Synthèse de haut-niveau

Olivier Muller

Le but de ce TP est de vous initier à la synthèse de haut niveau. Avec l'outil Vivado HLS, nous verrons comment synthétiser, simuler, optimiser et générer un circuit. Nous travaillerons sur une application de filtrage de type FIR.

Ex. 1 : Découverte de l'outil Vivado HLS et réalisation d'un FIR basique

Question 1 Après avoir récupérez les sources sur Chamilo, démarrez l'outil Vivado HLS et créez un nouveau projet en réalisant les commandes suivantes :

- Ouvrez un terminal pour y taper :
  `source /matieres/5MMMTSP/Xilinx/Vivado/2016.2/settings64.sh`
- Lancez ensuite l'outil avec la commande vivado_hls. Un "joli" GUI s'ouvrira.
- Cliquez sur "Create New Project"
- Donnez un nom (ex : fir.prj) et un répertoire à votre projet
- Ajoutez le fichier fir.c et précisez le nom de la fonction top, **fir** dans ce cas
- Ajoutez le fichier tb_fir.c
- Remplissez le champ Period (unité en ns) de manière à cibler une fréquence de 125MHz (chaud une division) et laissez le champ Uncertainty vide
- Cliquez sur le bouton "Part Selection", cherchez la référence xc7z010clg400-1 qui correspond au FPGA utilisé sur une carte Zybo et sélectionnez la.

Question 2 Explorez rapidement l'environnement de synthèse et les sources fournies en regardant en particulier les sources C.

Question 3 Lancez une simulation C pour vérifier que les sources fournies sont bien fonctionnelles. Pour cela, cliquez sur le menu "Project >Run C Simulation") et validez sans options. On remarquera au passage que l'outil fournit un debugger.

Question 4 Réalisez l'opération de synthèse de haut niveau en cliquant sur le triangle vert. L'outil vous fournira rapidement une solution. En regardant la console, indiquez les temps pris par les différentes étapes et l'étape la plus chronophage sur cet exemple. Analysez le rapport de synthèse fourni par l'outil. Vous indiquerez en particulier si l'outil respecte la contrainte en fréquence, la fréquence estimée du circuit généré, sa latence en précisant les détails liés aux boucles, les ressources utilisées. Soyez précis et rigoureux sur le choix des unités de mesure. Pensez-vous que la cible FPGA choisi nous permet d'envisager d'exploiter plus de parallélisme ?

Question 5 La fin du rapport de synthèse mentionne aussi les interfaces du circuit généré. Indiquez les signaux de contrôle par défaut (synchronisation basique et de type poignée de main). Précisez les interfaces retenues pour les 3 entrées-sorties de notre circuit.

Question 6 L'opération de synthèse a généré les sources RTL de votre circuit FIR dans 3 langages : VHDL, Verilog et SystemC. Depuis l'explorateur de fichier sur la gauche, ouvrez les sources VHDL dans le répertoire syn/vhdl. Que pouvez-vous dire à la lecture de ces sources ? Est-ce bien organisé ? facile à comprendre ? Vous parait-il envisageable de modifier ce code ? Sur quelle architecture s'appuie l'outil de HLS (PCPO ou pipeline) ? Combien d'états à la machine d'états ? Quel est l'encodage utilisé ?

L'exploration du code RTL nous a permis de trouver des informations intéressantes sur le circuit mais la méthode est fastidieuse. Heureusement, il existe des outils qui permettent d'analyser en détail la solution sans avoir à regarder le code.

Question 7 Cliquez sur la vue Analysis (en haut à droite de la fenêtre) pour faire apparaître les 5 outils d'analyse : vue hiérarchique (encart en haut à gauche), profil de performance (encart en bas à gauche) montrant en détail les performances par niveau hiérarchique, la vue performance (encart principal) montrant l'ordonnancement de chaque opération sur un état de la FSM, la vue ressource (encart principal, $2^{ème}$ onglet) montrant l'assignation des opérateurs et la vue profil de ressources (encart en bas à gauche, $2^{ème}$ onglet) donnant le détail des ressources par opérateur instancié. Utilisez ces outils pour déterminer à quel cycle est réalisé l'opération MAC et combien de cycles sont nécessaires pour réaliser un accès mémoire. Pour vous aider, vous pouvez utiliser l'astuce suivante : un clic droit sur une case mauve du diagramme de performance permet d'accéder à l'opération dans l'algorithme de départ (goto Source) et à l'opérateur dans le RTL d'arrivée (goto VHDL).

A ce stade, on a obtenu rapidement du code RTL généré par l'outil de HLS et des estimations sur sa qualité. Avant de chercher à l'améliorer, on va regarder le lien avec la suite du flot de conception.

Question 8 Pour réalisez une implantation complète, sélectionnez le menu "Solution >Export RTL", cochez la case **Evaluate**, choisissez le langage VHDL et lancez la bête. Dans la console, vous pourrez voir que Vivado HLS exporte votre IP au format IP-XACT, le fournit à l'outil Vivado, qui lance à son tour une synthèse logique puis un placement-routage. Au bout d'un temps non négligeable, vous obtiendrez un rapport précisant les ressources réellement utilisées par ce circuit, ainsi que la durée de son chemin critique. Indiquez et commentez vos résultats.

Comme vous l'avez vu en cours, il y a généralement 2 grandes étapes dans un flot de conception : une de génération et une de validation. L'outil Vivado HLS propose de valider son travail au travers d'une co-simulation C/RTL. Concrètement, l'outil génère un testbench RTL à partir des stimuli d'entrée du test C, puis lance automatiquement une simulation RTL de l'IP générée avec ce testbench, enfin les résultats de cette simulation sont réinjectées dans le test C pour vérifier la conformité.

Question 9 Lancez cette simulation en cliquant sur le menu "Solution > Run C/RTL Co-simulation". Choisissez VHDL dans la fenêtre qui s'ouvre puis OK. A l'issue de la simulation, vous obtiendrez les informations temporelles de cette simulation. Vous pouvez remarquer que les traces de la simulation peuvent être dumpées, ce qui permet de les visualiser dans un simulateur RTL.

Ex. 2 : Exploration architecturale

Nous allons maintenant explorer plusieurs architectures possibles. Vous pouvez créer de nouvelles solutions en cliquant sur "Project > New Solution" et en lui donnant un nom explicite.

Pour comparer vos solutions, vous pouvez utiliser la vue de comparaison accessible via "Project > Compare Reports".

Par défaut, Vivado HLS ne déroule pas les boucles et ne les pipeline pas. On peut lui indiquer de le faire en ajoutant des pragmas dans le corps de la boucle à modifier. Un déroulage de boucle d'un facteur x peut lui être demandé par l'ajout du code suivant :

```
#pragma HLS UNROLL factor=x
```

Son pipelinage peut lui être demandé par l'ajout du code suivant :

```
#pragma HLS PIPELINE
```

Question 1 En jouant sur ces 2 leviers, on peut synthétiser une dizaine de solutions différentes. Testez les en indiquant bien le cas échéant les limites de la solution et/ou de l'outil HLS. Réalisez ensuite une analyse comparative autour du compromis surface/performance. Vous prendrez le temps de conclure en indiquant la solution qui vous semble la plus pertinente.

Question 2 De l'analyse précédente, vous avez normalement constaté que les interfaces fournies limitent les performances atteignables par votre circuit. Une interface FIFO permettrait à notre IP FIR de limiter le trafic entrant à un échantillon par échantillon calculé. Quel serait le gain par rapport à la solution précédente ?

La réalisation d'interface nécessite de connaître de nombreuses subtilités de l'outil. Pour vous épargner les erreurs qui viennent avec, nous vous fournissons un squelette de code activable via la macro OPTIM.

Question 3 Que pouvez-vous dire de l'architecture décrite par ce squelette ?

Question 4 Complétez la fonction *fir_run* pour la rendre fonctionnelle, puis ajoutez les pragmas vous paraissant utile pour optimiser l'IP en temps. Notez qu'il est possible de demander l'explosion complète d'un tableau *bla* en registres grâce au pragma suivant :

```
#pragma HLS ARRAY_PARTITION variable=bla complete dim=0
```

Indiquez les performances temporelles obtenues pour fir_run et les ressources nécessaires.

Pour aller plus loin... Ex. 3 : J'ai fini et je m'ennuie

Vivado HLS est fourni avec plein de petits exemples de circuits synthétisables dont plusieurs versions de FIR.

Question 1 Avant de partir, je vous recommande donc de fermer le projet en cours, puis de cliquer sur "Open Examples Project" et enfin de naviguer selon vos envies.