

Laboratoire CSF

Prise en main des outils de simulation, synthèse et placement-routage

semestre printemps 2016 - 2017

Introduction

Ce premier labo vise la prise en main des outils de simulation et de synthèse. Nous y réaliserons une petite Unité arithmétique et logique, la simulerons et observerons le résultat en synthèse.

Cahier des charges

L'unité arithmétique et logique doit être capable d'effectuer 8 opérations, sur 2 vecteurs de 8 bits.

Les entrées sont :

| Nom | Taille | Description |
|----------|--------|-------------------|
| a_i | 8 | Premier opérande |
| b_i | 8 | Deuxième opérande |
| mode_i | 3 | Mode d'opération |
| carry_i | 1 | Retenue en entrée |
| carry_o | 1 | Retenue en sortie |
| result_o | 8 | Résultat |

Les résultats attendus sont les suivants :

| mode_i | s_o | c_o |
|--------|---|----------|
| "000" | a_i + b_i | retenue |
| "001" | a_i - b_i | retenue |
| "010" | a_i or b_i (bit par bit) | indéfini |
| "011" | a_i and b_i (bit par bit) | indéfini |
| "100" | a_i | indéfini |
| "101" | b_i | indéfini |
| "110" | result_o(0) = '1' si a_i vaut b_i sinon '0' | indéfini |
| "111" | 0 | indéfini |

Etape 1

Dessinez le schéma de cette ALU, en exploitant les blocs de base (portes logiques, mux, demux, décodeur, encodeur, comparateur, additionneur, ...).

Etape 2

Reprenez le code fourni et complétez-le pour effectuer les opérations 2 à 7 en écrivant le code de manière à décrire le schéma de l'étape 1. Utilisez le banc de test fourni pour stimuler votre système.

Pour ce faire, lancez le logiciel QuestaSim et placez-vous dans votre répertoire de travail. Ceci se fait via la console de QuestaSim, en utilisant la commande `ls`. La commande `pwd` vous indique votre répertoire courant. Allez ensuite dans le répertoire `sim`. Là, tapez la commande `do ../scripts/sim.do`. Ceci va lancer la compilation de votre fichier, ainsi que la simulation. Observez le chronogramme pour voir si le résultat attendu est bien celui que vous aviez prévu.

Etape 3

Votre code fonctionnant en simulation, nous allons le synthétiser en utilisant le logiciel Quartus 16 (Attention, pas Quartus II), d'Altera. Nous allons intégrer le système sur une carte DE1-SoC. Pour ce faire, créez un nouveau projet Quartus, sélectionnez votre répertoire `synth` et nommez le projet `de1_top`. Sélectionnez les fichiers sources VHDL fournis ainsi que votre fichier source `alu.vhd`. Sélectionnez parmi la famille Cyclone V le device 5CSEMA5F31C6 et dans l'onglet correspondant la board DE1-SoC. Une fois ces informations fournies la création du projet est terminée. Lancez maintenant la compilation (*Start Analysis & Synthesis, ctrl-k*) de vos fichiers sources.

Observez les erreurs potentielles, ainsi que les warnings. L'analyse des warnings est toujours importante, car certains peuvent être d'une importance capitale.

Vous devriez maintenant avoir la possibilité d'observer la quantité de ressources nécessaires à votre `alu`. (Dans *Analysis & Synthesis / Resource Utilization by Entity*).

Pour observer le schéma généré, sélectionnez *Tools/Netlist Viewers/RTL Viewer*. Que voyez-vous ? Pour observer le schéma généré, sélectionnez *Tools/Netlist Viewers/Technology Map Viewer (Post-Mapping)*. Que voyez-vous ?

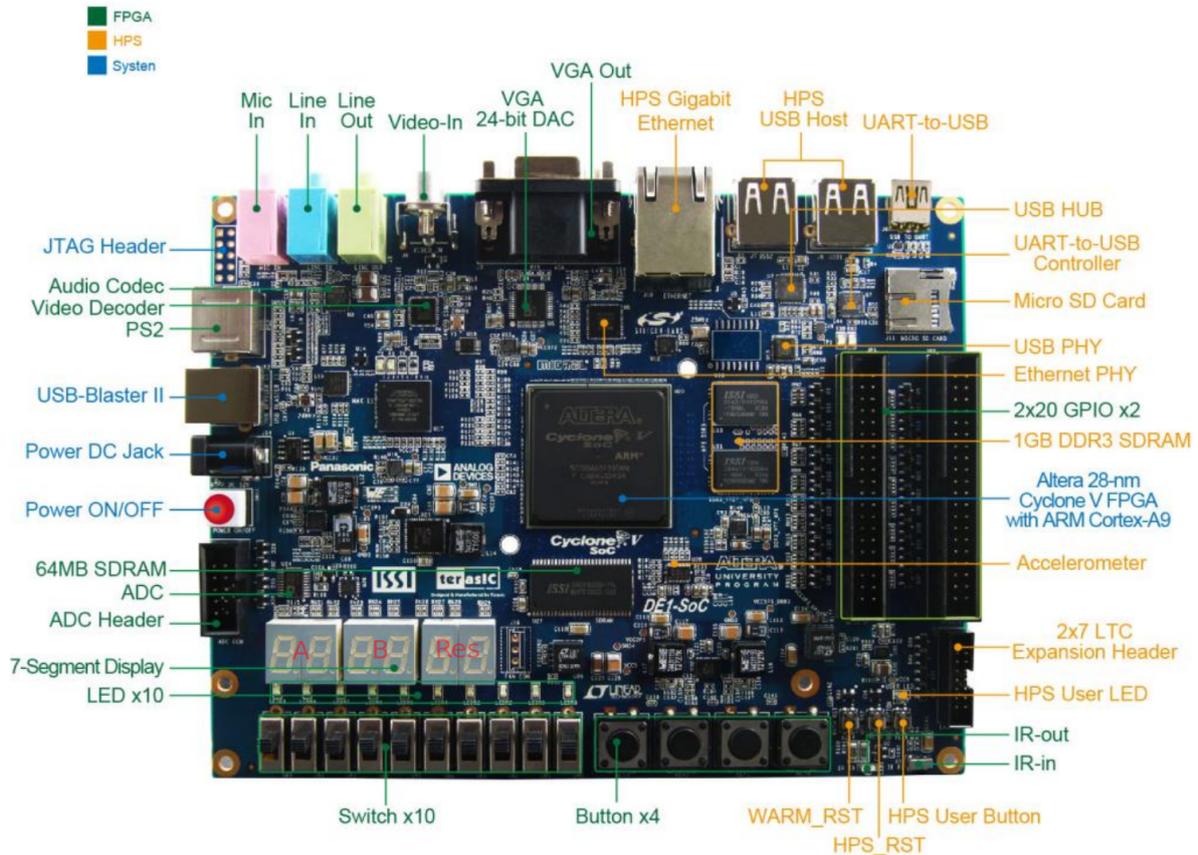
Quelle est la différence entre les deux schémas ? (entre les deux vues).

Etape 4

Ajoutez les opérations arithmétiques à votre design. Pour ce faire, exploitez la bibliothèque `ieee.numeric_std` en ajoutant `use ieee.numeric_std.all;` au début de votre fichier. Complétez les étapes 1 et 2.

Etape 5

Maintenant que votre ALU est prête, vous allez l'intégrer et la tester sur la FPGA. Il va falloir commencer par assigner les entrées-sorties de la carte, nous avons déjà configuré la plupart de celles-ci et vous pouvez importer le fichier `de1_top_pins.csv` avec le menu *Assignments/Import Assignments...* Ouvrez maintenant le **Pin Planner** avec le menu *Assignments/Pin Planner* ou l'icône correspondante et complétez les "**Location**" des pins non assignées à l'aide du manuel de la DE1-SoC, chapitre 3.6.1 User Push-Buttons, Switches and LEDs. (Le manuel est disponible sur la page du fabricant, terasIC). Il s'agit des entrées `push_buttons_i` du code VHDL `de1_top.vhd` et ces entrées doivent être assignés aux boutons poussoirs de la carte de développement.



La fonction des boutons est la suivante :

- `push_buttons_i[3]` : charge la valeur choisie grâce aux switches 0 à 7 dans le registre A et l'affiche.
- `push_buttons_i[2]` : charge la valeur choisie grâce aux switches 0 à 7 dans le registre B et l'affiche.
- `push_buttons_i[1]` : effectue l'opération choisie avec les switches 0 à 2 et affiche le résultat.
- `push_buttons_i[0]` : est le reset du système.

Compilez (Synthèse, Placement & Routage cette fois-ci) votre projet.

Observez-vous des *warnings* concernant des timings ? Il semble que votre système ne puisse fonctionner à la fréquence désirée par Quartus. Pour modifier cette contrainte, ouvrez l'utilitaire permettant de définir les contraintes de timing, via le menu *Tools->TimeQuest Timing Analyzer*. Vous pouvez indiquer à quelle fréquence d'horloge votre système doit pouvoir fonctionner. Choisissez une valeur raisonnable.

Relancez le placement-routage et observez les *warnings*. Tout devrait être OK. Vous êtes prêts à programmer la FPGA. Ceci se fait depuis le menu *Tools/Programmer* ou via l'icône correspondante.

Dans le menu *programmer* il faut que la carte soit reconnue et sélectionnée, si ce n'est pas le cas "No Hardware" est affiché, il faut aller dans *Hardware Setup...* et choisir le hardware, normalement nommé DE-SoC. Une fois le hardware sélectionné, cliquez sur *Auto Detect* et sélectionnez le device **5CSEMA5** qui correspond à votre FPGA. Une fois le device sélectionné cliquez sur *Change File ...* et spécifiez le fichier *.sof* généré par votre compilation (dans *output_files*). Cochez la case *Program/Configure* et appuyez sur *Start*. La barre de progression en haut à droite vous signalera si tout s'est bien passé.

Testez puis faites valider votre intégration par le professeur ou l'assistant.

Si la carte n'est pas reconnue

- Vérifiez que le câble USB est bien branché.
- Vérifiez que la carte de développement est allumée.
- Vérifiez que le matériel est reconnu par le pc avec `lsusb`. Il devrait y avoir une ligne avec un device **Altera**.
- Vérifiez que le service JTAG reconnaisse le matériel avec la commande `jtagconfig`, si rien n'est listé faite :

```
$ sudo killall -9 jtagd
$ sudo /opt/EDA/quartus16/quartus/bin/jtagd
```

Et vérifiez à nouveau avec :

```
$ jtagconfig
```

- Vérifiez la position des switches MSEL sur le dessous de la carte. MSEL[0..4] devrait valoir 01001. Regardez avec l'assistant si vous n'êtes pas sûr.
- Si rien ne fonctionne voir avec l'assistant.