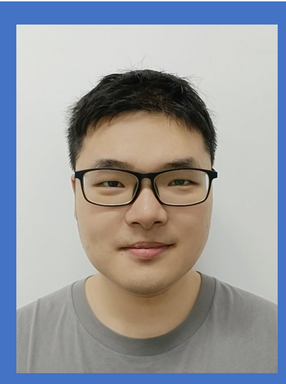
A Fully Reconfigurable Pipelined Architecture for FPGA-based Parallel PRBS Test Pattern Generators

Chengyang Zhu, Kezhu Song, Dongwei Zou, Zhuo Chen
State Key Laboratory of Particle Detection and Electronics, University of Science and Technology of China, Hefei, China
Email: cyzhu@mail.ustc.edu.cn



ABSTRACT

Serial links are widely used for data transfer in Data Acquisition (DAQ) Systems of High Energy Physics (HEP) experiments. Pseudo-Random Binary Sequences (PRBS) has seen wide application in high-speed serial wireline communication systems as test patterns for link characterization and testing. A flexible architecture for FPGA-based PRBS generators is proposed, with a focus on high throughput and full reconfigurability. In order to meet the demands of increasing data rates, the proposed architecture employs a parallel datapath with high scalability. The architecture is designed to be fully parametric and reconfigurable, which allows dynamic reconfiguration of all parameters of the PRBS generator on the fly, including polynomial, seed and output width. Reconfiguration of the parameters is achieved by simply writing to corresponding registers, without the need to re-synthesize or re-configure the FPGA device. A built-in bootstrap logic is used to convert parameter register values to internal states that are fed to the datapath to generate the output bit sequence. The datapath is pipelined to facilitate optimized timing performance on FPGA devices. The proposed design can be utilized to characterize serial link performance under a great variety of different test patterns rather than several selected ones, providing broader insights. The architecture is implemented in CHISEL and verified on an Intel Agilex-7 FPGA and a 106.25-Gbps serial link, where results show promising performance and scalability.

INTRODUCTION

PRBS is widely used as test patterns for serial link testing due to its structural simplicity and similar statistical characteristics to a truly random sequence. As shown in Fig. 1, a typical test setup comprises a PRBS generator that generates the test sequence to transmit, a PRBS checker that checks the received sequence, and a pair of SerDes transmitter and receiver with passive channel that make up the serial link.

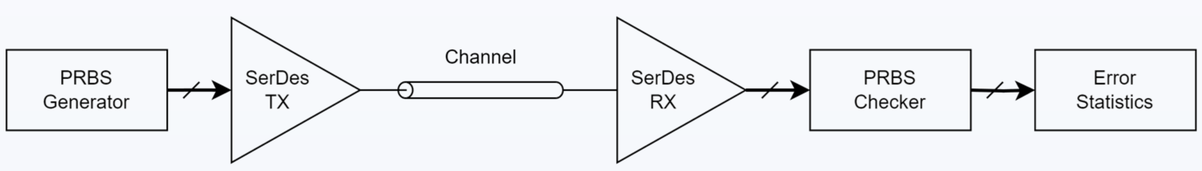


Fig. 1. Application diagram of PRBS in link error testing.

The structure of PRBS generators, as shown in Fig. 2, is also known as a Linear Feedback Shift Register (LFSR). The input taps of the feedback path is determined by the polynomial of the corresponding finite field. Parallelization is necessary to achieve higher throughput and meet the requirement of increasing data rates.

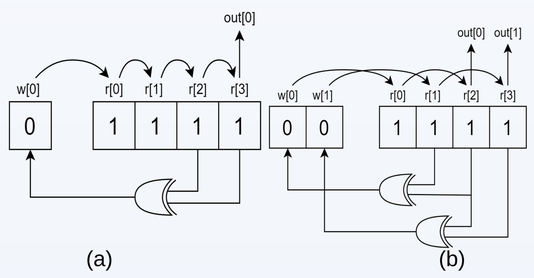


Fig. 2. Structural diagram of serial(a) and parallel(b) PRBS generators.

The polynomial is a key parameter of a PRBS. Different polynomials yield different sequences, as shown in Fig. 3, which might differ drastically in terms of statistical characteristics including baseline wander and clock content.

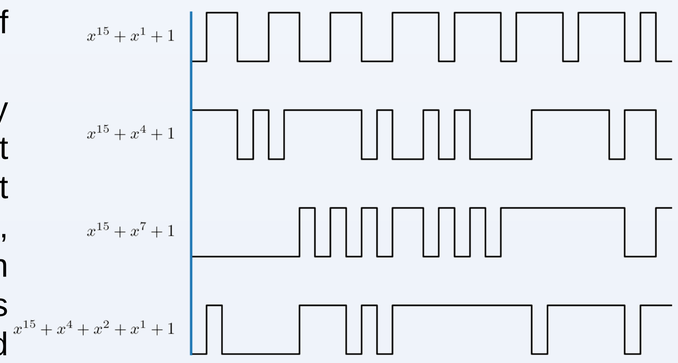


Fig. 3. Waveforms of PRBS15 with different polynomials.

A common practice when choosing polynomial is to choose the ones with the least non-zero terms to minimize tap count and reduce resource utilization. However, there is no guarantee that such polynomials come with the optimal statistical characteristics. To improve test coverage, it is desirable that the polynomial can be dynamically reconfigured at run time. Also, it is beneficial to have reconfigurable parallelism width to accommodate different link rates.

In this contribution, the proposed architecture is designed with the following considerations:

- Fully parametric and reconfigurable. All parameters of the PRBS can be modified at run time, including order, width, polynomial and seed.
- Pipelined datapath for optimal timing performance.
- Scalable parallel design supporting large widths for high-throughput system.
- Built-in bootstrap unit eliminates the need for any external pre-compute process.

PROPOSED ARCHITECTURE

There are three major parts in the proposed architecture: the core, the sequence generator and the bootstrap unit. The architectural block diagram is shown in Fig. 4.

- The core is composed of a set of registers that maintain the state of the PRBS and the next-state generation logic to update the registers per cycle.
- The sequence generator takes as input the core state registers and generate the parallel output sequence for each cycle.
- The bootstrap unit is used solely for reconfiguration. It is activated after a reconfiguration request is asserted by the user and serves the purpose of generating intermediate control signals for the core and the sequence generator logic. With the bootstrap unit, the user only needs to provide the desired parameters like polynomial and width and does not need to compute the transformation matrix themselves. The bootstrap unit is also responsible for preloading the core pipeline stages before a normal cyclic operation is ready.

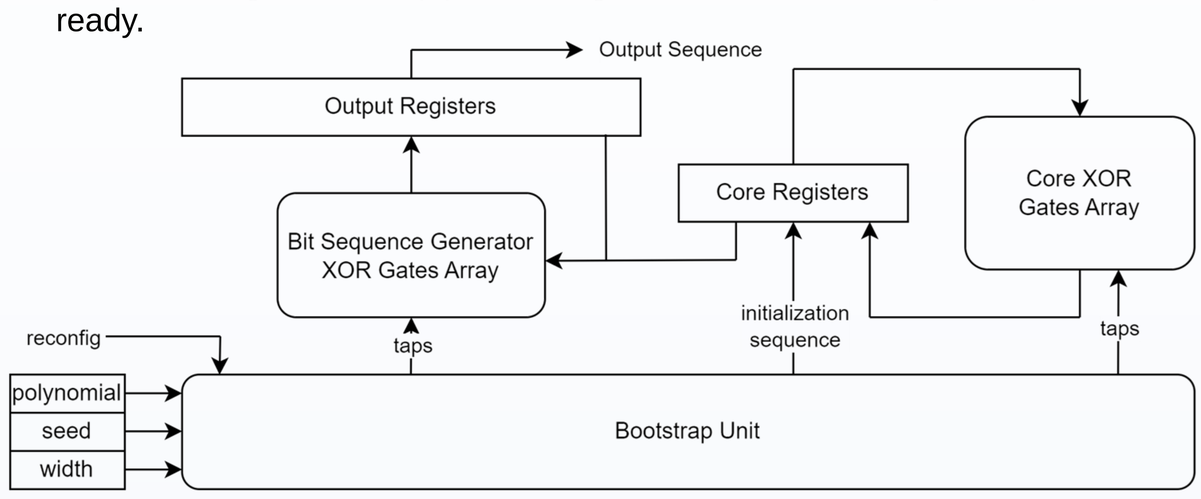


Fig. 4. Block diagram of proposed architecture.

The key to achieving dynamic reconfigurability is by implementing dynamic XOR gates, which are essentially wide XOR gates with input masks. Such dynamic XOR gates are the basic building blocks of both the core and the sequence generator logic.

In order to address the timing issue that arises with large input widths of gates, the XOR gates are reorganized into pipelined tree-structure that exploits the strengths of FPGA logic elements. The structural diagram is shown in Fig. 5.

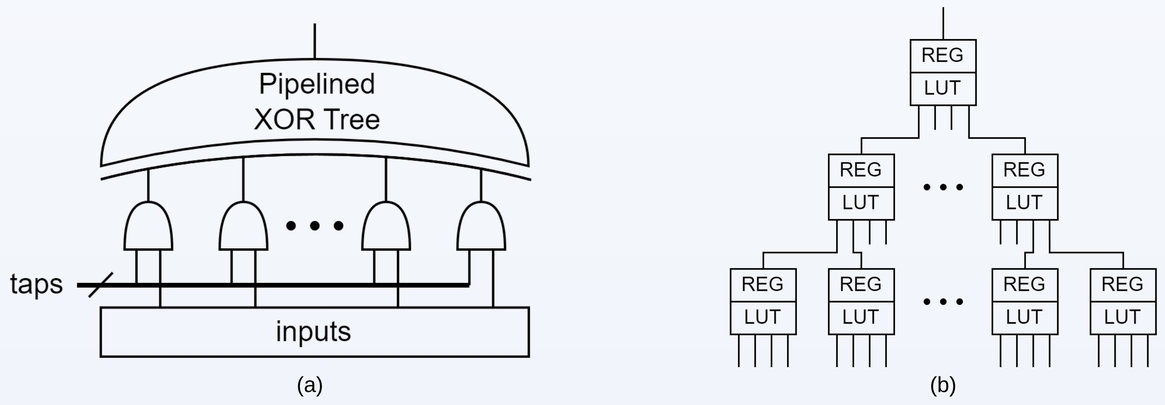


Fig. 5. (a) Dynamic XOR gate (b) Efficient pipelined tree-structure for implementing wide XOR gates.

A self-synchronizing PRBS checker can be built upon an existing PRBS generator by adding a error comparator, a synchronization Finite State Machine (FSM), and an extra multiplexed state register update path, as shown in Fig. 5.

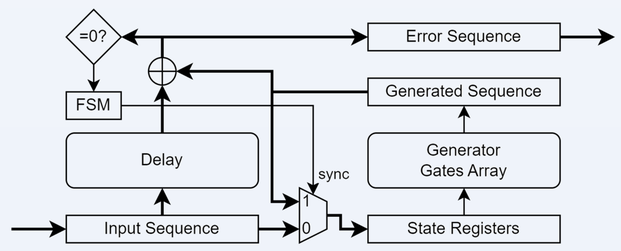


Fig. 5. Structural diagram of a self-synchronizing PRBS checker

The checker is able to recover the seed from received PRBS sequence (possibly with errors), eliminating the laborious process of manually synchronizing the PRBS generator at TX side and the PRBS checker at RX side.

DESIGN VERIFICATION

The design is synthesized and tested on an Intel Agilex-7 AGIB027R31B1E1VAA FPGA device. The resource utilization is listed below.

maxOrder	8	16	32	64
ALMs	1064	2040	4533	11639
Combinational ALUTs	1837	3522	7657	18552
Registers	1770	3725	7710	18819
Fmax (MHz)	575.71	554.32	552.18	514.93
Max. Throuput (Gbps)	147.38	141.91	141.36	131.82

TABLE I. Resource and timing of proposed architecture (maxWidth=256)

maxWidth	16	32	64	128	256
ALMs	1841	2017	2377	3099	4533
Combinational ALUTs	2294	2647	3374	4801	7657
Registers	2410	2767	3476	4889	7710
Fmax (MHz)	551.88	571.10	559.91	574.05	552.18
Max. Throuput (Gbps)	8.83	18.28	35.83	73.48	141.36

TABLE II. Resource and timing of proposed architecture (maxOrder=32)