

# EE20021 Coursework – Activity 2

## Microprocessor Register file (RF)

Uriel Martinez-Hernandez

### Introduction

You will implement a Register File (RF) to be used in a microprocessor. The name of your module will be **RegisterFile** and should have the port names described below. To get started, download the zip file 'Files for CW2-RegisterFile' from Moodle, where you will find a testbench file that you will need to complete to test the functionality of your register file module.

A register file often has more than one "port" that is a connection to the registers. Each port can access one register and may be read only or read/write. The core of the register file is a two-dimensional array which can be written as:

logic [**RegisterWidth-1:0**] Registers [**Number of Registers**];

Your register file will have 64 registers that are 16 bits width. This module should have the following connections and capabilities:

**Clock** – All the write operations in your register file will be synchronous with the positive edge of Clock signal.

**AddressA** – A 6-bit input signal that indicates the register address to be accessed by port A to read and write data.

**ReadDataA** – A 16-bit output signal that asynchronously accesses and reads the value of the register address indicated by *AddressA*.

**WriteData** – A 16-bit signal with data to be written to the register address indicated by *AddressA*. The writing process takes place on every clock cycle when the *WriteEnable* signal is active.

**WriteEnable** – Active high input indicates that a writing process in the register should occur.

**AddressB** – A 6-bit input signal that indicates the register address to be accessed by port B to read only.

**ReadDataB** – A 16-bit output signal that asynchronously access and reads the value of the register address indicated by *AddressB*.

## Activity 1 – Coding

Create a project named RegisterFile in Quartus Prime and create a SystemVerilog file named RegisterFile for the design your register file module. Create the register file module with the specifications described above. Make sure that the module and port names correspond to the names in the *RegisterFileTb.sv* testbench provided on Moodle.

## Activity 2 - Testbench

Complete the testbench *RegisterFileTb.sv* that has been provided on Moodle to test all the functionalities of the register file.

## IMPORTANT

You have the option to work in a group of **2 students maximum**. Add your name ( and the name of your teammate if you worked in a group) at the top of all the SystemVerilog files developed in this coursework activity.

Add clear comments and detailed description of all the components of your module, and do not forget to use meaningful names for the variables, ports, etcetera. Also remember that it is important to indent your code to make it easier to read by others. All these aspects will be considered for the mark of this module.