



# AMBA System-on-Chip

## Table des matières

1	Introduction	1
2	Lancement	1
3	Objectifs	1
3.1	Ecriture en mémoire . . . . .	2
3.2	Lecture de mémoire . . . . .	2
3.2.1	Todo . . . . .	2

## 1 Introduction

Ce laboratoire définit l'implémentation d'un système à processeur [NanoBlaze](#) comprenant les périphériques développés précédemment. Un tel système intégré sur puce est aussi abrégé **SoC (System on Chip)**.

Le processeur exécute un programme qui lit l'interface série, à travers le périphérique UART, et donne accès au contenu de la mémoire en lecture ou en écriture.

## 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 protocole série est celui utilisé par les collègues d'électronique de puissance pour la communication avec leurs DSPs. Chaque trame se compose comme suit :

- **header** : 0xAA
- **packet id** : numéro de la commande
- **cmd** : code de commande



- **data length** : nombre de bytes du contenu de la commande
- **data** : contenu de la commande
- **checksum** : somme de tous les bytes de la commande, en ignorant les reports

Chaque commande reçoit en retour une trame qui se construit de la même façon. La réponse retourne le même numéro et le même code que la commande, mais avec des données différentes.

### 3.1 Ecriture en mémoire

Pour une écriture d'un mot de 16 bits en mémoire, la trame de commande contient :

- **cmd** : 0x03
- **data length** : 4
- **data** :  $addr_l$ ,  $addr_h$ ,  $data_l$ ,  $data_h$

La réponse contient :

- **data length** : 0
- **data** : none

### 3.2 Lecture de mémoire

Pour une lecture d'un mot de 16 bits de la mémoire, la trame de commande contient :

- **cmd** : 0x04
- **data length** : 2
- **data** :  $data_l$ ,  $data_h$

La réponse contient :

- **data length** : 2
- **data** :  $data_l$ ,  $data_h$

#### 3.2.1 Todo

Le banc de test à disposition, **beamerSoc\_tb**, lit et écrit dans différents registres du système.

- Examiner le contenu du bloc **beamerSoc** et déterminer à quelles adresses sont les registres du périphérique de dessin de courbes.
- Etudier le code VHDL du générateur de stimuli **beamerSoc\_tester** et expliquer la séquence de test.
- Effectuer une simulation du système et vérifier son bon fonctionnement.
- Synthétiser le SoC et vérifier son fonctionnement à l'aide d'un oscilloscope.