

CGRA-ME 2.0 Installation Instructions

Omar Ragheb, Stephen Wicklund, Jason Anderson

2024-03-26

This lab's goals are to allow participants to explore the open-source CGRA-ME 2.0 platform by walking through the processes of mapping applications onto an architecture, modifying an architecture, generating RTL for the architecture, and performing a functional RTL simulation.

This document will cover instructions on installing a docker container with the required software to follow along with the lab.

1 Docker Installation

1.1 Linux

Docker installation guidelines for Linux are available here: [linux-install](#).

In this manual, we run docker as a sudo user, but if you would like not to use sudo or do not have sudo privileges, then follow these instructions: [manage-docker-as-a-non-root-user](#)

1.2 MAC

Docker installation guidelines for MAC are available here: [MAC-install](#).

Note that if you encounter the following error "ERROR [internal] load metadata for docker.io", when installing the docker on MAC OS please consult the following link [MAC-OS-error](#)

1.3 Windows

If you are running docker on a Windows machine, we recommend installing Ubuntu WSL and following the Docker for Linux instructions.

Alternatively, Docker for Windows is available here: [windows-install](#)

Note: Some of the commands in this tutorial may differ if using this method.

2 Using Visual Studio Code

This installation manual follows through creating a docker using a WSL Ubuntu terminal with some exceptions for (MAC users). Alternatively, you could use the Dockerfile on Visual Studio Code by installing the Docker extension and following the instructions provided by the extension.

The lab will involve modifying and exploring files inside the container. If you are comfortable with vim, you may use that, but if you would prefer a GUI editor, consider installing Dev Containers and using Remote Explorer with Visual Studio Code to explore the container.

3 Creating the Docker Image

3.1 Setting up the Dockerfile

Download the Dockerfile to your host machine from: https://www.eecg.utoronto.ca/~raghebom/date_cgrame/

Then set up an empty directory, call it "CGRA_ME", and place the downloaded Dockerfile into the directory. Using these commands:

```
# mkdir CGRA_ME
# cd CGRA_ME
# curl https://www.eecg.utoronto.ca/~raghebom/date_cgrame/Dockerfile > Dockerfile
```

3.2 Building the docker

Using the Dockerfile, we build a docker with the following command:

```
# sudo docker build -t cgrame .
```

Please note that if you are running the Dockerfile on a MAC with Apple silicon (i.e. MacBook that was released on 2021 or later) please use this command instead:

```
# sudo docker build -t cgrame . --platform=linux/amd64
```

If the build process takes longer than 500s on step[13/13], it indicates that there is an issue with ModelSim installation on the docker. If this occurs, comment out line :62 within the downloaded Dockerfile and restart the build. This change will cause ModelSim to be downloaded but not installed.

3.3 Running the docker

To run the docker once it is done building run this command:

```
# sudo docker run -it -v /sys:/sys:ro -v $(pwd):/build cgrame bash
```

3.4 Installing CGRA-ME 2.0

Navigate to the location of the CGRA-ME 2.0 installation using:

```
# cd tmp/cgra_me/cgra-me-release-2.0.2/
```

To install CGRA-ME 2.0 on the docker, we need to run the the following two commands:

```
# ./cgrame_env
# make
```

3.5 Saving the docker state

Now that you have CGRA-ME 2.0 compiled. It is best to save the state of the docker, such that the next time you run the docker you will have CGRA-ME 2.0 compiled. Otherwise, if you exit the docker without saving the state you would need to recompile CGRA-ME 2.0. To save the state first open a second terminal and then type the following command:

```
# sudo docker ps
```

Notice the terminal output contains a tag called "CONTAINER ID" with the ID of the current running docker under it. To save the compiled state, type the following command in the terminal:

```
# sudo docker commit $(CONTAINER ID) cgrame
```