

## Laboratoire 5: Utilisation des I/O de la partie FPGA via le HPS

Département: **TIC**

Unité d'enseignement: **SCF**

Auteur(s):

- **CECCHET Costantino**

Professeur:

- **DASSATTI Alberto**
- **YANN Thomas**

Assistant:

- **JACCARD Anthony I.**

Date:

- **16/04/2024**

*[Page de mise en page, laissée vide par intention]*

## Introduction

Pour ce laboratoire On a du installer Linux sur la DE1-SoC.

## Partie 1 : Verification du bon fonctionnement de linux sur la DE1-SoC

Pour cette partie nous avons du dans un premier temps utiliser l'image linux fournie pour verifier que la DE1-SoC fonctionne correctement.

Une fois l'image extraite il a fallu la copier sur une carte SD.

Cette commande permet de copier l'image sur la carte SD en utilisant un buffer de 4M ce qui permet de copier l'image plus rapidement et le status=progress permet d'afficher la progression de la copie.

**ATTENTION: Il est important de bien verifier que le device /dev/sdX correspond bien a la carte SD et pas a un autre device.**

Pour cela on a utilisé la commande suivante:

```
sudo dd if=DE1_SoC_SD.img of=/dev/sdX bs=4M status=progress
```

Une fois la copie terminée on a inséré la carte SD dans la DE1-SoC et on a démarré la carte.

On peut donc se connecter a la carte en utilisant un cable USB et un terminal série avec la commande suivante:

```
$ picocom -b 115200 /dev/ttyUSB0 -r -l
```

ici encore il est important de verifier que le device /dev/ttyUSB0 correspond bien au cable USB et pas a un autre device.

## Changement de version de linux

Pour cette partie nous avons du changer la version de linux sur la DE1-SoC.

Pour cela on a du compiler le kernel linux et le charger sur la carte.

Il a fallut tout d'abord telecharger le kernel linux sur le site officiel de linux.

Ensuite on a du compiler le kernel en utilisant la commande suivante:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- socfpga_defconfig  
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j8
```

ceci permet de compiler le kernel pour la DE1-SoC.

Une fois le kernel compilé on peut remplacer le kernel sur la carte SD par le nouveau kernel.

Pour cela on a du monter la partition boot de la carte SD et remplacer le kernel par le nouveau kernel.

Une fois le kernel remplacé on a pu redémarrer la carte et vérifier que le nouveau kernel fonctionne correctement.

## Modification du device tree

Il faut aller dans le dossier arch/arm/boot/dts et cree un fichier .dts pour la DE1-SoC.

On peut partir sur la base du fichier socfpga\_cyclone5\_de0\_sockit.dts qui est fourni avec le kernel.

On enlvera les GPIO et le noeud I2C car ils ne sont pas utilisés.

On ajoute ensuite le noeud pour le bridge HPS-FPGA.

```
&fpga_bridge0 {  
    reg = <0xff200000 0x100000>;  
    status = "okay";  
};
```

Ceci permet de lier le bridge HPS-FPGA au kernel.

On a ensuite du compiler le device tree en utilisant la commande suivante:

```
make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- -j8
```

Une fois le device tree compilé on a du remplacer le device tree sur la carte SD par le nouveau device tree.

Pour cela on a du monter la partition boot de la carte SD et remplacer le device tree par le nouveau device tree.

Une fois le device tree remplacé on a pu redémarrer la carte et vérifier que le nouveau device tree fonctionne correctement.

## INTEL FPGA Monitor

Une fois le boot réussi on a pu utiliser le INTEL FPGA Monitor pour charger notre ancien labo sur la carte.

Il a suffit de faire actions -> Download system.

une fois effectuée on peut tester notre IP

il faut tout d'abord compiler devmem2.c en utilisant la commande suivante:

```
arm-linux-gnueabihf-gcc-6.4.1 devmem2.c -o devmem2
```

puis on peut lire et ecrire dans les registres de l'IP en utilisant la commande suivante:

```
./devmem2 0xff200000 w 0x00000001
```

en voici le resultat:

```
root@socfpga:~# ./devmem2 0xff200000 w 0x00000001  
/dev/mem opened.  
Memory mapped at address 0xb6f9a000.  
Value at address 0xFF200000 (0xb6f9a000): 0xBADB100D  
Written 0x1; readback 0xBADB100D
```

## portage de l'application sur la DE1-SoC

Il faut maintenant porter l'application sur la DE1-SoC.

Il a fallut modifier les lectures et ecritures pour utiliser le même principe que devmem2.

C'est a dire utiliser le fichier /dev/mem pour lire et ecrire dans les registres de l'IP.

un read et un write utilisant le mmap pour lire et ecrire dans les registres de l'IP.

Puis on a du compiler l'application en utilisant la commande suivante:

```
arm-linux-gnueabi-gcc-6.4.1 -o <nomexec> <programme.c>
```

le copier sur la carte SD et le lancer sur la carte.

```
./<nomexec>
```

## Conclusion

Ce laboratoire nous a permis de voir comment installer linux sur la DE1-SoC et comment utiliser le bridge HPS-FPGA pour communiquer avec l'IP depuis linux.