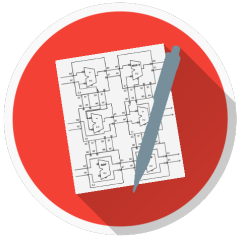
Documentation



CRC Configurator

Konstantin Lübeck, March 9, 2017

This document serves as documentation for the CRC Configurator.

If in doubt, use the source Luke.

Requirements

- `Java 1.8.0_40` or higher
- IntelliJ IDEA CE 2016.1 or higher

Compile

After cloning the repository from gitlab one has to compile the program into a `*.jar`. This is done by using IntelliJ IDEA CE 2016.

1. Open IntelliJ and click on *Import Project*
2. Choose the directory `idea` from the cloned repository and click on *OK*.
3. Choose *Create project from existing sources* and click on *Next*.
4. Set an appropriate *Project Name* (`crc_configurator`), make sure the file path is set to `crc_configurator/idea`, and click on *Next*. (Overwrite `.idea` when asked)
5. Click on *Next* two times in the library dialog. (Overwrite `idea.iml` when asked)
6. Choose the SDK (`Java 1.8.0_40` or higher) and click on *Next*.
7. Click on *Finish*.
8. Go to menu *File* → *Project Structure*.
9. Choose the *Artifacts*.
10. Click on the +-sign.
11. Choose *JAR* → *From modules with dependencies*.
12. Click on *OK* in the *Create JAR from Modules* dialog.
13. Click on *OK* in the *Project Sturcture* dialog.
14. Go to menu *Build* → *Build Artifacts...*
15. Choose Action *Build*.
16. Open a terminal and navigate to the cloned git repository.
17. In the git repository go to `idea/out/artifacts/idea_jar`
18. Rename the `idea.jar` to `crc_configurator.jar`.
19. Copy the `crc_configurator.jar` to a desired location.

Run

Go to the location of the `crc_configurator.jar` and execute the following command:

```
java -jar crc_configurator.jar
```

Usage

New File

To open a new CRC description file go to menu *File* → *New*. Set values for *Rows*, *Columns*, *Static Conf. Lines*, and *Dynamic Conf. Lines*, and click on *Create* to create the new CRC description file.

Set FU Functions

To set the possible FU functions for each PE go to the *Hardware Model* tab. Open the FU functions dialog for a PE by double clicking on the PE's FU. Check the checkboxes of the desired functions. If you want to apply the selected FU functions to all PEs check the checkbox *Apply to all PEs*. When you are done click on *Save*. The enabled FU functions can be seen below each PE in the *Hardware Model* tab.

Configure a CRC

To create a configuration for a CRC choose one of the *Static Configuration* or *Dynamic Configuration* tabs. To choose a FU operation for a PE right click or double click on the name of the operation in the FU (default *NOP*). To choose if the FU should treat values as signed or unsigned right click or double click on signed/unsigned (default unsigned). To set the driver for a FU input or a PE output right click or double click on the gray pad next to the FU or arrows and choose a driver. To set no driver click on the checked item in the context menu while choosing a driver.

Edit a CRC

To change the dimensions (rows and columns) or the amount for static and dynamic configuration lines go to menu *File* → *Edit*. Set the desired values for *Rows*, *Columns*, *Static Config. Lines*, and *Dynamic Config. Lines* and click on *Apply*. **Caution:** If a value is decreased data will be lost.

CRC/PE Configuration Bits

To get the CRC/PE configuration bits go to menu *File* → *Export Bits*. The *Export Bits* dialog displays the bits for the Verilog parameters.

Export Images of the Hardware Model and the Configurations

If you wish to export an image of the hardware model choose the *Hardware Model* tab and go to the menu *File* → *Export PNG*. Now choose a file name and a desired destination for the PNG image and click on *Save* to export the PNG image of the hardware model.
If you wish to export an image of a configuration choose a *Static Configuration* or *Dynamic Configuration* tab and go to the menu *File* → *Export PNG*. Now choose a file name and a desired destination for the PNG image and click on *Save* to export the PNG image of a configuration.

Export Verilog Code

In order to export Verilog code for the hardware model and the static configurations go to the menu *File* → *Export Verilog Code*. You can now choose if there should be FIFOs between all PEs or not by checking the *FIFOs between PEs* checkbox. Afterwards you can choose the location and the name of the Verilog file. You can either type the absolute path in the text filed *Path to Verilog File* or choose a desired destination and name graphically by clicking on *Choose*. **Caution:** if a file already exists at the chosen loaction it will be overwritten. It is also possible to generate a test bench for the Verilog code and a QuestaSim script for the simulation of the test bench with the Verilog code. In order to generate these two files check the *Generate test bench and QuestaSim script* checkbox. The location and the file names are set according to the Verilog file name. **Caution:** if files with those names already exist at the chosen location those will be overwritten.

To run the simulation with QuestaSim start QuestaSim an enter the following command:

```
do /path/to/questasim/script/verilog_file_name_questa_rtl.do
```

Command Line Interface

The CRC Configurator has a command line interface which provides shortcuts for opening a CRC description file, exporting configuration Bits and exporting images of the hardware model and configurations without using the GUI.
To simply open a file (with GUI) you can write the path to a CRC description file after the command which starts the CRC Configurator:

```
java -jar crc_configurator.jar /path/to/crc/description/file.json
```

If you wish to see the configuration bits for a CRC description file on the command line use the following command:

```
java -jar crc_configurator.jar /path/to/crc/description/file.json -eb
java -jar crc_configurator.jar /path/to/crc/description/file.json --export-bits
```

If you wish to generate images for the hardware model and all configurations for a CRC description file use the following command:

```
java -jar crc_configurator.jar /path/to/crc/description/file.json -epngs
java -jar crc_configurator.jar /path/to/crc/description/file.json --export-pngs
```

It is also possible to mix those export arguments.
By using the command line argument `-h` or `--help` you can see a short text which explains all possible command line arguments.