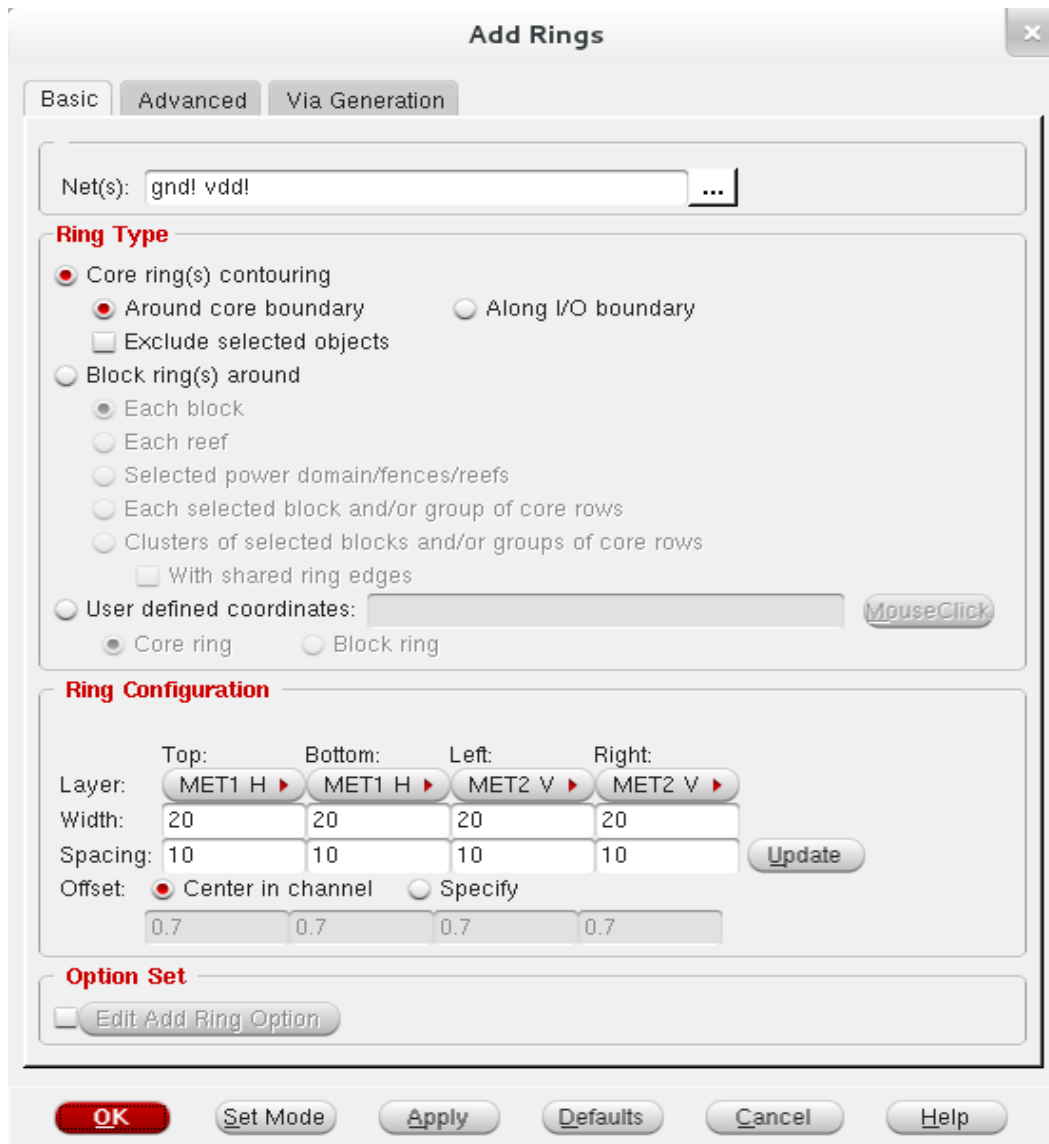


Figure 7 Spécification des anneaux d'alimentation

### c) Placement de bandes de conduction

Quatre paires de bandes de conduction sont définies et régulièrement espacées sur le cœur. Destinée à répartir l'approvisionnement en courant, leur largeur est donc logiquement un quart de celle des anneaux d'alimentation. Menu Power | Power Planning | Add Stripe... Dans la zone « Set Configuration » de l'onglet « Basic », remplir la zone de saisie « Net(s) » avec le nom des alimentations, soit ici vdd! et gnd!. Voir figure 7.

Définir la largeur et l'espacement des bandes de conduction pour chaque côté du cœur : width = 5 et spacing = 0.5.

Dans la zone « Set Pattern », sélectionner « Number of sets » et spécifier 4.

Dans la zone « Stripe Boundary », sélectionner « Core Ring ».

Dans la zone « First/Last Stripe », sélectionner « left » et « Relative from core or selected area » puis indiquer 100 dans les champs « from left » et « from right ».

Cliquez sur « OK » et vous obtenez une nouvelle vue du circuit comme cela est montré sur la figure 9.

**Add Stripes**

Basic Advanced Via Generation

**Set Configuration**

Net(s): gnd! vdd! ...

Layer: MET2 Direction: ☒ Vertical ☐ Horizontal

Width: 5 Spacing: 0.5 Update

**Set Pattern**

☐ Set-to-set distance: 100

☒ Number of sets: 4

☐ Bumps ☒ Over ☐ Between

☐ Over P/G pins Pin layer: Top pin layer ☐ Max pin width: 0

☒ Master name:  ☐ Selected blocks ☐ All blocks

**Stripe Boundary**

☒ Core ring

☐ Pad ring ☐ Inner ☒ Outer

☐ Design boundary ☒ Create pins

☐ Each selected block/domain/fence

☐ All domains

☐ Specify rectangular area

☐ Specify rectilinear area

**First/Last Stripe**

Start from: ☒ left ☐ right

☒ Relative from core or selected area

X from left: 100 X from right: 100

☐ Absolute locations

**Option Set**

☒ Edit Add Stripe Option

OK Set Mode Apply Defaults Cancel Help

Figure 8 Spécifications des bandes de conduction

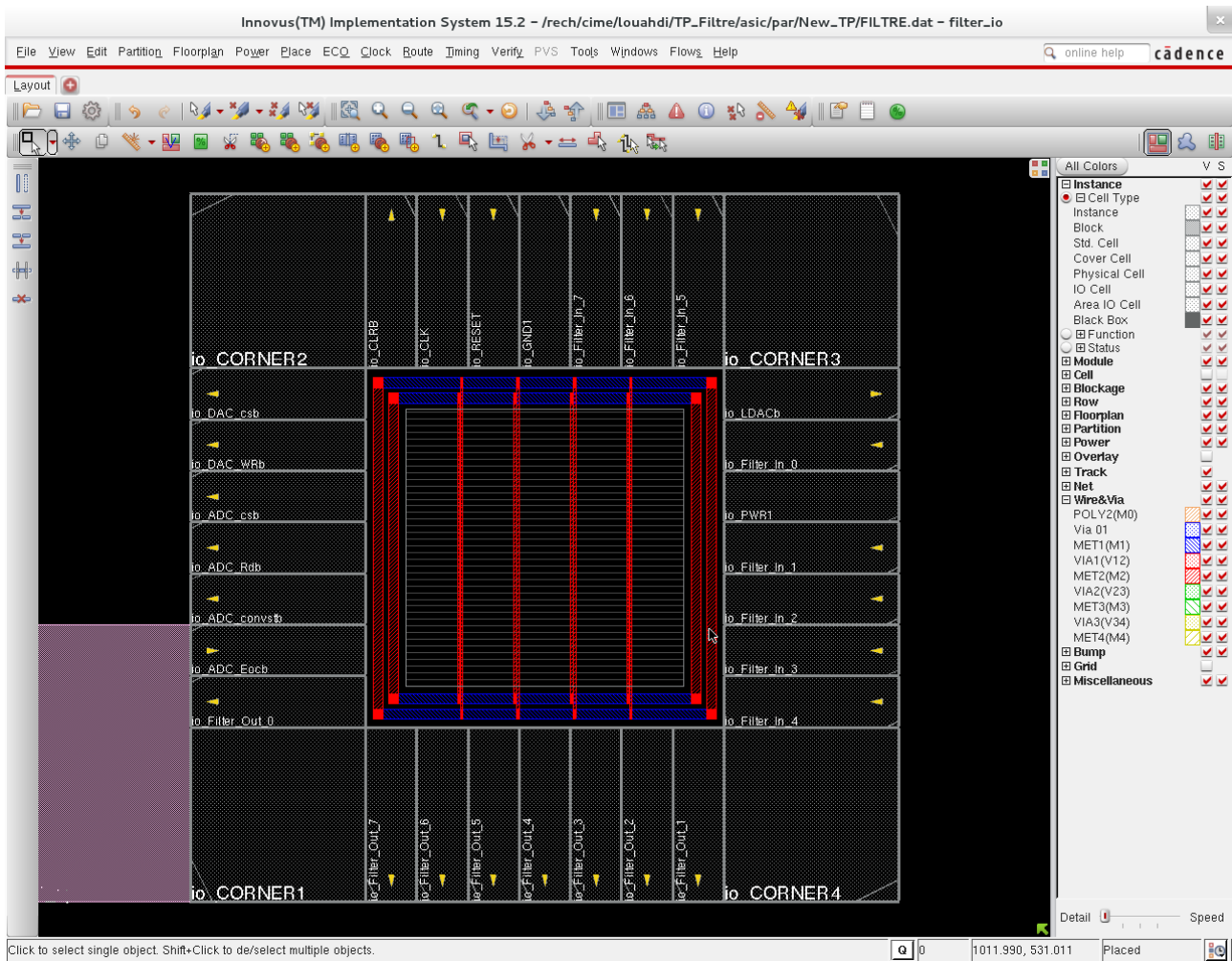


Figure 9 vue du circuit avec les anneaux et les bandes de conductions

#### d) Connexion des rails d'alimentation du cœur

Cette étape va définir un lien entre les bornes de puissance de chaque cellule et les plots d'alimentation du circuit. Le lien est obtenu par l'intermédiaire de deux noeuds globaux auxquels sont respectivement connectés d'une part les bornes de puissance des cellules composant le coeur et d'autre part les amplis de connexion des plots d'alimentation du circuit. D'où une procédure de configuration en deux étapes. Elle est suivie d'une troisième étape au cours de laquelle les rails d'alimentation seront effectivement routés.

Le nom par défaut pour désigner la borne d'alimentation positive des cellules du coeur est vdd! Ce nom est réemployé ici pour désigner le noeud global (*Global Net*) d'alimentation positive. Le nom par défaut pour désigner la borne d'alimentation négative des cellules du coeur est gnd! Ce nom est réemployé ici pour désigner le noeud global d'alimentation négative.

## Menu Power | Connect Global Nets...

Sélectionner « Pin » dans la zone « Connect ».

### Étape n°1)

Connexion des bornes de puissance des cellules composant le cœur aux nœuds globaux d'alimentation :

Taper '\*' dans « Instance Basename »

Taper vdd! dans les champs « Pin Name(s) » et « To Global Net » puis cliquer sur « Add to List »

Taper gnd! dans les champs « Pin Name(s) » et « To Global Net » puis cliquer sur « Add to List »

### Étape n°2)

Connexion des amplis de connexion des plots d'alimentation aux nœuds globaux d'alimentation :

Taper PWR1 dans « Instance Basename »

Taper 'A' dans le champ « Pin Name(s) »

Taper vdd! dans le champ « To Global Net » puis cliquer sur « Add to List »

Taper GND1 dans « Instance Basename »

Laisser 'A' dans le champ « Pin Name(s) »

Taper gnd! dans le champ « To Global Net » puis cliquer sur « Add to List » Appliquer les opérations indiquées dans la colonne de gauche « Connection List » en cliquant le bouton « Apply » puis fermer la fenêtre en cliquant « Cancel » (voir figure9). Il n'y a pour le moment aucun changement dans la zone de vue.

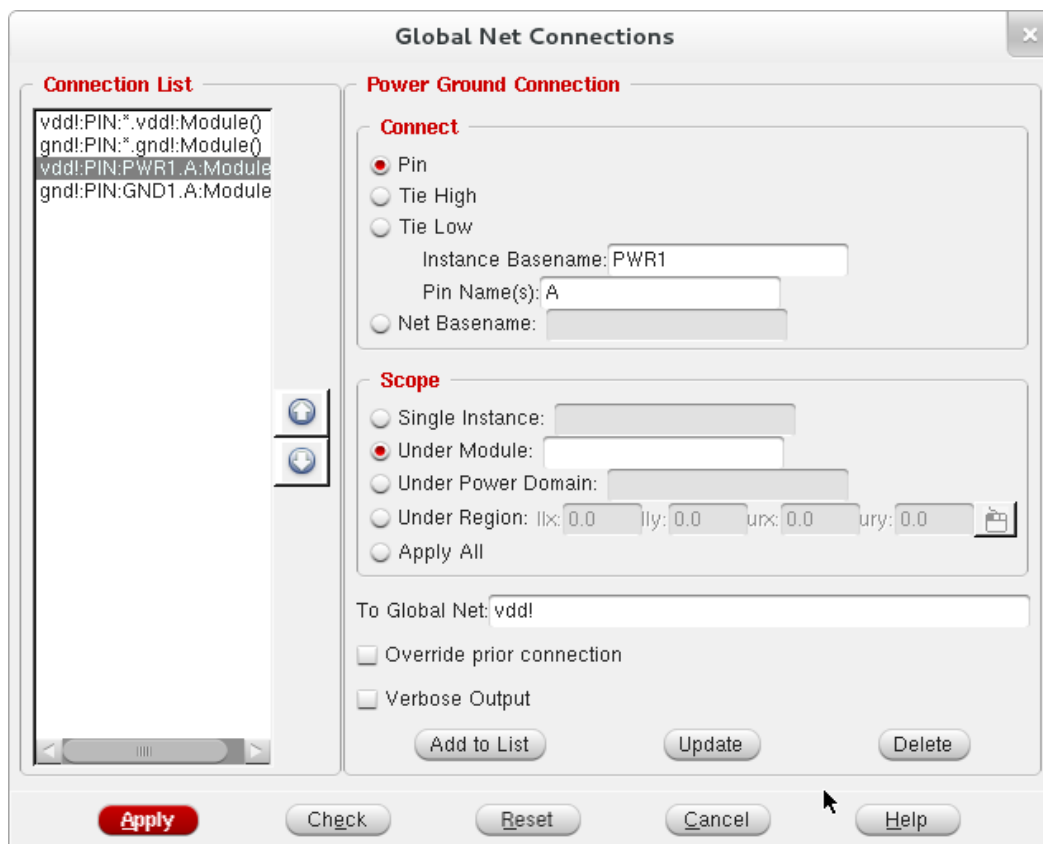


Figure 10 connection des nets d'alimentation

### Étape n°3) Routage des rails d'alimentation du cœur.

#### Menu

#### **Route | Special Route ...**

Dans l'onglet « Basic », la zone de saisie « Net(s) » sert à indiquer les noeuds qui seront à connecter aux bandes placées par dessus le coeur . Il s'agit ici des alimentations gnd! et vdd!.

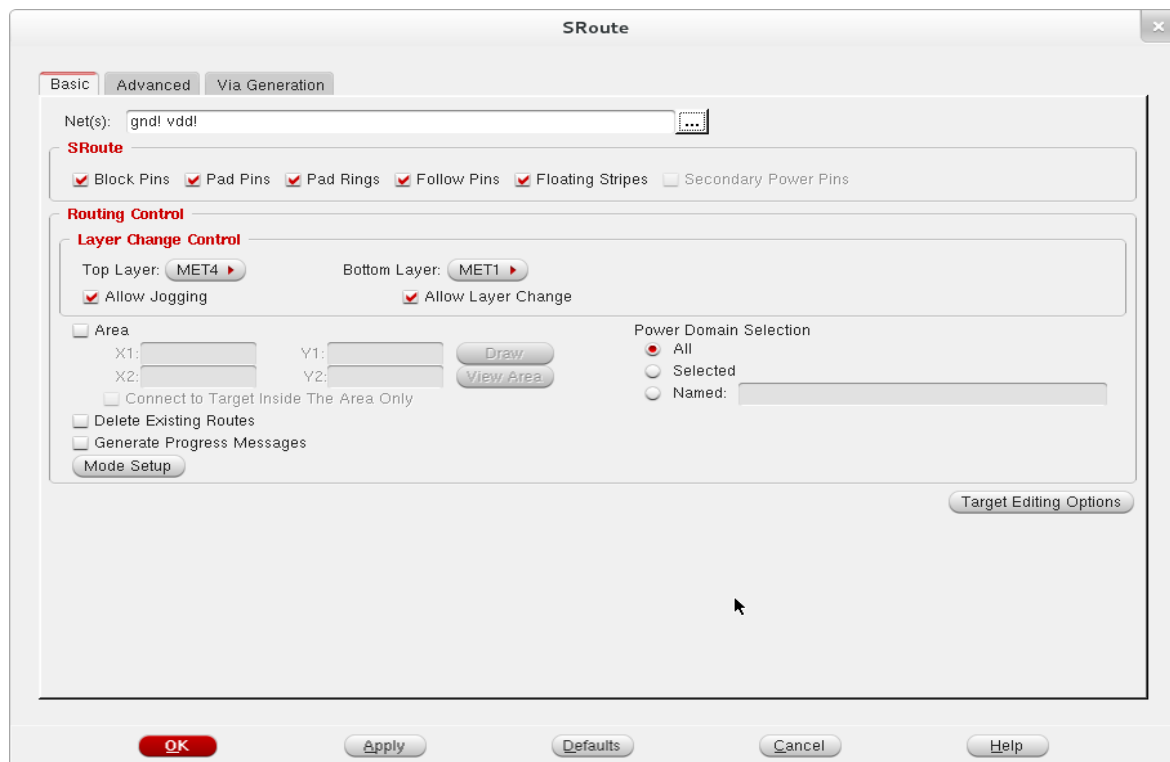
La zone « SRoute » sert à indiquer quel routage est à effectuer avec les anneaux d'alimentation et les bandes de conduction.

– « Block Pins » pour permettre de relier plusieurs bloc sur les mêmes anneaux d'alimentation les entourant.

Ceci n'a que peut de sens ici puisque le design n'a qu'un seul bloc.

– « Pad Pin » pour indiquer de connecter les plots d'alimentation à leur anneau d'alimentation respectif.

– « Pad Rings » (couronnes de plots) pour indiquer de créer une couronne de plots (ou plusieurs concentriques) si celle-ci n'est pas encore définie.



*Figure 11 Fenêtre special route*

– « Follow Pin » pour indiquer de relier les bornes d'alimentation des cellules standards aux rails d'alimentation qui vont être disposés en lignes à travers le coeur.

La zone « Routing control » sert à spécifier des contraintes de routage. La zone « Layer Change Control » permet d'indiquer les couches de métal autorisées pour le routage des alimentations et si le changement de couche est permis.

Si la case à cocher « Delete Existing Routes » est cochée, le routage est recommencé à zéro sinon le routage est incrémental.

Parmi les autres options, « Area » sert à spécifier de restreindre le routage à seulement une partie de la puce. « Power Domain Selection » sert restreindre le routage que sur un domaine d'alimentation (un noeud à une même tension d'alimentation). Ces options prennent tout leur sens pour les gros circuits.

Inscrire « gnd! vdd! » dans la zone de saisie « Net(s) » et laissez les valeurs par défaut pour les autres réglages (voir figure 11) ; au besoin cliquez le bouton « Defaults ».

Cliquez « OK », vous obtenez une vue comparable à celle de la figure 12.

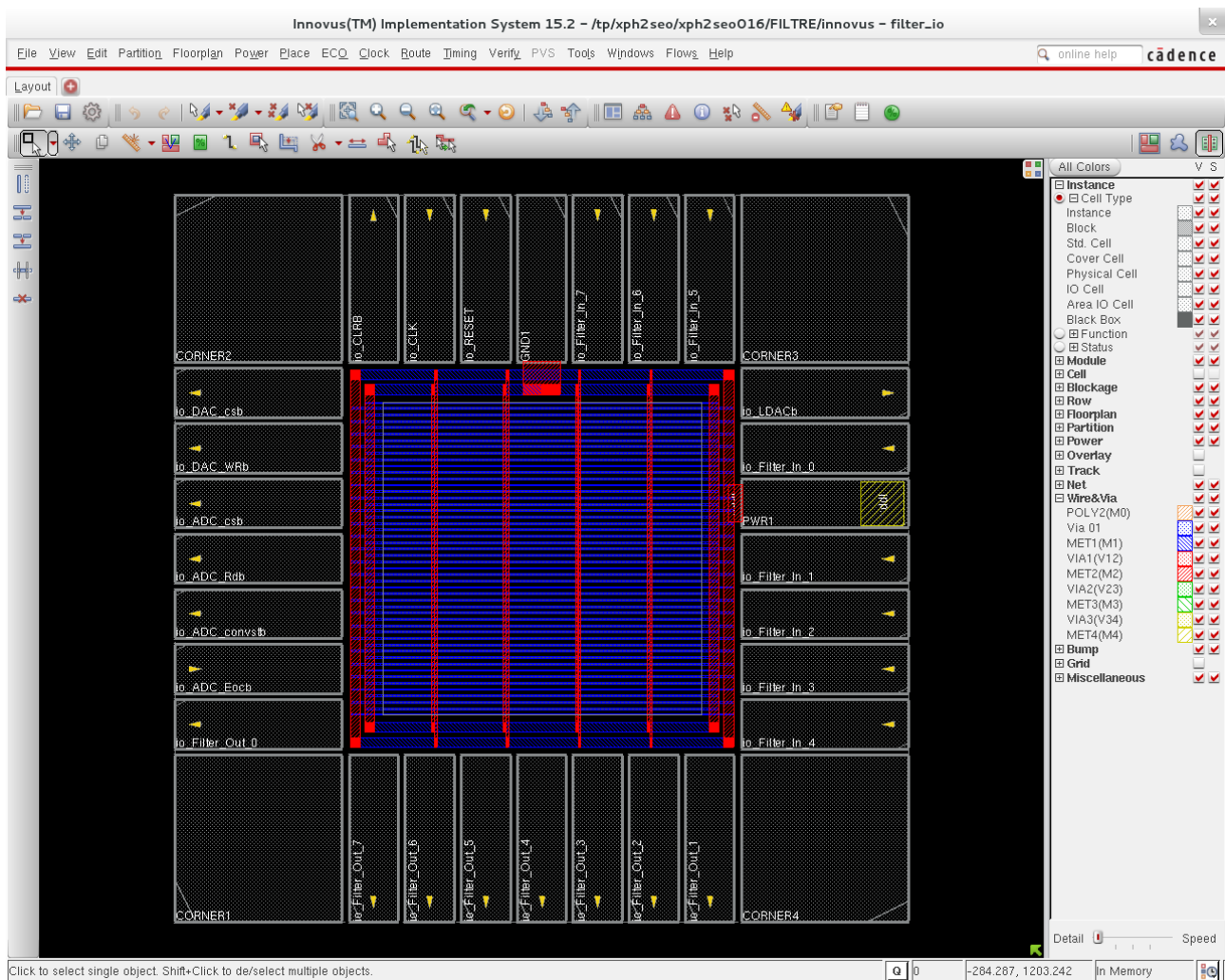


Figure 12 le circuit avec special route

Or, la bibliothèque technologique AMS utilisée ici est en 350 nm (0.35  $\mu\text{m}$ ). Il faut donc indiquer d'utiliser un noeud technologique supérieur ou égal à 130 nm par la commande suivante, à taper dans la console dans laquelle l'outil a été lancé :

***setDesignMode -process 130***



Pour un bon fonctionnement du circuit, des condensateurs de découplage doivent être placés entre les rails d'alimentation vdd! et gnd!. Ils sont placés par le script suivant (toujours à exécuter dans la console d'Innovus) :

***addEndCap -preCap ENDCAPL -postCap ENCAPR prefix ENDCAP***

Ensuite sélectionnez le menu

**Place | Place Standard Cells...**

- Sélectionnez « Run Full Placement »
- Dans la zone « Optimization options », sélectionnez « Include Pre-Place Optimization »
- Cliquer « OK »

Pour faire apparaître les cellules placées, sélectionnez la vue physique en cliquant l'icône physical

view à droite de la barre d'outils



Menu

**Place | Check Placement...**

Vérifiez que « Check Placement report to » est coché (un nom de fichier d'extension \*.checkPlace est proposé par défaut) puis cliquez « OK ».

Un rapport de placement a été généré dans la console. Si des violations apparaissent comme dans l'extrait de rapport de la figure 12, un placement plus raffiné est nécessaire.

```
innovus 17> Begin checking placement ... (start mem=1192.3M, init mem=1192.3M)
*info: Placed = 1204          (Fixed = 82)
*info: Unplaced = 0
Placement Density:54.93%(154536/281317)
Finished checkPlace (cpu: total=0:00:00.0, vio checks=0:00:00.0; mem=1192.3M)
```

Figure 13 Exemple d'un rapport de placement

## 6. Génération de l'arbre d'horloge

La génération de l'arbre d'horloge nécessite un fichier de contraintes qui spécifie les caractéristiques du signal d'horloge. C'est dans notre cas le fichier :

`${TP_PATH}/asic/par/CONSTRAINTS/ctgen.ctstch`

dont le contenu est reproduit figure.

### i. Synthèse de l'arbre d'horloge

Pour définir les paramètres globales de la Synthèse de l'arbre d'horloge il faut taper sur la console la commande :

***setCTSMode -engine ck***

```
# FirstEncounter(TM) Clock Synthesis Technology File Format
#
# -----
# Clock Root      : CLK
# Clock Name      : CLK
# Clock Period    : 20ns
# -----
AutoCTSRootPin io_Clk/Y
Period          20ns
MaxDelay         20ns # default value
MinDelay         0ns  # default value
MaxSkew          200ps # default value
SinkMaxTran      400ps # default value
BufMaxTran       400ps # default value
Buffer           BUF2 BUF4 BUF6 BUF8 BUF12 BUF15
NoGating         NO
DetailReport     YES
END
```

Menu

**Clock | Synthesize Clock Tree...**

Dans « Clock specification file », saisir

CONSTRAINTS/ctgen.ctstch puis cliquez « OK ».

Le compte rendu sur l'arbre d'horloge est enregistré  
dans le fichier

**`${TP_PATH}/asic/par/clock_report/clock.report`**

**ii. Visualisation de l'arbre d'horloge**

Menu

**Clock | Display | Display Clock Tree...**

– Dans la zone « Clock Selection », sélectionnez « Selected Clock » et renseignez le nom de l'horloge : `io_CLK/Y`.

– Dans la zone « Route Selection », sélectionnez « Pre-Route ». Ce choix permet d'afficher des évaluations des temps de propagation et de profondeur de l'arbre d'horloge.

– Dans la zone « Display Selection », sélectionnez « Display Clock Phase Delay » pour visualiser les différents niveaux de l'arbre d'horloge.

– Cliquez « Apply ». Dans la fenêtre principale, un ensemble de pastilles colorée indiquent symboliquement (les valeurs numériques sont écrites dans le rapport) le temps de propagation allant du bleu pour le chemin le plus court au rouge pour le chemin plus long.

– De nouveau dans la fenêtre « Display Clock Tree », sélectionnez la zone « Display Selection » les boutons radio « Display Clock Tree » et « Selected Level »

– Augmentez progressivement le nombre de niveaux, c'est à dire ici le nombre de fourches franchies,

en cliquant « Apply » à chaque fois. Un système de pastilles colorées du bleu au rouge montre les endroits du circuit atteint après le franchissement d'une, deux, trois, etc. fourches sur l'arbre d'horloge.

– Dans la zone « Route Selection », sélectionnez « Post-CTS » et dans la zone « Display Selection », sélectionnez les boutons radio « Display Clock Tree » et « All Level ».

Ces choix permettent d'afficher le routage que l'outil propose pour l'arbre d'horloge

– Cliquez « OK » pour fermer la fenêtre « Display Clock Tree ».

– L'arbre d'horloge d'horloge (et ses pastilles colorées) peut être masqué par le menu

**Clock | Display | Clear Clock Tree Display**

**Remarque :** À ce stade l'arbre d'horloge n'est pas encore routé à proprement parlé. Cette étape a juste déterminé la profondeur des arbres (nombre de fourches) en fonction de la disposition des cellules constituant le coeur. Il peut s'avérer nécessaire d'ajouter des étages d'amplification (*buffers*) pour garantir des temps de montées ou remettre en forme le signal d'horloge. C'est l'objet de l'étape d'optimisation cellules en place (*In-Place Optimization Post Clock-Tree Synthesis – IPO post-CTS*).

## 7. Remplissage des vides

Au cours de la fabrication, les dépôts et les enlèvements de matière se succèdent. Des procédés différents sont utilisés selon la quantité de matière à déposer ou retirer. Pour qu'un procédé soit le plus efficace possible, les surface à traiter doivent présenter une densité de motif la plus régulière possible. En effet, de petits motifs isolés seront gommés si un procédé destiné à réaliser de grands motifs est employé tandis que de larges motifs ne seront pas correctement reproduits si des procédés prévus pour garantir la finesse d'un petit motif est employé. C'est pourquoi, il est nécessaire de combler les vides qui peuvent exister entre les cellules placées dans le coeur ou ceux qu'il peut y avoir entre les plots dans la couronne. Les vides sont comblés par des remplisseurs (*fillers*) qui sont des cellules vides (et inactives) pour le coeur, et des plots vides (inutilisés) pour la couronne de plots. Ces opérations de remplissage vont être réalisées par l'exécution de scripts lancés depuis la console d'Innovus :

***setFillerMode -core {FILLANT1 FILLANT2 FILLANT5 FILLANT10 FILLANT25} -  
preserveUserOrder true***

ou par une seule commande de ligne :

soit ***addFiller -cell FILL25 FILL10 FILL5 FILL2 FILL1 -prefix FILLER***

soit ***addFiller -cell FILLRT25 FILLRT10 FILLRT5 FILLRT2 FILLRT1 -prefix FILLERT***

## 8. Routage effectif

Le circuit est désormais prêt pour le routage cependant il est impératif de commencer par les signaux spéciaux, aux contraintes les plus fortes. Le routage se fait donc en deux temps, d'abord l'arbre d'horloge puis le reste du circuit.

Le routage est réalisé en deux étapes : le routage global (*Global Routing*) et le routage détaillé (*Detailed Routing*). Le routage global est destiné à évaluer les zones de congestion potentielles et par suite à les éviter.

Pour cela l'outil découpe le design en blocs rectangulaires appelés cellules de routage globales (*global routing cells* – gcells), auxquelles il leur associe des noeuds correspondant aux signaux traversant ces blocs. Ensuite, il estime la possibilité de réussir le routage de ces cellules globales mais ne trace aucune piste. Une carte de congestion peut être générées. Le routage détaillé utilise les informations du routage global pour effectivement tracer les pistes d'interconnexion entre les différentes cellules qui composent le circuit.

Lancer le routage en exécutant la commande suivante :

***routeDesign***

## 9. Vérification

Après le routage, il est nécessaire de lancer une vérification sur les règles de dessins ainsi que sur les différentes connections.

Menu

**Verify | Verify Geometry...**

Puisque le placement a été fait en s'alignant sur la grille du fondeur, il faut s'assurer que:

- « Off Routing Grid » est décoché
- « Off Manufacturing Grid » est coché

Menu

**Verify | Verify Connectivity...**

Pour vérifier qu'il ne reste pas d'éléments non connectés (cf. figure).

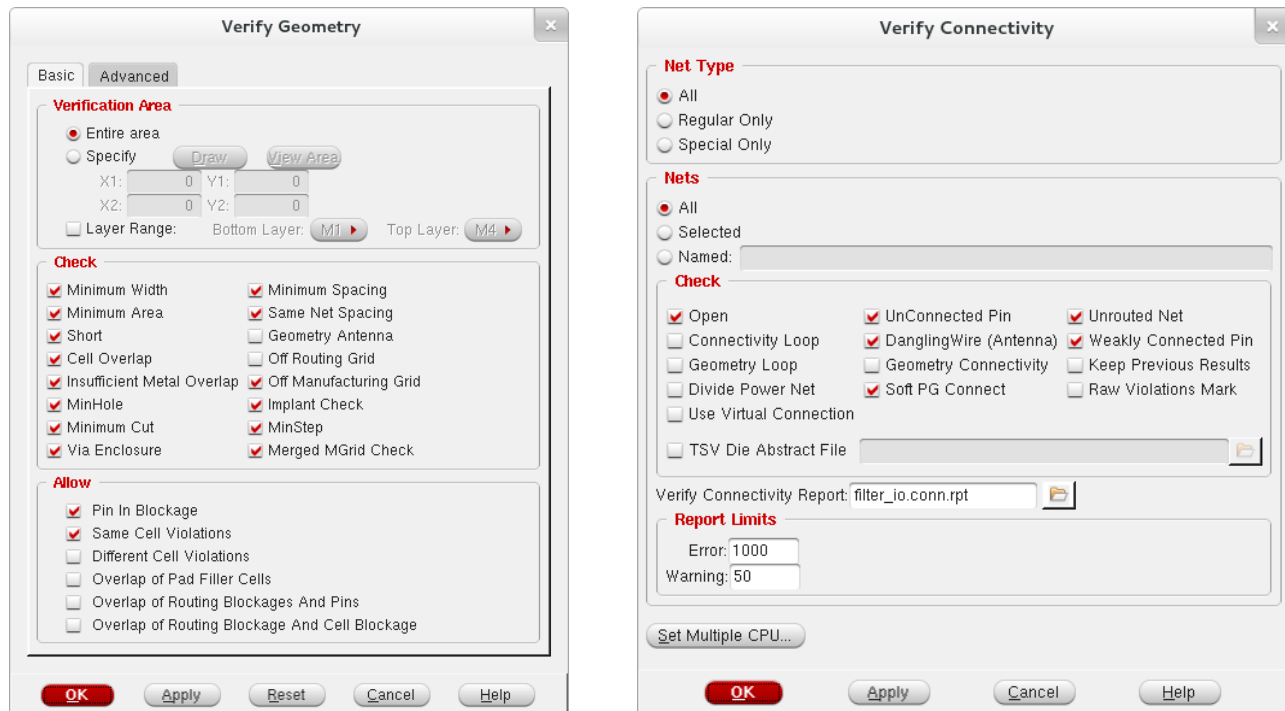


Figure 14 fenêtres de verification

## 10.Caractéristiques du circuit après P&R cible ASIC

Les principales caractéristiques du circuit (surface globale, nombre de cellules, nombre d'interconnexions, etc.) peuvent être rassemblées dans un récapitulatif (*Summary Report*).

Menu

**File | Report | Summary...**

## 11.Exportations des données de placement et routage

Après le placement et le routage, la réalisation du circuit se poursuivra par une simulation après placement et routage puis, si elle est satisfaisante, par le tracé des plans des masques (*layout*), des fichiers nécessaires pour le DRC, le LVS et les simulations post-routage.

Dans un premier temps créer un répertoire pour sauvegarder ces fichiers dans le répertoire : RESULTS.

Pour mener à bien ces opérations qui suivent ont besoin des informations sur les délais des interconnexions qui sont fortement influencés par la présence de RC parasites des interconnexions. Il est utilisé pour les simulations rétroannotées. Il faut donc d'abord procéder à l'extraction des RC parasites (*parasitic extraction*) puis enregistrer les informations des délais des interconnexions. Le format de fichier utilisé est le *Standard Delay Format* – SDF.

Menus

**Timing | Extract RC...**

**Timing | Write SDF...**

- ✓ **DEF** : Ce fichier permet d'importer le design du circuit dans cadence. On peut aussi utiliser le fichier gdsII. **File > Save > DEF...**
- ✓ **GDSII** : Ce fichier est une vue physique du circuit. Il sera importé dans cadence pour les vérifications DRC et LVS. **File > Save > GDSII...** Choisir le fichier XX.map pour le champ Map File Laisser « DesignLib » dans le champ « Library Name ».
- ✓ **Verilog** : **File > Save > Netlist...** Cette étape est nécessaire car des cellules (buffers) ont été ajoutées lors de la génération de l'arbre d'horloge, la netlist a donc changé

Le produit phare de la société Cadence est Virtuoso, un outil modulaire pour le tracé des plans de masque (*layout*) et pour la simulation électrique (*spice-like simulation*). Cadence utilise ses propres formats de données pour l'échange entre ses outils : *Design Exchange Format* (DEF) qui est appelé à être prochainement remplacé par le format *Open Access* (OA). L'exportation du circuits placé et routé dans ces formats est possible par le biais des menus « File | Save Design » et « File | Save | DEF ».

Quitter Innovus avant de continuer le TP.