Memory Elements (II)

CS31

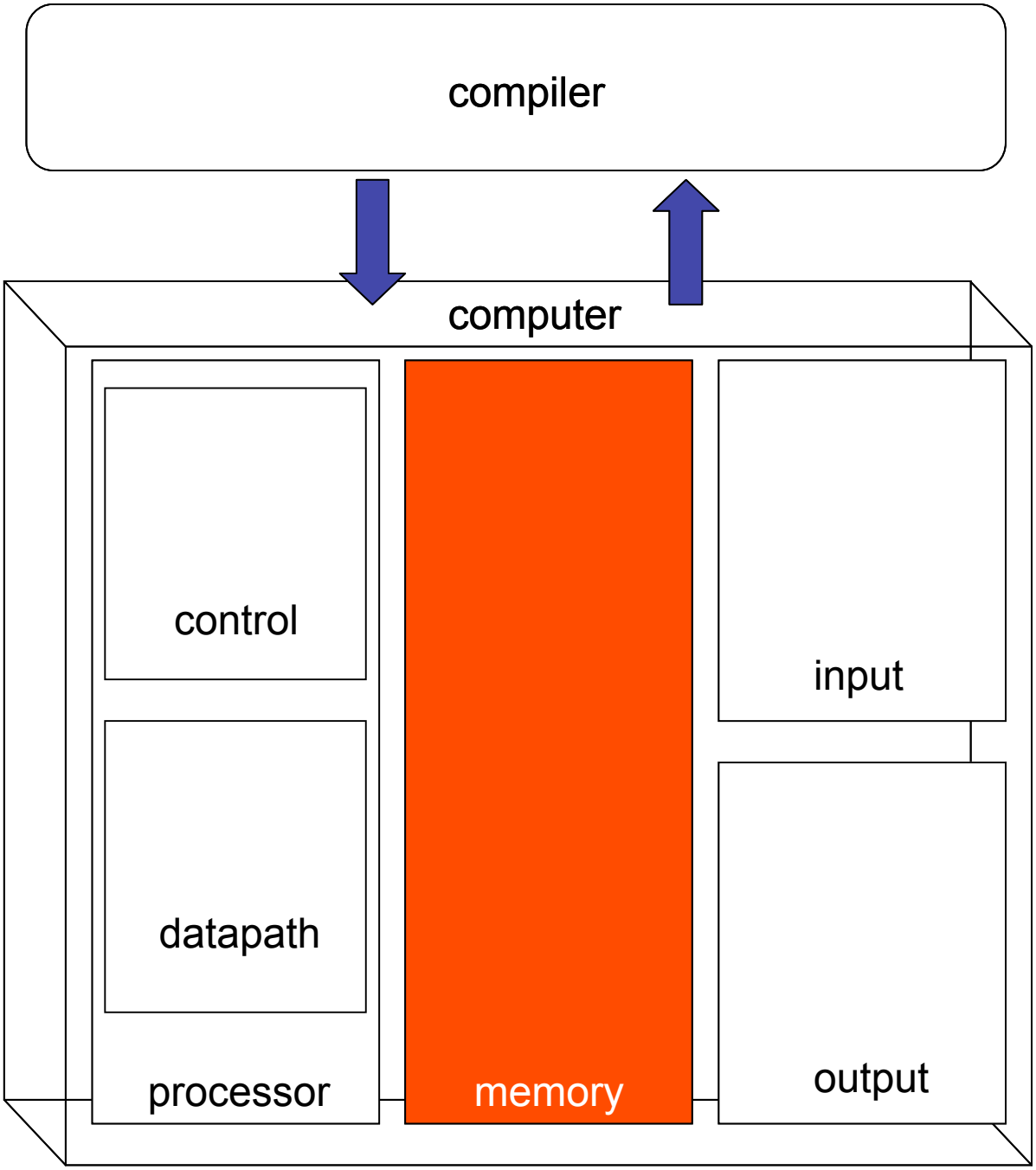
Pascal Van Hentenryck

Overview

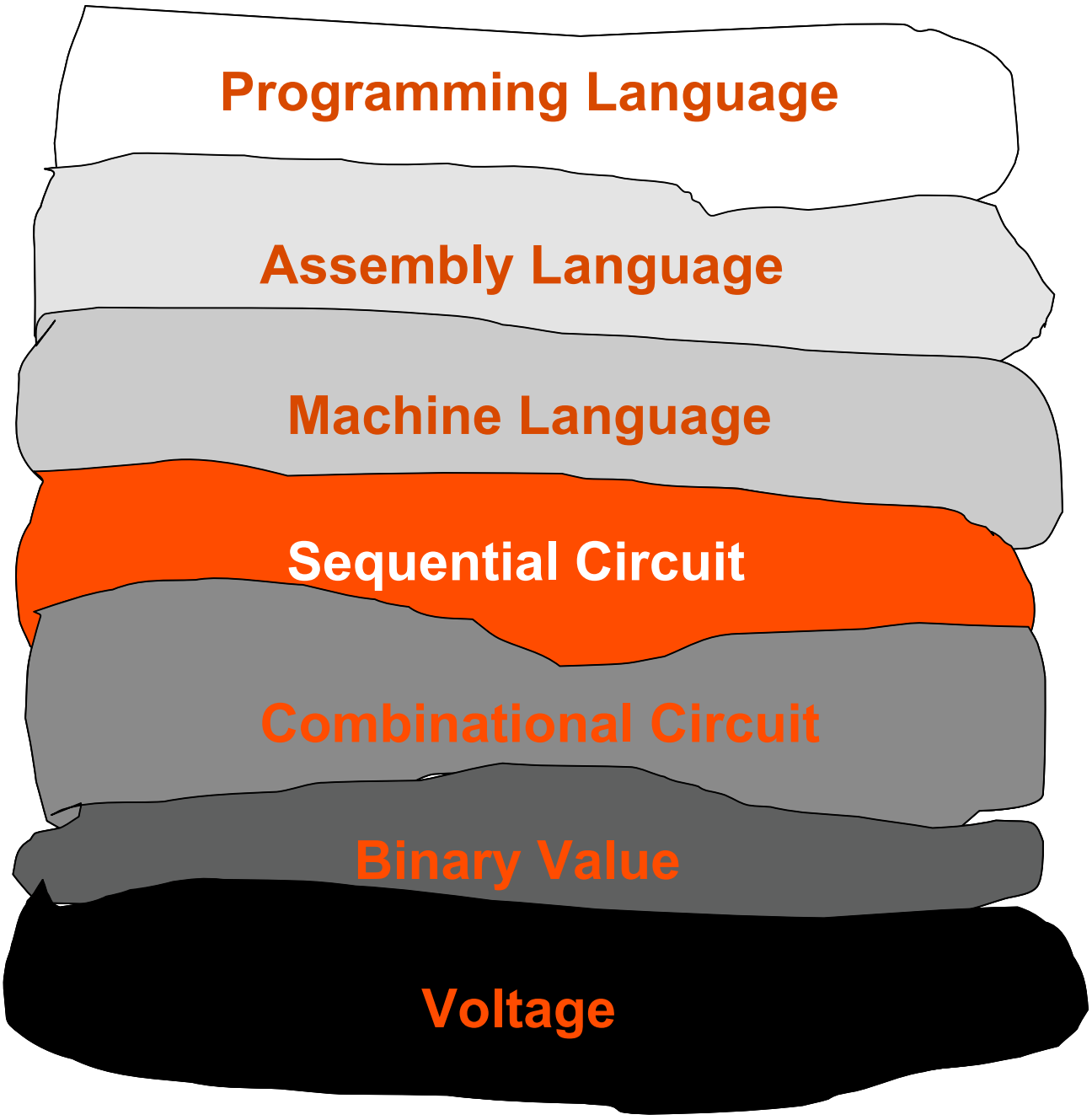
Memory Elements (II)

- registers
- register files

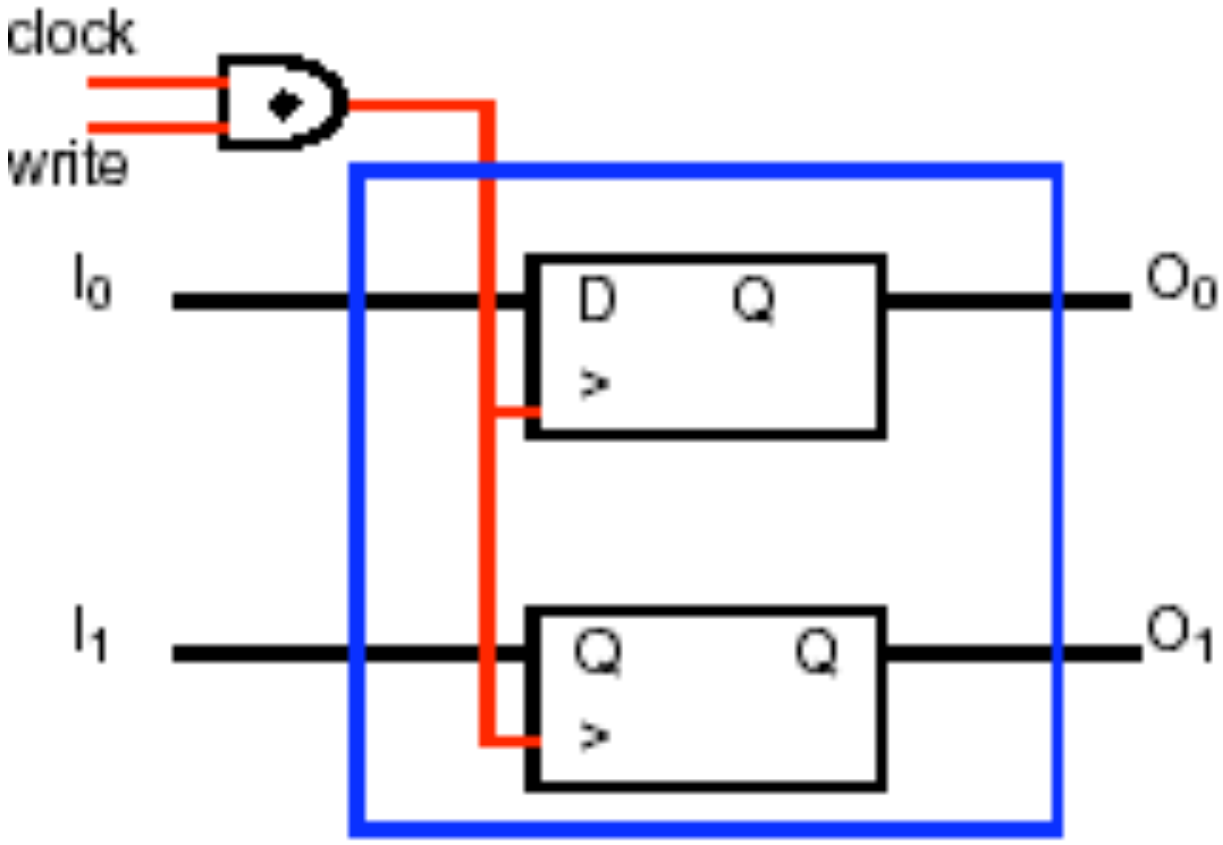
The Big Picture



Abstraction Hierarchy



2-bit Register

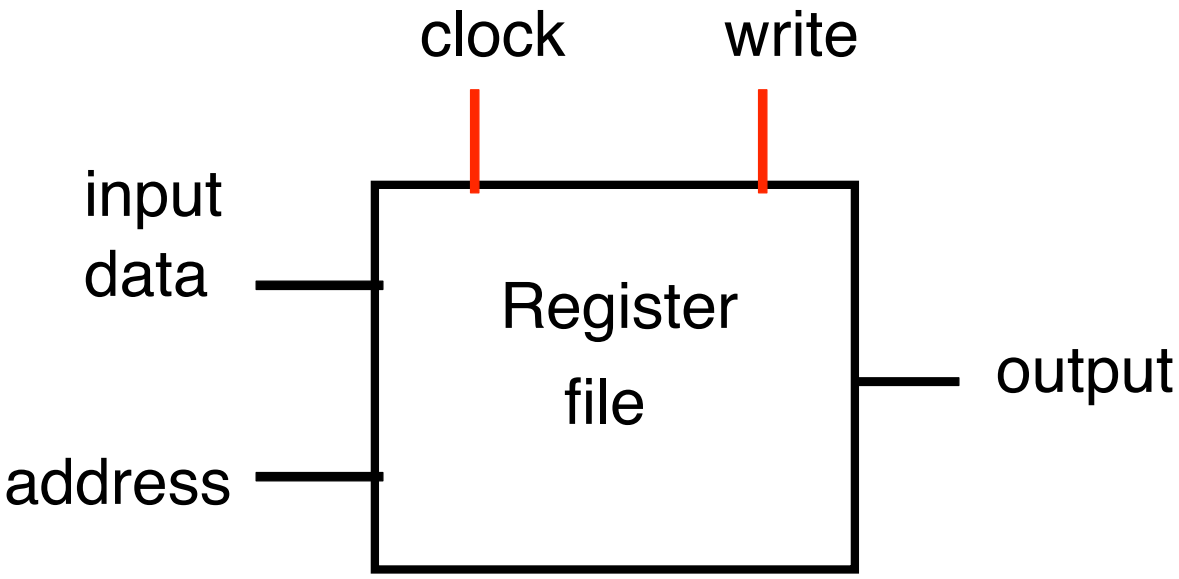


The register is written

- on the clock edge
- whenever the write signal is enabled

Register Files

A register file is a set of registers

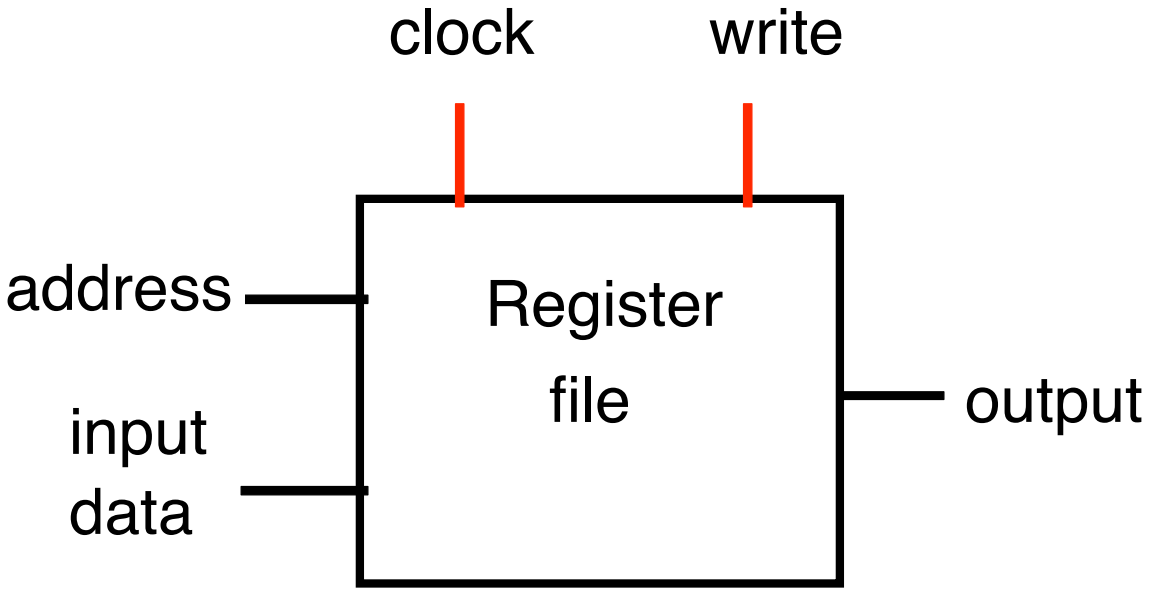


Construct a 4 register file with 8-bit registers

- size of address?
  2 bits addressing 4 registers
- size of input data?
  8-bits to write to the 8-bit registers
- size of output data?
  8-bits, also from the 8-bit registers

1-bit Register File

We want each register storing one bit

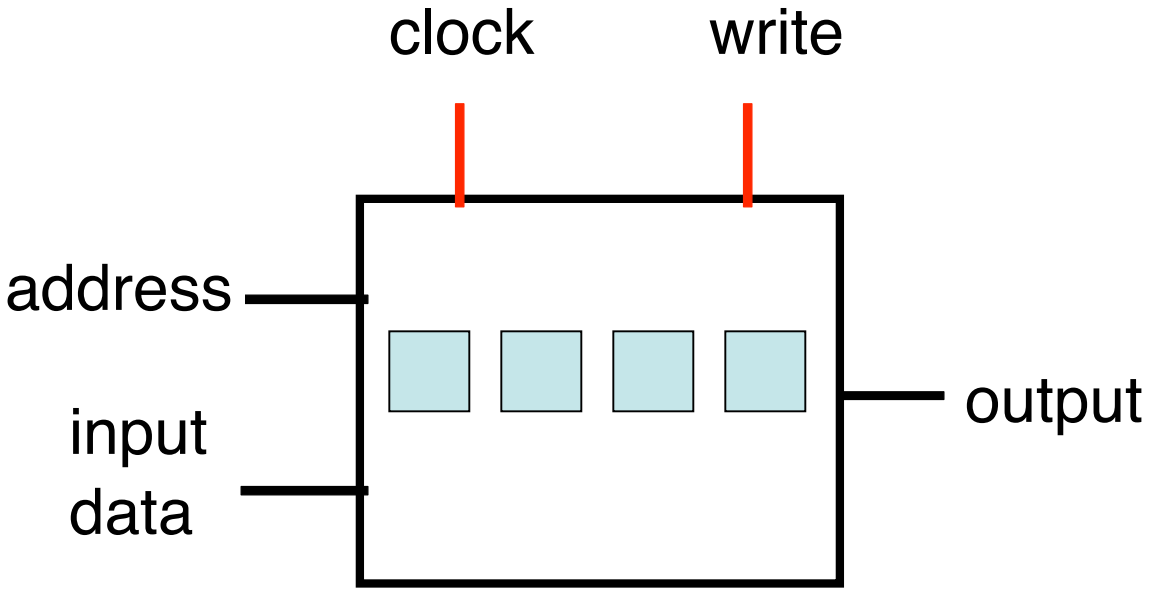


Proceed in two parts

- Read Part
- Write Part

1-bit Register File

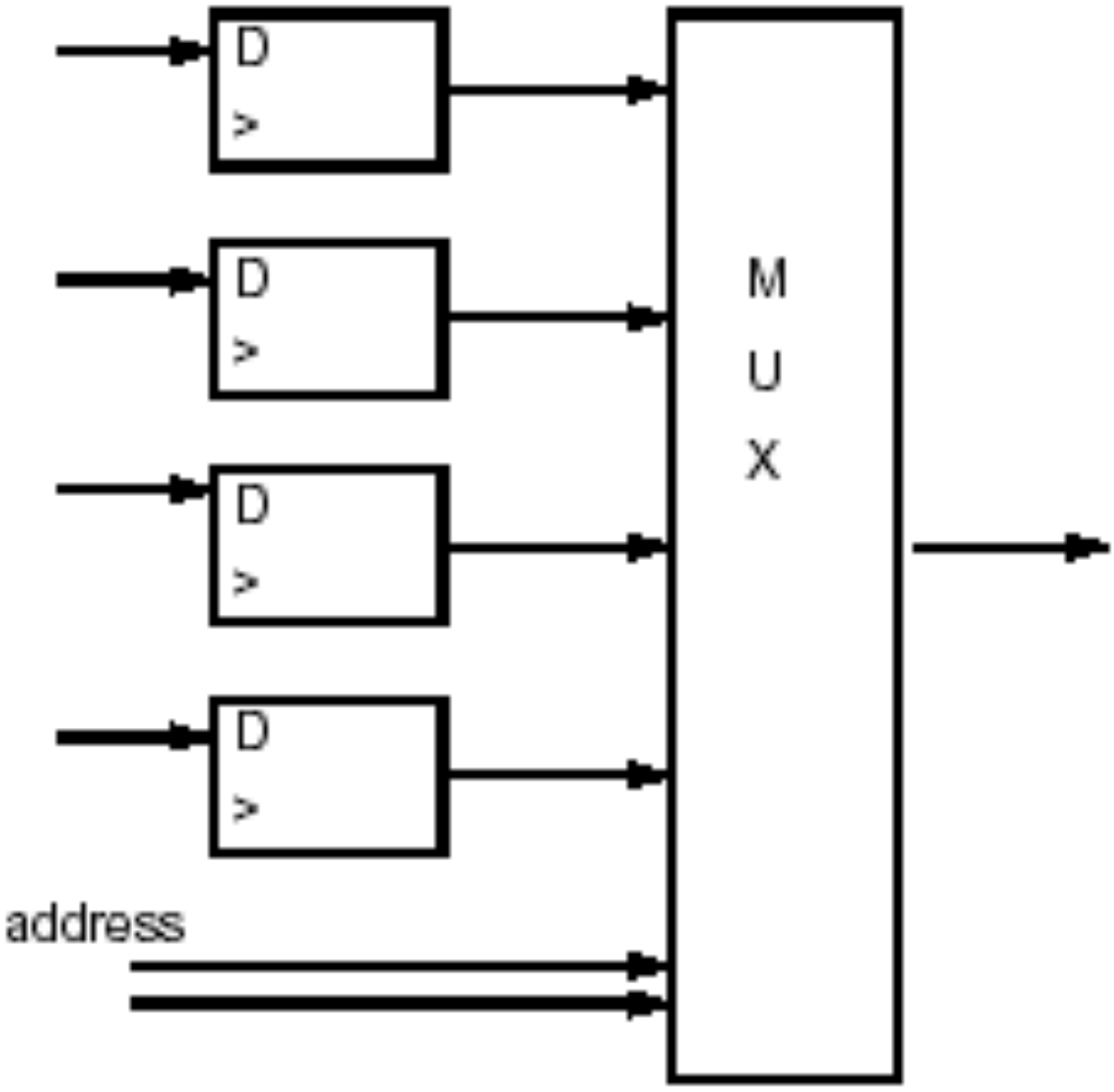
We want each register storing one bit



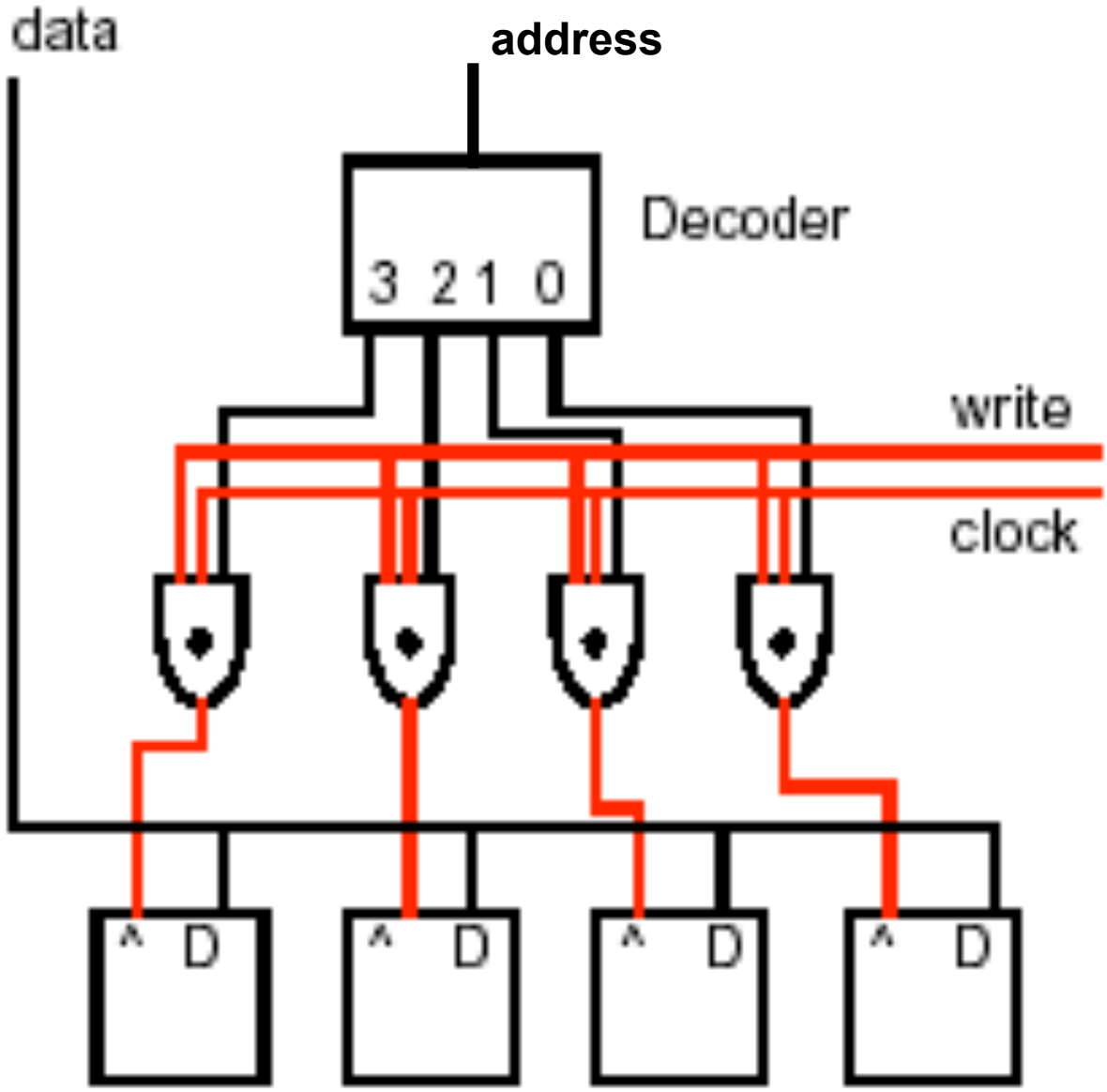
Proceed in two parts

- Read Part
- Write Part

Read Part

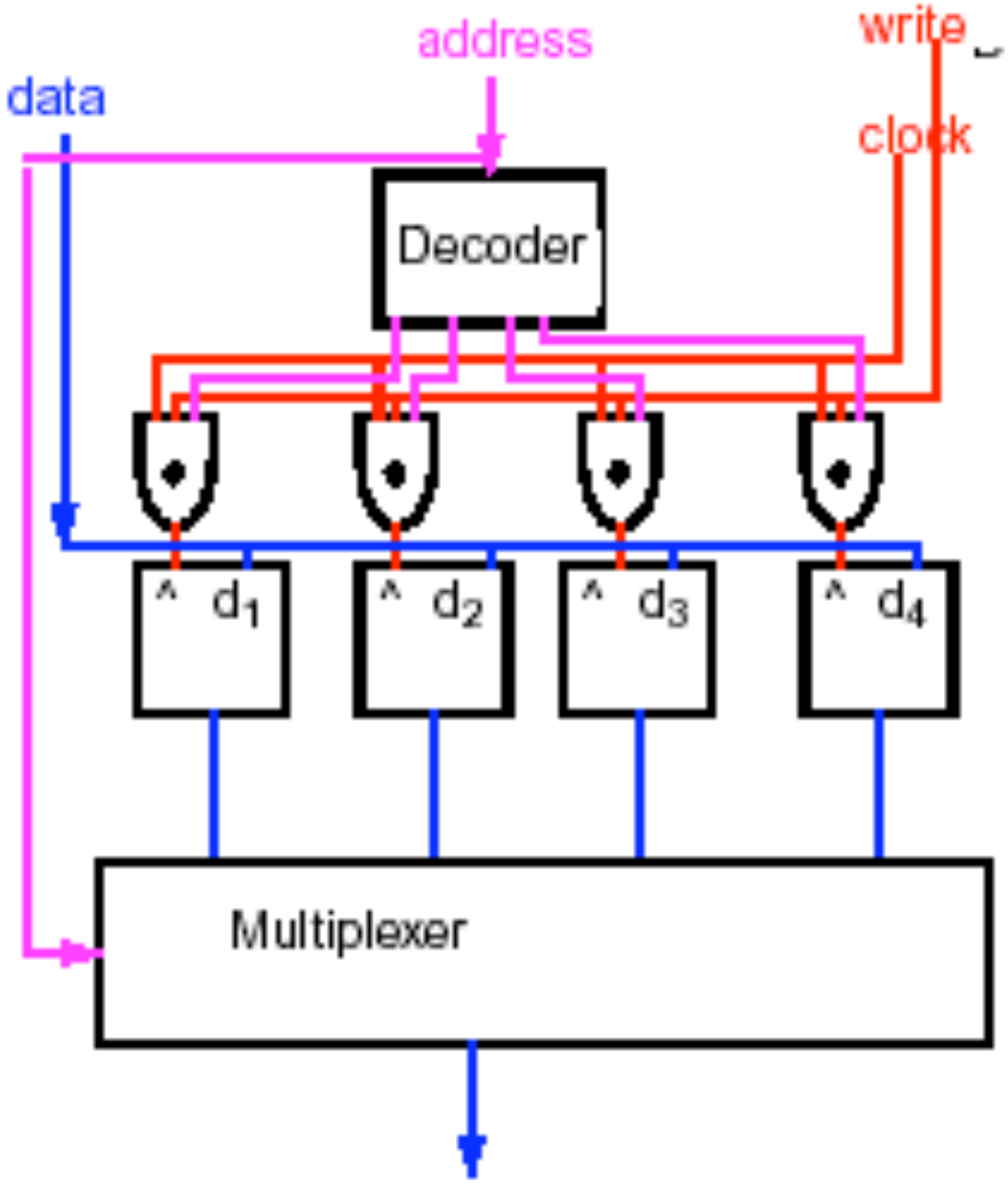


Write Part



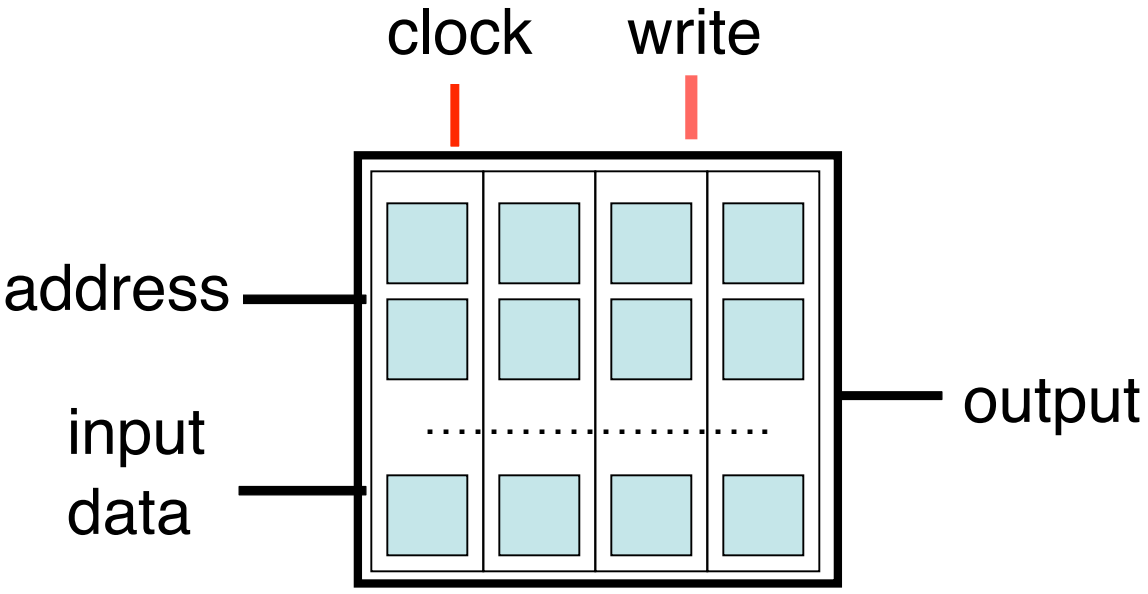
Note the clock input of the flip-flop

The Whole Thing



1-bit Register File

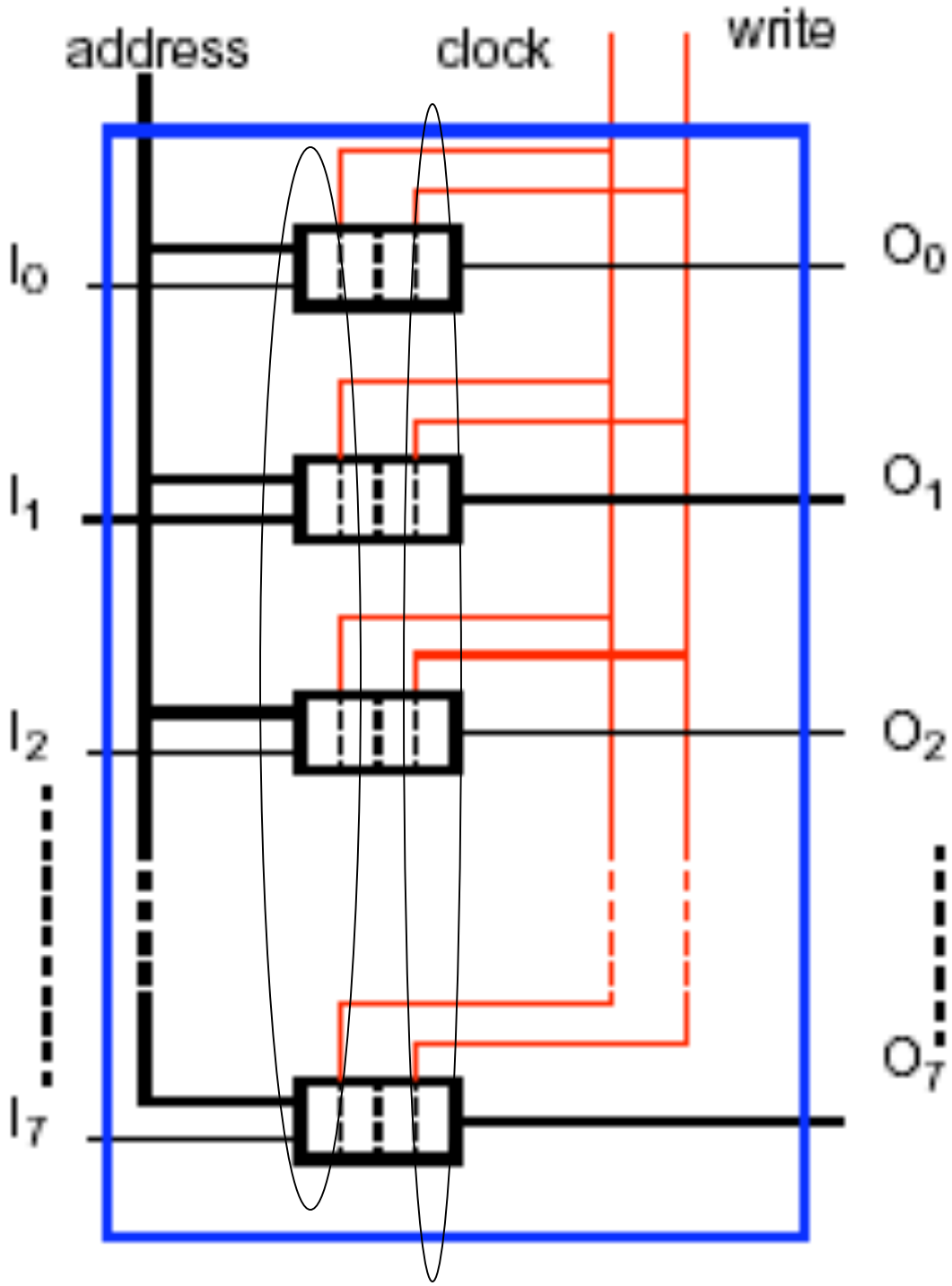
We want each register storing one bit



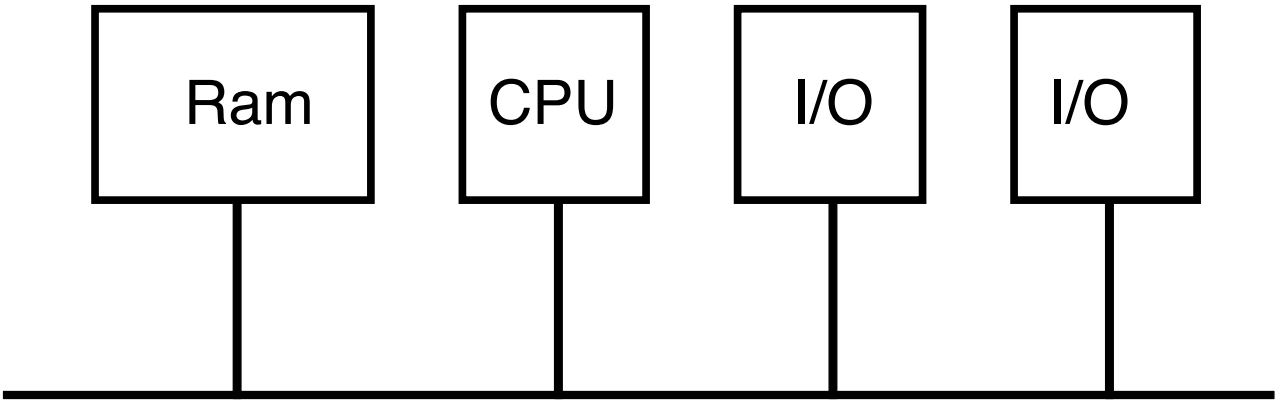
Proceed in two parts

- Read Part
- Write Part

8-bit Register File with 4 Registers



Bus



Motivation

- Many devices need to communicate with each other
- Many wires to connect them all
- A bus is a way to share the wires

Problem to solve

- Making sure that no two devices write on the bus at the same time

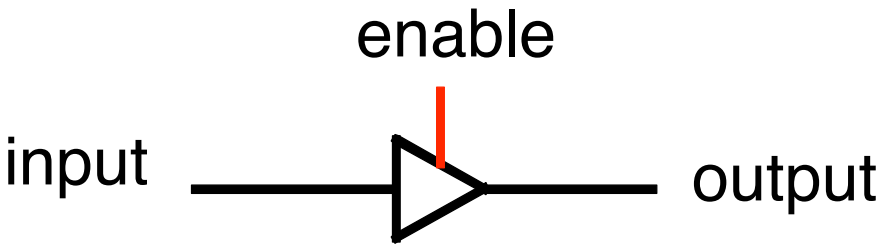
Tri-state

Basic Idea: three-states

- 1, 0, high impedance

Behavior

- when enable is asserted, the output is the input
- when enable is deasserted, the output has a high impedance value
- a high impedance value means that the output can be driven by some other device

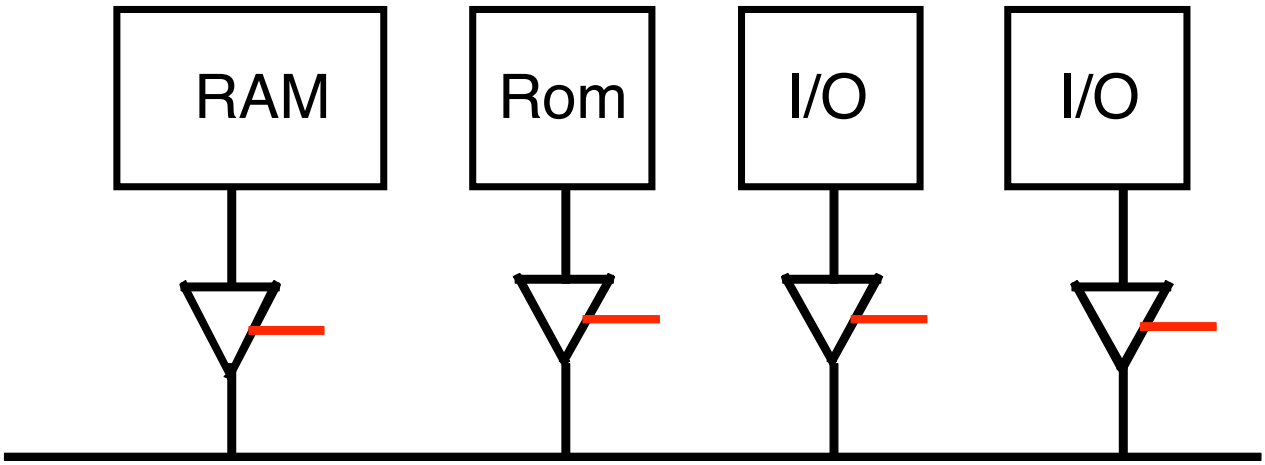


High Impedance

When an output is in the high impedance state, it is as though there were no connection

- Think of a very high, essentially infinite, resistance between the power supply, ground lines and the output
- Think of a water pipe
  1. Value 1 is 10 gallons/sec (high)
  2. Value 0 is 2 gallons/sec (low)
  3. High impedance is 0 gallons/sec (off)

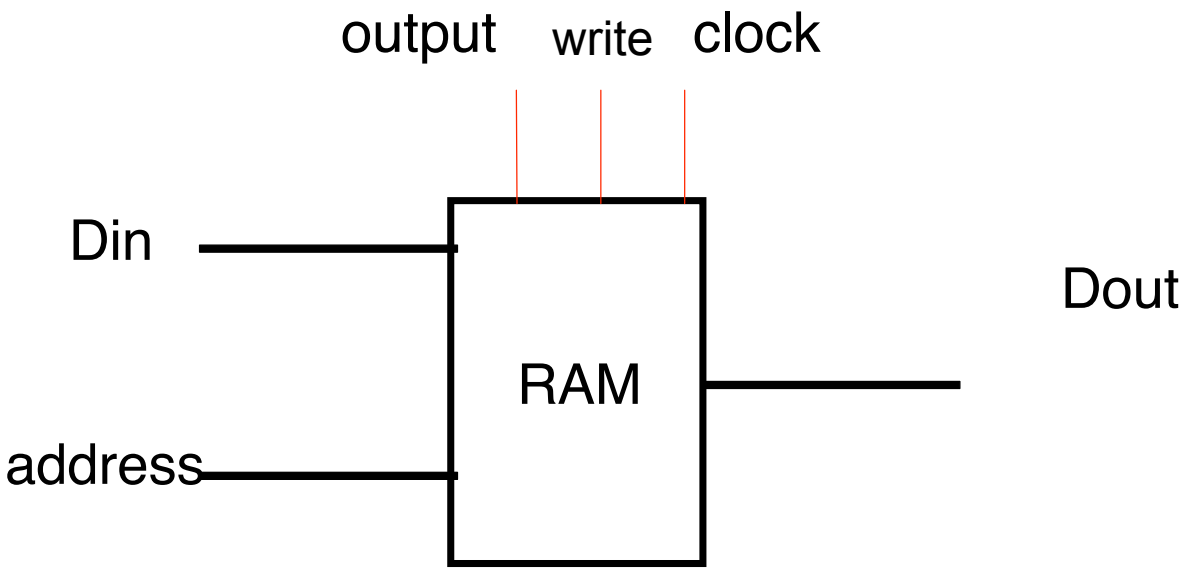
Bus Again



At any time

- only one of the tri-states can be enabled

RAM



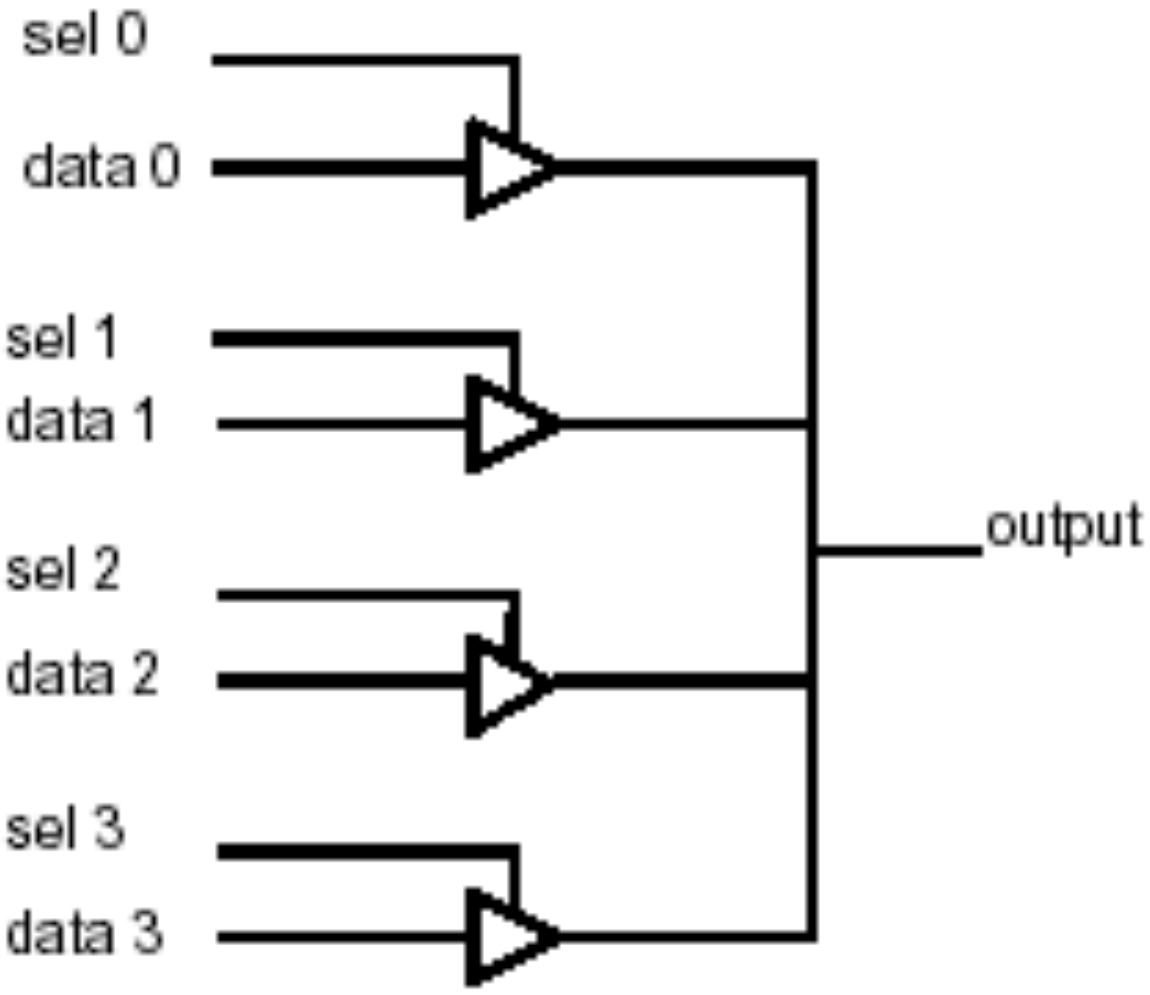
Control Bits

- Output bit: controls whether the data should be driven on Dout; useful to connect to a bus
- Write bit: controls whether the input should be written at the address

RAM Implementation

Too large to build a multiplexer

- use tri-states instead



RAM Implementation

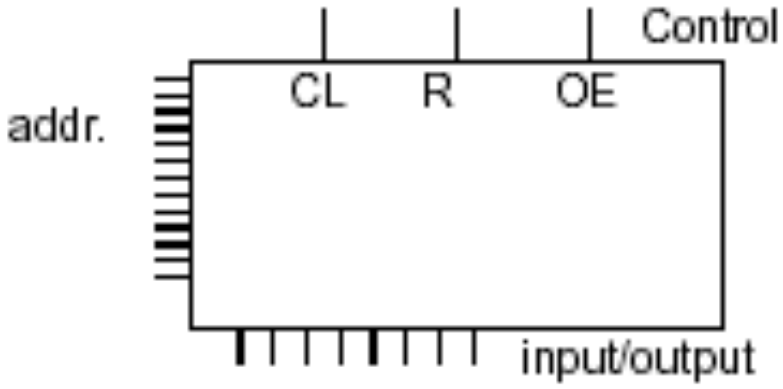
Basic Idea

- Integrate the tri-state in the flip-flop so that it can be connected to a shared wire
- One shared wire per output
- One Decoder

Main difference from a register file

- No multiplexer

RAM/ROM in Diglog



Basic Feature

- Inputs/Outputs share the same ports

Basic Control of the RAM/ROM

CL	R	OE	Action
0	1	0	read
0	0	x	write
0	1	1	noop
1	x	x	noop

The Good News

We have everything we need to build a computer now!