

Module d'approfondissement SEEE

Systèmes d'exploitation et environnements d'exécution embarqués
Prof. Daniel Rossier / Magali Fröhlich

Emulation de périphériques

Labo reptar2 (v2017.1.6)

Dans le cadre de ce laboratoire, une émulation de quelques périphériques simples de la plate-forme REPTAR sera réalisée dans l'environnement *Qemu*. Cette émulation sera ensuite reprise et utilisée dans les laboratoires suivants (*drivers*, *virtualisation*).

Consignes de laboratoire

- Ce laboratoire est à faire en binôme.
- Un rapport détaillé doit être élaboré par groupe et sera évalué.

Prologue – Mise à jour du workspace

- Assurez-vous que les dossiers *7seg_u-boot* et *miniapp_u-boot* sont présents dans *seee_student*.
- Ouvrez *Eclipse*.
- Dans la zone blanche du *Project Explorer*, faites un clic droit et choisissez *Import*.
- Allez dans la section *General* et choisissez *Existing Projects into Workspace*.
- Cliquez sur *Next*.
- Cliquez sur le bouton *Browse* à droite de *Select root directory*, sélectionnez le dossier *7seg_u-boot* et cliquez sur *OK*.
- Cliquez sur *Finish*.
- Répétez les étapes **c** à **g** pour *miniapp_u-boot*.

N'hésitez pas à vérifier que les deux projets importés se compilent bien.

Etape no. 1 – Environnement *Qemu* et machine *REPTAR*

Cette étape vous permet de vous familiariser avec l'environnement que nous utiliserons pour l'émulation de périphériques.

Dans cette étape, il est nécessaire de travailler avec l'application graphique *Qtemu*, qui constitue le *frontend* graphique de *Qemu*. L'application est développée en C++ et utilise la librairie Qt.

- A partir du répertoire *seee_student*, lancez le *frontend* graphique avec le script *stq* :

```
$ ./stq
```

b) Les fichiers sources de Qemu se trouvent dans le répertoire *qemu-reds*. Examinez les fichiers suivants :

- | | |
|---|---|
| • <i>hw/arm/reptar/reptar.c</i> | Emulation plate-forme REPTAR |
| • <i>hw/arm/reptar/reptar_sp6.c</i> | Emulation de la FPGA |
| • <i>hw/arm/reptar/reptar_clcd.c</i> | Emulation gestion du LCD4x20 |
| • <i>hw/arm/reptar/reptar_sp6_buttons.c</i> | Emulation gestion des boutons |
| • <i>hw/arm/reptar/reptar_sp6_emul.c</i> | Gateway entre Qemu et qtemu |

Vous trouverez également toute la documentation nécessaire sur la plate-forme Reptar dans le répertoire *doc*.

c) La compilation de Qemu pourra s'effectuer dans le répertoire *qemu-reds* directement, avec la commande `make` (utilisez `make -j4` ou `-j8` pour aller plus vite !).

Avant la toute première compilation, il est nécessaire de lancer le script *configure* avec la commande suivante :

```
$ ./configure --target-list=arm-softmmu --enable-debug --disable-attr --disable-docs
$ make -j8
```

(Vous pouvez retrouver la commande d'appel à *configure*, avec les paramètres adéquats, dans *qemu-reds/README*).

Par la suite, toute modification de l'émulateur nécessitera la recompilation avec la commande `make`. Vous pourrez également supprimer toutes les productions intermédiaires avec `make clean` (`make distclean` si vous souhaitez effacer la pré-configuration ; il faudra relancer le script *configure*).

D'autre part, il est possible de *debugger* l'émulateur *Qemu* en lançant d'abord *qtemu* seul (`./qemu/qtemu`), puis l'émulateur depuis *Eclipse* (une configuration de *debug* a été préinstallée).

Etape no. 2 – Emulation de la FPGA Spartan6

Dans cette étape, il s'agit de mettre en place la structure nécessaire à l'émulation de la FPGA intégrée à la plate-forme. La FPGA implémente des registres associés aux périphériques externes. Dans cette étape, il s'agit de s'assurer que l'accès aux adresses I/O en lecture et écriture fonctionne.

a) Complétez l'émulation de la FPGA afin de tester l'écriture et la lecture à l'une ou l'autre adresse dédiée à la FPGA (affichez simplement un message).

b) Testez les accès en lecture-écriture avec *U-Boot*.

Indications:

- Les fichiers à adapter sont les suivants : *reptar_sp6.c* et *reptar.c*
- Il faut adapter le code de la plate-forme (*retpar.c*) afin "d'instancier" le nouveau périphérique (cf cours).
- Consultez la documentation de la plate-forme REPTAR pour obtenir les informations relatives à la FPGA.

Etape no. 3 – Emulation des devices de type LED (output)

La FPGA est connectée à des LEDs qui sont visibles sur l'interface graphique. Cette étape consiste à implémenter le code d'émulation précédent afin de gérer l'accès aux LEDs reliées à la FPGA. Les interactions entre la FPGA et l'interface graphique doivent être gérées proprement.

Indications

- Il faut gérer un état interne, associé à la FPGA, pour maintenir les valeurs des registres.
- La communication avec le *frontend* graphique s'effectue à l'aide des fonctions définies dans le *Guide d'utilisation de l'infrastructure des salles A07/A09*.
- La commande `itbok` de U-Boot permet de lancer différents tests sur la plate-forme *Reptar*. Essayez le test no. **10**.

Etape no. 4 – Emulation de type boutons (input)

La FPGA est connectée à une série de boutons (*switches*) sur la plate-forme *Reptar*. Cette étape consiste à mettre en place la structure nécessaire à la gestion de ces boutons.

- a) Adaptez les fichiers nécessaires afin que l'émulation de votre périphérique (FPGA) puisse détecter la pression d'une touche, sans vous préoccuper pour le moment des interruptions.
- b) Le projet *sp6_buttons_u-boot* contient une application permettant de tester vos boutons (en mode *polling*). Compilez l'application et effectuez quelques tests.

Indications

- Les fichiers à adapter sont les suivants : *reptar_sp6.c* et *reptar_sp6_buttons.c*.
- Utilisez le *Guide d'utilisation de l'infrastructure des salles A07/A09* pour découvrir comment intercepter la pression d'un bouton avec le front end graphique.
- Utilisez la documentation sur les registres de la FPGA Spartan 6 de la plate-forme afin d'utiliser les bons registres.

Etape no. 5 – Gestion des interruptions (IRQ) avec les boutons

Complétez votre émulateur avec le code nécessaire à la gestion d'une interruption à niveau émise par la FPGA lorsqu'un bouton est pressé. L'interruption est censée être acquittée par le *driver*. Il faut donc gérer l'état interne associé à cette interruption.

- a) Commencez par adapter le code d'initialisation de la plate-forme (*reptar.c*) afin d'instancier une interruption en provenance de la FPGA ; l'interruption sera de type *niveau*.
- b) Testez que l'interruption fonctionne en configurant le contrôleur *GPIO* et en interrogeant le registre d'état, dans U-Boot. Les registres du microcontrôleur à utiliser sont les suivants :

GPIO_RISINGDETECT, GPIO_IRQENABLE1 et GPIO_IRQSTATUS1

De plus, l'interruption doit aussi être activée au niveau de la FPGA (cf documentation).

Indications

- Les fichiers à adapter sont les suivants : *reptar_sp6.c*, *reptar_sp6_buttons.c* et *reptar.c*.
- La ligne d'interruption de la FPGA est reliée à la *GPIO 10*.
- Vous trouverez les informations relatives au contrôleur *GPIO* du microcontrôleur *DM-3730* dans le document *AM/DM37x Multimedia Device - Technical Reference Manual*, disponible dans le répertoire *doc/reptar/components/cpu* (pages 3'517 et suivantes).

Etape no. 6 – Emulation de l'afficheur 7 segments

La FPGA est connectée à un afficheur 7 segments, visible sur l'émulateur. Cette étape consiste à mettre en place la gestion de cet afficheur 7 segments.

- a) Adaptez les fichiers nécessaires afin que l'émulation de votre périphérique (FPGA) puisse gérer les trois digits de l'afficheur 7 segments.
- b) Le dossier *7seg_u-boot* contient une application permettant de tester l'afficheur 7 segments : les chiffres de 0 à 9 doivent défiler progressivement : 012, puis 123, 234, 456, 567, ..., 901, 012, etc. Compilez l'application et effectuez quelques tests.

Indications

- Le fichier à adapter est le suivant : *reptar_sp6.c*
- Utilisez le *Guide d'utilisation de l'infrastructure des salles A07/A09* pour découvrir comment utiliser l'afficheur 7 segments sur le front end graphique.
- Utilisez la documentation sur les registres de la FPGA *Spartan 6* de la plate-forme afin d'utiliser les bons registres.

Etape no. 7 – Mini-application utilisant les boutons et l'afficheur 7 segments

- a) Le dossier *miniapp_u-boot* contient un chablon. Complétez-le afin de créer une application qui utilise les boutons SW2, SW5, SW4 et SW3, ainsi que l'afficheur 7 segments.
 - Lors d'un appui sur SW2, SW5 ou SW4, le digit respectivement à gauche, au centre ou au milieu est incrémenté de 1, modulo 10. Si un digit atteint 9, il reviendra à 0.
 - La valeur initiale de chaque digit, au démarrage de l'application, est 0 (on affichera 000).
 - Un appui sur SW3 quitte l'application.
 - Vous devrez gérer l'anti-rebond : le digit ne devra être incrémenté que si le bouton est pressé puis relâché (comme un appui sur une touche de sonnette par exemple).
- b) Testez votre application sur l'émulateur.
- c) Déployez et testez votre application sur la plate-forme réelle.

Indications

- Inspirez-vous des codes fournis pour les étapes 4 et 6.