INF1500 – logique des systèmes numériques

PARTIE II

Synthèse et implémentation avec Vivado

Introduction à l'utilisation de Vivado et à la technologie FPGA

Révisions :
Alexy Torres Aurora Dugo - V1.0

Septembre 2019

Département de génie informatique et de génie logiciel
École Polytechnique de Montréal

1 Synthèse et implémentation du design

Avant de commencer le deuxième TP, vous pouvez commencer par synthétiser et implémenter le design du TP1 afin d'être en mesure de l'envoyer sur la carte de développement pour le tester physiquement. Nous vous conseillons donc d,ouvrir un nouveau projet et de refaire un one_bit_full_adder. Cependant libre a vous d'implémenter for additionneur soustracteur 4 bits. Cependant il faudra modifier le fichier de contrainte en conséquence.

Que se passera-t-il ? Les étapes suivantes seront réalisées par les outils :

- la synthèse traduira la description de votre circuit en blocs disponibles dans la technologie utilisée
- la transformation (mapping) regroupera les composantes obtenues lors de la synthèse dans des blocs spécifiques du FPGA que nous utilisons
- la disposition (placement) choisira des endroits spécifiques sur le FPGA où disposer les blocs utilisés et choisira les broches du FPGA correspondant aux ports du design
- le routage (routing) établira les connexions électriques entre les blocs utilisés
- la configuration (configuration) convertira toute cette information en un fichier pouvant être téléchargé vers le FPGA pour le "programmer"

1.1 Fichier de contraintes

Pour cette introduction simple, nous allons synthétiser et implémenter l'additionneur complet de 1 bit. Nous aurons ainsi le fichier de contraintes d'utilisateur suivant, qui devra être enregistré quelque part dans le répertoire du design (souvenez-vous où) :

```
# LEDs
set_property -dict { PACKAGE_PIN H17 IOSTANDARD LVCMOS33 } [get_ports { cout }];
set_property -dict { PACKAGE_PIN K15 IOSTANDARD LVCMOS33 } [get_ports { s }];
# commutateurs
set_property -dict { PACKAGE_PIN J15 IOSTANDARD LVCMOS33 } [get_ports { a }];
set_property -dict { PACKAGE_PIN L16 IOSTANDARD LVCMOS33 } [get_ports { b }];
set_property -dict { PACKAGE_PIN M13 IOSTANDARD LVCMOS33 } [get_ports { cin }];
```

Les valeurs IOSTANDARD doivent être conformes au fichier Master_NEXYS4 disponible dans les ressources supplémentaires du laboratoire sur Moodle.
Ce fichier permet de "relier" les entrées/sorties du circuit a implémenter aux entrées/sorties de la carte physique. Ici, nous relions H17 et K15 (les LEDs en bas à droite de la carte) aux sorties cout et S de notre circuit. Notez que le symbole "#" est utiliser pour commenter le texte. Nommez ce fichier labo1.xdc.

Une fois fait, dans **Sources**, faites un clic droit sur **Constraints**, cliquez sur **Add Sources...** comme le montré la figure suivante :

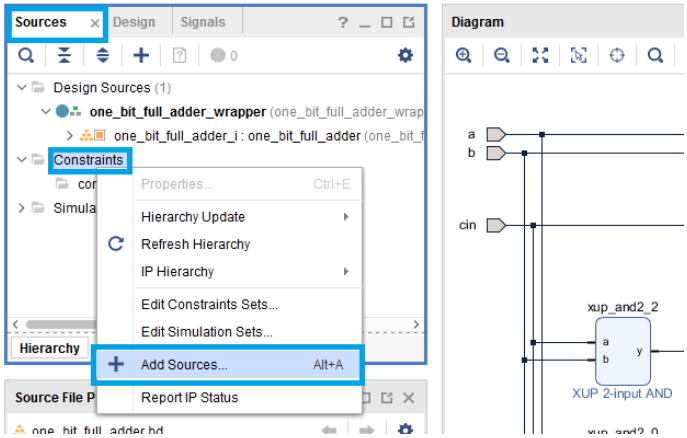

FIGURE 1 – Ajout de contraintes

Cochez la case **Add or Create Constraints**, puis cliquez sur **Next** puis **Add Files** pour choisir le fichier que vous venez de créer (illustré dans la figure ci-dessous).

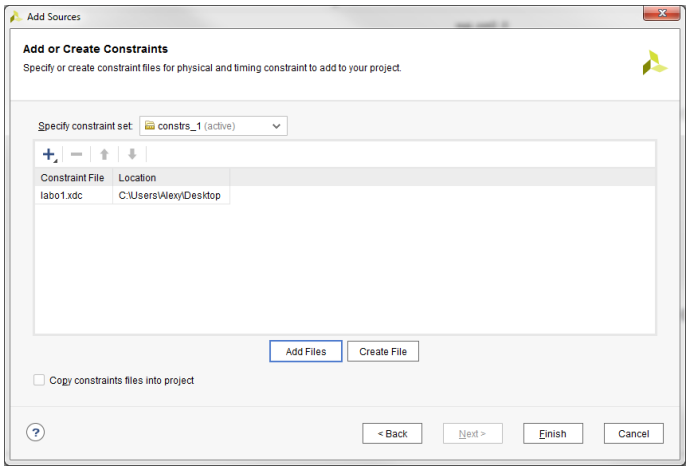

FIGURE 2 – Ajout de contraintes

Vous pouvez maintenant cliquer sur **Finish**.

1.2 Synthèse du circuit

Pour les opérations de **synthèse**, **implémenation** et**génération du bitstream**, il faut faire un clique gauche sur **Run Synthesis**, **Run Implementation** et **Generate Bitstream**, respectivement.

- Run Synthesis : permet de faire la synthèse du circuit en blocs configurables du FPGA.
- Run Implementation : permet de faire les mapping et routage nécessaires pour placer le tout sur le composant FPGA.
- Generate Bitstream : génère le fichier (.bit) utilisé pour programmer le FPGA.

Après chacune des étapes, un message de succès devrait s'afficher dans la console. Dans le cas contraire, il faut corriger les erreurs ou inspecter les avertissements (warnings) pour vous assurer qu'ils ne proviennent pas d'erreurs dans votre code. Chaque processus peut être long, attendez la fin du processus avant de lancer le suivant.

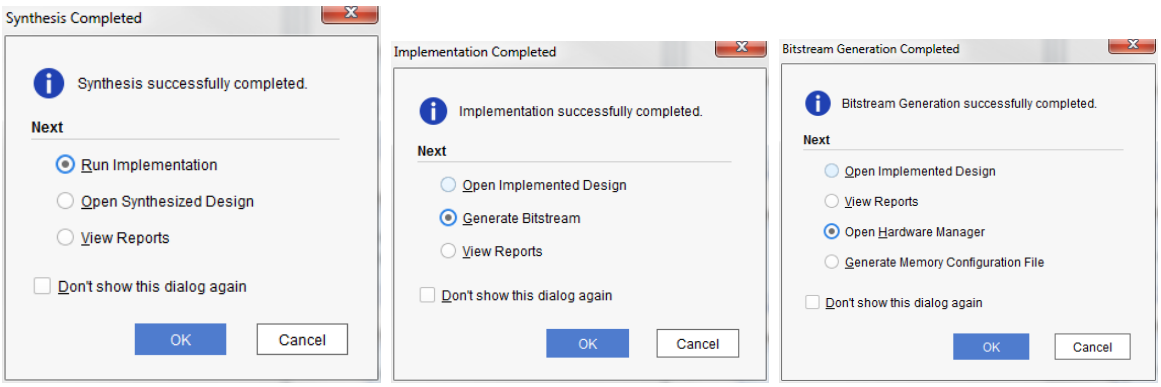

FIGURE 3 – Messages de fin de processus

L'erreur la plus fréquente est produite par une différence entre les noms des ports de votre circuit dans votre code VHDL et dans le fichier de contraintes .xdc.

2 Configuration du FPGA

Nous allons maintenant téléverser votre fichier bitstream sur la carte FPGA. Dans le **Flow Navigator**, cliquez sur **Open Hardware Manager** en bas de **Generate Bitstream**. Vérifiez que votre carte FPGA est bien allumée et branchée à l'ordinateur.
Le gestionnaire de matériel comporte un bandeau vert comprenant le texte "No hardware target is open. Open Target". Cliquez alors sur "Open target" puis "Auto connect". Vous devriez alors voir la fenêtre suivante :

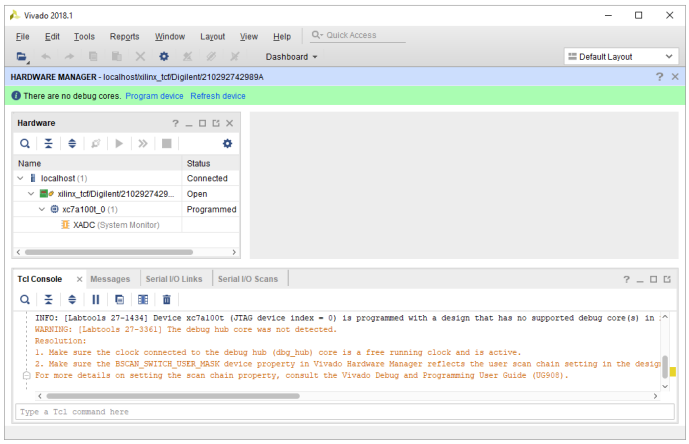

FIGURE 4 – Gestionnaire de matériel Vivado 2018.1

Vous trouverez à gauche de la fenêtre la boîte “Hardware”.

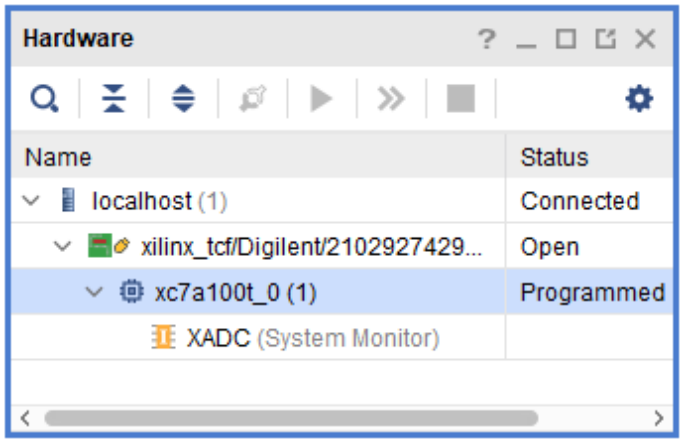

FIGURE 5 – boîte “Hardware” de Vivado 2018.1

Il faudra alors faire un clique-droit sur la board “xc7a100t_0 (1)” puis cliquer sur “Program Device”. Il est possible que le numéro entre parenthèse ne soit pas le même pour vous.

La fenêtre suivante devrait alors apparaître :

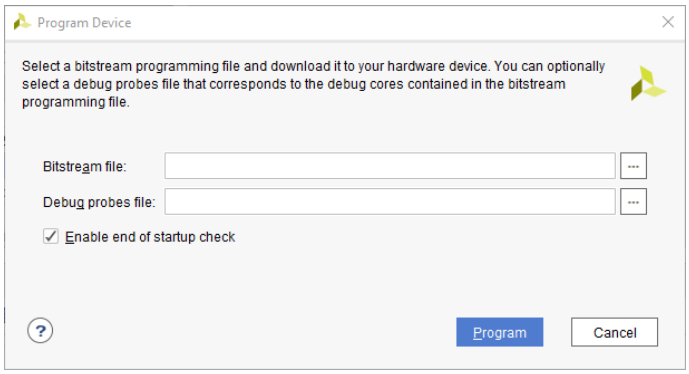

FIGURE 6 – Boîte de dialogue de programmation du FGPA

Sélectionnez alors votre fichier bitstream généré par Vivado lors de l'implémentation de votre circuit. Il devrait se trouver avec le chemin suivant :
Labo1\Labo1.runs\impl_1\one_bit_full_adder.bit
Cliquez enfin sur le bouton "Program" de la boite de dialogue. Une fois le processus terminé, si aucune erreur n'est retournée par Vivado, votre carte FPGA est programmée.

Vous pouvez maintenant vérifier que votre programme fonctionne sur la carte FPGA. Vous pourrez répéter l'étape de programmation de la carte avec le bitstream autant de fois que voulu après chaque implémentation.

Bravo, vous venez de finir la partie 2 de l'introduction à Vivado !