# Verification methodology of a multi-mode radiation-hard high-speed transceiver ASIC

Adithya Pulli  Iraklis Kremastiotis  Szymon Kulis

CERN, Geneva, Switzerland

## Introduction

lpGBT is a versatile radiation-hard ASIC designed to provide high-speed multi-purpose bidirectional links for high energy physics experiments. Some of its most important functional features which result in high verification complexity include:

- High-speed bidirectional links with configurable Forward Error Correction (FEC) codes and uplink data-rates
- Programmable electrical links with configurable data rates and link counts
- Programmable fixed-phase and phase-shifted clocks
- Tolerance to Single Event Upsets (SEUs) and Single Event Transients (SETs)
- Robust operation through watchdog and timeout actions
- Multiple boot options - Fuses, ROM, I2C
- Multiple configuration interfaces - Fuses, I2C, high-speed link, electrical link
- Programmable I2C master ports, GPIOs and reset generator

The verification strategy of the lpGBT was based on Coverage Driven Constrained Random Verification (CDCRV) at top level using Universal Verification Methodology (UVM). CDCRV enables tests to more efficiently explore the design space because of random nature of the stimuli. UVM facilitates the development of modular test cases at higher abstraction level which in turn improves the verification efficiency.

## UVM-based lpGBT Verification Environment

UVM environment is the central backbone of lpGBT verification effort which enables efficient test writing. The key design considerations include modularity, a high level of abstraction and randomization.

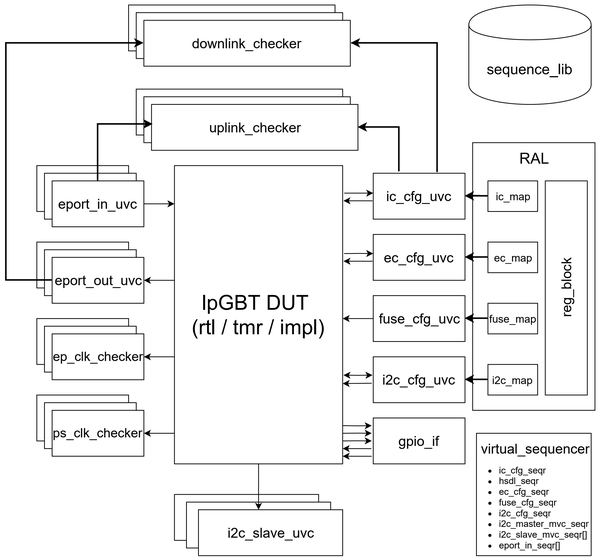



Figure 1. lpGBTv1 UVM environment.

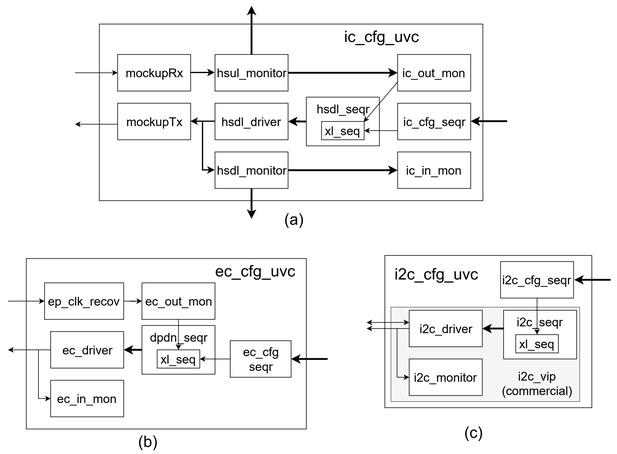



Figure 2. (a) IC configuration UVC, (b) EC configuration UVC, (c) I2C configuration UVC

Highlights of the lpGBT UVM environment include:

- Unified accesses to configuration space through Register Abstraction Layer (RAL) where multiple address maps are used to support various configuration access mechanisms
- SEE aware checkers which conditionally tolerate data upsets during SEE injection
- Layered agents which enable easier multiplexing of data and noise transactions
- Partially synthesizable drivers which are reused in post-silicon validation
- Supports RTL, triplicated RTL (rtl-tmr) and netlist versions of DUT

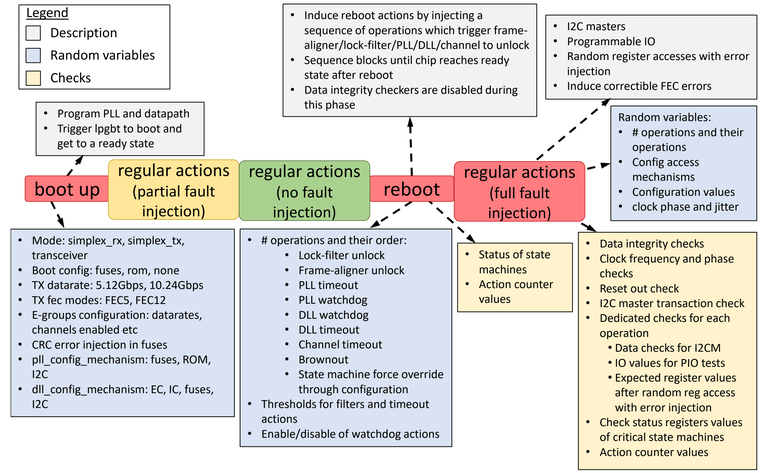
## Constrained Random Tests

The lpGBT test suite includes three highly random top-level UVM tests:

1. *lpgbt_rand_test* which randomly exercises most of the features of lpGBT
2. *lpgbt_reg_rw_test* which exhaustively verifies register instantiation and connectivity from all configuration ports
3. *lpgbt_fuse_burning_test* which exercises and verifies fuse burning flow

These UVM tests are supplemented by other standalone directed tests that target specific functionalities of the chip. Of the three UVM tests mentioned above, *lpgbt_rand_test* (Figure 3) is the most complex and most randomized. This test exercises, in a highly random fashion, typical use cases of lpGBT as indicated in boot up and regular actions in Figure 3. Additionally, it also exercises all recovery mechanisms (watchdog/timeouts/etc) to ensure that the chip always goes back to an operational state after prolonged error conditions. The methodology of exercising most of the features in one complex test enabled quick and efficient hitting of design corner cases which would be hard to imagine and exercise in a directed test.

The *lpgbt_rand_test* also includes SEE (both SEU and SET) injection. SEEs are injected into the DUT using *uvm_hdl_force* and *uvm_hdl_deposit* directives from a *uvm_sequence*. The list of nodes for fault injection is extracted using Xcelium Fault Set Generator (XFSG) tool and then categorized into triplicated and non-triplicated nodes based on design hierarchy. SEEs are injected one at a time, at random moments in time spaced by at least 50ns between two consecutive SEEs. Depending on the test phase, SEEs are injected either in all or none or only triplicated nodes as summarized in Figure 3.




| | Full injection | Partial injection | No injection |
|---|---|---|---|
| Fault injection scope | All nodes | Only triplicated nodes | No faults injected |
| Checkers | Enable SEE tolerance in end-to-end data checkers and clock phase & frequency checkers | All checkers enabled | All checkers are enabled |
| Fault injection frequency | Fclk/2 | Fclk/2 | 0Hz |

Figure 3. *lpgbt_rand_test* along with fault injection scheme

## Verification Signoff

Twenty person-months were spent on the verification of lpGBTv1 spread over ten calendar months between three engineers. The design and verification effort of lpGBT made use of a simplified version of git flow to organize the project. The build infrastructure of the design repository is based on tmake (build system for fault-tolerant ASICs). vManager was extensively used to manage regressions where each test is run multiple times with different random seeds. To improve execution efficiency and reduce the overall run time of tests, all tests were executed on Oracle Grid Engine (OGE) based batch system. This concurrent execution of tests reduced the overall regression run time by 60x compared to a simpler setup without parallel execution. In the course of lpGBTv1 verification activity, more than a hundred thousand simulations were run with this infrastructure. Various metrics collected to assess the progress of verification and determine sign-off quality, along with final scores are summarized in Table 1.

| Metric | Score | Remarks |
|---|---|---|
| Regression pass-rate | 98.24% | All the failing tests either point to a known testbench/modelling issue or a batch system failure |
| Code coverage | 94.35% | Collected in functional simulations without fault injection (SEE recovery mechanisms are not activated) |
| Functional coverage | 100% | |
| Fault coverage | 100% | SEUs per node = 2067, SETs per node = 214 |
| Bugs | 60 | 45 functional bugs and 15 SEU/SET vulnerabilities were discovered and fixed |

Table 1. Signoff metrics and final status