Microsoft

Getting Started with Catapult on TACC

Date: September 5, 2018
Version: V1.2.1
Author: Dan Zhang/Rob Moon

Revision History

Revision/ Date	Notes	Updated by
V0.4 07/05/2016	Initial version	Dan Zhang
V1.1 2/15/2017	Updated to latest SDK v1.1 release	Dan Zhang
V1.2 11/10/2017	Updated for SDK v1.2 release	Rob Moon
V1.2.1 9/5/2018	Updated for OpenCL and updating links	Rob Moon

Contents

1 Introduction

1.1 Overview

Project Catapult is an open research system deployed at Texas Advanced Computing Center (TACC) in partnership with Microsoft Research. The goal of the project is to investigate the use of field-programmable gate arrays (FPGAs) as data center accelerators to improve performance, reduce power consumption and open new research avenues of investigation, particularly in Machine Learning. Catapult is currently deployed in production Microsoft datacenters accelerating the Bing search engine.

1.2 Catapult System Specifications

The system consists of 384 two-socket Intel Xeon-based nodes, each with 64 GB of memory and an Intel Stratix V D5 FPGA with 8 GB of local DDR3 memory. FPGAs communicate to their host CPUs via a PCIe Gen3 x8 connection, providing 8GB/s maximum bandwidth, and each FPGA can read and write data stored on its host node using this connection.

The FPGAs are connected to one another via a dedicated network using high-speed serial links. This network forms a two dimensional 6x8 torus within a pod of 48 servers, and provides low latency communication between neighboring FPGAs. This design supports the use of multiple FPGAs to solve a single problem, while adding resilience to server and FPGA failures.

Each node has the following specifications:

- Two Xeon E5-2450, 2.1GHz, 8-core, 20MB Cache, 95W
- 64GB RAM
- Four 2TB 7.2k 3G SATA 3.5"; Two 480GB 6G Micron SATA SSD 2.5"
- Intel 82599 10GbE Mezz Card
- Mt Granite Altera/Intel Stratix V FPGA Card
- Operating System: Windows Server 2012

1.3 Getting Access

After your proposal is approved, the next step is to apply for a TACC account at https://portal.tacc.utexas.edu/account-request. To access Intel FPGA documentation, forums, and downloads, you will also need a MyIntel account.

After you successfully activated your accounts, read the Microsoft Research License agreement: http://research.microsoft.com/en-us/um/legal/MSR-LA_Software_Restricted%20Rights_Catapult_Academic_Shell.htm

Then, please forward the account confirmation emails to catapult@microsoft.com, along with:

- Your proposal
- Full legal name
- Country ofcitizenship
- Current residential address
- University name and department
- Along with the following statement:

  All of the code that I will pass through FAbRIC CAD tools (such as Verilog files, Bluespec files, etc.) and the files needed to process that code (such as Makefiles) is either already open source (GPL version 2 or above, BSD, or MIT licenses) or I have the right to make it open source and are hereby making all of the code that I pass through FAbRIC CAD tools open source by one of those licenses. I will provide access to my source code to the CAD tool vendors and the FAbRIC administrators immediately. The simplest way to do that is to provide a repository account to the FAbRIC administrators. By default, the CAD tool vendors and/or the FAbRIC administrators agree not to publish the code publicly for at least 12 months.

  I acknowledge that the tools, servers, and FPGAs are potentially subject to export controls under U.S. and other applicable Government laws and regulations. I will comply with these laws and regulations and agree to obtain all required Government authorizations.

  I acknowledge that my access to and use of Microsoft's Project Catapult Academic Shell and Driver and related hardware provided by Microsoft is governed by, and subject to, the terms and conditions of the Microsoft Research License Agreement for the Microsoft Project Catapult Academic Shell and Driver. By accessing or using Microsoft Project Catapult materials, I represent and warrant that I have read the agreement, and I agree to be bound by it.

Your application will be reviewed and, if approved, a dedicated Catapult node will be assigned to you for your research project.

1.4 Setting up TACC Multi-Factor Authentication

TACC requires users to use Multi-Factor Authentication (MFA) to log into their machines. This requirement now extends to the Catapult servers at TACC. To set up MFA, you can follow TACC's guide located here; we also summarize the steps below. We recommend using the Android or iOS smartphone app. The SMS method only works if you have a domestic US phone number; international users must use the smartphone app.

To set up MFA for smartphone apps, first download the Android or iOS TACC Multi-Factor Authentication app:

Android: https://play.google.com/store/apps/details?id=edu.utexas.tacc.authenticator

iOS: https://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware?id=705212751

To set up MFA, log in to the TACC portal at: https://portal.tacc.utexas.edu/

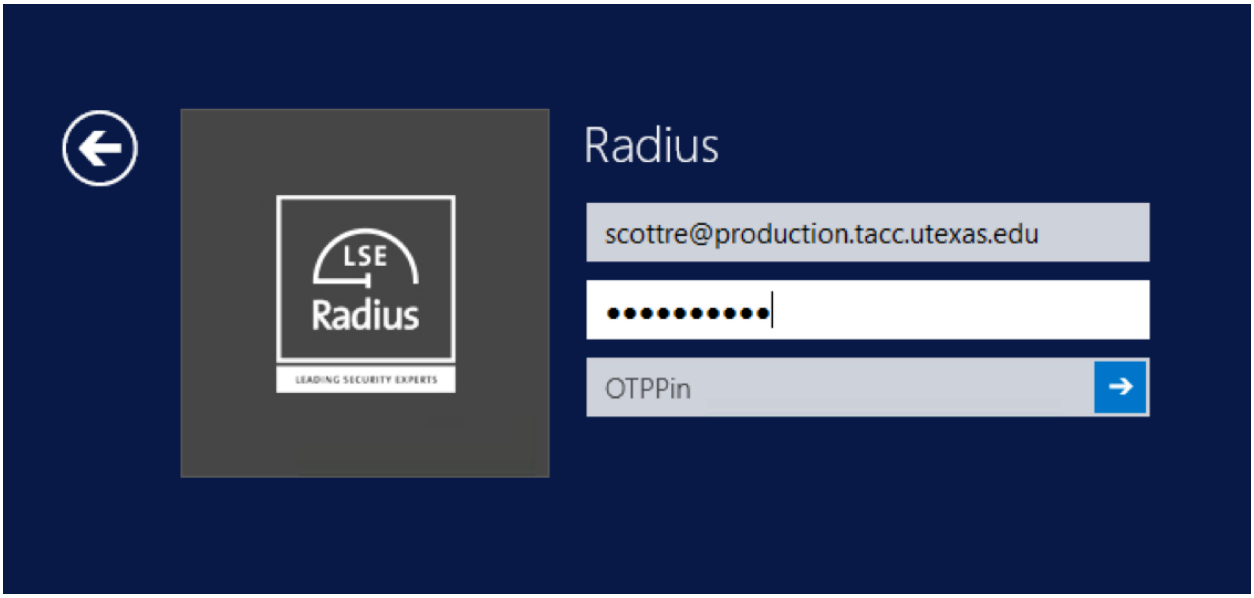
Click on "Manage Your Profile" on the upper right corner of the User Dashboard page. Then, click "Pair a Device", and then click on "Use the TACC Token App to pair". You will be shown a QR code. Open the

Android/iOS app and tap “+” to add a new pairing. Scan the QR code. The app will now begin generating token codes. Enter a valid token code into the website underneath the QR code. Your setup is now complete!

To set up MFA for SMS, log in to the TACC portal at: https://portal.tacc.utexas.edu/

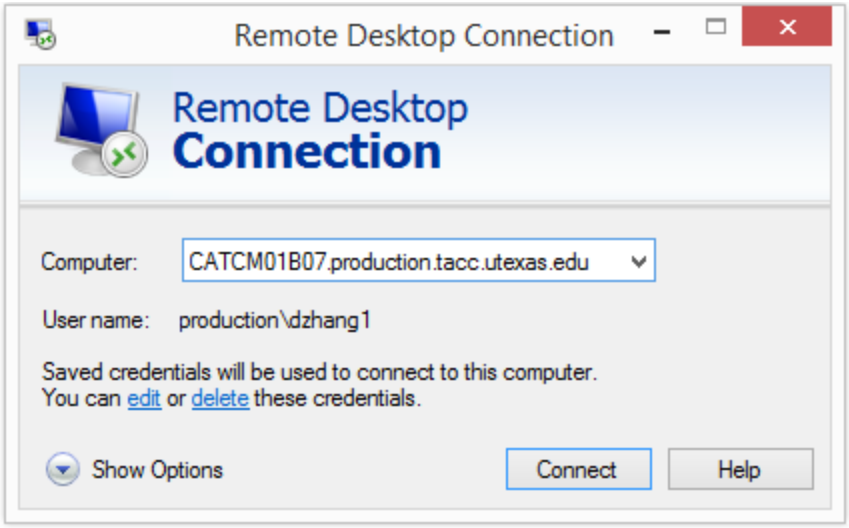
Click on “Manage Your Profile” on the upper right corner of the User Dashboard page. Then, click “Pair a Device”, and then click on “Use SMS on your mobile phone to pair”. Enter your phone number and click “Request code”. You’ll be sent an SMS with a 2-factor code. Enter the code and click “Confirm pairing”. Your setup is now complete!

When you Remote Desktop into the TACC Catapult machines, you may be asked to enter your OTPPin. If you are using the smartphone app, open the TACC smartphone app and enter the code provided. If you are using SMS, click on the blue arrow to the right of the OTPPin field. You should receive an SMS message with the code, which you can now type into the OTPPin field.



1.5 Connecting to the Catapult Servers

Launch Remote Desktop and connect to your assigned Catapult login node. The example below shows the user dzhang1 logging into the CATCM01B07 machine.



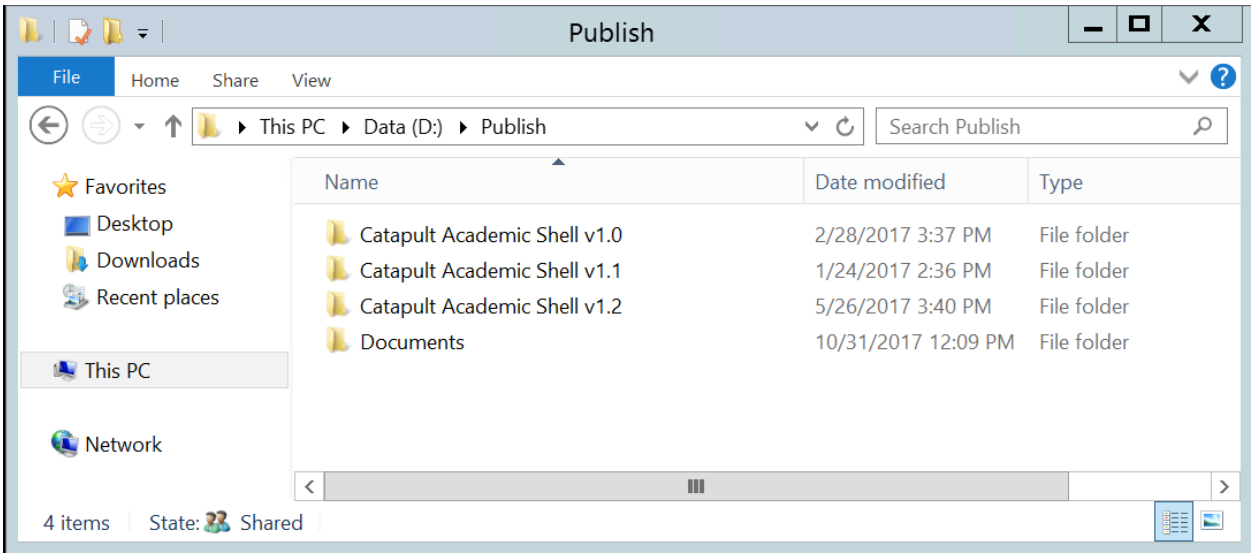
All Catapult machines will be in production.tacc.utexas.edu, and all TACC usernames will be in the "production" domain. Make sure you use backslash instead of forward slash. Your password will be the same password as that of your regular TACC account. You may be asked to provide a one-time pin upon login. If so, refer to the directions in Section 1.4 Setting Up TACC Multi-Factor Authentication.

1.6 Accessing the Catapult SDK and Documentation

To download the Catapult SDK and related documentation, log into your assigned Catapult machine in TACC and navigate to our public network folder here: \\cat-data\Publish

Inside, you'll see a folder hierarchy containing the various SDK releases and documentation. Due to licensing issues, we must distribute the Microsoft and Intel/Altera portions of the SDK separately. Navigate to the SDK release you wish to access (we recommend the most up-to-date version) and download the Microsoft and Altera/Intel Zip files. For example, for Catapult SDK v1.2, the Microsoft release is titled CatapultAcademic_MSFT_v1.2.zip, and the Altera/Intel IP portion is titled CatapultAcademic_Altera_v1.2.zip.

The Microsoft and Altera/Intel zip files already contain the full directory hierarchy. To install the SDK, simply unzip both to the E:\ root drive, which will unzip to E:\catapult\<version>. **Do not install the SDK to a different location since you may run into the Windows kernel 260-character maximum path limit.**



1.7 Getting Started with Windows Development

We recognize that many users are not used to developing in Windows and may prefer a GNU/Linux environment. Given the flexibility of Windows customization, it is possible to install many of the same GNU utilities and configure Windows PowerShell to behave in a similar way to Bash. Many people prefer to use Cygwin. For those who do not enjoy using Cygwin, we recommend the following:

1.7.1 Windows PowerShell

PowerShell is a Windows command line shell that is much more advanced than the basic Command Prompt (cmd). If not using Cygwin, we recommend that developers use PowerShell rather than the Command Prompt.

PowerShell is installed by default on all modern Windows operating systems. To launch PowerShell, hit the Windows key and type “powershell” (case insensitive). Windows PowerShell should appear: press Enter to run it. If you prefer, you may pin PowerShell to the task bar for ease of access (particularly if you wish to run PowerShell in Administrator mode). To do so, hit the Windows key and type “powershell”. Rather than pressing Enter, right-click on PowerShell and select “Pin to Taskbar”.

By default, auto-complete on Powershell (pressing Tab with a partial path) brings up the first valid path that matches the partial text. You may prefer a Linux-style autocomplete. This may be accomplished through the PSReadLine utility, which adds advanced features to the command line experience. Starting from PowerShell v5.0, PSReadLine is installed by default. If it is not, you may install it here: https://github.com/lzybkr/PSReadLine

The link above also contains the manual for PSReadLine. If you simply wish to change auto-complete, type:

```
Import-Module PSReadLine
```

```
Set-PSReadlineKeyHandler -Key Tab -Function Complete
```

1.7.2 GNU on Windows

GOW (GNU on Windows) is a lightweight alternative to Cygwin, installing over 100 open source UNIX applications compiled as native win32 binaries. You can download GOW from their website: https://github.com/bmatzelle/gow/wiki

2 Running "Hello World"

The Catapult machine image comes pre-loaded with the Catapult academic release. However, the release may be out of date, in which case you will need to manually install the updated release: refer to Section 1.6 Accessing the Catapult SDK and Documentation.

2.1 Catapult SDK Overview

The Catapult SDK release contains four folders. For more information, we suggest reading the Catapult User Guide, which is available in the Documents folder.

- Driver: The Catapult Windows drivers, FPGA programming utilities, and software header files and libraries.
- Roles: The Catapult user roles. Currently contains the golden role and several example roles demonstrating PCIe and DRAM. Add your roles here.
    - AcademicPCIeLoopback: Loopback test example demonstrating PCIe. Stores data from PCIe->FPGA into a queue, then sends the same data back out from FPGA->PCIe.
    - AcademicDRAMLoopback: Loopback test example demonstrating DRAM and PCIe. Loops data from FPGA -> PCIe -> DRAM -> PCIe -> FPGA.
    - AcademicGolden: Baseline "golden" image.
- Shells: The Catapult Academic shell supporting the Mt Granite board. Many of the shell components are encrypted to protect Microsoft IP. **Do not modify the shell!**
- Software: Software for interfacing with the user role.
    - LoopbackStressTest: stress test for measuring PCIe bandwidth. Demonstrates multiple ways to interface with the slot-based PCIe mechanism to achieve high bandwidth. The test is compatible with both AcademicPCIeLoopback and AcademicDRAMLoopback.

2.2 Setting up Intel Quartus II 15.1

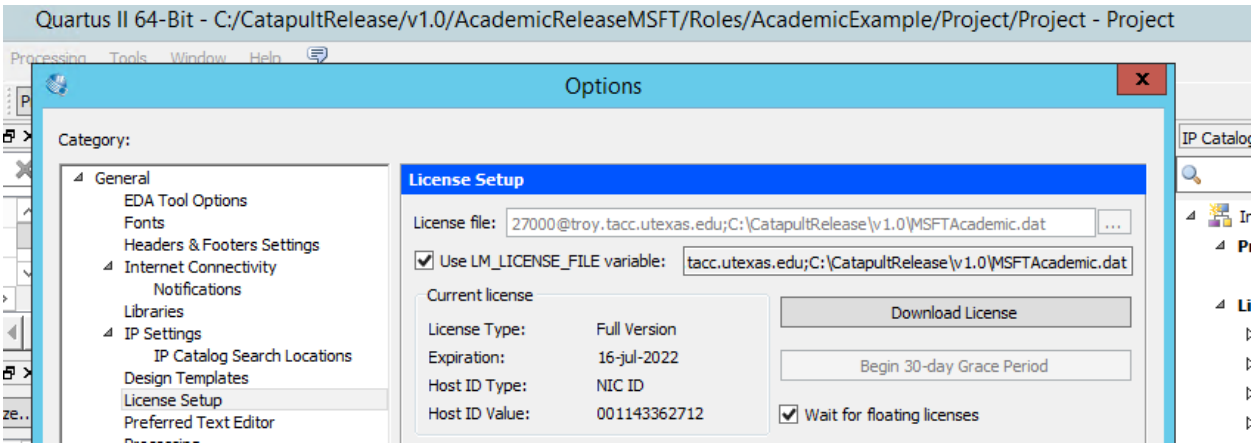
The Microsoft Catapult Academic SDK assumes that the designs will be synthesized using Intel Quartus II 15.1, which is pre-installed on the TACC machines. You will need to direct Quartus to the TACC license server. In addition, since many portions of the Catapult release are encrypted to protect Microsoft IP, Quartus also needs a decryption license to synthesize Catapult designs.

To set a system variable, hit Windows+X or go to the Control Panel, select “System”, select “Advanced System Settings”, select “Environment Variables”, select “New” under “System Variables”, and enter the following:

Variable name: LM_LICENSE_FILE

Variable value: 27000@license02.tacc.utexas.edu;E:\catapult\v1.1\MSFTAcademic.dat

This sets the Intel LM license variable to point to the TACC license server and the Catapult SDK decryption license. If you did not put MSFTAcademic.dat in the recommended location, change the above to correctly point to MSFTAcademic.dat. If Quartus has errors regarding the license file, go to the “Tools” menu and select “License Setup…” near the bottom. Make sure that “Use LM_LICENSE_FILE variable” is checked.



2.3 Synthesizing the Example FPGA Projects

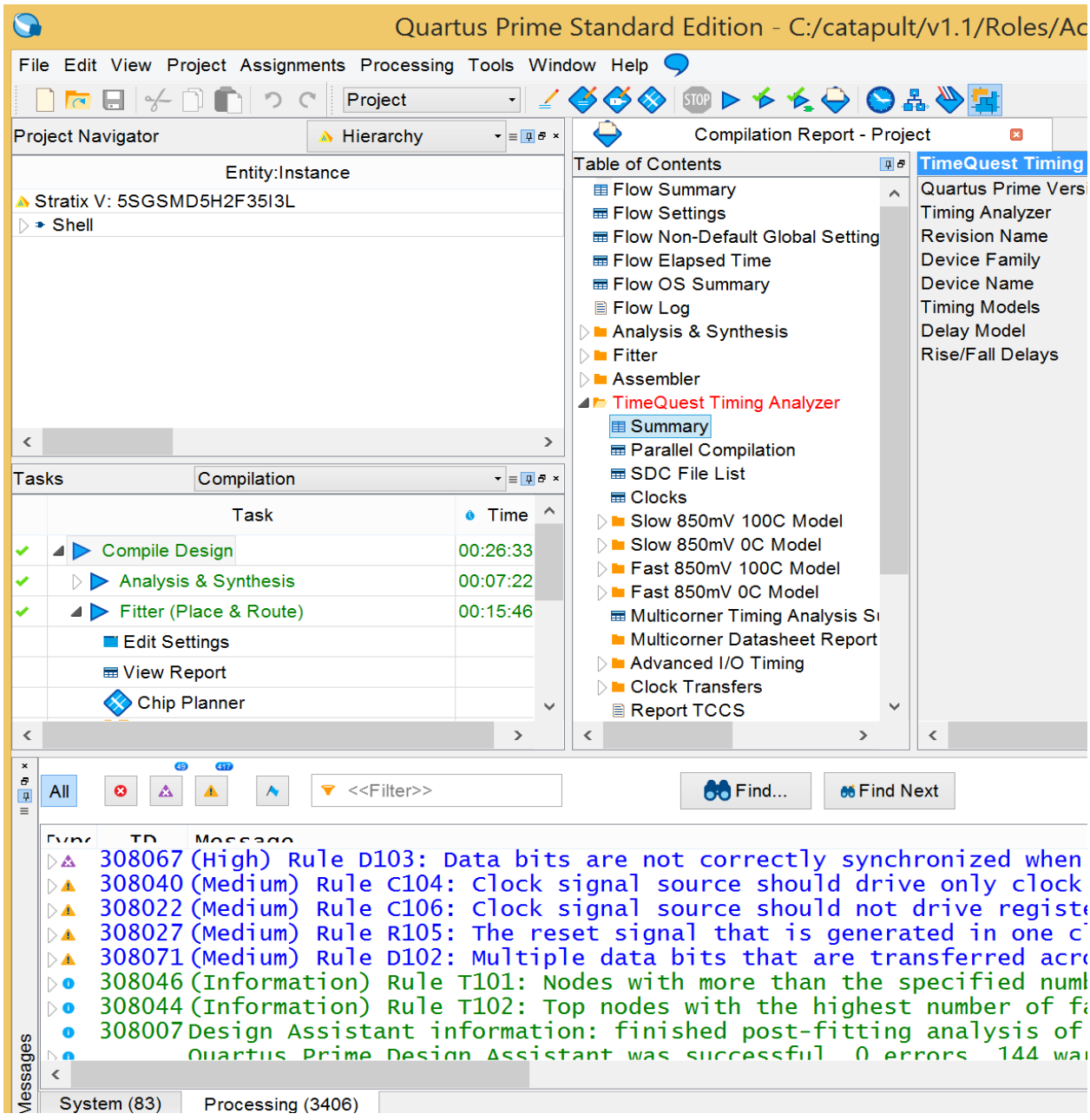
All projects can be synthesized using the same methodology. For this example, we will be synthesizing AcademicPCIeLoopback. Open the project file located in `E:\catapult\v1.1\Roles\AcademicPCIeLoopback\Project\Project.qpf`

After Quartus loads, double-click on “Compile Design” in the “Tasks” window to start the build process. Alternatively, you may go to the “Processing” menu and select “Start Compilation”. The build process should take around 30-40 minutes to complete. Since AcademicDRAMLoopback interfaces with DRAM, building this project will take much longer: around 1.5 to 2 hours to complete.

When the project finishes building, verify that there are no timing errors by expanding TimeQuest Timing Analyzer. The following should have black text rather than red:

- Slow 850mV 100C Model
- Slow 850mV 0C Model
- Fast 850mV 100C Model
- Fast 850mV 0C Model

You will see Unconstrained Paths in red with 2 illegal clocks. This is expected behavior and can be ignored.



2.4 Generating the Flash Memory Programming File

Catapult users must program the Mt Granite FPGA board through flash memory. To do so, you must convert the generated SOF bitfile to the required RPD format for the flash memory programmer tool.

The standard RPD conversion process using Quartus through the GUI is complicated, so we have provided a Powershell script that automates the process for you.

In Powershell, navigate to `E:\catapult\v1.2\Roles`

To use the script, type:

```
.\genRpd.ps1 -inputSof <path to sof> -outputRpd <path to rpd>
```

For example, if generating an RPD file for AcademicPCIeLoopback:

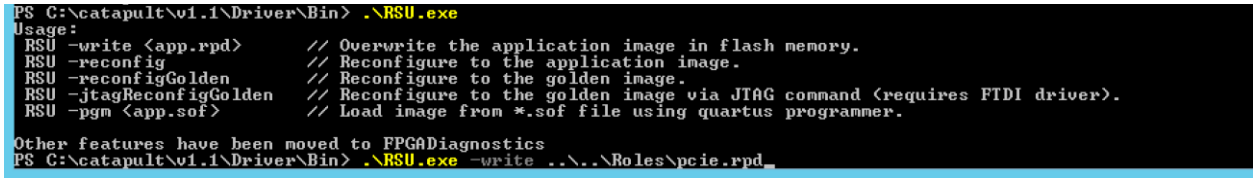
```
 .\genRpd.ps1 -inputSof
.\AcademicPCIeLoopback\Project\output_files\Project.sof -outputRpd pcie.rpd
```

The resulting `pcie.rpd` file is used to program the FPGA.

2.5 Programming the FPGA

In PowerShell with Administrator privilege, navigate to E:`\catapult\v1.2\Driver\Bin`

To program the flash memory, run `.\RSU.exe -write <path to RPD image file>`

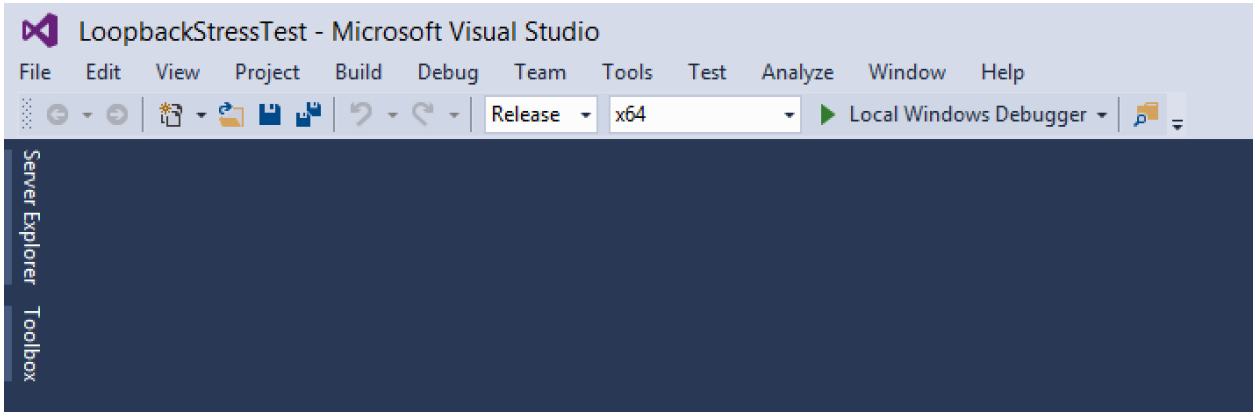


Programming the flash memory takes a few minutes. After this has completed, reconfigure the FPGA to the image on flash memory by running `.\RSU.exe -reconfig`

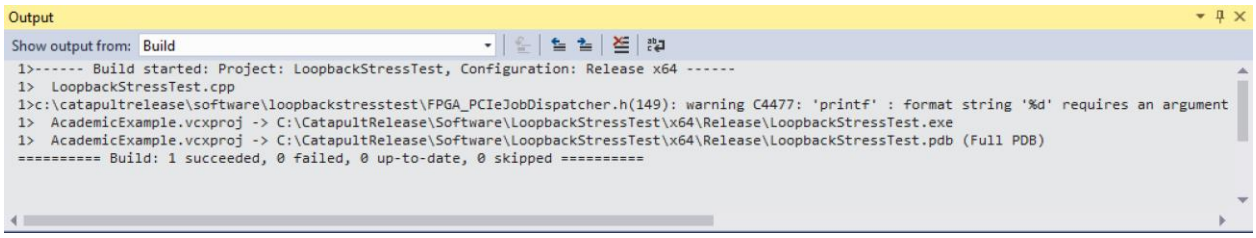
```
[hip-0]: Erasing/Writing/verifying sector 1f60000
[hip-0]: Erasing/Writing/verifying sector 1f70000
[hip-0]: Erasing/Writing/verifying sector 1f80000
[hip-0]: Erasing/Writing/verifying sector 1f90000
[hip-0]: Erasing/Writing/verifying sector 1fa0000
[hip-0]: Erasing/Writing/verifying sector 1fb0000
[hip-0]: Erasing/Writing/verifying sector 1fc0000
[hip-0]: Erasing/Writing/verifying sector 1fd0000
[hip-0]: Erasing/Writing/verifying sector 1fe0000
[hip-0]: Erasing/Writing/verifying sector 1ff0000
[hip-0]: Done at address 1ffffff
[hip-0]: Time 139.74 seconds
RSU total time: 139.79 seconds
PS C:\catapult\v1.1\Driver\Bin> .\RSU.exe -reconfig
Reconfiguring to the Application Image.
[hip-0]: Initializing PCIe HIP 0
Reconfig to app attempt #1
[hip-0]: ------------------------------------------------------------------------
[hip-0]: Before reconfiguration: <RSU Flash Slot:1>
[hip-0]: APP -> APP not supported, forcing a load of golden first (APP -> GOLDEN -> APP)
[hip-0]: Reconfiguring to flash slot 0 over flash, RSU 3.0 Detected
[hip-0]: Disabled AER on the upstream port
[hip-0]: RSU write to 0x4 = 0x0
[hip-0]: WARNING: RSU flash addr read back 0x10000
[hip-0]: PCIe link retrained to gen3x8
[hip-0]: Initializing PCIe HIP 0
[hip-0]: Reconfiguration Completed
[hip-0]: After reconfiguration: <RSU Flash Slot:0>
[hip-0]: Reconfiguring to flash slot 1 over flash, RSU 3.0 Detected
[hip-0]: Disabled AER on the upstream port
[hip-0]: RSU write to 0x4 = 0x10000
[hip-0]: PCIe link retrained to gen3x8
[hip-0]: Initializing PCIe HIP 0
[hip-0]: Reconfiguration Completed
[hip-0]: After reconfiguration: <RSU Flash Slot:1>
[hip-0]: RECONFIGURATION SUCCEEDED
RSU total time: 3.90 seconds
PS C:\catapult\v1.1\Driver\Bin> _
```

2.6 Compiling the LoopbackStressTest Software Program

Open the LoopbackStressTest project file in Microsoft Visual Studio 2015 located at `E:\catapult\v1.2\Software\LoopbackStressTest\LoopbackStressTest.vcxproj`. Make sure to select the build options "Release" and "x64" in the tool bar. Go to the Build menu and select "Build Solution", or hit Ctrl+Shift+B to generate the executable.



If the build succeeded, the Output window will display the following message:



2.7 Running the Hello World Application

By default, Visual Studio places the generated binary in x64\Release in the project directory. However, before running, the binary requires several dynamically linked libraries (*.dll) and ini files (*.ini) in the same directory as the binary. Copy all *.dll and *.ini files from:

```
E:\Catapult\v1.2\Driver\Bin
```

To:

```
E:\Catapult\v1.2\Software\LoopbackStressTest\x64\Release
```

Now, you're finally ready to run the application!

In PowerShell, navigate to the location of the generated binary, and type:

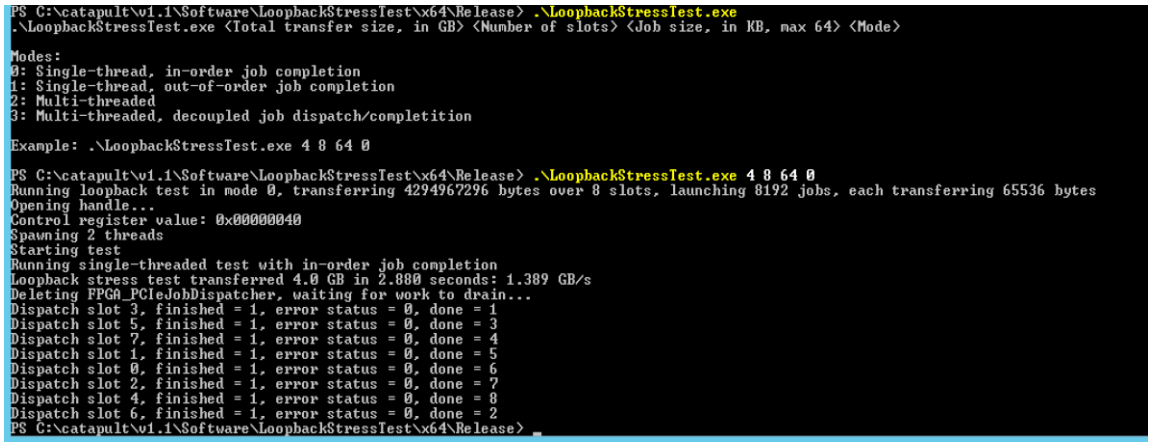
```
.\LoopbackStressTest.exe
```

This should show you a list of options. LoopbackStressTest is a simple program that demonstrates multiple ways to send data at high throughputs to the FPGA. Both AcademicPCIeLoopback and AcademicDRAMLoopback example roles return the same data back to the software.

To send 1GB of data over 8 slots, with each slot sending 64KB of data at a time using the simplest method which also checks the data, type:

```
.\LoopbackStressTest.exe 1 8 64 0
```

If the output is as follows, you have successfully run the LoopbackStressTest application. Congratulations!

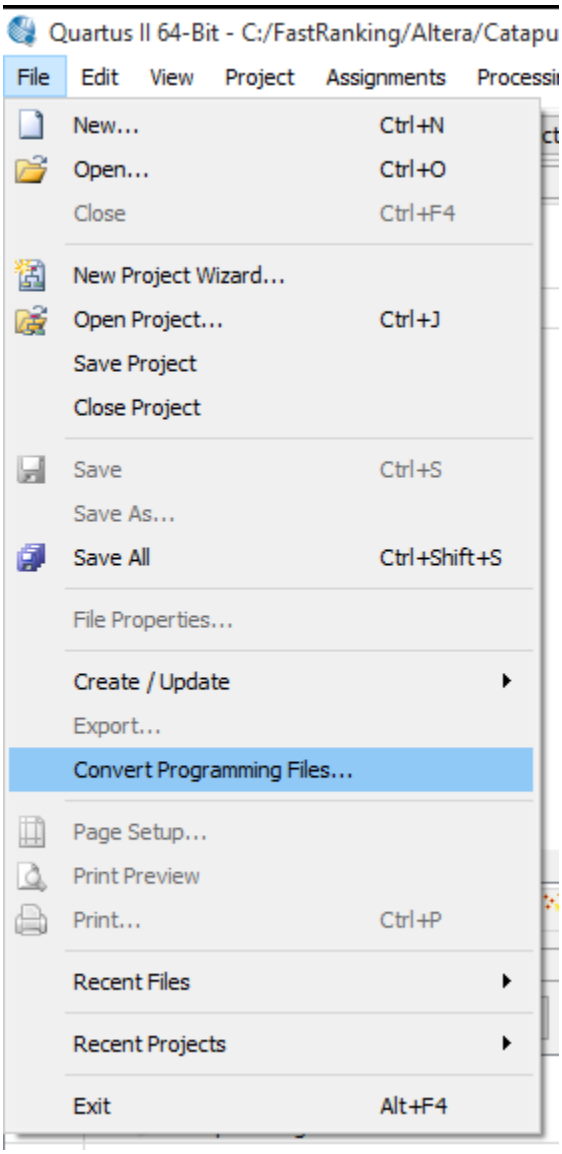

PS C:\catapult\v1.1\Software\LoopbackStressTest\x64\Release> .\LoopbackStressTest.exe
.\LoopbackStressTest.exe <Total transfer size, in GB> <Number of slots> <Job size, in KB, max 64> <Mode>
Modes:
0: Single-thread, in-order job completion
1: Single-thread, out-of-order job completion
2: Multi-threaded
3: Multi-threaded, decoupled job dispatch/completition
Example: .\LoopbackStressTest.exe 4 8 64 0
PS C:\catapult\v1.1\Software\LoopbackStressTest\x64\Release> .\LoopbackStressTest.exe 4 8 64 0
Running loopback test in mode 0, transferring 4294967296 bytes over 8 slots, launching 8192 jobs, each transferring 65536 bytes
Opening handle...
Control register value: 0x00000040
Spawning 2 threads
Starting test
Running single-threaded test with in-order job completion
Loopback stress test transferred 4.0 GB in 2.880 seconds: 1.389 GB/s
Deleting FPGA_PCIeJobDispatcher, waiting for work to drain...
Dispatch slot 3, finished = 1, error status = 0, done = 1
Dispatch slot 5, finished = 1, error status = 0, done = 3
Dispatch slot 7, finished = 1, error status = 0, done = 4
Dispatch slot 1, finished = 1, error status = 0, done = 5
Dispatch slot 0, finished = 1, error status = 0, done = 6
Dispatch slot 2, finished = 1, error status = 0, done = 7
Dispatch slot 4, finished = 1, error status = 0, done = 8
Dispatch slot 6, finished = 1, error status = 0, done = 2
PS C:\catapult\v1.1\Software\LoopbackStressTest\x64\Release>

3 Appendix

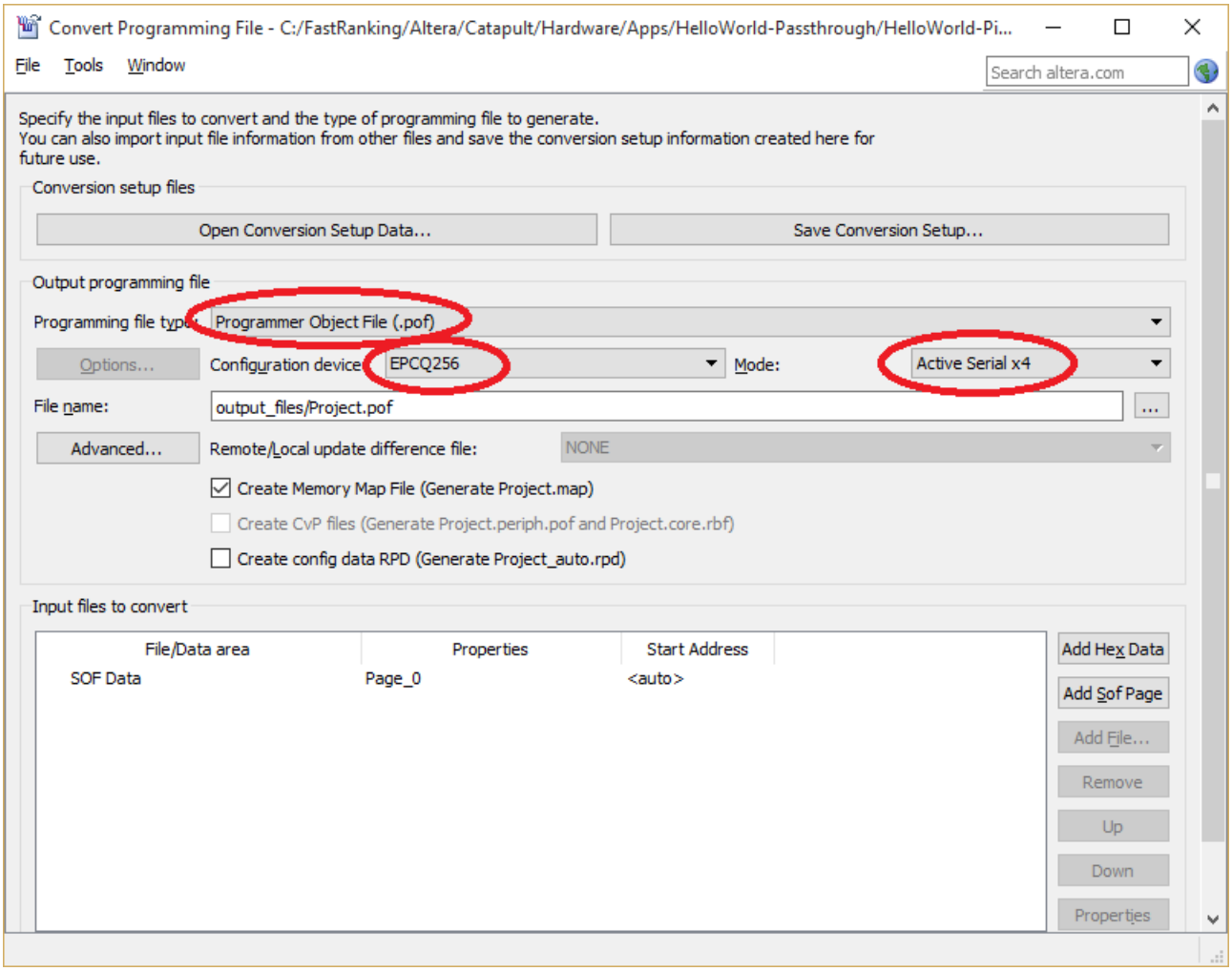
3.1 Manual RPD file generation

If for some reason the RPD generation script is not working, you may need to manually generate the RPD file instead.

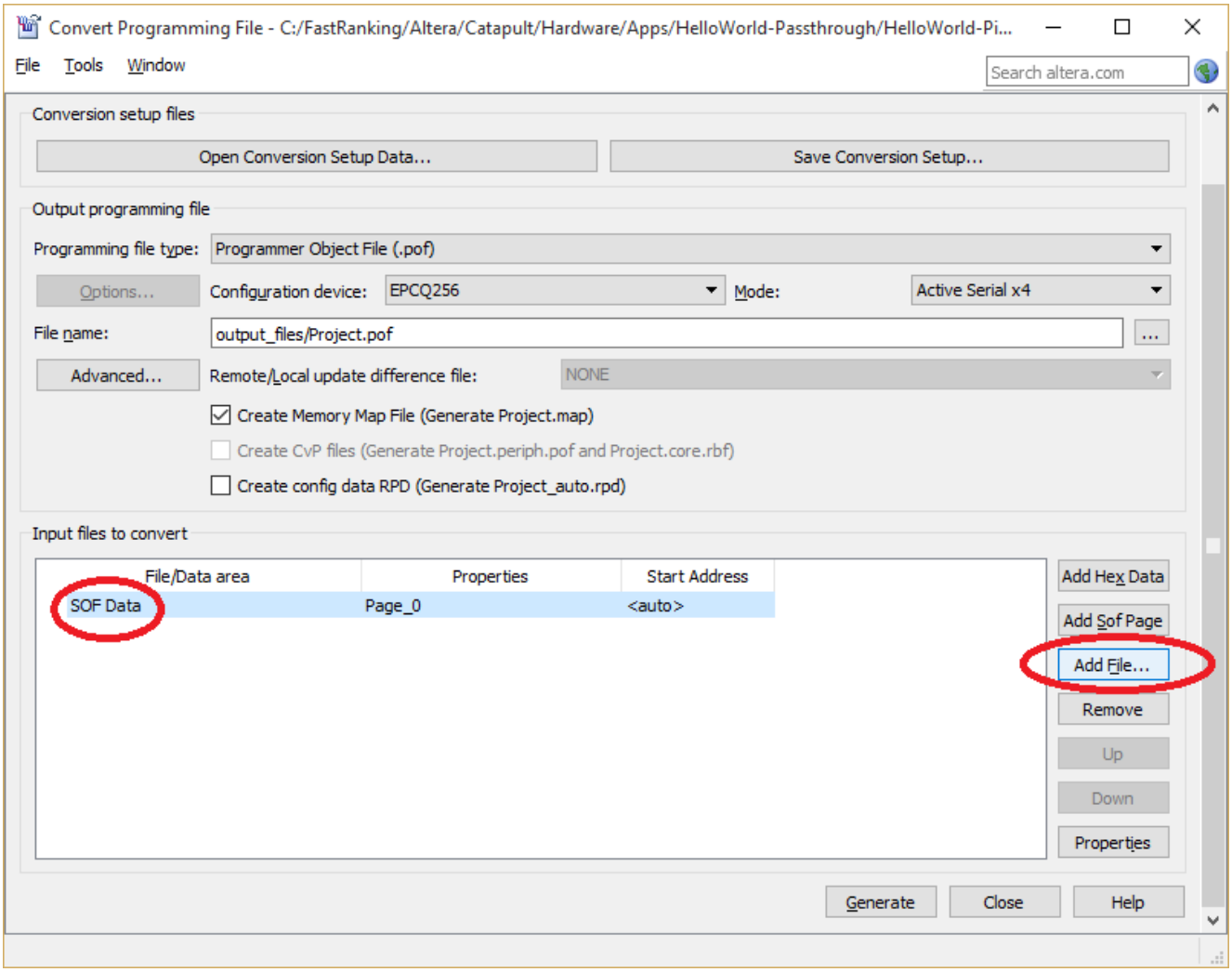
1. Open convert programming file GUI within Quartus



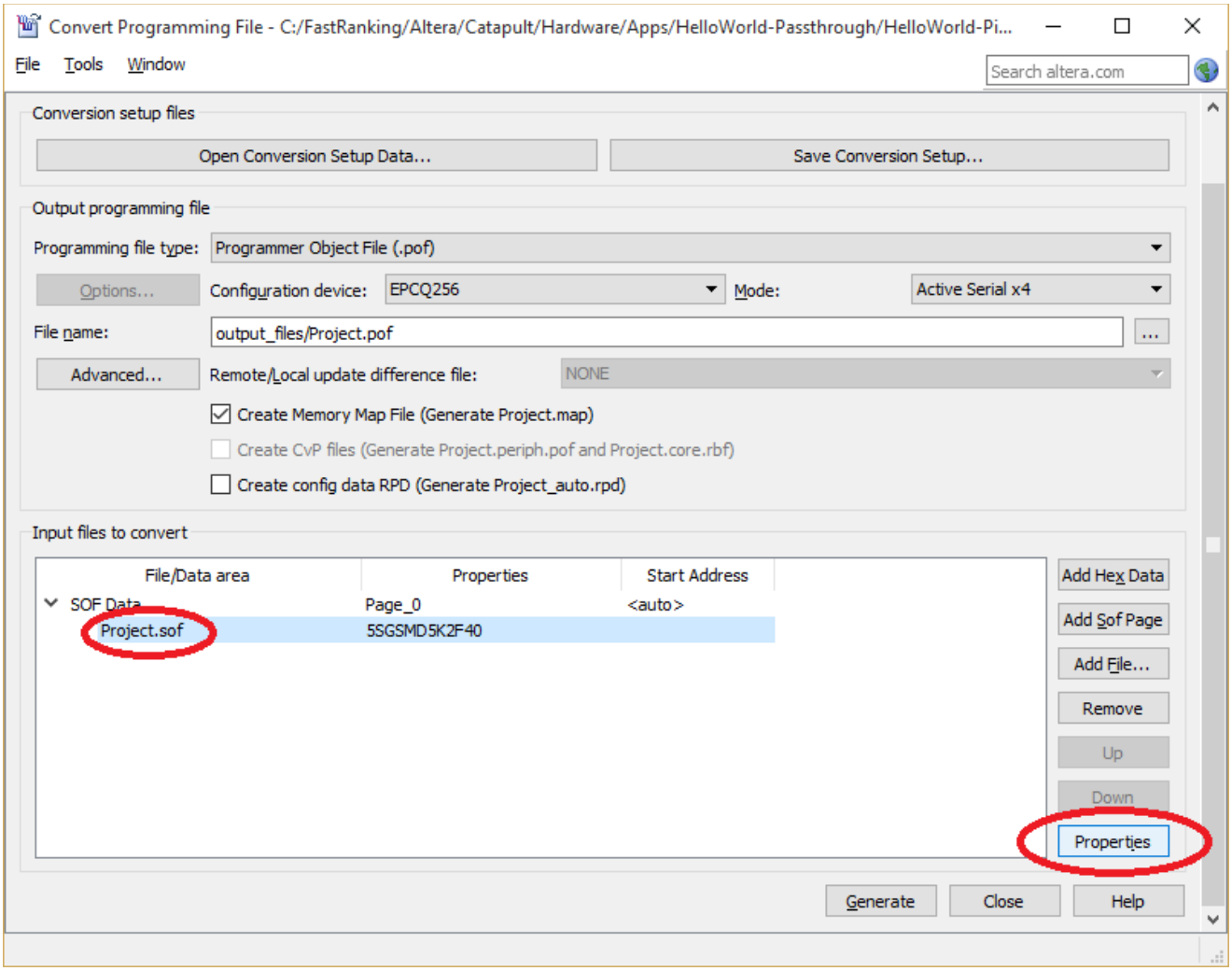
2. Ensure these options are used, choose your output file which will be of type .pof



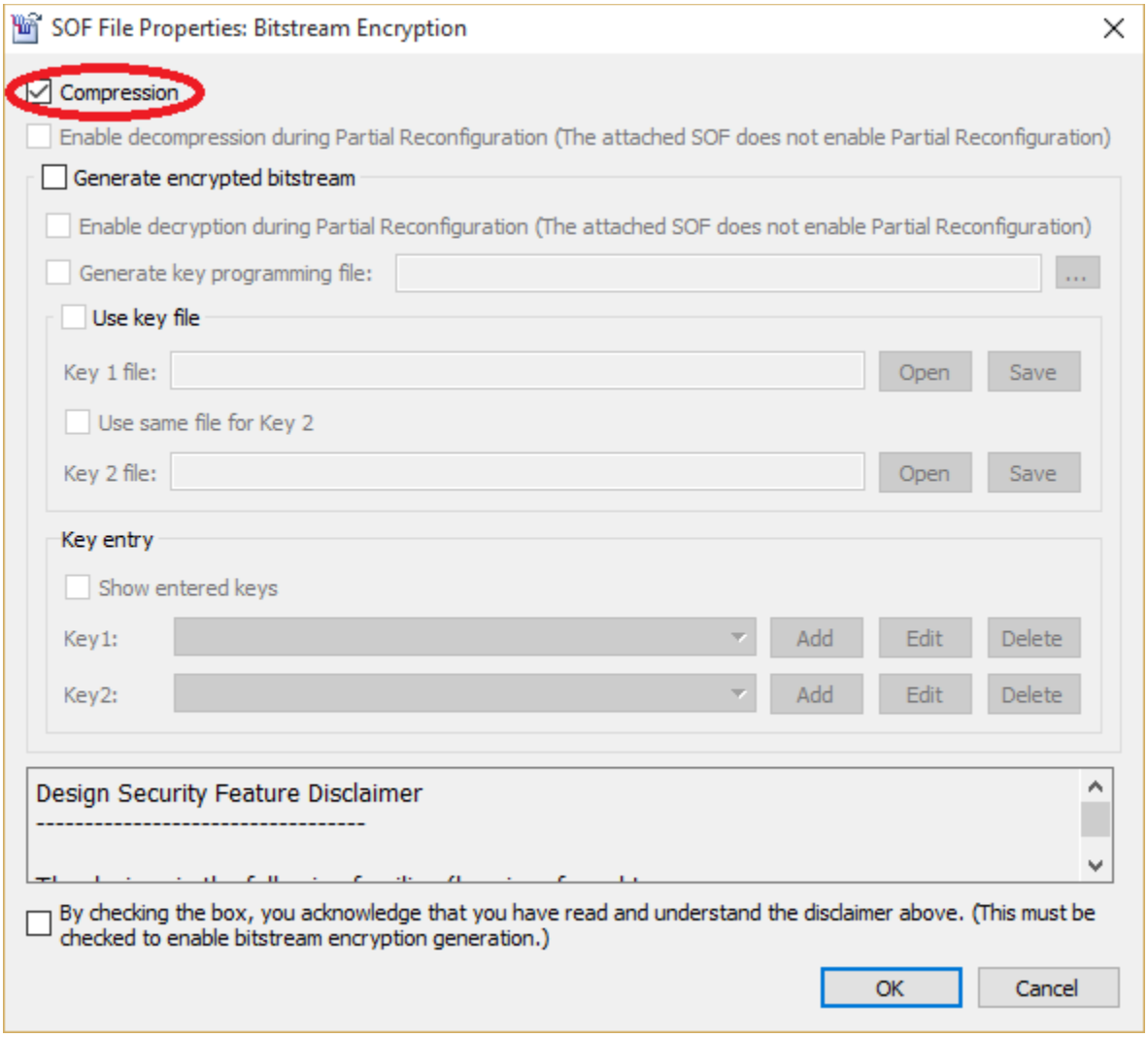
3. Click SOF Data, then the Add File button, use GUI to select the generated .sof file, located at \Project\output_files\Project.sof



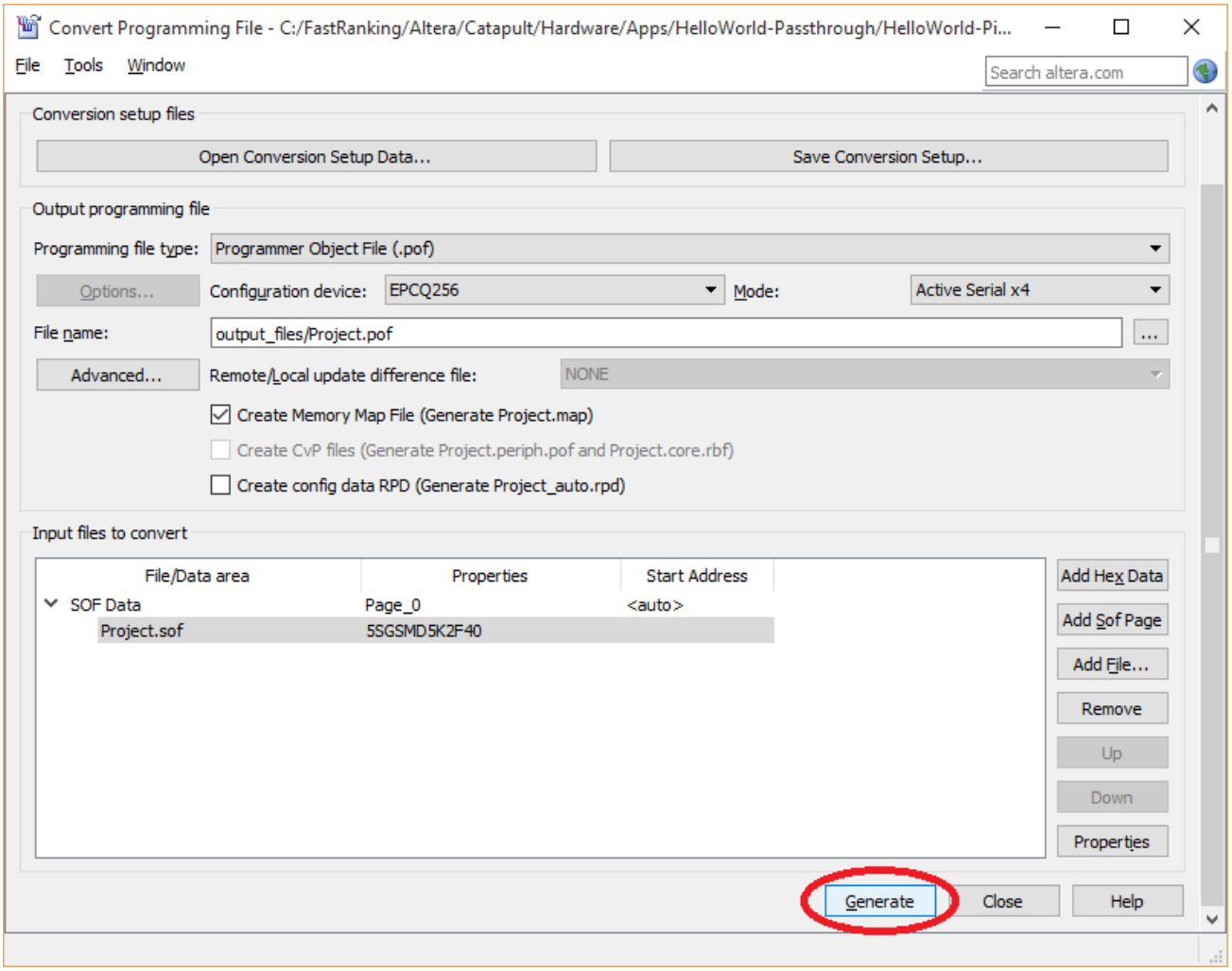
4. First click on the .sof file that was just added, then click the Properties button



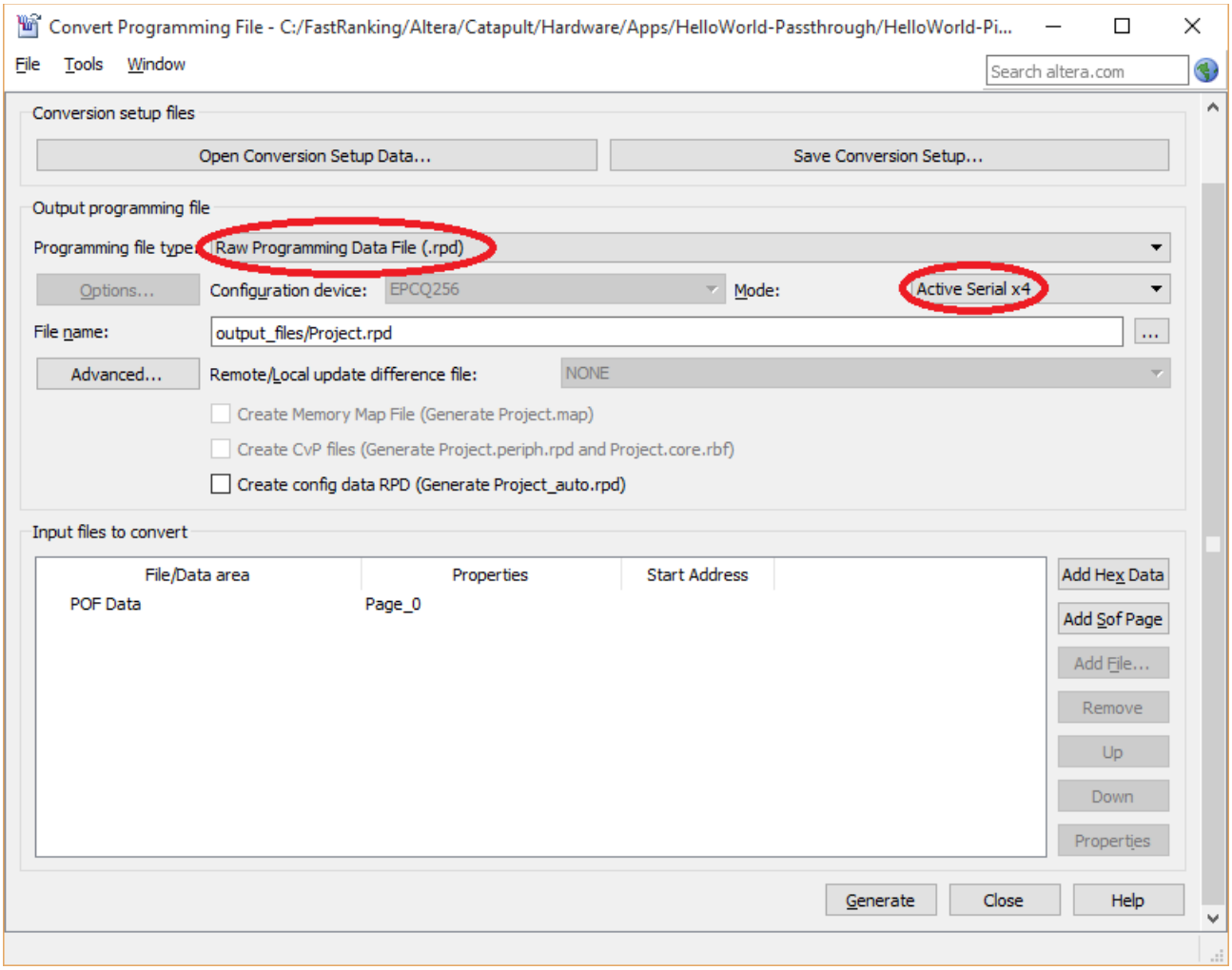
5. Use compression, then click the OK button at the bottom



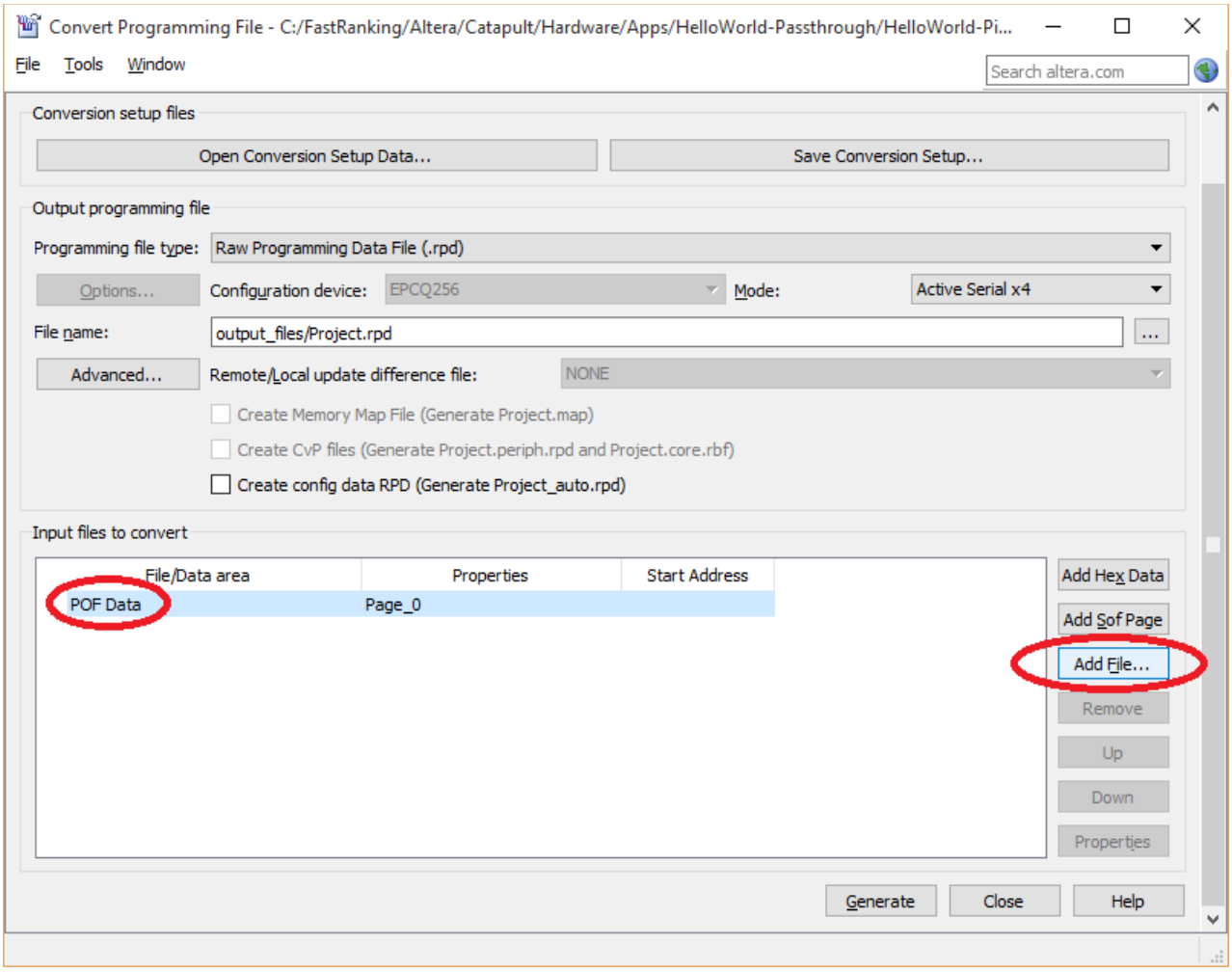
6. Click the Generate button – this produces the .pof. Overwrite the existing file or choose another file name.



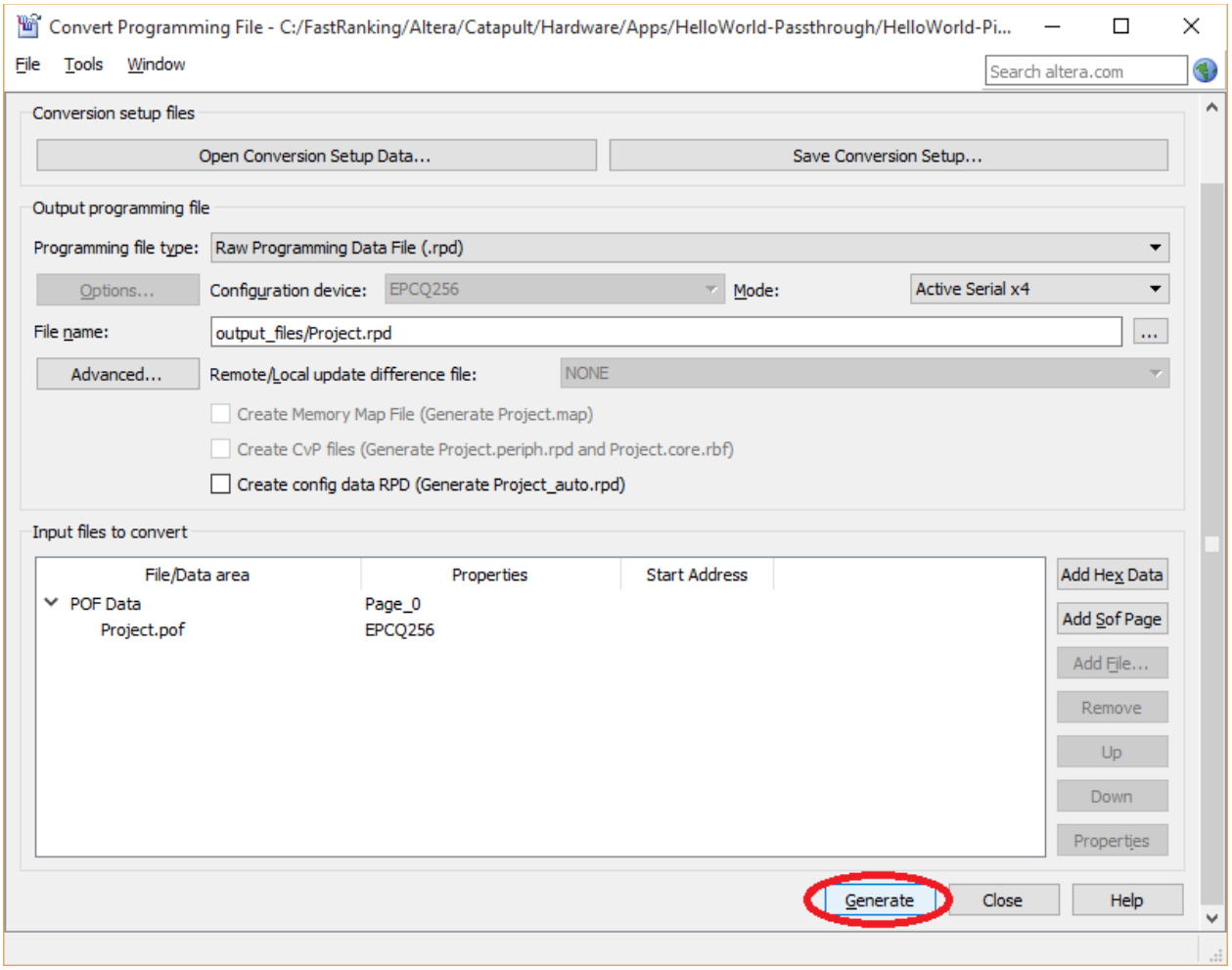
7. Now we convert the .pof to .rpd. Start by setting the options and choose your output file of type .rpd.



8. Click on POF Data, then click on Add File. Use the GUI to select the .pof file that was just created.



9. Generate


Convert Programming File - C:/FastRanking/Altera/Catapult/Hardware/Apps/HelloWorld-Passthrough/HelloWorld-Pi...
File
Tools
Window
Search altera.com
Conversion setup files
Open Conversion Setup Data...
Save Conversion Setup...
Output programming file
Programming file type:
Raw Programming Data File (.rpd)
Options...
Configuration device:
EPCQ256
Mode:
Active Serial x4
File name:
output_files/Project.rpd
Advanced...
Remote/Local update difference file:
NONE
Create Memory Map File (Generate Project.map)
Create CvP files (Generate Project.periph.rpd and Project.core.rbf)
Create config data RPD (Generate Project_auto.rpd)
Input files to convert
File/Data area
Properties
Start Address
POF Data
Page_0
Project.pof
EPCQ256
Add Hex Data
Add Sof Page
Add File...
Remove
Up
Down
Properties
Generate
Close
Help