

# TP3 : Réalisation de la vue " logique" de la partie Controle

1. Architecture de l'Amd2901
2. Réalisation de la partie contrôle
  1. Vérification des fichiers fournis
  2. Génération de la vue structurelle

## Architecture de l'Amd2901

La description générale de l'AMD2901 est donnée par

[?ftp://asim.lip6.fr/pub/amd2901/amd2901.pdf](ftp://asim.lip6.fr/pub/amd2901/amd2901.pdf)

Nous décomposons l'Amd2901 en 2 blocs : la partie contrôle qui regroupe la "glu" logique et la partie opérative (chemin de données).

- Le chemin de données contient les parties régulières de l'Amd2901 c'est à dire les registres et l'unité arithmétique et logique.
- La partie contrôle contient la logique irrégulière, c'est à dire le décodage des instructions

et le calcul des "drapeaux" (indicateurs, ou "Flags").



Les Fichiers fournis sont les suivants :

- squelette de la spécification de la partie contrôle de l'AMD2901
- spécification de la partie chemin de données de l'AMD2901
- fichier PYTHON de la partie chemin de données, à compléter
- description logique du coeur de l'AMD2901
- description logique du circuit contenant les plots et le coeur de l'AMD2901
- le fichier de vecteurs de test de l'AMD2901
- Catalogue des modèles
- Makefile pour automatiser la génération de l'AMD2901

Nous utiliserons la description hiérarchique suivante :



# Réalisation de la partie contrôle

## 1 Vérification des fichiers fournis

Récupérez le fichier CATAL dans votre répertoire de simulation. Il doit contenir les lignes suivantes :

- amd2901\_ctl C
- amd2901\_dpt C

Cela a pour effet d'indiquer au simulateur qu'il faut prendre les fichiers comportementaux (.vbe) de "amd2901\_ctl" et de "amd2901\_dpt". Après avoir complété le squelette du fichier amd2901\_ctl.vbe

Lancer la simulation avec **asimut**:

```
> asimut amd2901_chip pattern resultat
```

Vous pouvez contrôler le résultat en utilisant **XPAT** sur le fichier "resultat".

## 2 Génération de la vue structurelle

On souhaite réaliser la vue structurelle de la partie contrôle de l'Amd2901 à l'aide de la vue comportementale fournie. Pour ce faire on utilise de nouveau **BOOG** qui réalise

la synthèse logique avec les cellules pre-caractérisées de **SXLIB**.

```
> boog amd2901_ctl
```

Valider ensuite la vue structurelle obtenue en simulant le circuit complet avec les vecteurs de test qui vous sont fournis.

On remplacera la vue comportementale de la partie contrôle par sa vue structurelle en ôtant le nom **amd2901\_ctl** du fichier **CATAL**.

```
> asimut -zerodelay amd2901_chip pattern resultat
```

Notez que l'on réalise une simulation "sans délai" de la netlist. En cas de problème(s), n'hésitez pas à utiliser **XPAT**.