01signal.com

Quartus / Linux: Programming the FPGA with command-line

Command-line?

Yes, it much more convenient than the GUI programmer. Programming an FPGA is a repeated task, always the same file to the same FPGA on the same board connected to the computer. And somehow the GUI programming tools turn it into a daunting ceremony (and sometimes even a quiz, when it can't tell exactly which device is connected, so I'm supposed to nail the exact one).

With command line its literally picking the command from bash history, and press Enter. And surprisingly enough, the command line tool doesn't ask the silly questions that the GUI tool does.

First, some mucking about

Set up the environment:

```
$ /path/to/quartus/15.1/nios2eds/nios2_command_shell.sh
```

To list all devices found (cable auto-detected):

```
$ quartus_pgm --auto
Info: *******************************************************************
Info: Running Quartus Prime Programmer
    Info: Version 15.1.0 Build 185 10/21/2015 SJ Lite Edition
    Info: Copyright (C) 1991-2015 Altera Corporation. All rights reserved
[ ... ]
    Info: agreement for further details.
    Info: Processing started: Sun May 27 15:06:22 2018
Info: Command: quartus_pgm --auto
Info (213045): Using programming cable "USB-BlasterII [2-5.1]"
1) USB-BlasterII [2-5.1]
  02B040DD   5CGTFD9(A5|C5|D5|E5)/..
  020A40DD   5M2210Z/EPM2210

[ ... ]
```

Note that listing the devices as shown above is **not necessary** for programming. It might be useful to tell the position of the FPGA in the JTAG chain, maybe. Really something that is done once to explore the board.

jtagd

It's important to be aware of this deamon, which listens to TCP/IP port 1309: It's responsible for talking with the JTAG adapter through the USB bus, so both the GUI and command line programmer utilities rely on it. If there's no daemon running, both of these launch it.

But if you use multiple versions of Quartus, this may be a source of confusion, in particular if you make a first attempt to program an FPGA with an older version, and then try a newer one. That's because the newer version of Quartus will keep using the older version of jtagd, possibly failing to work with recent devices. Bottom line: If wonky things happen, this won't hurt:

```
$ killall jtagd
```

Programming

quartus_pgm displays most of its output in green. Generally speaking, if there's no red text, all went fine.

```
$ quartus_pgm -m jtag -o "p;path/to/file.sof"
```

Or add the position in the JTAG explicitly (in particular if it's not the first device). In this case it's @1, meaning it's the first device in the JTAG chain. If it's the second device, pick @2 etc.

```
$ quartus_pgm -m jtag -o "p;path/to/file.sof@1"
Info: *******************************************************************
Info: Running Quartus Prime Programmer
    Info: Version 15.1.0 Build 185 10/21/2015 SJ Lite Edition
    Info: Copyright (C) 1991-2015 Altera Corporation. All rights reserved
    Info: Your use of Altera Corporation's design tools, logic functions
    Info: and other software and tools, and its AMPP partner logic
    Info: functions, and any output files from any of the foregoing
    Info: (including device programming or simulation files), and any
    Info: associated documentation or information are expressly subject
    Info: to the terms and conditions of the Altera Program License
    Info: Subscription Agreement, the Altera Quartus Prime License Agreeme
    Info: the Altera MegaCore Function License Agreement, or other
    Info: applicable license agreement, including, without limitation,
```

```
    Info: that your use is for the sole purpose of programming logic
    Info: devices manufactured by Altera and sold by Altera or its
    Info: authorized distributors.  Please refer to the applicable
    Info: agreement for further details.
    Info: Processing started: Sun May 27 15:35:02 2018
Info: Command: quartus_pgm -m jtag -o p;path/to/file.sof@1
Info (213045): Using programming cable "USB-BlasterII [2-5.1]"
Info (213011): Using programming file p;path/to/file.sof@1 with checksum (
Info (209060): Started Programmer operation at Sun May 27 15:35:05 2018
Info (209016): Configuring device index 1
Info (209017): Device 1 contains JTAG ID code 0x02B040DD
Info (209007): Configuration succeeded -- 1 device(s) configured
Info (209011): Successfully performed operation(s)
Info (209061): Ended Programmer operation at Sun May 27 15:35:09 2018
Info: Quartus Prime Programmer was successful. 0 errors, 0 warnings
    Info: Peak virtual memory: 432 megabytes
    Info: Processing ended: Sun May 27 15:35:09 2018
    Info: Elapsed time: 00:00:07
    Info: Total CPU time (on all processors): 00:00:03
```

If anything goes wrong — device mismatch, a failure to scan the JTAG chain or whatever, it will be hard to miss because of the errors written in red. The sweet thing with the command line interface is that every attempt starts from fresh, so just turn the board on (the usual reason for errors) and give it another go.

Cyclone 10 GX FPGA development kit

This board caused me some extra trouble, so a few words about it. When connected, it appears as 09fb:6810, however after attempting to program the FPGA (note the "@2" in the end) with

```
$ quartus_pgm -m jtag -o "p;thecode.sof@2"
Error (213019): Can't scan JTAG chain. Error code 86.
```

it changes to 09fb:6010. So there's clearly some reprogramming of firmware (the log shows a disconnection and reconnection with the new ID). The board is detected as GX0000406 by the Quartus GUI Programming Tool, but clicking "Auto Detect" yields "Unable to scan device chain. Hardware is not connected".

OK, what about a scan?

```
$ quartus_pgm --auto
[ ... ]
Info (213045): Using programming cable "10CGX0000406 [1-5.1.2]"
1) 10CGX0000406 [1-5.1.2]
```

```
  Unable to read device chain - Hardware not attached
```

The problem in my case was apparently that the jtagd running was launched by an older version of Quartus, which didn't recognize Cyclone 10 devices. So follow the advice above, and kill it. After that, programming with the command above worked with Quartus Pro 17.1:

```
$ quartus_pgm --auto
[...]
Info (213045): Using programming cable "USB-BlasterII [1-5.1.2]"
1) USB-BlasterII [1-5.1.2]
  031820DD   10M08SA(.|ES)/10M08SC
  02E120DD   10CX220Y
```