Generate jic configuration file

To configure the flash memory of some modules with Intel FPGA(/SoCs from Trenz Electronic, a .jic file is needed. This file is built from the .sof file from the Quartus project and optionally the .hex file from the software application.
This is a short tutorial on how to create the .jic file.

More detailed instructions can be found directly on the Intel website:

- Nios II: AN 730: Nios II Processor Booting From Altera Serial Flash (EPCQ)
- Nios V: Nios V Processor Configuration and Booting Solutions

Table of contents

Prepare file conversion

Generate a .jic file only from the .sof file (without .hex file)

1. Compile the quartus project to generate a .sof file
2. Go to Convert .sof with .hex file (optional) to .jic file and ignore the steps to add the .hex file

Generate a .jic file from the .sof file with .hex file

Prepare Quartus Project:

1. Compile the quartus project to generate a .sof file
2. Make following changes in platform desginer:
    a. *Serial Flash Controller II Intel FPGA IP* or *Generic QUAD SPI Controller II Intel FPGA IP* (depending on the configuration flash memory) is needed to load the software application from the configuration flash memory
    b. Nios II:
        i. set *Nios II Processor Vectors Reset vector memory* to the selected flash memory controller
        ii. set *Nios II Processor Vectors Reset vector offset* to an address after the .sof file (minimum address: *start address* + *file size* of the .sof file)
    c. Nios V:
        i. set *Nios V Processor Vectors Reset Agent* to the selected flash memory controller
        ii. set *Nios V Processor Vectors Reset Offset* to an address after the .sof file (minimum address: *start address* + *file size* of the .sof file)
3. Recompile the whole quartus project

Prepare Nios II software project:

1. Open the software application in *Nios II SBT for Eclipse*
2. Recompile the software project
3. Right-click on project folder and select *Make Targets -> Build*
4. In the opened *Make Targets* window select *mem_init_generate* and click *Build*
    a. the .hex file is created in the *mem_init* folder in the software project directory

Prepare Nios V software project:

1. Recompile the software project
2. Open the Nios V Command Shell
3. Run following commands to generate the .hex file

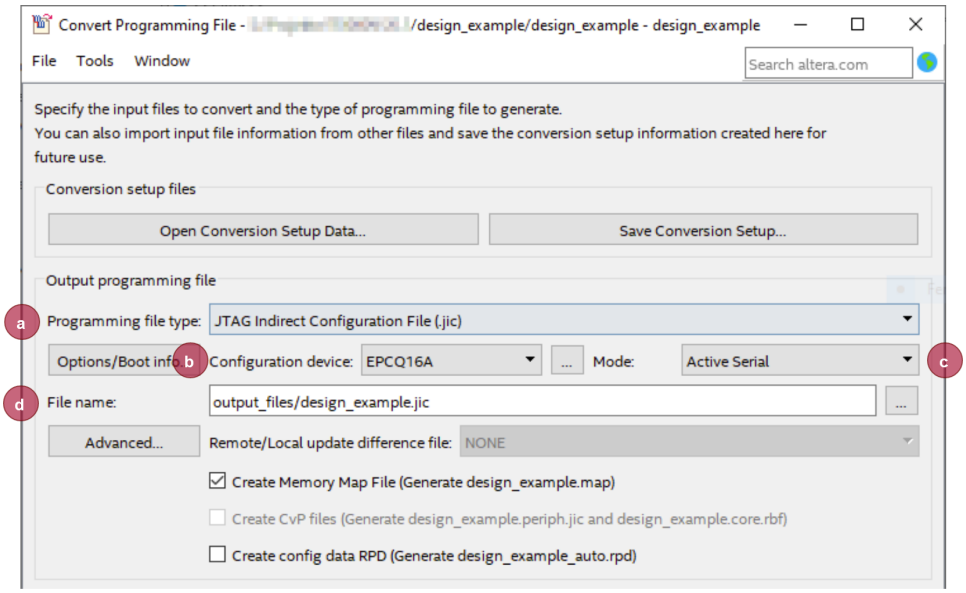
```
elf2flash --input path/to/sw_example.elf --output path/to/sw_example.
srec \
--reset  <flash memory controller base address + CPU reset offset> \
--base <flash memory controller base address> --end <flash memory
controller end address> \
--boot <quartus_installation_path>/niosv/components/bootloader
/niosv_bootloader.srec //check directory for correct bootloader file
name

riscv32-unknown-elf-objcopy --input-target srec --output-target ihex
path/to/sw_example.srec path/to/sw_example.hex
```

Convert .sof with .hex file (optional) to .jic file

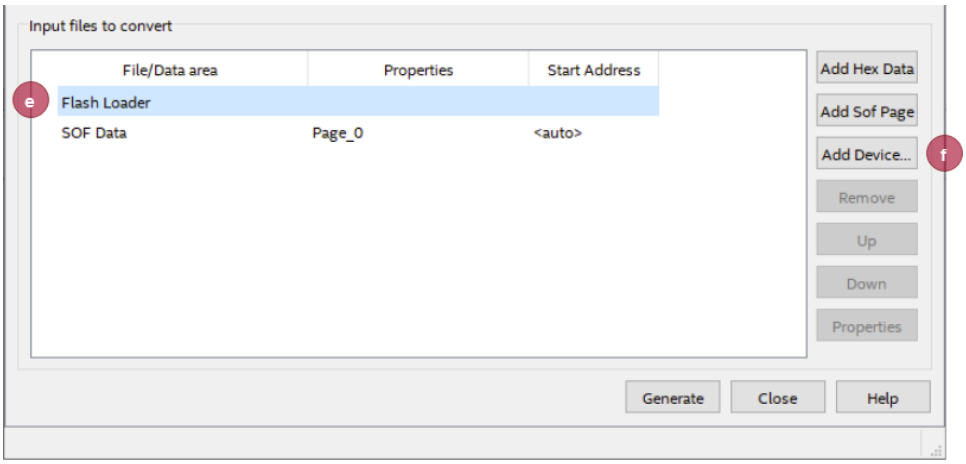
Do following steps to convert the .sof + .hex file to a .jic file:

1. Select *File Convert Programming files* from the quartus top menu
2. Make the following settings in the *Convert Programming File* window

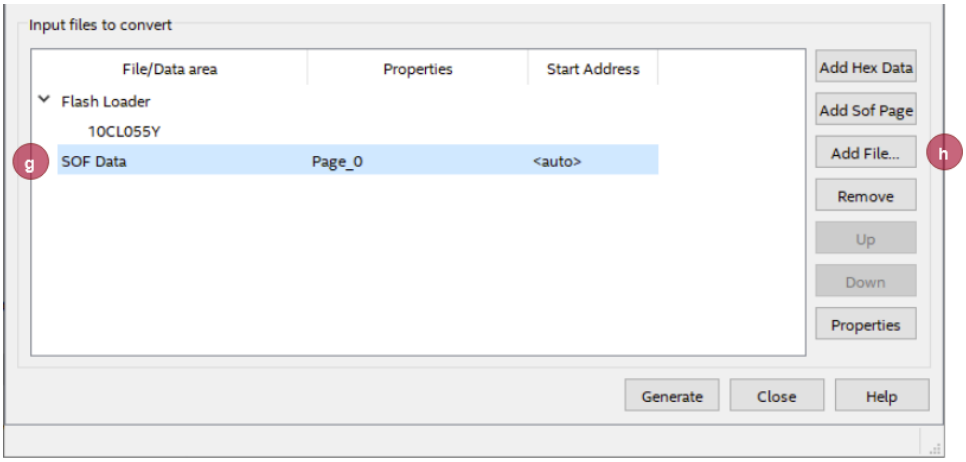


a. Programming file type: JTAG Indirect Configuration File (.jic)
b. Configuration device: depending on the configuration flash memory used on the module
   (if the memory device is not listed click "..." button next to the dropdown menu, select correct Device family and in the Configuration Device section the correct memory device)
c. Mode: Active Serial

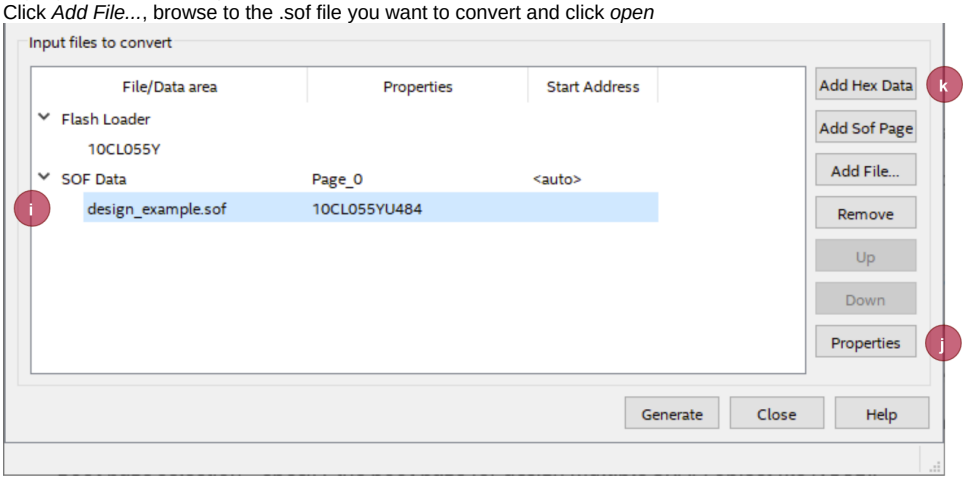
d. File name: specify the target directory and the output file name



e. Highlight *Flash Loader* in *Input files to convert* window
f. Click *Add Device...*, select correct Device family and Device name and click *OK*

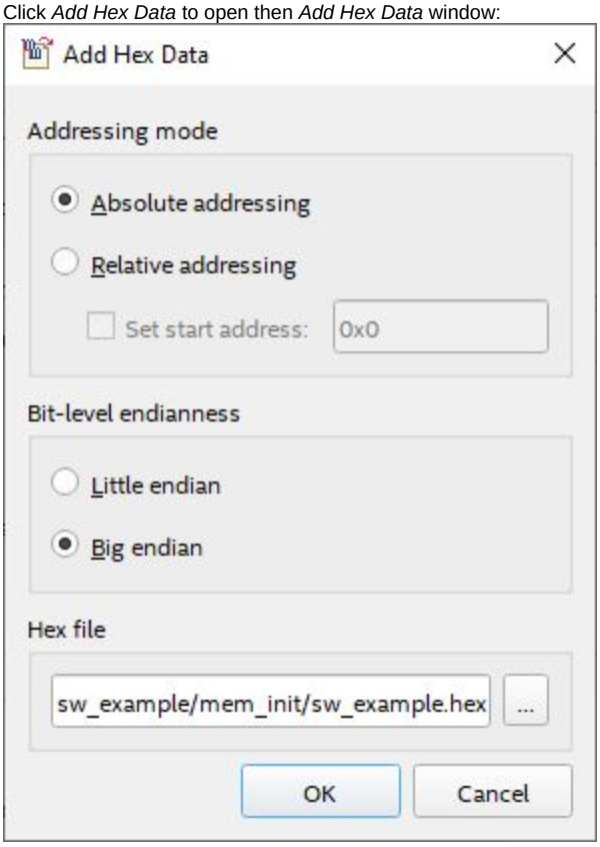


g. Highlight *SOF Data* in *Input files to convert* window
h. Click *Add File...*, browse to the .sof file you want to convert and click *open*



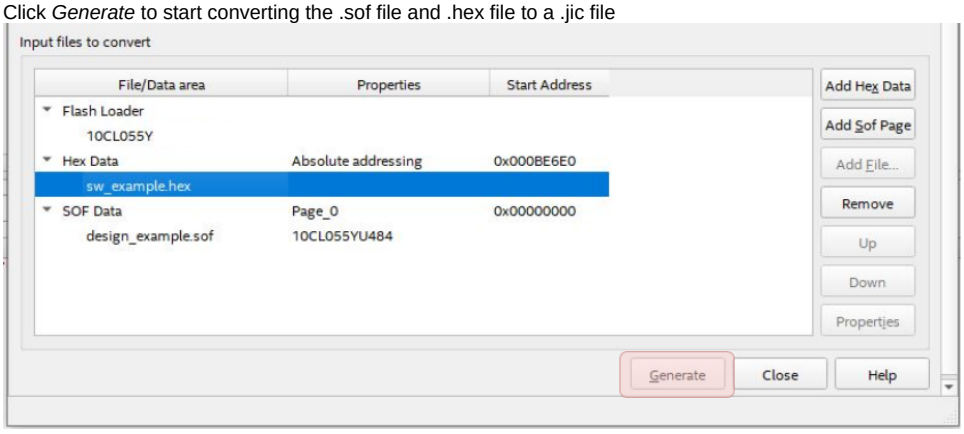
i. Highlight selected .sof file
j. Click *Properties*, enable *Compression* and click *OK*

k. Click *Add Hex Data* to open then *Add Hex Data* window:



i. set *Addressing mode* to *Absolute addressing*
ii. set *Bit-level endianness* to *Big endian*
iii. click "...", browse to the .hex file location and click *open*
iv. click *OK* to add the .hex file

3. Click *Generate* to start converting the .sof file and .hex file to a .jic file



You can use the Quartus Programmer to program the flash configuration device with the .jic file.

References

- AN 736: Nios II Processor Booting From Altera Serial Flash (EPCQ)

- Intel Quartus Prime Pro Edition User Guide: Programmer
- Intel Quartus Prime Standard Edition User Guide: Programmer