Tutorial Rapide des Outils de Synthèse et Simulation de programmes écrits en VHDL

AVANT DE COMMENCER

Organisation générale

Procédures de connexion et de déconnexion et manipulation de répertoires et fichiers :

Login "xph2seixxx" xx =001…031
Passwd : ??? pour tous les comptes
Dans une fenêtre xterm, placez vous dans le répertoire de travail concernant le TP
cd TP_VHD

ATTENTION (lire ce paragraphe avant toute manipulation de fichiers !) : Formation UNIX rapide !

- Le parcours des répertoires se fait avec les commandes LINUX (**cd nom de répertoire**, or **cd . .** pour remonter d'un niveau). L'affichage du contenu de répertoire se fait avec les commandes **ls**, ou **ll**.
- L'ouverture de fichier se fait avec la commande **nedit &** (on retrouve plusieurs commandes du Word…)
- L'impression de fichier se fait par commande, dans un terminal : **a2ps –Pbelledonne nom_de_fichier.vhd**, ou tout autre fichier ASCII (**et uniquement ASCII)**. Si vous souhaitez imprimer des documents pdf (ou postscript), utiliser la commande lpr – **Pbelledonne nom_de_fichier.pdf (ou ps)**

Si vous vous trompez de commande d'impression, l'imprimante imprimera des centaines de pages, bloquant ainsi tout accès à l'impression de n'importe quel utilisateur.
Réfléchissez deux fois avant d'exécuter les commandes d'impression et de suppression de fichiers. LINUX ne vous demande pas de confirmations !

Déconnexion :
Menu Actions -> Logout
ET ensuite taper logout !

Configuration de l'environnement VHDL

La configuration de l'environnement VHDL est réalisée par l'intermédiaire d'un fichier qui contient un ensemble de commandes. Ce fichier est par défaut dénommé "modelsim.ini" (un exemplaire est disponible dans le répertoire ~xph2seixxx/TP_VHD/config). Pour que ce fichier soit utilisé par défaut lors de l'exécution d'une commande (compilation, simulation…), il est nécessaire de définir une variable d'environnement dénommée "MODELSIM" comme suit : export MODELSIM=$HOME/TP_VHD/config/modelsim.ini Cette variable est définie dans le fichier config_FPGA à la racine de votre repertoire de TP.

Dans le fichier "modelsim.ini", doivent apparaître les liens entre bibliothèques logiques et physiques. Par exemple, si vous créez une bibliothèque de nom logique "lib_COMP", il faut définir le répertoire physique dans lequel elle est stockée :

lib_COMP = $HOME/TP_VHD/vhd/lib_COMP

L'ajout de ces liens se fera en utilisant la commande vmap (voir la section utilisation des outils de simulation)

Structure du TP

Répertoire contenant les codes sources des modèles du circuit à concevoir :

~xph2seixxx/TP_VHD/vhd

On y trouve des entités, des architectures, des paquetages, des configurations, une bibliothèque, etc.
Attention aux noms !
Ex : Entity : ALU

Architectures : A
Bibliothèque : lib_COMP

Répertoire contenant les codes sources des environnements de tests :

~ xph2seixxx /TP_VHD/bench

On y trouve des entités, des architectures, des paquetages, une bibliothèque, etc.

Ex : Entity : test_ALU
Architectures : test1
Bibliothèque : lib_BENCH

Répertoire contenant le résultat de la synthèse :

~ xph2seixxx /TP_VHD/synth

On y trouve des entités, des architectures, des paquetages, une bibliothèque, etc , résultant d'une opérationd e synthèse. On y stockera ici donc la netlist synthétisée. Utiliser les mêmes noms que dans les fichiers avant synthèse !

Ex : Entity : ALU
Architectures : A
Bibliothèque : lib_SYNTH

AVANT TOUT TRAVAIL EN VHDL PENSER A EXECUTER LA COMMANDE
source config_FPGA
A PARTIR DU REPERTOIRE TP_VHD et cela dans toute nouvelle fenêtre de commande.

UTILISATION DES OUTILS DE SIMULATION

Création d'une bibliothèque de compilation

Cette opération est nécessaire avant la première compilation d'une unité de conception dans la bibliothèque lib_NOM !

vlib lib_NOM
***ex de commande* : vlib lib_COMP**

Création d'un lien entre le nom de la bibliothèque de compilation et son emplacement physique

Cette opération est nécessaire avant la première compilation d'une unité de conception dans la bibliothèque lib_NOM !

vmap lib_NOM emplacement/lib_NOM
***ex de commande* : vmap lib_COMP $HOME/TP_VHD/vhd/lib_COMP**

Compilation dune description VHDL

La commande suivante effectue la compilation des unités de conception contenues dans le fichier source "source.vhd", et stocke les résultats compilés sans erreur dans la bibliothèque de travail lib_NOM :

vcom +acc –work lib_NOM ficher_ a_compiler.vhd

Simulation avec ModelSim

La commande suivante lance le simulateur (à exécuter dans le répertoire bench.) :

vsim &

Tous les fichiers présents dans le répertoire de travail courant seront accessibles en cours de session, par exemple les fichiers de commande du simulateur (cf. §Créer et utiliser un fichier de commandes).

1. Charger une unité de conception à simuler

Dans la fenêtre Workspace, qui représente la structure du projet, sélectionner la bibliothèque lib_BENCH et charger le composant à simuler.

Le chargement d'une nouvelle unité à simuler pourra être fait par le menu Simulate -> Start simulation. Dans ce menu se trouve un onglet pour spécifier l'unité de temps à utiliser pendant la simulation ("ns" est l'unité par défaut).

2. Observer les signaux de l'unité de conception

La fenêtre "Objects" est ouverte lors du chargement d'une unité de conception. Cette fenêtre montre les signaux d'entrée/sortie du bloc sélectionné dans la fenêtre Workspace (signaux définis dans l'entité).
La fenêtre "Objects" peut être obtenue aussi à partir du menu View -> Debug Windows.

3, Observer les chronogrammes (ondes) de simulation (waves)

La définition des signaux à afficher sous forme de chronogrammes peut être réalisée dans la fenêtre"Objects" (menu pop-up obtenu par le bouton droit de la souris, commande Add to Wave). La liste de signaux à ajouter peut correspondre à :

"Selected signals" : signaux sélectionnés dans la fenêtre" Objects" avant d'activer la commande.
"Signals in Region" : ports de l'entité.
"Signals in Design" : tous les signaux internes.

Il est aussi possible d'utiliser la commande "add wave nom_du_signal" ou "add wave /*" pour afficher tous les signaux de l'entité de plus haut niveau hiérarchique.

4,. Construire des stimuli

Bien qu'il soit fortement recommandé de construire les stimuli avec un programme VHDL de test, voici un moyen simple pour créer rapidement des signaux de façon interactive à l'aide du simulateur.
La commande de base pour construire des formes d'ondes est la commande "force" (aussi disponible à partir du menu pop-up de la fenêtre "Objects").

Exemples :	force x 000	positionne le signal x à la valeur 000 à partir de l'instant de simulation courant
	force x(1) 1	positionne le BIT 1 du signal x à la valeur 1 à partir de l'instant de simulation courant
	force clk 1 30, 0 80 -repeat 100	crée une forme d'onde de période 100, avec clk = 1 à l'instant 30 et clk = 0 à l'instant 80

5.Exécuter la simulation

L'exécution est lancée à partir du menu Simulate →Run.
La commande "Step" permet de suivre l'évolution pas à pas de la simulation en visualisant le code source.
La commande "run time" exécute "time" unité de temps de simulation (ex : run 100).
La commande suivante exécute la simulation pour toujours : run time'high.

6. Analyser les résultats de simulation

Dans la fenêtre des chronogrammes, le curseur peut être déplacé ; les valeurs des signaux sont affichées, ainsi que le temps pointé par le curseur. Il est possible d'ajouter d'autres curseurs, de façon à effectuer des mesures d'intervalles de temps (menu "Add -> Cursor").

7. Sauvegarder les résultats de simulation

Il est possible de sauvegarder les chronogrammes dans un fichier Postscript :
menu "File -> Print Postscript", impression dans un fichier.

8. Créer et utiliser un fichier de commandes

Il peut être utile de pouvoir reproduire les stimuli qui ont été construits pour tester une unité de conception (Attention, cela fait double emploi avec les programmes de test écrits en VHDL). Pour cela, il faut créer un fichier qui reproduit les commandes utilisées pour créer les stimuli. Le nom du fichier n'est pas contraint, l'extension si elle existe peut être quelconque. Il est conseillé d'ouvrir un fichier texte dans le répertoire courant, et de le mettre à jour au fur et à mesure que les stimuli sont élaborés.
La commande "do nom_fichier" permet d'exécuter le fichier de commandes et ainsi reproduire les stimuli.

UTILISATION DES OUTILS DE SYNTHESE

Rappel de règles simples d'écriture de code VHDL

En VHDL, un processus est synthétisé en une logique combinatoire pure si :
- tous les signaux lus appartiennent à la liste de sensibilité du processus,
- toutes les variables locales sont affectées avant d'être lues,
- tous les signaux de sortie sont affectés, et ceci dans toutes les branches du programme.

En VHDL, un processus est synthétisé en une logique séquentielle si :

- **on utilise une liste de sensibilité**
- **les signaux asynchrones se trouvent en liste de sensibilité, avec la clock**
- **il y a une clock et une condition de clock edge**
- **il n'y a pas de else au statement if clock'event and clock='1', etc**

Synthèse de programmes VHDL avec Precision (Mentor)

Lancer l'outil "precision"

Pour lancer l'outil de synthèse logique Precision, se placer dans le répertoire « **synth** » et entrer la commande suivante, dans un terminal :

"**precision &**"

Créer un nouveau projet par « Project -> New Project » et donner le nom de votre choix, par exemple ALU. Le logiciel va créer deux répertoires à l'intérieur du synth, « ALU_impl » pour la description du projet et « ALU_temp » pour les fichiers temporaires. Ne les effacez pas !

A. Demarrage

Définir les modèles à synthétiser :

L'icône "Add Input Files" permet de spécifier les fichiers qui contiennent les entités et architectures à synthétiser.
Working Directory : "~xxx/TP_VHD /synth". Très important !
Open files : se placer dans "~x1appxxx/TP_VHD/vhd" et choisir le fichier de l'UAL.(du type *.vhd)

Choix de la technologie cible :

A l'aide de l'icône « Setup »choisir :
Famille : Xilinx ®Spartan3
Circuit : 3s200ft256
Speed grade : -5
Laisser les autres options par défaut. Sélectionner la fréquence, ou le délai (sans changer les valeurs).

B. Options d'optimisation:

Dans le menu Tools/Set Options : Il est possible d'optimiser la synthèse, soit en minimisant la surface, soit en améliorant les performances temporelles par la commande « Optimization ». Il est aussi possible de choisir le codage des FSMs (Input/Fsm encoding).

C. Spécifier le format de sortie du résultat de synthèse :

Le menu « Tools -> Set options », puis "Output" permet de définir le format de sortie de la "netlist" obtenue après synthèse. Il faut spécifier le nom du fichier de sauvegarde :
"ALU_synth" ainsi que le format "VHDL".
Dévalider l'option « Generate Vendor Constraint File ».

E. Synthèse : menus "Compile" et "Synthesize"

La synthèse peut être lancée à partir des menus « Compile », pour la première étape, et « synthesize » pour la synhèse finale, " accessibles à gauche de l'écran.

Le compte-rendu de ces opérations apparaît dans la fenêtre transcript.

F. Visualisation du résultat de la synthèse :

Pour visualiser le schéma portes produit par l'étape de synthèse, il suffit de double-cliquer sur l'un des symboles du champ « Design center » représentant des portes logiques.

Le premier permet de visualiser le schéma correspondant à la vue RTL de la modélisation VHDL.

Le deuxième permet de visualiser le résultat de la synthèse obtenu avec les portes logiques de la bibliothèque cible spécifiée.

Pour avoir la description de l'étape de synthèse et la description synthétisée du du composant, cliquer sur le symbole « Area report ".

Vérifier la présence du fichier «alu_synth.vhd » dans le répertoire « ALU_impl », résultat de la synthèse, avant de passer à la suite.

Lancer « Project -> Save project » avant de quitter « precision ».

Annexe - Important

Paquetages standard

Cette partie présente les paquetages mis à la disposition du concepteur, et qui définissent un environnement standard pour la simulation et la synthèse. Les codes sources sont dans le répertoire : /softs/modeltech/vhdl_src/

• Paquetage "STANDARD" : ce paquetage définit les types "boolean, bit, character, severity_level, integer, real, time, string, bit_vector, les sous types natural, positive ainsi que la fonction "now". Le code source se trouve dans le répertoire : /softs/modeltech/vhdl_src/std/

• Paquetage "TEXTIO" : ce paquetage définit les primitives d'entrée/sortie ASCII de VHDL. Le code source se trouve dans le répertoire : /softs/modeltech/vhdl_src/std/

• Paquetage "STD_LOGIC_1164" : ce paquetage définit un type bit étendu à neuf états dénommé "STD_ULOGIC" et un sous type résolu dénommé "STD_LOGIC". Il définit également les types composites "STD_ULOGIC_VECTOR et STD_LOGIC_VECTOR" ainsi que les fonctions logiques usuelles sur ces types. Le code source se trouve dans le répertoire : /softs/modeltech/vhdl_src/ieee/

• Paquetage "NUMERIC_BIT et NUMERIC_STD" : ces paquetages définissent les fonctions arithmétiques qui permettent de travailler sur des vecteurs de BIT et STD_LOGIC_VECTOR respectivement. Le code source se trouve dans le répertoire : /softs/modeltech/vhdl_src/ieee/

• Paquetage "STD_LOGIC_ARITH" : ce paquetage définit les fonctions arithmétiques de la même façon que les paquetages standard NUMERIC_BIT et NUMERIC_STD, mais il assure la compatibilité avec les versions antérieures des outils de synthèse. Le code source se trouve dans le répertoire : /softs/modeltech/vhdl_src/arithmetic/ le nom du fichier source est "std_arit.vhd".

Il suffit donc d'inclure les clauses suivantes au début des programmes pour avoir accès à ces paquetages :

```
library IEEE ;
use IEEE.STD_LOGIC_1164.all ;

et
use IEEE.NUMERIC_STD.all ;

ou

library arithmetic ;
use arithmetic.STD_LOGIC_ARITH.all ;
```