

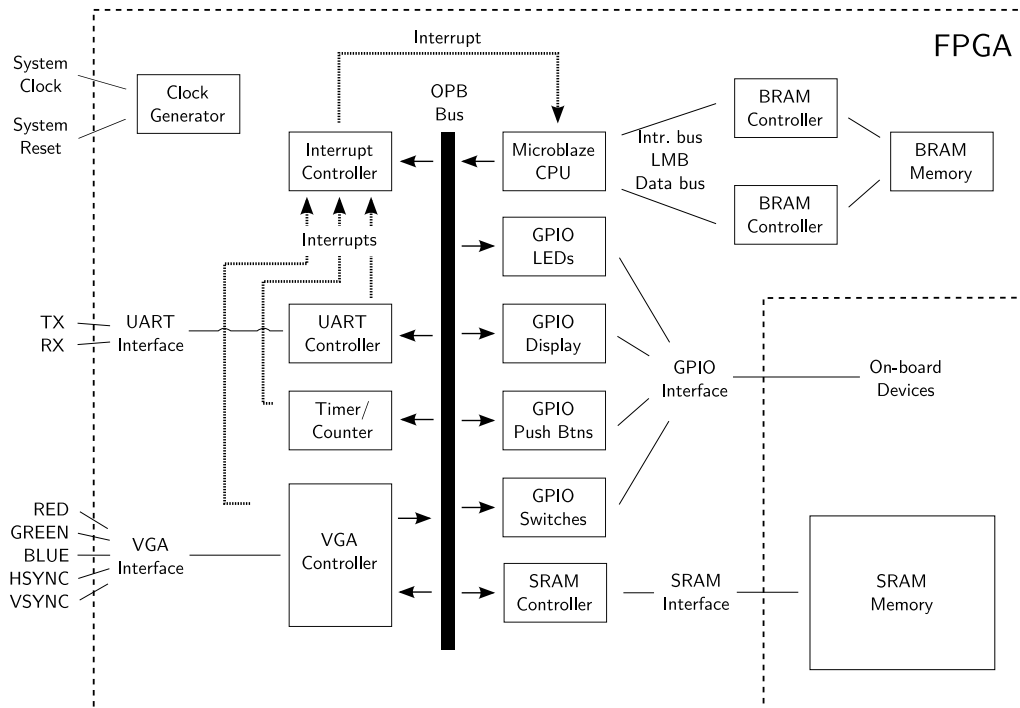
Modélisation TLM en SystemC

TP n°3 : Demo de la plate-forme FPGA

Introduction

Voici la plate-forme que vous allez modéliser en TLM dans ce TP.

La plate-forme est composée d'une carte Spartan-3 de Digilent avec un circuit intégré FPGA Xilinx sur lequel nous avons synthétisé plusieurs composants matériels. On y trouve notamment un processeur généraliste type Microblaze, des mémoires, un timer, un contrôleur écran VGA et des boutons poussoirs.

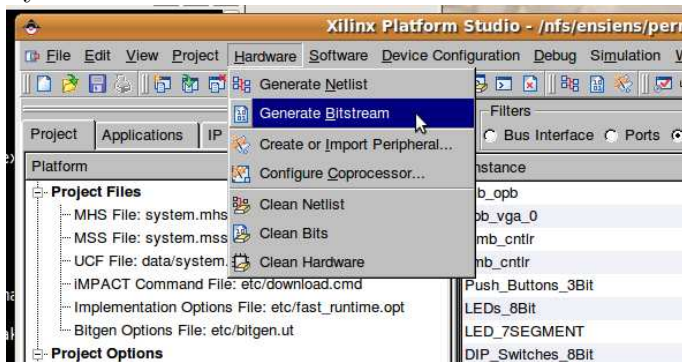


Certains composants de la carte ne seront pas utilisés par le logiciel embarqué, et donc ne seront pas modélisés en TLM.

Synthèse de la plate-forme matérielle

L'étape de synthèse du matériel consiste à générer les composants matériels qui seront dans le FPGA.

- Placez-vous dans le répertoire de la plateforme : `cd cours-tlm/TPs/tp3/`.
- Lancez EDK : `~moy/bin/startEDK-8.1i.sh` (**attention**, la version installée par défaut de startEDK ne permet pas la programmation du FPGA¹, bien prendre celle se trouvant dans `~moy/bin`)
- Ouvrez le projet `fpga/system.xmp`.
- Synthétisez le matériel :

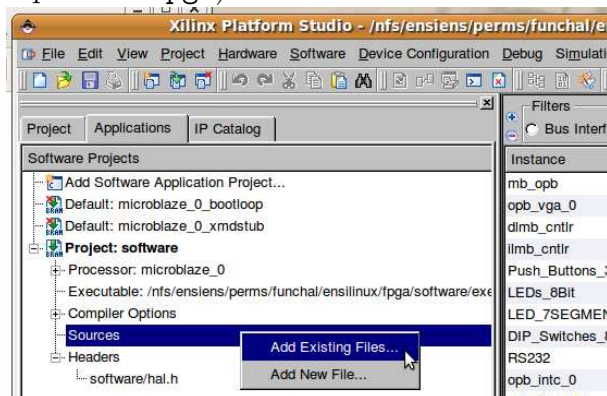


La synthèse prend plusieurs minutes, profitez-en pour lire la suite du sujet, brancher le FPGA, ...

Compilation du logiciel

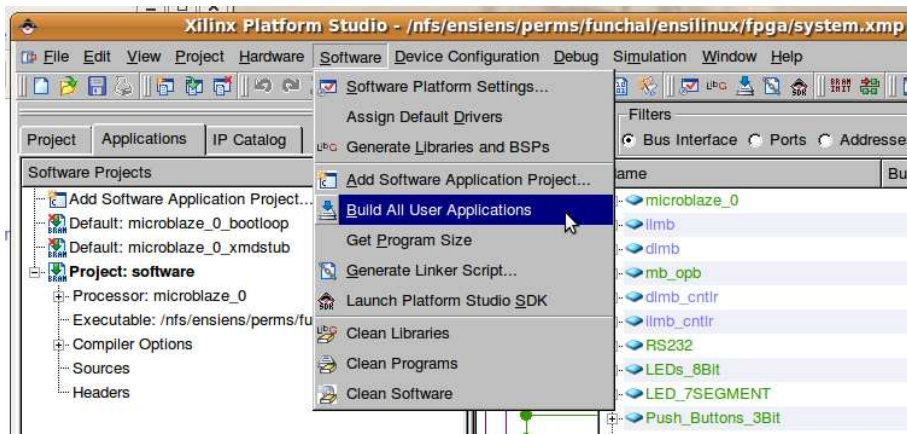
Le logiciel utilisé est le même que vous allez ensuite faire tourner dans la plate-forme TLM.

- Rajoutez au projet le source `main.c` (dans le répertoire `tp3/software/`, en dehors du répertoire `fpga`) :



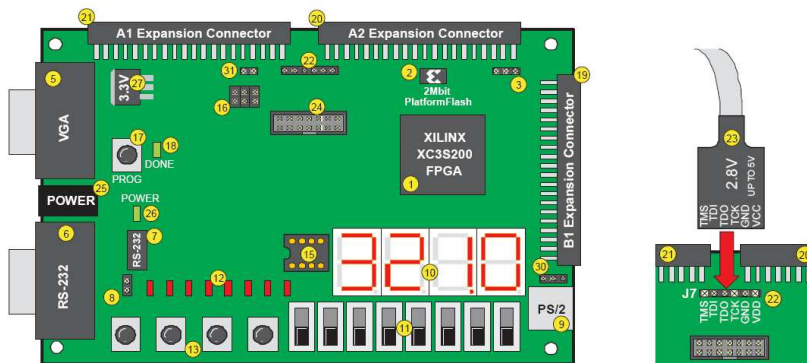
- Compilez le logiciel :

1. Le problème est lié à la compatibilité 32 bits (installation d'EDK) / 64 bits (reste de l'installation du système). Pour la programmation du FPGA via impact, il est nécessaire de charger une bibliothèque via `LD_PRELOAD`, mais positionner `LD_PRELOAD` pendant la compilation du logiciel embarqué casse la compilation.



Téléchargement vers le FPGA

- Branchez l'écran, le lien série, le câble de programmation JTAG et l'alimentation.



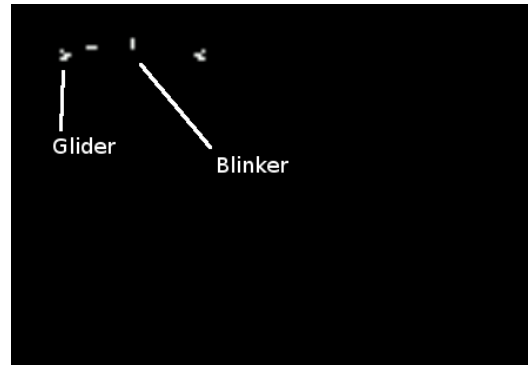
- Dans une autre fenêtre de terminal, lancez le script `fpga/serial` pour monitorer le lien série. Utilisez `~.` (tilde point) pour déconnecter.
- Téléchargez le bitstream vers le FPGA :



Le jeu de la vie

Si vous avez réussi, l'écran doit afficher le jeu de la vie (http://en.wikipedia.org/wiki/Conway%27s_Game_of_Life), en partant avec deux gliders, deux blinkers et un point (qui meurt à la première itération). Vous pouvez confirmer aussi que le lien série affiche les `printfs` correctement. Si vous appuyez sur la touche `BTN0`, deux gliders seront rajoutés. Essayez d'appuyer sur `BTN3` (soft reset) et `PROG` (hard reset).

Modification et recompilation du logiciel embarqué



Regardez le code source du logiciel (`software/main.c`), et si le temps le permet, modifiez ce fichier et décommentez la ligne `draw_gun(0, 30)` ; au début de la fonction `__start()`, puis recompilez et reprogrammez le FPGA. Vous devriez voir le « glider gun » apparaître (un motif classique du jeu de la vie, cf. http://en.wikipedia.org/wiki/Gun_%28cellular_automaton%29 pour les détails).

Détails techniques sur la plate-forme

Contenu de la carte

La carte Digilent Spartan-3 Starter Board contient :

- une puce FPGA Xilinx XC3S1000-4FT256 ;
- un mémoire SRAM asynchrone 1M octets ;
- un port VGA ;
- un port RS232 (série) ;
- un port PS/2 (non utilisé) ;
- 4 displays 7-segments ;
- 8 interrupteurs ;
- 8 leds ;
- 5 boutons poussoirs (2 utilisés pour softreset/hardreset, 3 libres) ;
- un oscillateur à 50MHz.
- des régulateurs de tension, des ports d'expansion, etc...

Les composants synthétisés sur FPGA

- un processeur Microblaze 32-bit sans pipeline ni cache ;
- un générateur d'horloge ;
- un timer ;
- deux contrôleurs de mémoire interne, un pour les instructions, un pour les données ;
- un bloc de mémoire interne type BRAM de 16K octets ;
- un contrôleur d'interruptions ;
- un contrôleur série (UART) ;
- un contrôleur écran VGA ;
- un contrôleur mémoire externe type SRAM asynchrone ;
- quatre modules entrée/sortie (GPIO) ;

L'espace d'adressage (vue Microblaze)

Composant	Adresse de début	Taille	Dans la plate-forme TLM ?
Mémoire BRAM (interne)	0x00000000	0x00004000	oui
Mémoire SRAM (externe)	0x20100000	0x00100000	oui
GPIO (boutons poussoirs)	0x40000000	0x00010000	oui
GPIO (leds)	0x40020000	0x00010000	non
GPIO (display)	0x40040000	0x00010000	non
GPIO (interrupteurs)	0x40060000	0x00010000	non
Contrôleur série (UART)	0x40600000	0x00010000	non
Contrôleur d'interruptions	0x41200000	0x00010000	partiellement
Timer/Compteur	0x41C00000	0x00010000	oui
Contrôleur VGA	0x73A00000	0x00010000	oui