TangerineSDR

Notes on Quartus II and NIOS Installation for Ubuntu 18.04 and 20.04

White Paper

Version Number: 0.1

Version Date: July 27, 2021

Document Number:

VERSION HISTORY

Version Number	Implemented By	Revision Date	Approved By	Approval Date	Description of Change
0.1	T. McDermott	July 27, 2021			Original Issue

1. Scope

The TangerineSDR project Data Engine Version 1 is based on the MAX 10 FPGA by Altera. Altera was acquired by Intel.

Much of the documentation available is mixed between Altera and Intel branding. Additionally much of the documentation seems to have broken links after the transition to the Intel website URL.

The last version of Quartus II supporting the MAX 10 FPGA is version 20.1.

Additionally TSDR uses the NIOS processor embedded within the FPGA as a soft processor (consuming Logic Elements, not hard-silicon). There is a separate GUI for NIOS software development that can generate FPGA flash programming.

This white paper outlines a recipe for installing Quartus II and NIOS that has been tested on real and virtual Ubuntu 20.04 operating systems. It was also tested on a virtual Ubuntu 18.04 instance.

2. Quartus II installation

The Quartus II software version 20.1 is available on the Intel website. It requires a user registration in order to download. There is a free version of Quartus and Modelsim. These instructions assume that you have download the Quartus II Lite edition (the free edition) for the Linux operating system.

https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/download.html

After about version 16 the Eclipse development environment that NIOS needs was removed from the installer.

Part 3 of this document outlines how to get NIOS operational.

The Quartus II installation is relatively straight forward. Install it per the insctrucitons from the Intel website.

The installation will create a desktop launcher ICON which must be located on, well, the desktop. The desktop file is normally placed in /home/user, copy or move it to the desktop. The file is called “Quartus (Quartus Prime 20.1) Lite Edition.desktop”.

On the desktop right-click that icon and select 'Allow Launching' if it doesn't automatically convert to a launcher.

The large task buttons on the main screen:

- Compare Editions
- Buy Software
- Documentation
- Training
- Support
- What's New
- Notifications

do not work.

There is an error message:
14565 Can't connect to the Intel FPGA website to check for updates.

Most references to the Intel website seem to result in broken links.

3. Eclipse, Java, and NIOS

A separate download for the Eclipse development environment is now required. This is a single download file with a somewhat messy installation process.

That download also bundles a binary Java Runtime Environment (JRE) that does not function properly in Ubuntu 18.04 or 20.04 and must be replaced. Additionally the replacement Java JRE does not correctly handle some graphics references, and that replacement JRE configuration needs to be altered.

3.1. Installing Eclipse

Download the file: eclipse-cpp-mars-2-linux-gtk-x86_64.tar
From the Intel site:
https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html

Use the Tab "Select by software" then select
NIOS II Embedded Design Suite Lite
Then select the version 20.1 and download.

You need to have already registered as a user on the Intel site. The below directions were I think from Altera, and are repeated here. On my system the Quartus installation directory was /home/<user>/intelFPGA_lite/20.1

To install the Nios® II SBT for Eclipse, follow the steps below:

1. Download CDT 8.8.1 which is Eclipse C/C++ IDE for Mars.2 [Use the link above]

CDT 8.8.1 for Linux [it should have the filename as above]

[You should have already done this step, above].
Note: Try the mirror links if the main source link doesn't work.

2. Extract the downloaded file into <Intel Quartus installation directory>/nios2eds/bin. You should see the <Intel Quartus installation directory>/nios2eds/bin/eclipse folder after extraction is done.

3. Rename the <Intel Quartus installation directory>/nios2eds/bin/eclipse folder to <Intel Quartus installation directory>/nios2eds/bin/eclipse_nios2

Note: the following step can be a little confusing. Read it carefully. In step 3 above a directory was created that contains the plugins needed.

4. Extract the <Intel Quartus installation directory>/nios2eds/bin/eclipse_nios2_plugins.zip (or tar.gz for Linux) to <Intel Quartus installation directory>/nios2eds/bin. The extraction will override files in <Intel Quartus installation directory>/nios2eds/bin/eclipse_nios2.

5. Verify the extraction is done correctly by making sure you see the <Intel Quartus installation directory>/nios2eds/bin/eclipse_nios2/plugin_customization.ini file

3.2. Java installation and re-direction

The Java runtime environment JRE that is bundled above doesn't work. Install the Java 8 headless JRE using the following commands:

1. Install java 8 java runtime:

```
$ sudo apt install openjdk-8-jre-headless
```

2. Find where your java package is located:

```
$ update-alternatives --list java

/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
```

This will list where the new java is installed. We will use only the path upto ../jre not bin or java subdirectories.

Create a soft link to the new jre64 and disable the JRE provided by Eclipse. If you installed Quartus somewhere else, adjust the paths accordingly.

```
$ cd /intelFPGA_lite/20.1/quartus/linux64

$ mv jre64 jre64_old

$ ln -s /usr/lib/jvm/java-8-openjdk-amd64/jre jre64
```

This will make quartus use the JRE pointed at by the sofltink rather than the one installed by the Eclipse installer.

3.3. Update Java Configuration

The headless version of Java needs to disable the accessibility.properties.

This can be done by editing the accessibility.properties file for OpenJDK 8.

```
$ sudo nano /etc/java-8-openjdk/accessibility.properties
```

Comment out the following line:

#assistive_technologies=org.GNOME.Accessibility.AtkWrapper

4. USB Blaster II

The FPGA MAX10 development board can be programmed via USB using a USB Blaster II that is built into the development board. For Linux no special drivers are needed. However the USB interface does need to be identified to Ubuntu via udev rules.

```
$ sudo nano /etc/udev/rules.d/51-usbblaster.rules
```

Enter the following text exactly as 3 lines (not 5 lines) of text. Each SUBSYSTEM== line is a single line of text. MS Word wraps the line when the text is larger font making it sometimes appear as more lines.

```
# USB-Blaster II
SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6010", MODE="0666", GROUP="plugdev"
SUBSYSTEM=="usb", ATTR{idVendor}=="09fb", ATTR{idProduct}=="6810", MODE="0666", GROUP="plugdev"
```

Then write out and save the file.
Here is the above in larger font, but unfortunately line-wrapped for display:

```
# USB-Blaster II
SUBSYSTEM=="usb", ATTR{idVendor}=="09fb",
ATTR{idProduct}=="6010", MODE="0666", GROUP="plugdev"
SUBSYSTEM=="usb", ATTR{idVendor}=="09fb",
ATTR{idProduct}=="6810", MODE="0666", GROUP="plugdev"
```

Restart the udev service:

```
$ sudo service udev restart
```

Note: the USB Blaster II will not work on a virtual machine unless the IOMMU group containing the appropriate USB interface has been passed through to the virtual machine.

5. Final comments

This should allow you to launch Quartus II, the programmer using USB, and NIOS/Eclipse environment. Additionally the NIOS command windows should also launch without throwing exceptions. You will need to have an actual MAX10 device connected via USB in order for the tools to find something to program.