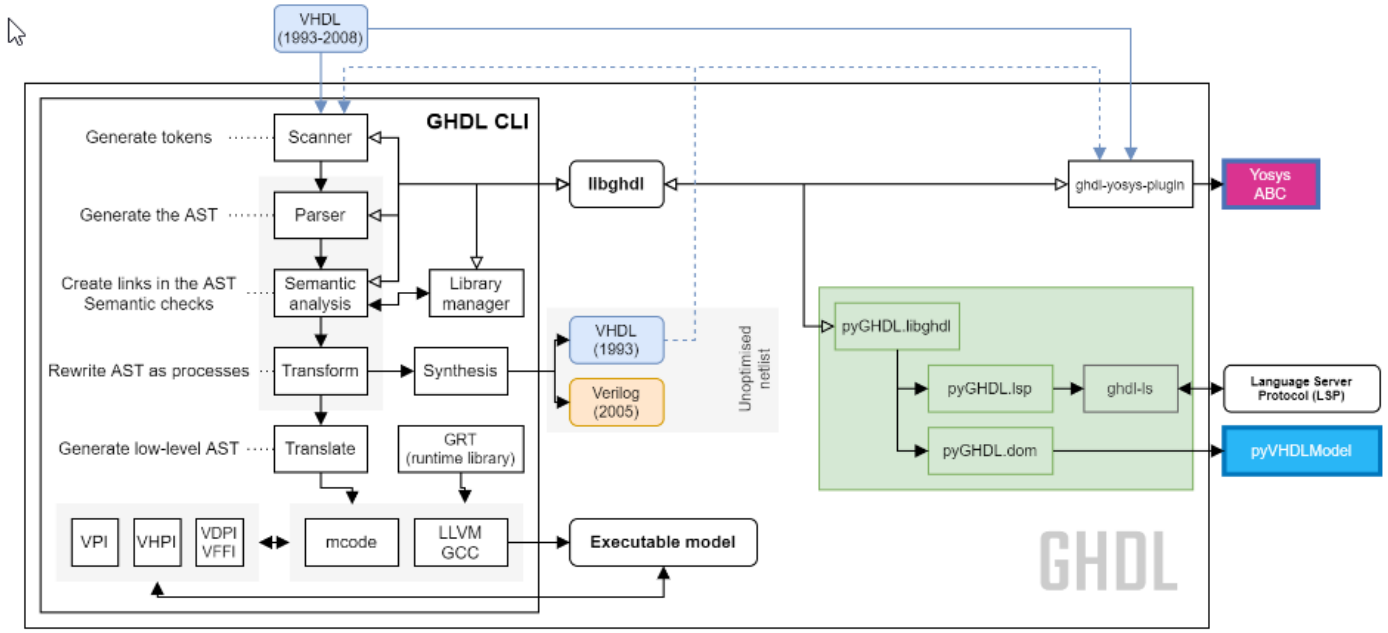
Outils de simulation VHDL

INF3500
Conception et réalisation de systèmes numériques

POLYTECHNIQUE
MONTRÉAL

GHDL et gtkWave (simulation VHDL)

- Vous pouvez installer ghdl et gtkwave sur Windows avec des versions Windows. Vous pouvez aussi vous installer une distribution linux sur votre PC Windows et pouvoir rouler GHDL et mode CLI sur Linux.



GHDL : Windows
Installation de GHDL et gtkWave

GHDL

- Aller sur le site suivant : https://github.com/ghdl/ghdl et trouver la version la plus récente pour Windows (par exemple : ghdl-MINGW32.zip).
- Créer un répertoire EDA sur le C de votre ordinateur
- Ouvrir le fichier ZIP et extraire le répertoire GHDL pour le mettre dans votre répertoire EDA.
- Pour partir l'outil, vous devez aller dans le répertoire bin/ et utiliser l'exécutable : gtkwave.exe

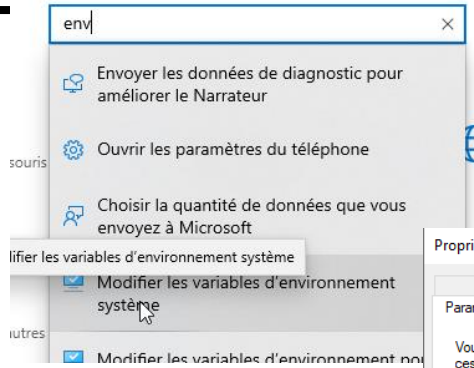
gtkWave

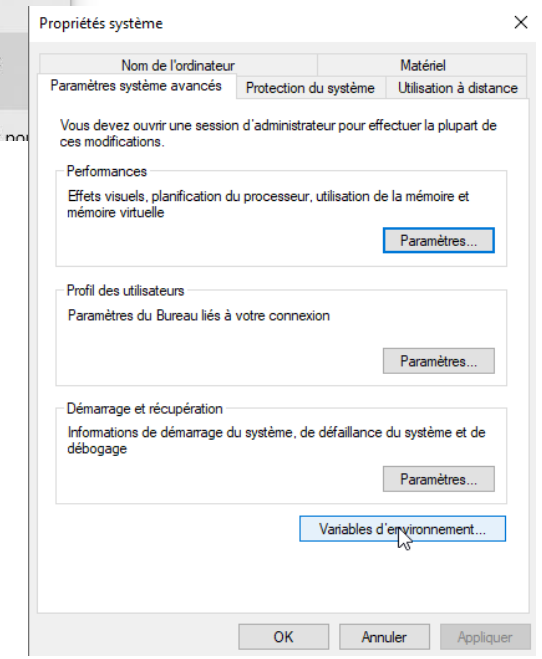
- Pour visualiser vos signaux de simulation (wave) vous devrez utiliser gtkWave (https://sourceforge.net/projects/gtkwave/ mais attention de bien aller chercher la version Windows gtkwave-3.3.98-bin-win64.zip).
- Ouvrir le fichier ZIP et extraire le répertoire gtkWave64 pour le mettre dans votre répertoire EDA.
- Pour partir l'outil, vous devez aller dans le répertoire bin/ et utiliser l'exécutable : gtkwave.exe (vous pouvez vous créer un raccourcit)

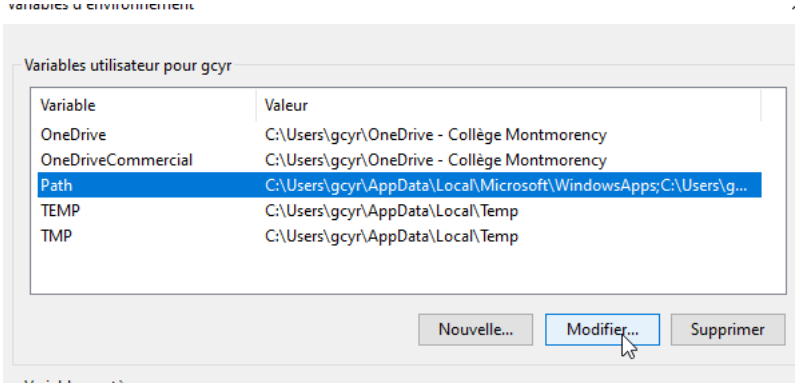
GHDL : Windows
Modifier la variable PATH

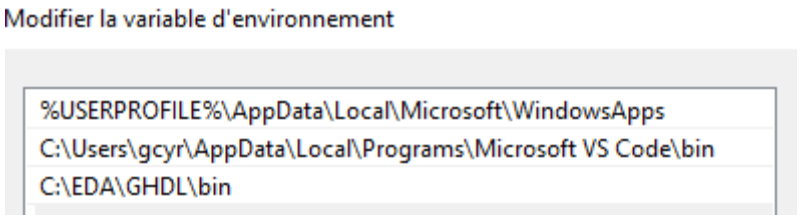
Pour faciliter l'utilisation des exécutables sur Windows (interface ligne de commande), il est préférable d'ajouter le chemin des exécutables dans la variable PATH de Windows.

- Dans le « settings » de l'ordinateur, chercher et choisir : « Modifier les variables d'environnement système »
- Puis choisir « variable d'environnement »
- Modifier la variable « Path »
- Ajouter le chemin pour GHDL



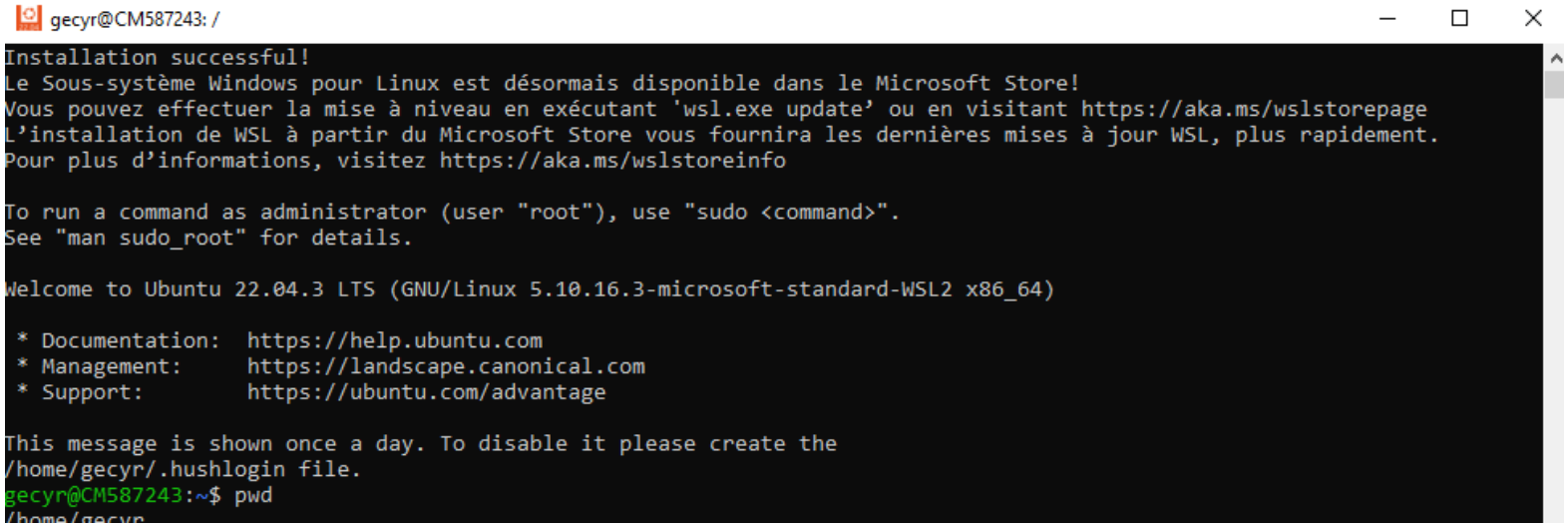






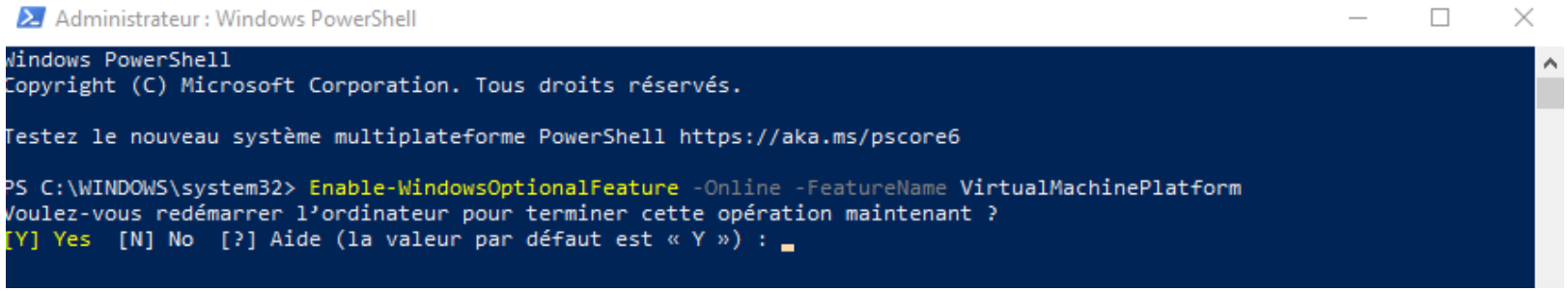
GHDL : Linux
Installation d'une distribution Linux sur Windows

- Pour installer une disctribution Linux sur Windows, vous pouvez utiliser WSL (Windows Subsystem for Linux) sur Windows. Pour l'installer, vous pouvez aller dans PowerShell en administrateur et taper : wsl.exe – install.
- Si cela ne fonctionne pas, suivez les étapes données à la page suivante pour le faire manuellement.

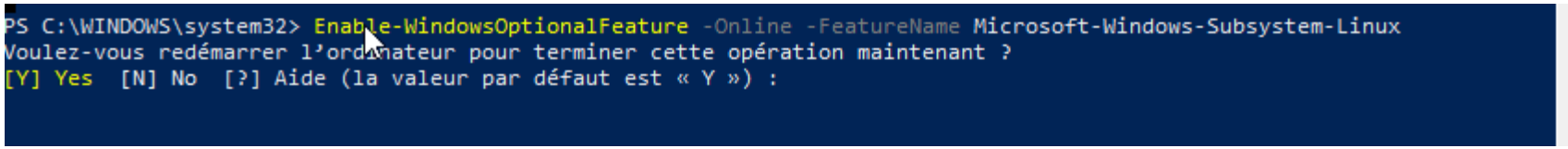


GHDL : Linux
Installer le WSL sur Windows manuellement

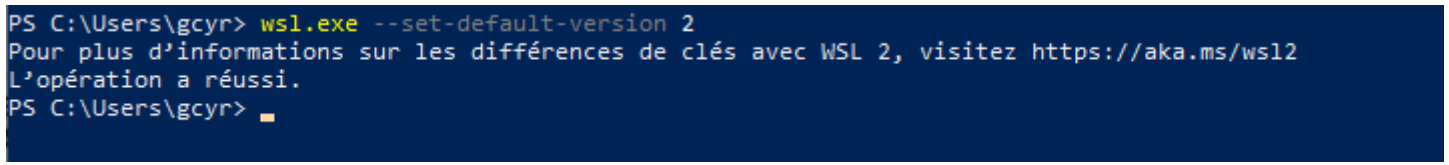
- Ouvrir PowerShell en mode administrateur puis faire la commande suivante et redémarrer l'ordinateur.

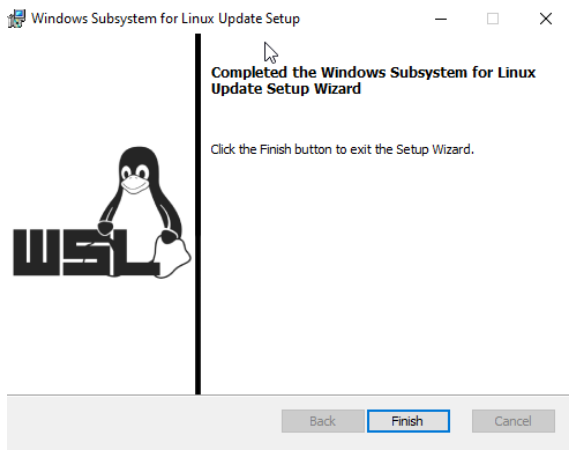


- Ouvrir PowerShell en mode administrateur puis faire la commande suivante et redémarrer l'ordinateur.



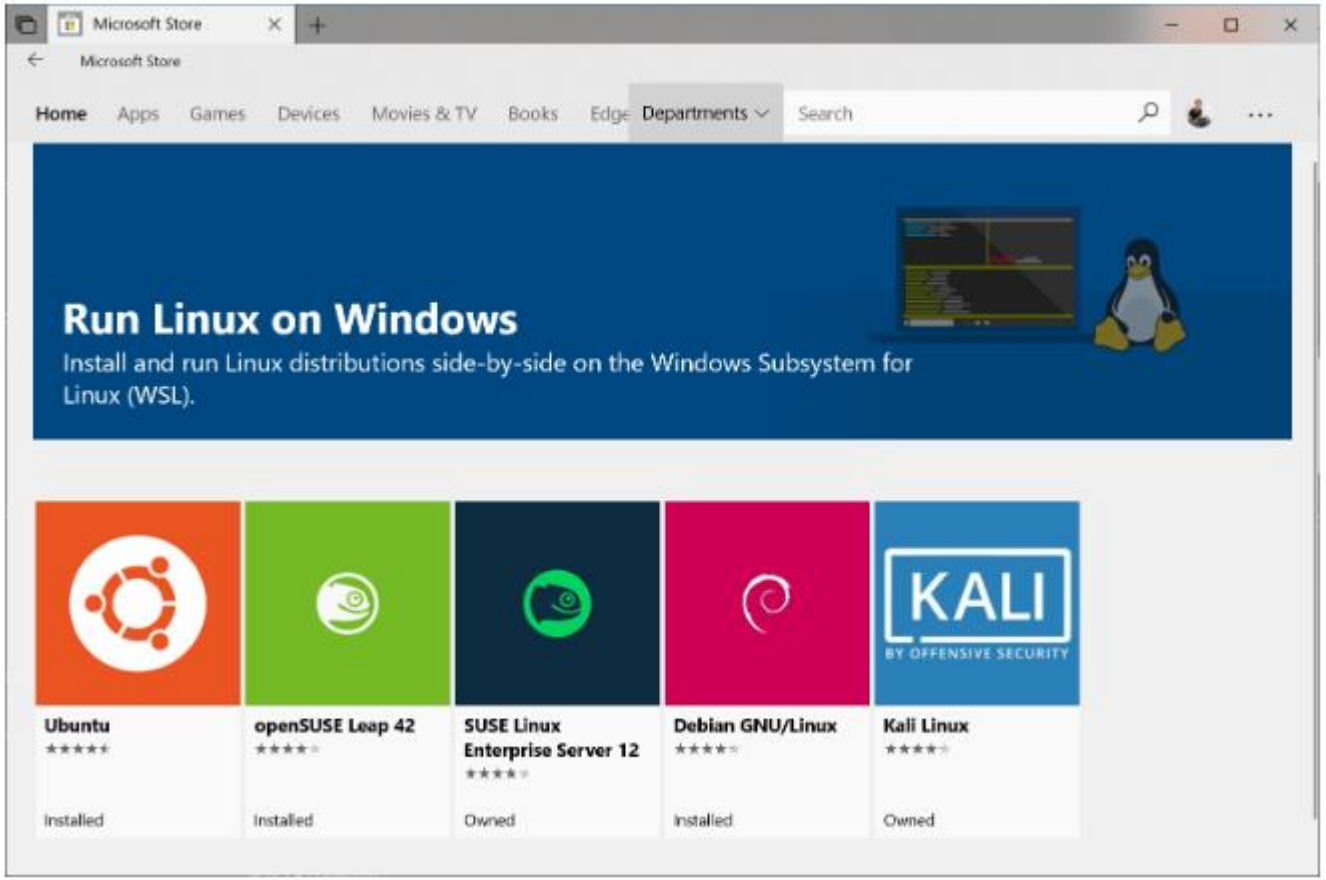
- Télécharger le package de mise à jour du noyau Linux : wsl_update_x64.msi
- Installer wls :





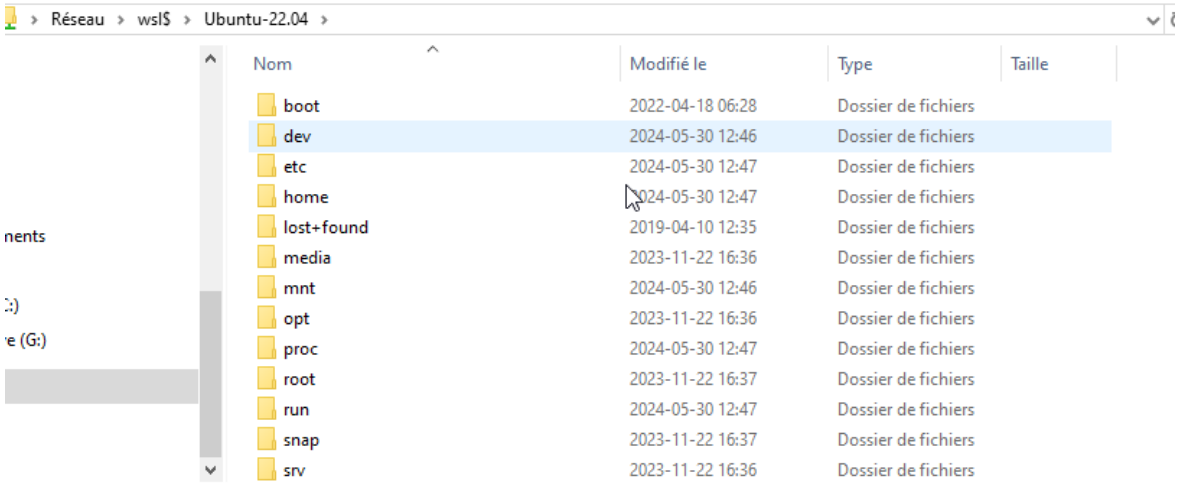
GHDL : Linux
Installer une distribution Linux pour Windows

- Choisir votre distribution Linux et l'installer (vous pouvez choisir Ubuntu 20.04)

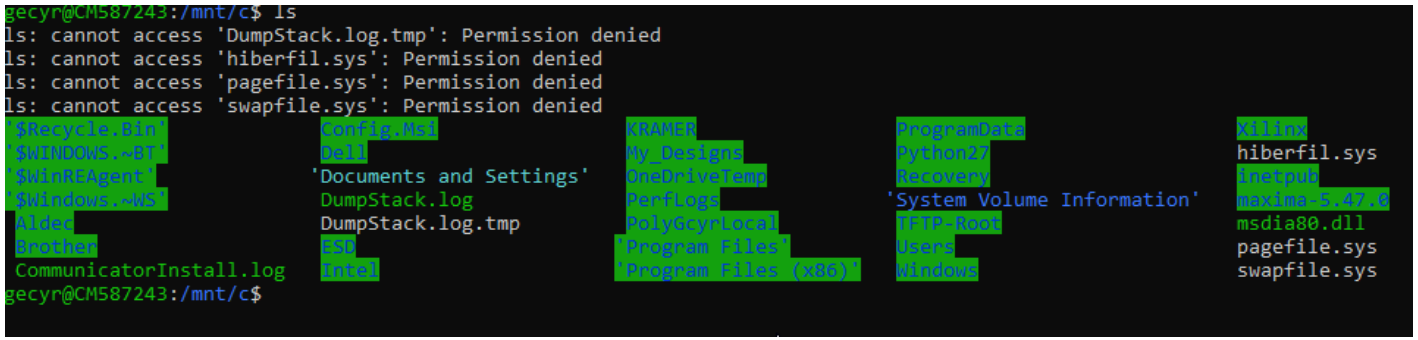


GHDL : Linux
Accéder aux fichiers de Linux à Windows et Windows à Linux

- Pour accéder aux fichiers Linux de l'explorateur Windows taper : \\wsl$ dans la barre de navigation



- Pour accéder aux répertoire Windows sour Linux taper : cd /mnt/c



GHDL : Linux
Installer les packages sur Linux

- Si vous avez installé Ubuntu 20.04, vous aurez besoin de certains packages qui ne sont pas disponibles. Sur le Shell Linux entrer les trois commandes suivantes :

```
sudo apt install make
sudo add-apt-repository universe
sudo apt update
sudo apt install csh
sudo apt install tclsh
sudo apt install tk8.6-dev
sudo apt install tcl8.5-dev
sudo apt install –y
sudo apt install –y  gpert
sudo apt install libbz2-dev
sudo apt install libgtk2.0-dev libglib2.0-dev
```

GHDL : Linux
Installation de GHDL et gtkWave

GHDL

- GHDL est nativement utilisé sur Linux, mais peut aussi être installé sur Windows, et utiliser par une interface ligne de commande.
- Vous pouvez télécharger et installer l'installateur pour Windows. La version la plus simple à installer sur WSL est la version « ghdl-installer-0.29.1.exe » sur le site : http://ghdl.free.fr/download.html. Ce n'est pas la plus récente (elle ne supporte pas le VHDL 2008), mais elle est facile à installer puisque c'est un exécutable Windows.
- Par la suite, vous devrez faire un Makefile pour compiler vos fichiers et l'exécuter sur votre Shell Linux.

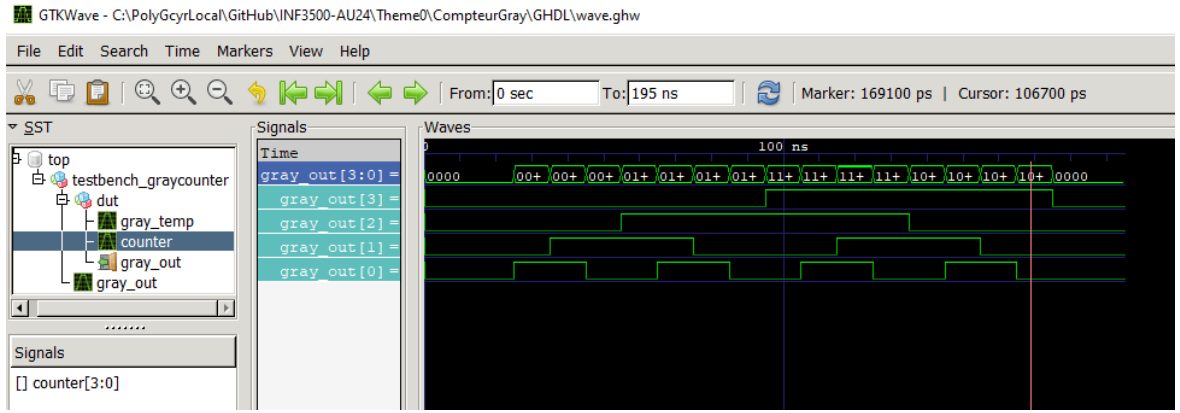
gtkWave

- Nous allons utiliser la même version que celle qui a été installée dans la partie GHDL : Windows, donc rien de plus à faire ici.

GHDL : Linux
Utilisation de GHDL et gtkWave

- Par la suite, vous devrez faire un Makefile pour compiler vos fichiers et l'exécuter sur votre Shell Linux. Voici un exemple pour mon compteur gray

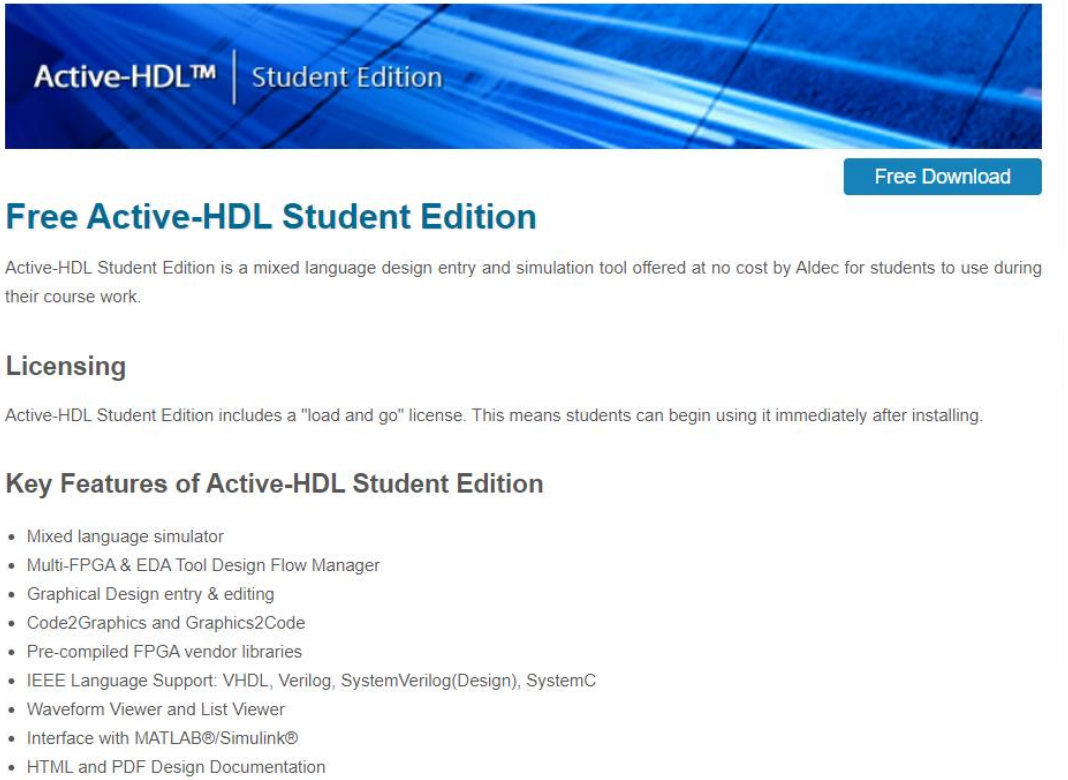
```
GHDL = ghdl.exe
FLAGS ="--std=93"

all:
	$(GHDL) -a $(FLAGS) GrayCounter.vhd Testbench_GrayCounter.vhd
	$(GHDL) -e $(FLAGS) Testbench_GrayCounter
	$(GHDL) -r $(FLAGS) Testbench_GrayCounter --wave=wave.ghw --stop-time=1us
```



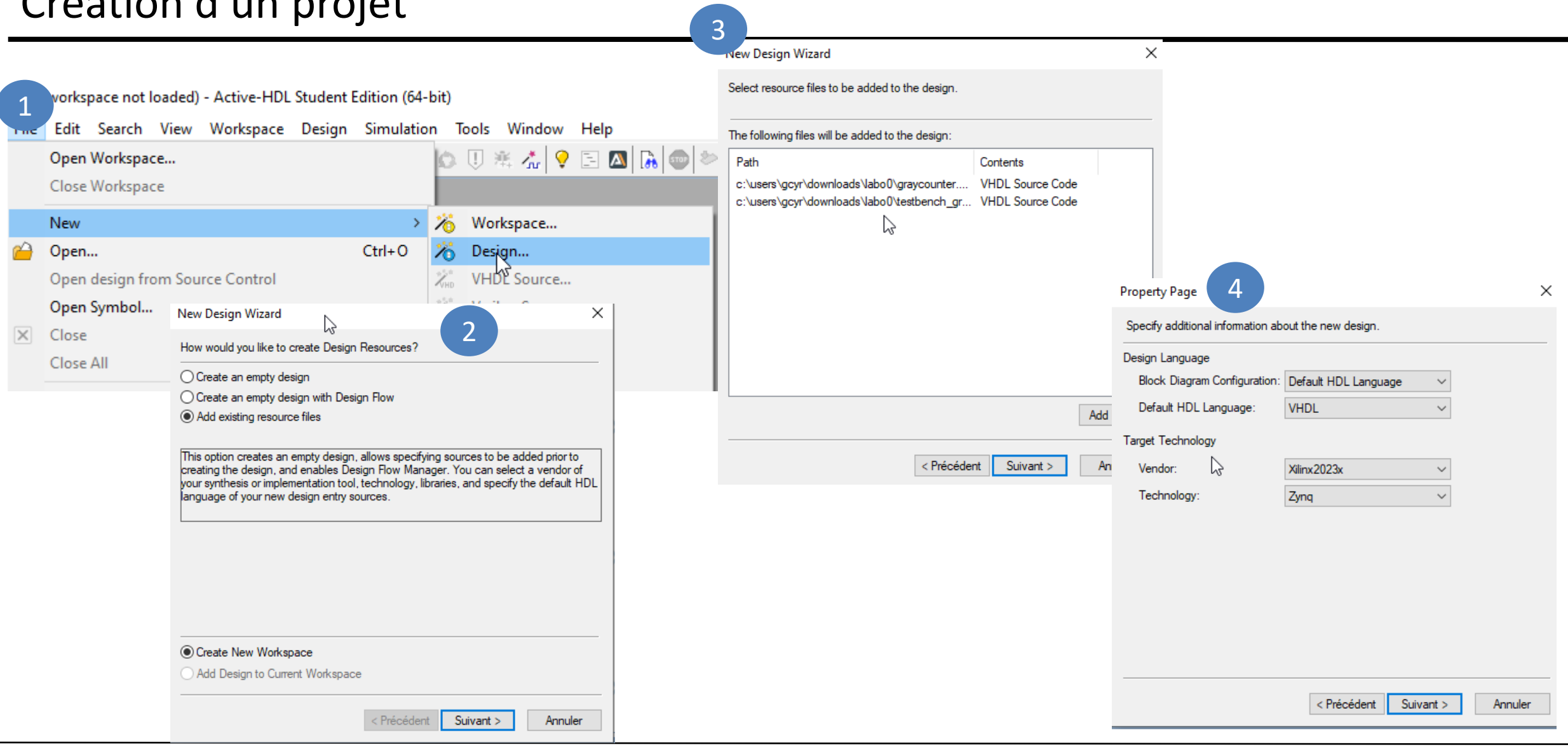
- En allant dans le répertoire de vos fichiers, vous n'avez qu'à faire : make
- Le script fait 3 étapes :
  - Compilation (avec le –a)
  - Élaboration (avec le –e)
  - Simulation (avec le –r)
- Par la suite, un fichier .ghw sera créé et vous pourrez l'ouvrir dans gtkWave pour voir le résultat de la simulation

Active-HDL de Aldec (Simulateur HDL)

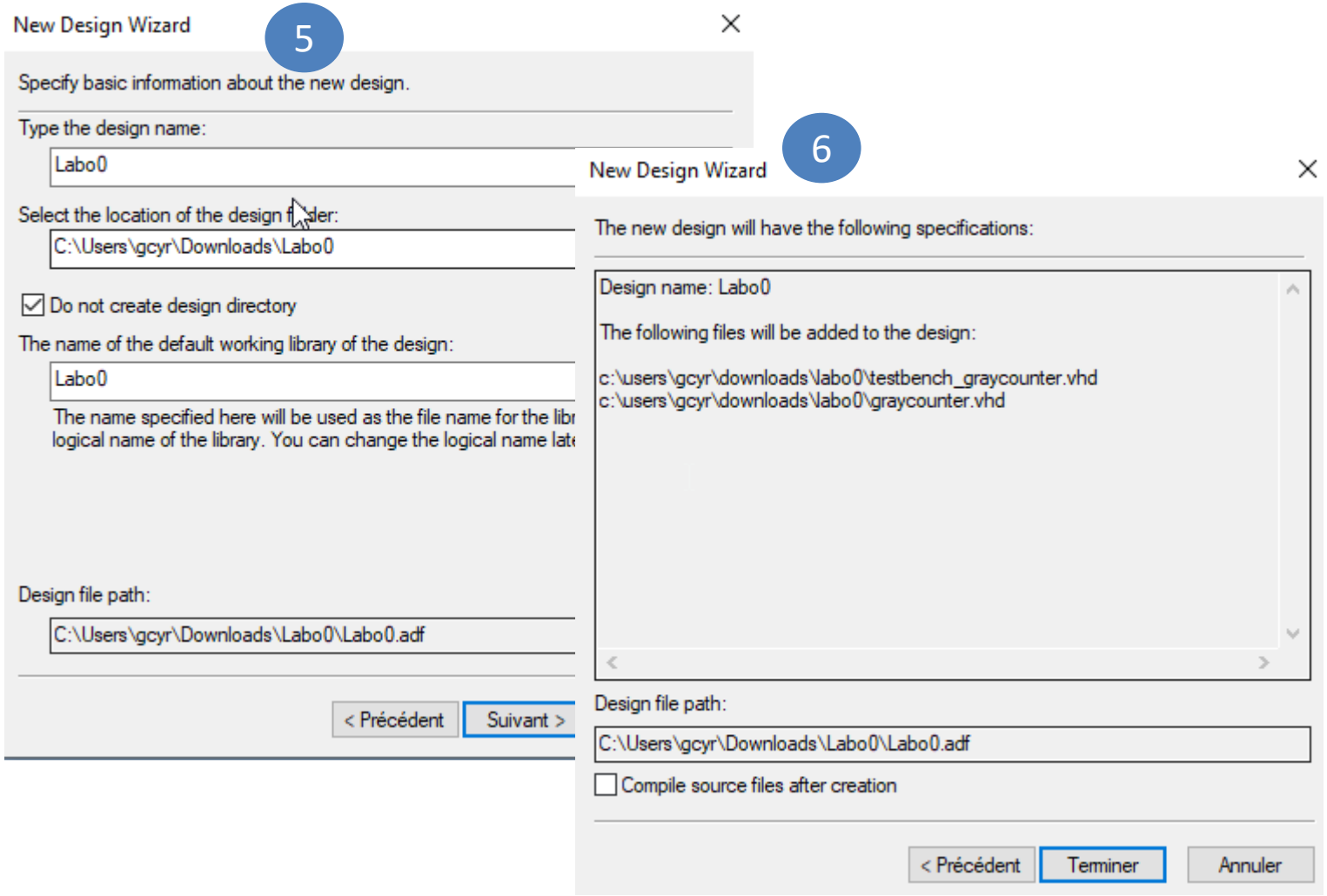
- Si vous décidez d'utiliser Active-HDL pour vos simulations, vous pouvez télécharger la version la plus récente étudiant et gratuite : https://www.aldec.com/en/products/fpga_simulation/active_hdl_student
- Pour savoir comment créer votre projet dans Active-HDL, vous pouvez regarder les étapes dans : https://www.aldec.com/en/support/resources/documentation/articles/1054

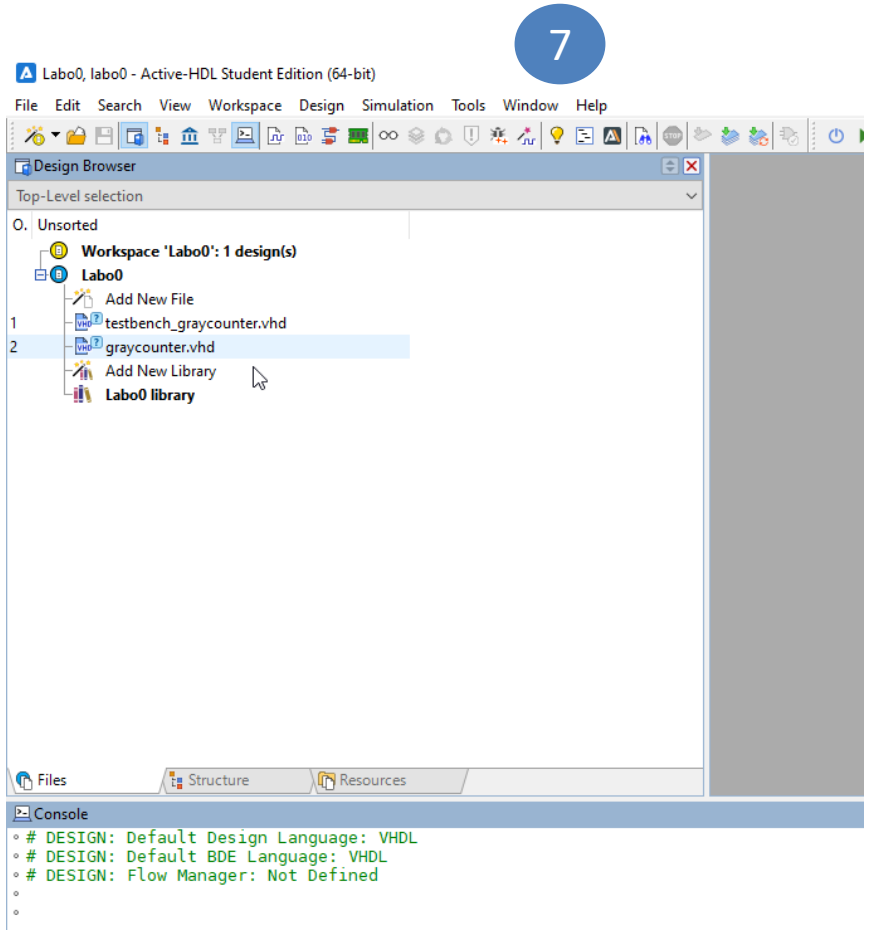


Active-HDL
Création d'un projet

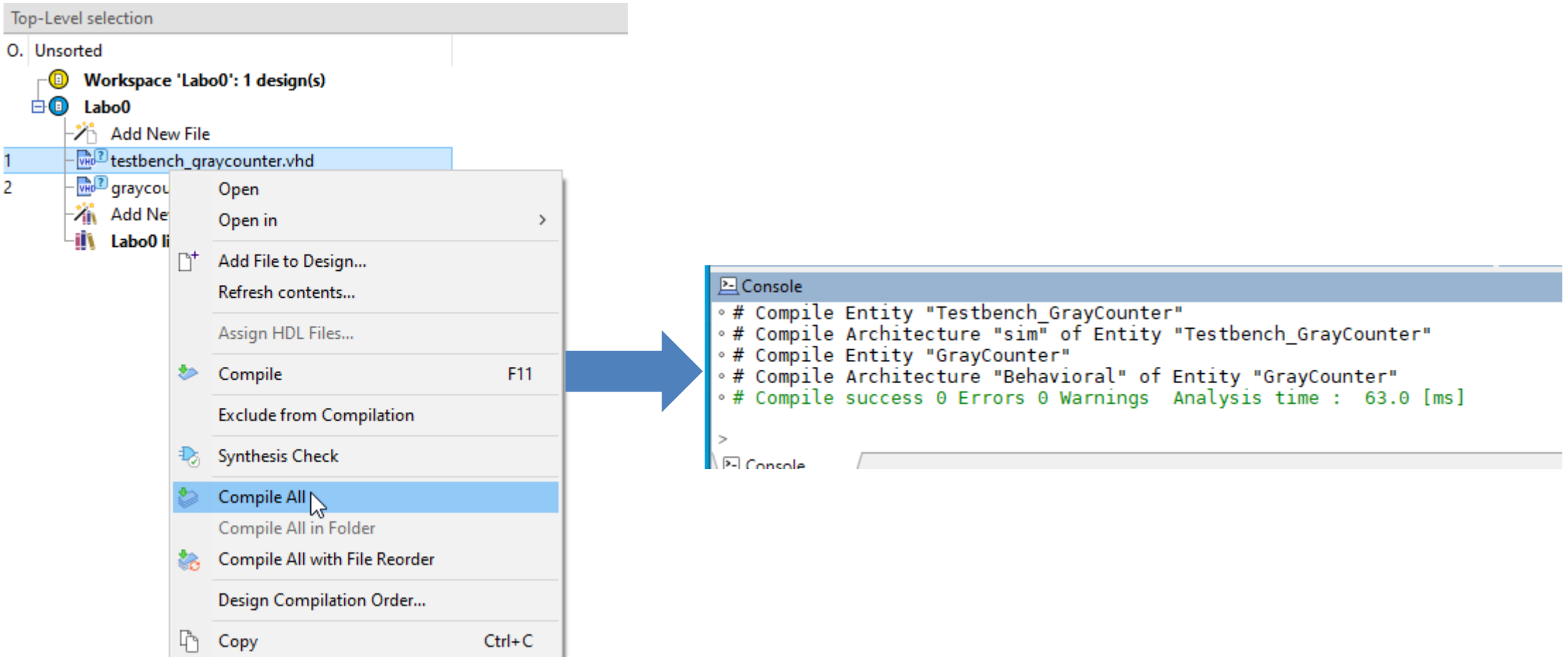


Active-HDL
Création d'un projet (suite)

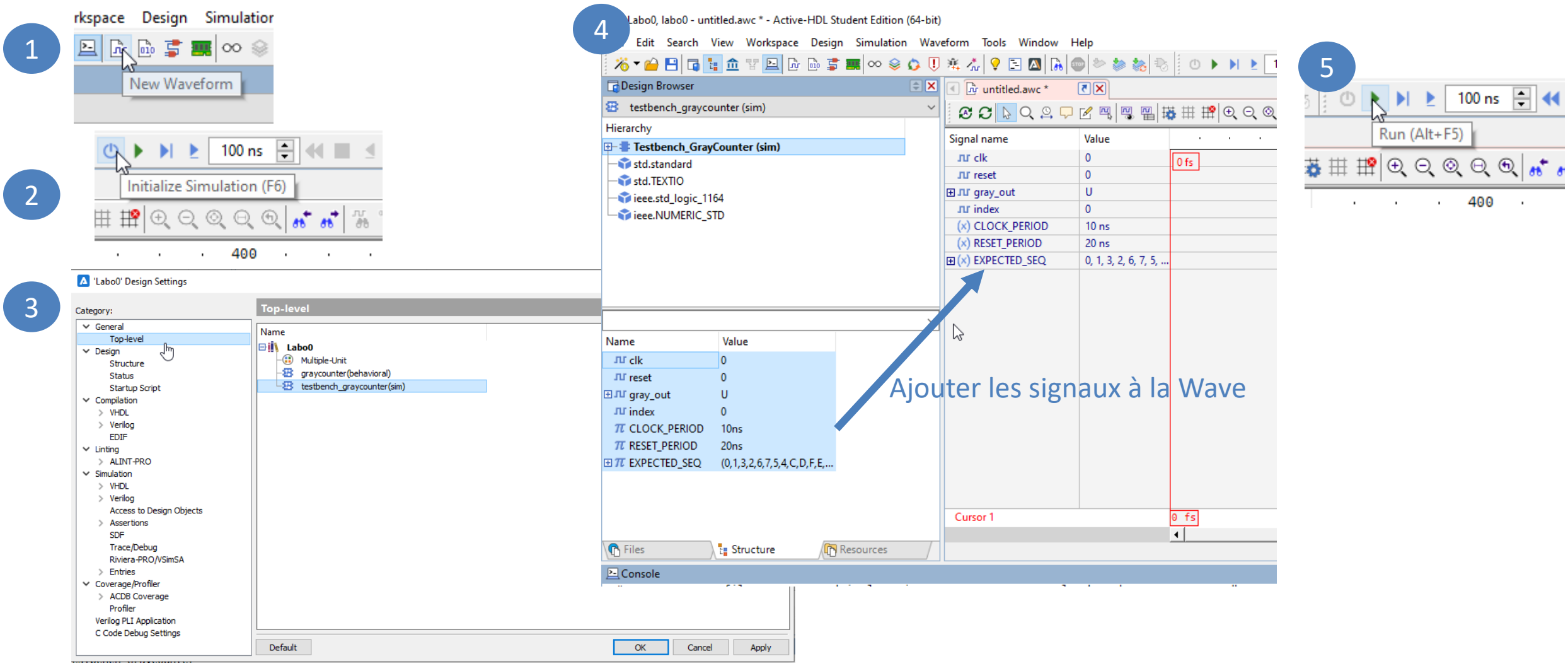




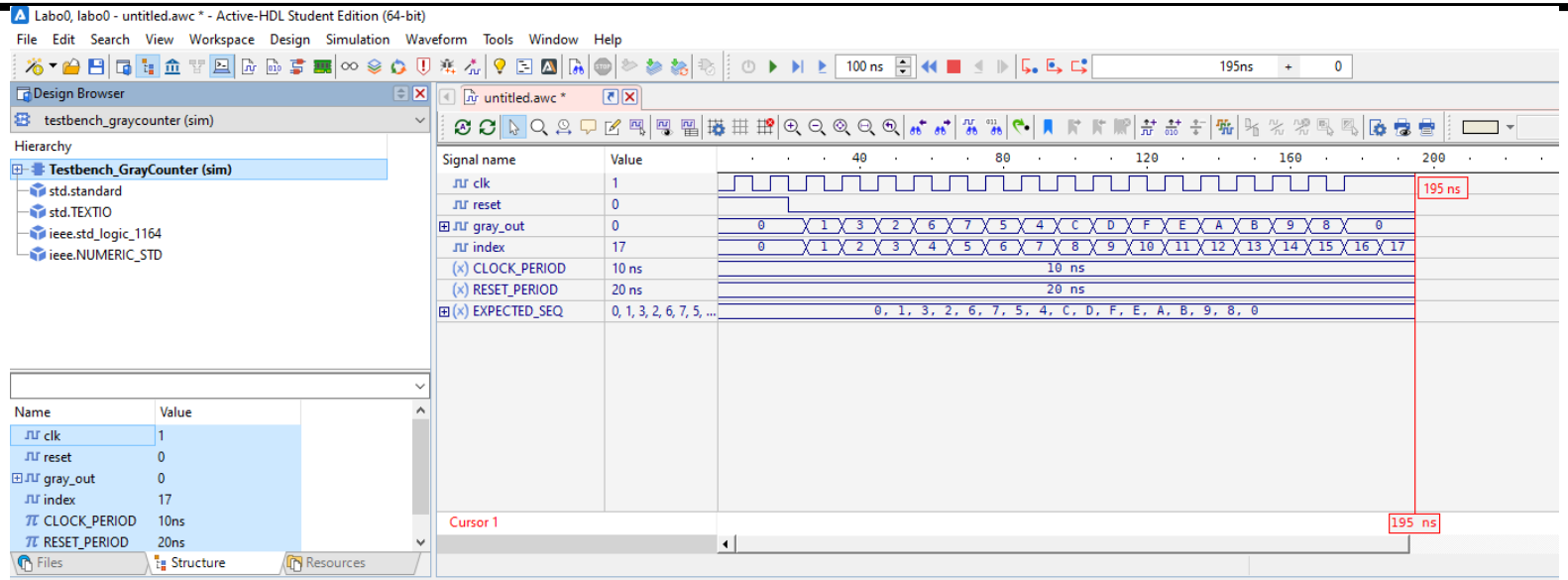
Active-HDL
Compilation



Active-HDL Simulation



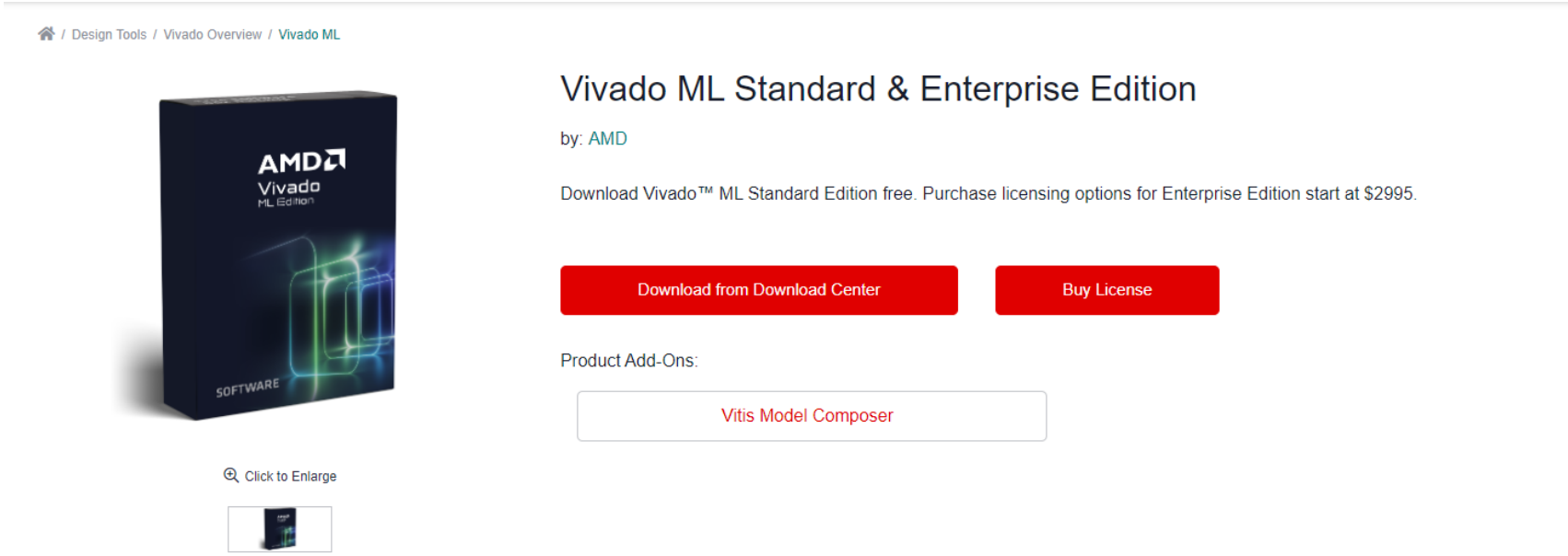
Active-HDL
Résultats de simulation




```
# Waveform file 'untitled.awc' connected to 'C:/Users/gcyr/Downloads/Labo0/src/wave.asdb'.
run
# EXECUTION:: NOTE   : Comparing value = 0
# EXECUTION:: Time: 25 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 1
# EXECUTION:: Time: 35 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 2
# EXECUTION:: Time: 45 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 3
# EXECUTION:: Time: 55 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 4
# EXECUTION:: Time: 65 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 5
# EXECUTION:: Time: 75 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 6
# EXECUTION:: Time: 85 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 7
# EXECUTION:: Time: 95 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 8
# EXECUTION:: Time: 105 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 9
# EXECUTION:: Time: 115 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 10
# EXECUTION:: Time: 125 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 11
# EXECUTION:: Time: 135 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 12
# EXECUTION:: Time: 145 ns,  Iteration: 0,  Instance: /Testbench_GrayCounter,  Process: stimulus_process.
# EXECUTION:: NOTE   : Comparing value = 13
```

Vivado (outil de développement Xilix)

- Pour utiliser et installer qu'un seul outil pour la compilation, simulation, synthèse et implémentation, on vous suggère d'utiliser Vivado pour simuler.
- Vous pouvez télécharger la version la plus récente (2024.1) et gratuite :https://www.xilinx.com/products/design-tools/vivado/vivado-buy.html

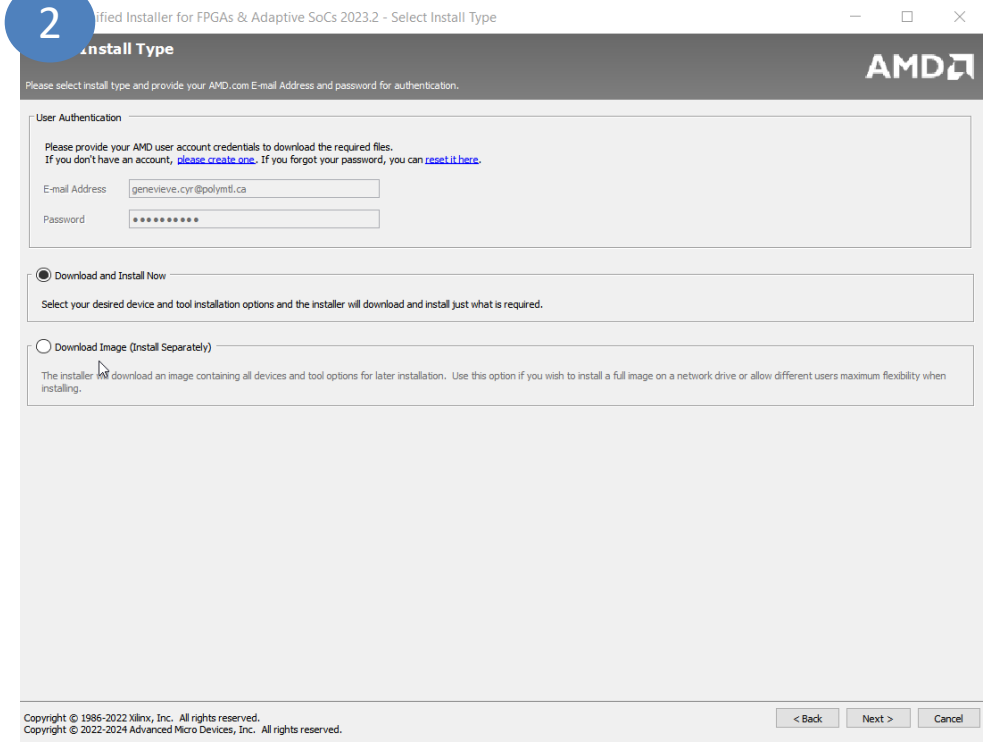


Vivado Installation

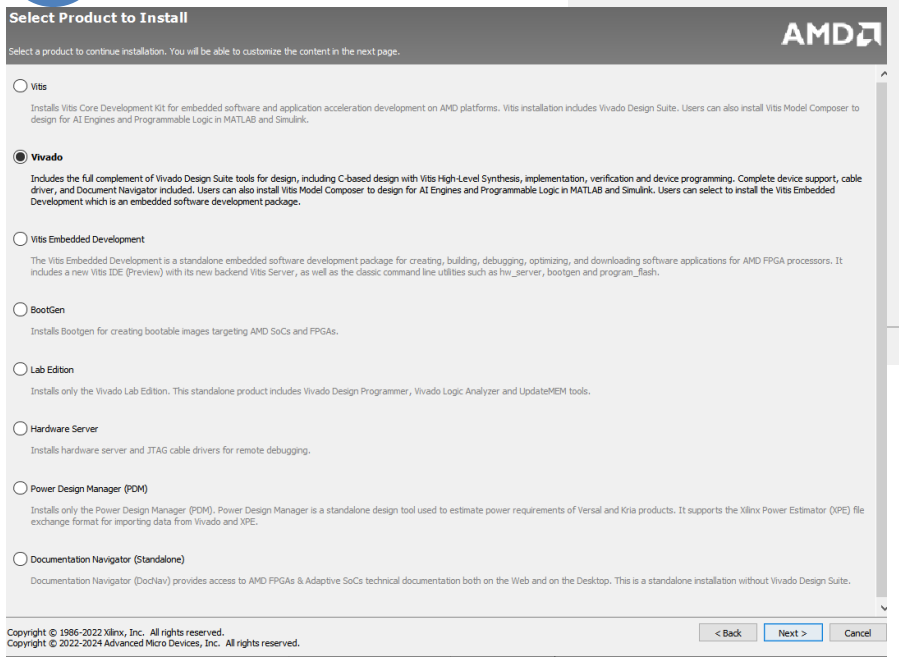
1

Vous devrez d'abord vous identifier avec un compte créer sur : https://sso-registration-qa.amd.com/ pour pouvoir vous identifier lors de l'installation par la suite.

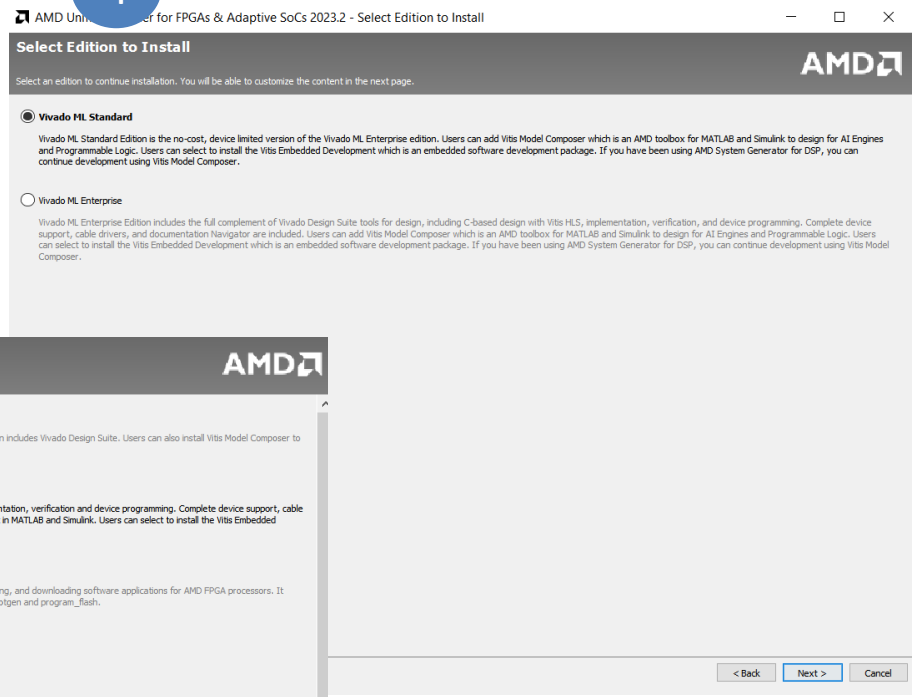
2



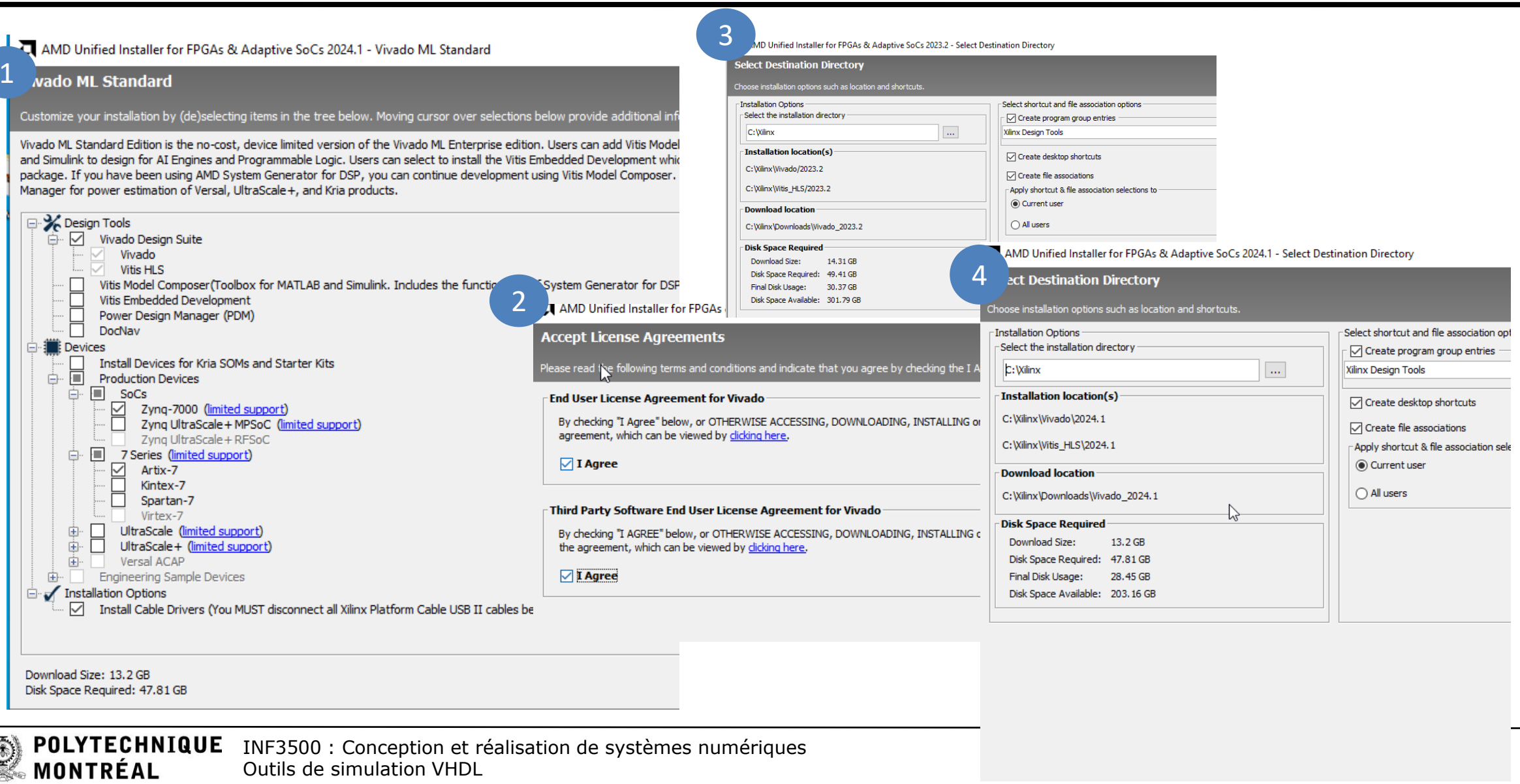
3



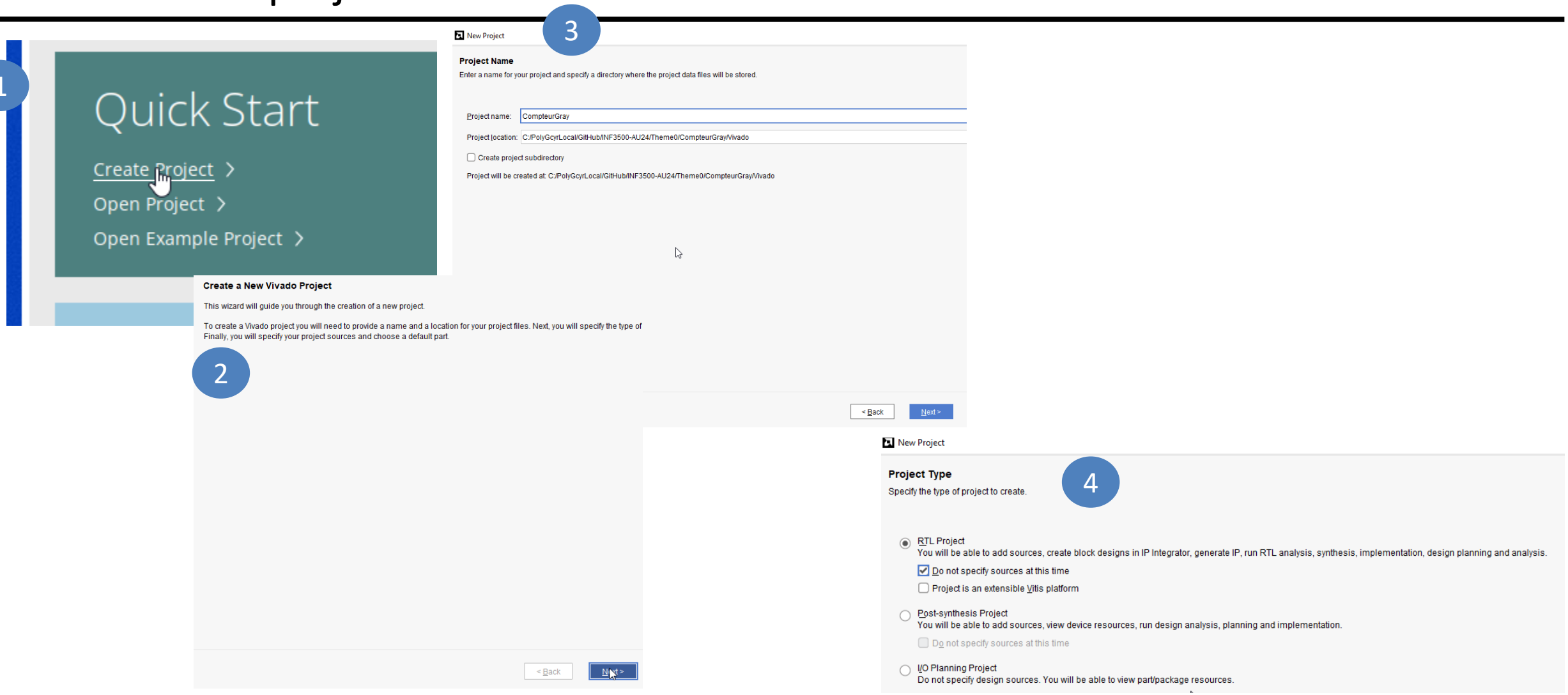
4



Vivado
Installation (suite)

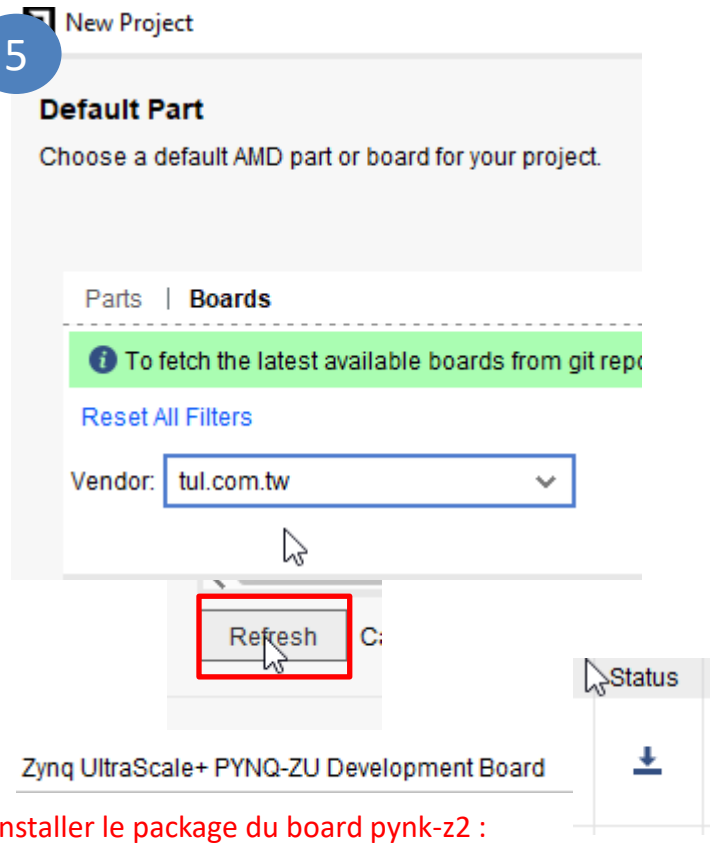


Vivado
Création d'un projet

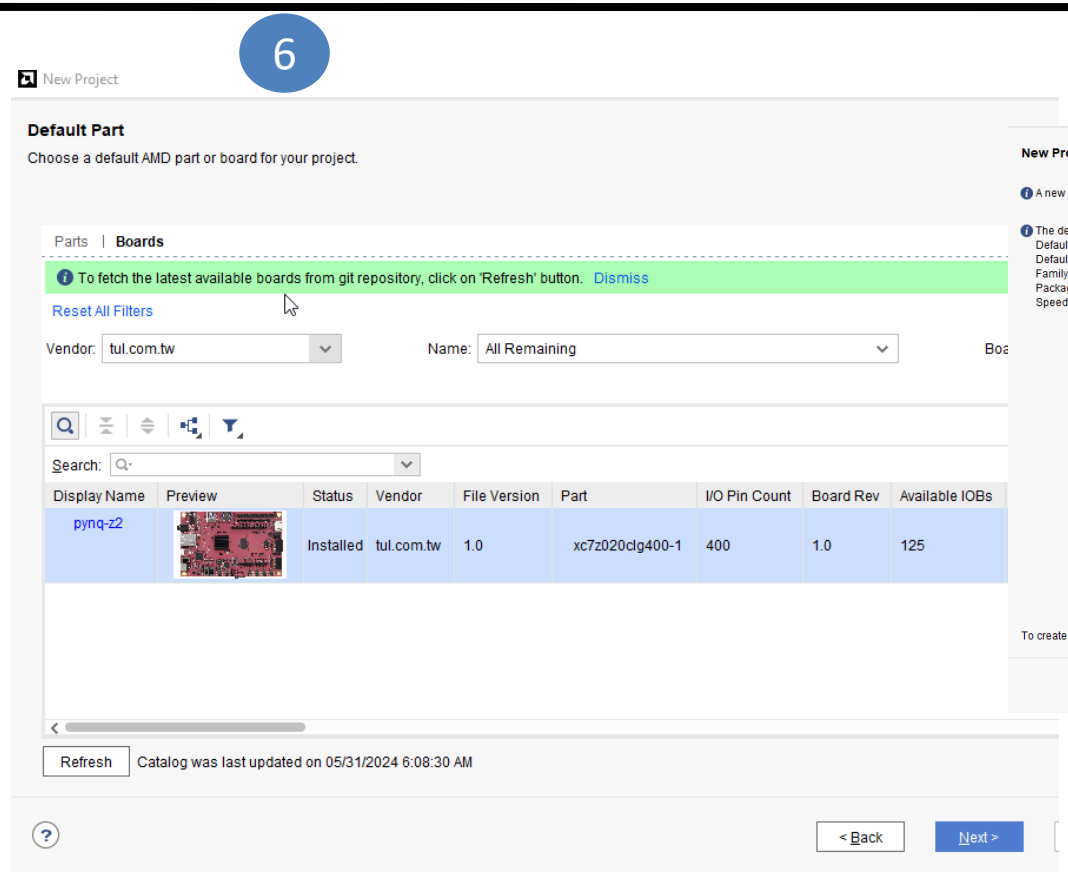


Vivado
Création d'un projet (suite)

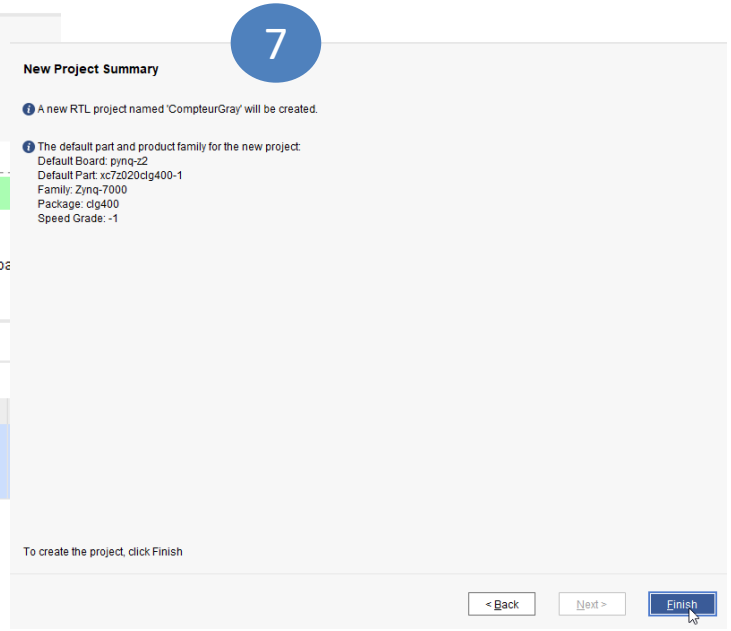
5



6



7



Installer le package du board pynk-z2 :

- Choisir tul.com.tw
- Appuyer sur refresh
- Un composant apparaîtra (Zynq UltraScale+PynqzU Development Board)
- Appuyer sur Téléchargement (flèche vers le bas dans status)

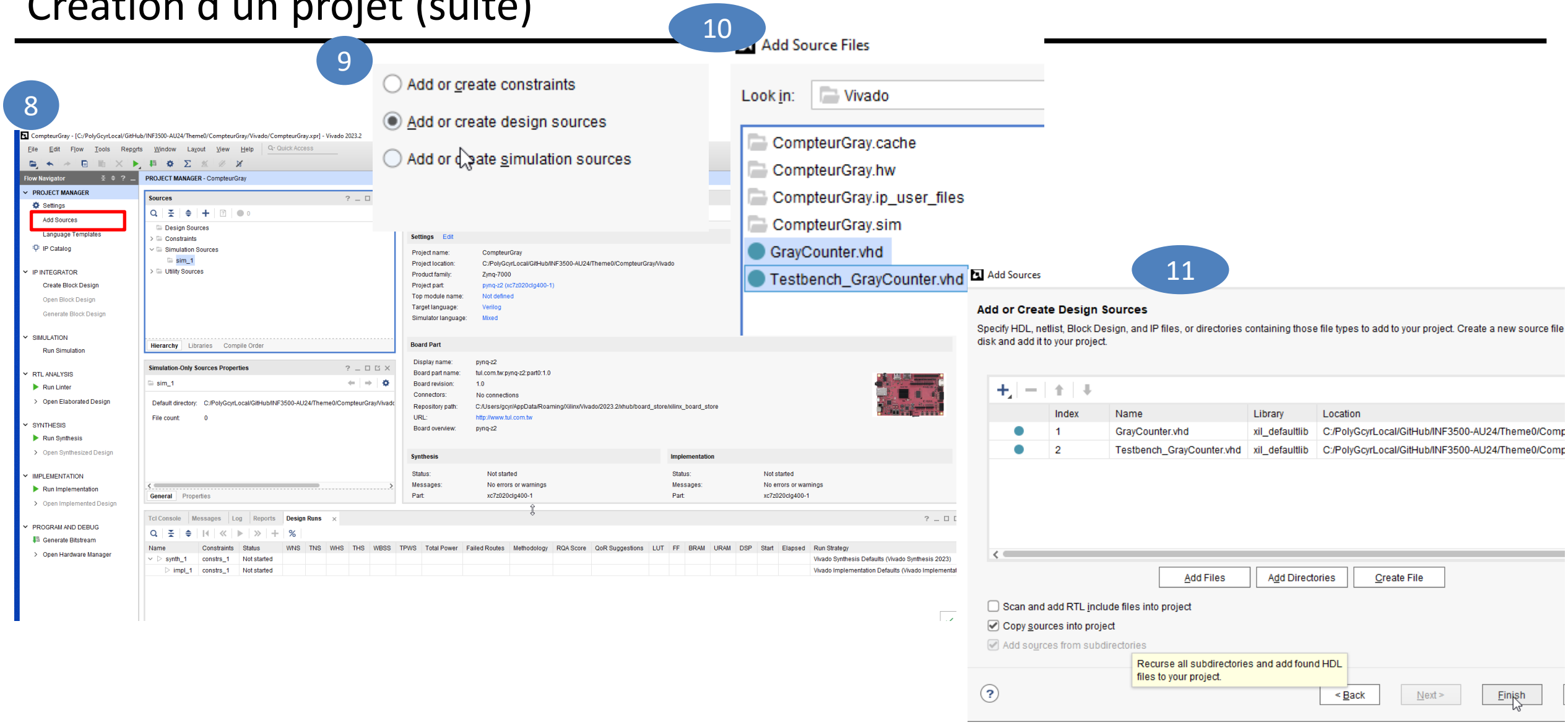
Vivado
Création d'un projet (suite)

8

9

10

11



Vivado
Simulation