Modifying the Rocket Core on the Zedboard

Ashton Johnson & Ian Swepston

April 30, 2017

1 Purpose

This document describes how to modify the RISC-V Rocket core that can be deployed to a Zedboard. For information about the rocket chip, see Rocket-Chip on Github.

2 Assumptions

This document assumes the following items are installed on your computer.

- Java JDK
- Scala Build Tool (SBT) (available via apt)
- Xilinx Vivado 2016.2
- fpga-zynq repository

3 Prerequisites

- Download the fpga-zynq repository by performing the following from the command line:

```
git clone https://github.com/ucb-bar/fpga-zynq
cd fpga-zynq
export TOP=$(pwd)
```

- Initialize the sub-repositories:

```
make init-submodules
```

4 Generate default core (optional)

Generating the default configuration is simple. If everything as been configured correctly, we only require two steps:

- Move to the target board directory

```
cd $TOP/zedboard
```

- Generate the Vivado project

```
make rocket
```

- This process will take less than 10 minutes. At the end of this, you should see a new folder in the folder named `zedboard_rockethchip_ZyncConfig`.
- Generate the bitstream:

```
make project
```

- Generate the `boot.bin` file:

```
make fpga-image-zedboard/boot.bin
```

- You can upload the image to the SD card:

```
make load-sd SD=path_to_mounted_sdcard
```

5 Modifying the core

+In the next section we will modify the core by adding and referencing new configurations. Creating new configurations will allow us to keep the existing ones.

5.1 Adding L2 cache

L2 caches are not utilized by default. Modification requires 3 steps:

1. Creating a new rocket-chip configuration
2. Creating a new fpga-zynq configuration. This will reference the new rocket-configuration.
3. Generating the RTL for the new fpga-zynq configuration.

5.1.1 Rocket-chip Configuration

- We begin by creating a new rocket-chip configuration. Open up the following file using your favorite text editor. We use emacs here:

```
emacs $TOP/rocket-chip/src/main/scala/rocketchip/Configs.scala
```

- Locate the following line:

```
class DefaultFPGAConfig extends Config(new FPGAConfig ++ new BaseConfig)
```

- This is the default configuration for FPGAs. We will create a new config that builds upon this. Add the following:

```
class DefaultFPGAConfigL2 extends Config(  new WithL2Capacity(256)
++ new WithL2Cache ++ new DefaultFPGAConfig)
```

- You see here we gave this a new name, and we added two arguments. The arguments are self-explanatory, but the `WithL2Capacity(256)` is very important for targeting the Zedboard. If this is not included, `WithL2Capacity(2048)` is inferred. This is is 2048K Bytes of L2 cache, which certainly will not fit on the Zedboard. The maximum that can be instantiated is `WithL2Capacity(256)`

- Save this file.

5.1.2 fpga-zynq Configuration

- We now need to create a new fpga-zynq configuration that will utilize our new rocket-chip configuration. Open the configuration file for this:

```
emacs $TOP/common/src/main/scala/Configs.scala
```

- Locate the following line:

```
class ZynqConfig extends Config(new WithZynqAdapter ++ new DefaultFPGAConfig)
```

- This is the default configuration for our FPGA project. We will create a new configuration upon this but references our new rocket-chip configuration. Add the following:

```
class ZynqL2Config extends Config(new WithZynqAdapter ++ new DefaultFPGAConfigL2)
```

- Save this file.

5.1.3 Vivado Project

- Now that we have the two configuration files modified, we will need to run make rocket again, but reference out new fpga-zynq configuration. Run the following commands

```
cd $TOP/zedboard
```

- Build the new Vivado project by running:

```
make rocket CONFIG=ZynqL2Config
```

- At the end of this, you should see a new folder in the folder named `zedboard_rockethchip_ZynqL2Config`.

5.1.4 Build Bitstream

- Now the the RTL files have been generated, we need to build the bitstream that is loaded into the Programmable Logic of the Zynq on the Zedboard. Ensure that you are in the zedboard folder:

```
cd $TOP/zedboard
```

- Here we will make the project as we did before, but must specify our new configuration:

```
make project CONFIG=ZynqL2Config
```

5.1.5 Update Boot Image

- To load in the new configuration, a new `boot.bin` file will need to be created. Here we also need to specify out new configuration:

```
make fpga-images-zedboard/boot.bin CONFIG=ZynqL2Config
```

5.1.6 Upload Boot Image

- Now you just need to upload the boot image to the SD card:

```
make load-sd SD=path_to_mounted_sdcard
```

6 Conclusion

You should have successfully modified the rocket core, and loaded the new core onto the FPGA with this tutorial. We encourage you to look around the .scala files both in the fpga-zynq folders, and down int eh rocket-chip folders. Specifically, these are located in these two locations:

`/fpga-zynq/common/src/main/scala/`
`/fpga-zynq/rocket-chip/src/main/scala/`