Avant-G/X SEDC Module IP - Lattice Radiant Software

User Guide

FPGA-IPUG-02232-1.0

December 2023

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 REDUDANCIES, 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

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
CRC32	Cyclic Redundancy Check – 32 bits
CRAM	Configuration Random Access Memory
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
SoC	System-on-a-Chip
ASR	Address shift register
DSR	Data shift register

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 via LMMI
- Supports SEC without the need for external memory access

2. Functional Description

2.1. Block Diagram

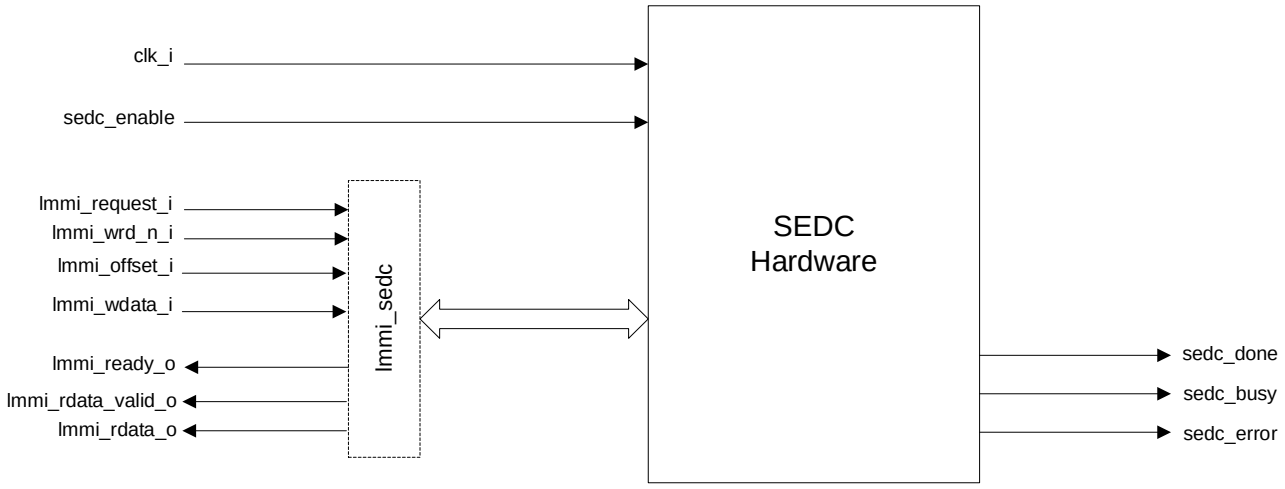

Figure 2.1. Generic SEDC Block Diagram

2.2. Functional Overview

The 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 parallels with ECC. Check the Signal Description section for information on which signals are to be connected.

2.3. Signal Description

Table 2.1. SEDC Module Ports

Pin Name	Direction	Width(Bits)	Description
clk_i	IN	1	System clock
rst_n_i	IN	1	Asynchronous active low system reset
sedc_enable	IN	1	When set to High, SED/SEC is enabled
sedc_done	OUT	1	SED/SEC is finished
sedc_busy	OUT	1	SED/SEC is running
sedc_error	OUT	1	SED/SEC has detected error
LMMI			
lmmi_request_i	IN	1	Start transaction
lmmi_wrrd_n_i	IN	1	Write = HIGH, Read = LOW
lmmi_offset_i	IN	8	Register offset, starting at offset 0
lmmi_wdata_i	IN	16	Write data
lmmi_ready_o	OUT	1	Ready to start new transaction
lmmi_rdata_valid_o	OUT	1	Read transaction is complete and lmmi_rdata_o contains valid data
lmmi_rdata_o	OUT	16	Read data

2.4. Attributes Summary

Table 2.2 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: These attributes are only for simulating errors and do not change the generated IP.

Table 2.2. Attributes Summary

Attribute	Selectable Values	Default	Description
CRAM Error Injection			
Inject 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.
Inject 2-bit non-correctable CRAM error	Checked, Unchecked	Unchecked	When checked, 2-bit non-correctable CRAM error is injected in the simulation.
Inject CRC CRAM Error	Checked, Unchecked	Unchecked	When checked, CRC CRAM error is injected in the simulation.
CRAM 1-bit Bit Error Location	0 – 1023	0	Indicates the target DSR bit location of 1-bit error.
CRAM 1-bit Frame Error Location	0 – 16383	0	Indicates the target ASR bit location of 1-bit error.
CRAM 1-bit Region Error Location	0 – 31	0	Indicates the target region location of 1-bit error.
INIT Error Injection			
Inject 1-bit correctable INIT error	Checked, Unchecked	Unchecked	When checked, 1-bit correctable INIT error is injected in the simulation. Error location is set in other attributes.
Inject 2-bit non-correctable INIT error	Checked, Unchecked	Unchecked	When checked, 2-bit non-correctable INIT error is injected in the simulation.
Inject CRC INIT Error	Checked, Unchecked	Unchecked	When checked, CRC INIT error is injected in the simulation.
INIT 1-bit Bit Error Location	0 – 15	0	Indicates the INIT register bit location of 1-bit error.
INIT 1-bit Offset Error Location	0 – 16383	0	Indicates the INIT register offset location of 1-bit error.
INIT 1-bit ID Error Location	0 – 4095	0	Indicates the INIT ID of 1-bit error.

2.5. Register Description

2.5.1. SEDC Status Register 0 0x0

Table 2.3. SEDC Status 0

Bits	Name	Access	Width	Reset
[15]	reg_init_errcrc	read-only	1	0x0
[14]	reg_init_errm	read-only	1	0x0
[13]	reg_init_err1	read-only	1	0x0
[12]	reg_init_err	read-only	1	0x0
[11:8]	*reserved*	read-only	4	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

Bits	Name	Access	Width	Reset
[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_init_errcrc
When set to 1, a CRC error in INIT registers has been detected.

reg_init_errm
When set to 1, an uncorrectable error in INIT registers has been detected.

reg_init_err1
When set to 1, a 1-bit error in INIT registers has been detected.

reg_init_err
When set to 1, an error in INIT registers has been detected.

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.

2.5.2. SEDC Status Register 1 0x1

Table 2.4. 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.

2.5.3. SEDC Status Register 2 0x2

Table 2.5. 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.

2.5.4. SEDC Status Register 3 0x3

Table 2.6. 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.

2.5.5. SEDC Status Register 4 0x4

Table 2.7. SEDC Status Register 4

Bits	Name	Access	Width	Reset
[15:4]	*reserved*	read-only	12	0x0
[3:0]	reg_init_bit_errloc	read-only	4	0x0

reg_init_bit_errloc

Indicates the INIT register bit position of the last 1-bit error.

2.5.6. SEDC Status Register 5 0x5

Table 2.8. SEDC Status Register 5

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

reg_init_offset_errloc

Indicates the INIT register offset of the last 1-bit error.

2.5.7. SEDC Status Register 6 0x6

Table 2.9. SEDC Status Register 6

Bits	Name	Access	Width	Reset
[15:12]	*reserved*	read-only	4	0x0
[11:0]	reg_init_id_errloc	read-only	12	0x0

reg_init_id_errloc

Indicates the INIT ID of the last 1-bit error.

2.5.8. SEDC Configuration Register 0 0x7

Table 2.10. 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

1 – When 1-bit error is detected, SEDC core automatically corrects errors and the scan will continue without stopping.
0 – When 1-bit error is detected, SEDC core stops scanning and waits for you to write to the *reg_sedc_resume* register.

2.5.9. SEDC Configuration Register 1 0x8

Table 2.11. 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	0x0

reg_clk_sedc_div
Indicates the divider value for SEDC clock.

2.5.10. SEDC Configuration Register 2 0x9

Table 2.12. 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
0 – SEDC scan is continuous.
1 – SEDC scan is one-time for both CRAM and INIT. 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.

2.6. Sample Advance SEDC Applications

The following sections describe the Advance SEDC Application 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.

2.6.1. 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.

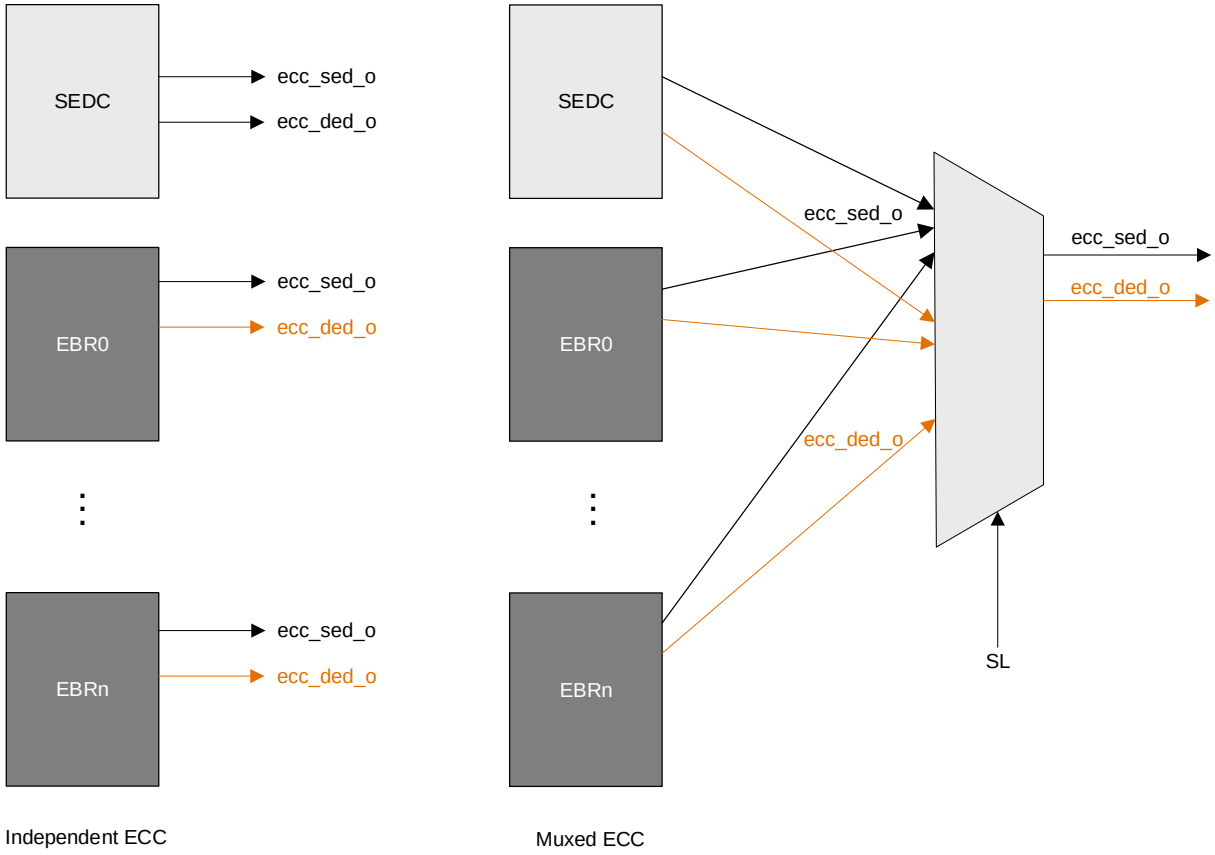

Figure 2.2. SEDC + EBR Sample Implementation

2.6.2. Automatic System Refresh

The SEDC can be interfaces to the global system reset, which restarts the entire system if it encounters a SER 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.

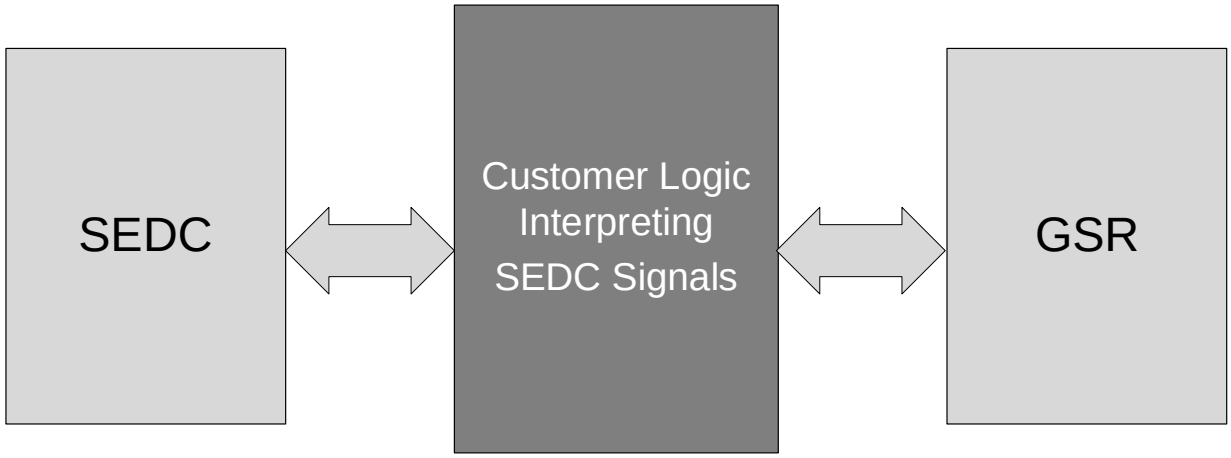

Figure 2.3. Automatic System Refresh Block Diagram

2.6.3. 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 these interrupts.

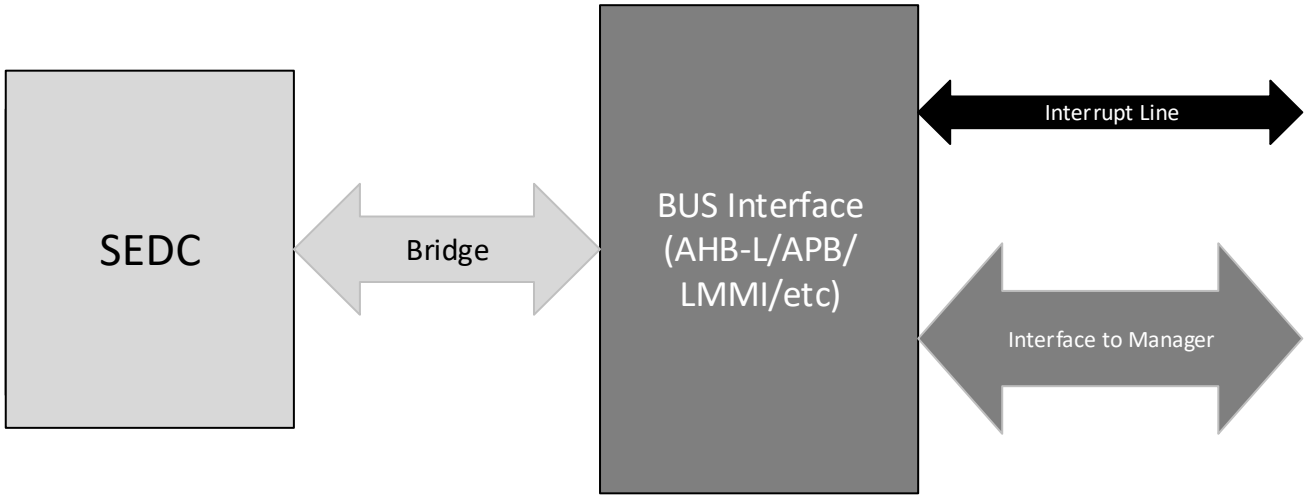

Figure 2.4. Bus Interface Block Diagram

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

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. From the Lattice Radiant Software, select the **IP Catalog** tab and select **IP on Local > Architecture Modules**.

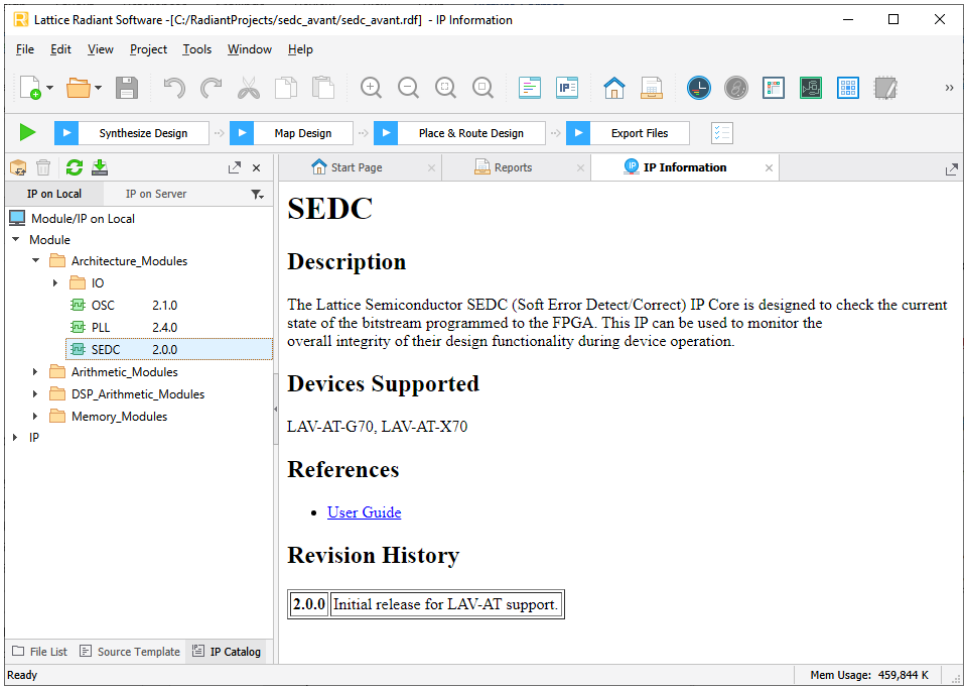

Figure 3.1. IP on Local, with SEDC Selected Under Architecture

2. Double-click on **SEDC IP** to open the **Module/IP Block Wizard**. Enter the values in the **Component name** and **Create in** fields and click **Next**.

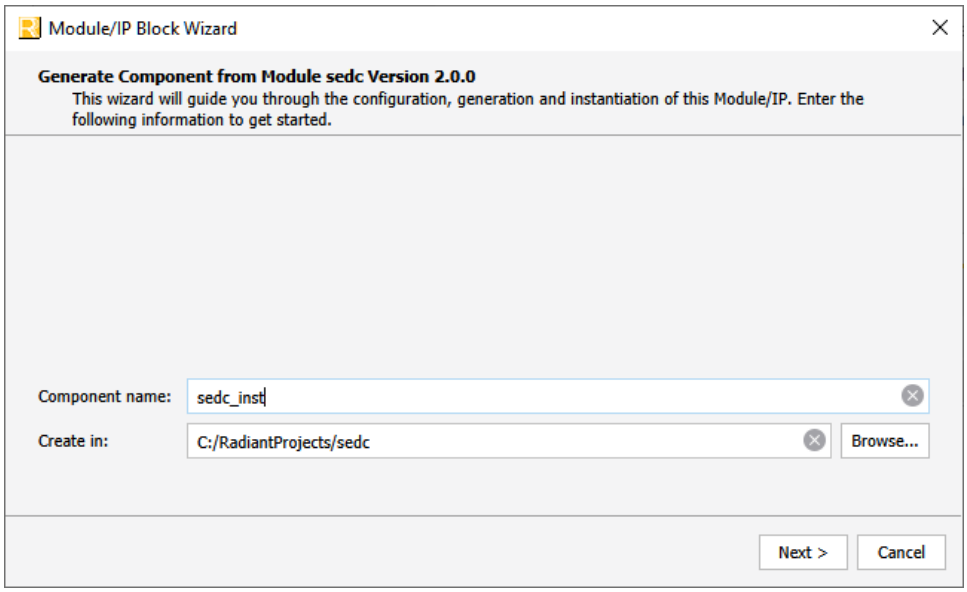

Figure 3.2. Module/IP Block Wizard

3. Update the configuration parameters as necessary as shown in Figure 3.3.

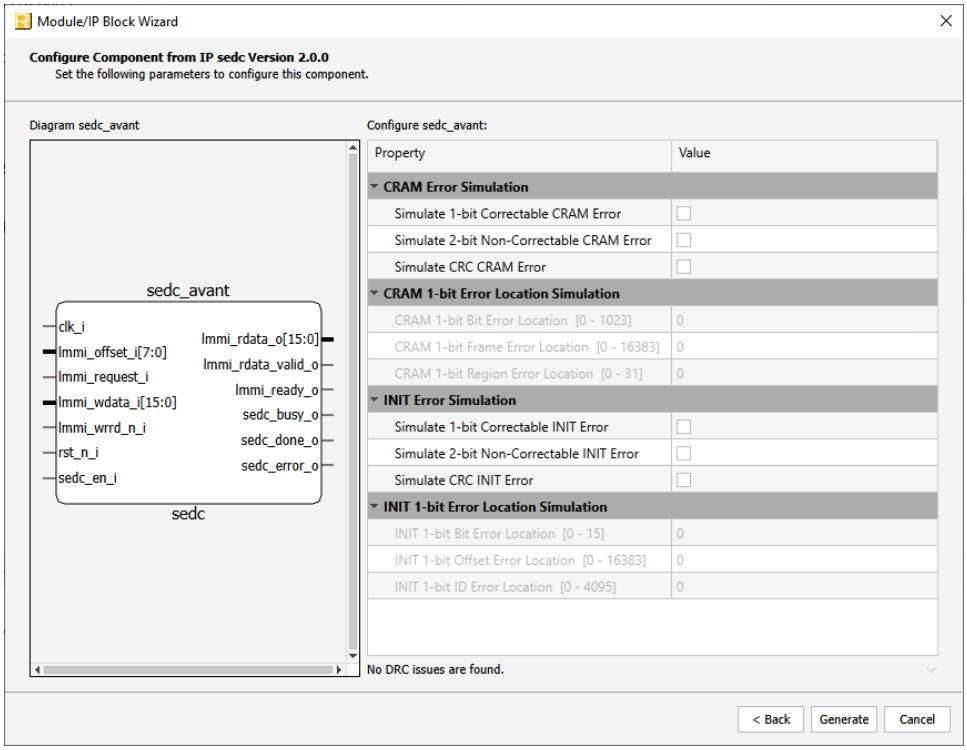

Figure 3.3. Configuration Window

4. Click **Generate**. After generation, the IP is automatically added to the active project as shown in Figure 3.4.

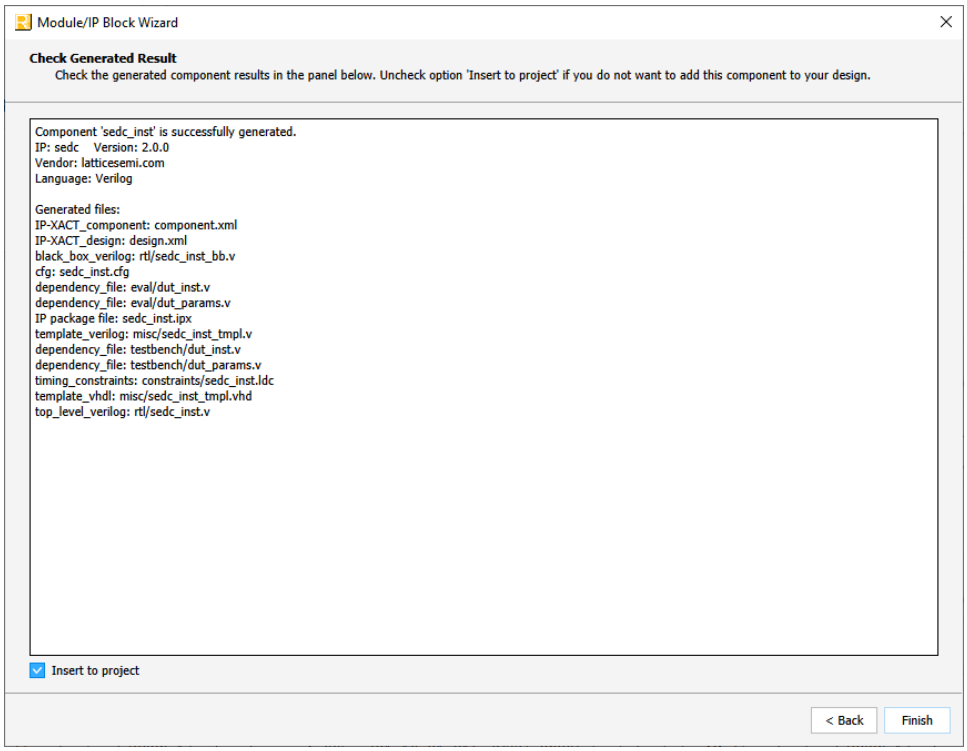

Figure 3.4. SEDC File List

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 3.2.

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 3.5.

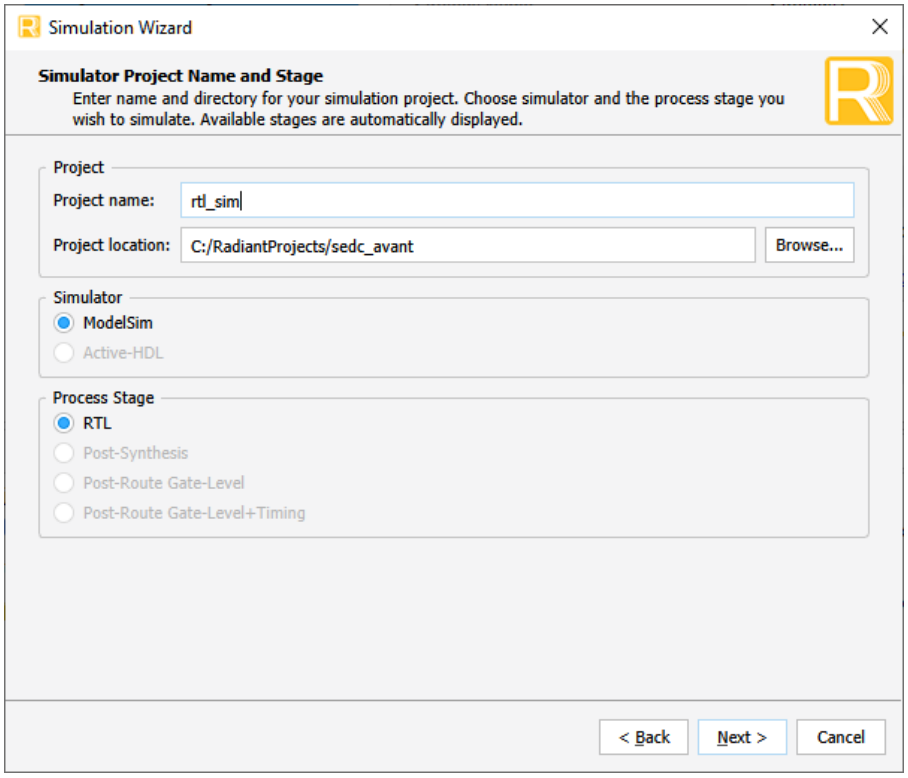

Figure 3.5. Simulation Wizard

2. Click **Next** to open the **Add and Reorder Source** window as shown in Figure 3.6.

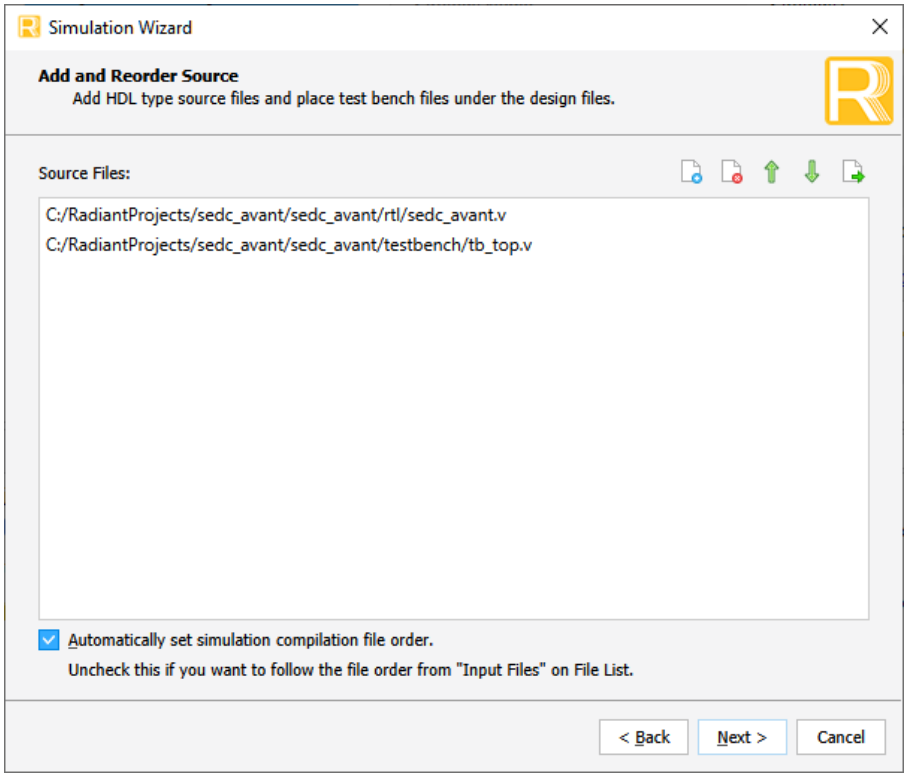

Figure 3.6. Adding and Reordering Source

3. Click **Next**. The **Summary** window is shown in Figure 3.7. Click **Finish** to run the simulation.

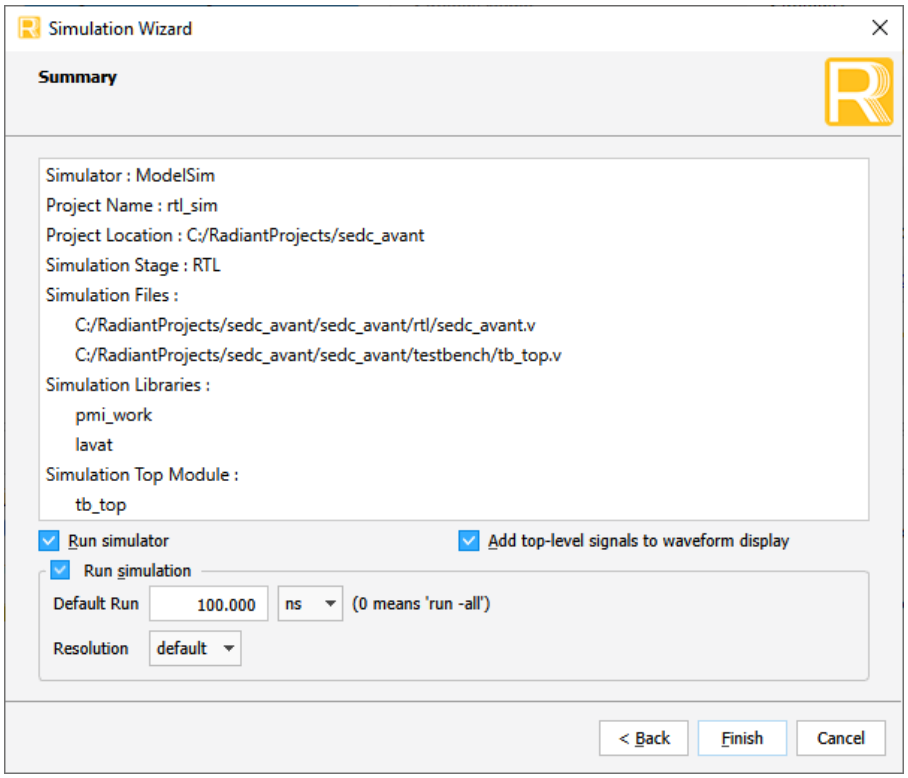

Figure 3.7. Simulation Wizard Summary

References

- Lattice Avant-G web page
- Lattice Avant-X web page
- Lattice Radiant Software web page
- Lattice Insights web page

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 1.0, 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. • 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.

www.latticesemi.com