SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

Master Informatique

M1 Spécialité SESI

UE 4I108 - J.Denoulet

TP – FPGA1

SYSTEMES PROGRAMMABLES

1ERE PARTIE – SYNTHESE VHDL SUR FPGA

Le double objectif de ce TP est de :

- Prendre en main la chaîne de conception ***Xilinx Vivado*** ainsi que la carte ***FPGA Nexys4*** sur laquelle nous travaillerons.
- De comprendre comment fonctionne l'outil de synthèse de ***Vivado*** et de quelle façon il va interpréter un code VHDL imprécis ou erroné.

I) Prise en main de la carte et des outils

1) Présentation de la carte Digilent Nexys

La carte de développement ***Nexys A7*** (ainsi que la ***Nexys4-DDR***, qui est en fait la même carte sous un autre nom) dispose (notamment) des éléments suivants

- ***FPGA Xilinx Artix-7*** : C'est le composant principal. Ce circuit reconfigurable nous permettra d'implémenter les systèmes numériques des TP à venir.
- ***Oscillateur*** : La carte comporte un oscillateur externe permettant de fournir au FPGA une horloge de fréquence 100 MHz.
- ***Interrupteur ON/OFF:*** Self-explanatory…
- ***Connecteur USB*** : L'USB permet d'alimenter la carte et de programmer le FPGA. Le connecteur sert également de port JTAG et UART qui seront utilisés par la suite.
- ***Connecteurs PMOD*** : ils permettent de connecter la carte Nexys à d'autres systèmes et d'amener des signaux d'entrées/sorties au FPGA.

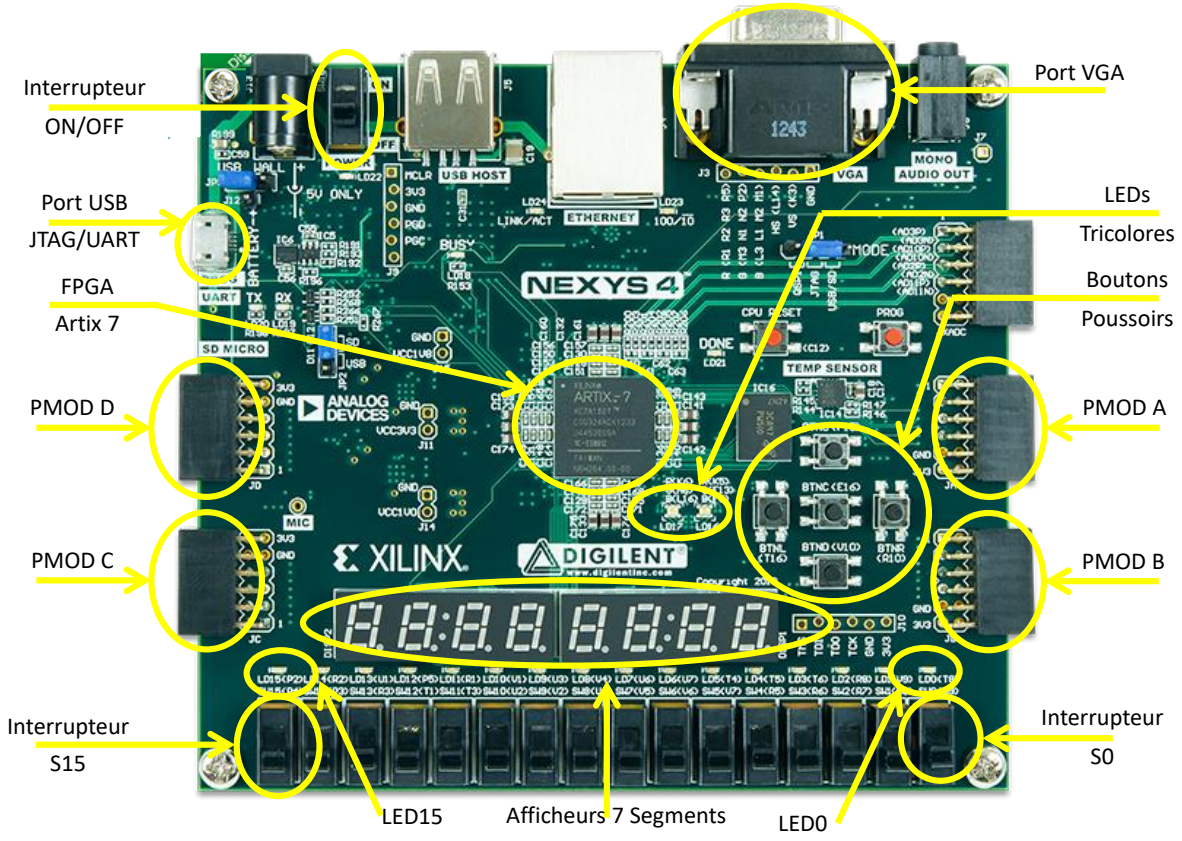
- ***Interrupteurs***: La carte possède 16 interrupteurs. L'interrupteur 15 est à gauche, le 0 à droite. Chaque interrupteur est relié à une broche du FPGA.
    - Si l'interrupteur est tiré vers le bas, la broche est à l'état 0.
    - Si l'interrupteur est tiré vers le haut, la broche est à l'état 1.

- ***LED*** : La carte possède 16 LED vertes. La LED15 est à gauche, la LED0 à droite. Chaque LED est reliée à une broche du FPGA.
    - Si la broche est à l'état 1, la LED est allumée.
    - Si la broche est à l'état 0, la LED est éteinte.

  La carte possède également 2 LED tricolores, chacune commandée par 3 broches du FPGA pour allumer ou éteindre le rouge, vert ou bleu de la LED.

- ***Boutons poussoirs*** : Ces 5 boutons poussoirs (Left, Right, Up, Down, Center) reliés à une broche du FPGA.
    - Si on appuie sur un bouton, un 1 logique est transmis.
    - Sinon on transmet un 0.

- ***Afficheurs 7 seg***: Ces 8 afficheurs 7 segments possèdent un bus de données commun qui permet d'allumer ou d'éteindre chaque segment, et une entrée de validation propre à chaque afficheur. Tous ces signaux sont actifs au niveau bas.



- Les deux cartes étant identiques, le brochage du FPGA avec les différents périphériques est le même pour la ***Nexys A7*** et la ***Nexys4 DDR***. Ce sera donc le même fichier de configuration qu'il faudra utiliser.

2) Création d'un projet avec le logiciel Vivado

- Avant d'ouvrir le logiciel, aller sur le site Moodle de l'UE, dans la section TP, pour télécharger l'archive ***Ressources TP 1ère partie*** qui contient tous les fichiers sources nécessaires au TP. Extraire sur le disque dur du PC le répertoire ***FPGA1_TP1*** à partir de cette archive.

- Ouvrir le logiciel Vivado. (***Menu Démarrer → Tous les Programmes → Xilinx Design Tools → Vivado 2020.2.***
    - Si vous ne trouvez pas Vivado dans le Menu Démarrer, ouvrir un explorateur Windows et lancer le fichier ***C:/Xilinx/Vivado/2020.2/bin/vivado.bat***

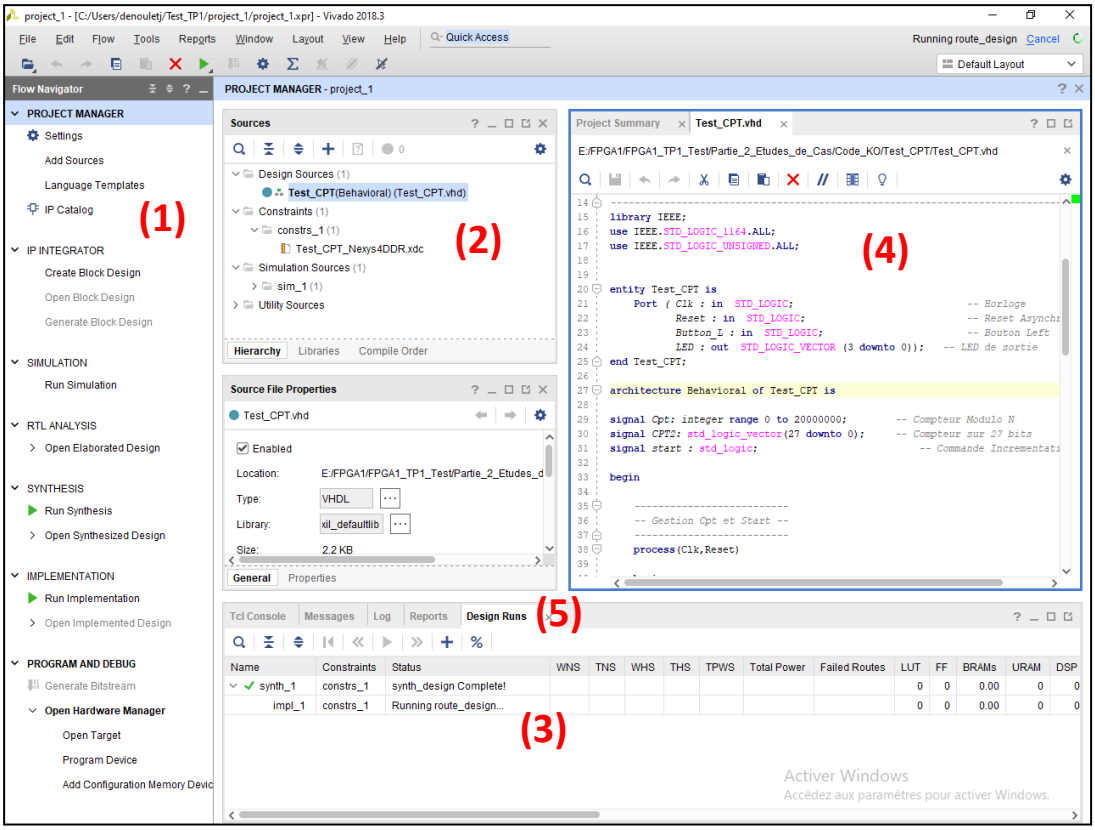
- Cliquer sur ***Create Project*** puis ***Next***. Choisir un nom pour le projet et choisir son répertoire source.
    - Attention : ne pas mettre d'epsaces, d'accents ou de caractères spéciaux dans le nom du projet.

- Cliquer deux fois sur ***Next***. Dans la fenêtre suivante, sélectionner ***VHDL*** comme ***Target Language*** puis cliquer deux fois sur ***Next***.

- Dans la fenêtre suivante pour choisir le FPGA, sélectionner les filtres suivants :
    - Family : ***Artix-7***
    - Package : ***CSG324***
    - Speed : ***-3***

  Sélectionner le FPGA ***xc7a100*** puis cliquer sur ***Next*** puis ***Finish***.

A l'écran s'affiche alors un ensemble de fenêtres similaire à la figure ci-dessous.



(1) ***Flow Navigator*** : Permet de démarrer les différentes phases du flot de conception, de la création des sources VHDL à la simulation, l'implémentation et jusqu'à la programmation du FPGA.

(2) ***Sources*** : Liste les différentes sources (VHDL, Testbenchs, Fichier de contraintes, etc…) du projet ainsi que leur hiérarchie.

(3) ***Console/Messages*** : Indique les messages générés par Vivado. Parmi les onglets disponibles, ***Reports*** (5) permet d'accéder aux rapports de synthèse et d'implémentation.

(4) ***Editeur***: Permet d'éditer les sources du projet ou de consulter les rapports d'implémentation.

3) Création d'un module VHDL

- Dans le ***Flow Navigator***, cliquer sur ***Add Sources***. Choisir ***Add or Create Design Sources***, puis cliquer sur ***Next***. Cliquer sur ***Create File***, puis choisir ***VHDL*** comme type de fichier et donner comme nom ***Test*** au fichier. Cliquer sur ***OK*** puis sur ***Finish***.

- Indiquer ensuite les entrées/sorties du module VHDL: 3 ports d'entrée appelés ***SW2***, ***SW1***, ***SW0*** et un port de sortie sur 3 bits appelé ***LED(2 :0)***. Terminer la création de la source VHDL.

- Dans la fenêtre ***Sources***, en déroulant le dossier ***Design Sources***, vous verrez que le fichier ***Test.vhd*** a été ajouté. Double cliquer sur ce fichier pour l'ouvrir dans la fenêtre ***Editeur***.
    - Vous constaterez que l'entité du module VHDL a déjà été pré-remplie conformément aux indications que vous avez fournies précédemment.

- Ecrire l'architecture du module ***Test***.
    - La ***LED(0)*** prend la valeur de ***SW0***.
    - La ***LED(1)*** prend la valeur de ***SW1***.
    - La ***LED(2)*** prend la sortie d'un ET logique entre les 3 interrupteurs.

4) Testbench et simulation

- Pour créer un fichier Testbench, cliquer à nouveau sur ***Add Sources*** et choisir ***Add or Create Simulation Sources***. Choisir ***Add Files***. Aller chercher le fichier ***TB_Test.vhd*** dans le répertoire ***FPGA1_TP1***. Vérifier que l'option ***Copy sources into project*** est cochée puis cliquer sur ***Finish***.
    - Dans la fenêtre ***Sources***, dérouler le répertoire ***Simulation Sources*** puis ***sim_1***. Vérifier que le fichier ***TB_Test*** est bien présent et que le fichier ***Test*** est bien associé au Testbench.
    - Ouvrir le fichier ***TB_Test.vhd*** pour voir comment se déroule la simulation.

- Cliquer sur ***Run Simulation*** puis ***Run Behavioral Simulation***.
    - Si jamais le code VHDL comporte des erreurs, corrigez-les à l'aide des informations de la ***Console***.
        - Les erreurs peuvent être soulignées en rouge directement dans le fichier VHDL ou bien indiquées suite à la compilation dans la Console Tcl de Vviado. Remonter bien dans la liste des messages d'erreurs pour retrouver les premières lignes de code qui posent problème.
    - Avec les commandes de zoom à gauche du chronogramme, visualiser l'ensemble de la simulation et vérifier le bon comportement de l'architecture.

5) Implémentation sur la carte FPGA

Pour implémenter l'architecture dans le FPGA, il faut ajouter un fichier de contraintes au projet. Ce fichier indique notamment sur quelles broches du circuit mapper les entrées/sorties du code VHDL.

- Cliquer sur ***Add Sources*** puis ***Add or Create Constraints*** pour ajouter le fichier ***Test_Nexys4DDR.xdc*** du répertoire ***FPGA1_TP1***.

- Ouvrir le fichier ***Test.xdc*** en déroulant dans la fenêtre ***Sources*** le répertoire ***Constraints***. Dans les parenthèses get_ports, changer le nom des ports pour les remplacer par ceux de votre description VHDL.
    - ***IMPORTANT – LE FICHIER XDC EST CASE SENSITIVE A L'INVERSE DES FICHIERS VHDL***

- Cliquer sur ***Run Synthesis*** puis dans la fenêtre qui s'ouvre cliquer sur ***OK***.
    - Cette procédure analyse les sources VHDL et les transforme en cellules logiques de base.

- Lorsque la synthèse est terminée, cliquer sur ***Run Implementation***
    - Au cours de cette phase, Vivado va placer dans le FPGA les cellules logiques identifiées lors de la synthèse, conformément aux contraintes données par le fichier ***XDC***. Les cellules placées sont ensuite routées les unes avec les autres.

- A la fin de l'implémentation, cliquer sur ***Generate Bitsream*** puis sur ***OK*** pour générer le fichier de configuration du FPGA.

- Cliquer ensuite sur ***Open Hardware Manager*** pour programmer le FPGA.
    - Connecter la carte au port USB du PC et allumer la carte avec l'interrupteur ON/OFF
    - Dans le bandeau vert en haut de l'écran, cliquer sur ***Open New Target***.
    - Cliquer autant de fois que nécessaire sur ***Next*** puis ***Finish***.
    - A nouveau dans le bandeau vert, cliquer sur ***Program Device.*** Vérifier que le chemin du bitstream (Test.bit) est bien spécifié avant de lancer la programmation du FPGA..
        - Si ce n'est pas le cas, le bitstream se trouve à l'endroit suivant : C :\Users\1234567\My_Project\MyProject.runs\impl_1\Test.bit

- Le FPGA est à présent programmé.
    - Vérifier en jouant sur les interrupteurs que les LEDs s'allument conformément au code VHDL.

II) Cas d'études – Synthèse VHDL.

Dans les parties suivantes, nous allons travailler avec plusieurs applications dont le code VHDL vous sera fourni. Nous allons voir en quoi ces architectures sont bien ou mal interprétées par l'outil de synthèse de Vivado et, le cas échéant, comment écrire un code qui soit parfaitement compréhensible et synthétisable par l'outil.

- L'architecture 1) propose deux compteurs imbriqués dont la valeur des poids forts est affichée sur les LEDs.
- Les architectures 2) et 3) réalisent un système d'allumage/clignotement de LEDs selon les impulsions sur les boutons de la carte.

1) Compteurs Imbriqués

❖ Observation

- Reprendre le projet précédent en enlevant toutes les sources (sélectionner toutes les sources, puis cliquer avec le bouton droit et choisir ***Remove File from project***)

- Ajouter au projet les fichiers ***Test_CPT.vhd*** et ***Test_CPT_Nexys4DDR.xdc***.
    - Quelle est la fonctionnalité décrite dans le VHDL?
    - A quel résultat peut-on s'attendre sur la carte ?

- Implémenter le design.
    - Astuce : vous pouvez cliquer directement sur ***Generate Bitstream*** pour faire d'un trait toutes les étapes de l'implémentation. (Le flot s'arrêtera en cas d'erreur)
- Tester sur la carte. Que se passe-t-il ?

❖ Analyse

- Dans le ***Flow Navigator***, dérouler la ligne ***Open Synthesized Design*** et cliquer sur ***Schematic*** pour générer un schéma RTL de l'architecture synthétisée.
    - Est-ce que cela correspond au code VHDL de départ ?

- En revenant dans le ***Project Manager*** (fenêtre ***Flow Navigator***), aller dans la fenêtre ***Console*** et cliquer sur l'onglet ***Reports***. Afficher le ***Synthesis Report***. faire une recherché sur l'expression « **[synth** » et analyser les messages ***WARNING***.
    - Que comprenez-vous ? Y a-t-il quelque chose qui vous semble anormal ?
    - En faisant le lien entre les Warning et le Schematic vu précédemment, quel signal dans la description VHDL peut avoir posé problème lors de la synthèse ?

- En regardant précisément la déclaration et la description VHDL du signal en question, trouver l'erreur et apporter une modification.

❖ Correction et vérification

- Une fois l'erreur corrigée, refaire une synthèse et ouvrir le rapport.
    - Vérifier en fin de rapport qu'il n'y a plus de warnings.
    - Recharger le schéma RTL de l'architecture synthétisée pour constater la différence

- Poursuivre l'implémentation et porter le design sur la carte. Valider le bon fonctionnement de l'architecture.

2) Compteur d'Impulsions

❖ Observation

- Retirer toutes les sources de votre projet et ajouter les fichiers ***Impulse_Count.vhd*** et ***Impulse_Count_Nexys4DDR.xdc***.
    - Quelle est la fonctionnalité décrite par le code VHDL ?

- Ajouter au projet le testbench ***TB_Impulse_Count.vhd***.
    - Analyser le Testbench pour savoir ce que l'on va effectuer dans la simulation.
    - Simuler et vérifier le bon comportement de l'architecture.

- Implémenter et vérifier le bon comportement sur la carte. Que constatez-vous ?
    - NB : il y a un phénomène de rebond sur les boutons de la carte qui peut entraîner plusieurs fronts du signal pour un seul et même appui sur le bouton.

❖ Analyse

- Regarder les warnings générés en synthèse et générer le ***Schematic*** du système synthétisé.
    - Quelle différence y a-t-il entre la description VHDL et l'architecture synthétisée ?

- Dans le ***Flow Navigator***, cliquer sur ***Project Settings*** et vérifier que le ***Target Language*** est bien VHDL (sinon, faire la modification).
- Relancer ensuite une simulation, mais en choisissant cette fois la ***Post Synthesis Functional Simulation***.
    - L'outil va simuler un fichier VHDL correspondant à l'architecture synthétisée par ***Vivado***. Ce fichier, généré lors de la synthèse est consultable dans le simulateur en cliquant sur le module ***UUT*** dans la fenêtre ***Scopes***.
    - Ouvrir et regarder la structure du code. Comment est-il composé ?
    - Comparer le chronogramme de la simulation post-synthèse avec la simulation pré-synthèse vue précédemment. Le comportement est-il différent ?
    - Que constatez-vous entre le comportement de la simulation post synthèse et celui du design testé sur la carte ?.

❖ Correction et vérification

- Comment modifier le code VHDL pour faire en sorte que le compteur d'impulsions fonctionne correctement ?
    - Décrire cette nouvelle architecture en VHDL
    - Modifier le testbench en conséquence puis lancer une simulation (***Behavioral Simulation***) pour valider fonctionnellement cette nouvelle description.

- Modifier le fichier de contraintes ***XDC*** de la façon suivante :
    - Pour les cartes ***Nexsy4DDR*** et ***Nexys A7***
        - Décommenter les lignes 7,8 sur le port d'horloge en faisant attention que le nom corresponde bien à celui de votre code VHDL.
        - Passer en commentaire la ligne 29 (***set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets Button_C_IBUF]***)

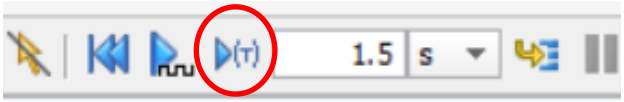
- Implémenter votre design. Examiner le rapport de synthèse et le schéma RTL pour vérifier la bonne compréhension de votre code par l'outil, puis tester sur la carte.

3) Décodeur & Machine à Etats

❖ Observation

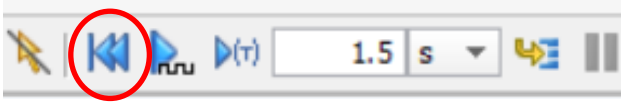
- Dans le même projet, ajouter de nouvelles sources (***Design Sources***) correspondant aux fichiers suivants :
    - ***Selector.vhd, FSM.vhd*** et ***Top.vhd***
    - Analyser les codes pour comprendre quelle est la fonctionnalité et la hiérarchie du système.

- Nous allons à présent simuler le système.
    - Pour cela, ajouter le testbench ***TB_Top.vhd***.
    - Dans la fenêtre ***Sources***, dans le répertoire ***Simulation Sources***, cliquer avec le bouton droit sur ***TB_Top.vhd*** et choisir ***Set as Top***. Lancer le simulateur en mode ***Behavioral***.
    - Régler le pas de temps à 1,5s puis cliquer sur l'icône ci-dessous pour faire avancer la simulation du temps désiré.
        - *NB1 : La décimale est indiquée par un point et non une virgule (i.e. 1.5 s et non pas 1,5 s)*
        - *NB2 : Attention, le temps de simulation pour arriver jusqu'à 1,5s est assez long…*



    - Vérifier le bon fonctionnement du code.

- ***IMPORTANT :*** Avant de quitter le simulateur, cliquer sur l'icône ***Restart*** pour supprimer le fichier du chronogramme précédent (qui occupe une taille de +1Go sur le disque) et qui vous fera perdre du temps et de la place sur la clé USB ou le cloud sur lequel vous allez rapatrier votre projet après le TP.



- Remplacer le précédent fichier ***XDC*** par ***Top_Nexys4DDR.xdc***.

- Si ce n'est pas déjà le cas, mettre le fichier ***Top.vhd*** comme top level du système (***Set as Top***)

- Implémenter le design et tester sur la carte.
    - Appuyer 3 fois sur le bouton Left (sélection du mode 1) puis 1 fois sur le bouton Right (validation), ce qui doit entraîner un clignotement des ***LEDs***. Que constatez-vous ?
    - Faites un bref reset du système (Tirer le Switch 0 vers le bas, puis le remettre vers le haut). Cette action devrait normalement éteindre les LED. Que se passe-t-il ?

❖ Analyse et Correction

- Ouvrir le rapport de synthèse et repérer les warnings..
    - Quelle(s) erreur(s) dans le code a pu provoquer cela ?

- Corriger le code de ***Selector.vhd*** et de ***FSM.vhd*** et implémenter à nouveau.
    - Vérifier dans le rapport de synthèse que le ou les warnings ont disparu.
    - Valider sur la carte le bon fonctionnement du design