

# TP2 : Modélisation structurelle avec Stratus

1. Avant propos
2. 1 Travail à effectuer
  1. 1.1 Familiarisation avec **Stratus**
  2. 1.2 Circuit addaccu
  3. 1.3 Circuit addsubaccu
3. 2 Compte rendu

## Avant propos

Dans ce TP, nous souhaitons réaliser un générateur de circuit addaccu amélioré avec comme paramètre, entre autres, le nombre de bits.



Le circuit addaccu a deux niveaux de hiérarchie : dans **addaccu** sont instanciés trois blocs **mux**, **add** et **reg**. Les blocs **mux**, **add** et **reg** doivent être décrits par une netlist paramétrable de cellules **sxlib**, dans le langage **Stratus**. En effet, **Stratus** permet de décrire des netlists paramétrables.

Fichiers fournis :

- La netlist en "**Stratus**" du bloc "**mux**"
- Script pour la création de la netlist

Ce bloc a la fonctionnalité suivante :

```
si (cmd==0) alors s <= i0 sinon s <= i1
```

i0, i1 et s ayant un nombre de bit paramétrable.

## 1 Travail à effectuer

### 1.1 Familiarisation avec Stratus

- Récupérer les deux fichiers permettant de créer le bloc **mux** et les étudier.
- Créer une instance de mux sur 22 bits. Pour ce faire, il faut exécuter le script fourni avec le bon paramètre.

```
> python gen_mux.py -n 2
```

Si le script s'effectue sans erreur, un fichier **.vst** est normalement généré. Vous pouvez vérifier qu'il décrit bien le circuit voulu. Ce bloc peut lui-même être instancié dans une netlist grâce à la méthode **Inst**.

### 1.2 Circuit addaccu

- Ecrire les blocs **add** et **reg** avec **Stratus** en utilisant exclusivement les cellules de la bibliothèque **sxlib**. Ces deux blocs prennent comme paramètre le nombre de bits. En outre, ils vérifient que leur paramètre est compris entre 2 et 64 (ce n'est pas fait dans mux).
- Ecrire le circuit **addaccu** en **Stratus** Ce circuit instancie les trois blocs précédents (**mux**, **add** et **reg**). Le circuit **addaccu** prend également comme paramètre le nombre de bits.

- Ecrire un fichier **Makefile** paramétrable permettant de produire chaque composant et le circuit **addaccu**.

### 1.3 Circuit **addsubaccu**

- Maintenant, nous souhaitons que l'**addaccu** puisse effectuer soit des additions, soit des soustractions. Un nouveau paramètre sera donc à apporter pour choisir la fonction à effectuer (Vous avez le choix pour le nom et les valeurs possibles de ce paramètre). Ce nouveau composant sera sur le même schéma que le précédent, avec des modifications à apporter au circuit et/ou les composants. Créer un nouveau composant, appelé **addsubaccu** qui prend en compte cette nouvelle contrainte.
- Ecrire un fichier **Makefile** paramétrable permettant de produire chaque composant et le circuit **addsubaccu**.

## 2 Compte rendu

Vous rédigerez un compte-rendu d'une page maximum pour ce TP. Vous explicitez **en détail** les choix que vous avez fait pour modifier le circuit **addaccu** et/ou ses composants de façon à créer le circuit **addsubaccu**.

Vous fournirez tous les fichiers écrits, avec les **Makefile** permettant d'effectuer la génération des deux circuits (et l'effacement des fichiers générés).