EENG 428 / ENAS 968
Cloud FPGA
Prof. Jakub Szefer

Setting-up GUI and Running Simulations on F1 Developer AMIs

Authors:
Shanquan Tian, Adam Wolnikowski, and Jakub Szefer
Yale University
New Haven, CT 06511, USA

Revised:
October 29, 2019

Contents

1 Configuring Development VM to Enable a GUI

This section explains how to set up and enable Graphical User Interface (GUI) on the development VM so that simulations can be run with a graphical waveform output. The instructions were adapted from `https://devopscube.com/how-to-setup-gui-for-amazon-ec2-rhel-7-instance/`.

This section assumes the reader has an "FPGA Developer AMI" (with CentOS Linux) running on a `c4.4xlarge` instance and that they can SSH into the instance. Also, *Public IP* of the instance will be needed to make the remote desktop connection, it can be obtained from *EC2 Dashboard* → *Instances* page under *Description* tab once the target instance is selected.

First, start the instance and SSH into the instance.

Next, install GUI components:

```
$ sudo yum -y update
$ sudo yum groupinstall -y "Server with GUI"
$ sudo systemctl set-default graphical.target
$ sudo systemctl default
```

Then, reboot the VM instance:

1. Open the Amazon EC2 console.
2. In the navigation pane, choose *Instances.*
3. Select the instance and choose *Actions* → *Instance State* → *Reboot.*
4. Choose *Yes, Reboot* when prompted for confirmation.

You can also reboot from the command line using the AWS CLI tools (in below command, make sure to input correct instance ID):

```
$ aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```

Then, set up XRDP, an open-source Remote Desktop Protocol server (note the first command should be all on one line, the back slash at end of line below is used if you type command over two lines in the terminal):

```
$ sudo rpm -Uvh https://li.nux.ro/download/nux/dextop/el7/x86_64/ \
nux-dextop-release-0-1.el7.nux.noarch.rpm
$ sudo yum install -y xrdp tigervnc-server
$ sudo chcon --type=bin_t /usr/sbin/xrdp
$ sudo chcon --type=bin_t /usr/sbin/xrdp-sesman
```

If there are errors with the chcon commands, alternative commands may work:

```
$sudo chcon -h system_u:object_r:bin_t:s0 /usr/sbin/xrdp
$sudo chcon -h system_u:object_r:bin_t:s0 /usr/sbin/xrdp-sesman
```

Next, start and enable the XRDP service:

```
$ sudo systemctl start xrdp
$ sudo systemctl enable xrdp
```

Then, update firewall configuration to allow connections on port `3389`, the default Remote Desktop Protocol port:

```
$ sudo firewall-cmd --permanent --add-port=3389/tcp
$ sudo firewall-cmd --reload
```

Note, this command should update the security group settings to open up RDP port `3389`. If there are connection problems, go to *EC2 Dasbhoard* → *Instances* → select your instance → *Description* tab → *Security Groups* to edit the security settings and add inbound TCP connections on port `3389` from `0.0.0.0/0`.

Next, set a password for the "centos" user (by default there is no password since usually one logs in with SSH and a public-private key pair):

```
$ sudo passwd centos
```

Finally, set password for root as you will be prompted for cloud user password for network proxy and color. To set this password, login as root and set the password:

```
$ sudo su
$ passwd
```

2 Connecting to Development VM via RDP

To make the remote desktop connection, you will need RDP software as well as set the password for the *centos* user.

2.1 RDP Software

For Windows and Mac OS computers, *Microsoft Remote Desktop* application works well. A free version should be available from the university. If you experience problems with the quality of the display images, you may want to try to change color quality to "medium" under "Display" settings.

For Linux machines, *Remmina* application works well on Ubuntu. If you have problem with *Remmina*, one common issue is that you may have to set the color profile to GFX RFX (32 bpp) for *Remmina* to work well. Another possible issue is the type of security protocol used, if you're not able to make a connection, go to advance settings, then security, and set the connection security to RDP.

2.2 Configure *centos* User Password

When you make a connection, use the Public DNS address shown on the *Connect To Your Instance* window in the *Instances* page of the *EC2 Dashboard*, or use the Pubic IP address, it can be obtained from *EC2 Dashboard* → *Instances* page under *Description* tab once the target instance is selected.

For the connections, use the *centos* user name, and the password you set in the above Section 1.

3 Simulating the Design: Hello World Example Test

First, connect to the `c4.4xlarge` VM instance using remote desktop.

Once using GUI to view the desktop of the developer VM, open terminal and setup the HDK and SDK environments (same as is done when using SSH):

```
$ cd aws-fpga/
$ source hdk_setup.sh
$ source sdk_setup.sh
```

Now export `CL_DIR` variable to specify the top level directory of the Hello World example:

```
$ cd aws-fpga/
$ export HDK_DIR=$(pwd)/hdk
$ cd $HDK_DIR/cl/examples/cl_hello_world
$ export CL_DIR=$(pwd)
```

Next you can compile the simulation:

```
$ cd $CL_DIR/verif/scripts
$ make TEST=test_hello_world
```

The output of `make` will take a few minutes. Eventually in the terminal you should see output similar to:

```
Writing 0xDEAD_BEEF to address 0x00000500
Reading 0xefbeadde from address 0x00000500
TEST PASSED
```

This indicates the simulation has finished and the tests have passed.

3.1 Generating Waveforms of the Hello World Example Test

When running the simulation, by default, not all the signals are saved in the waveform database (later used by the Vivado waveform viewer to view the signals). To ensure all signals' transitions are saved into the database the `waves.tcl` script needs to be edited.

Open the `$CL_DIR/verif/scripts/waves.tcl` file. In the file, change `add_wave /` to `add_wave -recursive /`. Now re-run the simulation to save all the waveform signals. This change will cause the simulation to run longer and waveform database to grow (but should not exceed 100MB).

3.2 Using Waveform Viewer to View Waveforms of Hello World Test

Because the Amazon F1 designs use proprietary (Xilinx) IP modules for PCIe, DRAM, etc. the designs can not be simulated easily with open-source tools such as `iverilog` and viewed with `gtkwave`. Instead, to view the simulation waveform in detail, you need to do with Xilinx Vivado tools, using the `vivado` command from the terminal.

First, for `vivado`, a configuration file is needed to show the waveform. Go to the simulation directory:

```
$ cd $CL_DIR/verif/
$ cd sim/vivado/test_hello_world/
```

Then create a simple `open_waves.tcl` script in the directory. Here, the `echo` command is used to write contents to a file, a single `>` is used to create a new file and write a line of text to it (delete any prior file contents if they existed), and `>>` is used to append a line to the file. Thus, below commands are used to create a file with two lines of text which are the TCL commands needed by `vivado` to show the waveforms.

```
$ echo 'current_fileset' > open_waves.tcl
$ echo 'open_wave_database tb.wdb' >> open_waves.tcl
```

Now start Xilinx Vivado tool to view the waveforms (use the `&` at the end of the command line so that you can keep using the terminal with the Vivado GUI is opened).

```
$ vivado -source open_waves.tcl &
```

In the waveform GUI you can now add signals and see the waveforms as a function of the simulated time, and use them to debug the deign.

4 Acknowledgement

This tutorial document was based on information available from Amazon in their documentation, and various online tutorials referenced in the the text. Most of the information was taken from `https://github.com/aws/aws-fpga/blob/master/hdk/docs/RTL_Simulating_CL_Designs.md`.