

Show title & tags ✓

The Islamic University of Gaza

Computer Architecture Lab - 2023

LAB (4) || Register File Design

Eng: Amal I. Mahfouz, Eng: Hassan Sammour

Objectives

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

Register File Design

A register file is a component of a processor that stores a small amount of data called registers, which are small, high-speed storage locations for holding data that the processor is currently using or manipulating. A register file typically consists of multiple registers that can be read from and written to in order to transfer data between different stages of a processor's pipeline.

How does an Register File Design unit work?

1. Register File Organization:

The register file is typically organized as an array of registers, each of which is identified by a unique address or index number. For example, a 32-bit register file might contain 32 registers, each of which is 32 bits wide, and can be addressed using a 5-bit index value.

2. Register Read:

When an instruction requires the value of a register, the register file reads the value from the specified register and provides it as an output to the rest of the processor. The read operation takes place in a single clock cycle.

3. Register Write:

When an instruction needs to write a value to a register, the register file accepts the new value and updates the specified register. The write operation also takes place in a single clock cycle.

4. Simultaneous Access:

The register file design is often optimized to allow multiple read and write operations to be performed simultaneously. For example, some designs may allow two read operations and one write operation to take place in a single clock cycle, or vice versa.

5. Address Generation:

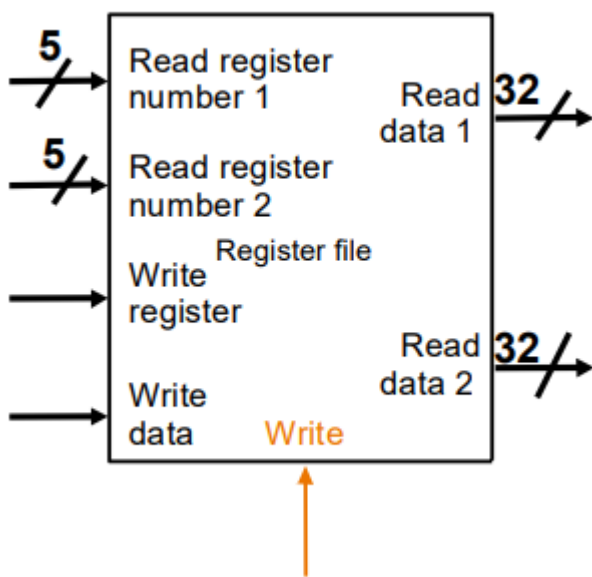
The address of the register to be accessed is generated by the instruction decoder, which typically includes an index field that identifies the register number. This index field is used to select the appropriate register from the register file array.

6. Register File Control:

The register file design unit may include various control signals that govern the read and write operations, such as clock enable, write enable, and read enable signals. These control signals ensure that the register file operates correctly and efficiently within the processor's pipeline.

Overall, a register file design unit provides a fast and efficient mechanism for storing and accessing data within a processor. By organizing registers in an array and providing fast read and write operations, the register file enables the processor to quickly and easily manipulate data as it executes instructions.

A block diagram of MIPS register file includes 32 32-bit general purpose registers



Input/output Descriptions:

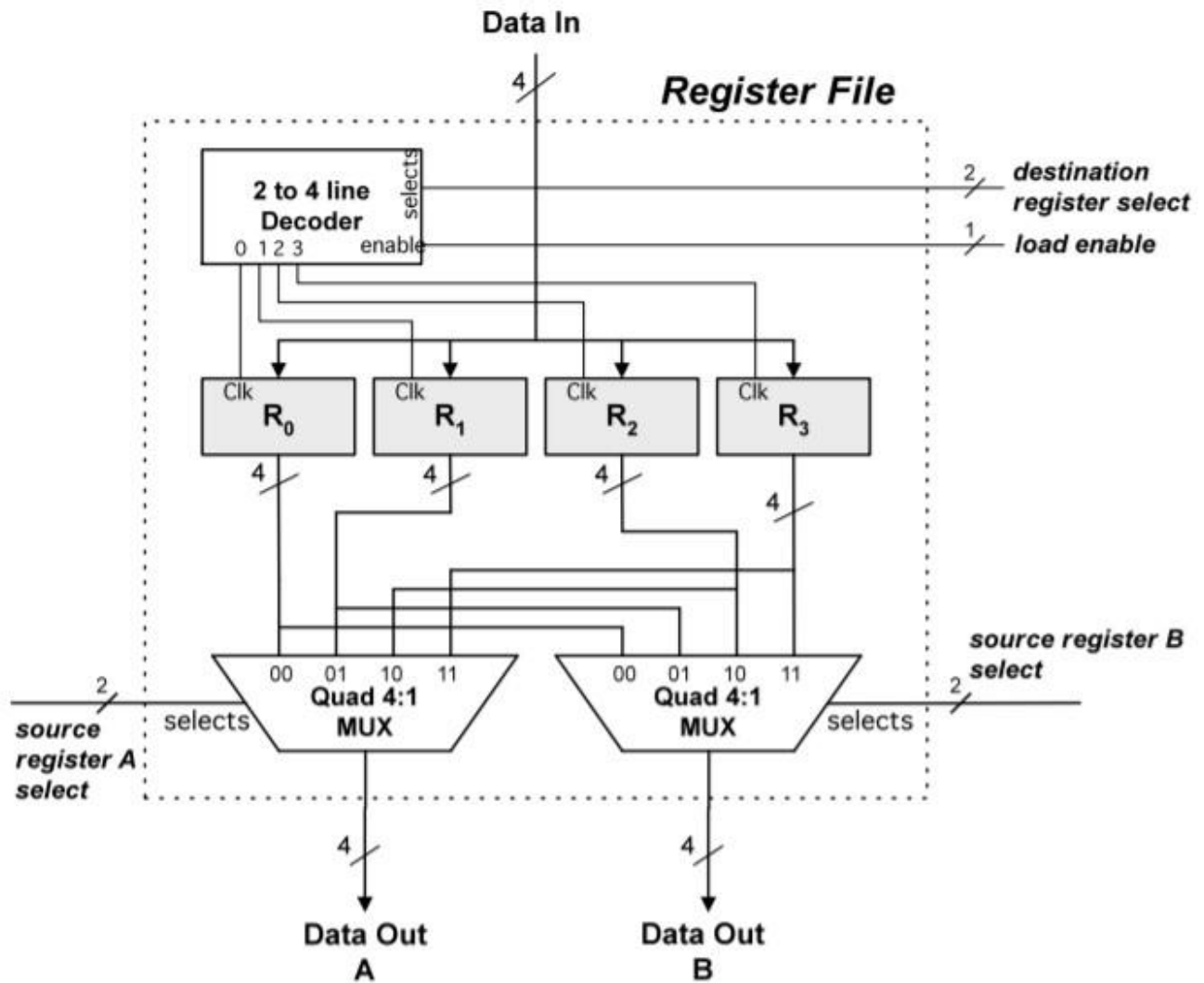
- Read Register1 specifies which register should be read from. That register's data should be presented on the ReadData1 output.
- Read Register2 specifies which register should be read from. That register's data should be presented on the ReadData2 output.
- Write Register specifies which register should be written to.
- Write Data is the data that will be written to the register specified by WriteRegister. On a rising clock edge, the data will be written to the specified register.
- Write Enable, if true, will cause the WriteData to be written to the register specified by Write-Register; otherwise, WriteData will be ignored and no register will be written to.
- Clock conveys the clock signal.

Comprison between Register File and Memory (SRAM)

Register File	Memory (SRAM)
m n -bit storage words	m n -bit storage words
Few words (<256), many ports, <u>dedicated</u> read/write ports, so that read and write operations <u>can</u> be done simultaneously.	Many words (>1024), few ports, <u>shared</u> read/write ports. Read and write operations <u>cannot</u> 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

Here is a picture of what your register file sub circuit, symbolically, looks like. Bold labels indicate that an input/output contains more than 1 bit of data.



Assignment Statements:

1. Design a 16 Register File.
2. Connect Register File in previous problem with ALU that designed in Lab 2.