

Rapport IOT

Thomas Lerchundi M2PGI

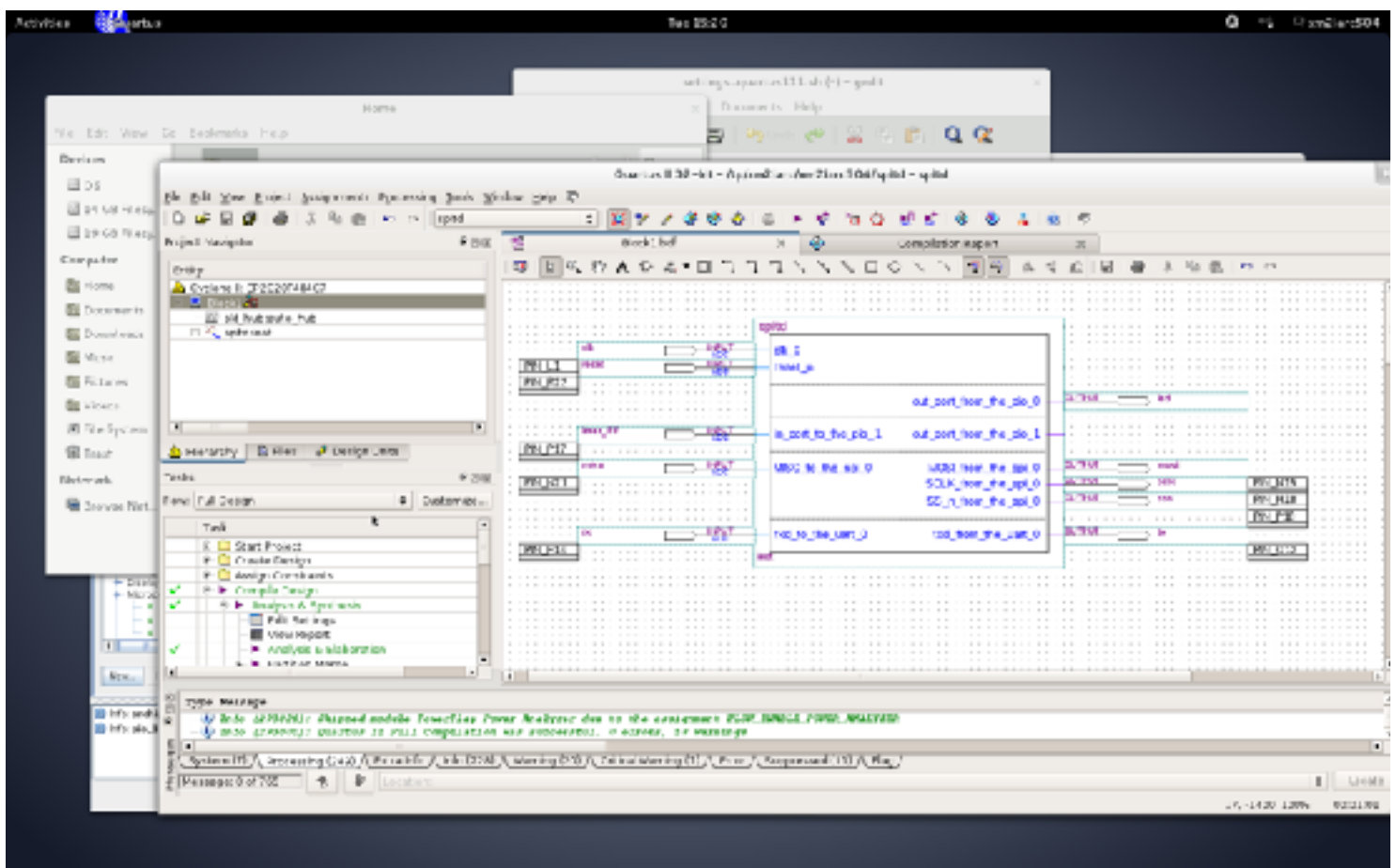
1. Synthétise du processeur du processeur

Dans cette section, nous avons rigoureusement suivi les indications fournis en Tp (pour la partie configuration). Nous avons utilisé FPGA afin de synthétiser un processeur Nios 2.

Premièrement , en spécifiant les différents composants du processeur(mémoire, interface SPI...), nous avons pu généré le processeur.

Il s'agissait ensuite d'ajouter l'ensemble des entrées/sorties du micro-processeur, puis nous avons pu le générer.

Voici-ci contre le diagramme obtenu :



2. Programmation de la FPGA

Nous avons réalisé la reprogrammation des protocole de lecture et d'écriture sur l'interface SPI.

voici les fonctions programmée :

```
13 // Completer la fonction de lecture SPI
14 static int ReadSPI_0(alt_u8 adresse){
15     alt_u8 data_miso; // contiendra la donnée lue
16     //Instruction de demande de lecture via TXDATA
17     alt_u16 seq = (adresse<<8)&0x3F00;
18     IOWR_ALTERA_AVALON_PIO_DATA(SPI_0_BASE+4,seq );
19     while(IORD_ALTERA_AVALON_PIO_DATA(SPI_0_BASE+8)&0x0020 ==0 ){
20         // Attente active tant que le TMT n'indique pas que la donnée peut être lue
21     }
22     //Lecture effective des données.
23     data_miso =IORD_ALTERA_AVALON_PIO_DATA(SPI_0_BASE);
24 }
25 return data_miso;
26 }
27
28 // Completer la fonction d'écriture SPI
29 static void WriteSPI_0(alt_u8 adresse, alt_u8 data_mosi){
30     //Construction de la séquence MOSI
31     alt_u8 entete=(adresse&0x3F)+0x80;
32     alt_u16 seq = (entete << 8) + data_mosi;
33     while(IORD_ALTERA_AVALON_PIO_DATA(SPI_0_BASE+8)&0x0020 ==0 ){
34         // Attente active tant que le TMT n'indique pas que la donnée peut être écrite
35     }
36     //Instruction de demande de lecture via TXDATA
37     IOWR_ALTERA_AVALON_PIO_DATA(SPI_0_BASE+4,seq );
38 }
39 }
```

Cette partie fut très intéressante, notamment par le fait de comprendre le fonctionnement des registres TXDATA, RX DATA et STATUS. Elle nous a permis de comprendre dans quelle circonstances ces registres était réellement utiliser dans le cadre du protocole SPI.

De plus, ce fut l'occasion de (re)découvrir l'utilisation d'opérations binaire en C.

Cependant, Faute de temps (et d'anglais). Nous n'avons pas eu le temps de tester le programme sur la carte dans le monde physique. Il aurait été intéressant de voir le déroulement des communications avec le module CYPRESS.

Conclusion/Impressions Personnelles

De manière général, il fut réellement intéressant de synthétiser un micro-processeur, et de le charger sur une vraie carte en utilisant les brochages adéquats. De plus, comprendre le fonction des registres dans l'interface SPI fut très instructif.

Cependant, il reste une certaine amertume, du fait de ne pas avoir pu tester notre programme par manque de temps. Si l'organisation le permet pour les années à venir, il serait intéressant de consacrer un créneau de plus à ce TP, car la prise en main et la configuration initiale requiert un temps considérable.