

TP6 : Routage manuel "overcell"

1. 1 Explications
2. 2 Travail à effectuer

Nous avons vu comment dessiner entièrement une cellule puis comment réaliser

une cellule simple instanciant d'autres cellules. Nous allons maintenant réaliser un cellule plus avancée nécessitant un routage manuel "overcell".

1 Explications

Le but est de créer un compteur de bits tel que décrit ci-dessous. 

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:

- $Z_0 = \neg A \cdot \neg B \cdot \neg C$
- $Z_1 = (A \cdot \neg B \cdot \neg C) + (\neg A \cdot B \cdot \neg C) + (\neg A \cdot \neg B \cdot C)$
- $Z_2 = (A \cdot B \cdot \neg C) + (A \cdot \neg B \cdot C) + (\neg A \cdot B \cdot C)$
- $Z_3 = A \cdot B \cdot C$

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



Nous avons vu comment dessiner entièrement une cellule puis comment réaliser une cellule simple instanciant d'autres cellules. Nous allons maintenant réaliser un cellule plus avancée nécessitant un routage manuel "overcell".

Notre cellule finale contiendra donc 10 portes NAND3 et 5 inverseurs. Regardons plus précisément les caractéristiques de chaque cellule :

- La cellule NAND3 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.



- La cellule INVERSEUR 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.



- Ces cellules précaractérisées sont conçues pour être aboutables dans les deux directions X et Y. On souhaite avoir le placement suivant de notre cellule :



Ci-dessous un exemple de routage "overcell" pour la génération du signal de sortie



2 Travail à effectuer

- Décrire le comportement de la cellule compteur de bits dans un fichier .vbe
- Saisir sous GRAAL le dessin de la cellule en instanciant des portes `inv_x1` et `na3_x1` de la `sxlib`
- Dessiner sous GRAAL le routage "overcell"
- Valider les règles de dessin symbolique en lançant DRUC sous GRAAL

- Extraire la netlist de l'inverseur au format .al avec COUGAR

- Extraire le VHDL comportemental avec YAGLE