HLS Compiler Setup for Linux

Yannic Schneider - 08.03.2019

Inhaltsverzeichnis

Prerequisites

Software

Linux Ubuntu here
Quartus lite18.1.0.625 – found here (requires free intel account)

Installation

HLS Compiler

The following steps assume a working copy of Ubuntu.

After the installation of Ubuntu is completed, the repositories should be refreshed with

```
apt-get update upgrade
```

After the installation of Quartus Prime, we need to install the required packages with

```
sudo apt-get install build-essential
```

Finally, we add modelsim to the path (adjust for your quartus location). To permanently keep this change, we edit the bashrc file:

```
gedit ~/.bashrc
```

At the end of the file we add:

```
export PATH=$PATH:home/parallels/intelFPGA_lite/18.1/modelsim_ase/bin
```

To run the hls compiler, respectively the `i++` command, the `init_hls.sh` in the hls folder of the quartus install directory must be run. This can be avoided by adding the following variables to the path using the export command (adjust for your quartus install path):
We can again add it to the the bashrc file of course, just like before

```
/home/parallels/intelFPGA_lite/18.1/hls/bin
/home/parallels/intelFPGA_lite/18.1/hls/host/linux64/lib
```

Finally, we should add the location of the shared libraries to the LD_LIBRARY_PATH:

```
export LD_LIBRARY_PATH="/home/parallels/intelFPGA_lite/18.1/hls/host/linux64/lib/"
```

The end of the bashrc file should now look like this:

```
export QSYS_ROOTDIR="/home/parallels/intelFPGA_lite/18.1/quartus/sopc_builder/bin"
export PATH=$PATH:/home/parallels/intelFPGA_lite/18.1/modelsim_ase/bin
export PATH=$PATH:/home/parallels/intelFPGA_lite/18.1/hls/bin
export PATH=$PATH:/home/parallels/intelFPGA_lite/18.1/hls/host/linux64/lib
export LD_LIBRARY_PATH="/home/parallels/intelFPGA_lite/18.1/hls/host/linux64/lib/"
```

where the first line was added with the installation of quartus.

Now the setup can be tested with an example in the examples folder inside the hls directory from before. Navigate inside an example folder
(e.g. `/home/parallels/intelFPGA_lite/18.1/hls/examples/counter/`)

To create a x86-64 binary, we can use the following command:

```
make test-x86-64
```

To create a hardware executable, we can use the following command:

```
make test-fpga
```

In both cases the console echoes the i++ command that was used to create the file.

The created files can be executed with the ./test-x86-64, or ./test-fpga command.