

# Projet VHDL

Mehmed Blazevic & Orphée Antoniadis

Hepia 2018

## 1 Introduction

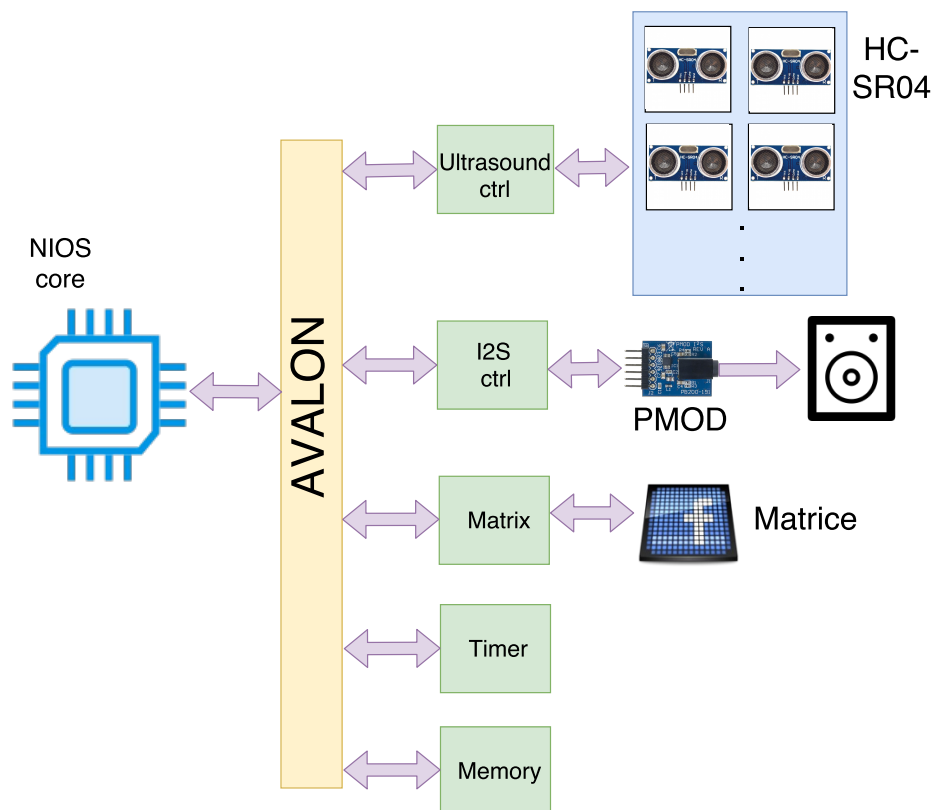
Dans le cadre du cours de VHDL, FPGA & SoPC, il nous a été demandé de réaliser un projet. Le projet s'étale sur tout le semestre et le thème est libre. Nous avons tout de même certaines contraintes :

- Utiliser un périphérique existant (SPI, I2C, UART, ...)
- Créer un périphérique spécifique

## 2 Notre idée

Nous voulions travailler avec la musique. On pensait faire un instrument un peu différent de ce que l'on a l'habitude de voir. L'idée est de travailler avec l'espace. Nous avons décidé d'utiliser des capteurs ultrasons, afin de connaître la distance entre une main et ce capteur. En fonction de la distance que l'on mesure avec la main, nous avons dans l'idée de produire une notes de musique plus ou moins forte. L'idée était de placer 8 capteurs de distances alignés, afin de jouer avec un piano "virtuel".

### 2.1 Schema



## 3 Drivers

### 3.1 Capteur ultrasons

Le module ultrason que nous avons utilisé est le HCSR04 de Cytron Technologies. Ce module est composé de 4 pins, le Vcc, le Gnd, une entrée appelée trig et une sortie appelée echo. Il utilise son propre protocole qui très simple. Lorsque l'on souhaite obtenir la distance, il faut mettre la pin trig à l'état haut pendant  $10\mu s$ . Il faut ensuite remettre le trig à l'état bas, le module envoie une onde ultrason et met la pin echo à l'état haut. L'écho est remis à l'état bas lorsque l'onde ultrason a fait l'allé-retour. Pour obtenir la distance, il faut donc mesurer le temps que l'écho a été à l'état haut. A noter que le module ultrason fonctionne en 5V et la DE0 fonctionne en 3V3. Nous avons donc du faire un petit diviseur de tension sur la pin echo. Le trig étant une entrée, les 3V3 de la FPGA suffisent à considérer le signal comme étant à l'état haut.

Le driver que nous avons développé reproduit le comportement décrit ci-dessus. Nous avons utilisé une machine à états pour le reproduire. Pour éliminer le bruit, nous faisons une moyenne sur 100 échantillons. L'interface avalon développée est très simple elle aussi. Il n'y a qu'un seul registre en lecture seule. Ce registre contient le nombre de coups d'horloges de la DE0 pendant lesquels l'écho était à l'état haut. Au niveau du code C il suffit de convertir ces coups d'horloge et utiliser la vitesse du son afin d'obtenir la distance entre le module et l'obstacle. Notre driver fonctionnait mais nous avons remarqué que la distance calculée n'était pas très précise mais assez pour notre projet.

### 3.2 I2S