

# TP2 : Modélisation Structurelle VHDL

1. Objectifs
2. A) Génération procédurale des stimuli
3. B) Description structurelle au niveau blocs
4. C) Modélisation Comportementale des blocs
5. D) Simulation logique
6. Compte-rendu

## Objectifs

Le but de cette seconde séance de TP est d'utiliser le langage VHDL pour décrire, puis simuler une description structurelle du composant *addaccu*, présenté dans le premier TP, en utilisant une bibliothèque de cellules pré-caractérisées (en anglais *standard cells library*).

Comme on souhaite construire une description hiérarchique multi-niveaux, on procédera par étape:

- On commencera par décrire le composant *addaccu* comme l'instanciation de trois blocs fonctionnels : le bloc **mux**, le bloc **adder**, et le bloc **accu**.
- On décrira ensuite la structure interne de chacun de ces trois blocs comme un schéma utilisant les cellules de base de la bibliothèque de cellules précaractérisées *SXLIB*.

Un deuxième objectif est d'introduire le langage de description de stimuli **genpat**.

## A) Génération procédurale des stimuli

Dans le premier TP, vous avez écrit "à la main" le fichier *stimuli.pat* décrivant les valeurs à appliquer sur les entrées du circuit. Cette méthode est assez fastidieuse, et elle est source d'erreurs. Pour faciliter la description des scénarios de simulation, vous pouvez utiliser le langage **genpat**. **genpat** est un ensemble de fonctions écrites en langage C, qui apporte au concepteur de circuit toute la puissance d'expression du langage C (boucles, expression conditionnelles, etc.) pour décrire les scénarios de simulation.

On rappelle que vous pouvez obtenir des informations détaillées sur n'importe quel outil de la chaîne de CAO *ALLIANCE* en tapant (par exemple) la commande :

```
>man genpat
```

```
>man AFFECT
```

Les noms des fonctions **genpat** sont en majuscules. La fonction la plus importante du langage **genpat** est la fonction **AFFECT()** qui permet d'assigner une nouvelle valeur à un signal particulier X à une certaine date T. Cette fonction permet donc de spécifier des *événements*. Chaque fonction du langage **genpat** possède son propre man :

```
>man AFFECT
```

Il faut donc écrire un fichier *stimuli.c* respectant la syntaxe du langage C, et c'est l'exécution de ce programme C qui générera le fichier *stimuli.pat* utilisable par **asimut**. On utilise pour cela la commande:

```
>genpat stimuli
```

Voici quelques suggestions utiles pour écrire le fichier *stimuli.c* :

- Ecrire une fonction C indépendante pour le signal d'horloge, qui est très régulier (on conservera une période de 10 ns, avec un rapport cyclique de 50%).
- faites en sorte que la valeur stockée dans l'accumulateur possède une valeur bien définie, en sélectionnant l'entrée a du multiplexeur (au moyen de la commande sel) dans les tous premiers cycles.

Vérifiez que le fichier *stimuli.pat* généré correspond à ce que vous attendez en utilisant l'outil **xpat**.

```
>xpat stimuli
```

## **B) Description structurelle au niveau blocs**

## **C) Modélisation Comportementale des blocs**

## **D) Simulation logique**

## **Compte-rendu**