IL2234

Digital Systems Design and Verification using Hardware Description Languages

Project Milestone 2

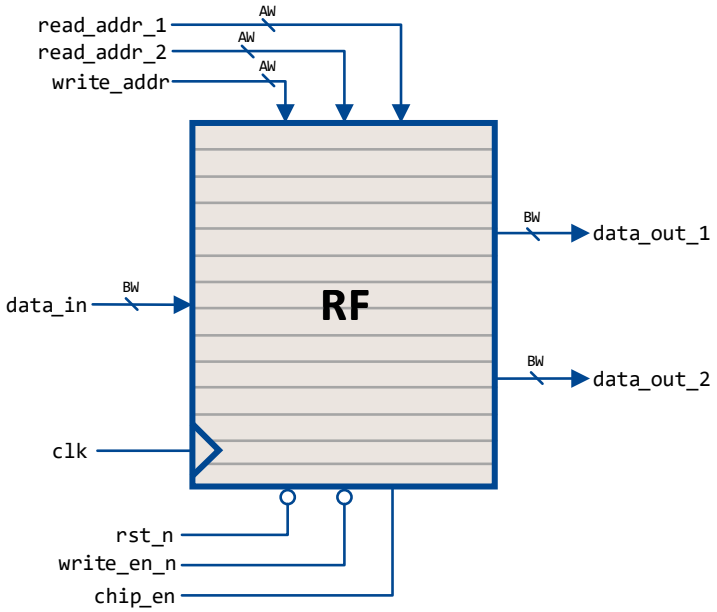
Sequential Circuits

Introduction

In this milestone, we will build the second component of our microprocessor: the Register File (RF). We will design and model the RF circuit based on the SystemVerilog modelling concepts learned in the class. Additionally, we will develop a simple testbench to test the basic functionality of the RF.

Overview

Below is a simple diagram of the RF



The inputs and outputs are described in the table below:

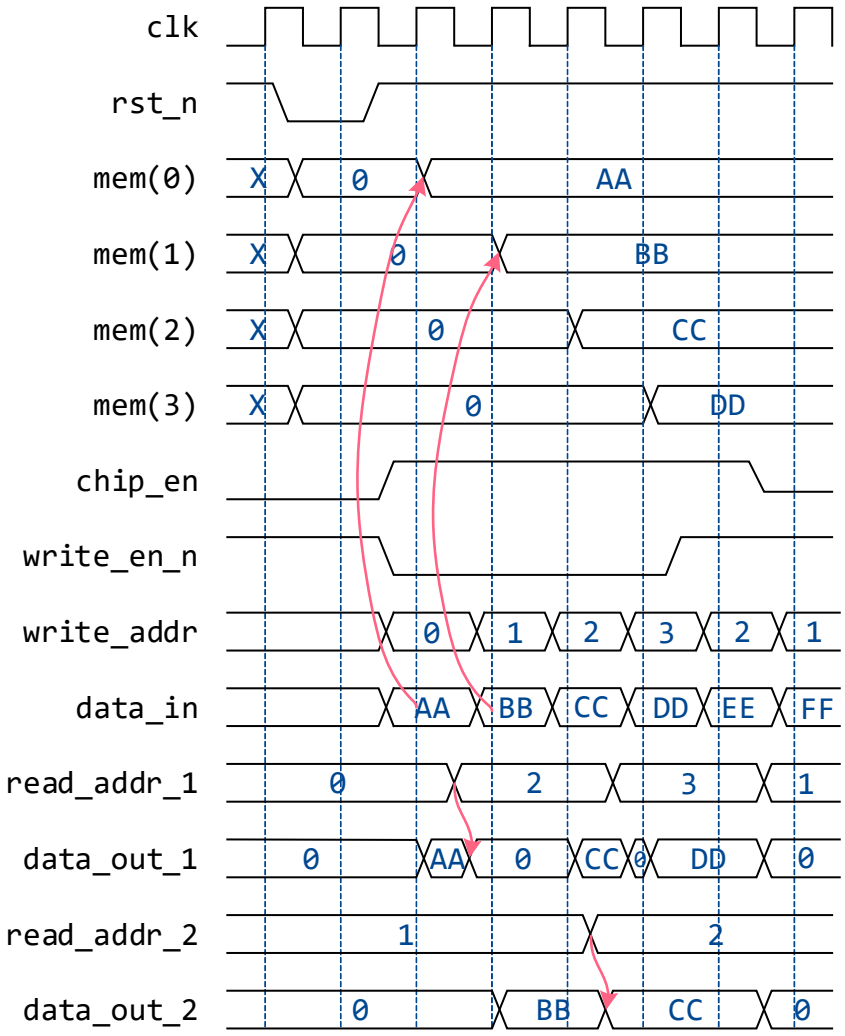
Name	Direction	Type	Bitwidth	Description
clk	Input	logic	1	Clock input
rst_n	Input	logic	1	Asynchronous active low reset signal
data_in	Input	logic signed	BW [1]	Data input port
data_out_1	Output	logic signed	BW [1]	Data output port 1
data_out_2	Output	logic signed	BW [1]	Data output port 2
read_addr_1	Input	logic unsigned	log2(DEPTH)[2]	Address for output port 1
read_addr_2	Input	logic unsigned	log2(DEPTH)[2]	Address for output port 2
write_addr	Input	logic unsigned	log2(DEPTH)[2]	Address for input port
write_en_n	Input	logic	1	Active low read enable
chip_en	Input	logic	1	Active high chip enable

The RF behaviour is defined according to the following table:

Function	clk	rst_n	chip_en	write_en_n	Memory	data_out_1	data_out_2
Reset	X	0	X	X	mem(all) = 0	0	0
Standby	X	1	0	X	No change	0	0
Read	X	1	1	X	No change	mem(read_addr_1)	mem(read_addr_2)
Write		1	1	0	mem(write_addr) = data_in	mem(read_addr_1)	mem(read_addr_2)

A timing diagram showing a typical operation of the register file is shown below:

[1] BW is a configurable parameter that defines the bitwidth of the RF data
[2] DEPTH is a configurable parameter that defines the number of rows of the RF



Tasks

1. Design the RF according to the specifications presented earlier. Write your code in System Verilog using **only synthesizable** constructs. Follow the same naming for the signals indicated in the diagrams and tables.

2. Write a testbench that generates read and writes on the RF. Use randomised data for the data and addresses. Use the printing functions to display the contents of the RF in the terminal and check against the expected results.

3. Implement the ALU design from milestone 1, targeting the FPGA development board provided to you. Modify the 7-segment module to support negative numbers and use the switches and displays to connect the input operands and show the result. Use the LEDs to display the status of the ONZ flags.

Deliverables

Use GitHub Classroom to submit your RTL and testbench files.

The GitHub Classroom repositories contain a skeleton of the RTL and testbench files for the RF that you may use for your development.

Bonus tasks

(1 points) Implement a module that uses PWM to modify the brightness of the LEDs and 7-segment displays. Hint: You can use the clock divider as a starting point.

(2 point) Implement a multiplexing system that allows you to rotate the displayed data on the 7-segments to be able to show 8-digit numbers. Test it with a 32 bit ALU