Avant-G/X SEDC Module IP

User Guide

FPGA-IPUG-02232-1.2

November 2024

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
ASR	Address shift register
CRAM	Configuration Random Access Memory
CRC32	Cyclic Redundancy Check – 32 bits
DSR	Data shift register
ECC	Error Correcting Code
GSR	Global System Reset
LMMI	Lattice Memory Mapped Interface
SEC	Single Error Correct
SED	Single Error Detect
SEDC	Single Error Detect/Correct
SEI	Soft Error Injector
SEU	Single Event Upset
SoC	System-on-a-Chip

1. Introduction

This document provides technical information about the SEDC (Single Error Detect/Correct Module) that is supported in Lattice FPGA devices built on the Lattice Avant™ platform. This aims to provide information essential for IP/System developers, Verification and Software for integration, testing and validation. In general, design specification from RTL up to IP packaging, IP generation, and integration with Lattice Radiant™ software are covered in this document.

- Supports both SEC (Single Error Correct) and SED (Single Error Detect) modes
- Provides information on the Soft Error location
- Supports SEC without the need for external memory access

2. Functional Description

2.1. Functional Overview

Soft Error Detection (SED), when enabled, performs reads of DSR frames in user mode and performs ECC (Error Correcting Code) Calculations for the Soft Error Detection of SRAM content. Once an error is detected/corrected, a notification is sent to the fabric and SED resumes. If more than one-bit error is detected within one frame notification, a Double Error Detect is sent to the fabric. In addition, the CRC32 (Cyclic Redundancy Check – 32 bits) is done on the entire CRAM (Configuration Random Access Memory) data in parallel with ECC.

2.2. Attributes Summary

Table 2.1 shows the configurable attributes of the Lattice Avant SEDC IP Core for simulation. The attributes can be configured through the Module/IP Block Wizard.

Note: The attributes under CRAM Error Location Simulation are designed exclusively for simulating errors and have no impact on the generated IP.

Table 2.1. Attributes Summary

Attribute	Selectable Values	Default	Description
Configuration			
Clock Divider	2-256	2	Indicates the divider value for SEDC clock.
CRAM Error Simulation			
Simulate 1-bit Correctable CRAM error	Checked, Unchecked	Unchecked	When checked, 1-bit correctable CRAM error is injected in the simulation. Error location is set in other attributes.
Simulate 2-bit Non-Correctable CRAM error	Checked, Unchecked	Unchecked	When checked, 2-bit non-correctable CRAM error is injected in the simulation.
Simulate CRC CRAM Error	Checked, Unchecked	Unchecked	When checked, CRC CRAM error is injected in the simulation.
CRAM 1-bit Error Location Simulation			
CRAM 1-bit Bit Error Location	0 – 1023	0	Indicates the target DSR bit location of 1-bit error. Applicable only if *Simulate 1-bit Correctable CRAM Error* is checked.
CRAM 1-bit Frame Error Location	0 – 16383	0	Indicates the target ASR bit location of 1-bit error. Applicable only if *Simulate 1-bit Correctable CRAM Error* is checked.
CRAM 1-bit Region Error Location	0 – 31	0	Indicates the target region location of 1-bit error. Applicable only if *Simulate 1-bit Correctable CRAM Error* is checked.

2.3. Register Description

2.3.1. SEDC Status Register 0 0x0

Table 2.2. SEDC Status 0

Bits	Name	Access	Width	Reset
[15:8]	*reserved*	read-only	8	0x0
[7]	reg_cram_errcrc	read-only	1	0x0
[6]	reg_cram_errm	read-only	1	0x0
[5]	reg_cram_err1	read-only	1	0x0
[4]	reg_cram_err	read-only	1	0x0
[3]	*reserved*	read-only	1	0x0
[2]	reg_sedc_done	read-only	1	0x0
[1]	reg_sedc_busy	read-only	1	0x0
[0]	reg_sedc_enable	read-only	1	0x0

reg_cram_errcrc
When set to 1, a CRC error in CRAM registers has been detected.

reg_cram_errm
When set to 1, an uncorrectable error in CRAM registers has been detected.

reg_cram_err1
When set to 1, a 1-bit error in CRAM registers has been detected.

reg_cram_err
When set to 1, an error in CRAM registers has been detected.

reg_sedc_done
When set to 1, SEDC one-time scan has finished.

reg_sedc_busy
When set to 1, SEDC is busy.

reg_sedc_enable
When set to 1, SEDC is enabled in fabric.

2.3.2. SEDC Status Register 1 0x1

Table 2.3. SEDC Status Register 1

Bits	Name	Access	Width	Reset
[15:10]	*reserved*	read-only	6	0x0
[9:0]	reg_cram_bit_errloc	read-only	10	0x0

reg_cram_bit_errloc
Indicates the DSR bit location of the last 1-bit error.

2.3.3. SEDC Status Register 2 0x2

Table 2.4. SEDC Status Register 2

Bits	Name	Access	Width	Reset
[15:14]	*reserved*	read-only	2	0x0
[13:0]	reg_cram_frm_errloc	read-only	14	0x0

reg_cram_frm_errloc
Indicates the ASR frame location of the last 1-bit error.

2.3.4. SEDC Status Register 3 0x3

Table 2.5. SEDC Status Register 3

Bits	Name	Access	Width	Reset
[15:5]	*reserved*	read-only	11	0x0
[4:0]	reg_cram_rgn_errloc	read-only	5	0x0

reg_cram_rgn_errloc

Indicates the region location of the last 1-bit error.

2.3.5. SEDC Configuration Register 0 0x7

Table 2.6. SEDC Configuration Register 0

Bits	Name	Access	Width	Reset
[15:2]	*reserved*	read-only	14	0x0
[1]	reg_sedc_resume	read-write	1	0x0
[0]	reg_auto_err_correct	read-write	1	0x0

reg_sedc_resume

If *reg_auto_err_correct* is set to 0, SEDC core will stop the scan when error is detected. Write 1 to this register to resume the scan.

reg_sedc_auto_correct

Must be set as 0 – When 1-bit error is detected, SEDC core stops scanning and waits for you to write to the *reg_sedc_resume* register.

2.3.6. SEDC Configuration Register 1 0x8

Table 2.7. SEDC Configuration Register 1

Bits	Name	Access	Width	Reset
[15:8]	*reserved*	read-only	8	0x0
[7:0]	reg_clk_sedc_div	read-only	8	Equal to attribute *Clock Divider – 1.*

reg_clk_sedc_div

Indicates the divider value for SEDC clock.

2.3.7. SEDC Configuration Register 2 0x9

Table 2.8. SEDC Configuration Register 2

Bits	Name	Access	Width	Reset
[15:1]	*reserved*	read-only	15	0x0
[0]	reg_sedc_single_scan_en	read-write	1	0x0

reg_sedc_single_scan_en

Must be set as 1 – SEDC scan is one-time for CRAM. The scan will start when *sedc_enable* is asserted. If the scan has finished without error, *sedc_done* will be asserted. If the scan has finished with error, the corresponding SEDC error status will be asserted. You can toggle sedc_ena at the end of a scan cycle for SEDC to scan continuously.

2.4. Sample Advanced SEDC Applications

The following sections describe advanced SEDC applications when combined with other IPs. This section is not part of the core SEDC IP but may serve as a guideline on how the SEDC IP can be used as a user solution.

These applications can be combined to provide maximum flexibility in the designs.

2.4.1. SEDC Integration with EBR ECC

The EBR modules have a separate ECC port with the one in the SEDC block. While the SEDC block automatically sweeps the information in the bitstream, the EBR module's ECC functions are triggered only when an access is made to a specific memory location. For more information on ECC on EBR memory, refer to EBR Memory Modules – Lattice Radiant Software User Guide (FPGA-IPUG-02190). Since multiple EBRs can be used in a single design, you can implement error detection with either a mux[with selector] or as independent IPs.

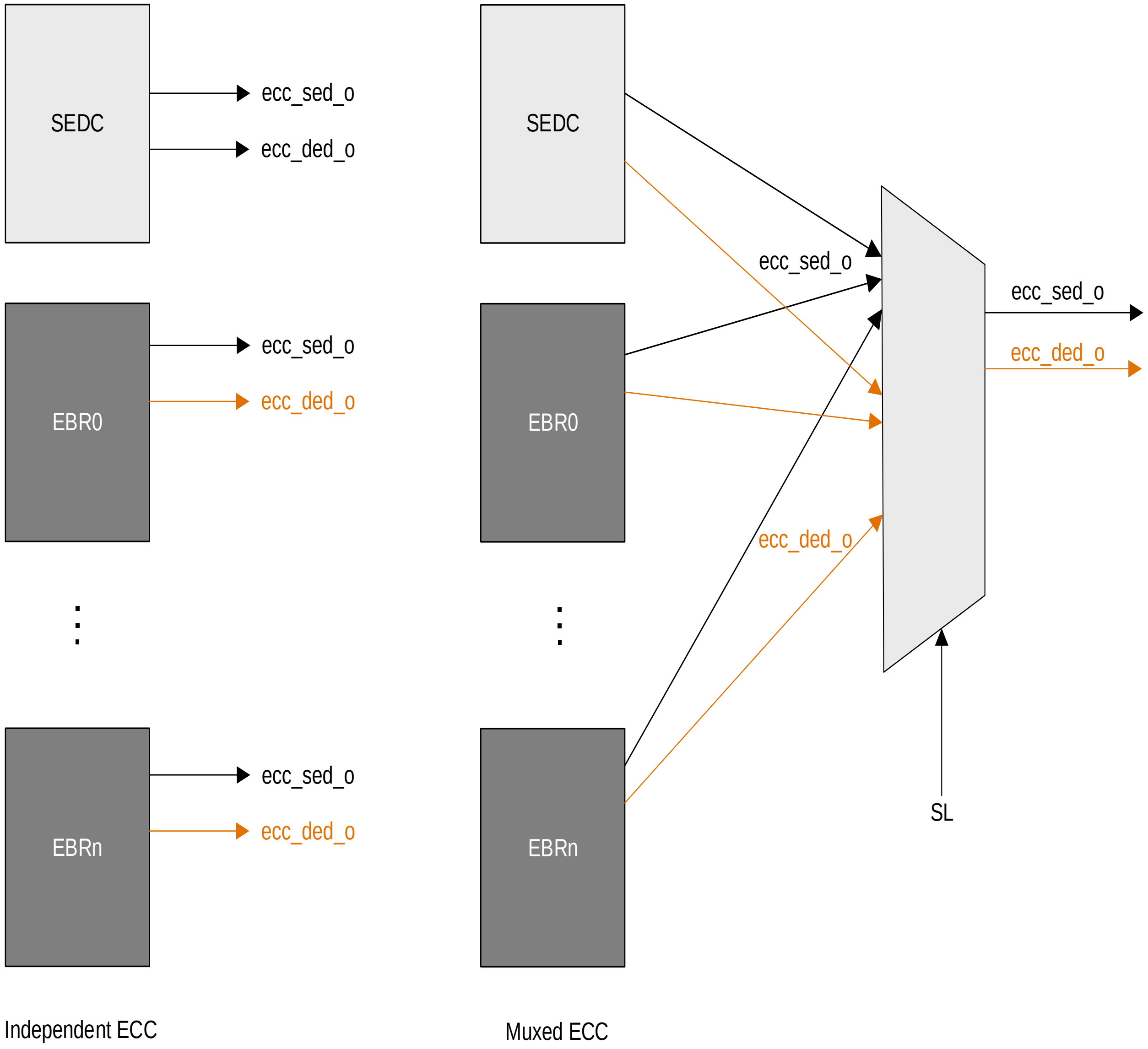


Figure 2.1. SEDC + EBR Sample Implementation

2.4.2. Automatic System Refresh

The SEDC can be interfaced with the global system reset, which restarts the entire system if it encounters an SEU error. For more information on the pins and parameters of the GSR (Global System Reset) primitive, check the included *Help* section in the Lattice Radiant software.

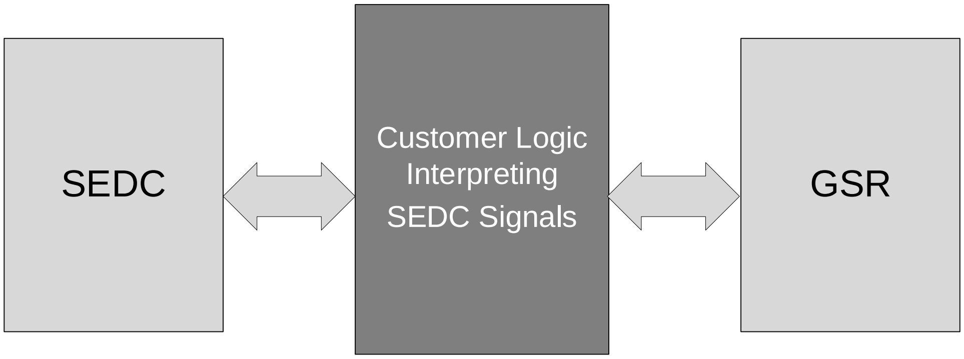


Figure 2.2. Automatic System Refresh Block Diagram

2.4.3. System Control on Error Event

The SEDC can also be interfaced with a bridge coupled with SoC design applications. In this setup, the SEDC can be used to monitor the overall system health and a manager can be designed to respond to interrupts.

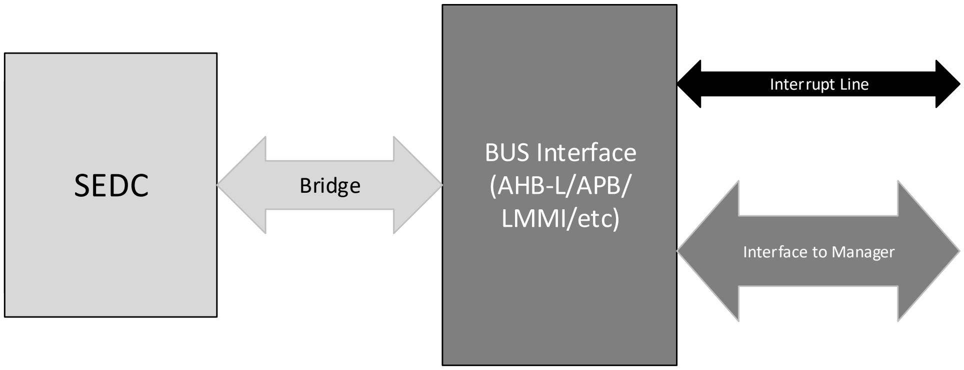


Figure 2.3. Bus Interface Block Diagram

3. IP Generation and Evaluation

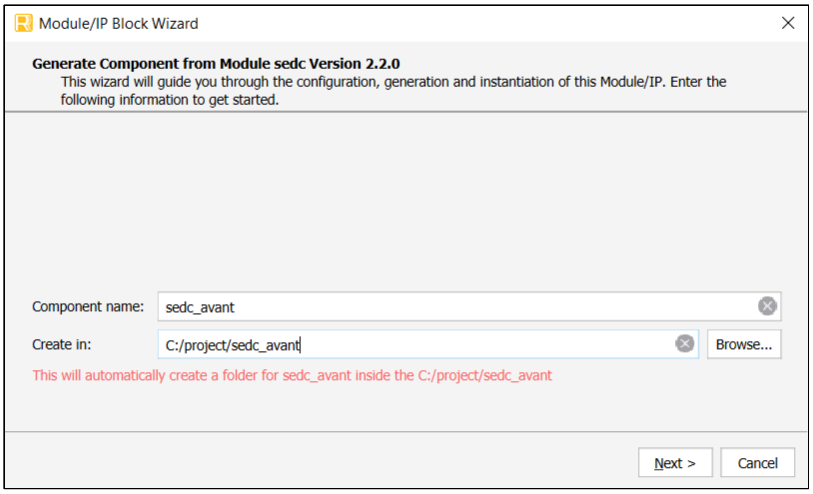
This section provides information on how to generate and instantiate the Module/IP using the Lattice Radiant software.

3.1. Generating and Synthesizing the IP

The Module/IP Block Wizard in Lattice Radiant Software allows you to generate, create, or open modules for the target device.

To generate the Module/IP:

1. Create a new Lattice Radiant software project or open an existing project.
2. In the **IP Catalog** tab, double-click **SEDC** under **IP on Local** > **Architecture Modules** category. The **Module/IP Block Wizard** opens as shown in Figure 3.1. Enter values in the **Component name** and the **Create in** fields and click **Next**.



Module/IP Block Wizard

Generate Component from Module sedc Version 2.2.0

This wizard will guide you through the configuration, generation and instantiation of this Module/IP. Enter the following information to get started.

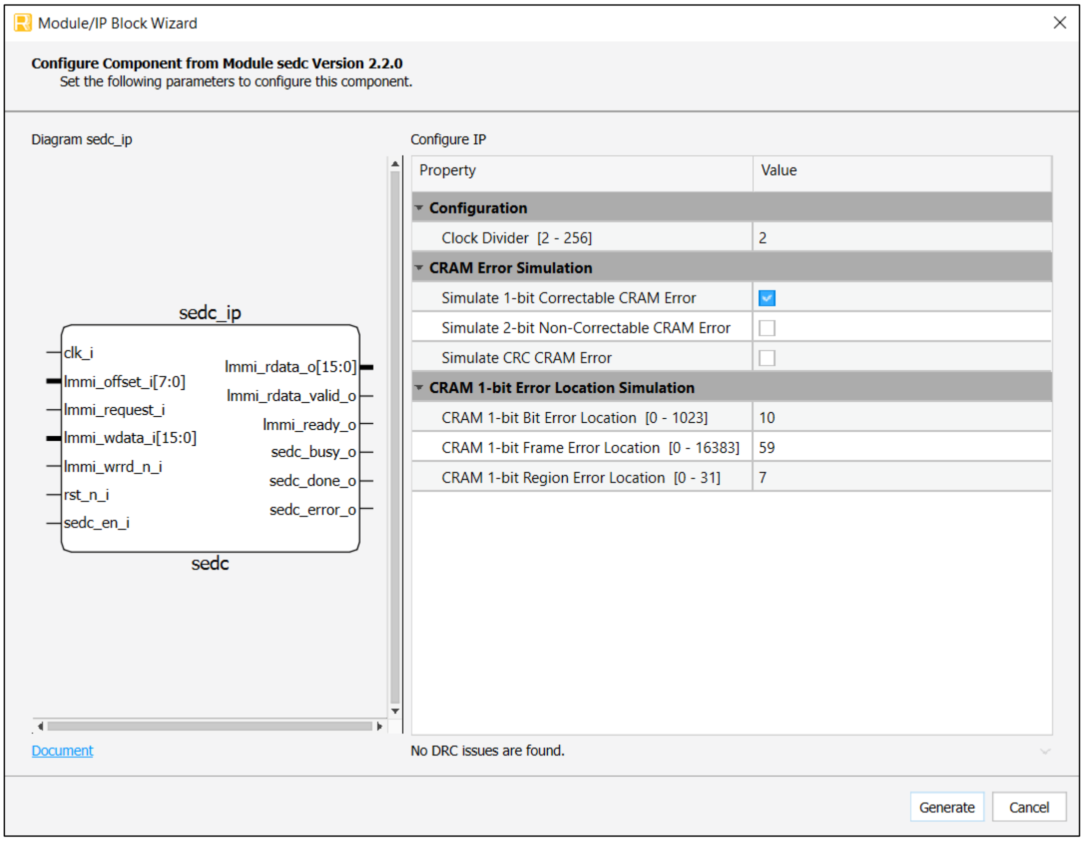
Component name: sedc_avant

Create in: C:/project/sedc_avant

This will automatically create a folder for sedc_avant inside the C:/project/sedc_avant

Figure 3.1. Module/IP Block Wizard

3. In the next **Module/IP Block Wizard** window, customize the selected SEDC IP using fields and check boxes. Figure 3.2 shows an example configuration of the SEDC IP. For details on the configuration options, refer to the Attributes Summary section.



Module/IP Block Wizard

Configure Component from Module sedc Version 2.2.0
Set the following parameters to configure this component.

Diagram sedc_ip

Configure IP

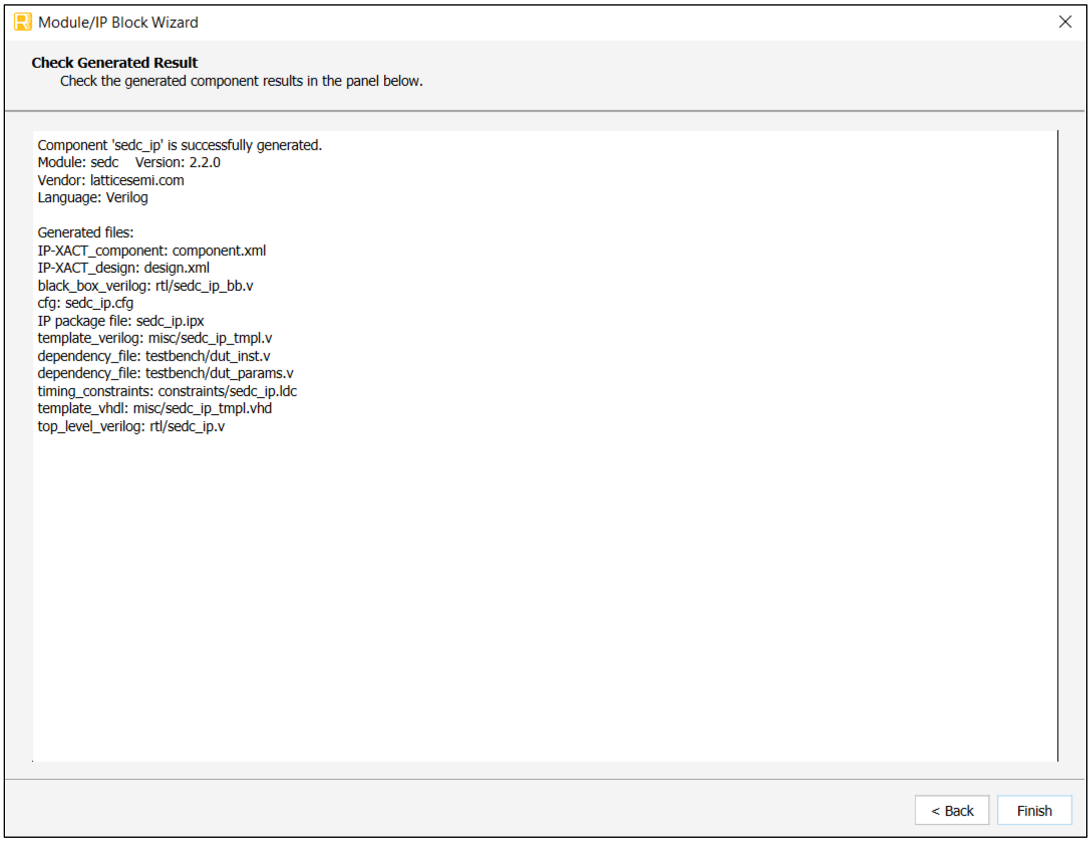
Property	Value
Configuration	
Clock Divider [2 - 256]	2
CRAM Error Simulation	
Simulate 1-bit Correctable CRAM Error	☑
Simulate 2-bit Non-Correctable CRAM Error	□
Simulate CRC CRAM Error	□
CRAM 1-bit Error Location Simulation	
CRAM 1-bit Bit Error Location [0 - 1023]	10
CRAM 1-bit Frame Error Location [0 - 16383]	59
CRAM 1-bit Region Error Location [0 - 31]	7

Document

No DRC issues are found.

Figure 3.2. Configuration Window

4. Click **Generate**. The **Check Generating Result** dialog box opens, showing design block messages and results as shown in Figure 3.3.



Check Generated Result
Check the generated component results in the panel below.

```
Component 'sedc_ip' is successfully generated.
Module: sedc    Version: 2.2.0
Vendor: latticesemi.com
Language: Verilog

Generated files:
IP-XACT_component: component.xml
IP-XACT_design: design.xml
black_box_verilog: rtl/sedc_ip_bb.v
cfg: sedc_ip.cfg
IP package file: sedc_ip.ipx
template_verilog: misc/sedc_ip_tmpl.v
dependency_file: testbench/dut_inst.v
dependency_file: testbench/dut_params.v
timing_constraints: constraints/sedc_ip.ldc
template_vhdl: misc/sedc_ip_tmpl.vhd
top_level_verilog: rtl/sedc_ip.v
```

Figure 3.3. SEDC File List

5. Click the **Finish** button. All the generated files are placed under the directory paths in the **Create in** and the **Component name** fields shown in Figure 3.1.

The generated SEDC IP Core package includes the closed-box (<Component name>_bb.v) and instance templates (<Component name>_tmpl.v/vhd) that can be used to instantiate the core in a top-level design. An example RTL top-level reference source file (<Component name>.v) that can be used as an instantiation template for the IP core is also provided. You can use this top-level reference as the starting template for the top-level of your complete design. The generated files are listed in Table 3.1.

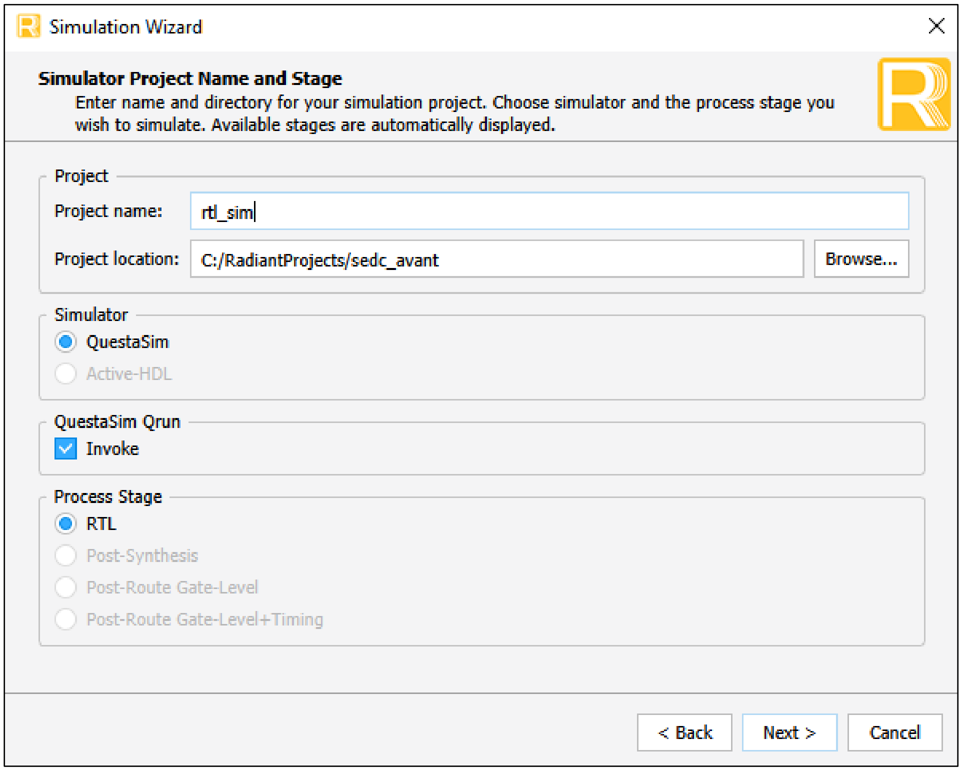
Table 3.1. Generated File List

Attribute	Description
<Component name>.ipx	This file contains the information on the files associated to the generated IP.
<Component name>.cfg	This file contains the attribute values used in IP configuration.
component.xml	This file contains the ipxact:component information of the IP.
design.xml	This file documents the configuration attributes of the IP in IP-XACT 2014 format.
rtl/<Component name>.v	This file provides an example RTL top file that instantiates the IP core.
rtl/<Component name>_bb.v	This file provides the synthesis closed-box.
misc/<Component name>_tmpl.v misc /<Component name>_tmpl.vhd	These files provide instance templates for the IP core.

3.2. Running Functional Simulation

The testbench included in this IP can only be used in RTL simulation. Post-synthesis and Post-Route simulations are not supported.

1. Click the icon located on the **Toolbar** to initiate the **Simulation Wizard** shown in Figure 3.4.



Simulation Wizard

Simulator Project Name and Stage

Enter name and directory for your simulation project. Choose simulator and the process stage you wish to simulate. Available stages are automatically displayed.

Project

Project name: rtl_sim

Project location: C:/RadiantProjects/sedc_avant

Simulator

● QuestaSim

○ Active-HDL

QuestaSim Qrun

☑ Invoke

Process Stage

● RTL

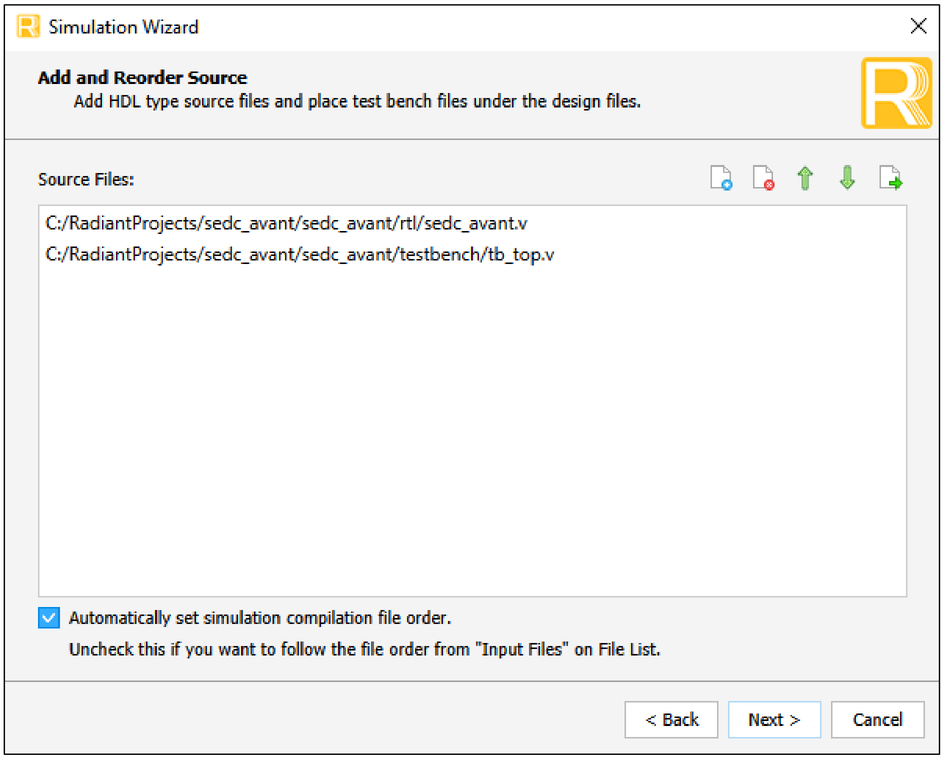
○ Post-Synthesis

○ Post-Route Gate-Level

○ Post-Route Gate-Level+Timing

Figure 3.4. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 3.5.



Simulation Wizard

Add and Reorder Source

Add HDL type source files and place test bench files under the design files.

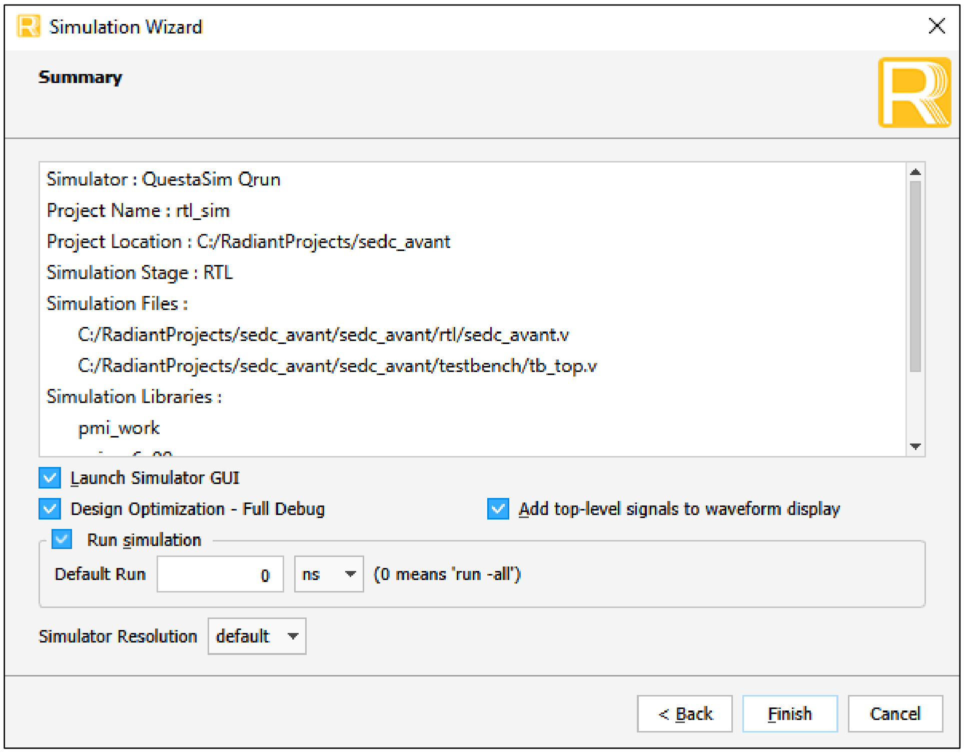
Source Files:

C:/RadiantProjects/sedc_avant/sedc_avant/rtl/sedc_avant.v
C:/RadiantProjects/sedc_avant/sedc_avant/testbench/tb_top.v

☑ Automatically set simulation compilation file order.
Uncheck this if you want to follow the file order from "Input Files" on File List.

Figure 3.5. Adding and Reordering Source

3. Click **Next**. The **Summary** window is shown in Figure 3.6. Click **Finish** to run the simulation.



Simulation Wizard

Summary

Simulator : QuestaSim Qrun
Project Name : rtl_sim
Project Location : C:/RadiantProjects/sedc_avant
Simulation Stage : RTL
Simulation Files :
C:/RadiantProjects/sedc_avant/sedc_avant/rtl/sedc_avant.v
C:/RadiantProjects/sedc_avant/sedc_avant/testbench/tb_top.v
Simulation Libraries :
pmi_work

☑ Launch Simulator GUI

☑ Design Optimization - Full Debug ☑ Add top-level signals to waveform display

☑ Run simulation

Default Run 0 ns (0 means 'run -all')

Simulator Resolution default

Figure 3.6. Simulation Wizard Summary

References

- EBR Memory Modules – Lattice Radiant Software User Guide (FPGA-IPUG-02190)
- Avant-G web page
- Avant-X web page
- Lattice Radiant FPGA design software
- Lattice Insights for Lattice Semiconductor training courses and learning plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.2, November 2024

Section	Change Summary
All	Minor editorial changes.
Abbreviations in This Document	• Updated section title, description, and table header. • Rearranged items in alphabetical order. • Added *SEU*.
Introduction	Removed LMMI reference in relation to information on soft error location.
Functional Description	• Removed the Block Diagram section. • Removed the Signal Description section. • Removed reference to INIT Error Simulation in the Attributes Summary section. • In Table 2.1. Attributes Summary: • Updated selectable value range and default value for Clock Divider. • Removed INIT Error Simulation attributes. • In the SEDC Status Register 0 0x0section: • Removed INIT-related registers and updated bits to reserved in Table 2.2. SEDC Status 0. • Removed INIT-related register descriptions. • Removed the SEDC Status Register 4 0x4, SEDC Status Register 5 0x5, and SEDC Status Register 6 0x6 sections. • In the SEDC Configuration Register 0 0x7 section: • Removed reg_sedc_auto_correct description for setting 1 and updated description for setting 0. • In the SEDC Configuration Register 2 0x9 section: • Removed reg_sedc_single_scan_en description for setting 0 and updated description for setting 1.
IP Generation	Updated Figure 3.1. Module/IP Block Wizard through Figure 3.3. SEDC File List.
References	Added EBR Memory Modules – Lattice Radiant Software User Guide.

Revision 1.1, July 2024

Section	Change Summary
All	• Changed the document name from *Avant-G/X SEDC Module IP – Lattice Radiant Software* to *Avant-G/X SEDC Module IP*. • Made editorial fixes.
Functional Description	• Updated the *Direction* of the following *Pin Name* in Table 2.1. SEDC Module Ports: • *clk_i* • *rst_n_i* • *sedc_enable* • *sedc_done* • *sedc_busy* • *sedc_error* • Updated the *Direction* of the following *LMNI* in Tale 2.1. SEDC Module Ports: • *lmmi_request_i* • *lmmi_wrrd_n_i* • *lmmi_offset_i* • *lmmi_wdata_i* • *lmmi_ready_o* • *lmmi_rdata_valid_o* • *lmmi_rdata_o* • Updated the note in section 2.4. Attributes Summary. • In Table 2.2. Attributes Summary:

Section	Change Summary
	• Added configuration. • Updated the table header from *CRAM Error Injecction* to *CRAM Error Simulation*. • Updated the following *CRAM Error Simulatiom*: • Simulate 1-bit Correctable CRAM error • Simulate 2-bit Non-Correctable CRAM error • Simulate CRC CRAM Error • Added *CRAM 1-bit Error Location Simulation* section. • Updated the description of the following *CRAM 1-bit Error Location Simulation*. • CRAM 1-bit Bit Error Location • CRAM 1-bit Frame Error Location • CRAM 1-bit Region Error Location • Updated the table header from INIT Error Injection to INIT Error Simulation. • Updated the following INIT Error Simulation. • Simulate 1-bit Correctable INIT error • Simulate 2-bit Non-Correctable INIT error • Simulate CRC INIT Error • Added INIT 1-bit Error Location Simulation section. • Updated the description of the following INIT 1-bit Error Location Simulation. • INIT 1-bit Bit Error Location • INIT 1-bit Offset Error Location • INIT 1-bit ID Error Location • Updated the *Reset* description of the *Bits [7:0]* in Table 2.11. SEDC Configuration Register 1.
IP Generation and Evaluation	• Updated the existing steps in 3.1. Generating and Synthesizing the IP section. • Updated Figure 3.1. Module/IP Block Wizard – Figure 3.3 in 3.1. Generating and Synthesizing the IP section. • Updated Figure 3.4 – Figure 3.6 in 3.2 Running Functional Simulation section.
References	Updated this section.

Revision 1.0, December 2023

Section	Change Summary
All	• Changed the document name from *Avant SEDC Module IP – Lattice Radiant Software* to *Avant-G/X SEDC Module IP – Lattice Radiant Software*. • Minor adjustments to ensure the document is consistent with Lattice Semiconductor's inclusive language policy. • Updated the content to better suit second person point of view.
Disclaimers	Updated boilerplate.
Acronyms in This Document	Added Address Shift Register and Data Shift Register to the list.
Introduction	Updated the Introduction section by updating the product features: • Supports both SEC (Single Error Correct) and SED (Single Error Detect) modes • Provides information on the Soft Error location via LMMI • Supports SEC without the need for external memory access
Functional Description	• Added Attributes Summary section. • Removed previous sections *2.5.1 SED/SEC Status Register 0 0x120B – 2.5.12 SED/SEC Interrupt Enable Register 0x1216*. • Added new sections 2.5.1 SEDC Status Register 0 0x0 – 2.5.10 SEDC Configuration Register 2 0x9. • Added header number for section 2.6 Sample Advance SEDC Applications. • Updated header numbers for sections 2.6.1 SEDC Integration with EBR ECC, 2.6.2 Automatic System Refresh, and 2.6.3 System Control on Error Event.
IP Generation and Evaluation	• Updated the existing steps in 3.1 Generating and Synthesizing the IP section to be more concise.

Section	Change Summary
	• Added step 5, and Table 3.1 to the existing steps in 3.1 Generating and Synthesizing the IP section. • Updated Figure 3.1 and Figure 3.3 in 3.1 Generating and Synthesizing the IP section. • Removed section *3.2 Generated Files and Top-Level Directory Structure* and replaced with section 3.2 Running Functional Simulation. • Updated Figure 3.5 – Figure 3.7 in 3.2 Running Functional Simulation section. • Removed *Note: Register addresses used in simulation are different from actual addresses detailed in Register Description section. Refer to the register list in testbench/tb_top.v in the generated IP for the list of register addresses used in simulation* from step 3 of 3.2 Running Functional Simulation section.
References	Added this section.

Revision 0.8, June 2023

Section	Change Summary
All	Initial preliminary release.

LATTICE SEMICONDUCTOR®

www.latticesemi.com