Compa Family CPLDs Device Soft Error Recovery Application Guide

(AN03006, V1.2)

(12.07.2021)

Revisions History

Document Revisions

Version	Date of Release	Revisions
V1.2	12.07.2021	Initial release

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
DRM	Dedicated RAM Module

Table of Contents

Tables

Figures

Chapter 1 Overview

1.1 Introduction

This document serves as the application manual for the soft recovery of the Compa family CPLD devices launched by Shenzhen Pango Microsystems Co., Ltd. This document primarily introduces the function description, design architecture, interface definition, interface timing and reference designs of soft error recovery.

1.2 Main Functions

The main functions of soft error recovery are as follows:

- Performs error detection and correction multiple times.
- Independent error detection and correction as well as automatic error correction
- Soft error recovery without interrupting the current operating state of devices
- Error correction through internal flash or external SPI flash

1.3 Design Information

Table 1-1 Design Information

Soft Error Recovery Application	
Supported Devices	Compa family CPLD devices (excluding PGC1KL[1])
Supported User Interface	User defined
Provided Design Files	
Soft Error Recovery Reference Design	Verilog file
Constraint File	fdc file
Development Tools	
Design Tools	Pango Design Suite 2019.4 and later versions

Note

1: PGC1KL devices only support error detection, not correction.

Chapter 2 Function Description

The soft error recovery function of the Compa family includes 2 operation processes: error detection and error correction. Users can start error detection using I/O control or by user logic generating an error detection trigger signal; similarly, users can initiate error correction using I/O control or automatically correct errors based on error detection results.

2.1 Design Architecture

The soft error recovery function includes error detection and error correction.

The error detection function is implemented by reading back serial data from configuration memory for CRC calculation and checking the calculation results. A 32-bit CRC algorithm is used, with an initial CRC value of 0, and the readback data includes all data stored in configuration memory except for DRM initialization data. Each readback of 32-bit data is used for a CRC calculation, and the result is used for CRC calculation with the next 32-bit readback data, obtaining a CRC value through multiple iterations. The 'device's configuration control system determines if there is an error in configuration memory by comparing the value stored in the RBCRC register with the calculated value of readback data. If the CRC result of the readback does not match the value in the RBCRC register, an error signal is set high.

Correction is achieved through dynamic reconfiguration. Dynamic reconfiguration refers to refreshing configuration memory without resetting the device or exiting user mode. Therefore, the correction process does not interrupt the current operating state of the device. Dynamic reconfiguration can be completed by internal flash or external SPI flash.

Both error detection and error correction are controlled and implemented by 'on chip hardware module RBCRC, with minimal pins control required by the user. The logic block diagram is shown below.

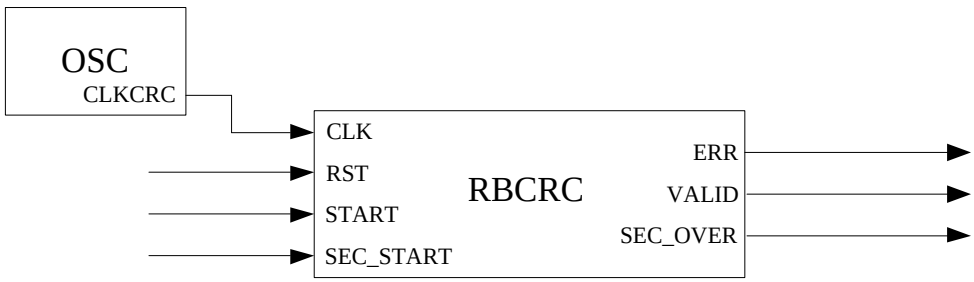

Figure 2-1 RBCRC Function Block Diagram

During design, users can use OSC by instantiating GTP_OSC_E2 and control the hardware module RBCRC by instantiating GTP_RBCRC, thereby completing soft error recovery operations.

2.2 GTP_OSC_E2 Interface List

Table 2-1 GTP_OSC_E2 Interface List

Port	Direction	Function Description
CLKCRC	Output	Output clock, providing the clock for GTP_RBCRC with a default frequency of 2.08MHz, which can be modified via PDS software
EN_N	Input	OSC output enable signal. When it is 1'b1, OSC is disabled; when it is 1'b0, OSC is enabled

2.3 GTP_RBCRC Interface List

Table 2-2 GTP_RBCRC Interface List

Port	Direction	Function Description
RST	Input	Reset signal, active-high, used to reset the ERR error flag and clear the readback CRC initial value. Module operation cannot be initiated when RST and START are both active; reset operations can only be performed before error detection starts and after it ends, not during the error detection process (from START being pulled high to VALID being pulled high)
CLK	Input	Clock input, with clock signal provided by CLKCRC of GTP_OSC_E2 through a hardware connection. The clock input range is 2.08-33.25MHz, all supported by GTP_OSC_E2
START	Input	Error detection starts, active-high
ERR	Output	Error detection error flag; high level indicates an error is detected
VALID	Output	Error detection end flag, high level lasting for one CLK cycle
SEC_START	Input	Error correction starts, active-high
SEC_OVER	Output	Error correction end flag, high level lasting for one CLK cycle

2.4 GTP_RBCRC Interface Timing

This section primarily introduces the timing for two types of soft error recovery.

➢ Manual error detection and automatic correction

Manually trigger the error detection process, and once an error is detected, generate the error correction start signal by logically judging the ERR and VALID signal states to complete the soft error recovery operation. The timing is shown below.

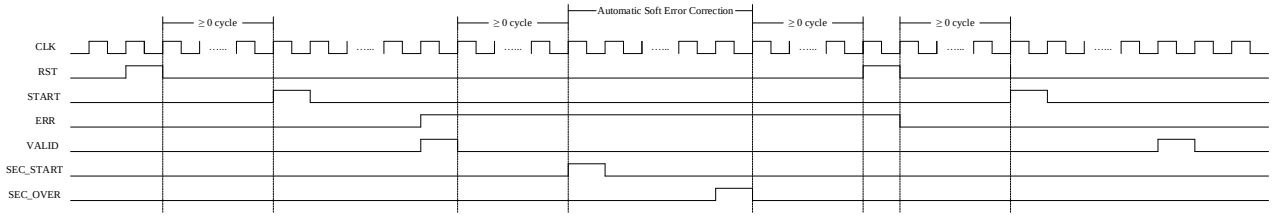

Figure 2-2 Error Detection and Automatic Correction Timing Diagram

➢ Cyclic error detection and correction

After the device is powered up, it remains in an error detection state. When the ERR output is high, it indicates that an error has been detected and the user can initiate error correction. The timing is shown below.

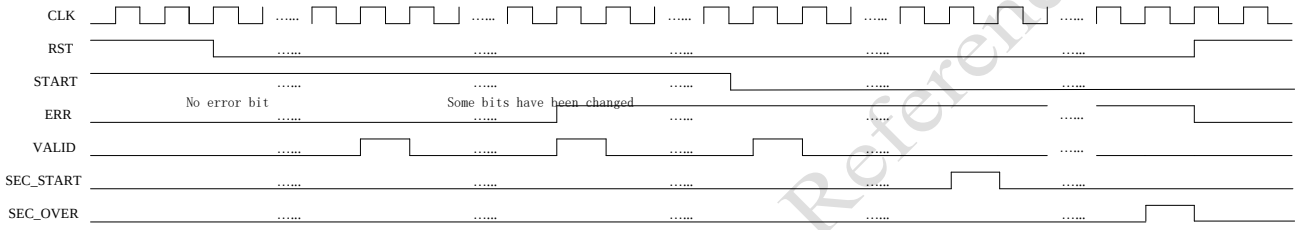

Figure 2-3 Cyclic Error Detection and Correction Timing Diagram

Chapter 3 Reference Designs

3.1 Reference Design Functions

This reference design implements the soft error recovery function featuring cyclic error detection and manual correction, as shown in Figure 3. During the soft error recovery process, there is a heartbeat indicator, which stops after the recovery is completed.

3.2 Reference Design Interface List

Table 3-1 Soft Error Recovery Interface List

Port	Direction	Function Description
sec_rstn	Input	System reset signal, active-low
dec_start_in	Input	Soft error detection input signal, active-high
sec_start_in	Input	Soft error correction input signal, controlled by a key, with a transition pulse generated
sec_err	Output	Soft error detection error indicator, active-high
sec_over	Output	Soft error correction completion indicator, active-high
sec_heartbeat	Output	Soft error recovery heartbeat signal. There is a heartbeat during error detection and correction, and no heartbeat if there is an error or it is not working

3.3 Resource Report

Table 3-2 Reference Design Resource Report

Device	LUT Used	FF Used	DRM Used	OSC Used	PLL Used
PGC7KD	25	22	0	1	0

3.4 Reference Design Board Validation

3.4.1 PDS Software Setup

Before compiling the project, check and enable the OSC's CLKCRC clock, which is disabled by default in PDS software. The steps are as follows:

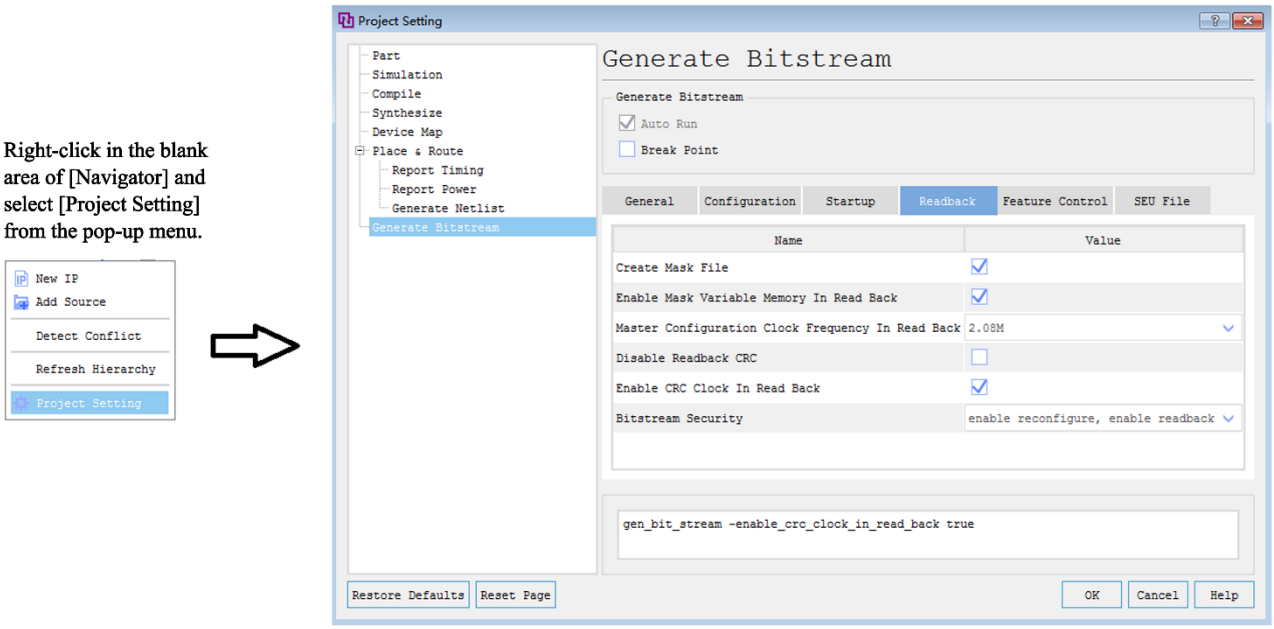

Figure 3-1Enabling CLKCRC Clock

Select [Enable CRC Clock In Read Back] to enable the CLKCRC clock; select [Enable Mask Variable Memory In Read Back] to prevent the initial data of Distributed RAM from participating in the CRC calculation during readback CRC check, thus avoiding false error detection.

Set the CLKCRC clock frequency, as shown in the following figure.

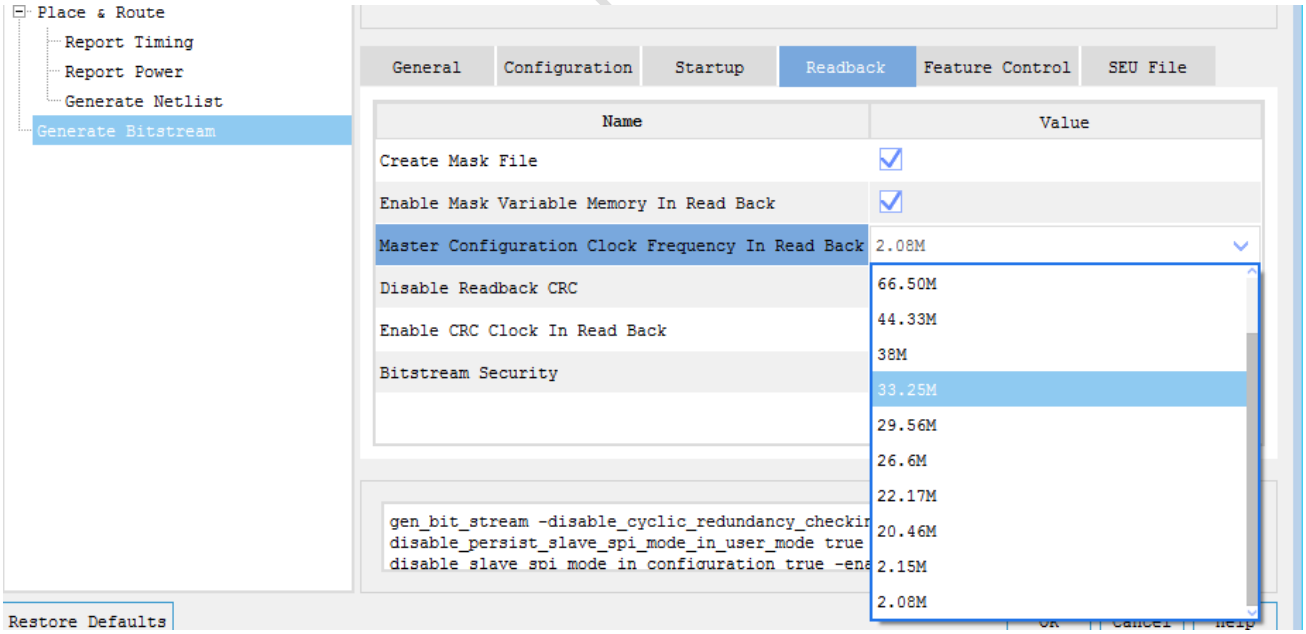

Figure 3-2 Setting CLKCRC Clock Frequency

3.4.2 Generate Error Bitstream

PDS software can inject a 1-bit error into the user bitstream to simulate user logic corruption caused

by external interference in actual applications. Similarly, in [Project Settings], use the SEU File to generate an error-injected bitstream. The operation interface is shown in the figure below.

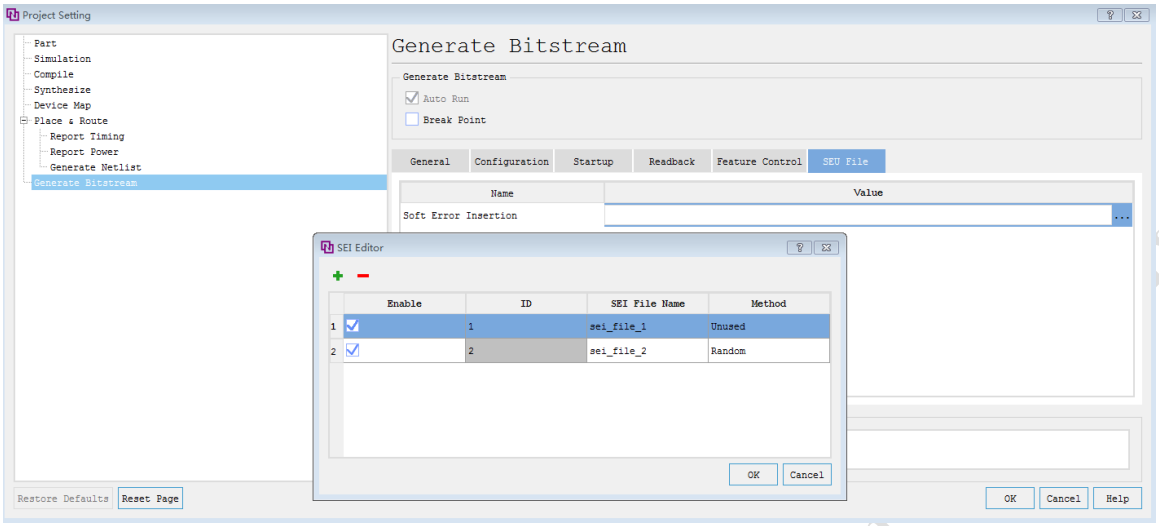

Figure 3-3 Generating an error-injected bitstream

After completing the settings, click [OK]. An error-injected bitstream is generated concurrently with the bitstream.

3. 4. 3 Verification Process for Internal Flash Soft Error Recovery

- Select MA for Boot Select of the feature control bit
- Program the bitstream into the internal flash
- Reset the device or reload the configuration memory to make it operating
- Pull the dec_start_in signal high to continue error detection, at which point the sec_heartbeat heartbeat signal is valid
- Pull the dec_start_in signal low to load the error-injected bitstream, simulating a bitstream error
- Pull the dec_start_in signal high again to start error detection, with sec_err high and the sec_heartbeat heartbeat signal low
- Pulling the dec_start_in signal low and initiate the sec_start_in for soft error correction, during which the sec_heartbeat resumes beating and stops after error correction is complete

3. 4. 4 Master SPI Mode Verification Process

- Select MSPI for Boot Select of the feature control bit
- Program the bitstream into the external SPI flash
- Ensure the device operates properly, other operation processes are consistent with the process of internal flash soft error recovery verification

Chapter 4 Application Considerations

- GTP_OSC_E2 also has an output clock CLKOUT, which provides clock signal for user logic. It is not used in this application and can be left unconnected
- The CLK of GTP_RBCRC can be left unconnected, or 1 wire-type signal can be defined to connect with the CLKCRC of GTP_OSC_E2, but it must not be connected to other clock signals
- The supported clock frequencies for the CLK of GTP_RBCRC are not continuous, which are determined only by the output frequencies of CLKCRC from GTP_OSC_E2, and not all selectable frequencies are listed in the PDS software
- To inject an error, the dec_start_in signal must be pulled low; otherwise, error injection will fail
- Error detection and error correction cannot occur simultaneously; during correction, the dec_start_in signal must be pulled low
- When using the master SPI mode, ensure that VCCIO2 is powered up before VCC
- After error correction is completed, GTP_RBCRC must be reset to clear the error detection error flag

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage licence, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its family products at any time without prior notice.
4. The information contained in this document is intended to assist users in resolving application-related issues. While we strive for accuracy, we cannot guarantee that the document is entirely free from flaws. Should any functional abnormalities and performance degradation arise due to deviation from the prescribed procedures outlined herein, our company will neither be held liable nor concede that such issues stem from product deficiencies. The solutions presented in this document are just one of the feasible options and cannot cover all application scenarios. Consequently, if users encounter functional abnormalities or performance degradation despite adhering to the prescribed procedures outlined herein, we cannot assure that such issues are indicative of product deficiencies.