

Simularea Verilog utilizând Modelsim

Opritoiu Flavius
flavius.opritoiu@cs.upt.ro

18 septembrie 2023

Simularea Verilog utilizând mediul modelsim

Obiective:

- ▶ Compilarea unui design Verilog folosind Modelsim
- ▶ Simularea unei soluții Verilog în mediul Modelsim

Suport de instalare Windows/Linux pentru modelsim: [▶ link](#).

Notă: Este necesar un cont Intel pentru descărcare.

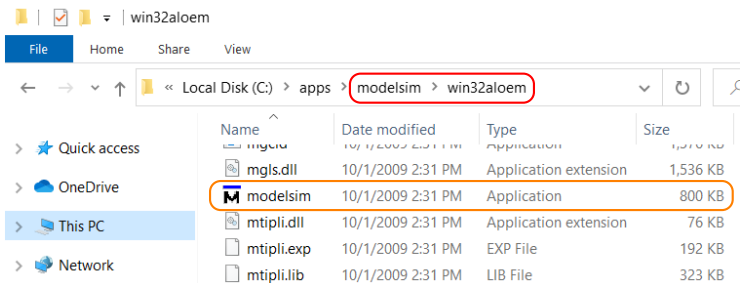
Acest ghid folosește versiunea "Modelsim Altera Starter Edition 6.5b". Aceasta ediție nu necesită o licență.

Notă: Mediul Modelsim include propriul editor de cod.

Instalare modelsim portabilă pentru Windows

O versiune modelsim portabilă rulând pe sisteme Microsoft Windows, poate fi descărcată de [▶ aici](#).

După extragerea arhivei, mediul Modelsim poate fi lansat rulând "modelsim\win32aloem\modelsim.exe".



Instalare modelsim tip Docker pentru macOS

O imagine Docker a mediului modelsim a fost construită pentru sisteme macOS. Procedura descrisă în slide-urile următoare a fost testată sub macOS Catalina, versiunea 10.15.7.

Pas 1: Instalare Docker Desktop for macOS, descrisă [▶ aici](#).

Pas 2: Încărcarea imaginii Docker Modelsim:

- i. Porneste Docker Desktop for macOS
- ii. Descarcă imaginea arhivată de [▶ aici](#)
- iii. Deschide aplicația Terminal și, considerând arhiva ca fiind salvată în ~/Downloads, rulează comenzile următoare:

```
cd ~/Downloads  
tar xJf ./msim_dking.tar.xz  
docker load -i ./modelsim_v6.5.tar
```
- iv. (opțional) Fișierele ./msim_dking.tar.xz și ./modelsim_v6.5.tar pot fi șterse

Instalare modelsim tip Docker pentru macOS (contin.)

Pas 3: Instalare și configurare XQuartz:

- i. Instalare XQuartz, descrisă [aici](#).

Notă: se acceptă solicitarea de log out de la finalul instalării

- ii. Pornește aplicația XQuartz. În noua fereastră (întitulată "xterm") accesează `XQuartz menu` `Preferences` și în tab-ul Security bifează opțiunea "Allow connections from network clients"
- iii. Repornește calculatorul

Pas 4: Descarcă script-ul `start_modelsim_macos.command`.

Notă: Script-ul poate fi salvat oriunde (ex. Desktop, pentru comoditate)

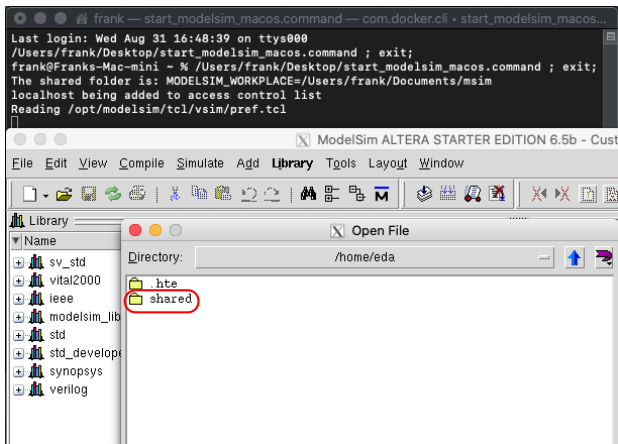
Pas 5: Configurarea directorului de partajat cu mediul Modelsim: deschide script-ul într-un editor text (ex. TextEdit) și, în linia 3, modifică variabila `MODELSIM_WORKPLACE` pentru a indica directorul de partajat.

Important: Directorul partajat este accesibil în mediul Modelsim environment prin calea de fișiere `/home/eda/shared`.

Instalare modelsim tip Docker pentru macOS (contin.)

Pentru lansarea mediului Modelsim în macOS:

- i. Rulează Docker Desktop for macOS
- ii. Rulează script-ul `start_modelsim_macos.command`
- iii. Conținutul partajt este accesibil prin meniul **File** **Open**, selectând folder-ul "shared"



Instalare modelsim portabilă pentru Ubuntu

O versiune portabilă a mediului Modelsim, care poate rula sub sistemul Ubuntu, poate fi descărcată de [aici](#). Acest mediu a fost testat sub Ubuntu 20.04 LTS și 22.04 LTS, versiuni de 64-bit.

Pentru sisteme Ubuntu de 64-bit, dependențele packetului Modelsim pot fi instalate prin comenzile următoare:

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install libc6:i386 libncurses5:i386
sudo apt-get install libx11-6:i386 libxtst6:i386
sudo apt-get install libstdc++6:i386
```

Comenzile de mai sus sunt incluse în fișierul "INSTALL" din folder-ul rădăcină al arhivei mediului Modelsim.

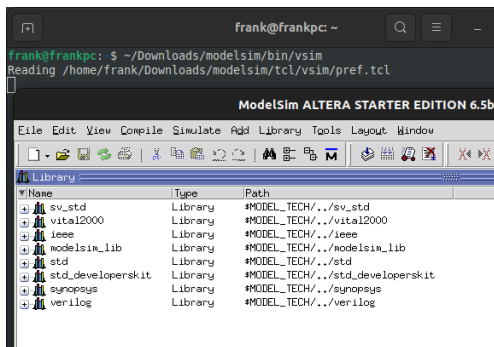
Instalare modelsim portabilă pentru Ubuntu (contin.)

Considerând arhiva salvată în `~/Downloads`, se dezarchivează prin:

```
cd ~/Downloads && tar xJf ./modelsim.tar.xz
```

Mediul Modelsim este lansat prin comanda:

```
~/Downloads/modelsim/bin/vsim
```



Este posibil ca această instalare portabilă să poată fi adaptată și altor distribuții Linux. În caz contrar, se poate utiliza instalarea tip Docker pentru Linux, prezentată în slide-urile următoare.

Instalare modelsim tip Docker pentru Linux

O imagine Docker a mediului `modelsim` a fost construită pentru a fi utilizată sub sisteme Linux. Procedura descrisă în slide-urile următoare a fost testată sub Ubuntu 22.04 LTS, 64-bit.

Pas 1: Instalare Docker Engine pentru distribuția utilizată: [▶ link](#).

Pas 2: Încărcarea imaginii Docker Modelsim:

- i. Descarcă imaginea arhivată de [▶ aici](#)
- ii. Deschide un terminal și, considerând arhiva ca fiind salvată în `~/Downloads`, rulează comenzile următoare:

```
cd ~/Downloads
tar xJf ./msim_dking.tar.xz
sudo docker load -i ./modelsim_v6.5.tar
```
- iii. (opțional) Fișierele `./msim_dking.tar.xz` și `./modelsim_v6.5.tar` pot fi șterse

Instalare modelsim tip Docker pentru Linux (contin.)

Pas 3: Descarcă script-ul `start_modelsim_linux.sh`. Considerând script-ul salvat în `~/Downloads`, va fi marcat ca executabil prin:

```
chmod +x ~/Downloads/start_modelsim_linux.sh
```

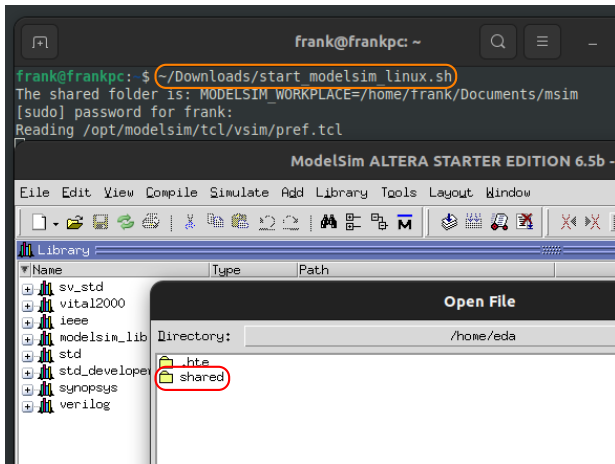
Pas 4: Configurarea directorului de partajat cu mediul Modelsim: deschide script-ul într-un editor text (ex. gedit) și, în linia 3, modifică variabila `MODELSIM_WORKPLACE` pentru a indica directorul de partajat.

Important: Directorul partajat este accesibil în mediul Modelsim prin calea de fișiere `/home/eda/shared`.

Instalare modelsim tip Docker pentru Linux (contin.)

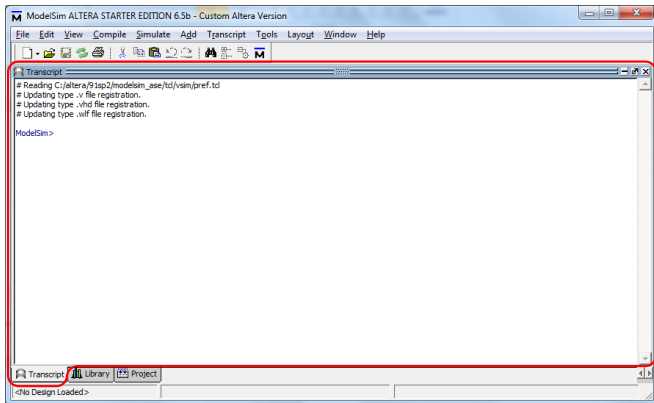
Pentru lansarea mediului Modelsim în Linux:

- i. Rulează script-ul `start_modelsim_linux.sh` într-un terminal
- ii. Conținutul partajat este accesibil prin meniul **File** > **Open**, selectând folder-ul "shared"



Mediul modelsim

Fereastra Transcript este marcată în imaginea de mai jos:



Toate comenzile sunt lansate din fereastra Transcript

Crearea unui proiect în modelsim

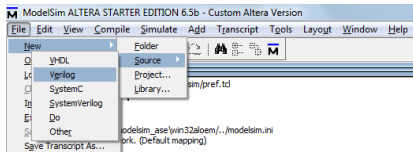
Se va selecta sau crea un folder pentru stocarea fișierelor proiectului. Acest exemplu folosește folderul "d:\student\msim".

Schimbați folderul curent al Modelsim rulând în fereastra Transcript comanda:

```
cd d:/student/msim
```

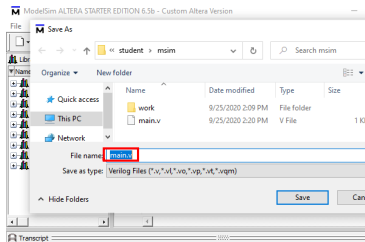
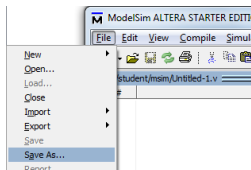
Scrierea codului Verilog în mediul modelsim

Meniul **File** **New** **Source** **Verilog** creează un nou fișier.



Se salvează fișierul din meniul **File** **Save As...** înainte de a adăuga cod și, pentru acest proiect, va fi numit *main.v*.

Important: Nu uitați să atașați extensia ".v" numelui *main.v*



Cod Verilog exemplu utilizând `modelsim`

Adăugați următorul cod Verilog în fișierul *main.v*

```
1 module main;  
2     initial  
3         $display(" Hello , _world!");  
4 endmodule
```

Pentru compilarea și simularea modului Verilog, se descarcă script-ul *run_helloworld.txt* de [aici](#).

Compilarea si simulare utilizând modelsim

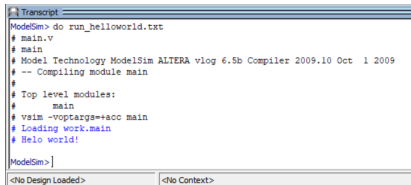
Se va executa următoarea comandă în fereastra Transcript.

```
do run_helloworld.txt
```

Nota: Fereastra Transcript oferă facilități de tip *shell* permițând accesarea comenzilor anterioare prin folosirea tastei ↑.

Script-ul *run_helloworld.txt* va compila fișierul sursă *main.v* ulterior lansând în execuție simularea modului *main*.

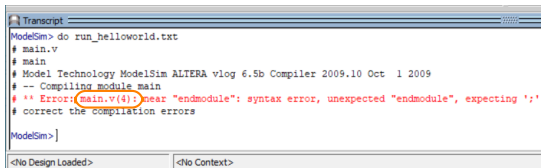
Dacă nu apare nicio eroare la compilare sau simulare, rezultatul din fereastra Transcript poate fi văzut în figura de mai jos:



```
Transcript
ModelSim> do run_helloworld.txt
# main.v
# main
# Model Technology ModelSim ALTERA vlog 6.5b Compiler 2009.10 Oct 1 2009
# -- Compiling module main
#
# Top level modules:
#   main
# vsim -voptargs==acc main
# Loading work.main
# Helo world!
ModelSim>
<No Design Loaded> | <No Context>
```


Erori de compilare utilizând modelsim

Figura de mai jos prezintă fereastra Transcript după semnalarea unei erori de compilare. Este evidențiată indicația numelui de fișier și a liniei conținând erorare



```
Transcript
ModelSim> do run_helloworld.txt
# main.v
# main
# Model Technology ModelSim ALTERA vlog 6.5b Compiler 2009.10 Oct 1 2009
# -- Compiling module main
# ** Error: main.v(4): near "endmodule": syntax error, unexpected "endmodule", expecting ';'
# correct the compilation errors

ModelSim> ]
<No Design Loaded> <No Context>
```

După corectarea erorilor de implementare, va fi rulată aceeași comandă în fereastra Transcript.