Report for Backend Module of Mixed-Signal IC

Team-SuperKings

Member 1- Priyansh Singh 210775

Member2- Anand Gaurav 231040014

1. Introduction

This project aims to design a backend module for a mixed-signal IC that manages initialization and dynamic control of analog components such as an operational amplifier (opamp), ring oscillator, temperature sensor, ADC, and core logic. The backend ensures correct startup sequencing, gain configuration, and adaptive bias control based on temperature readings.

2. Design Overview

The backend interfaces with an external FPGA controller and ensures smooth chip initialization. It controls various modules through outputs, manages gain configuration via serial data, and adjusts bias and clock frequency based on ADC outputs.

2.1 Inputs and Outputs

Inputs:

- i_resetbAll: Active low reset for the entire chip.
- i_clk: 500 MHz main clock.
- i_sclk: Serial clock for gain configuration.
- i_sdin: Serial data input.
- i_RO_clk: Ring oscillator output.
- i_ADCout[3:0]: ADC output from the temperature sensor.

Outputs:

- o_ready: Signals completion of startup.
- o_resetb_amp: Active low reset for the opamp.
- o_gain[2:0]: Controls the opamp gain.
- o_Ibias_2x: Controls opamp bias current.
- o_enableRO: Enables the ring oscillator.
- o_resetb_core: Active low reset for the core logic.
- o_core_clk: Clock signal to core logic.

3. Functional Breakdown

3.1 State Machine Design

The backend follows a Finite State Machine (FSM) with 8 states:

1. RESET: All outputs and registers reset.
2. WAIT_SERIAL: Wait for serial data input.
3. ENABLE_RO: Enable the ring oscillator.
4. WAIT_RO: Waits for 5 clock cycles after enabling the oscillator.
5. CHECK_ADC: Checks the ADC output and configures bias settings
6. ENABLE_MODULES: Enable opamp and core logic.
7. WAIT_MODULES: Waits for 5 clock cycles after enabling modules
8. READY: Set o_ready and monitor ADC continuously.

3.2 Serial Data Handling

The module captures 5 bits of serial data (d0-d4). Only d2, d3, and d4 are used to configure the 3-bit opamp gain (o_gain). A shift register receives data on the rising edge of i_sclk.

3.3 ADC Moving Average Filter

A 4-tap moving average filter smooths ADC output:

ADCavg=(ADCout[0]+ADCout[1]+ADCout[2]+ADCout[3])/4

Bias and Clock Control:

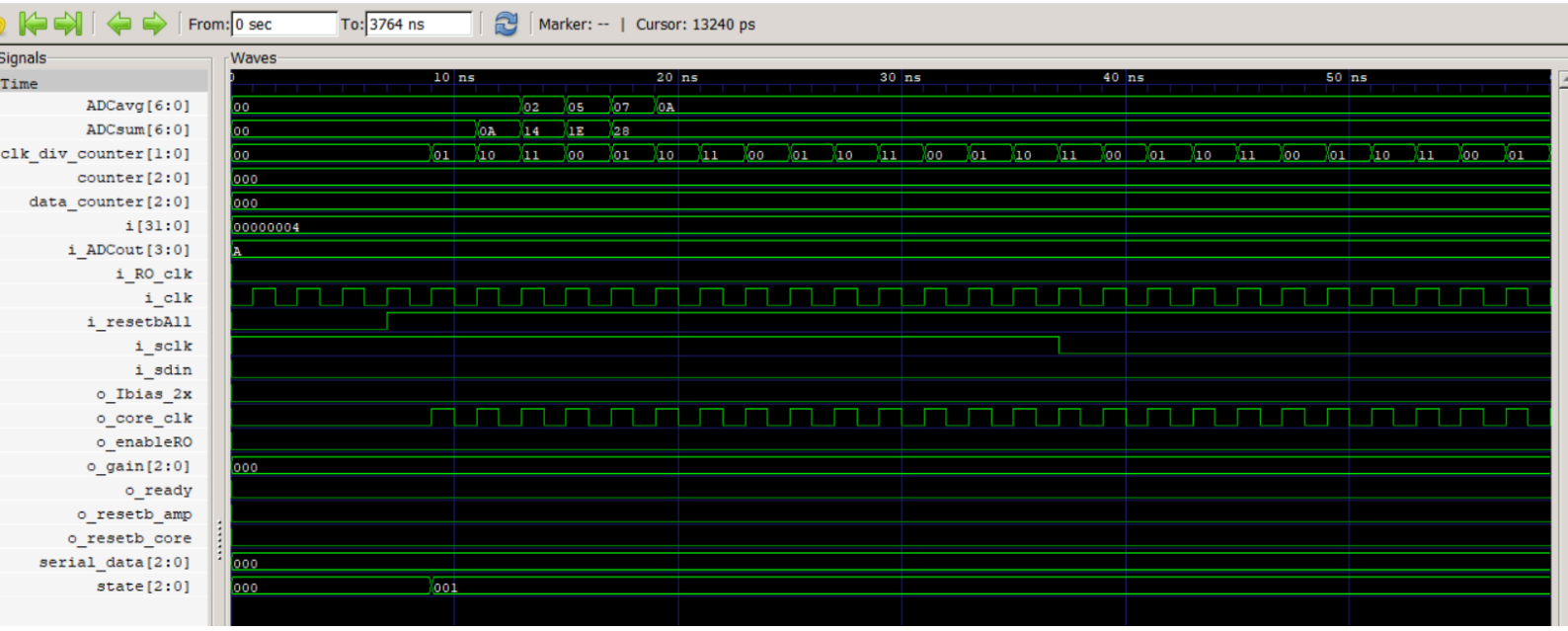
- ADCavg ≤ 12: o_Ibias_2x = 0 (normal bias), o_core_clk = i_clk
- ADCavg > 12: o_Ibias_2x = 1 (double bias), o_core_clk = i_clk/4
- 8 ≤ ADCavg ≤ 12: Hold current values

The design and functionalities of various parts of the code and debugging to get the final code involved efforts by both the team members equally. With Priyansh Singh focussing more on the state machine design(3.1),clock control and report making, and Anand Gaurav focussing more on serial data handling(3.2) and ADC filter(3.3) .

4. Startup Sequence Implementation with waveform:

The startup sequence follows these steps:

When i_resetbAll = 0, all outputs of the backend should be pulled to 0. All internal registers should be either set or reset. When i_resetbAll becomes 1, the start up sequence is initiated.

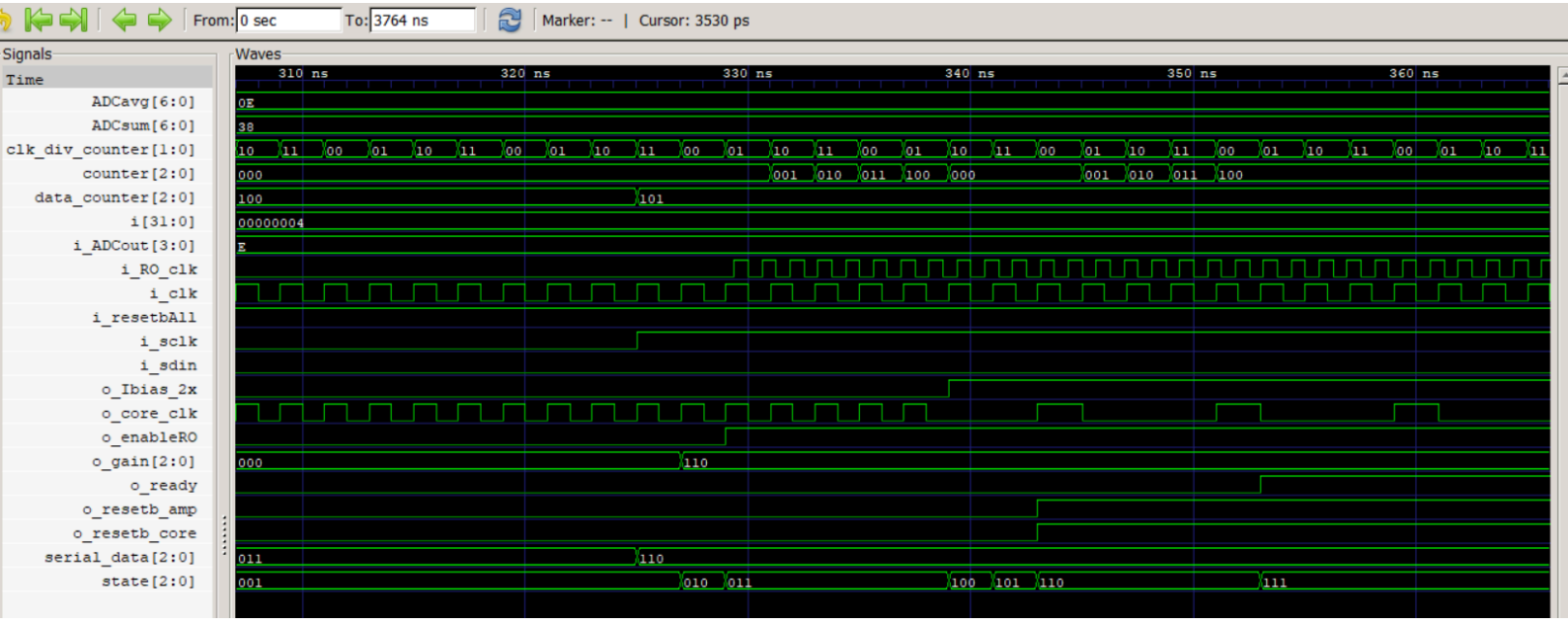


Wait for serial data, configure o_gain.

Enable ring oscillator (o_enableRO = 1).

Wait 5 clock cycles.

Compute ADC average, set o_Ibias_2x (going to 1 this time) and o_core_clk accordingly.



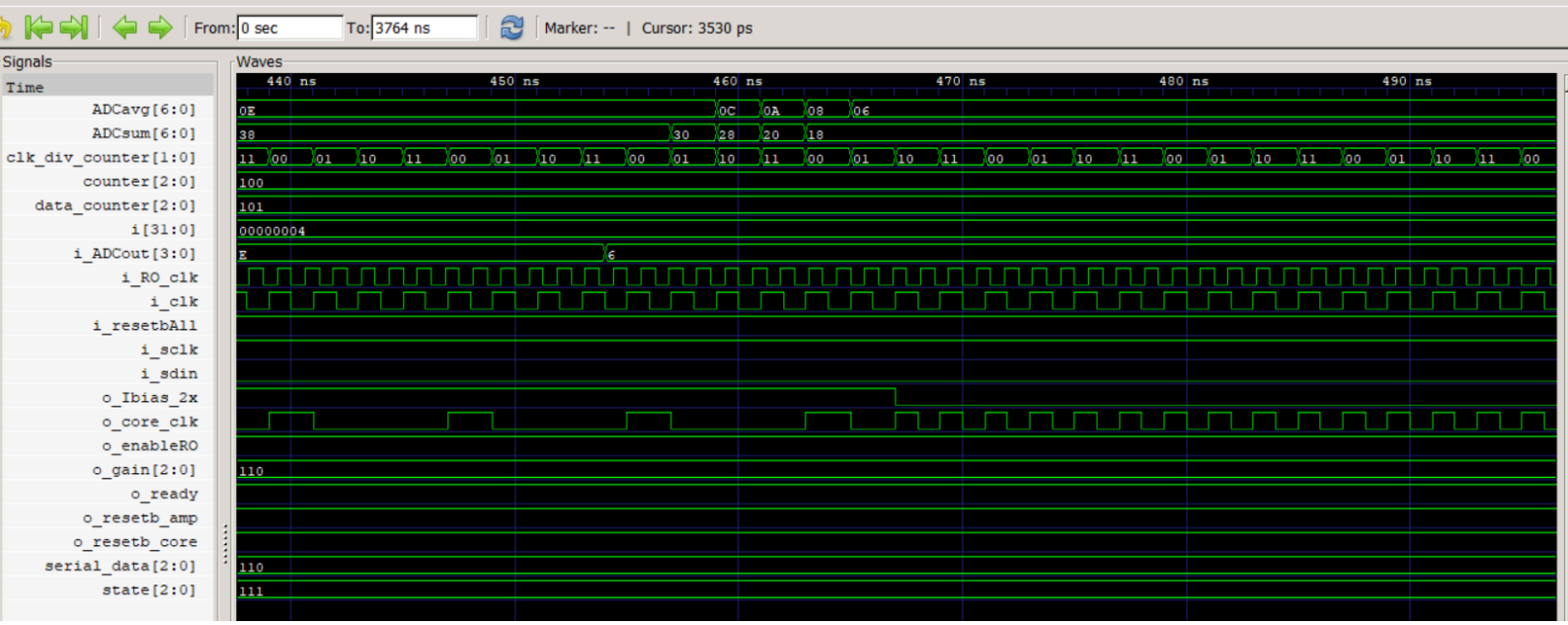
(this gain 110 is obtained by using the sample FPGA file uploaded in the resources)

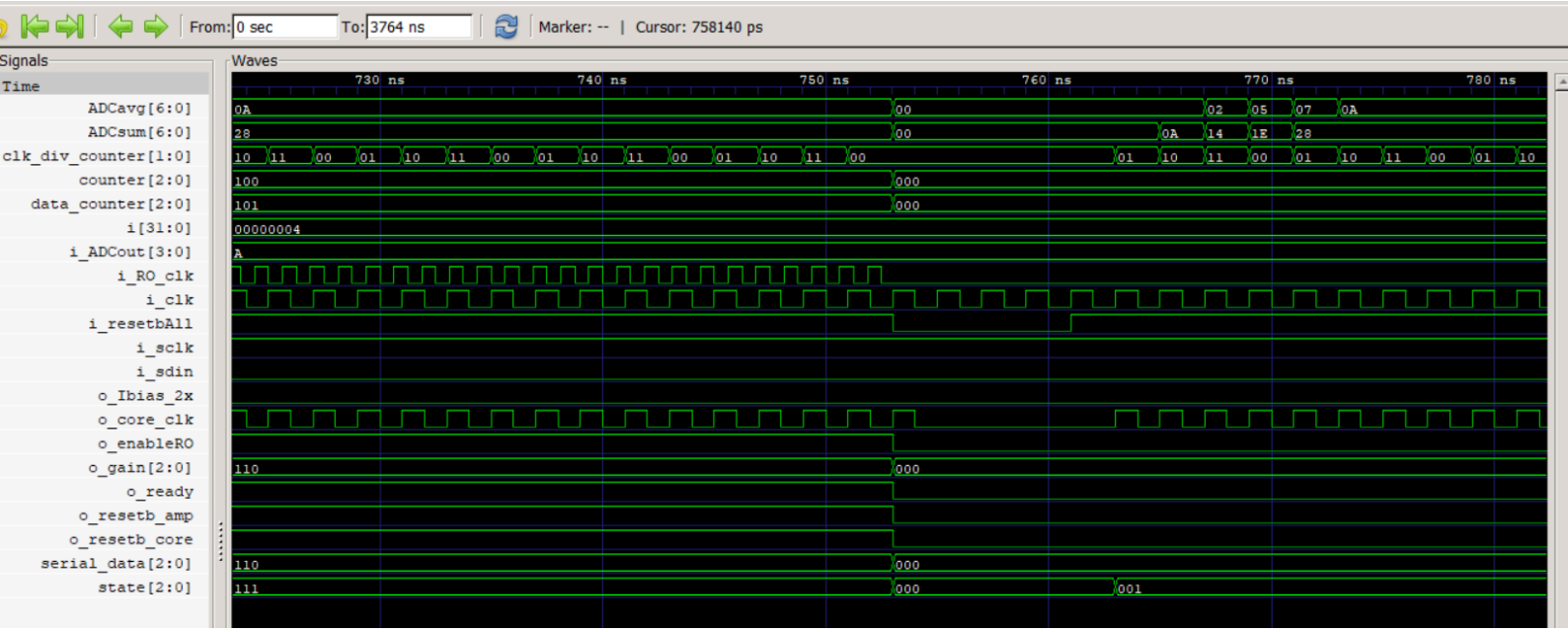
Release o_resetb_amp and o_resetb_core.

Wait another 5 clock cycles.

Set o_ready = 1.

Monitor ADC continuously for temperature changes.





Reset all outputs (o_resetb_amp, o_resetb_core, o_ready = 0) with i_resetbAll going to 0.

5. Conclusion

The backend module successfully handles the startup sequence, gain configuration, temperature-based bias control, and dynamic clock adjustment. The FSM ensures proper synchronization between serial data reception, ADC filtering, and module enablement. Continuous monitoring keeps the system adaptive to temperature changes.