

1 Prise en main de ModelSim sur le composant alu_reg vu en TD

Afin de prendre en main le logiciel de simulation ModelSim, il est demandé de simuler l'exercice sur l'unité arithmétique et logique vu en TD. Tous les fichiers nécessaires se trouvent dans le répertoire alu_reg. Son contenu est présenté dans le tableau 1. Toutes les informations sur les commandes sont données dans le document Guide ModelSim.

Nom du répertoire	Description
bench	Contient les environnements de simulations et les programmes de tests
config	Contient les scripts de configuration et les fichiers d'initialisation
libs	Contiendra les bibliothèques VHDL que vous compilerez : lib_VHDL : bibliothèque comportementale (avant synthèse) lib_BENCH : l'environnement de test Note : les répertoires correspondant apparaîtront lors de la création de ces bibliothèques
vhd	Contient les sources VHDL

TABLE 1 – Structure des répertoires du dossier alu_reg

Travail à réaliser :

- aller dans le dossier alu_reg
- dans le répertoire config, exécuter la commande source config_RTL
- créer les librairies nécessaires à la compilation ModelSim
 - aller dans le répertoire libs
 - exécuter la commande vlib lib_VHDL
 - exécuter la commande vlib lib_BENCH
- faire le lien entre le nom logique des bibliothèques et leur emplacement
 - exécuter la commande vmap lib_VHDL \${TP_PATH}/libs/lib_VHDL
 - exécuter la commande vmap lib_BENCH \${TP_PATH}/libs/lib_BENCH
- compiler le code vhd de la description du composant alu_reg
 - aller dans le répertoire vhd
 - vcom -work lib_VHDL alu_reg.vhd
 - vcom -work lib_VHDL full_adder.vhd
- compiler le bench
 - aller dans le répertoire bench
 - vcom -work lib_BENCH bench.vhd
- lancer le simulateur par la commande vsim&
- observer les chronogrammes

2 Environnement pour le travail sur le filtre

2.1 Structure des répertoires

L'environnement de travail sur le filtre se trouve dans le répertoire filtre. Le tableau 2 décrit son contenu. Vous aurez à vous reporter à ce tableau tout au long du TP pour savoir où trouver les données ou pour vous situer au bon endroit lors de l'exécution de certaines commandes.

Nom du répertoire	Description
bench	Contient les environnements de simulations et les programmes de tests
chronogramme	Chronogramme du composant FSM à compléter
config	Contient les scripts de configuration et les fichiers d'initialisation
fpga	Dédié à la cible FPGA
fpga/userconstraints	Contient le fichier des contraintes de placement et d'affectation des broches du FPGA
fpga/par	Dédié au placement et routage sur FPGA (Xilinx ISE™)
fpga/synth	Dédié à la synthèse VHDL
libs	Contiendra les bibliothèques VHDL que vous compilerez : lib_VHDL : bibliothèque comportementale (avant synthèse) lib_BENCH : l'environnement de test lib_FPGA_SYNTH : bibliothèque de compilation du fichier de synthèse sur FPGA lib_FPGA_PAR : bibliothèque de compilation pour le placement et routage pour le FPGA Note : les répertoires correspondant apparaîtront lors de la création de ces bibliothèques
spyglass	Dédié à la vérification du code VHDL avec l'outil Lint
vhd	Contient les sources VHDL

TABLE 2 – Structure des répertoires du dossier filtre

2.2 Avant de commencer à travailler

La première chose à faire avant de commencer à travailler est de sourcer¹ le script de configuration correspondant à l'étape du TP dans laquelle vous vous trouvez :

- config_RTL : configuration pour la simulation comportementale au niveau RTL
- config_FPGA : configuration une implantation sur FPGA

Tous ces fichiers de configuration ont été placés dans le sous-répertoire "config". Ils définissent tous une variable d'environnement appelée TP_PATH qui transporte le chemin absolu de la base de l'arborescence du TP. Vous pouvez utiliser cette variable d'environnement pour exécuter vos commandes ou écrire vos scripts.

3 Présentation et spécification du filtre numérique

Ce TP propose d'étudier et de réaliser sur cible FPGA un filtre numérique à réponse impulsionnelle fine à 32 coefficients. Ceci permettra la prise en main de toutes les étapes de la conception et la validation de circuits numériques.

Une étude sous Matlab a permis de spécifier la réponse fréquentielle voulue. Les courbes obtenues sont reproduites sur la figure 1. Les trente-deux coefficients correspondant sont donnés dans le tableau 3.

3.1 Étude et spécification du filtre numérique

En notant In_n le $n^{\text{ème}}$ échantillon du signal d'entrée, Out_n le $n^{\text{ème}}$, a_i le $i^{\text{ème}}$ coefficient du filtre et N la taille du filtre, ici $N = 32$, l'expression analytique du filtrage est :

$$Out_n = \sum_{i=0}^{i=N-1} a_i \cdot In_{n-i}$$

L'architecture canonique est donnée à la figure 2, on observe que celle-ci est trop coûteuse car le nombre d'opérateurs arithmétiques est trop important. C'est pourquoi, l'architecture optimisée de la figure 3 lui est préférée.

Cette architecture est détaillée dans la figure 4. Sa description en VHDL est consignée par l'ensemble des fichiers du répertoire vhd. Le bloc FSM permet de contrôler le calcul d'une valeur de sortie du filtre. Les blocs

1. Dans le contexte d'une utilisation des outils de CAO dans un environnement Linux, l'expression « sourcer un script » désigne (par contamination du nom de la commande) le lancement du script à l'aide de la commande Linux source plutôt que par une exécution simple sur la ligne de commande.

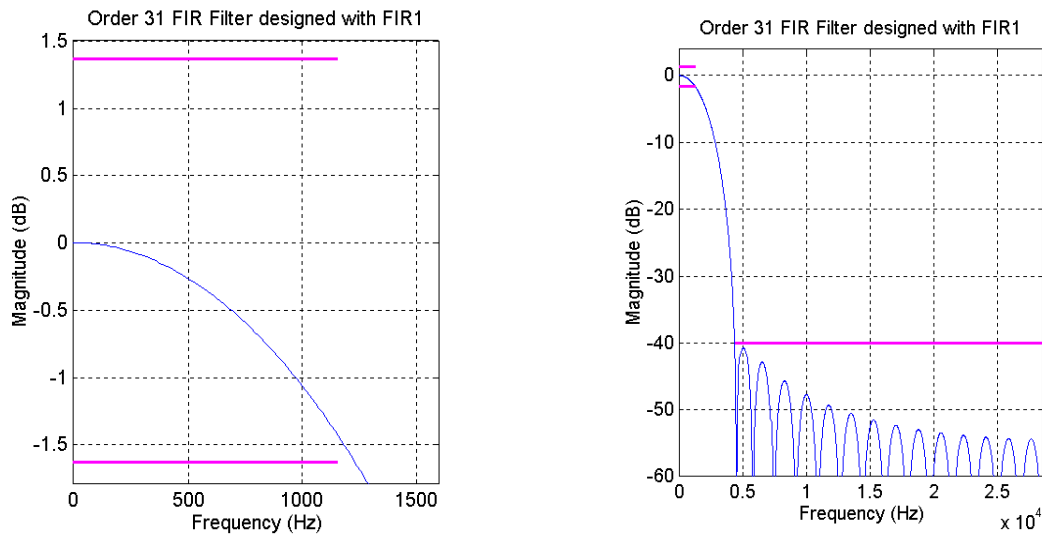


FIGURE 1 – Réponse fréquentielle du filtre

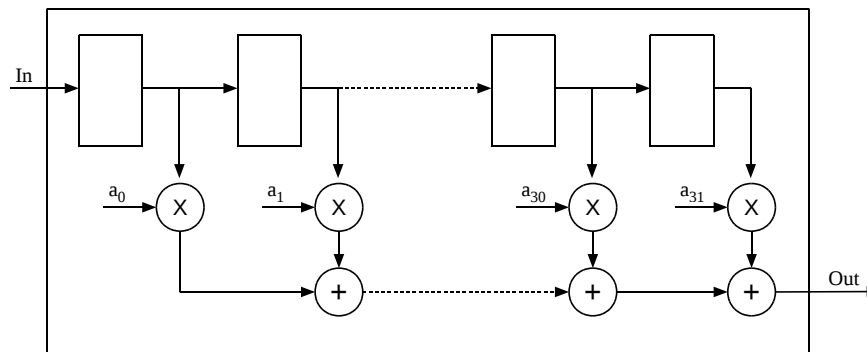


FIGURE 2 – Architecture canonique

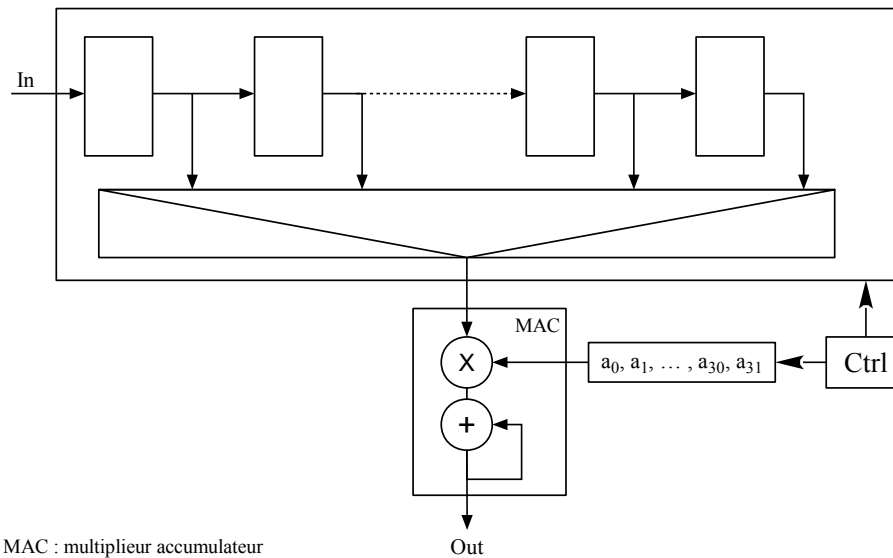


FIGURE 3 – Architecture séquentielle

Binaire	Hexadécimal	Décimal ar- rondi	Décimal	Erreur
0 0 0 0 1 1 0 1	0D	0.0508	0.0512	0.0004
0 0 0 1 0 1 0 1	15	0.0820	0.0832	0.0012
0 0 0 1 1 1 1 1	1F	0.1211	0.1245	0.0034
0 0 1 0 1 1 0 0	2C	0.1719	0.1754	0.0035
0 0 1 1 1 1 0 0	3C	0.2344	0.2355	0.0011
0 1 0 0 1 1 0 1	4D	0.3008	0.3039	0.0031
0 1 1 0 0 0 0 1	61	0.3789	0.3791	0.0002
0 1 1 1 0 1 0 1	75	0.4570	0.4591	0.0020
1 0 0 0 1 0 1 0	8A	0.5391	0.5412	0.0021
1 0 0 1 1 1 1 1	9F	0.6211	0.6226	0.0015
1 0 1 1 0 0 1 1	B3	0.6992	0.7001	0.0009
1 1 0 0 0 1 0 1	C5	0.7695	0.7707	0.0012
1 1 0 1 0 1 0 0	D4	0.8281	0.8314	0.0033
1 1 1 0 0 0 0 1	E1	0.8789	0.8795	0.0006
1 1 1 0 1 0 0 1	E9	0.9102	0.9128	0.0026
1 1 1 0 1 1 1 0	EE	0.9297	0.9298	0.0001
1 1 1 0 1 1 1 0	EE	0.9297	0.9298	0.0001
1 1 1 0 1 0 0 1	E9	0.9102	0.9128	0.0026
1 1 1 0 0 0 0 1	E1	0.8789	0.8795	0.0006
1 1 0 1 0 1 0 0	D4	0.8281	0.8314	0.0033
1 1 0 0 0 1 0 1	C5	0.7695	0.7707	0.0012
1 0 1 1 0 0 1 1	B3	0.6992	0.7001	0.0009
1 0 0 1 1 1 1 1	9F	0.6211	0.6226	0.0015
1 0 0 0 1 0 1 0	8A	0.5391	0.5412	0.0021
0 1 1 1 0 1 0 1	75	0.4570	0.4591	0.0020
0 1 1 0 0 0 0 1	61	0.3789	0.3791	0.0002
0 1 0 0 1 1 0 1	4D	0.3008	0.3039	0.0031
0 0 1 1 1 1 0 0	3C	0.2344	0.2355	0.0011
0 0 1 0 1 1 0 0	2C	0.1719	0.1754	0.0035
0 0 0 1 1 1 1 1	1F	0.1211	0.1245	0.0034
0 0 0 1 0 1 0 1	15	0.0820	0.0832	0.0012
0 0 0 0 1 1 0 1	0D	0.0508	0.0512	0.0004

TABLE 3 – Liste des 32 coefficients du filtre numérique

ADC_FSM et DAC_FSM permettent d'effectuer les demandes de conversions. Ces blocs implémentent les protocoles de communication avec les convertisseurs. Les chronogrammes présentés sur les figures 5 et illustrent le comportement de ces deux blocs.

Le fichier de test «bench/bench.vhd» simule le fonctionnement des convertisseurs. Il fournit des signaux en entrée du filtre et observe ceux en sortie en tenant compte des caractéristiques fonctionnelles et temporelles des signaux allant ou provenant des convertisseurs. Ce fichier simule une réponse impulsionnelle. Le fichier «bench/-bench_sinus.vhd» simule une entrée sinusoïdale.

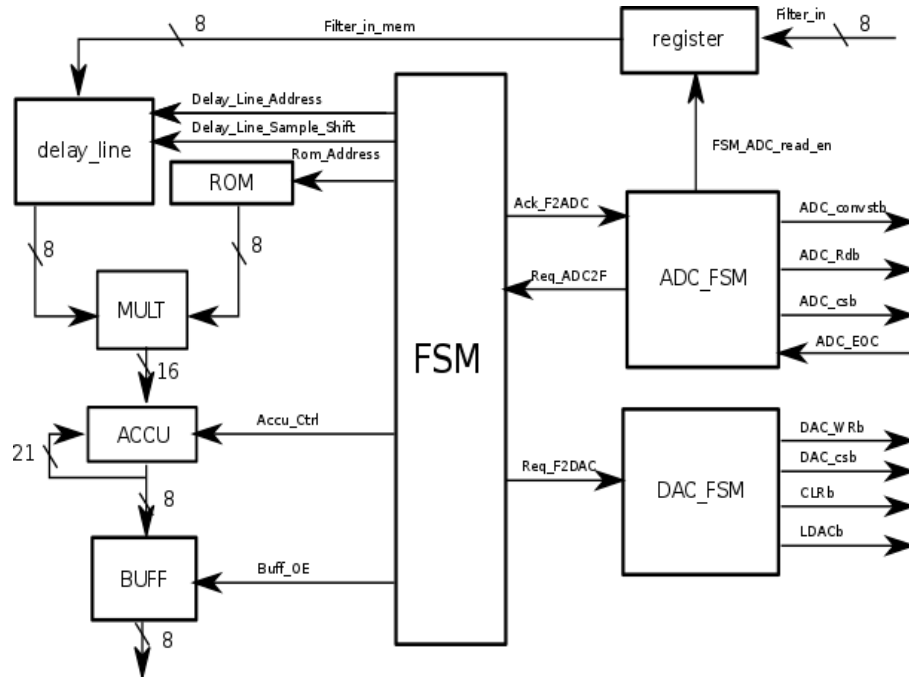


FIGURE 4 – Architecture séquentielle détaillée

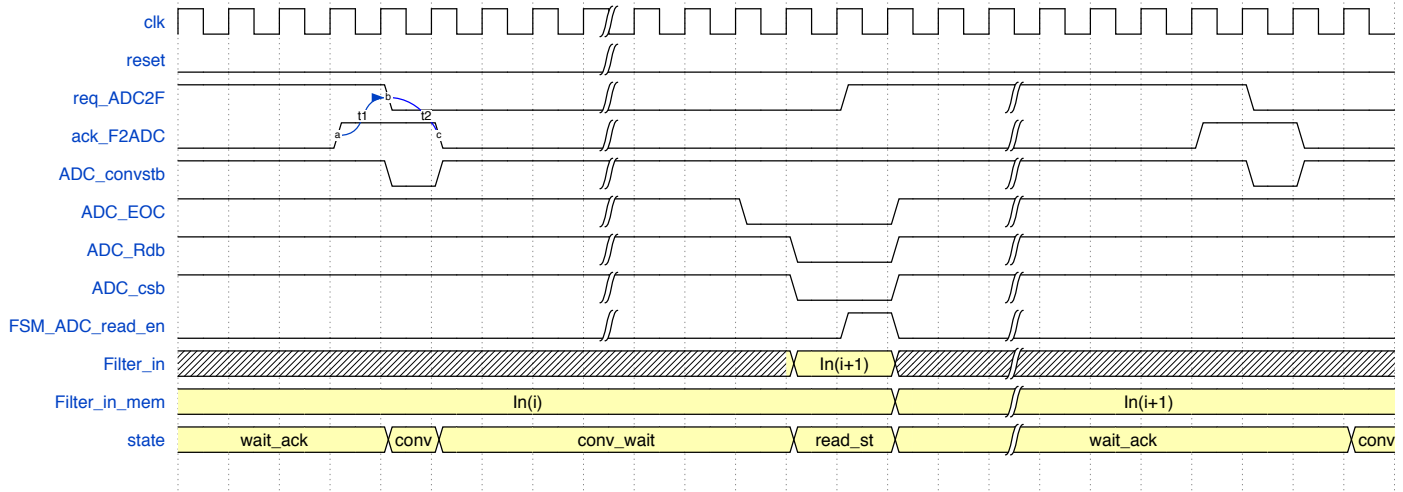


FIGURE 5 – Protocole de communication entre la FSM contrôlant le calcul et la FSM contrôlant le CAN

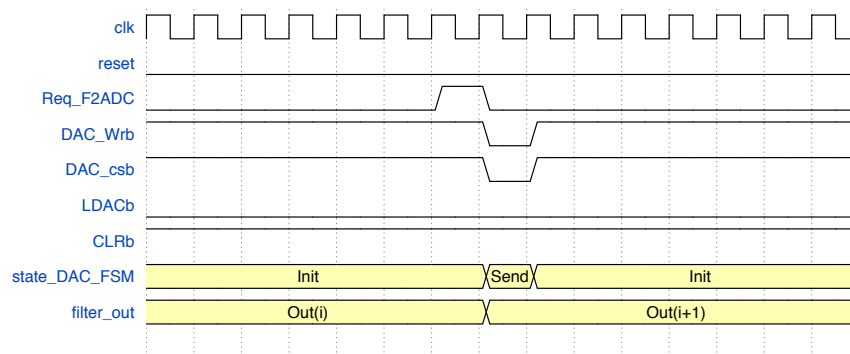


FIGURE 6 – Protocole de communication entre la FSM contrôlant le calcul et la FSM contrôlant le CNA

3.2 Travail demandé lors de la partie conception du filtre RIF

1. Comprendre et commenter l'ensemble du code source donné dans les répertoires "vhd" et "bench".
2. Faire un chronogramme sur wavedrom spécifiant le fonctionnement du système (compléter le fichier chronogramme/filtre.json).
3. Extraire les machines à états finis correspondant aux composants FSM et ADC_FSM. Les dessiner sous forme de graphe de transitions d'états.
4. Compléter le fichier "vhd/fsm.vhd". Il s'agit d'écrire en VHDL l'architecture associée à l'entité fsm du contrôleur du filtre. Le contrôleur doit être tel que le filtre puisse s'interfacer avec les FSMs communiquant avec les systèmes de conversion analogique/numérique et numérique/analogique.
5. Compléter le fichier "vhd/fsm_ADC.vhd". Il s'agit d'écrire en VHDL l'architecture associée à l'entité fsm_ADC qui communique avec les systèmes de conversion analogique/numérique (cf. annexe).
6. A l'aide de l'outil d'Atrenta SpyGlass™ vérifier le vhd du fichier fsm et de l'ensemble des fichiers VHDL en utilisant le module «Lint» (voir le Tutorial Spyglass dans le répertoire doc)
7. Modifier les benchs. Le fichier de test "bench/bench.vhd" simule le fonctionnement des convertisseurs. Il fournit des signaux en entrée du filtre et consomme ceux en sortie en tenant compte des caractéristiques fonctionnelles et temporelles des signaux allant ou provenant des convertisseurs (cf. documentation technique en annexe). Ce programme de test, que vous avez à compléter, devra vérifier que le filtre est fonctionnellement correct en appliquant les stimulus appropriés de façon à examiner la réponse impulsionnelle, la réponse indicielle ou la réponse à une somme de sinusoïdes. Expliquez votre démarche.
8. Simuler le filtre complet dans l'environnement de test développé pour vérifier son bon fonctionnement. Corriger toutes les erreurs de conception en plus des erreurs de syntaxe. Argumenter avec précision pourquoi le fonctionnement du filtre est correct.

Pour effectuer des simulations comportementales avec le logiciel ModelSim, vous devez au préalable sourcer le script de configuration config_RTL. Le mini guide fourni sur ModelSim donne les principales commandes à connaître pour utiliser ce logiciel. N'hésitez pas à consulter aussi l'aide du logiciel (le mini guide indique où trouver cette aide).

Un compte rendu synthétique, décrivant l'ensemble de vos travaux, est à rendre à la fin de la deuxième séance.

4 Réalisation du filtre sur FPGA

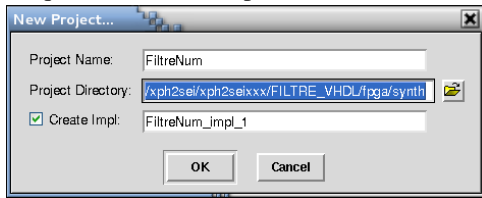
L'objectif de cette partie de TP est de se familiariser avec le flot de synthèse et validation d'un circuit numérique sur cible FPGA. Pour cette partie vous devez sourcer le fichier de configuration config_FPGA.

4.1 Synthèse sur FPGA de la description VHDL avec l'outil Precision de Mentor Graphics

4.1.1 Lancer l'outil Précision

1. Sourcer le script de configuration config_FPGA et assurer vous que vous êtes dans le répertoire \${TP_PATH}/fpga/synth
2. Démarrer l'outil en exécutant la commande precision&
3. Créer un nouveau projet via le menu « Project | New Project »
4. Donner le nom de votre choix à votre projet, par exemple « FiltreNum », en renseignant la zone de saisie « Project Name ».
5. Assurer vous que le champ « Project Directory » désigne bien le répertoire \${TP_PATH}/fpga/synth
6. La zone de saisie « Create Impl » donne le nom du répertoire qui contiendra la description du circuit synthétisé. Le nom par défaut est le nom du projet (champ « Project Name ») suivie de « impl_1 » (adapter ce nom si besoin est). La case à cocher doit être cochée pour que le répertoire soit créé.

7. Cliquer OK, le sous répertoire destiné à contenir la description du circuit synthétisé est alors créé.



4.1.2 Choix de la technologie cible

A l'aide de l'icône « Setup Design », ouvrir la fenêtre « Project Settings ». Il peut être nécessaire de cliquer l'onglet « Design » dans la colonne de gauche pour voir cette icône. Choisir :

- Technology : Xilinx Spartan3
- Device : 3s200ft256
- Speed Grade : -5

Cocher Set Frequency et donner la fréquence de cadencement de la carte cible (50 MHz). Laisser les autres options par défaut, Valider (OK) et sauvegarder le projet.

4.1.3 Définir les modèles à synthétiser

- L'icône « Add Input Files » (colonne de gauche onglet « Design »), permet de spécifier les fichiers qui contiennent les entités et les architectures à synthétiser.
- Se placer dans `${TP_PATH}/vhd` et choisir tous les fichiers source du filtre (fichiers * .vhd).
- **ATTENTION** : Dans Precision, le fichier contenant le module englobant (top module) du projet est mis en gras et en bas de la liste « Input Files » dans la zone « Project Files ». Dans ce projet, le module englobant est contenu dans le fichier filter.vhd. Il faut donc le désigner comme tel. Pour cela, sélectionner le fichier filter.vhd cliquer droit et sélectionner « Move File to Bottom ».

4.1.4 Options

Il est possible d'optimiser la synthèse, soit en minimisant la surface, soit en améliorant les performances temporelles. Pour accéder aux options d'optimisation sélectionner le menu « Tools | Options | Optimization Settings ».

4.1.5 Spécifier le format de sortie du résultat de synthèse

Le menu « Tools | Options | Output Options » permet de définir le format de sortie de la liste de porte (netlist) obtenue après synthèse ainsi que le nom du fichier dans lequel la synthèse sera enregistrée.

- Spécifier le nom du fichier de sauvegarde dans le champ « Output File Base Name » : filtre_synth.
- Sélectionner les formats EDIF et VHDL.
- Décocher l'option « Generate Vendor Constraint File ».

Avec ces choix la synthèse produira deux fichiers :

1. filtre_synth.edf nécessaire à l'étape de placement et routage qui suit
2. filtre_synth.vhd destiné à la simulation après synthèse

Ces fichiers seront enregistrés dans le sous répertoire FiltreNum_impl_1 selon notre exemple.

4.1.6 Réalisation de la synthèse

La synthèse peut être lancée à partir des icônes « Compile », pour la première étape, et « Synthesize » pour la synthèse finale. Il peut être nécessaire de cliquer l'onglet « Design » dans la colonne de gauche pour voir ces icônes. Le compte-rendu de ces opérations apparaît dans l'onglet « Transcript ».

4.1.7 Visualisation du résultat de la synthèse

Les résultats de la synthèse sont visualisables par le biais des icônes de l'onglet « Schematics » dans la colonne de gauche :

- L'icône « View RTL Schematic » permet de visualiser la vue RTL de la modélisation VHDL.
- L'icône « View Tech Schematic » permet de visualiser le schéma de portes obtenue avec la bibliothèque de portes logiques de la technologie ciblée.
- L'icône « View Critical Path » permet de visualiser le chemin critique dans le circuit synthétisé.

Pour connaître les statistiques de performances du circuit synthétisé, cliquer sur l'icône « Area report » pour le taux d'occupation du FPGA et sur l'icône « Report Timing » pour les timings. Ces icônes deviennent visibles en cliquant sur l'onglet « Design Analysis » dans la colonne de gauche. Penser à relever les valeurs.

4.1.8 Enregistrement de la synthèse

Le résultat de la synthèse est conservé dans le fichier filtre_synth.edf.

- Vérifiez que les fichiers filtre_synth.edf et filtre_synth.vhd sont bien présents dans le répertoire créé pour la synthèse (FiltreNum_impl_1 dans notre exemple).
- Procéder ensuite à l'enregistrement du projet en sélectionnant « Project | Save Project » avant de quitter l'outil Precision.

4.2 Placement et routage cible FPGA avec l'outil ISE de Xilinx

L'objectif de cette étape du flot de conception cible FPGA est d'implanter physiquement le filtre sur la Spartan 3 de Xilinx.

4.2.1 Le fichier des contraintes de l'utilisateur

Même si un FPGA est une technologie très flexible, il n'est pas possible d'affecter n'importe quel signal à n'importe quelle broche. En effet, les broches sont regroupées en ports qui peuvent être associés à des fonctions spéciales comme un port sériel, un port JTAG ou un décodeur pour un afficheur. Même dans le cas le plus courant d'entrées/sorties tout ou rien à usage général, deux ports différents n'acceptent pas forcément la même plage de tension (par exemple un domaine en 0-5V et un autre en 0-3V), la sortance peut être différente et certaines terminaisons sont soit en entrée soit en sortie alors que d'autres sont bidirectionnelles. En outre, certaines broches sont dédiés aux alimentations et au moins une est réservée pour le signal d'horloge. Lorsque le FPGA est déjà câblé sur une carte électronique prête à l'emploi, comme celle utilisée dans ce TP, les contraintes sur l'affectation des ports et des broches sont encore plus importantes. Par exemple, une entrée tout ou rien initialement bidirectionnelle sur le FPGA vide, peut, une fois le FPGA câblé, être désormais en entrée uniquement (cas d'un bouton poussoir) ou en sortie seulement (cas d'une LED).

Il faut donc renseigner l'outil de placement et routage sur la correspondance qu'il doit respecter entre les ports physiques du FPGA et les ports formels définis par l'entité du module englobant dans la description matérielle. Il faudra également spécifier la direction pour les ports bidirectionnels, la plage de tension, la sortance, etc. Toutes ces informations sont consignées dans le fichier des contraintes de l'utilisateur (User Constraint File – UCF).

Dans le cas de ce TP, le fichier des contraintes de l'utilisateur a été préparé pour vous et placé dans le répertoire \$ { TP_PATH } /fpga/userconstraints. Regarder et comprendre ces contraintes sans modifier ce fichier !

4.2.2 Lancement de l'outil ISE et paramétrage du projet

Rappel : vous devez exécuter le script de configuration config_FPGA avant de pouvoir utiliser l'outil ISE.

- Se placer dans le répertoire \$ { TP_PATH } /fpga/par et démarrer l'outil en exécutant la commande ise&
- Sélectionner le menu «Project | New Project»
- Dans la fenêtre «New Project Wizard – Create New Project», renseigner
 - le nom du projet pour le placement et le routage, par exemple « FiltreNum »
 - l'emplacement du répertoire pour le placement et le routage le « Working Directory », Indiquer le chemin \$ { TP_PATH } /fpga/par
 - sélectionner « EDIF » dans la liste déroulante « Top-level source type »
 - cliquer « Next »
- Dans la fenêtre « New Project Wizard – Import EDIF/NGC Project », indiquer

- le fichier edf créer lors de la synthèse (cf. § 7), dans la zone de saisie « Input Design »
- le fichier \${TP_PATH}/fpga/userconstraints/filtre.ucf dans la zone de saisie « Constraint file »
- Décocher la case « Copy the constraints file to the project directory » de façon à ce que le fichier des contraintes de l'utilisateur *.ucf ne soit pas dupliqué.
- cliquer « Next »
- Dans la fenêtre « New Project Wizard – Project Settings », les caractéristiques du FPGA (Family, Device, Package, Speed) doivent être automatiquement remplies. S'assurer que le champ « Preferred language » est positionné sur VHDL.
- Cliquer « Next » puis « Finish »

4.2.3 Placement et routage

- Les commandes de placement et routage sont regroupées dans le panneau de gauche « Processes » au sein d'une arborescence appelée « Implement Design » (voir figure 7).
- Les commandes de placement et routage donnent lieu à des rapports qui sont accessibles en dépliant l'arborescence.
- Le rapport général (Design Overview => Summary) qui peut être visualisé par l'onglet « Design Summary » dans le panneau de droite, existe aussi sous forme d'un fichier HTML (FILTER_summary.html) dans le répertoire du projet de placement routage. Il est ainsi possible de consulter ce rapport sans devoir relancer l'outil.

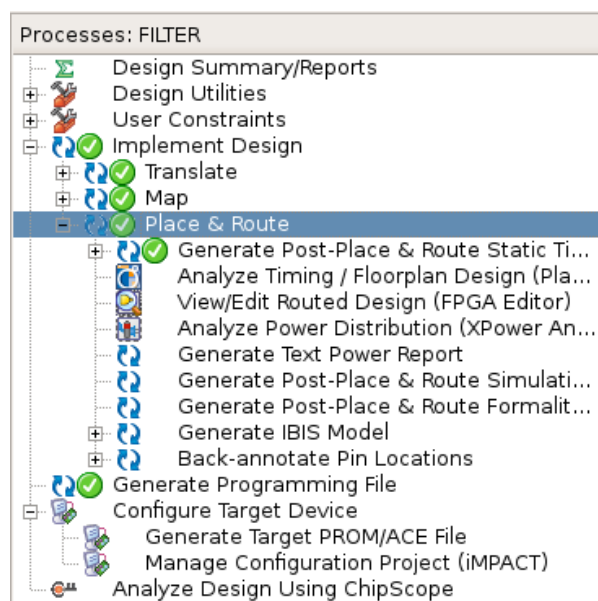


FIGURE 7 – Arborescence du flot de placement routage

Génération des modèles de simulation temporelle : Double-cliquer sur la ligne « Generate Post-place & Route Simulation Model » dans le panneau « Processes ». Dans le Process Properties modifiez le model de simulation en VHDL, par défaut il est en Verilog.

Emplacement des fichiers produits pour la simulation temporelle : en supposant que l'entité du module englobant (top module) ait été appelée « FILTER » et que le nom du projet pour le placement et le routage soit « FiltreNum », les fichiers de timing nécessaires à la simulation après placement et routage sont automatiquement baptisés FILTER_timesim.sdf et FILTER_timesim.vhd. Ils sont générés dans le sous répertoire : \${TP_PATH}/fpga/par/FiltreNum/netgen/par/

4.2.4 Génération du fichier de configuration du FPGA

Cette opération va créer un fichier qui sera chargé dans le FPGA pour changer sa configuration de façon à refléter le circuit que vous venez de placer et router. Ce type de fichier est appelé bitstream en anglais : double-cliquer sur la ligne « Generate Programming File » dans le panneau « Processes ».

Attention : le brochage peut ne pas avoir été effectué correctement. Vérifier dans le fichier que le brochage a été effectué. Si ce n'est pas le cas, refaire toutes les étapes précédentes.

5 Test du filtre sur cible FPGA

5.1 Programmation de la carte FPGA Xilinx Spartan3 et test en environnement réel

Se reporter aux annexes (voir répertoire doc) pour connaître l'emplacement et le branchement des connecteurs sur le banc de test.

5.1.1 Configuration du FPGA

L'outil de programmation des FPGA Xilinx s'appelle iMPACT. Pour charger un fichier de configuration « bitstream » dans un FPGA Xilinx, procéder comme suit :

- Mettre sous tension la carte Spartan 3 Starter Board.
- Dans une fenêtre terminal, taper la commande '**impact**'.
- Valider par « OK », le message d'information disant que le répertoire de travail a été changé et a été positionné sur le répertoire courant.
- Dans la fenêtre « iMPACT Project », sélectionner « create a new project » et désigner le répertoire où créer le projet (\$ { TP_PATH } /fpga/par) grâce au bouton « Browse ». Cliquer « OK ».
- Dans la fenêtre « iMPACT – Welcome to iMPACT », s'assurer que « Configure devices using Boundary-Scan (JTAG) » est sélectionné et que la liste déroulante est positionnée sur « Automatically connect to a cable and identify Boundary -Scan chain » puis cliquer « Finish ».
- Si le message « Warning :iMPACT :923 – cable not found » apparaît, c'est que le port JTAG n'est pas connecté au port USB du PC (attention au sens du câble) ou que la carte FPGA n'est pas sous tension ou que le pilote USB n'est pas chargé (cf. ci-dessus les scripts à sourcer).
- Une fenêtre « Assign New Configuration File » apparaît pour indiquer la configuration du composant xc3s200, : c'est le FPGA. Parcourir l'arborescence pour retrouver le fichier de configuration généré au §III.2.4) (bitstream d'extension *.bit). Sélectionner ce fichier puis cliquer « Open ».
- La fenêtre « Assign New Configuration File » réapparaît pour la configuration d'un composant xcf02s : c'est une EEPROM présente sur la carte FPGA mais qui n'est pas utilisée dans ce TP. Continuer en cliquant « Bypass » puis valider par OK le message d'information suivant.
- Dans l'onglet « Boundary Scan » de la fenêtre d'iMPACT (cf. figure 8), faire un clic droit sur l'icône du FPGA xc3s200,, puis sélectionner « Program »
- Le FPGA est reconfiguré en une dizaine de secondes.

5.1.2 Test du circuit implanté sur FPGA

En vous aidant du schéma d'implantation simplifié et de la photo du banc de test matériel (cf. § IV des annexes), assurez vous que le générateur basse fréquence (Géné BF) est bien connecté à l'entrée AnaIn de la chaîne A et renvoyée sur une des entrées de l'oscilloscope. Vérifiez que la sortie AnaOut de la chaîne A est reliée à l'oscilloscope. Visualisez les signaux de contrôles du CAN et du CNA de la chaîne A.

A l'aide du Géné BF et de l'oscilloscope, contrôlez le fonctionnement du filtre numérique.

Afin de lever les doutes sur la chaîne de numérisation, en cas de non fonctionnement de votre circuit, les fichiers de configuration de référence suivants sont mis à votre disposition dans le répertoire /softslin/configCAO/Xilinx :

- le fichier test_sans_filtre_spartan3.bit valide le fonctionnement de la chaîne de numérisation sans filtrage
- le fichier valid_cna_spartan3.bit valide le fonctionnement du convertisseur numérique vers analogique

ATTENTION : s'il n'est pas possible de faire fonctionner le banc de test avec les fichiers de configuration de référence indiqués ci-dessus, réinitialiser la carte FPGA en débranchant le connecteur d'alimentation, le port JTAG et le Géné BF.

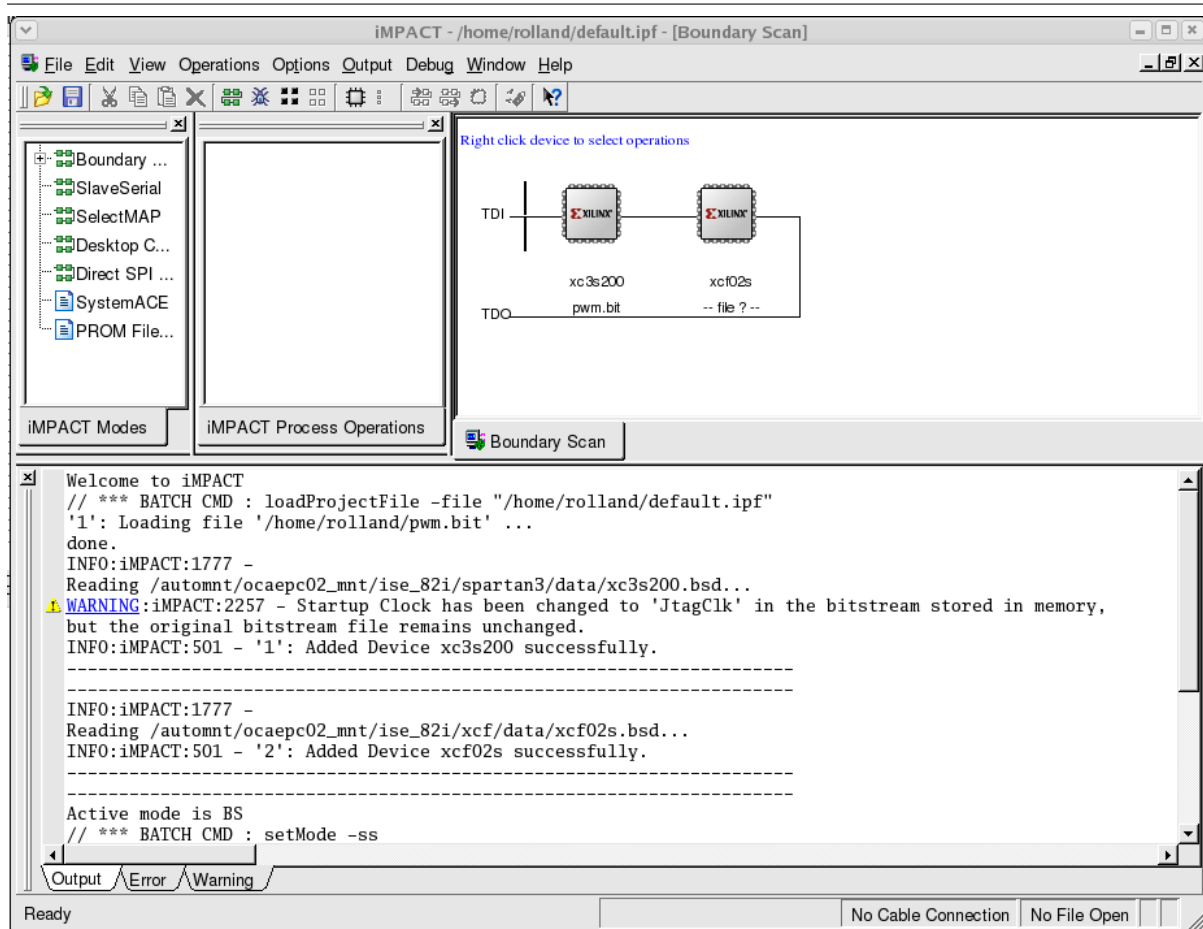


FIGURE 8 – Fenêtre d'iMPACT

6 Travail demandé

6.1 Travaux à réaliser et questions à traiter

1. Réaliser la synthèse du filtre à partir de l'ensemble du code source du répertoire vhd sur la technologie Xilinx Spartan3.
2. Le résultat de synthèse est-il conforme à votre intuition ? commenter.
3. A quoi correspond l'opération « compile » dans Precision ?
4. Qu'effectue l'outil Precision lors de la phase « synthesize » ? Lors de la phase « optimize » ?
5. Réaliser le placement et routage sur le FPGA Spartan3 de Xilinx.
6. Regarder le routage obtenu dans Xilinx FPGA Editor.
7. Que dire du nombre de LUTs utilisées après placement-routage ? Qu'en est-il du nombre de registres ?
8. À quoi sert le fichier ucf ? Que contient-il ?
9. Simuler le filtre complet obtenu après placement et routage, toujours dans l'environnement de test développé, mais en tenant compte des délais de propagation introduits dans les portes et les interconnexions. Vérifier que le fonctionnement du filtre reste correct.
10. Qu'est-ce que le packaging VITAL ? À quoi sert-il ?
11. Effectuer la validation du filtre sur la carte de test en chargeant le fichier de configuration (fichier *.bit) dans le circuit Xilinx à programmer.
12. Produire un compte-rendu final.

6.2 Compte-rendu final

Un compte-rendu décrivant l'ensemble de vos travaux sera à rendre à la fin de la quatrième séance. Il devra en particulier inclure les indications suivantes avec une explication :

- les réponses aux différentes questions,
- le programme de test bench, en détaillant la modélisation de l'environnement et en argumentant avec précision pourquoi ce test permet d'assurer que le fonctionnement du filtre est correct,
- le rapport commenté sur la surface du filtre (# CLB, # FF, # IO) après la synthèse et après le placement et routage,
- le délai du chemin critique après le placement et routage,
- en déduire à quelle vitesse maximale le filtre pourrait fonctionner sur le FPGA choisi, après la synthèse et après le placement et routage,
- le schéma niveau portes logiques après la synthèse,
- à la fréquence de fonctionnement choisie de 50 MHz, quelle est la fréquence maximale d'échantillonnage du système ? Répondre à cette question de deux façons différentes : calcul théorique et mesure,
- préciser la fréquence de coupure du filtre à -3dB (résultat de mesure).