MAPD-modA-FinalProject-Report

Alireza Molla Ali Hosseini

May 2023

1 Report

VHDL implementation of a 4-tap FIR filter

1.1 Design Overview

Figure 1 provides an overall view of the design.

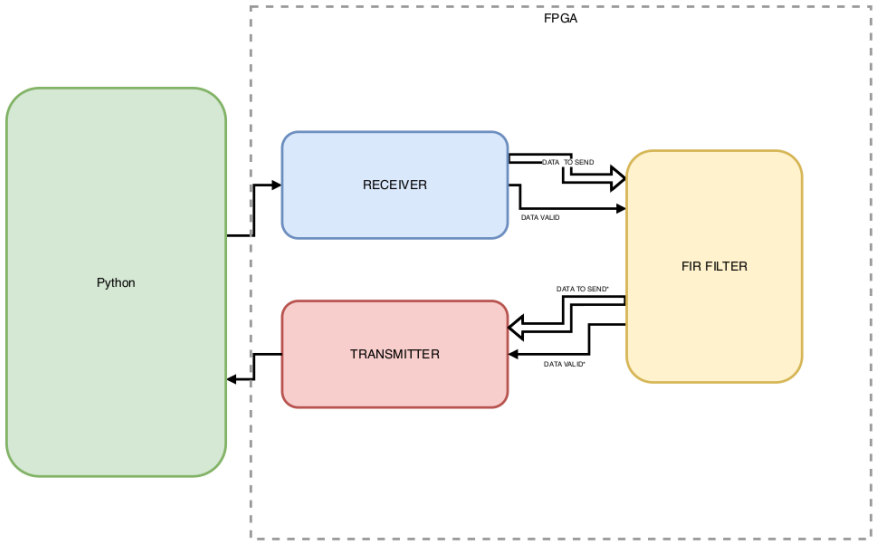

Figure 1: Design overview

1.2 Receiver

1.2.1 Design Overview

The UART receiver is responsible for receiving and decoding serial data transmitted using the UART protocol. It converts the serial data into parallel format

and provides the received data to the system. The use of a state machine ensures proper synchronization and data reception. The baud rate generator generates the clock signal required for the serial reception.Figure 2 shows the design of the receiver.

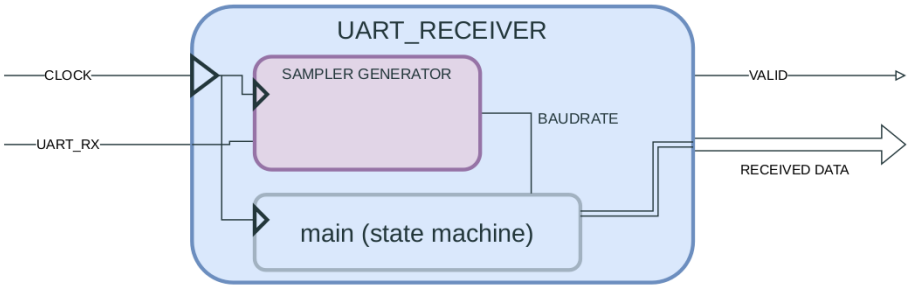

Figure 2: Receiver Design

1.2.2 Design Architecture

The UART receiver is implemented using a structural architecture style in VHDL. The architecture describes the functionality and structure of the receiver, including the state machine and the baud rate generator.

1.2.3 Entity

The entity "UART-Receiver" defines the inputs and outputs of the receiver. It has the following ports:

- **clk** The clock input for synchronous operation.
- **uart-rx** The serial data input for reception.
- **data-valid** The data-valid signal to validate the input data.
- **data-to-send** The output parallel data received.

1.2.4 Signals

The architecture includes the following signals:

- **state** A signal of type state representing the current state of the receiver state machine.
- **data-to-send** A signal of type std-logic-vector of width 8 representing the data
- **baudrate-out** A signal of type std-logic representing the output of baudrate generator.
- **data-valid** A signal of std-logic representing the validation of data.

1.2.5 Components

The receiver architecture includes a component declaration for the Baud Rate Generator. The baud rate generator is responsible for generating the clock signal with the desired baud rate. The clock frequency is divided by the BAUD-DIVISOR to obtain the baud clock.

1.2.6 Instantiation

The sampler-generator (baud rate) component is instantiated to create baudrate-output.

1.2.7 Receiver

The receiver state machine operates as follows:

On the rising edge of the clock (clk) the state machine transitions through different states to receive and decode the serial data. When the state machine is in start state, the state machine starts the reception process. The state machine goes through a series of states to receive each bit of the data in a serial format and reconstruct it into parallel format (uart-rx) in data-to-send. After receiving all the bits, the state machine outputs the parallel data data-to-send and returns to the idle state, waiting for the next reception.

1.3 FIR Filter

1.3.1 Design Overview

The FIR filter is a digital filter that performs a linear convolution between the input samples and a set of filter coefficients. In this implementation, we are designing a 4-tap FIR filter using VHDL. The filter has a data width of 8 bits for both the input and output samples. The filter coefficients are unsigned values. The filter includes a D flip-flop for delaying the input samples.

1.3.2 Design Architecture

The FIR filter is implemented using the behavioral architecture style in VHDL. The architecture is named "Behavioral" and describes the functionality and behavior of the filter. It includes the necessary signals and components to perform the filtering operation.

1.3.3 Entity

The entity "my-fir" defines the inputs and outputs of the filter. It has the following ports:

- **clk** The clock input for synchronous operation.
- **valid-in** The valid-in input for validation of the input data.

- **Xin** The input sample to be filtered.
- **valid-o** The output to validate the output data
- **Yout** The filtered output sample.

1.3.4 Signals

The architecture includes the following signals:

- **Coefficients** Signals of type signed that holds the filter coefficients. They have a width of 8 bits.
- **Multiplication** Signals of type signed that holds the Multiplication outcomes. They have a width of 16 bits.
- **Adder** Signals of type signed that holds the Adders outcomes. They have a width of 16 bits.
- **D flip-flop** Signals of type signed that holds the D flip-flop outputs. They have a width of 16 bits.

1.3.5 Components

The filter architecture includes a component declaration for the D flip-flop. The D flip-flop is a sequential element used for delaying the input samples by one clock cycle. The component declaration specifies the generic parameter DATA-WIDTH, which is set to 8 for this design.

1.3.6 Instantiation

The D flip-flop component is instantiated four times to create four D flip-flops (dff0, dff1, dff2, and dff3). Each D flip-flop is used to delay the input samples in a pipeline fashion.

1.3.7 Filter Operation

The filter operates as follows:

On the rising edge of the clock (clk), if the valid-in is 1, the output sample (Yout) is obtained by truncating the accumulated value to the appropriate width (8 bits). The Procedure of the Filter is shown in Figure 3.

1.4 Transmitter

1.4.1 Design Overview

The UART transmitter state machine is responsible for transmitting data serially using the UART protocol. It converts parallel data into a serial format and transmits it over a single data line.Figure 4 shows the design of the transmitter.

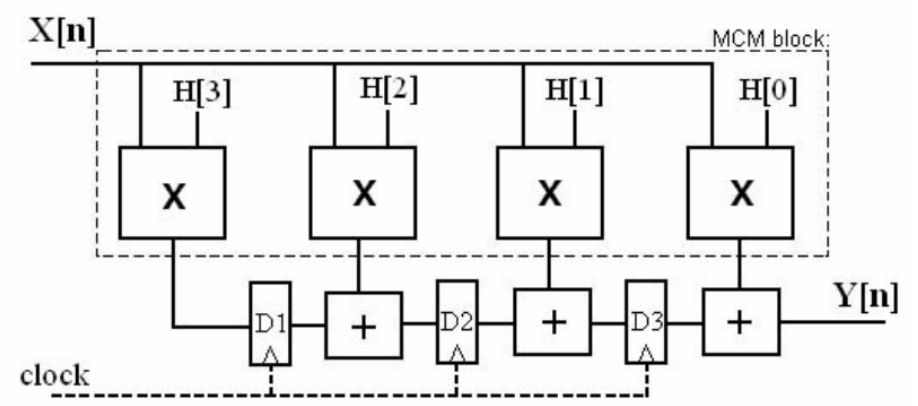

Figure 3: FIR Filter Operation

1.4.2 Design Architecture

The UART transmitter is implemented using a structural architecture style in VHDL. The architecture describes the functionality and structure of the transmitter, including the state machine and the baud rate generator.

1.4.3 Entity

The entity "UART-Transmitter" defines the inputs and outputs of the transmitter. It has the following ports:

- **clk** The clock input for synchronous operation.
- **data-valid** The validation input for validating the input sample.
- **data-to-send** The input parallel data to be transmitted.
- **busy** The busy signal avoiding change of the input data.
- **uart-tx** The serial data output for transmission.

1.4.4 Signals

The architecture includes the following signals:

- **state** A signal of type state representing the current state of the receiver state machine.
- **data-to-send** A signal of type std-logic-vector of width 8 representing the data
- **baudrate-out** A signal of type std-logic representing the output of baudrate generator.

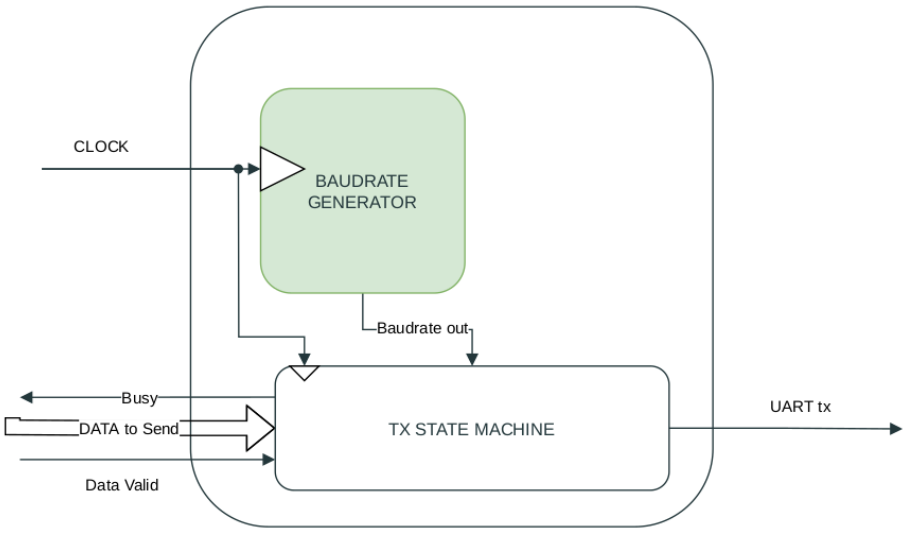

Figure 4: Transmitter Design

1.4.5 Components

The transmitter architecture includes a component declaration for the Baud Rate Generator. The baud rate generator is responsible for generating the clock signal with the desired baud rate. The clock frequency is divided by the BAUD-DIVISOR to obtain the baud clock.

1.4.6 Transmitter State Machine

The transmitter state machine operates as follows:

On the rising edge of the clock (clk) the state machine transitions through different states to transmit the data serially. When the state machine is in start state, the state machine starts the transmission process by loading the parallel data (data-to-send) into the uart-tx and initializing. The state machine then goes through a series of states to shift out the data bit by bit in a serial format. After transmitting all the bits, the state machine returns to the idle state and waits for the next transmission.

1.5 Conclusion

This VHDL implementation of a 4-tap FIR filter with a data width of 8, unsigned coefficients, and a D flip-flop provides a means to perform digital filtering on input samples. The design is based on the principles discussed in the paper "VHDL Generation of Optimized FIR Filters" and follows a behavioral architecture style. It allows for the efficient implementation of FIR filters in VHDL

and can be further customized to suit specific filter requirements.