EE120A Logic Design
Department of Electrical Engineering
University of California – Riverside

Instructor: Roman Chomko

LABORATORY # 7
L A B M A N U A L

Register File Design

Objectives

1. Design of register files, synthesis and implementation;
2. Emulation of register file performance

Equipment

- PC or compatible
- Digilent's Basys Spartan-3E FPGA Evaluation Board

Software

- Xilinx ISE Design Software Suite 10.1
- ModelSim XE III modeling software
- Digilent's Adept ExPort Software

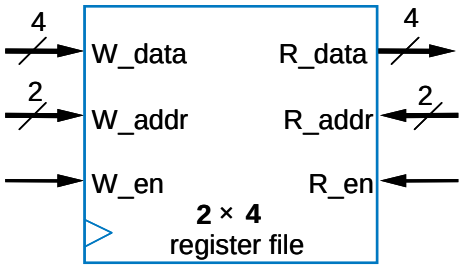
Parts

- N/A

Background – Register Files

A typical datapath has multiple registers. The construction of a bus system with a large number of registers requires different techniques. A set of registers having common operations performed on them maybe organized into a register file*. A block diagram of a 2x4 register file is shown in **Figure L7-1**. Note that in general a register file is denoted as $\mathbf{2^m \times n}$, where m is the number of register address bits and n is the number of bits per register. In our case it is better to use a notation 2^2 x 4 register file†.

Figure L7-1. Block Diagram of a 2x4 Register File



Register files have a memory-like behavior. However, they are very different from memories like SRAM, ROM, etc. as will be explained in more detail in class. They are generally used as fast temporary storage. Brief comparative properties are outlined in **Table L7-1** below.

Register File	Memory (SRAM)
m n-bit storage words	m n-bit storage words
Few words (<256), many ports, dedicated read/write ports, so that read and write operations can be done simultaneously.	Many words (>1024), few ports, shared read/write ports. Read and write operations cannot be done simultaneously.
Writing to a register takes just one clock cycle. Read op doesn't require a clock or additional control inputs.	Writing to a register can take many clock cycles. Read op requires a clock and other CTRL.
Logically Static Content	Logically Dynamic Content
Synchronous	Different implementations but mostly asynchronous‡

Table L7-1. Register Files vs. SRAM Memory

An example of a **2^2 x 5** (or **4 x 32**) register file architecture was given in **Lecture 17** and is provided here for convenience in **Figure L7-2**.

* See **Lecture 17** on the mirror display application
† Both notations can be used but the latter is preferred
‡ For an application developer it does sometimes contain circuitry that makes it look like synchronous.

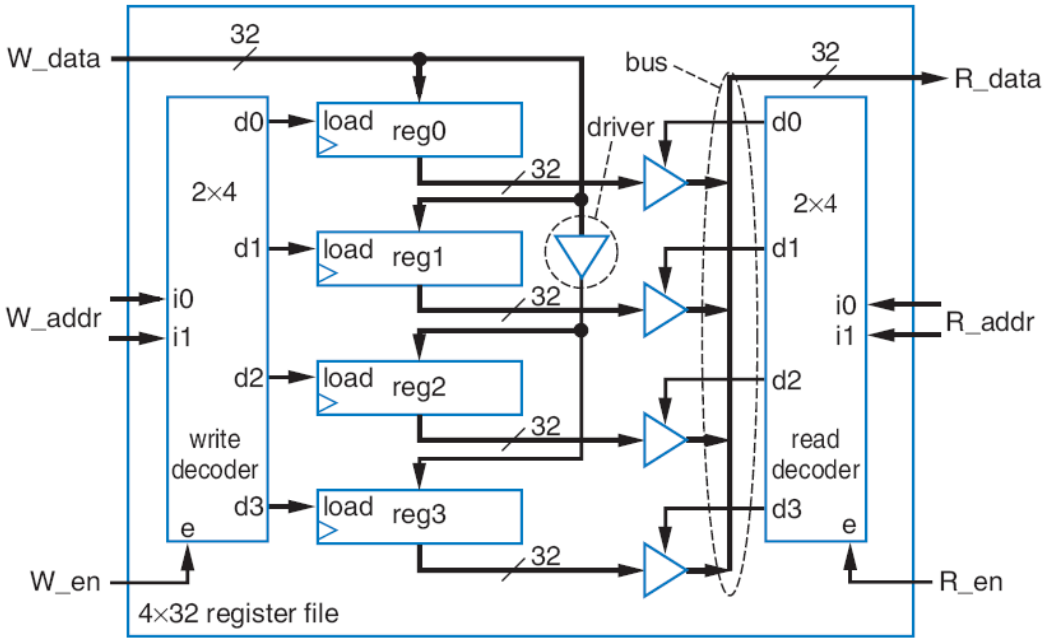


Figure L7-2. Implementation of a **2^2 x 5** Register File

In summary, a register file is faster and more flexible. However, due to the circuit size of a FF (flip-flop), a register file is feasible only for small storage. Note that a register file is a typical design component of any CPU.

Specification

In this FPGA application development assignment, we will implement a **2^2 x 4** register file and emulate its performance on the Digilent's Basys Board.

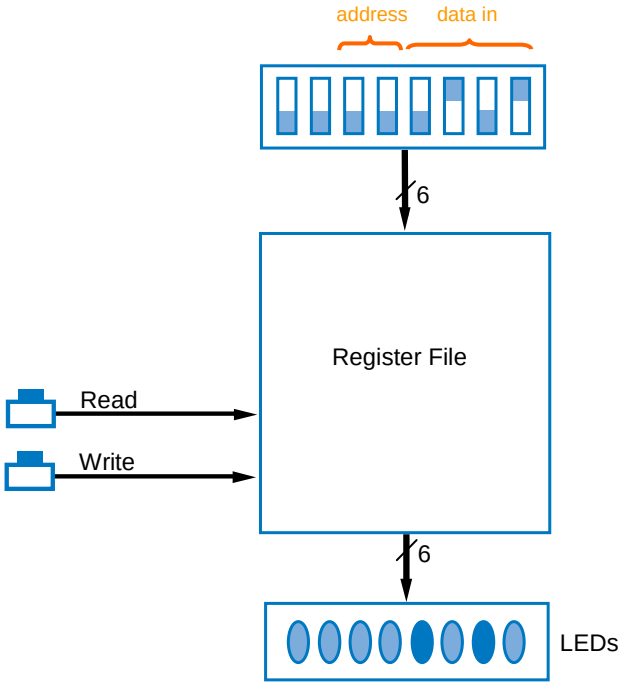


Figure L7-3. Register File and Basys Board implementation hint

The application functions as follows:

- 2 address switches provide the address (4 registers);
- 4-bit bytes ("data in") will be written into the register file when "Write" button is pressed. LED's should not react to this operation;
- 4-bit bytes will be read out when a "Read" button is pressed: 4 rightmost LED's will indicate the value and the following two LED's will show the address indicated by the address switches.

Demonstration

Demostrate that the application performs according to specs.

Procedures

1. Xilinx ISE Design and Synthesis environment;
2. Creation of Configuration files;
3. Usage of Adept ExPort download software;

Presentation and Report

Must be presented according to the general EE120A lab guidelines posted in iLearn.

Prelab

1. Review Lectures 16, 17;
2. Try to answer all the questions, prepare logic truth tables, do all necessary computations