

# TP6 : Routage "overcell" d'un opérateur de décodage

1. 1 Objectif
2. 2 Spécification fonctionnelle
3. 3 Travail à effectuer
  1. 3.1 Saisie du schéma
  2. 3.2 Placement / Routage
  3. 3.3 Validation du routage
4. 4 Compte rendu

## 1 Objectif

Nous avons vu dans le TP5 comment dessiner une cellule précaractérisée respectant le gabarit de la bibliothèque **SXLIB**. Le but de ce TP6 est de réaliser le dessin des masques d'un bloc combinatoire résultant de l'interconnexion d'une quinzaine de cellules de la bibliothèque **SXLIB**.

On va donc effectuer *à la main* (c'est à dire en utilisant l'éditeur interactif **graal**), le placement et le routage de cet opérateur, pour mieux comprendre les problèmes que doivent résoudre les outils de placement/routage automatiques.

## 2 Spécification fonctionnelle

On souhaite réaliser le décodeur 3 vers 4 tel que décrit ci-dessous.

① La sortie  $Z_i$  prend la valeur 1 quand le nombre de bits d'entrée ayant la valeur 1 est égal à  $i$ . Les 3 autres sorties prennent la valeur 0.

Les équations sont les suivantes :

- $Z_0 = \overline{A}.\overline{B}.\overline{C}$
- $Z_1 = (A.\overline{B}.\overline{C}) + (\overline{A}.B.\overline{C}) + (\overline{A}.\overline{B}.C)$
- $Z_2 = (A.B.\overline{C}) + (A.\overline{B}.C) + (\overline{A}.B.C)$
- $Z_3 = A.B.C$

De ces équations, on déduit le schéma en portes logiques inverseuses :

①

Le bloc à réaliser contient donc 10 portes NAND3 et 5 inverseurs. Regardons plus précisément les caractéristiques de chaque cellule :

- La cellule `inv_x1` a une largeur de 3 pitches.
- Le signal d'entrée  $E$  est accessible sur 6 pistes de routage.
- Le signal de sortie  $S$  est accessible sur 7 pistes de routage.
  
- La cellule `na3_x1` a une largeur de 5 pitches.
- Les 3 signaux d'entrée  $E_1$ ,  $E_2$  et  $E_3$  sont accessibles sur 6 pistes de routage.
- Le signal de sortie  $S$  est accessible sur 7 pistes de routage.

① ①

On propose le placement ci-dessous pour cet opérateur, mais vous pouvez choisir un autre placement si vous le souhaitez :



Le dessin ci-dessous illustre un routage partiel correspondant à quelques signaux et utilisant des fils de Metal2 , Metal3, ainsi que des via12 et des vias23.



## 3 Travail à effectuer

### 3.1 Saisie du schéma

- Utiliser le langage **STRATUS** pour décrire le schéma proposé ci dessus, et générer le fichier **decodeur.vst** correspondant à la description structurelle VHDL de cet opérateur.
- Valider ce schéma en écrivant quelques stimuli, et en simulant avec **Stratus**.

### 3.2 Placement / Routage

- Saisir sous **graal** le dessin du bloc *decoder* en instanciant les 5 portes *inv\_x1* et les 10 portes *na3\_x1*.
- Dessiner les fils de routage sous **graal**, et utiliser la commande *equi* pour vérifier la connectivité de chacun des signaux.
- Vérifier l'absence de violation des règles de dessin en lançant la commande *druc* sous **graal**. Pour que cette vérification soit significative, il faut préalablement "mettre à plat" le bloc *decoder*, en utilisant la commande *real.flat*.

### 3.3 Validation du routage

- Extraire la netlist du bloc *decoder* au format **.al** avec l'outil **cougar** mais sans descendre au niveau des transistors : On veut obtenir une netlist de cellules, et non une netlist de transistors.
- Vérifier que les deux netlists définies par les fichiers **decoder.vst** et **decoder.al** sont isomorphes en utilisant l'outil **lvx**.
- Créer un fichier Makefile automatisant la procédure de validation.

## 4 Compte rendu

Vous rendrez un compte rendu d'une page maximum pour ce TP.

Vous joindrez vos fichier sans oublier le Makefile.