HLS Compiler Setup for Linux

Yannic Schneider - 08.03.2019

Inhaltsverzeichnis

Prerequisites

Software

Cent OS Linux 7.6 1810 – found here
Quartus lite18.1.0.625 – found here (requires free intel account)

Installation

HLS Compiler

The following steps assume a working copy of CentOS.

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

```
sudo yum update
```

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

```
sudo yum groupinstall "Development Tools" "Additional Development"
```

To support the simulation with Model Sim we need to install the following packages:

```
sudo yum install -y glibc.i686 glibc-devel.i686 libX11.i686 \
 libXext.i686 libXft.i686 libgcc.i686 libgcc.x86_64 \
 libstdc++.i686 libstdc++-devel.i686 ncurses-devel.i686 \
 qt.i686 qt-x11.i686
```

Finally, we add modelsim to the path (adjust for your quartus location):

```
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):

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

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.