Technical Documentation for Future Engineers

VHDL Project: SPI-Based RF Front-End Configuration

Overview

This VHDL project provides an interface for configuring RF front-end devices via SPI. The design includes a top-level module that orchestrates SPI communication, a local oscillator (LO) controller, an I/O expander controller, and an SPI master module.

Project Structure

1. Top-Level Module (Top_Level.vhd)

- **Purpose**: Coordinates data flow between the FPGA and RF front-end components.
- **Interfaces**:
    - **Clock & Reset**: clk_in, reset_in
    - **LO Configuration Interface**: lo_start, lo_busy, lo_done, lo_write_en, lo_write_addr, lo_write_data
    - **Expander Configuration Interface**: exp_start, exp_busy, exp_done, exp_write_en, exp_write_data, exp_read_data
    - **SPI Lines**: sck, mosi, miso_exp, cs_exp_n, cs_lo_n
    - **Additional LO Signals**: muxout_lo, lock_detect
    - **Reset Expander**: reset_exp_n
- **Functionality**:
    - Routes SPI transactions to either the LO or expander controller based on priority.
    - Converts parallel configuration data into serial SPI commands.

2. SPI Master (SPI_Master_8bit.vhd)

- **Purpose**: Implements an 8-bit SPI master for serial data transmission.
- **Interfaces**:
    - Clock, reset, control, data (input/output), and SPI signals.
- **State Machine**:

    - IDLE → SHIFT (data transmission) → COMPLETE
- **Features**:
    - Operates in SPI Mode 0 (CPOL=0, CPHA=0).
    - Shifts data in and out synchronously with sck.

3. Local Oscillator Controller (LO_Controller.vhd)

- **Purpose**: Manages the programming of LO registers through SPI.
- **Registers**:
    - Stores six 32-bit configuration values.
- **State Machine**:
    - Controls sequential transmission of register values.
    - Asserts cs_lo_n during transaction.
    - Updates lock_detect status.

4. I/O Expander Controller (Expander_Controller.vhd)

- **Purpose**: Communicates with an SPI-based I/O expander.
- **Operations**:
    - Read and write sequences with two SPI transactions per read operation.
- **State Machine**:
    - Writes: Control byte → Register address → Data.
    - Reads: Control byte → Register address → (Second cycle) Control byte → Read data.

5. Testbench (Top_Level_tb.vhd)

- **Purpose**: Verifies correct functionality through simulated stimuli.
- **Tests Included**:
    - LO and expander write transactions.
    - Read-back of expander values.
    - Overlapping transactions to test SPI arbitration.

Future Work

- **Integration with GNU Radio**: Modify the top-level module to accept register values via Ethernet or another communication interface.
- **Expand LO Control**: Add error handling for failed register writes.
- **Optimize SPI Performance**: Increase SPI clock rate if necessary.