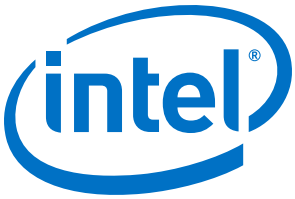


Error Message Register Unloader Intel FPGA IP Core User Guide

Updated for Intel® Quartus® Prime Design Suite: **18.0**

Subscribe
Send Feedback

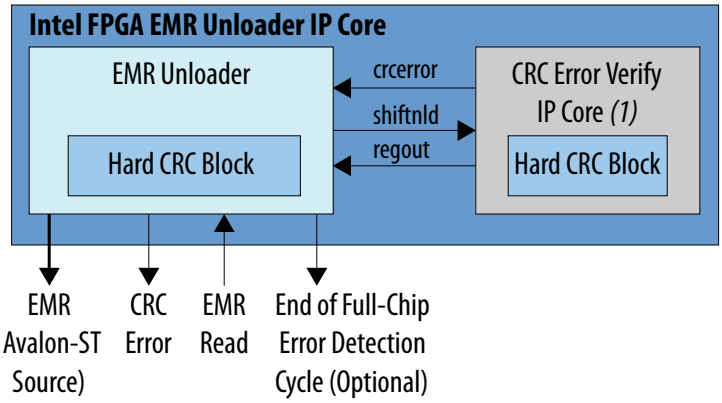
UG-01162 | 2018.05.23
Latest document on the web: **PDF** | **HTML**

Contents

1. Error Message Register Unloader Intel® FPGA IP Core User Guide

The Error Message Register Unloader Intel® FPGA IP core (altera_emr_unloader) reads and stores data from the hardened error detection circuitry in supported Intel FPGA devices. You can use the Error Message Register Unloader IP core's Avalon® Streaming (Avalon-ST) logic interface to read the device EMR.

Figure 1. Error Message Register Unloader Block Diagram



Note:

1. Only Stratix IV and Arria II devices use the CRC Error Verify IP core.

When hardware updates the EMR content, the IP core reads (or unloads) and de-serializes the EMR content, and allows other logic (such as the Intel FPGA Advanced SEU Detection IP core, Intel FPGA Fault Injection IP core, or user logic) to access the EMR content simultaneously.

1.1. Features

- Retrieves and stores the error register message contents for Intel FPGA devices
- Permits injection of an EMR register content value without changing CRAM bits
- Avalon (-ST) interface
- Easy instantiation with the parameter editor GUI
- Generates VHDL or Verilog HDL synthesis files

1.2. IP Core Device Support

The following devices support the Error Message Register Unloader IP core:

Table 1. IP Core Device Support

Design Software	IP Core Device Support
Intel Quartus® Prime Pro Edition	Intel Arria® 10 and Intel Cyclone® 10 GX devices
Intel Quartus Prime Standard Edition	Arria V, Arria II GX/GZ, Intel Arria 10, Cyclone V, Stratix® IV, and Stratix V devices

1.3. Resource Utilization and Performance

The Intel Quartus Prime software generates the following resource estimate for the Cyclone V (5CGXFC7C7F23C8) FPGA device. Results for other supported devices are similar.

Table 2. Error Message Register Unloader IP Core Device Resource Utilization

Device	ALMs	Logic Registers		M20K
		Primary	Secondary	
5CGXFC7C7F23C8	37	128	33	0

1.4. Functional Description

Supported Intel FPGA devices have an error message register that indicates the occurrence of a CRC error in the configuration RAM (CRAM). CRAM errors can occur because of a single event upset (SEU).

You can use the Error Message Register Unloader IP core's Avalon-ST logic interface to access the FPGA device EMR. For example, you can use the Error Message Register Unloader IP core with the Intel FPGA Fault Injection and Intel FPGA Advanced SEU Detection IP cores to access device EMR information.

The Error Message Register Unloader IP core monitors the device EMR. When hardware updates the EMR content, the IP core reads (or unloads) and de-serializes the EMR content. The IP core allows other logic (such as the Intel FPGA Advanced SEU Detection IP core, Intel FPGA Fault Injection IP core, or user logic) to access the EMR content simultaneously.

As shown in the #unique_1/unique_1_Connect_42_image_fbb_3mm_gs on page 3, the Error Message Register Unloader IP core instantiates the CRC Error Verify IP core for some devices.

Note: For more information on SEU support for your FPGA device, refer to the device handbook's SEU mitigation chapter.

1.4.1. Error Message Register

Some single event upset (SEU) FPGA devices contain built-in error detection circuitry to detect a flip in any of the device's CRAM bits due to a soft error.

The bit assignments for the device EMR vary by device family. For details on the EMR bits for your FPGA device family, refer to the device handbook's SEU mitigation chapter.

1.4.2. Signals

Table 3. Error Message Register Unloader Signals

Signal	Width	Direction	Description
clock	1	Input	Input clock signal.
reset	1	Input	Active-high logic reset signal.
emr_read	1	Input	Optional. This active-high signal initiates rereading the current EMR content. The EMR content updates when the device detects a new error. The EMR contains the error until a new error is detected, even if internal or external scrubbing corrects the error.
crcerror	1	Output	Indicates detection of a CRC error. This signal synchronizes to the clock port of the Error Message Register Unloader IP core.
crcerror_pin	1	Output	Connect this signal to the CRC_Error pin. This signal is synchronous to the device's internal oscillator.
crcerror_clk	1	Input	CRC Error Verify IP core input clock signal.
crcerror_reset	1	Input	CRC Error Verify IP core active-high logic reset signal.
emr[N-1:0]	46, 67, or 78	Output	This data port contains the device's error message register contents, as defined in the device handbook SEU mitigation chapter: • Intel Arria 10 and Intel Cyclone 10 GX devices have 78-bit EMRs • Stratix V, Arria V, and Cyclone V devices have 67-bit EMRs • Older devices have 46-bit EMRs The EMR output signals comply with the Avalon-ST interface definition. *N* is 46, 67, or 78.
emr_valid	1	Output	Active high when the emr signal contents are valid. This signal complies with the Avalon interface definition.
emr_error	1	Output	This signal is active high when the current EMR output transfer has an error and should be ignored. Typically, this signal indicates that the EMR input clock is too slow. This signal complies with the Avalon interface definition.
endoffullchip	1	Output	Optional output signal that indicates the end of each full-chip error detection cycle for the entire device. Intel Arria 10, Intel Cyclone 10 GX, Stratix V, Arria V, and Cyclone V devices only.

1.4.3. Timing

The Error Message Register Unloader IP core requires two clock cycles for the device error message circuitry, plus the following additional Error Message Register Unloader input clock cycles to unload EMR content: $N + 3$ where N is the emr signal width.

- 122 clock cycles for Intel Arria 10 and Intel Cyclone 10 GX devices
- 70 clock cycles for Stratix V, Arria V, and Cyclone V devices
- 49 clock cycles for Stratix IV and Arria II GZ/GX devices

1.4.3.1. IP Timing Behavior (Intel Arria 10 and Intel Cyclone 10 GX Devices)

The following waveforms show the Error Message Register Unloader IP core timing behavior for Intel Arria 10 and Intel Cyclone 10 GX devices.

Figure 2. emr_valid Signal for Correctable Errors (0 < Column-Based Type < 3'b111) Timing Diagram

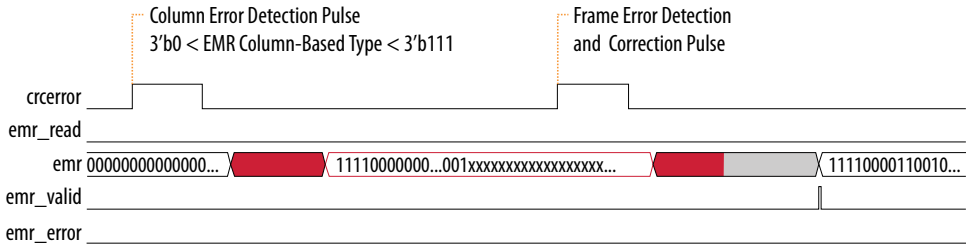


Figure 3. emr_valid Signal for Correctable Errors after Power Up Only (Column-Based Type == 3'b0)

Note: When first loaded with the bitstream, the FPGA executes Frame-based EDCRC once, calculates the column-based check bit and turns it into column-based EDCRC. This timing diagram is referring to the error detected during frame-based EDCRC.

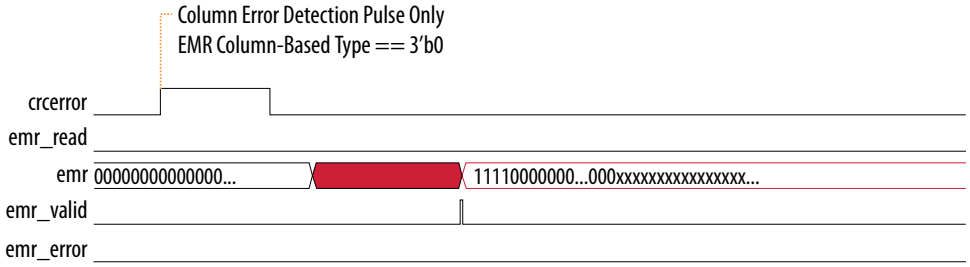


Figure 4. emr_valid Signal for Uncorrectable Errors

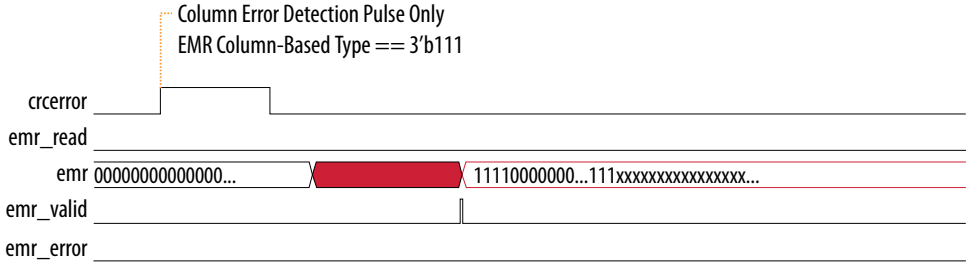
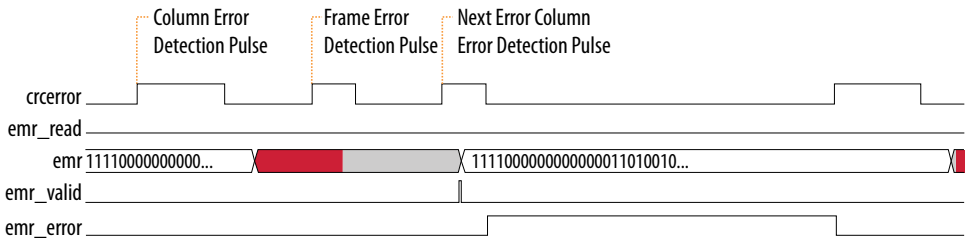


Figure 5. emr_error Timing Diagram



1.4.3.2. All Other Device Timing

The following waveforms show the Error Message Register Unloader IP core timing behavior for Stratix V, Stratix IV, Arria V, Arria II GZ/GX, and Cyclone V devices.

Figure 6. emr_read Timing Diagram

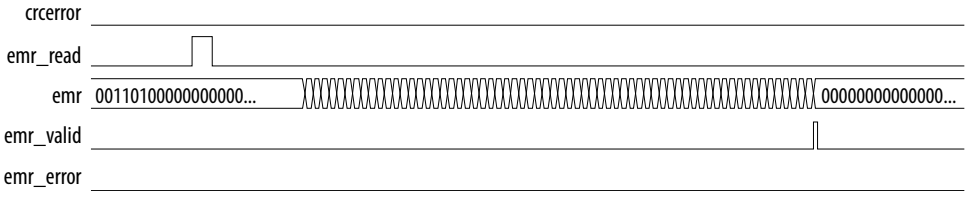


Figure 7. emr_valid Timing Diagram

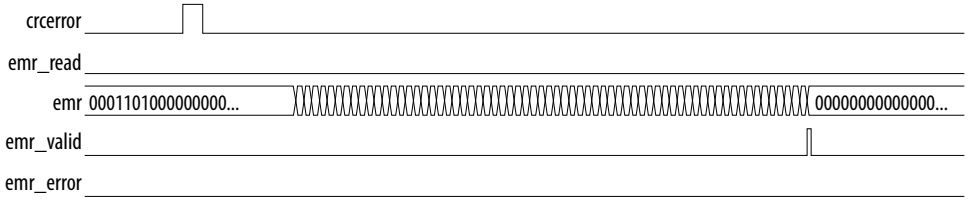
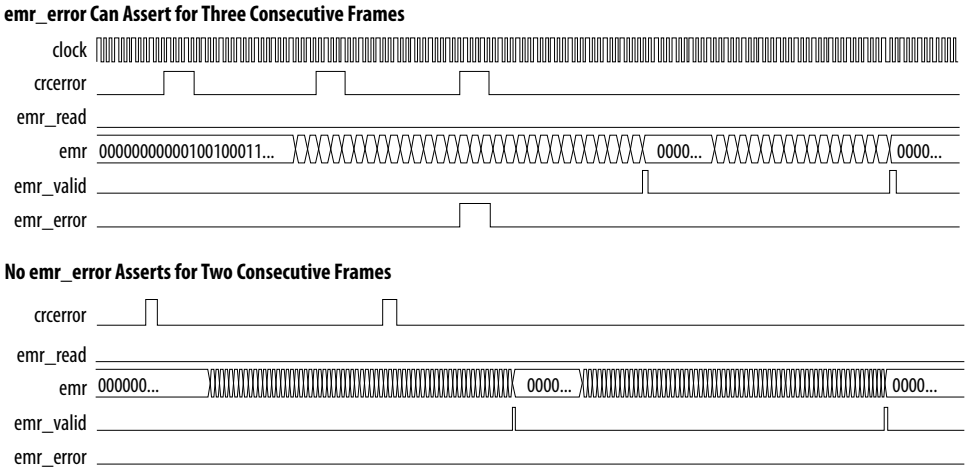


Figure 8. Example EMR Errors Timing Diagram



- In the case of 2 consecutive SEU errors, the IP core asserts `emr_error` for the lost EMR content.
- The IP core asserts `emr_error` if it detects the falling edge of the `crcerror` pulse for the next error, before the IP core loads the previous content of the EMR user update register into the user shift register.
- The rising edge of `crcerror` deasserts `emr_error`.
- `emr_error` is a critical system state and can indicate that the Error Message Register Unloader input clock is too slow.

1.5. Parameter Settings

Table 4. Error Message Register Unloader Parameters

Parameter	Value	Default	Description
CRC error check clock divisor	1, 2, 4, 8, 16, 32, 64, 128, 256	2	Indicates the error detection clock divisor value to apply to the internal oscillator. The divided clock drives the internal CRC function. This setting must match the `ERROR_CHECK_FREQUENCY_DIVISOR` Intel Quartus Prime Settings File (**.qsf**) setting, otherwise the software issues a warning. Stratix IV and Arria II devices do not support a value of 1.
Enable Virtual JTAG CRC error injection	On, off	Off	Enables in-system sources and probes (ISSP) functionality to inject the EMR register content via the JTAG interface without changing the CRAM value. Use this interface to troubleshoot user logic that is connected to the core.
Input clock frequency	Any	50 MHz	Specifies the frequency of the Error Message Register Unloader IP core input clock. This option is applicable when the **Input clock is driven from Internal Oscillator** parameter is off.
Input clock is driven from Internal Oscillator	On, off	Off	Indicates that the internal oscillator provides the core input clock. Enable this parameter if an internal oscillator drives the user design's core input clock. *Note:* The frequency of the internal oscillator is not affected by the CRC error check clock divisor.
CRC Error Verify input clock frequency	10 - 50 MHz	50 MHz	Specifies CRC Error Verify IP core (`ALTERA_CRCERROR_VERIFY`) input clock frequency. Stratix IV and Arria II devices only.
Completion of full chip Error Detection cycle	On, off	Off	Optional. Turn on to assert this signal at the end of each full chip error detection cycle. Stratix V, Intel Arria 10, Arria V, Cyclone V, and Intel Cyclone 10 GX devices only.

1.6. Installing and Licensing Intel FPGA IP Cores

The Intel Quartus Prime software installation includes the Intel FPGA IP library. This library provides many useful IP cores for your production use without the need for an additional license. Some Intel FPGA IP cores require purchase of a separate license for production use. The Intel FPGA IP Evaluation Mode allows you to evaluate these licensed Intel FPGA IP cores in simulation and hardware, before deciding to purchase a

full production IP core license. You only need to purchase a full production license for licensed Intel IP cores after you complete hardware testing and are ready to use the IP in production.

The Intel Quartus Prime software installs IP cores in the following locations by default:

Figure 9. IP Core Installation Path

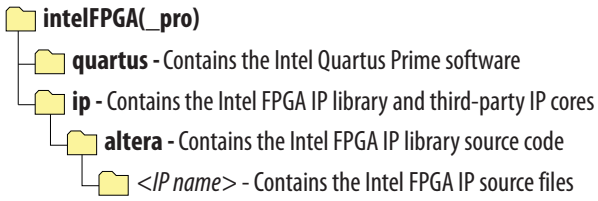


Table 5. IP Core Installation Locations

Location	Software	Platform
`<drive>:\intelFPGA_pro\quartus\ip\altera`	Intel Quartus Prime Pro Edition	Windows*
`<drive>:\intelFPGA\quartus\ip\altera`	Intel Quartus Prime Standard Edition	Windows
`<home directory>:/intelFPGA_pro/quartus/ip/altera`	Intel Quartus Prime Pro Edition	Linux*
`<home directory>:/intelFPGA/quartus/ip/altera`	Intel Quartus Prime Standard Edition	Linux

1.7. Customizing and Generating IP Cores

You can customize IP cores to support a wide variety of applications. The Intel Quartus Prime IP Catalog and parameter editor allow you to quickly select and configure IP core ports, features, and output files.

1.7.1. IP Catalog and Parameter Editor

The IP Catalog displays the IP cores available for your project, including Intel FPGA IP and other IP that you add to the IP Catalog search path.. Use the following features of the IP Catalog to locate and customize an IP core:

- Filter IP Catalog to **Show IP for active device family** or **Show IP for all device families**. If you have no project open, select the **Device Family** in IP Catalog.
- Type in the Search field to locate any full or partial IP core name in IP Catalog.
- Right-click an IP core name in IP Catalog to display details about supported devices, to open the IP core's installation folder, and for links to IP documentation.
- Click **Search for Partner IP** to access partner IP information on the web.

The parameter editor prompts you to specify an IP variation name, optional ports, and output file generation options. The parameter editor generates a top-level Intel Quartus Prime IP file (`.ip`) for an IP variation in Intel Quartus Prime Pro Edition projects.

The parameter editor generates a top-level Quartus IP file (`.qip`) for an IP variation in Intel Quartus Prime Standard Edition projects. These files represent the IP variation in the project, and store parameterization information.

Figure 10. IP Parameter Editor (Intel Quartus Prime Pro Edition)

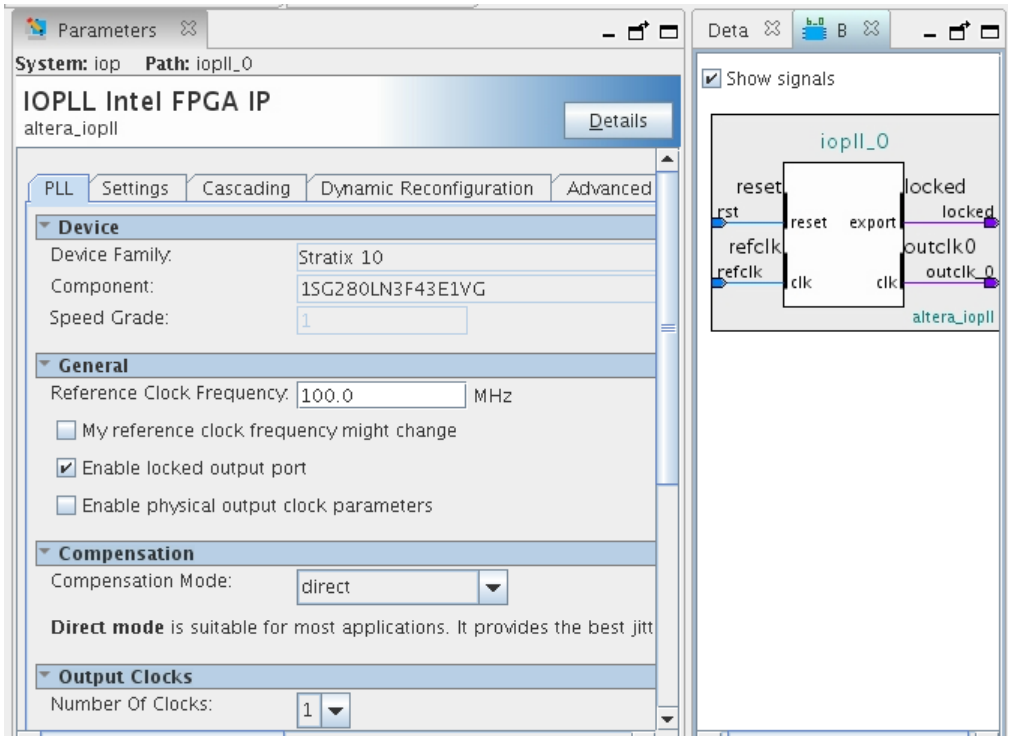
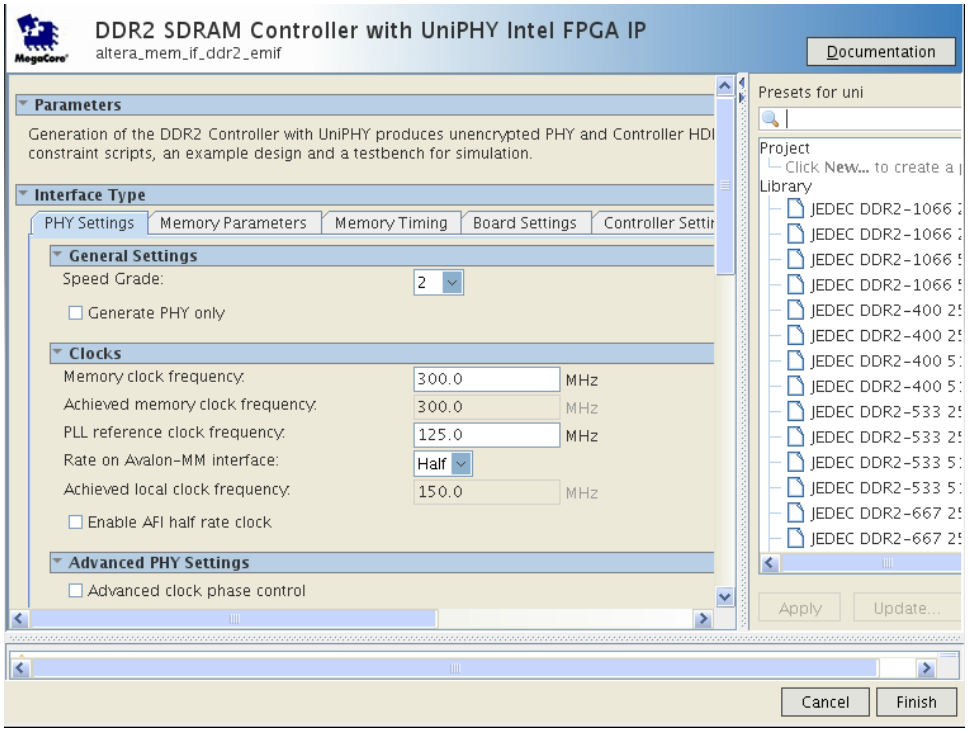


Figure 11. IP Parameter Editor (Intel Quartus Prime Standard Edition)



1.7.2. The Parameter Editor

The parameter editor helps you to configure IP core ports, parameters, and output file generation options. The basic parameter editor controls include the following:

- Use the **Presets** window to apply preset parameter values for specific applications (for select cores).
- Use the **Details** window to view port and parameter descriptions, and click links to documentation.
- Click **Generate ➤ Generate Testbench System** to generate a testbench system (for select cores).
- Click **Generate ➤ Generate Example Design** to generate an example design (for select cores).
- Click **Validate System Integrity** to validate a system's generic components against companion files. (Platform Designer systems only)
- Click **Sync All System Info** to validate a system's generic components against companion files. (Platform Designer systems only)

The IP Catalog is also available in Platform Designer (**View ➤ IP Catalog**). The Platform Designer IP Catalog includes exclusive system interconnect, video and image processing, and other system-level IP that are not available in the Intel Quartus Prime IP Catalog. Refer to *Creating a System with Platform Designer* or *Creating a System with Platform Designer (Standard)* for information on use of IP in Platform Designer (Standard) and Platform Designer, respectively.

Related Information

- Creating a System with Platform Designer
- Creating a System with Platform Designer (Standard) (Standard)

1.7.3. Specifying IP Core Parameters and Options

Follow these steps to specify IP core parameters and options.

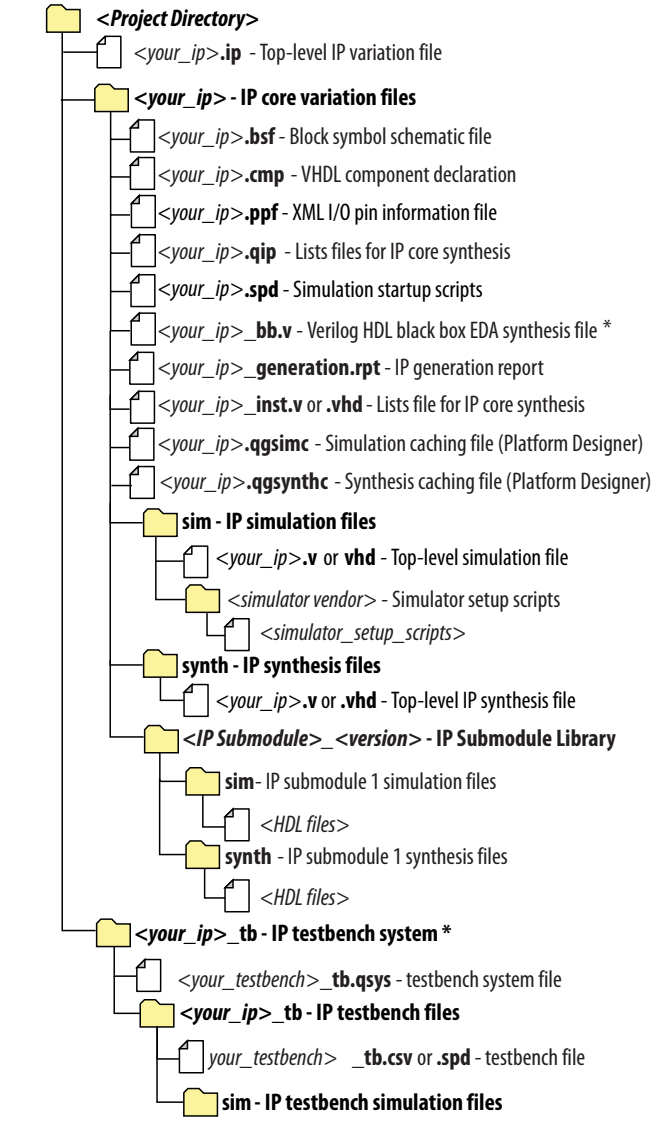
1. In the Platform Designer IP Catalog (**Tools ➤ IP Catalog**), locate and double-click the name of the IP core to customize. The parameter editor appears.
2. Specify a top-level name for your custom IP variation. This name identifies the IP core variation files in your project. If prompted, also specify the target FPGA device family and output file HDL preference. Click **OK**.
3. Specify parameters and options for your IP variation:
   - Optionally select preset parameter values. Presets specify all initial parameter values for specific applications (where provided).
   - Specify parameters defining the IP core functionality, port configurations, and device-specific features.
   - Specify options for generation of a timing netlist, simulation model, testbench, or example design (where applicable).
   - Specify options for processing the IP core files in other EDA tools.
4. Click **Finish** to generate synthesis and other optional files matching your IP variation specifications. The parameter editor generates the top-level `.qsys` IP variation file and HDL files for synthesis and simulation. Some IP cores also simultaneously generate a testbench or example design for hardware testing.
5. To generate a simulation testbench, click **Generate ➤ Generate Testbench System**. **Generate Testbench System** is not available for some IP cores that do not provide a simulation testbench.
6. To generate a top-level HDL example for hardware verification, click **Generate ➤ HDL Example**. **Generate ➤ HDL Example** is not available for some IP cores.

The top-level IP variation is added to the current Intel Quartus Prime project. Click **Project ➤ Add/Remove Files in Project** to manually add a `.qsys` (Intel Quartus Prime Standard Edition) or `.ip` (Intel Quartus Prime Pro Edition) file to a project. Make appropriate pin assignments to connect ports.

1.7.3.1. IP Core Generation Output (Intel Quartus Prime Pro Edition)

The Intel Quartus Prime software generates the following output file structure for individual IP cores that are not part of a Platform Designer system.

Figure 12. Individual IP Core Generation Output (Intel Quartus Prime Pro Edition)



* If supported and enabled for your IP core variation.

Table 6. Output Files of Intel FPGA IP Generation

File Name	Description
`<your_ip>.ip`	Top-level IP variation file that contains the parameterization of an IP core in your project. If the IP variation is part of a Platform Designer system, the parameter editor also generates a `.qsys` file.
`<your_ip>.cmp`	The VHDL Component Declaration (`.cmp`) file is a text file that contains local generic and port definitions that you use in VHDL design files.
`<your_ip>_generation.rpt`	IP or Platform Designer generation log file. Displays a summary of the messages during IP generation.

continued...

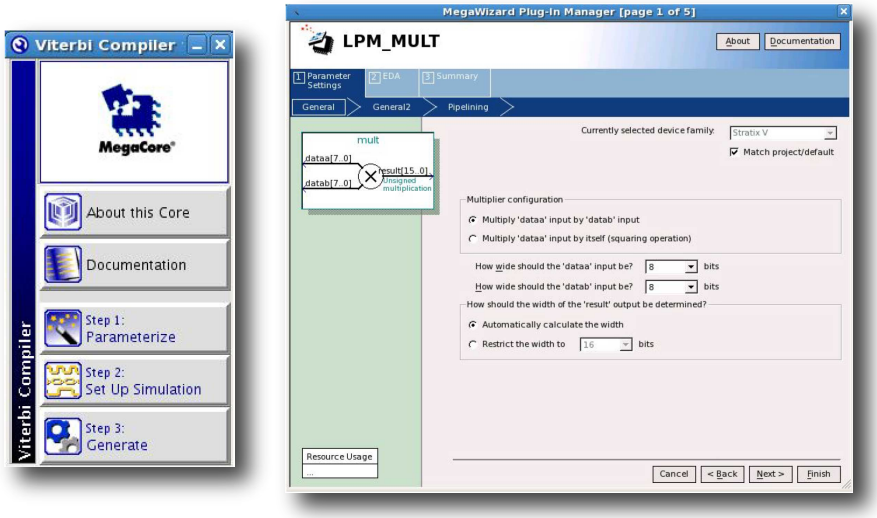
File Name	Description
<*your_ip*>.qgsimc (Platform Designer systems only)	Simulation caching file that compares the .qsys and .ip files with the current parameterization of the Platform Designer system and IP core. This comparison determines if Platform Designer can skip regeneration of the HDL.
<*your_ip*>.qgsynth (Platform Designer systems only)	Synthesis caching file that compares the .qsys and .ip files with the current parameterization of the Platform Designer system and IP core. This comparison determines if Platform Designer can skip regeneration of the HDL.
<*your_ip*>.qip	Contains all information to integrate and compile the IP component.
<*your_ip*>.csv	Contains information about the upgrade status of the IP component.
<*your_ip*>.bsf	A symbol representation of the IP variation for use in Block Diagram Files (.bdf).
<*your_ip*>.spd	Input file that ip-make-simscript requires to generate simulation scripts. The .spd file contains a list of files you generate for simulation, along with information about memories that you initialize.
<*your_ip*>.ppf	The Pin Planner File (.ppf) stores the port and node assignments for IP components you create for use with the Pin Planner.
<*your_ip*>_bb.v	Use the Verilog blackbox (_bb.v) file as an empty module declaration for use as a blackbox.
<*your_ip*>_inst.v or _inst.vhd	HDL example instantiation template. Copy and paste the contents of this file into your HDL file to instantiate the IP variation.
<*your_ip*>.regmap	If the IP contains register information, the Intel Quartus Prime software generates the .regmap file. The .regmap file describes the register map information of master and slave interfaces. This file complements the .sopcinfo file by providing more detailed register information about the system. This file enables register display views and user customizable statistics in System Console.
<*your_ip*>.svd	Allows HPS System Debug tools to view the register maps of peripherals that connect to HPS within a Platform Designer system. During synthesis, the Intel Quartus Prime software stores the .svd files for slave interface visible to the System Console masters in the .sof file in the debug session. System Console reads this section, which Platform Designer queries for register map information. For system slaves, Platform Designer accesses the registers by name.
<*your_ip*>.v <*your_ip*>.vhd	HDL files that instantiate each submodule or child IP core for synthesis or simulation.
mentor/	Contains a msim_setup.tcl script to set up and run a simulation.
aldec/	Contains a script rivierapro_setup.tcl to setup and run a simulation.
/synopsys/vcs /synopsys/vcsmx	Contains a shell script vcs_setup.sh to set up and run a simulation. Contains a shell script vcsmx_setup.sh and synopsys_sim.setup file to set up and run a simulation.
/cadence	Contains a shell script ncsim_setup.sh and other setup files to set up and run an simulation.
/xcelium	Contains an Parallel simulator shell script xcelium_setup.sh and other setup files to set up and run a simulation.
/submodules	Contains HDL files for the IP core submodule.
<*IP submodule*>/	Platform Designer generates /synth and /sim sub-directories for each IP submodule directory that Platform Designer generates.

1.7.4. Specifying IP Core Parameters and Options (Legacy Parameter Editors)

Some IP cores use a legacy version of the parameter editor for configuration and generation. Use the following steps to configure and generate an IP variation using a legacy parameter editor.

Note: The legacy parameter editor generates a different output file structure than the latest parameter editor. Refer to *Specifying IP Core Parameters and Options* for configuration of IP cores that use the latest parameter editor.

Figure 13. Legacy Parameter Editors



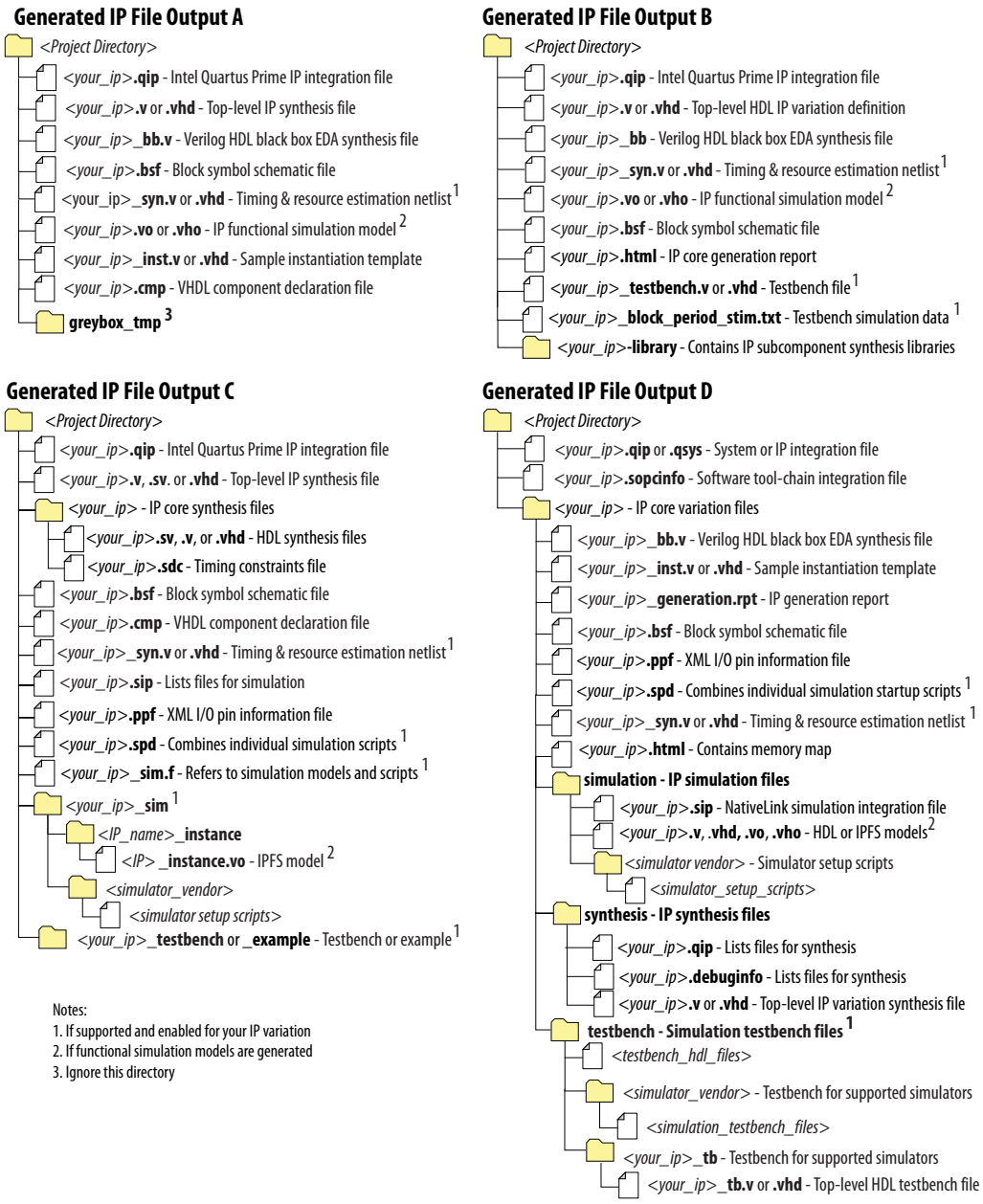
1. In the IP Catalog (**Tools ➤ IP Catalog**), locate and double-click the name of the IP core to customize. The parameter editor appears.
2. Specify a top-level name and output HDL file type for your IP variation. This name identifies the IP core variation files in your project. Click **OK**.
3. Specify the parameters and options for your IP variation in the parameter editor. Refer to your IP core user guide for information about specific IP core parameters.
4. Click **Finish** or **Generate** (depending on the parameter editor version). The parameter editor generates the files for your IP variation according to your specifications. Click **Exit** if prompted when generation is complete. The parameter editor adds the top-level `.qip` file to the current project automatically.

   Note: To manually add an IP variation generated with legacy parameter editor to a project, click **Project ➤ Add/Remove Files in Project** and add the IP variation `.qip` file.

1.7.4.1. IP Core Generation Output (Intel Quartus Prime Standard Edition)

The Intel Quartus Prime Standard Edition software generates one of the following output file structures for individual IP cores that use one of the legacy parameter editors.

Figure 14. IP Core Generated Files (Legacy Parameter Editors)



Notes:
1. If supported and enabled for your IP variation
2. If functional simulation models are generated
3. Ignore this directory

1.8. Document Revision History for Error Message Register Unloader Intel FPGA IP IP Core User Guide

Document Version	Intel Quartus Prime Version	Changes
2018.05.23	18.0	• Renamed IP from *Intel FPGA Error Message Register Unloader IP core* to *Error Message Register Unloader Intel FPGA IP core*. • Updated figures *emr_valid Signal for Correctable Errors after Power Up Only (Column-Based Type == 3'b0)* and *emr_valid Signal for Uncorrectable Errors*.

Date	Version	Changes
December 2017	2017.12.18	• Renamed the document as *Intel FPGA Error Message Register Unloader IP Core User Guide*. • Updated the "IP Core Device Support" table. • Updated for latest branding standards. • Made editorial updates throughout the document.
July 2017	2017.07.15	• Added Intel Cyclone 10 GX device support. • Changed V-Type to Column-Based Type in IP timing diagrams. • Provided separate parameterization instructions for Intel Quartus Prime Pro Edition and Intel Quartus Prime Standard Edition. • Updated for latest branding standards.
May 2016	2016.05.02	• Removed feature bullet about Verilog HDL RTL support. • Changed Quartus II references to Quartus Prime.
June 2015	2015.06.12	Updated Arria 10 support details.
December 2014	2014.12.15	Initial release.