TP7 : Placement et routage du circuit AMD2901

1 Travail sur le coeur : Préplacement des structures régulières

à partir de amd2901_core.py effectuez les étapes suivantes dans la méthode Layout :

- Placer le chemin de données : fonction Place ()
- Agrandir la boite d'aboutement du coeur : fonction ResizeAb ()
- Placer les rails de rappels d'alimentation dans le coeur : fonctions AlimVerticalRail () et AlimHorizontalRail ()
- Placer les connecteurs du coeur : fonction AlimConnectors ()

ATTENTION : La logique "irrégulière" constituant la partie contrôle n'a pas besoin

d'être placée explicitement. Cela sera fait automatiquement par la suite !

Vérifiez le résultat

```
> ./execute_amd2901_core.py
```

2 Travail sur le cicuit complet

Prenez le fichier amd2901_chip.py et complétez la méthode Layout.

2.1 Placement de la couronne de plots et du coeur

Dans le fichier amd2901_chip.py fourni, les plots sont instanciés dans la méthode

Netlist :

```
def Netlist ( self ) :
...
  Inst ( "pck_px", "p_ck"
        , map = { ?pad? : self.ck
                , ?ck? : cki
                , ?vddi? : self.vdd
                , ?vssi? : self.vss
                , ?vdde? : self.vdde
```

```
                , ?vsse? : self.vsse
                }

        )
```

Il vous faut donc, dans la méthode Layout :

- Définir la taille de la boîte d'aboutement globale du circuit de façon à ce que

  les plots puissent être placés à la périphérie : fonction DefAb () (Commencer par définir une boite d'aboutement de 4000 par 4000 et vous essaierez ensuite de la diminuer)

- Placer le coeur du circuit au centre de la boîte d'aboutement du chip : fonction PlaceCentric ()

- Définir sur quelle face et dans quel ordre vous souhaitez placer les plots. Cela se

  fait à l'aide des 4 fonctions : PadNorth (), PadSouth (), PadEast () et PadWest ().

- Vérifiez le résultat :

```
> ./execute_amd2901_chip.py
```

2.2 Routage des alimentations

Vous devez utiliser la fonction PowerRing () pour créer la grille d'alimentation. Vérifiez le résultat :

```
> ./execute_amd2901_chip.py
```

2.3 Placement de la logique irrégulière

C'est le placeur Mistral qui se charge de placer les cellules de la partie de contrôle. Il détecte quelles sont les cellules qui n'ont pas été placées et complète le placement en utilisant les zones "vides". Pour appeler le placeur Mistral, vous devez faire appel à la fonction PlaceGlue ()

Attention : Pour pouvoir placer automatiquement la logique "irrégulière", il faut que les plots soient placés. L'outil de placement du flot CORIOLIS place les cellules en se basant sur les attirances de celles-ci vers les plots ainsi que vers les cellules déjà placées.

Vérifiez le résultat :

```
> ./execute_amd2901_chip.py
```

Le placement automatique se termine par l'appel à la fonction FillCell () qui effectue le placement automatique de cellules de bourrage.

Vérifiez le résultat :

```
> ./execute_amd2901_chip.py
```

2.4 Routage des signaux d'horloge

Vous devez utiliser la fonction RouteCk () qui route le signal d'horloge. Vérifiez le résultat :

```
> ./execute_amd2901_chip.py
```

2.5 Routage des signaux logiques

Routez automatiquement tous les signaux autres que le signal d'horloge et les signaux d'alimentation en utilisant NERO de la manière suivante :

```
> nero -V -p amd2901_chip amd2901_chip amd2901_chip_r
```

L'option -p indique que vous transmettez un placement, à savoir celui du chip. Le troisième argument est la netlist du chip, le quatrième est le fichier résultat.

NOTA BENE : La variable MBK_CATA_LIB ne doit contenir qu'une seule fois les chemins d'accès aux bibliothèques.

2.6 Validation du chip

- On validera le travail de NERO avec les outils druc, cougar et lvx.

```
> druc amd2901_chip_r
> export MBK_OUT_LO=al
> cougar -f amd2901_chip_r
> lvx vst al amd2901_chip amd2901_chip_r -f
```

- Simulez à nouveau la netlist extraite avec ASIMUT. Précisez le format de la netlist

dans la variable d'entrée MBK_IN_LO avant la simulation.

```
> export MBK_IN_LO=al
```

Faites attention au fichier CATAL'"

- Pour connaitre le nombre de transistors, on effectue une extraction du circuit au niveau

transistor :

```
> cougar -v -t amd2901_chip_r amd2901_chip_r_t
```

Conclusion

Ce TP vous a permis de passer par la plupart des étapes nécessaires à la conception "back-end" et la validation d'un circuit réalisé en cellules précaractérisées avec préplacement des parties régulières.

Ces mêmes outils seront utilisés pour laréalisation du processeur MIPS R3000. Le compte-rendu du TP doit comporter :

Vos logins, vos noms et prénoms, et vos répertoires de travail pour ce TP (laissez libre accès à vos répertoires en lecture !). Une description exacte de la méthodologie employée, incluant les éventuels probl èmes rencontrés.

Pour l'amd2901, décrivez le flot de conception. Quels choix avez-vous retenus pour le placement des colonnes du chemin de données, votre circuit est-il limité par les plots ou par la taille du coeur (pad limited ou core limited)... Quels sont les résultats donnés par lvx... Les schémas sont appréciés.

Les Makeles du flot total. (Les Makefiles seront testés à la fin de ce TP) NE PAS JOINDRE DE LISTINGS DE FICHIERS (SAUF LES MAKEFILES). Merci et bon courage !