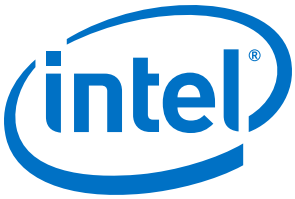


AN 860: Using Intel® Arria® 10 SoC FPGA Early I/O Release

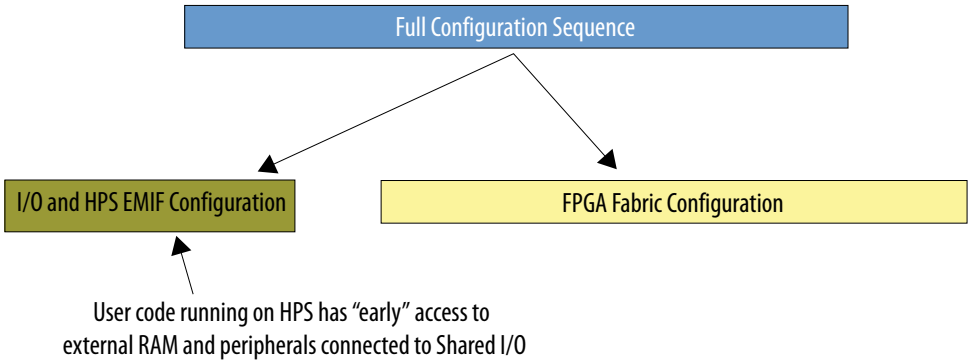
Subscribe
Send Feedback

AN-860 | 2020.10.22
Latest document on the web: PDF | HTML

Contents

1. Introduction

The Intel® Arria® 10 SoC FPGA device supports Early I/O Release. This feature splits the FPGA configuration sequence into two parts. The first part configures the FPGA I/O, the Shared I/O and enables the HPS External Memory Interface (EMIF) if present. The second part of the sequence configures the FPGA fabric.



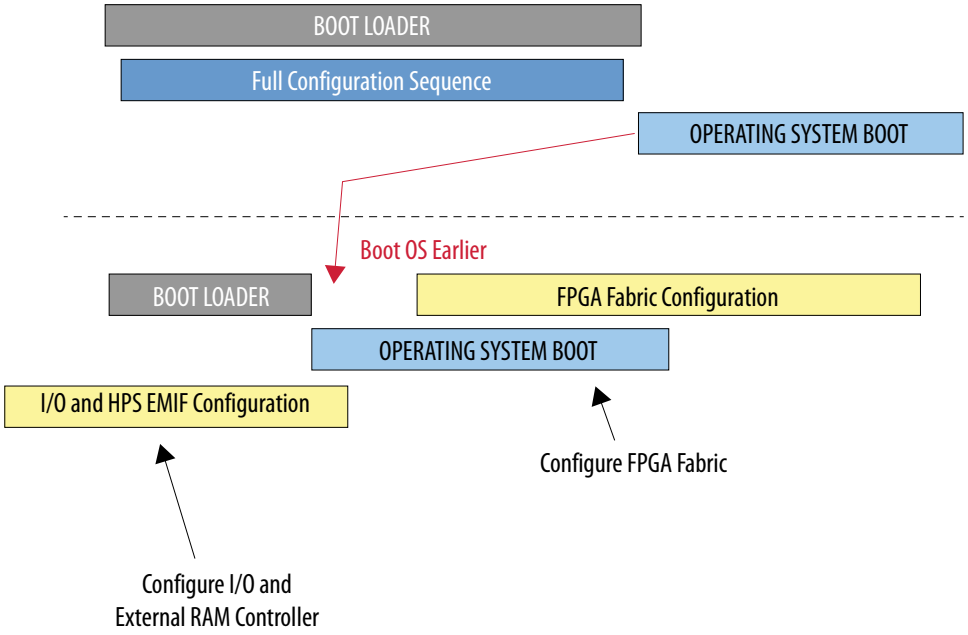
Splitting the configuration sequence provides HPS access to Shared I/O and EMIF before the FPGA fabric is configured. This allows more flexibility for designs that need faster boot times or alternate boot sources.

Enabling Early I/O Release in an Intel Arria 10 SoC FPGA design is optional and adds some pin and usage restrictions. Instructions for enabling and debugging the Early I/O Release feature and usage restrictions are provided in this document.

1.1. Early I/O Release Use Cases

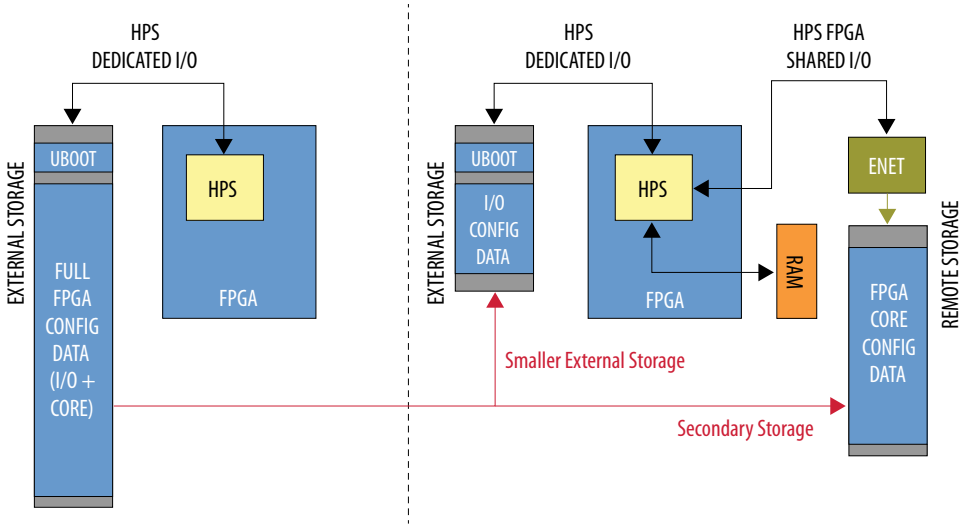
There are a few different reasons why you might choose to enable the Early I/O Release feature of the Intel Arria 10 SoC FPGA device. The typical reason is to speed up system boot and configuration time. By gaining early access to a large pool of system RAM connected the HPS EMIF interface, the boot software can more efficiently load the bulk of the FPGA configuration image from mass storage. Restricting boot code to on-chip RAM typically impedes bulk transfers because of the limited code and buffer space.

Figure 1. Early I/O Release



Some designers might want to boot the HPS operating system (OS) immediately. This action can be accomplished by postponing FPGA fabric configuration until the OS has booted. Booting the OS quickly is achieved by using the boot loader to only configure the FPGA I/O, Shared I/O and external memory before FPGA fabric is configured, then allowing the OS to boot and configure the FPGA fabric.

Figure 2. Early I/O Release allows FPGA fabric image to be accessed through a Shared I/O Interface



Another reason to enable the Early I/O Release feature is to gain boot software access to an interface connected to Shared I/O, such as a secondary mass storage device or network connection. This secondary access allows the primary storage connected to the HPS dedicated I/O pins to be smaller and possibly even write protected.

Related Information

- Intel Arria 10 Hard Processor System Technical Reference Manual
  Refer to section "Functional Description of the HPS I/O" for more details about the types of I/O available to HPS.
- Intel Arria 10 Hard Processor System Technical Reference Manual
  Refer to section "Early I/O Release FPGA Configuration Flow Through HPS" for more details about Early I/O Release.

1.2. Early I/O Release Flow

When using Early I/O release as part of the FPGA design compilation process, two Raw Binary Format (**.rbf**) configuration files are generated: a peripheral **.rbf** file and a core **.rbf** file. Together, these configuration files contain the same data as a combined configuration **.rbf** file that is generated when the Early I/O Release feature is not used.

The peripheral **.rbf** file is loaded first and configures the FPGA I/O, Shared I/O and HPS EMIF. The core **.rbf** is loaded next and completes the FPGA configuration sequence by configuring the FPGA fabric.

After the peripheral **.rbf** is successfully loaded, the Intel Arria 10 SoC FPGA HPS EMIF pins are released and the interface begins calibration.

2. Design Restrictions

There are design restrictions when enabling the Intel Arria 10 SoC FPGA Early I/O Release feature. You must evaluate these restrictions during the design planning process. These restrictions should also be considered before you attempt to enable the feature for an existing design to avoid unnecessary design debug effort.

2.1. Peripheral and Core RBF are a Matched Pair

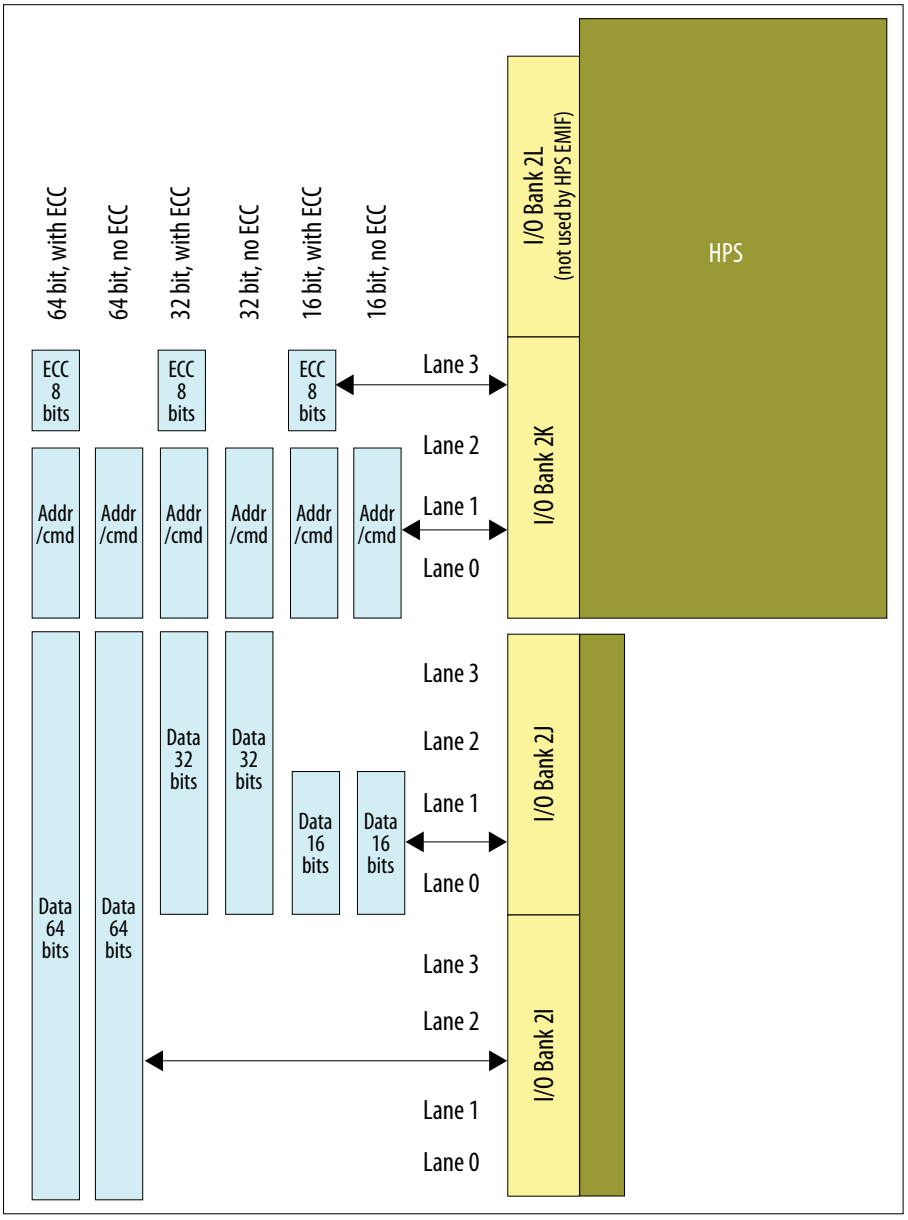
Although most of the signal and clock routing information is contained in the core **.rbf**, some of the routing information for paths between the FPGA core logic to the FPGA I/O pins is in the peripheral **.rbf**. Therefore, the peripheral **.rbf** and core **.rbf** files for a specific build of a design are a matched pair and must be not be mixed with **.rbf** files from another build.

2.2. EMIF Pin Usage Restrictions

The FPGA I/O bank lanes released for use by the EMIF are limited to the minimum required based on the external memory interface configuration. All pins used by the EMIF interface must be placed within these bank lanes. For example, no EMIF pins, including CLK or RZQ pins, should be placed in the 2I bank if the interface width is not 64-bits wide. Furthermore, no EMIF pins should be placed in bank 2J, lanes 2 and 3, if the interface is 16-bits wide.

Placing Intel Arria 10 SoC FPGA HPS EMIF pins in bank lanes that are not activated by the Early I/O Release feature will fail to calibrate. A design that violates this restriction may pass calibration when the Early I/O Release feature is disabled, and fail only when the feature is enabled. The Platform Designer tools flag this condition as a warning when building with the Early I/O Release feature disabled, and escalates this condition to an error when building with the feature enabled.

Figure 3. Intel Arria 10 SoC EMIF Pin Options

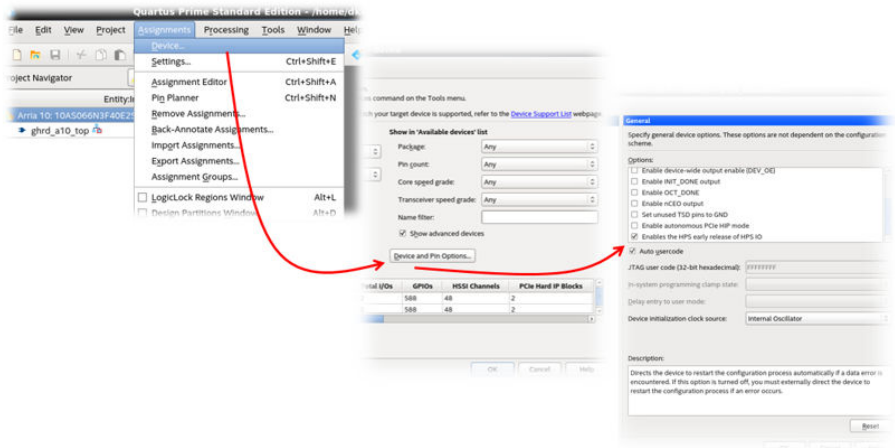


3. Enabling I/O Release

There are multiple steps to enabling the Intel Arria 10 SoC FPGA Early I/O Release feature in a design. The feature needs to be enabled in the Intel Quartus® Prime design project and in the Platform Designer design if using the Intel Arria 10 SoC HPS EMIF component. A command line option is also required when converting the design SRAM Object Format (**.sof**) configuration file to the required **.rbf** files. You must configure the boot loader to load either just the peripheral **.rbf**, or both the peripheral and core **.rbf** files (U-boot is provided below as an example).

Note: The operating system may also need to be configured to load core **.rbf** if not loaded as part of the boot loader.

3.1. Intel Quartus Prime Project Settings



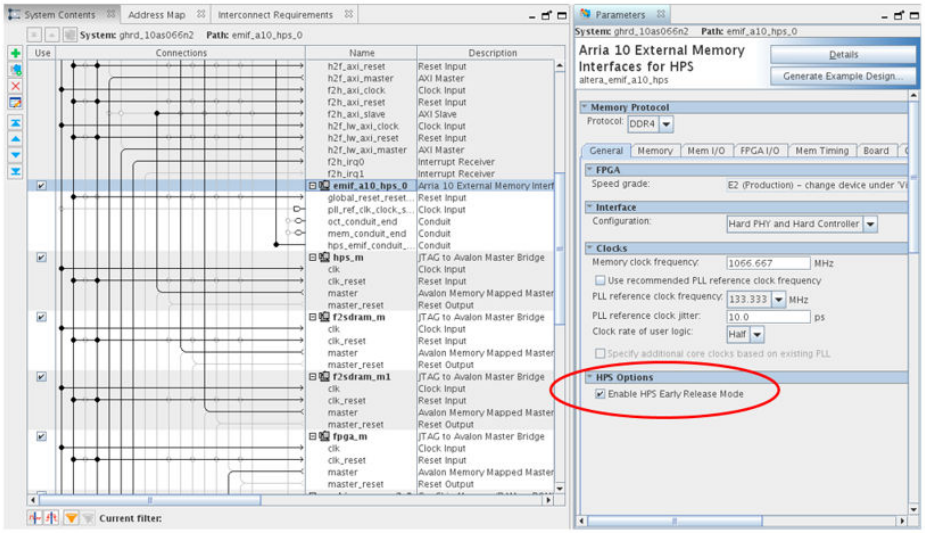
Follow the steps below

1. Enabling Early I/O Release in the Intel Quartus Prime project settings requires you to check one checkbox. To access the checkbox option, select the **Assignments ➤ Device** drop down menu item.
2. Click the **Device and Pin Options** box in the dialog window.
3. Once the **Device and Pin Options** dialog window is available, scroll the **Options** list to the bottom to find the **Enables the HPS early release of HPS IO** option and ensure that it is checked.
4. Selecting this option commands the FPGA compilation tools to set the **Early I/O Release option** bit in the assembled configuration stream.

3.2. Platform Designer Component Settings

Early I/O Release is enabled by setting an option in the Intel Arria 10 HPS EMIF component. This option is set using the Platform Designer design tool.

Open the **Parameters** window for the Intel Arria 10 HPS EMIF component and ensure that the **Enable HPS Early Release Mode** option in the **General** tab is checked.



Selecting this option parameter in the Intel Arria 10 HPS EMIF disables the component reset signal from the FPGA core logic. Setting this option is important to ensure that the EMIF does not accidentally reset when loading the FPGA core image.

When not using the Early I/O Release feature, the Intel Arria 10 HPS EMIF component is held in reset until the FPGA core is fully configured. However, when using the Early I/O Release feature, the EMIF is released from reset before the FPGA core is configured. This option disables any connection from the FPGA core logic to the EMIF reset.

3.3. Creating Split RBF Images

After the Intel Quartus Prime FPGA project has been successfully compiled and a **.sof** configuration file has been generated, a pair of split **.rbf** files must be generated using the **quartus_cpf** command line utility. Use the `-c` option to indicate that this is a conversion step, and use the `--hps` option to indicate that the **.rbf** should be split into peripheral and core files. The remaining arguments are the input **.sof** file and the output **.rbf** file base name.

```
$ quartus_cpf --convert --hps -o bitstream_compression=on
output_files/ghrd_10as066n2.sof output_files/ghrd_10as066n2.rbf
```

```
Info: ******************************************************************
Info: Running Quartus Prime Convert_programming_file
    Info: Version 18.0.1 Build 261 06/28/2018 SJ Pro Edition
    Info: Copyright (C) 2018  Intel Corporation. All rights reserved.
    Info: Your use of Intel 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 Intel Program License
    Info: Subscription Agreement, the Intel Quartus Prime License Agreement,
    Info: the Intel FPGA IP License Agreement, or other applicable license
    Info: agreement, including, without limitation, that your use is for
    Info: the sole purpose of programming logic devices manufactured by
    Info: Intel and sold by Intel or its authorized distributors.  Please
    Info: refer to the applicable agreement for further details.
    Info: Processing started: Sun Jul 15 23:05:21 2018
Info: Command: quartus_cpf --convert --hps -o bitstream_compression=on
output_files/ghrd_10as066n2.sof output_files/ghrd_10as066n2.rbf
Info: Quartus Prime Convert_programming_file was successful. 0 errors, 0
warnings
    Info: Peak virtual memory: 1297 megabytes
    Info: Processing ended: Sun Jul 15 23:05:45 2018
    Info: Elapsed time: 00:00:24
    Info: Total CPU time (on all processors): 00:00:58
$ ls output_files/*.rbf
output_files/ghrd_10as066n2.core.rbf  output_files/ghrd_10as066n2.periph.rbf
```

The command example listed above results in two new generated files: `ghrd_10as066n2.core.rbf` and `ghrd_10as066n2.periph.rbf`. These are the split **.rbf** files. The `ghrd_10as066n2.periph.rbf` must be loaded first, followed by the `ghrd_10as066n2.core.rbf` file.

3.4. Configuring U-Boot to use Split RBF Files

This section assumes that you are familiar with the `U-boot` build flow as defined by Intel Arria 10 SoC FPGA user documentation. For reference, Intel recommends you review the information provided in the Getting Started section of the RocketBoards.org support website and the Intel SoC FPGA Embedded Development Suite (EDS) User Guide. Links are provided at the end of this section.

After compiling a Intel Quartus Prime project that contains an Intel Arria 10 SoC FPGA HPS component, the generated meta data in the `hps_isw_handoff/hps.xml` file contains entries used by the Intel SoC EDS `bsp-editor` tool. The entry that indicates the Early I/O Release feature has been enabled is listed below.

```
<option_flags>
<config name='chosen.early-release-fpga-config' value='1' />
</option_flags>
```

The `bsp-editor` tool uses this entry determine if the `U-boot` device tree should enable the Early I/O Release feature.

If the value attribute is '0', then the `bsp-editor` tool generates a `U-boot devicetree.dts` file containing the following `chosen` block. `U-boot` loads the listed `cff-file` as a combined (peripheral + core) **.rbf** file.

```
chosen {cff-file = "socfpga.rbf";/* Bootloader setting: uboot.rbf_filename */};
```

If the value attribute is '1', then the `bsp-editor` tool adds the highlighted field to the `chosen` block as shown below. This block is generated when the Intel Quartus Prime project is compiled with the Early I/O Release feature enabled.

```
chosen {cff-file = "socfpga.rbf"; /* Bootloader setting: uboot.rbf_filename */};
```

The addition of the `early-release-fpga-config` field modifies the boot behavior of `U-boot`. `U-boot` loads the listed `cff-file` at startup as a peripheral **.rbf** instead of a combined **.rbf**. `U-boot` will print the following message on startup to indicate a successful peripheral **.rbf** file load.

```
FPGA: writing socfpga.rbf ...
FPGA: Early Release Succeeded.
```

The core **.rbf** file must be loaded later. The core **.rbf** file can be loaded using `U-boot` or postponed until the operating system has booted. If using `U-boot` to load the core **.rbf**, the `fpga loadfs` (or similar) command must be added to the `bootcmd` variable, entered at the `U-boot` command line, or executed via a `U-boot` script. The following is an example `U-boot` environment variable setup to load a core **.rbf** as part of the `bootcmd` mechanism.

```
bootcmd=run core_rbf_prog; run mmcload; run mmcboot
core_rbf_prog=fpga loadfs 0 mmc 0:1 ${rbfcoreimage} core
rbfcoreimage=ghrd_10as066n2.core.rbf
```

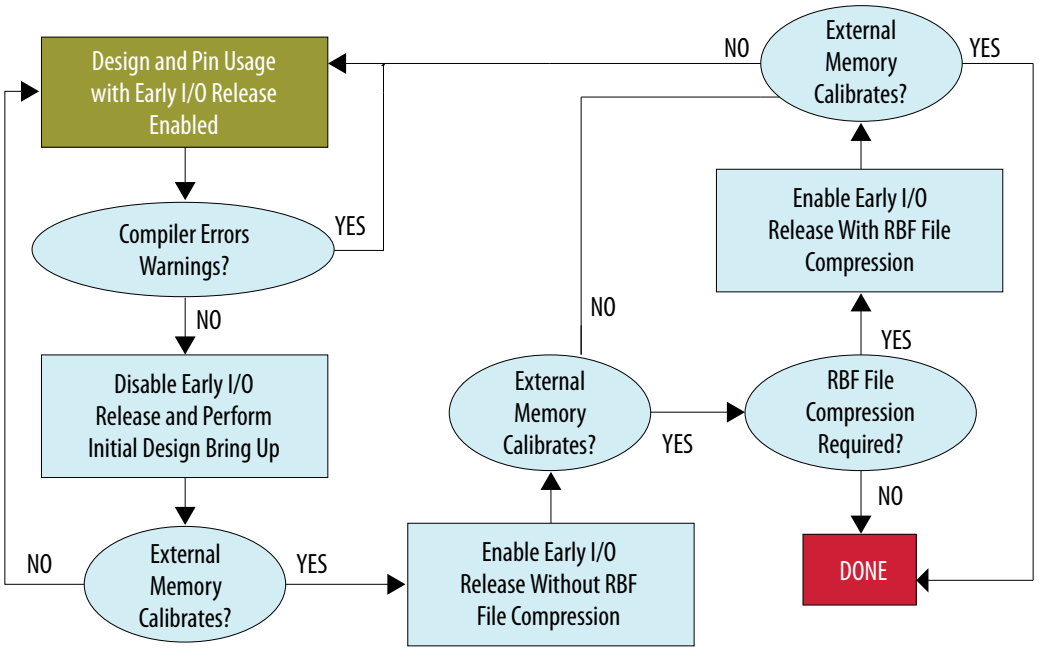
Related Information

- Getting Started RocketBoards.org
  Select **Board** to be "Altera Arria 10 SoC Board" and the latest available **Tool Version** then select the desired **Task**
- Intel SoC FPGA Embedded Development Suite User Guide
  Refer to "Boot Tools User Guide" chapter.

4. Debugging Early I/O Release

If a design uses the Intel Arria 10 SoC FPGA Early I/O Release feature, Intel recommends you enable the feature in the Intel Quartus Prime project and Platform Designer design file when the design process begins. The feature enables the Intel Quartus Prime compilation tools to check for pin usage restrictions.

When bringing up the design in hardware for the first time, temporarily disabling the Early I/O Release feature and following the normal combined **.rbf** configuration flow can help confirm that external memory is configured and calibrating properly. Intel recommends you avoid using **.rbf** file compression when initially testing a design.



After you confirm that configuration and external memory calibration is successful with Early I/O release disabled, Intel recommends you enable the Early I/O Release without **.rbf** file compression. Testing the Early I/O Release feature without **.rbf** file compression helps determine that your design is properly implementing the flow as described in this document.

If required by the design, you must use **.rbf** file compression with the Early I/O Release feature and validate using the previous bring up steps.

A. Document Revision History for Using Intel Arria 10 SoC FPGA Early I/O Release

Document Version	Changes
2020.10.22	Corrected the figure: *Intel Arria 10 SoC EMIF Pin Options*.
2018.10.11	Initial release.