



# Beamer avec AMBA

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Lancement</b>	<b>1</b>
<b>3</b>	<b>Objectifs</b>	<b>1</b>
3.0.1	Todo . . . . .	2
3.1	Pilotage des mémoires de points à dessiner . . . . .	2
3.1.1	Todo . . . . .	2
3.2	Vitesse de dessin . . . . .	2
3.2.1	Todo . . . . .	3
3.3	Mode de test du calculateur d'interpolation . . . . .	3
3.3.1	Todo . . . . .	3

## 1 Introduction

Ce laboratoire reprend le circuit d'interpolation et le but est de l'interfacer au bus AMBA. Le circuit utilisé présente une manière de réaliser une interface de bus générique qui peut s'adapter facilement à d'autres systèmes que le bus AMBA-Lite pris comme exemple dans cette série de laboratoires. Le circuit pourrait facilement s'adapter à un autre type de bus parallèle et même à un bus série.

## 2 Lancement

Le circuit se trouve dans la librairie **AhbLiteComponents**. Le banc de test dans la librairie **AhbLiteComponents\_test**.

*Rappel : le programme de modélisation doit être lancé à travers le fichier **systemOnChip.bat**.*

## 3 Objectifs

Le périphérique fournit deux sorties modulées  $\Sigma\Delta$  permettant le dessin d'une courbe sur un oscilloscope



Les registres accédés en écriture par le microprocesseur sont :

- adresse 00 : registre de commande
  - bit 0 : **run**, détermine si une courbe est dessinée ou bien si le système est arrêté
  - bit 1 : **updatePattern**, définit si on charge des nouveaux points dans les mémoires ou si on est en mode de dessin
  - bit 2 : **interpolateLinear**, définit si la ligne qui relie deux points successifs est une droite ou une spline
  - bits de poids fort (MSB) : **patternSize**, aligné à gauche, spécifie le nombre de points de la courbe à dessiner
- adresse 01 : registre de vitesse, donne la période entre le calcul de 2 points consécutifs du dessin
- adresse 02 : registre des points de la courbe en X
- adresse 03 : registre des points de la courbe en Y

Les registres accédés en lecture sont :

- adresse 00 : registre des commande, permet de relire les valeurs écrites précédemment
- adresse 01 : registre des vitesse, permet de relire la valeur écrite précédemment

### 3.0.1 Todo

- Ecrire le code VHDL du décodeur d'adresse **periphAddressDecoder**.
- Ecrire le code VHDL des registres de commande et de vitesse du périphérique **ahbBeamer**.

## 3.1 Pilotage des mémoires de points à dessiner

Les mémoires contenant la liste des points à dessiner font usage de blocs spécifiques de mémoire à l'intérieur du circuit programmable. Le contrôle des mémoires comporte 2 modes distincts :

- en mode **updatePattern**, chaque nouvel accès en écriture au registre concerné déclenche une écriture dans la mémoire et une incrémentation du compteur d'adresse
- en mode de dessin de courbes, chaque nouvelle demande d'un point par le signal **newPolynom** provoque l'incrémentation du compteur d'adresse

Au moment du passage à '1' du signal **updatePattern**, le compteur d'adresse est remis à zéro.

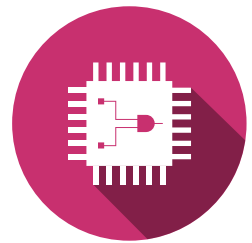
En mode de dessin de courbes, le compteur d'adresse est remis à zéro lorsque le compteur atteint la valeur **patternLength-1**. Ceci n'est pas le cas en mode **updatePattern**.

### 3.1.1 Todo

- Ecrire le code VHDL des blocs de contrôle des mémoires des points des courbes en X et en Y du périphérique **ahbBeamer**.

## 3.2 Vitesse de dessin

Le registre de vitesse permet au périphérique de travailler à une vitesse plus lente que la fréquence d'horloge.



### 3.2.1 Todo

- Ecrire le code VHDL du compteur générant les impulsions **interpolateEn** dans le bloc **periphSpeedController**.

## 3.3 Mode de test du calculateur d'interpolation

Le bouton **selSinCos** permet de dessiner un cercle en mode X-Y indépendamment de toute valeur écrite dans un registre du périphérique.

### 3.3.1 Todo

- Ecrire le code VHDL des de la table générant les valeurs de sinus et de cosinus dans le bloc **sinCosTable**.
- A l'aide du banc de test à disposition, simuler le système et vérifier le bon fonctionnement du périphérique **ahbBeamer**.