EE214
SPI Communication to DAC via Multimeter

Tanisha Hase (23B3984)
Varun Mishra (23B3985)

Project Overview

This project focuses on retrieving a digital value from a register which was used to store the digital value received from the ADC in the previous task and transmitting it to a DAC (Digital-to-Analog Converter) using SPI. The DAC, with a reference voltage (Vref) of 3.3V, converts the digital data into an analog signal. This signal is then measured with a multimeter to verify that it corresponds to the original input voltage.

System Overview

Components

ADC: Sends the digital signal to the FPGA.

Microcontroller/FPGA: The central unit responsible for generating the SPI signals and controlling the communication.

DAC: Converts the digital signal from the microcontroller into an analog signal.

Multimeter: Measures the output voltage from the DAC to verify accuracy.

SPI Protocol and Signal Flow

- **Master-Slave Configuration**: In this setup, the FPGA operates as the master, controlling the **Serial Clock (SCLK)**, **Chip Select (CS)**, and **Master Out Slave In (MOSI)** lines for communication with two connected slave devices: an ADC and a DAC. Both chip select lines (**CS1** and **CS2**) are initially set to ‘1’ (inactive), preventing any unintentional communication.

1. **ADC Communication (CS1)**: To initiate communication with the ADC, **CS1** is set to ‘0’, activating the ADC and allowing the master (FPGA) to send configuration and data requests over the **MOSI** line. During this phase, the clock signal (**SCLK**) is controlled by the FPGA to synchronize data transfer with the ADC.
2. **DAC Communication (CS2)**: Once communication with the ADC is complete, **CS1** is set back to ‘1’ to deactivate the ADC, and **CS2** is pulled to ‘0’ to activate the DAC. This enables data transfer from the FPGA to the DAC, where the digital data is converted into an analog signal based on the reference voltage.

This sequential activation of **CS1** and **CS2** ensures only one slave is active at a time, avoiding data conflicts and maintaining accurate communication. The **SCLK** and **MOSI** signals are shared across both devices, with the FPGA controlling which slave responds based on the CS signals.

- **Clock Division**: The master clock is divided to match the SPI requirements.
- **Data Transmission Process**: Data is sequentially sent to the DAC, accompanied by control signals to ensure proper timing and data integrity.
- **DAC Configuration**: The DAC uses a specific configuration (0111) as defined in the setup.

Implementation

The code is written in VHDL, implementing SPI communication with the DAC and an ADC. The key processes of the code are explained below:

Clock Divider Process:

- Divides the input clock signal to a lower frequency suitable for SPI communication.
- The `clk_counter` resets every 1000 cycles, toggling `sclk_gen` to create the SPI clock (SCLK).

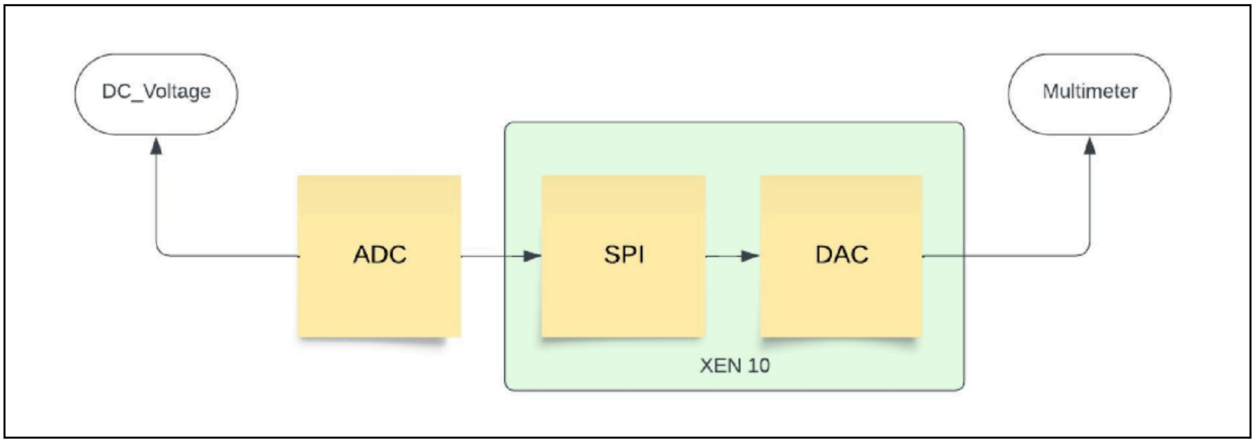
Data Transmission Process:

- Manages the transmission of data through the MOSI line in sync with `sclk_gen`.

- Uses bit_counter to track the transmission state, setting cs_gen1 for the ADC and cs_gen2 for the DAC as required.

Data Reception and Conversion:

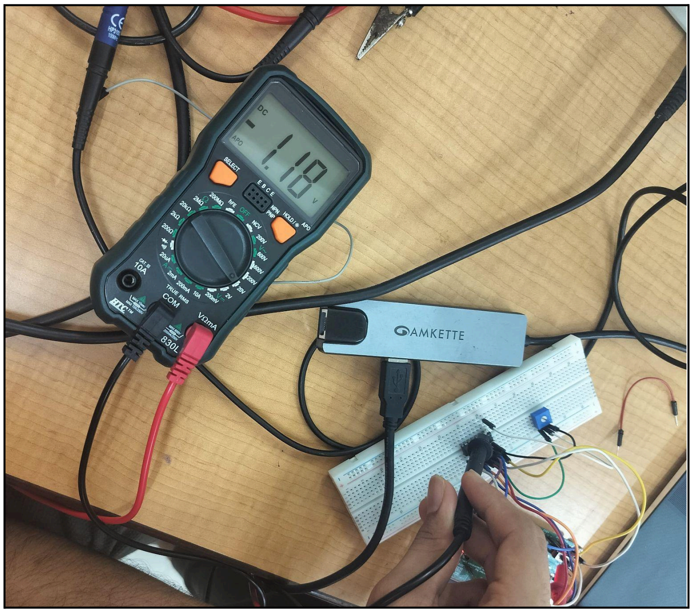
- Reads data from the ADC into data_in and sends it to the DAC through SDI to recreate the analog signal.
- Configures SDI for DAC communication, transmitting data bits while toggling CS and tracking the bit position.



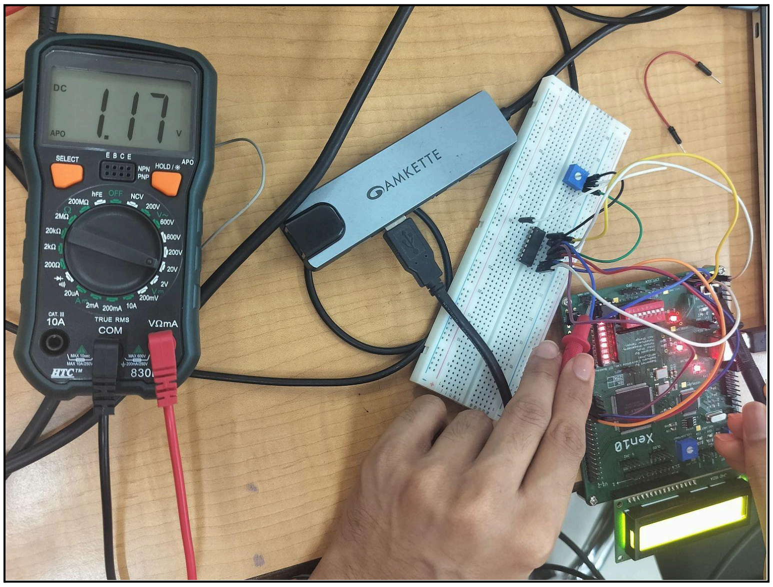
Observations and Results

Output Consistency: The multimeter displayed voltage values consistent with expected DAC output, confirming proper data transmission and conversion.

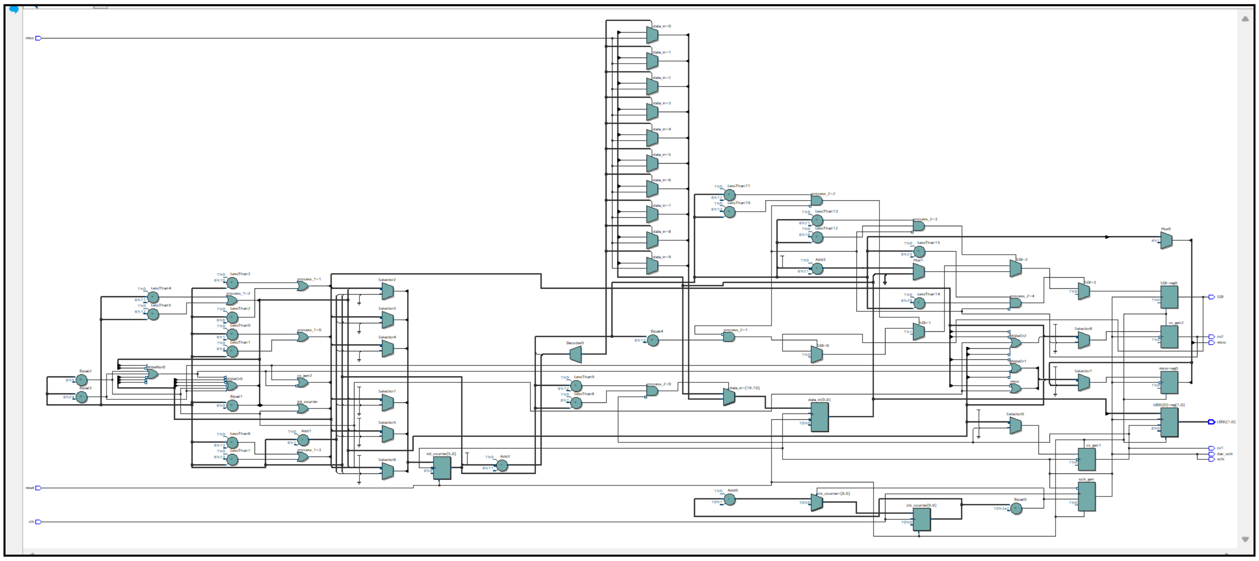
Error Analysis: Any minor discrepancies could stem from DAC resolution limits or voltage drift in Vref.



ADC Input



DAC Output



Work Distribution

Varun: Developed the VHDL code for SPI communication between the FPGA and DAC.

Tanisha: Set up and connected hardware. Also prepared the final report, detailing the code functionality and testing results.

Testing: Conducted together to verify the analog output on the multimeter, troubleshoot issues, and confirm the setup's accuracy.

Conclusion

This project successfully demonstrated SPI communication between an FPGA and a DAC, enabling accurate digital-to-analog conversion. The multimeter verification confirmed that the analog output was consistent with the intended values, validating the effectiveness of the SPI-DAC integration.