Comprehensive Guide for Installing and Troubleshooting ModelSim and Its Dependencies

Maryam Shahangian
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, Oklahoma
maryam.shahangian@okstate.edu

Hritom Das
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, Oklahoma
hritom.das @okstate.edu

Introduction

In modern digital systems and circuit analysis, software tools like ModelSim play a crucial role in ensuring the reliability and functionality of designs. ModelSim is widely used for simulating digital circuits, analyzing logical systems, and verifying hardware designs. Whether for educational purposes, research projects, or professional engineering tasks, having ModelSim configured properly is essential to avoid delays and technical setbacks. This guide brings together all the steps and solutions to address common issues faced during the installation and configuration of ModelSim. From setting up the software to resolving environmental variable dependencies and installing prerequisites such as Visual C++ Redistributable, this article covers everything you need to know to ensure a smooth and error-free experience with ModelSim. Additionally, we provide insights into using text editors like Visual Studio Code, terminal tools, and cross-platform solutions for Linux and Mac.

I. Siemens ModelSim Overview

ModelSim is a versatile software owned by Siemens, widely used for circuit simulation and verification. Initially developed by Mentor Graphics Corporation, it remains a powerful tool for graduate students and professionals alike.

- **ModelSim SE and Linux Links:** Provided on Canvas.
- **Mac Users:** Use Parallels Desktop for running ModelSim. Parallels Desktop Link

II. Installing Modelsim

The installation files for ModelSim are available at the following Dropbox link: ModelSim Installation Files

- **Executable File (.exe):** This file is sufficient for installation.
- **ZIP File (.zip):** Only needed for advanced features, likely unnecessary for most users.

III. Linux and Mac Setup for ModelSim

Window Users

- Open PowerShell or Command Prompt by searching for "cmd" or "powershell" in the Windows search bar.
- Use terminal commands to run simulations and interact with ModelSim.

Linux Users

Linux offers an excellent environment for running ModelSim, particularly for users accustomed to command-line workflows.

- **Terminal Access:** Use terminal commands for ModelSim operations. WSL (Windows Subsystem for Linux) is also supported on Windows systems. Learn more about WSL
- **License Server Access:** Ensure the environment variable LM_LICENSE_FILE is correctly configured.
- Linux terminals provide robust support for simulation tools. Ensure dependencies like Make and GCC are installed.

Mac Users

ModelSim does not natively support Mac OS. However, users can run ModelSim using virtualization tools like Parallels Desktop.

- **Parallels Desktop:** Install Parallels to create a Windows virtual machine. Parallels Desktop Link
- **License Configuration:** Follow the same steps to configure LM_LICENSE_FILE as in Windows or Linux.

IV. Importance of Environmental Variables

What Are Environmental Variables?

Environmental variables communicate specific information to the operating system to set up the working environment for software. They help the system manage file locations, user data, and software configurations.

Role of Environmental Variables in ModelSim

One critical environmental variable for running ModelSim is LM_LICENSE_FILE. This variable specifies the license location required by the software. Without proper configuration, ModelSim will not function.

Setting the LM_LICENSE_FILE Variable

To use the associated license for ModelSim, set the variable for LM_LICENSE_FILE to 1717@trelaina.ecen.okstate.edu

This tells the software to fetch the license from the appropriate server.

Common Issues and Solutions

1. **ModelSim Fails to Run:**
   - **Cause:** LM_LICENSE_FILE is not set correctly.
   - **Solution:** Verify and correctly set the variable value.
2. **Insufficient Error Information from ModelSim:**
   - **Cause:** An incorrect or missing variable may not trigger clear error messages.
   - **Solution:** Reconfigure the variable and ensure VPN connectivity for off-campus users.
3. **VPN Requirement:**
   - If you are off-campus, connect to the OSU VPN to access the license. VPN link: OSU VPN

V. Installing Visual C++ Redistributable

Why Is Visual C++ Redistributable Important?

Many applications developed using Visual Studio depend on **Visual C++ Redistributable** to run. This package provides essential libraries required by the software.

Common Issues Without This Package

1. **DLL Missing Errors:**
   - Messages such as "MSVCR120.dll is missing" or "MSVCP120.dll is missing" indicate the absence of this package.
2. **Application Crashes:**
   - Without this package, applications relying on Visual Studio libraries may crash during execution.

Download and Installation

Download the **Visual C++ Redistributable for Visual Studio 2013** from the official Microsoft website: Download Visual C++ Redistributable

Installation Steps

- Select the appropriate version for your system (x86 or x64).
- Download and run the installer.
- Restart the system if prompted and re-run ModelSim.

VI. Using Text Editors and Terminal Tools

Choosing the Right Text Editor

A good text editor is essential for writing code efficiently. While simple editors like Notepad may suffice for basic tasks, advanced editors like **Microsoft Visual Studio Code** offer a wide range of features for productivity.

- **Microsoft Visual Studio Code:**
  A lightweight but powerful editor supporting extensions for Verilog/VHDL and debugging. Download here: Visual Studio Code

Why Is Visual Studio Code Suitable?

- **Versatility:** Supports multiple languages like JavaScript, Python, C++, and Verilog.
- **Extensions:** Offers numerous extensions to enhance capabilities.
- **Free and Open Source:** Completely free and open-source.
- **User-Friendly:** Features a simple and intuitive user interface.

Other Recommended Editors

1. **Notepad++ (Windows):** Notepad++
2. **Emacs (All Platforms):** Emacs
3. **VIM/VI (All Platforms):** VIM
4. **Sublime Text (All Platforms):** Sublime Text
5. **Atom:** Atom

Conclusion

ModelSim stands as a cornerstone in the field of digital design and circuit simulation. Its versatility and reliability make it indispensable for professionals and students alike. However, its full potential can only be realized when installed and configured correctly. By addressing common challenges and providing detailed solutions, this guide empowers you to seamlessly integrate ModelSim into your workflow. From configuring critical environmental variables like `LM_LICENSE_FILE` to ensuring all dependencies, such as Visual C++ Redistributable, are in place, each step has been outlined to eliminate potential roadblocks. Additionally, leveraging tools like Visual Studio Code alongside the power of C++ can significantly enhance your productivity and deepen your understanding of system-level design. This guide is not just a solution to technical issues but an invitation to explore the broader capabilities of ModelSim. By mastering these foundational steps, you position yourself at the forefront of innovation in digital systems and hardware design. Remember, every successful simulation is a step closer to transforming ideas into reality. Let this guide be the starting point for your journey toward excellence in digital engineering.