FPGA Targeted Design: PRBS-15 Block and Pattern Detector

This project involves the design of a Pseudo-Random Binary Sequence (PRBS-15) generator and a Pattern Detector using Verilog.

Table of Contents

Project Description

This project provides an FPGA-targeted design in Verilog, containing:

- A PRBS-15 block that outputs a custom pseudo-random sequence.
- A pattern detector block that identifies when a specific pattern repeats correctly n times.
- A serial-to-parallel module to handle resource limitations on the FPGA.

Deliverables

1. **Custom PRBS-15 Block**:

    - Takes a 4-byte pattern as input.
    - Outputs these 4 bytes after each other n times.
    - Implements a custom PRBS equation with XOR of bit 13 and bit 14.

2. **Pattern Detector**:

    - FSM-based.
    - Detects if the pattern is correctly repeated n times.
    - Raises a flag once the pattern is found.

3. **Serial to Parallel Converter**:

    - Handles limited FPGA resources by receiving 8-bit inputs serially and converting them into a 32-bit parallel output.

4. **Top Module**:

    - Integrates the PRBS-15, Pattern Detector, and Serial-to-Parallel blocks.

5. **Testbenches**:

   - Provided for each block and the top-level module to validate the functionality.

Design Overview

PRBS-15 Block

This module generates a pseudo-random binary sequence using a custom PRBS-15 equation. It accepts a 32-bit input pattern and repeats it n times before applying a shift-left operation with XORing of bit 13 and bit 14 to generate the PRBS sequence.

Ports:

- **clk**: Clock signal.
- **rst_n**: Active-low reset signal.
- **pattern_in**: 32-bit input pattern.
- **n_repeats**: 8-bit input specifying the number of times to repeat the pattern.
- **prbs_out**: 8-bit output PRBS sequence.

Internal Registers:

- **pattern_counter**: Count the patterns to reach **n_repeats**.
- **byte_counter**: Count the 4 Bytes for every pattern.

Pattern Detector

This block uses a finite state machine (FSM) to detect if a given pattern is repeated n times in the input data stream. The FSM transitions between IDLE, MATCHING, and FOUND states to ensure that the pattern is correctly detected.

Ports:

- **clk**: Clock signal.
- **rst_n**: Active-low reset signal.
- **data_in**: 8-bit input data stream from the PRBS block.
- **pattern**: 32-bit input pattern to detect.
- **n_repeats**: 8-bit input specifying the number of times the pattern should repeat.
- **pattern_found**: Output flag that is asserted when the pattern is detected.

Serial to Parallel Converter

To account for FPGA resource constraints, this module allows serial data input of 8 bits, which are then converted into a 32-bit parallel output. The serial-to-parallel converter accumulates data over multiple cycles before outputting the full 32-bit data.

Ports:

- **clk**: Clock signal.
- **rst_n**: Active-low reset signal.

- **data_serial**: 8-bit serial input.
- **data_parallel**: 8-bit parallel output.
- **finish**: Signal asserted once the data has been completely received and converted to parallel.

Top Module

The top module integrates the PRBS-15 block, Pattern Detector, and Serial to Parallel Converter. It connects the prbs_out from the PRBS block to the data_in of the Pattern Detector, ensuring that the detector checks the PRBS sequence for repeated patterns.

Ports:

- **clk**: Clock signal.
- **rst_n**: Active-low reset signal.
- **pattern_in**: 32-bit input pattern.
- **n_repeats**: 8-bit input specifying the number of repeats.
- **pattern_found**: Output flag indicating if the pattern was detected.
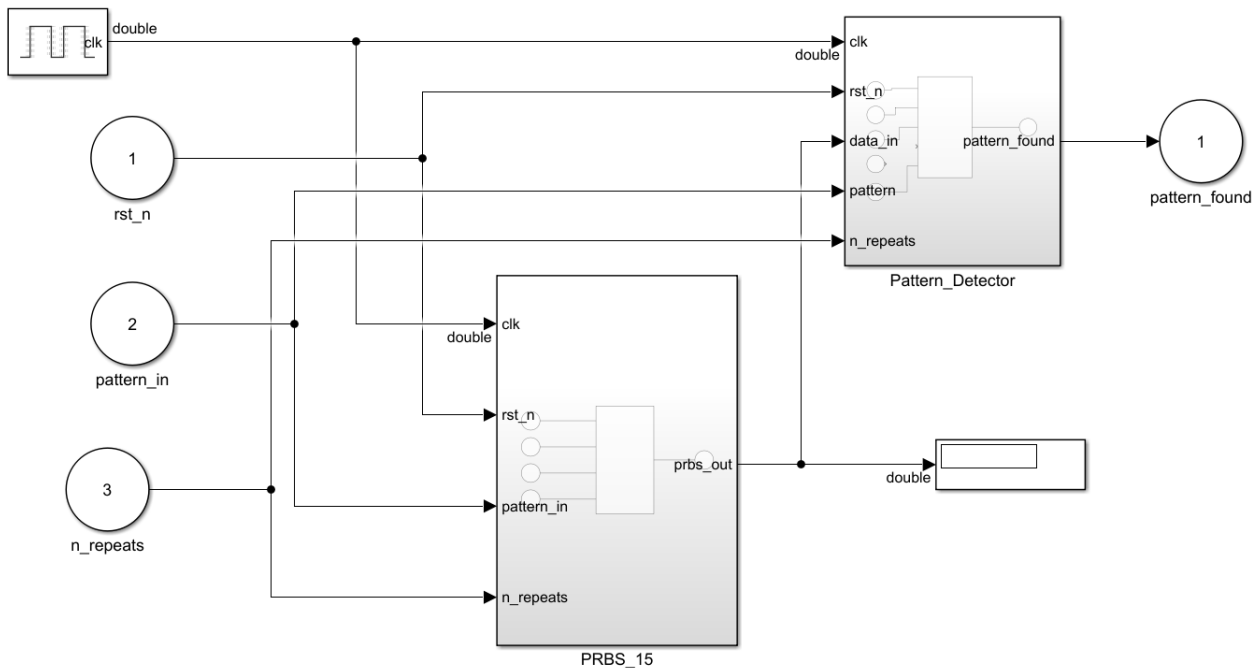
Testbenches

Testbenches are provided for:

1. **PRBS-15 Block**: Validates the PRBS output for different input patterns and repeat counts.
2. **Pattern Detector**: Tests whether the detector correctly identifies patterns.
3. **Serial to Parallel Converter**: Ensures the serial data is correctly converted to parallel.
4. **Top Module**: Verifies the integration of the PRBS-15, Pattern Detector, and Serial-to-Parallel blocks.
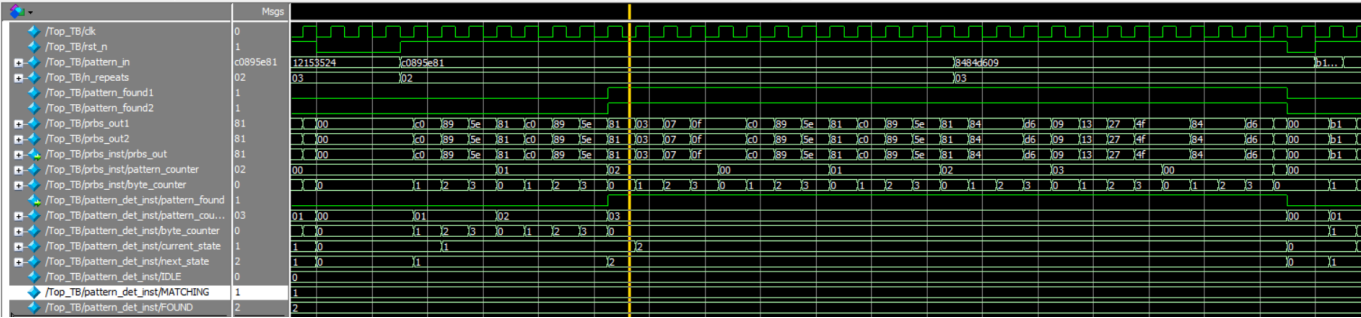
Bitstream Generation

For FPGA deployment, the bitstream generation includes the following considerations:

- A **serial-to-parallel converter** is introduced to handle 8-bit serial inputs in environments with limited resources.
- The overall system is synthesized and implemented on the target FPGA using Xilinx Vivado.