Running ARTIK 530 with FPGA through USB: Pipelined AES128 Example

Version 0 Revision 1

August 15, 2018 (August 5, 2018)

Future Design Systems, Inc.
www.future-ds.com / contact@future-ds.com

Abstract

This document addresses how to run Samsung ARTIK 530 Development Board with FPGA, where CON-FMC board connects ARTIK 530 to FPGA through USB.

Table of Contents

1 Overview

Future Design Systems' CON-FMC™ is a FMC[1] board and it connects ARTIK to the FPGA board through USB. This enables any program running on the ARTIK to communicate with hardware block in the FPGA using a simple C/C++ API. This feature can offload CPU to FPGA, where computing intensive function can be run by the FPGA.

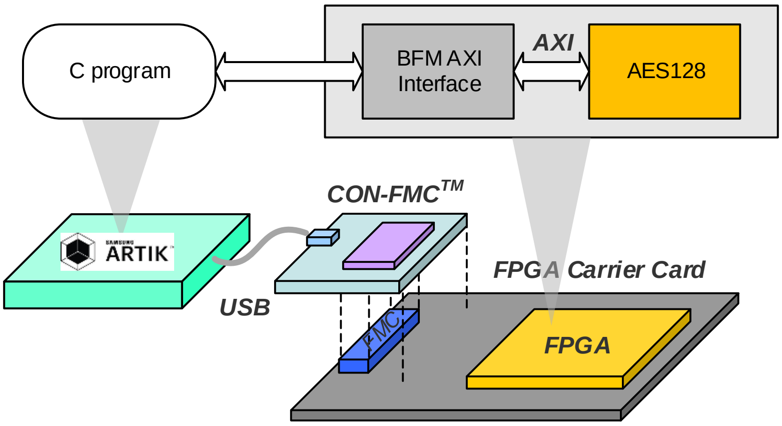


Figure 1: Conceptual structure

Figure 1 shows an example case, where 'AES128' is chosen as an example design. ARTIK runs AES application and the FPGA takes care of actual AES calculation. This implies that FPGA can provide offloading of computation.

2 Setup

Figure 2 shows overall environment to run ARTIK 530 along with ZedBoard, where ARM on the ARTIK board runs user program and FPGA on the ZedBoard runs user design. CON-FMC provides generic communication channel between the ARTIK board and the ZedBoard.

As shown in Figure 2, four components (ARTIK, CON-FMC, FPGA board, PC) are connected accordingly.

1. Make sure that all power is turned off
2. Mount CON-FMC board on the ZedBoard through FMC connector
3. Connect USB between ARTIK and CON-FMC
4. Connect USB-serial between ARTIK and PC
5. Connect USB-JTAG between ZedBoard and PC
6. Turn on power of ZedBoard and ARTIK

[1] FPGA Mezzanine Card (FMC) is an ANSI/VITA (VMEbus International Trade Association) 57.1 standard.

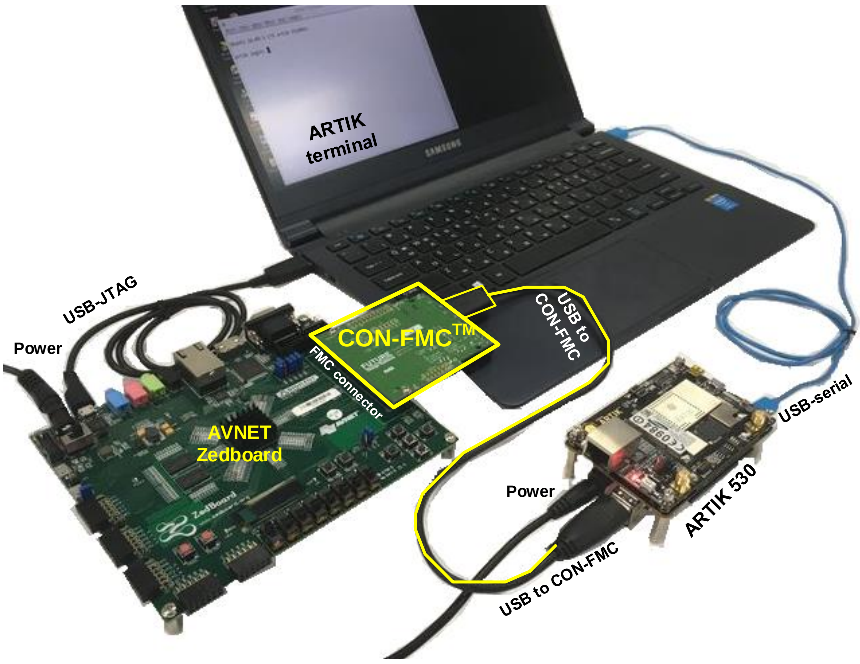


Figure 2: Overall setup

To run this environment following steps are required:

- CON-FMC package should be installed on ARTIK 530 (this step is explained in 'section 3 SW installation')
- FPGA board should be programmed for user design (this step is explained in 'section 4.2 FPGA configuration')

3 SW installation

CON-FMC package should be installed on ARTIK. More details refer to 'Appendix A'

- Get the package from Future Design Systems, which will be 'confmc.armv7l.ubuntu.tgz'.
  - Alternatively, download from github:
    - https://github.com/github-fds/confmc.armv7l.ubuntu
- Untar the package, if requires
- run 'coninstall.sh'

If 'libusb-1.0' is not installed yet, it should be ready since CON-FMC utilizes 'libusb-1.0'. For more details, refer to 'Appendix B'.

By default, CON-FMC package will be installed in '/opt/confmc/2018.06' directory, where '2018.06' stands for version of the package and the version can be differ.

4 Example

As shown in Figure 1, a pipelined AES[2] example[4] is used to test this environment. Figure 3 shows a functional diagram of AES HW block, which has AMBA AXI interface.

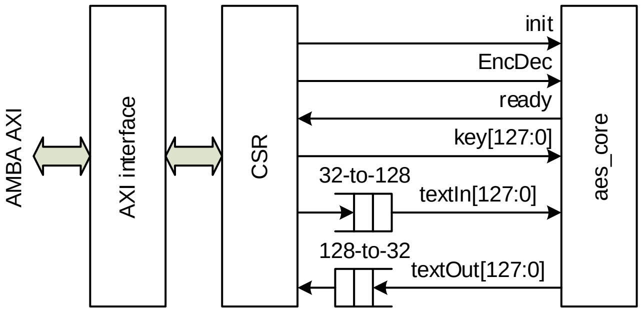


Figure 3: AES block

This example reads RGB pixel data from BMP file and creates two BMP files; one uses encrypted data the other uses decrypted data.

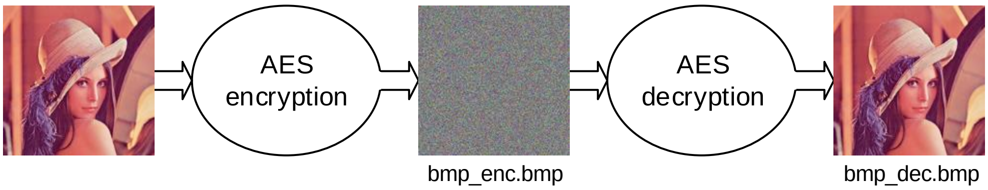


Figure 4: AES data flow

- Get the example package from Future Design Systems, which will be ‘fex_0006_amba_axi_aes.tgz’.
  - ✧ Alternatively, download from github:
    - ➢ https://github.com/github-fds/examples_B
- Untar the package

To run this example, three steps are required.

1. FPGA configuration
2. Compiling application
3. Running application

4.1 Quick start

Followings are a quick step to run.

1. Make sure ‘CON-FMC package’ and ‘LibUSB’ are installed.

[2] Advanced Encryption Standard (AES) implanting Rijndael algorithm.

2. Do not forget to run ‘/opt/confmc/2018.06/setting.sh’
3. Go to ‘$(PROJECT)/hw/pnr/vivado.zed.lpc/download’ directory
4. Invoke ‘make’
5. Go to ‘$(PROJECT)/sw.native/test_img_bmp’ directory
6. Invoke ‘make’
7. Invoke ‘make run’
8. Check options with ‘-h’ option such as ‘./test -h’

4.2 FPGA configuration

At first the FPGA on the ZedBoard should be ready to run with ARTIK by downloading FPGA bitstream. For details refer to ‘Appendix C’.

This step requires followings:

- FPGA bit-stream
  - Design for FPGA (for this example it is AES128 design)
  - It will be ‘fpga.bit’ in ‘fex_0006_amba_axi_aes/hw/pnr/vivado.zed.lpc’ directory.
- FPGA programming package, such as ISE or Vivado depending FPGA (for ZedBoard, Vivado is required)
  - ISE Labtool or Vivado Labtool is enough and these package can be download from Xilinx website and do not need any license.

4.3 Compiling

Following shows how to compile the example.

1. Set environment by calling ‘settins.sh’
2. Invoke ‘make’

```
[root@artik] ls
BmpHandle/  Clean.bat*  Clean.csh*  Clean.sh*  INDEX.txt  Makefile  images/
src/        src.aes/
[root@artik] source /opt/confmc/2018.06/settings.sh
[root@artik] make
gcc -c   -DTRX_BFM -DTRX_AXI -DRIGOR -DVERBOSE -std=gnu99 -O -
I/opt/confmc/2018.06/include -Isrc -I/opt/confmc/2018.06/hwlib/trx_axi/drv/c -Isrc.aes -
IBmpHandle/src -I../../iplib/aes128_axi/api/c -o obj/main.o src/main.c
…. not all shown …
gcc -o test obj/main.o obj/arg_parser.o obj/trx_axi_api.o obj/aes128_api.o obj/rijndael.o
obj/bmp_handle.o obj/test_bench.o  -L/opt/confmc/2018.06/lib/linux_armv7l -lconapi -lusb-1.0
[root@artik] ls
BmpHandle/  Clean.bat*  Clean.csh*  Clean.sh*  INDEX.txt  Makefile  images/
obj/        src/        src.aes/    test*
[root@artik] ./test -h
[Usage] ./test [options]
       -c   cid    card id: 0
       -i   img    image file
       -b   num    burst length: 256
       -m   mod    0=SW, 3=HW, 1=HW-SW, 2=SW-HW: 0
       -r          compare result
       -v   num    verbose level (default: 0)
```

```
     -h          print help message
Eg: ./test -c 0 -i lena.jpg -b 256 -m 3 -r
[root@artik]
```

This example has some options and following two are important.

- '-i img': specify BMP file to encrypt and decript.
- '-m mod': specify operation mode
  - '-m 0': means both encryption and decryption use software.
  - '-m 3': means both encryption and decryption use HW
  - '-m 1': means encryption by HW and decryption by SW

4.4 Running program

Following shows software only case.

```
 [root@artik] ./test -i images/lena_512x512.bmp -m 0
Encryption (SW): 11.138 secs [70.609-Kbyte/sec of 786432:512x512]
Decryption (SW): 11.079 secs [70.982-Kbyte/sec of 786432:512x512]
```

Following shows all hardware case, in which AES encryption and decryption are carried out by HW IP in the FPGA.

```
[root@artik] ./test -i images/lena_512x512.bmp -m 3
Encryption (HW): 0.960 secs [819.037-Kbyte/sec of 786432:512x512]
Decryption (HW): 0.893 secs [881.014-Kbyte/sec of 786432:512x512]
```

After completion, there will be three BMP files, which are

- bmp_enc.bmp: encrypted BMP
- bmp_dec.bmp: decrypted BMP
- bmp_org.bmp: replicated BMP of the source BMP

5 Performance

Since this environment makes use of FPGA as an additional computation component along with CPU, it gives offloading CPU to FPGA and can get acceleration depending on hardware and application.

Figure 5 shows elapse time to encrypt images of different size, where pure software execution has built with different optimization options; no-optimization, '-O' and '-Ofast'. HW-assisted execution can give x2 to x45 times fast result.

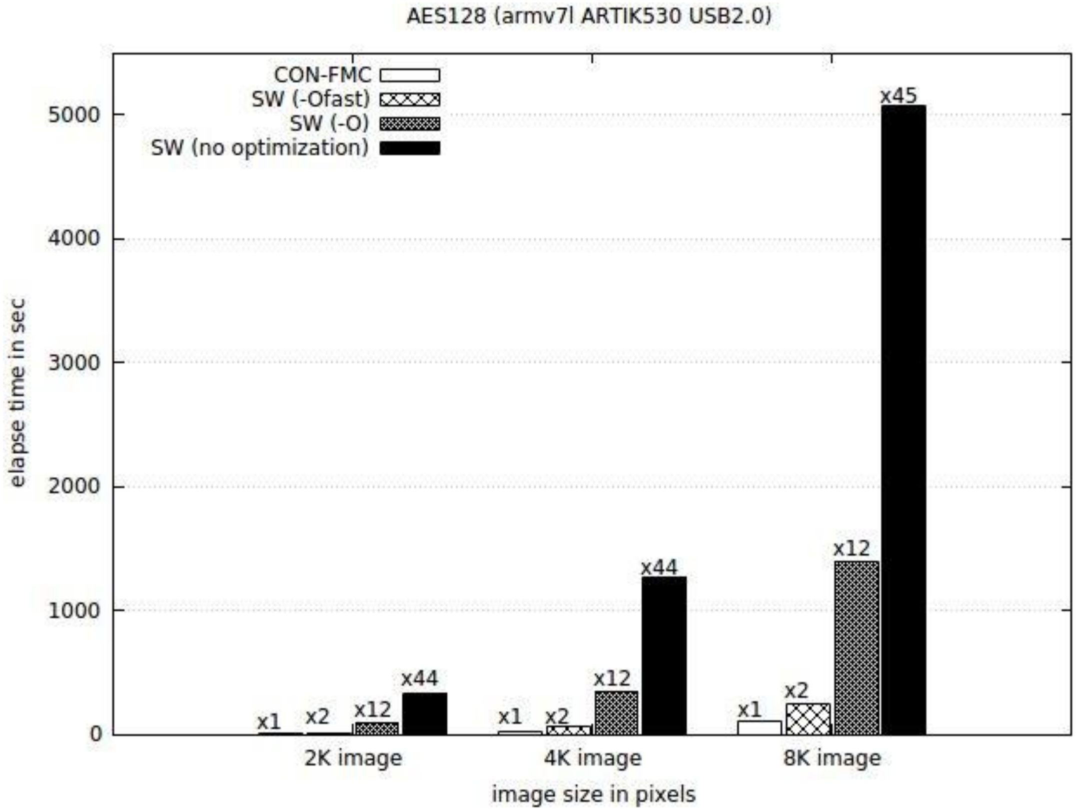


Figure 5: Elapse time comparison

6 References

[1] Samsung Semiconductor, Samsung ARTIK 530 Development Board User Guide, 2017.

[2] Avnet, ZedBoard (Zynq™Evaluation and Development) Hardware User's Guide, 2014.

[3] Future Design Systems, CON-FMC™ User Manual, FDS-TD-2018-03-001, 2018.

[4] Future Design Systems, Pipelined AES128 Rijndael with AMBA AXI, FDS-TD-2018-06-001, 2018.

Wish list

□

Revision history

□ 2018.08.05: Started by Ando Ki (adki@future-ds.com)

Appendix A: SW installation: CON-FMC

This section guides you to install CON-FMC software package step-by-step.

6.1 Get the package

Get CON-FMC software package from github[3].

```
[root@artik] git clone https://github.com/github-fds/confmc.armv7l.ubuntu.git
```

This will make ‘confmc.armv7l.ubuntu’ directory.

6.2 Install the software

Now install CON-FMC software package

```
[root@artik] cd confmc.armv7l.ubuntu
[root@artik] ./coninstall.sh -dst /opt/confmc/2018.06
/opt writable

Do you want to see full "End-User-License-Agreement for CON-FMC"
Yes (Y) or no (n): y

….. End User License Agreement ….

Do you agree "End-User-License-Agreement for CON-FMC"
Yes (Y) or no (n): y
Thank you and proceed to install CON-FMC software

Override "/etc/udev/rules.d/51-fds-rule.rules"
Yes (Y) or no (n): y

You may need to run followings or re-boot the system.
$ sudo udevadm control --reload-rules
$ sudo service udev restart
$ sudo udevadm trigger

CON-FMC software package has been installed at /opt/confmc/2018.06
Do not forget to run /opt/confmc/2018.06/settings.sh before using CON-FMC software.

[root@artik]
```

Appendix B: SW installation: LibUSB

This section guides you to install LibUSB package step-by-step, which is required to use CON-FMC.

[3] This step may requires network setting. For example, following two lines should be added in ‘/etc/resolve.conf’ file; ‘nameserver 128.0.1.1’ and ‘nameserver 192.168.1.1’

6.3 Check if installed or not

Following command checks LibUSB installation.

```
[root@artik] ldconfig -p | grep libusb
        libusb-1.0.so.0 (libc6,hard-float) => /lib/arm-linux-gnueabihf/libusb-1.0.so.0
        libusb-1.0.so (libc6,hard-float) => /usr/lib/arm-linux-gnueabihf/libusb-1.0.so
        libusb-0.1.so.4 (libc6,hard-float) => /lib/arm-linux-gnueabihf/libusb-0.1.so.4
```

6.4 Install LibUSB

If you do not see something like above message, then LibUSB installation is required.

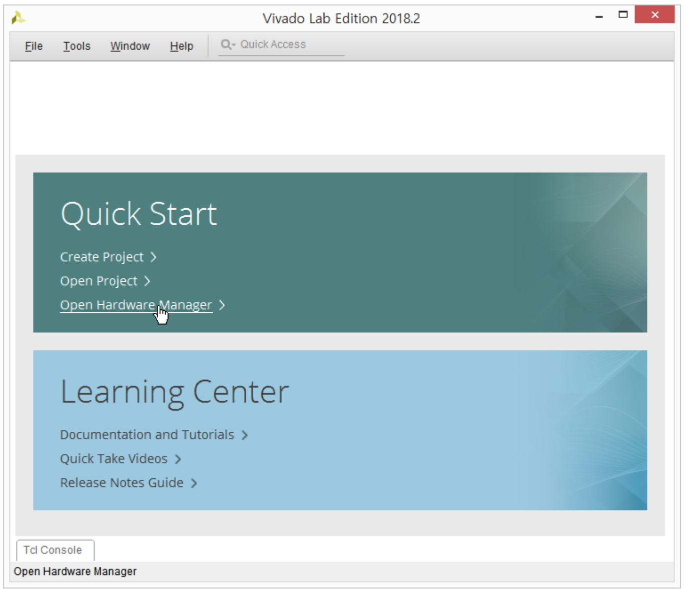
```
[root@artik] apt-get install libusb-1.0.0-dev
```

Appendix C: FPGA configuration

This section explains how to download bit-stream to the FPGA. Note that this example only use PL region.

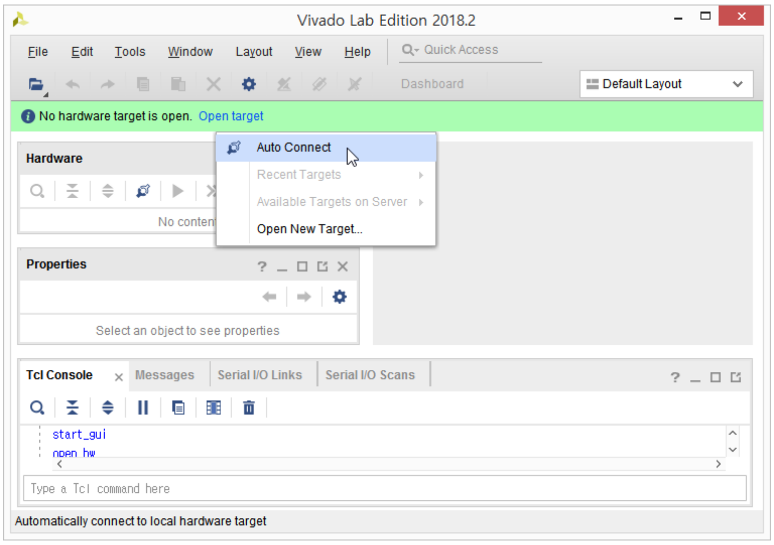
1. Invoke 'Vivado'[4]

2. Select "Hardware Manager".



3. Select "Open target" → "Auto Connect"

[4] There is a license-free version, called Vivado Lab Edition or Vivado Lab Solution, which is available from https://www.xilinx.com/support/download.html.



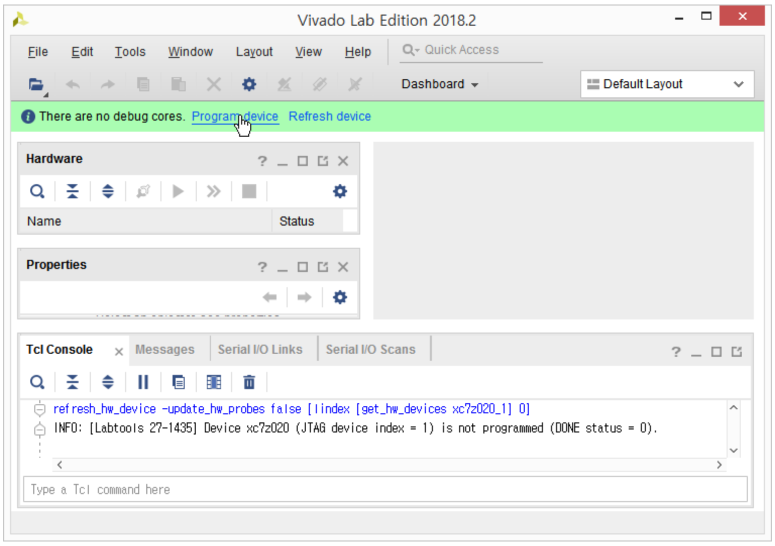
No hardware target is open. Open target

Select an object to see properties

```
start_gui
open_hw
```

Automatically connect to local hardware target

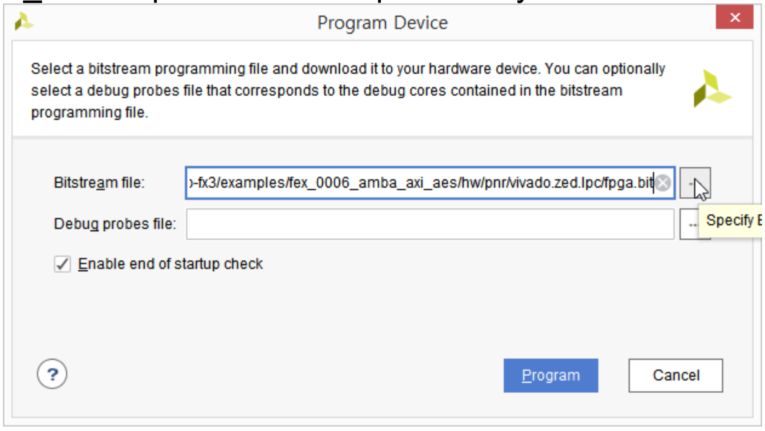
4. Select “Program device”



There are no debug cores. Program device Refresh device

```
refresh_hw_device -update_hw_probes false [lindex [get_hw_devices xc7z020_1] 0]
INFO: [Labtools 27-1435] Device xc7z020 (JTAG device index = 1) is not programmed (DONE status = 0).
```

5. Select proper bit file; ‘fpga.bit’ in ‘fex_0006_amba_axi_aes/hw/pnr/vivado.zed.lp’ directory



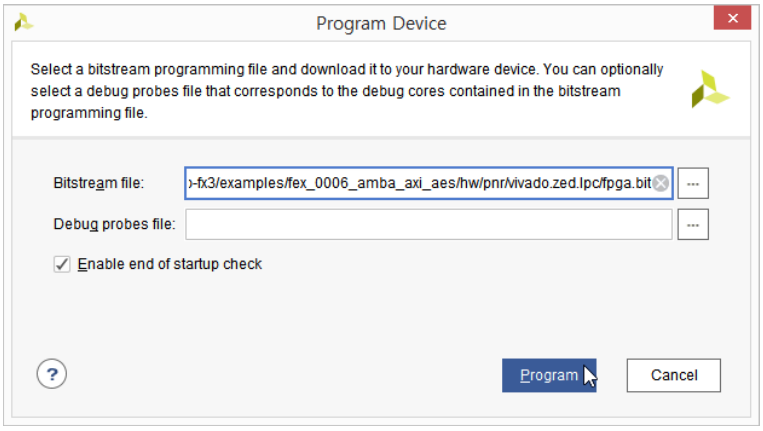
Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file:)-fx3/examples/fex_0006_amba_axi_aes/hw/pnr/vivado.zed.lp/fpga.bit

Debug probes file:

☑ Enable end of startup check

6. Select “Program”



Program Device

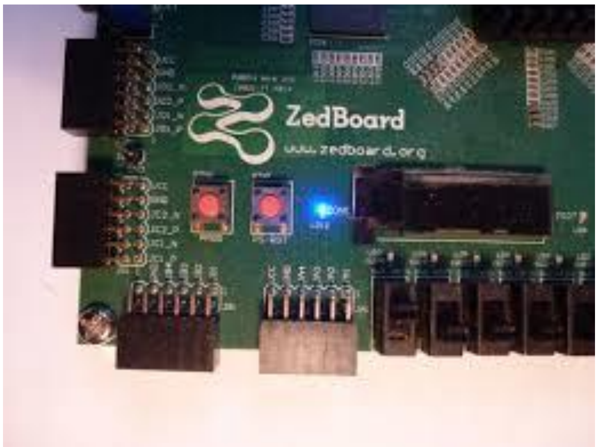
Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: ›-fx3/examples/fex_0006_amba_axi_aes/hw/pnr/vivado.zed.lpc/fpga.bit

Debug probes file:

☑ Enable end of startup check

7. Make sure that the configuration done LED should be lit blue.



– End of document –