

# Notice d'utilisation des outils logiciels pour le VHDL

Ce petit guide doit vous permettre de vous familiariser avec l'outil de simulation Modelsim. L'ensemble des outils logiciels utilisés pour le VHDL peuvent être installé gratuitement sur Windows ou Linux.

Lien pour télécharger le logiciel Modelsim version 9.1 sur **Windows** :

<https://dropsu.sorbonne-universite.fr/s/qXQZ7Q8ZXsEdaq>

## 1 Edition du code VHDL

Modelsime fournit un éditeur VHDL mais il n'est pas très performant. Vous pouvez utiliser Modelsim pour faire la simulation en utilisant un éditeur externe.

Par exemple, sur Windows, NotePad++ ou VSCode sont des éditeurs de VHDL gratuits et fort agréables. Ils appliquent une coloration syntaxique pour le VHDL et permettent d'écrire en mode colonne (très pratique pour l'édition du code VHDL)

Notepad++ : <https://notepad-plus-plus.org/download/v7.5.5.html>

VS Code avec extension pour le VHDL : <https://code.visualstudio.com>

## 2 Compilation et simulation avec Modelsim

Il existe deux méthodes pour simuler un composant, soit vous le simulez en éditant des stimuli avec le simulateur, soit en écrivant un banc de test (méthode vivement conseillée). Pour gagner du temps on peut aussi utiliser un script de simulation.

Lancer le logiciel Modelsim :

- sur linux en tapant la commande vsim dans un terminal
- sur Windows en allant dans le menu **Démarrer → Programmes → Programmation → Modelsim**

### 2.1 Simuler en éditant des stimuli

Une fois Modelsim lancé, il faut changer de répertoire afin de se placer dans le répertoire du fichier VHDL à simuler :

**File → Change Directory** → sélectionné le répertoire de votre fichier VHDL

Créer une nouvelle librairie de travail :

**File → New → Library**

- Cochez le bouton *a new library and a logical mapping to it*
- Mettre le nom *work* pour *Library Name* et *Library Physical Name*
- Cliquez sur Ok

Compiler votre ou vos fichier(s) VHDL :

**Compile → Compile**

- Choisissez le fichier VHDL à compiler et cliquez sur compile
- Refaites la même opération avec les autres fichiers.
- Quand vous avez terminé cliquez sur Done.

**Remarque :** Si vous avez plusieurs fichiers, commencez toujours par ceux qui sont de plus basse hiérarchie et finissez par celui qui est de plus haute hiérarchie qu'on nomme souvent le top-level.

Lancer le simulateur :

**Simulate → Simulate**

- Développer work,
- Cliquez sur le nom de l'entité à simuler
- Cliquez sur OK


Ouvrir la fenêtre signals et wave :

**View → Objects**

**View → Wave**

Sélectionnez les signaux dans la fenêtre Objects et faites-les glisser jusqu'à la fenêtre Wave.

Dans la fenêtre Wave, sélectionnez un signal d'entrée, cliquez sur le bouton droit de la souris et vous pouvez mettre une valeur constante à votre signal en choisissant Force, ou le faire varier comme un signal périodique carré en choisissant Clock. Affectez une valeur valide à tous vos signaux d'entrées.

Appuyez sur l'icône . A chaque appui sur cette icône vous simulez un pas de temps, ce pas étant par défaut de 100 ns.

Pour terminer la simulation :

**Simulate → End Simulation**

## 2.2 Simuler en utilisant un banc de test (test-bench)

Le banc de test est un fichier VHDL qui permet de générer les stimuli nécessaires à tester la validité de votre description. Lorsqu'il est complet il permet également de vérifier les sorties de votre design, on parle alors d'un banc de test auto-vérifiant.

Pour faire la simulation avec un banc de test, il faut reprendre exactement les mêmes étapes que la partie 2.1 sauf qu'il faut compiler le composant à simuler (design.vhd) et le banc de test (design\_tb.vhd). Ensuite il faut lancer la simulation du banc de test et non pas du composant à simuler :

**Simulate → Simulate**

- Développer work,
- Cliquez sur le fichier du banc de test (design\_tb.vhd)
- Cliquez sur OK

Ouvrir les fenêtres Objects et Wave :

**View → Objects**

**View → Wave**

Sélectionnez les signaux dans la fenêtre Objects et faites-les glisser jusqu'à la fenêtre wave. Lancer la simulation :

Cliquer sur l'icône  ou **Simulate → Run → Run -All**

## 2.3 Simuler en utilisant un script de simulation tcl/tk

L'utilisation d'un script de simulation permet de gagner beaucoup de temps à chaque simulation, celui-ci effectue un certain nombre d'étapes en une seule. Le script utilise le langage tcl/tk qui très utilisé dans les flots VHDL pour écrire des scripts de simulation ou de synthèse (une ligne de commentaire commence par #).

**Lorsque vous lancez Modelsim, il faut toujours commencer par changer de répertoire, afin de se placer dans le répertoire de travail : répertoire dans lequel se trouve le script de simulation et le banc de test.**

**File → Change Directory →** sélectionné le répertoire de travail

Voici comment écrire votre script de simulation :

```
# création de la librairie de travail work
vlib work

#compilation des fichiers vhdl, l'ordre de compilation est important
#commencer par le package, les composants, le top-level et enfin
# le test-bench
#vcom -93 packages.vhd
#vcom -93 sources.vhd
#vcom -93 testbench.vhd
#Exemple :
vcom -93 counter_pkg.vhd
vcom -93 counter.vhd
vcom -93 counter_tb.vhd

#lancer la simulation avec le nom du testbench (si il existe)
#sinon avec le nom du top-level design
#pour chacun des cas précisez le nom de l'entité, pas le nom du fichier
vsim -novopt counter_tb(RTL)
#entre parenthèse vous pouvez préciser le nom de l'architecture

#pour visualiser tout les signaux du design:
view signals
add wave *

#lancer la simulation
run -all
```

Sauvegarder le fichier avec le nom **simu.do**

Pour l'exécuter, lancer Modelsim et taper la commande **do simu.do** dans le transcript.