Mograsim

Table of Contents

A Modular Graphical Simulator for Teaching Computer Architecture at All Levels

1. Quick Links

Directly install Mograsim in Eclipse by dragging this button into a running Eclipse workspace:

Drag to your running Eclipse* workspace. *Requires Eclipse Marketplace Client

- GitHub Repository
- Local Git Mirror
- Eclipse Marketplace
- Additional Documentation Material
- PDF-Version of this Documentation

2. About

Mograsim is a modular, graphical simulator for teaching microprogramming, ISAs and circuit logic in a way that allows for a smooth transition between those levels.

The aim is to give students a better understanding how these layers blend in a machine, and what the purpose of microprogramming is. In more detail, Mograsim allows:

- Programming and running simple assembler on the machine

this is still in development

- Defining an ISA (Instruction Set) to use in the assembler

this is still in development

- Microprogram the CPU to implement the specified ISA
- View the Circuit Logic operating in detail
- Executing the microprogram step by step with custom speed

Mograsim focuses especially on the AMD Am2900 Family microprocessors in a specific 16-bit arrangement used by the Technical University of Munich as an example.

Project and Motivation

Microprogramming is a difficult topic to teach, because it is usually hidden from the programmer but much more complex than simple binary logic. We are taking a look how teaching micro-programming can be improved by using a simulator that seamlessly visualizes the processes happening across all levels of computer architecture when a program is being executed. The result is an Eclipse plugin that can be used to simulate a microprogrammable machine, which is built from AMD's 2900 circuit family and used as an educational example at the Technical University of Munich. This is realized by a VHDL-`std_logic`-like sequential discrete event simulator and modular circuit models.

3. Getting started with Mograsim

3.1. Prerequisites

- Java 11 or later
- Eclipse 2019-03 or later

3.2. Installing Mograsim

First, you need start Eclipse.

1. Open the Eclipse Marketplace using "Help" → "Eclipse Marketplace..."
2. Search for "Mograsim"
3. Select Mograsim by the Mograsim Team and click "Install"
4. Read and accept the license, and click on "Finish".
5. Confirm that you want to install unsigned content.
6. Wait for the installation to complete (may take a while). When prompted, restart Eclipse.

Alternative: You can install Mograsim manually and set the update site yourself:

1. Go to "Help" → "Install New Software...".
2. Add the Mograsim update site:
   a. Click on "Add...".
   b. Next to "Location:", enter the update site address. This is http://mograsim.net/updatesite/ and http://mograsim.net/updatesite-unstable/ for the latest development versions.
   c. Click on "Add".
3. Tick "Mograsim".
4. Click on "Next >" two times, read and accept the license, and click on "Finish".
5. Confirm that you want to install unsigned content.
6. Wait for the installation to complete (may take a while). When prompted, restart Eclipse.

3.3. Enabling the Launch Action Set

Go to "Window" → "Perspective" → "Open Perspective" and select "Mograsim" or click "Other..." and then select Mograsim.
This can also be done by the perspective switcher on the right.

3.4. Creating a new Mograsim project

1. Create a new Mograsim project. Go to "File" → "New" → "Project...", select "Mograsim" → "Mograsim Project".

2. Give it a project name and a Mograsim machine and finish.

Alternative: You can create a general Project and add the Mograsim nature to the new project and set it up:

1. Open the properties dialog of the new project. (Right-click on it, select Properties.)
2. Go to the "Project Natures" page, click on "Add...". If a confirmation dialog pops up, confirm.
3. Select "Mograsim Project Nature"; click on "OK".
4. Click on "Apply and Close" and re-open the properties dialog.
5. Go to the new "Mograsim" page, select `Am2900Simple`, click on "Apply and Close".

3.5. Writing a Microprogram

1. Create a new file with the extension `.mpm`. (Right-click on the project → "New" → "File"; enter the filename; click on "Finish".)
2. The Mograsim instruction editor should open. If not, right-click on the MPM file → "Open With" → "Other..."; select "Instruction Editor"; click on "OK".
3. Write a microprogram.

The MPROM is currently hard-coded to be Opcode * 0x10.

Every cell differing from the default value is highlighted with a cursive font and green background.

3.6. Optional: Initial Main Memory

1. Create a new file with the extension `.mem` (as described above).
2. The Mograsim memory editor should open. If not, open it as described above.
3. Write the memory contents.

Each table row contains one (16 bit wide) memory cell. The two text fields labeled "Address" and "Number of cells" only refer to the cells displayed simultaneously in the editor. The editor internally retains all 65536 addressable cells.

3.7. Opening the Views

How to open the views "Simulation", "Debug", "Memory" and "Registers".

Go to "Window" → "Show View" → "Other...", select a view; click on "Open".

The Simulation view is in the category "Mograsim" and the other views in "Debug".

It is recommended to move the Simulation view to the Editor pane.

3.8. Creating a Machine Launch Configuration

1. Click on the little triangle next to the "Launch" symbol in the toolbar; click on "Run configurations…".
2. Right-click on "Mograsim machine" → "New Configuration".
3. Enter the Mograsim project containing both the MPM and MEM file, as well as these files.
4. If you don’t have a MEM file, leave the according field blank. This causes the memory to be initialized with 0.
5. Click "Run". The Simulation view now should contain a rectangle containing either the text "Am2900" in a very small font or a huge mess of smaller rectangles connected by colored lines.

The machine doesn’t start running yet since it starts paused.

3.9. An Introduction to All Views

One quick way to set up Mograsim and open all windows is by switching to the Mograsim perspective. This can be done in the upper right corner of Eclipse. If you have already done that: great, just read on.

Go to the window icon with a plus "Open Perspective" and select Mograsim. This will already present to you most of the views and editors that we will talk about in more detail in the following sections.

3.9.1. Simulation View

Move the view around by dragging with the left mouse button held. Zoom in and out by either scrolling up or down or by dragging down and up with the middle mouse button held.

Using the slider or by directly entering a number in the text field, the machine can be slowed down or sped up.

Also, a "step by step execution" mode can be enabled. Step by step execution means that the machine is automatically paused on each rising edge of the clock.

At the bottom of the Simulation view, a single instruction table row is displayed. This row contains the instruction currently being executed. The MPM can be modified by this line. This is not recommended, however, because changes done here are not reflected in the MPM file, and will be undone if the MPM file is hot-replaced (see below), even if the changed row didn’t change in the hot replace. Also, as for a hot replace, these changes doesn’t affect the currently active microinstruction.

3.9.2. Memory View

1. Select the preferred numerical representation
2. Right-click on the table → "Format…".
3. Select 16 units per row and one unit per column. (8 units per row if 16 don’t fit on the screen.)

The table now displays the contents of the currently running machine. At this moment, this should be equal to the contents of the initial memory specified when creating the launch configuration. This table updates automatically if the machine writes a memory cell. In this case, for a short time, this cell is highlighted using a red font and with a small delta symbol in the cell's corner.

3.9.3. Register View

Expand the register group you would like to view.

It should contain the registers R0-R15 as well as the Q register. All of them should be 0, displayed as a bit string

It is possible to change the register contents via this view by clicking on the old value and entering a new bitstring. It is not recommended to use `Z` in these bitstrings.

3.9.4. Debug View

Lists all currently running machines. The Simulation, Memory, and Register view follow the selection done here. A Mograsim machine launch contains one debug target, which contains one thread labeled "pseudo thread". The pseudo thread contains one stack frame labeled "pseudo stack frame" as long as the machine is paused. Unfortunately, it is required by Eclipse that threads don't have stack frames as long as they are running. For the Memory view to work, the pseudo thread or pseudo stack frame of a machine launch needs to be selected; for the Register view, the stack frame needs to be selected.

3.9.5. Toolbar

The toolbar also contains some relevant buttons: "Resume", "Suspend", and "Terminate". Suspend not only causes the clock to stop, but freezes the other components and wires. Because of this, some components can seem to be in an incorrect state, for example, the two inputs of a NAND gate could be 1, while the output also is 1.

3.10. Starting the Machine

Set the simulation speed to the lowest setting. In the toolbar, click the "Resume" button. Slowly increase the simulation speed until the wires in the Simulation view start flickering. The machine is now running and starts executing the microprogram.

3.11. Some Notes

1. If the MPM file used by a machine launch is opened in the Instruction editor, the row corresponding to the microinstruction currently being executed is highlighted there using a bold font and yellow background.
2. If the MPM file changes, after confirmation, the changes are hot-replaced into the machine. Changes in the MPM file don't affect the currently active microinstruction, however.
3. Some of the behavior described here can be changed in the Eclipse Preferences (in the pages "Mograsim" and "General" → "Appearance" → "Colors and Fonts"). For these cases the default

behavior is described.

Troubleshooting

If nothing Mograsim-related seems to be installed after installing Mograsim, make sure Eclipse uses Java 11 or later.

1. Go to "Help" → "About Eclipse IDE", click on "Installation Details", go to the tab "Configuration".
2. Search the line starting with "java.version=". (This line is in the "System properties" block, which usually occupies the first 200 lines.)
3. If this line specifies a version less than 11, configure your Eclipse installation to use a Java 11 JRE (or later). See Eclipse Wiki.

4. Customization

The following options are available to adjust Mograsim to your needs and personal preferences.

4.1. Colors and Visuals

For Colors in Views and Editors and their Fonts

1. open the Eclipse Preferences ("Window" → "Preferences")
2. and go to "General" → "Appearance" → "Colors and Fonts".

You will find a Mograsim Category there, with some sub-categories.

Some advanced settings for the visualization can also be found in the preferences under "Mograsim".

4.2. Controls and Debugging

Open the Mograsim Eclipse Preferences "Window" → "Preferences" → "Mograsim". In the first section, you will find settings to configure the navigation in the simulation view.

You can also enable the HLS *(High Level State)* Debug Shell, an expert option to view and set a machine's status in a separate window.

5. Simulation Model

The simulation model consists of components connected by wires.

Some components consist of other components and wires. The components contained in such a component are called the subcomponents of this component.

Some wires carry one bit, and some carry multiple bits.

Each bit carried by a wire is one of `1`, `0`, `U`, `Z`, or `X`. These values are IEEE 1164-compliant. (We don't use `H`, `L`, or `W`.) See IEEE_1164 on Wikipedia for an explanation of these values. Note: We also use X to denote illegal states. For example, when two components try to put different values on the D bus, the conflicting bits on the D bus will become `X`.

The simulation model can be viewed in the Simulation view: Each rectangle represents a component. Zooming in a component reveals its inner structure (if it has one). The colored lines connecting these rectangles represent wires. (Multiple-bit wires are displayed thicker than the single-bit wires, and always are black. For single-bit wires, the color represents their current value. These colors can be configured in the Eclipse preferences.

Table 1. Default Color Configuration

`std_logic`	**Color**
`0`	grey
`1`	green
`X`	red
`Z`	yellow
`U`	cyan