SEDC Module - Lattice Radiant Software

User Guide

FPGA-IPUG-02066-1.0

December 2019

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 and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. 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. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
AHB-L	Advance High Performance Bus Lite
APB	Advance Peripheral Bus
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

1. Introduction

This document provides technical information about the SEDC (Single Error Detect/Correct Module) that is supported in CrossLink-NX™. 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
- Output information on Soft Error location
- Supports SEC without the need for external memory access
- Support for SEI (Soft Error Injector) tool
- Including support to target errors to a specific block of logic
- Flexible handling of SEC events -> Freeze I/O, GSR, and others
- Integrated handling of oscillator and other resources (including configuration access for Freeze I/O, GSR)

2. Functional Description

2.1. Block Diagram

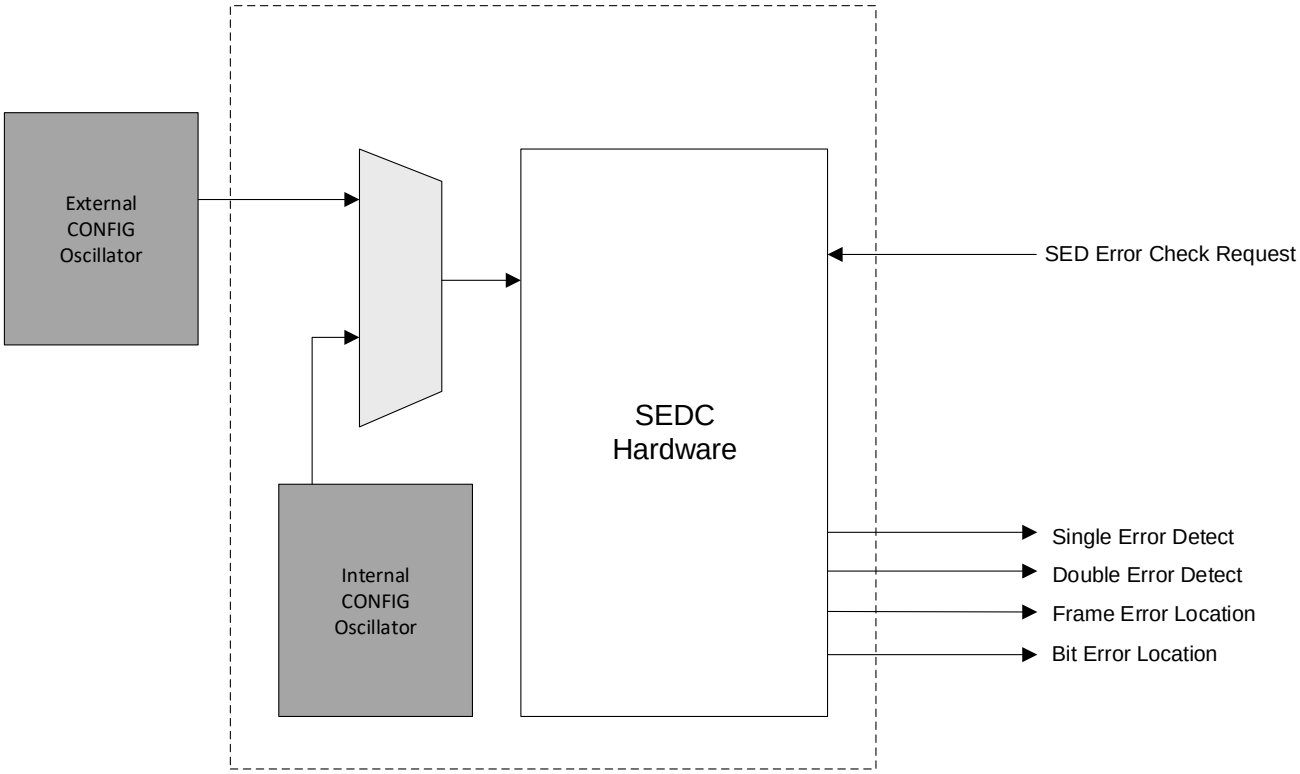

Figure 2.1. Generic SEDC Block Diagram

Note: The MUX control signal is the Use CONFIG_CLK Externally in the Attribute Summary.

2.2. Functional Overview

Soft Error Detection (SED) when enabled performs reads of DSR frames in user mode, and performs ECC (Error Correcting Code) Calculations for Soft Error Detection of SRAM content. Once an error is detected/corrected, notification is sent to the fabric and SED should resume. 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.

2.3. Signal Description

Table 2.1 SEDC Module Ports

Pin Name	Direction	Width(Bits)	Description
sed_en_i	IN	1	When HIGH, this would initiate the SED request once a single error rate has been generated.
sedc_cof_i	IN	1	When LOW, the SEDC engine stops once an error is detected. Setting to HIGH causes the SEDC to continue checking for errors, and could change the value of the frame ECC location, and DSR ECC location.
sedc_en_i	IN	1	When HIGH, this indicates that the SEDC IP is ready to accept SED requests.
sedc_mode_i	IN	1	When HIGH, this indicates that the IP would generate SED results, for as long as sedc_start_i is HIGH. When LOW, the IP would only generate SED results when sedc_start_i is triggered from LOW to HIGH.
sedc_start_i	IN	1	When triggered from LOW to HIGH, it starts generating SEDC results. If sedc_mode_i is HIGH, it would continuously generate SED results as long sedc_start_i is HIGH.
oscclk_sedc_i	IN	1	Input Clock of the SEDC IP. (Not shown if USE_CONFIG_RST = 1.)
sedc_rst_i	IN	1	Input RST pin of the SEDC IP. (Not shown if USE_CONFIG_RST = 1)
sedc_busy_o	OUT	1	Indicates if the SEDC IP is currently busy.
sedc_err_o	OUT	1	Indicates 1-bit SED error (correctable) detected. Latches until cleared by setting sedc_en_i to LOW.
sedc_errc_o	OUT	1	Indicate SED Error detected. Current error.
sedc_errcrc_o	OUT	1	Indicates CRC error detected after reading all frames.
sedc_errm_o	OUT	1	Indicates that more than one bit error detected within 1 BCH (DSR) frame; sticky until SED/SEC disabled.
sedc_frm_errloc_o	OUT	16	—
sedc_dsr_errloc_o	OUT	13	—

2.3.1. SEDC Timing Information

- SED can be started using sedc_start_i. Starting SED using sedc_start_i has two operational modes depending on the setting of sedc_mode_i.
- If sedc_mode_i is HIGH, SED runs continuously as long as sedc_start_i is HIGH.
- If sedc_mode_i is LOW, SED runs once every time there is a LOW to HIGH transition on sedc_start_i
- When sed_en_i is high and SED/SEC is enabled SEC runs when a single bit error is detected by the SEC logic, which flips the error bit by shift back through the DSR data for that frame.

2.3.2. SEDC Timing Diagrams

Figure 2.2 and Figure 2.3 show the different SEDC operations, which are when sedc_mode = 1 / sedc_mode = 0.

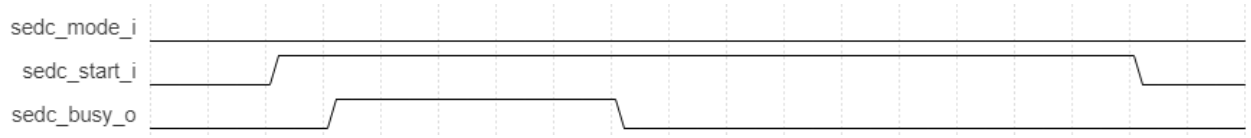

Figure 2.2. SEDC_Mode Deasserted

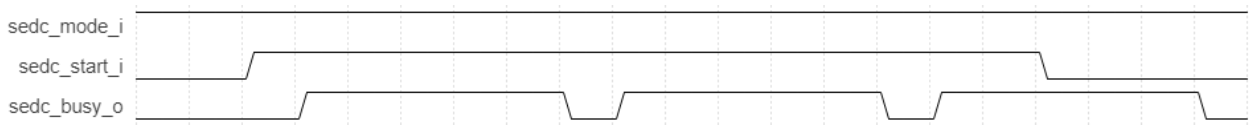

Figure 2.3. SEDC_Mode Asserted

2.4. Attribute Summary

Table 2.2 SEDC Attribute

Attribute Name	Values	Default	Description
Use CONFIG_CLK Externally	0 or 1	0	Instantiates the CONFIG_CLKRST_CORE module, to generate the required clock and signals to proper IP operation. Alternatively, this can be connected to an external CONFIG_CLKRST_CORE module which you can customize.

2.5. Sample Advance SEDC Applications

The following sections describe the Advance SEDC Application when combined with other IPs. This section is no longer part of the core SEDC IP, but serves as a guideline on possible ways that the SEDC can be used as a user solution.

2.5.1. 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 Memory Modules – Lattice Radiant Software User Guide (FPGA-IPUG-02033). 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.

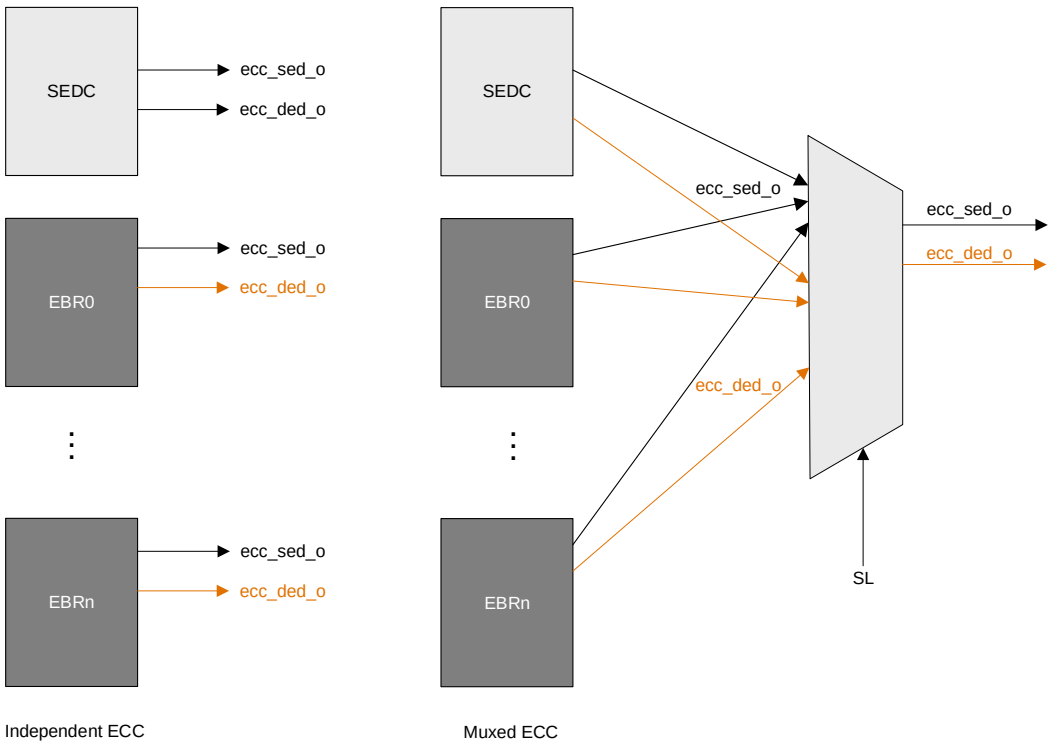

Figure 2.4. SEDC + EBR Sample Implementation

2.5.2. 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, please check the included *help* section in the Radiant software.

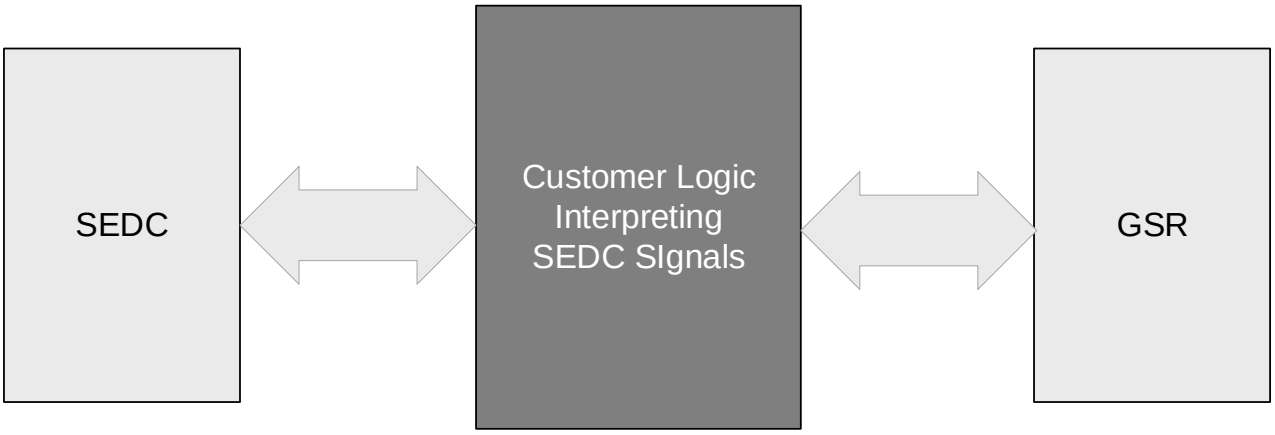

Figure 2.5. Automatic System Refresh Block Diagram

2.5.3. 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 master can be designed to respond these interrupts.

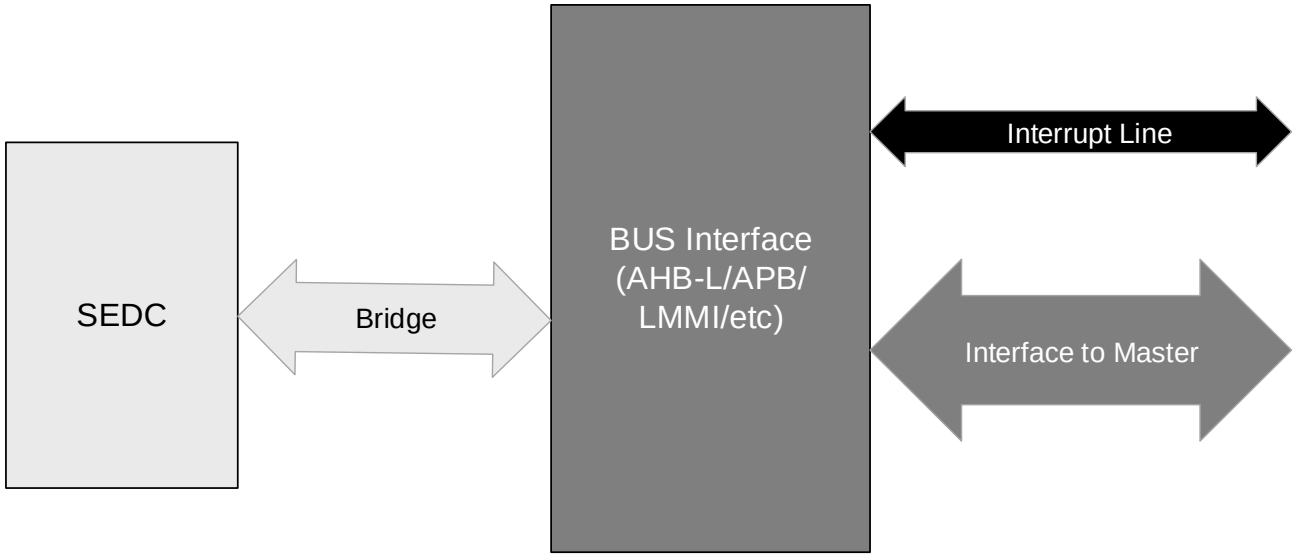

Figure 2.6. Bus Interface Block Diagram

These applications can also be combined, to provide you maximum flexibility in your 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**.

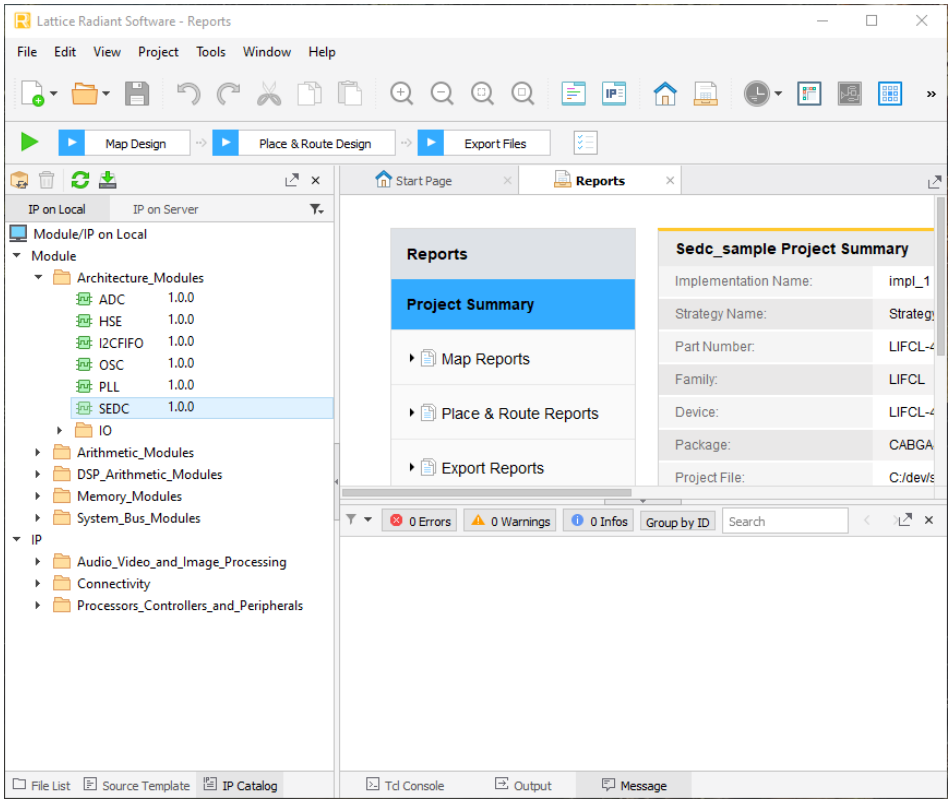

Figure 3.1. IP on Local, with SEDC Selected Under Architecture

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

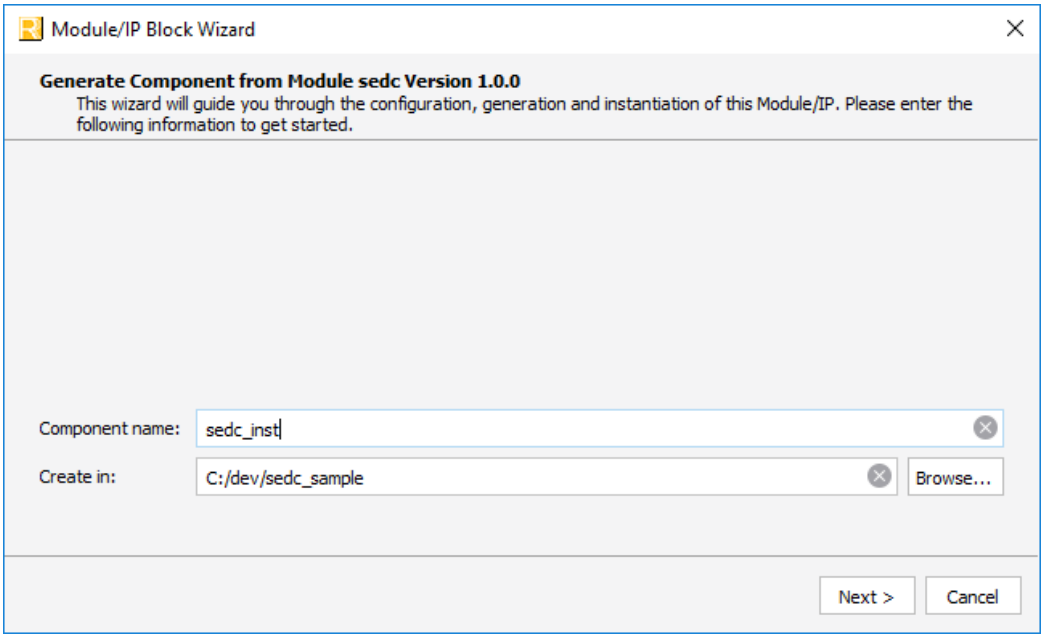

Figure 3.2. Module/IP Block Wizard

3. Click **Next**.
4. Update the configuration parameters as necessary as shown in Figure 3.3.

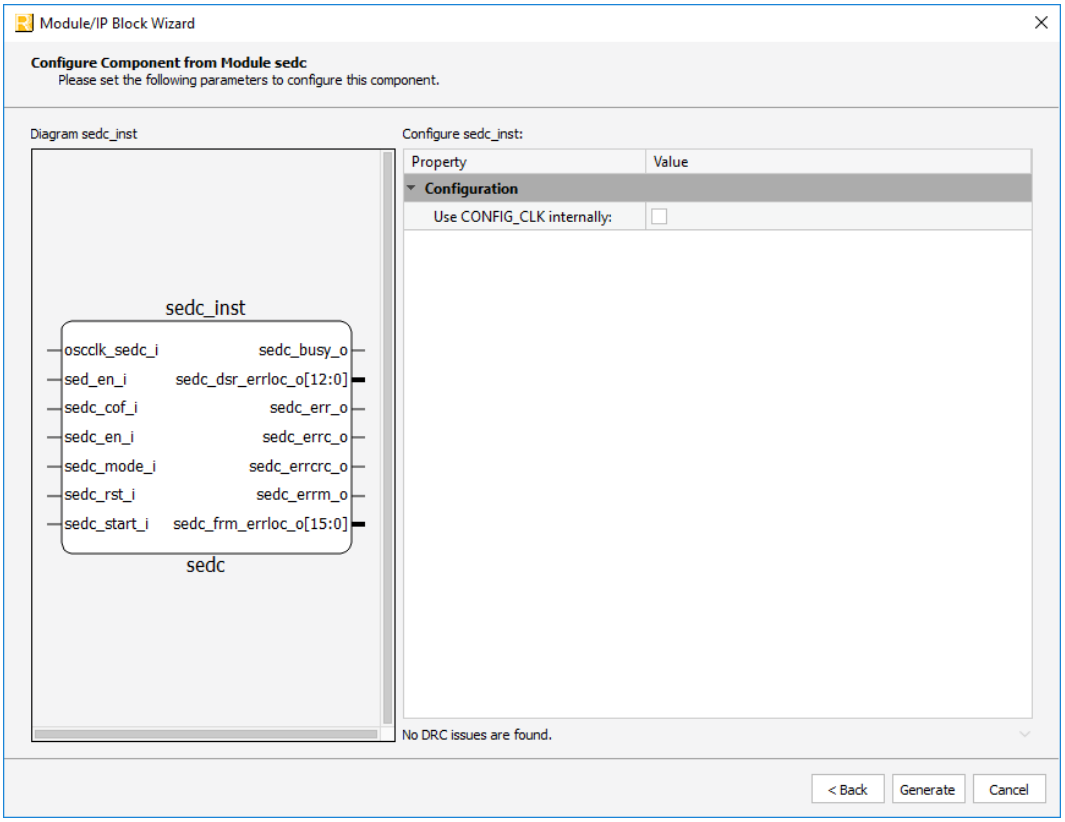

Figure 3.3. Configuration Window

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

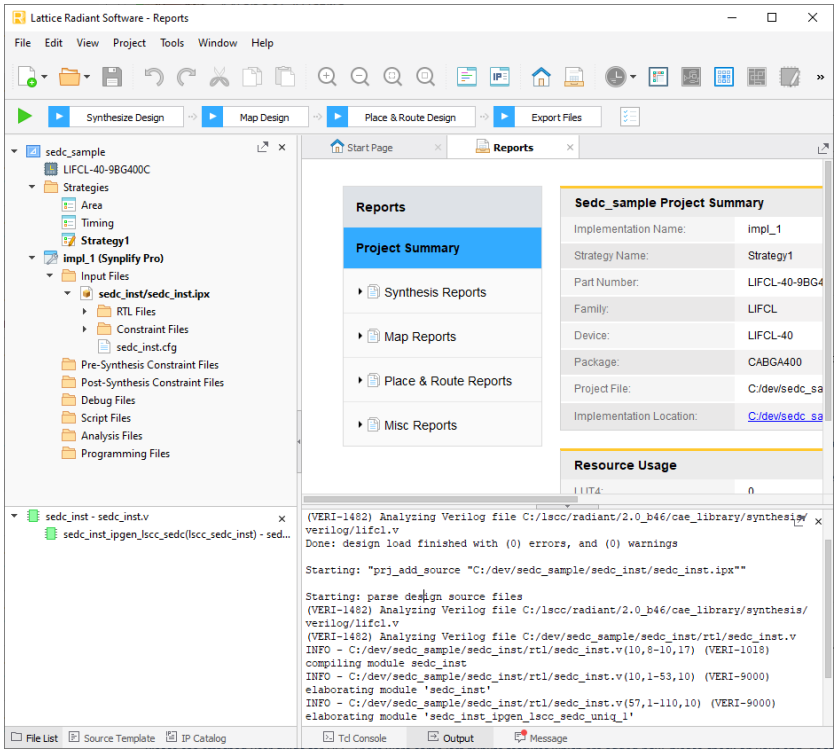

Figure 3.4. SEDC File List

3.2. Generated Files and Top Level Directory Structure

Table 3.1 SEDC Generated File Description

File	Sim	Synthesis	Description
<IP_name>.v	Yes	Yes	Top Level RTL file with the selected configuration. The main IP file
<IP_name>.cfg	—	—	This file contains the configuration options used to recreate or modify the core in the IP Platform.
<IP_name>.ipx	—	—	The IPX file holds references to all of the elements of an IP or Module after it is generated from the IP Platform user interface. The file is used to bring in the appropriate files during the design implementation and analysis. It is also used to re-load parameter settings into the IP Platform when the IP/Module is being re-generated.

3.3. Instantiating the Module

To instantiate the module, type the module name and instance name following the Verilog format. Alternatively, a sample instance can be seen in the /misc/ folder or the sample module instance in *<IP_name>_bb.v*.

Technical Support Assistance

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

Revision History

Document Revision 1.0, Lattice Radiant SW version 2.0, December 2019

Section	Change Summary
All	Initial release

www.latticesemi.com