

# CSC258 Winter 2016

## Lecture 7



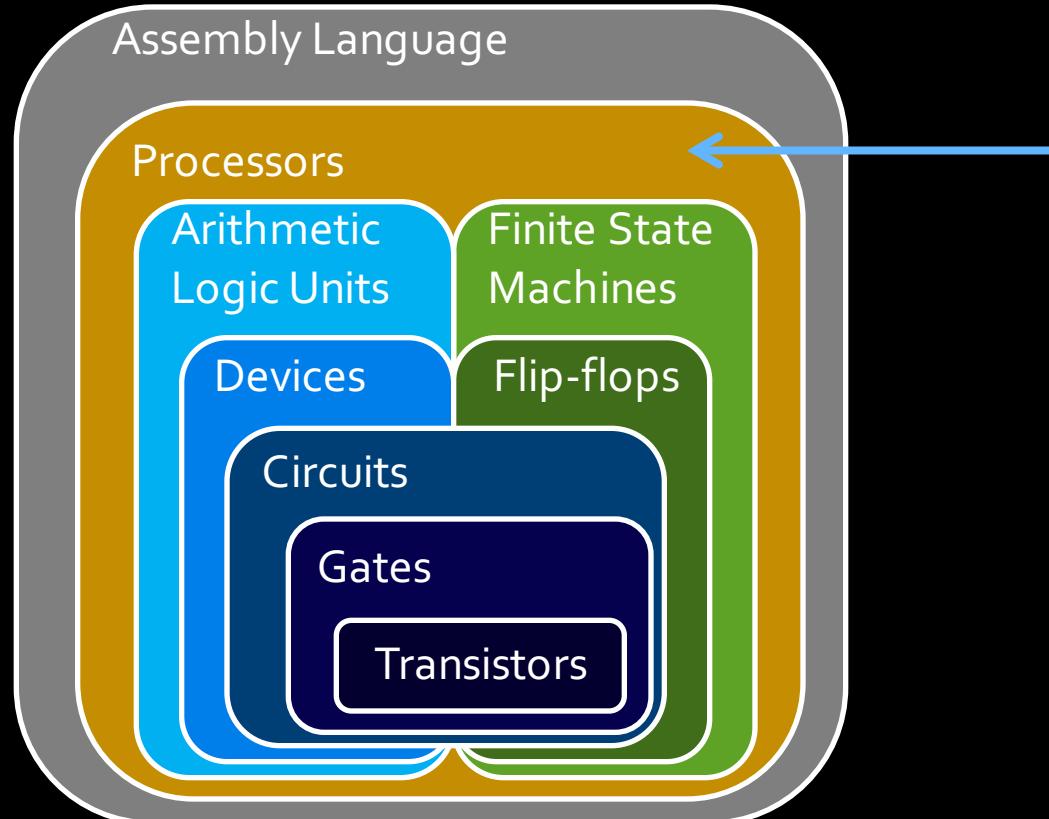
**KEEP  
CALM  
AND  
NEVER  
GIVE UP**



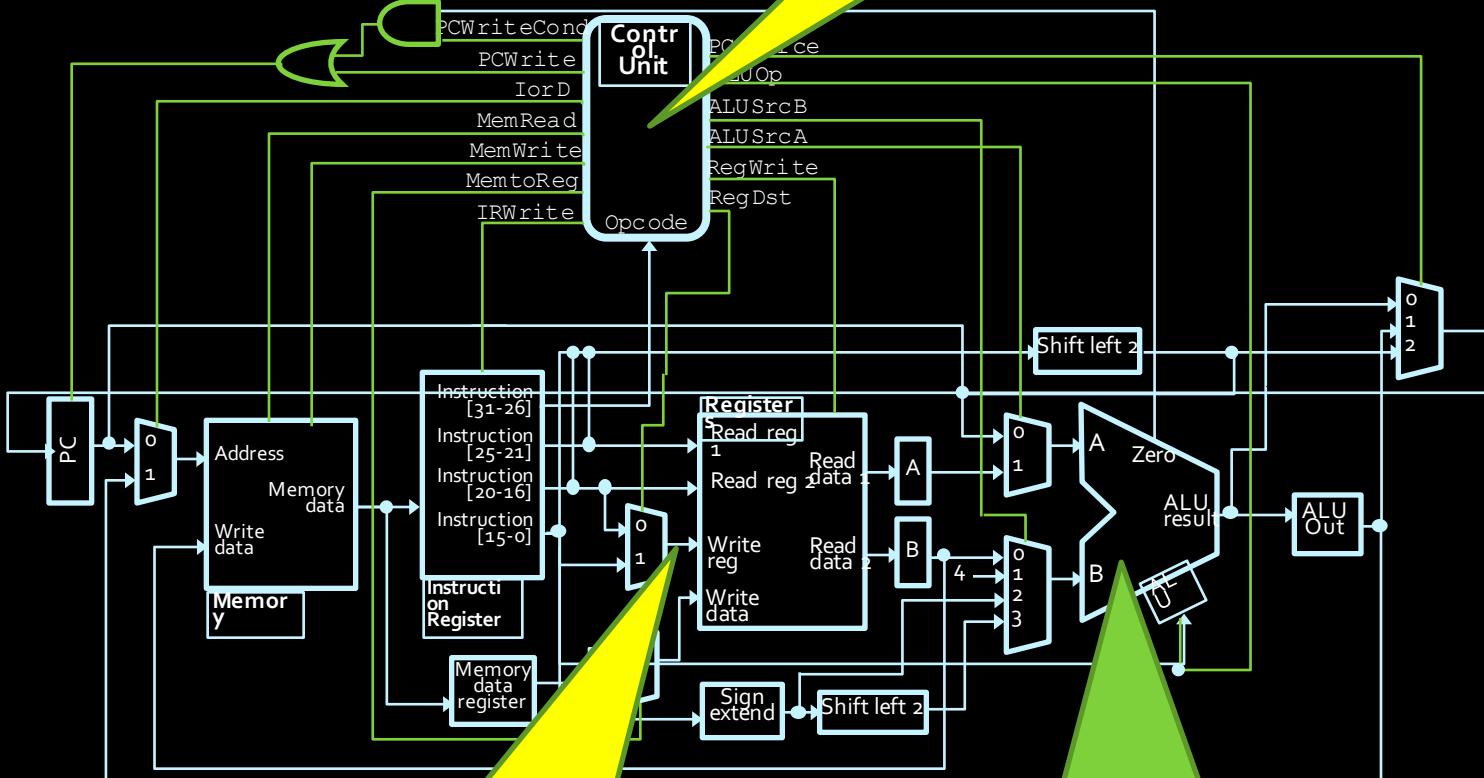
Drop date: Mar 6

JK, don't do it yet, you will get your midterm marks back before that, then you can decide.

# Recap: We are here



# The Blueprint of a microprocessor



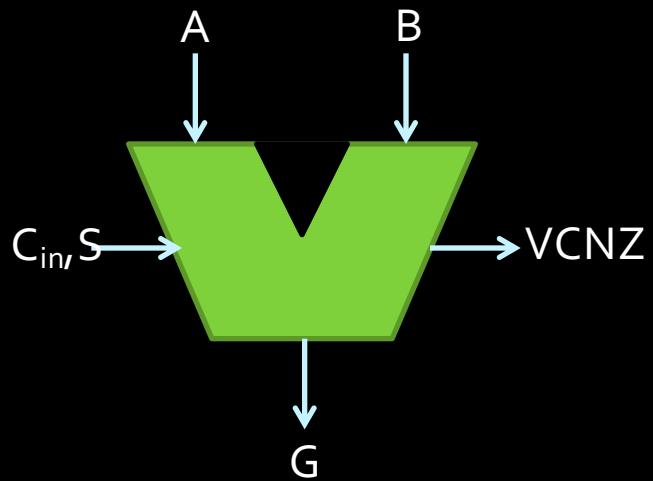
The Storage Thing

The Controller Thing

The Arithmetic Thing

# We've learned the arithmetic thing

ALU



With ALU, we can do addition, subtraction, logic operations, etc.

Multiplication was a bit trickier

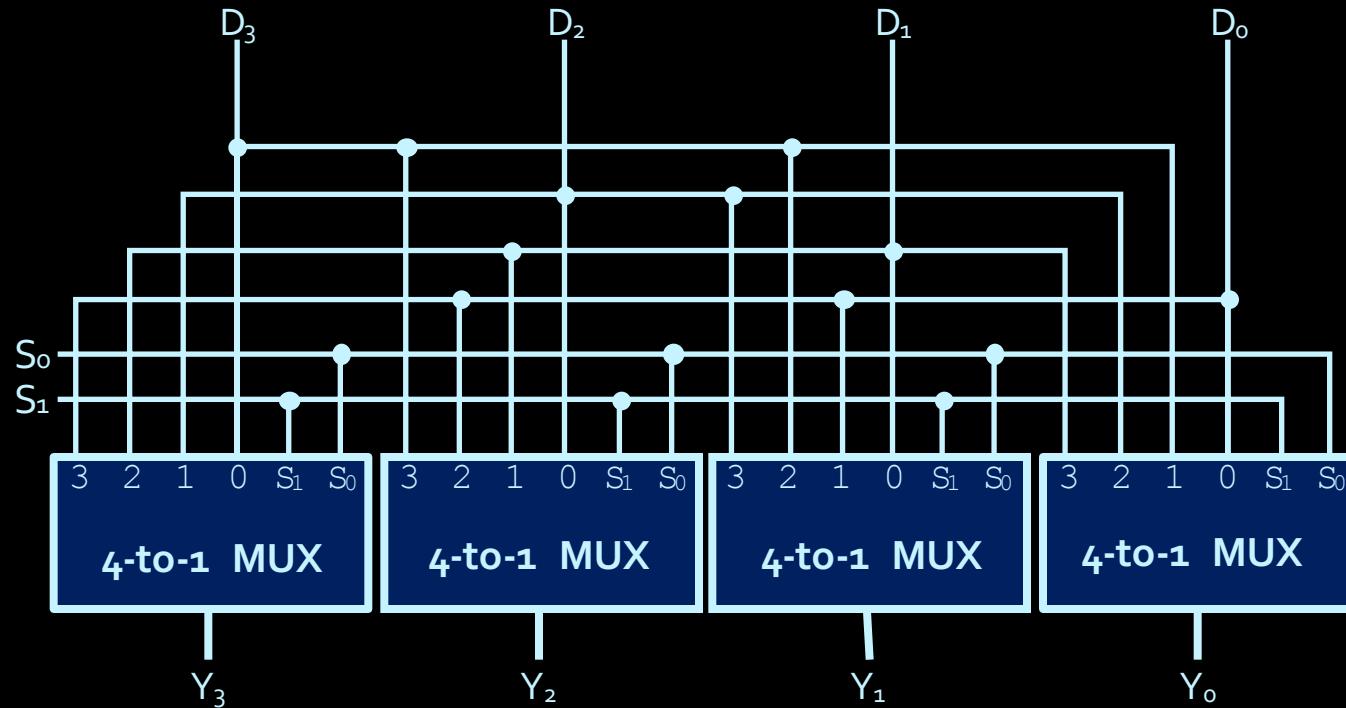
- Booth algorithm

To implement multiplication using Booth algorithm, we basically repeatedly do three things

- Addition
- Subtraction (kind-of an addition)
- Shift

We learned how to do addition and subtraction.  
How to do shift?

# Shifter unit



- **Barrel shifter** acts alongside ALU to shift data elements according to  $S_0$  and  $S_1$ .

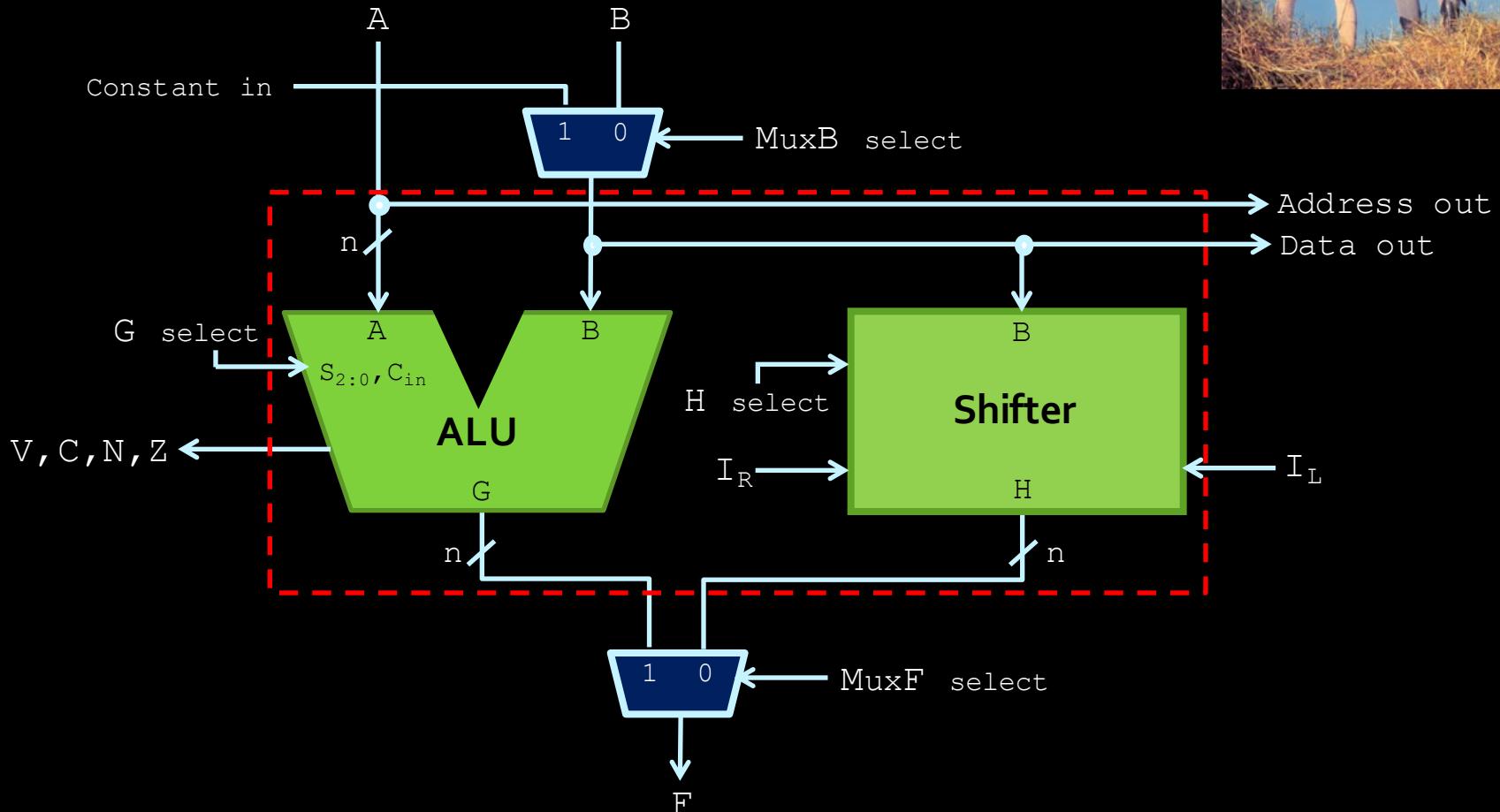
If  $S_1S_0 = 00$ :  $Y_3Y_2Y_1Y_0 = D_3D_2D_1D_0$   
If  $S_1S_0 = 01$ :  $Y_3Y_2Y_1Y_0 = D_2D_1D_0D_3$

# ALU partnered with Shifter

- We can implement all kinds of arithmetic functions, like...
- $(A+B)*(A-B) - 3*A*B$



# Function Unit



So where do  $A$  and  $B$  come from?

# The “Storage Thing”

aka: the register file and main memory



# Computer memory hierarchy

In terms of data (food) access speed

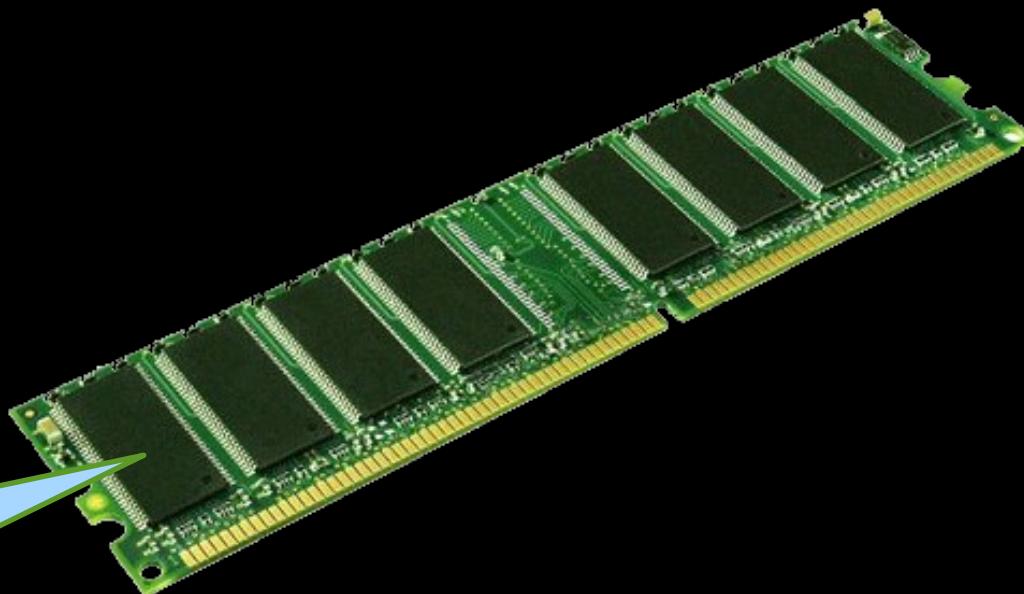
- Register: that plate in front of you
- Cache: the fridge in the kitchen
- Memory: the grocery store downstairs
- Hard disk: the farm in the suburb
- Network: the farm in a different country

# Memory and registers

- There are units in the CPU that store multiple data values for use by the CPU:
  - Registers: Small number of fast memory units that allow multiple values to be read and written simultaneously.
  - Main memory: Larger grid of memory cells that are used to store the main information to be processed by the CPU.

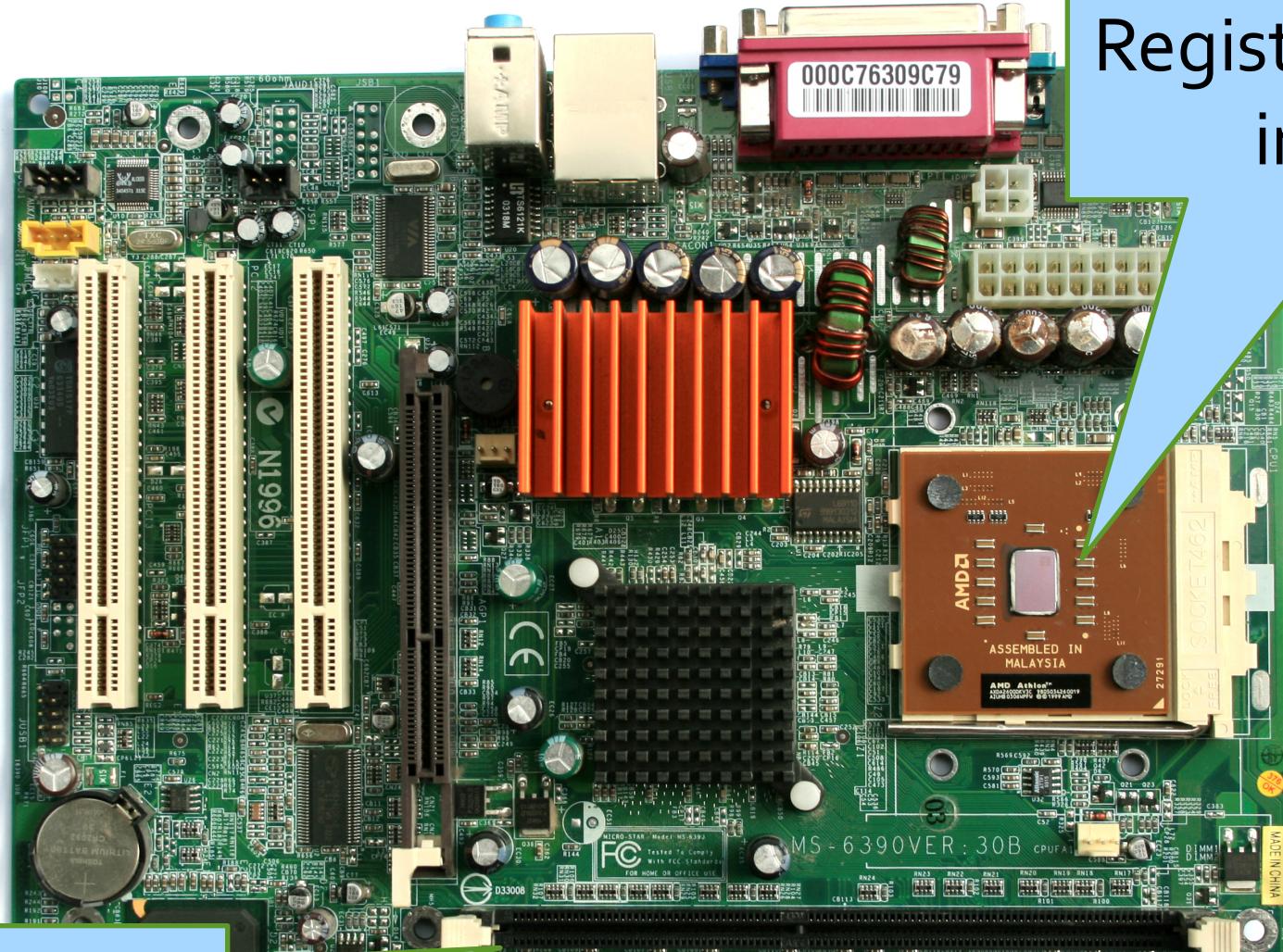


Registers are in here

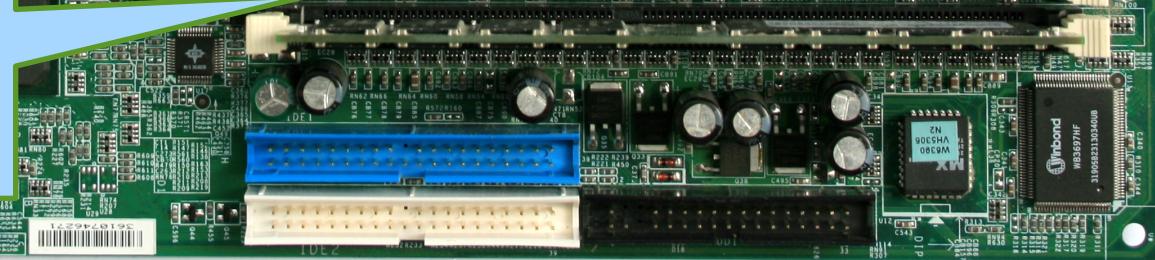


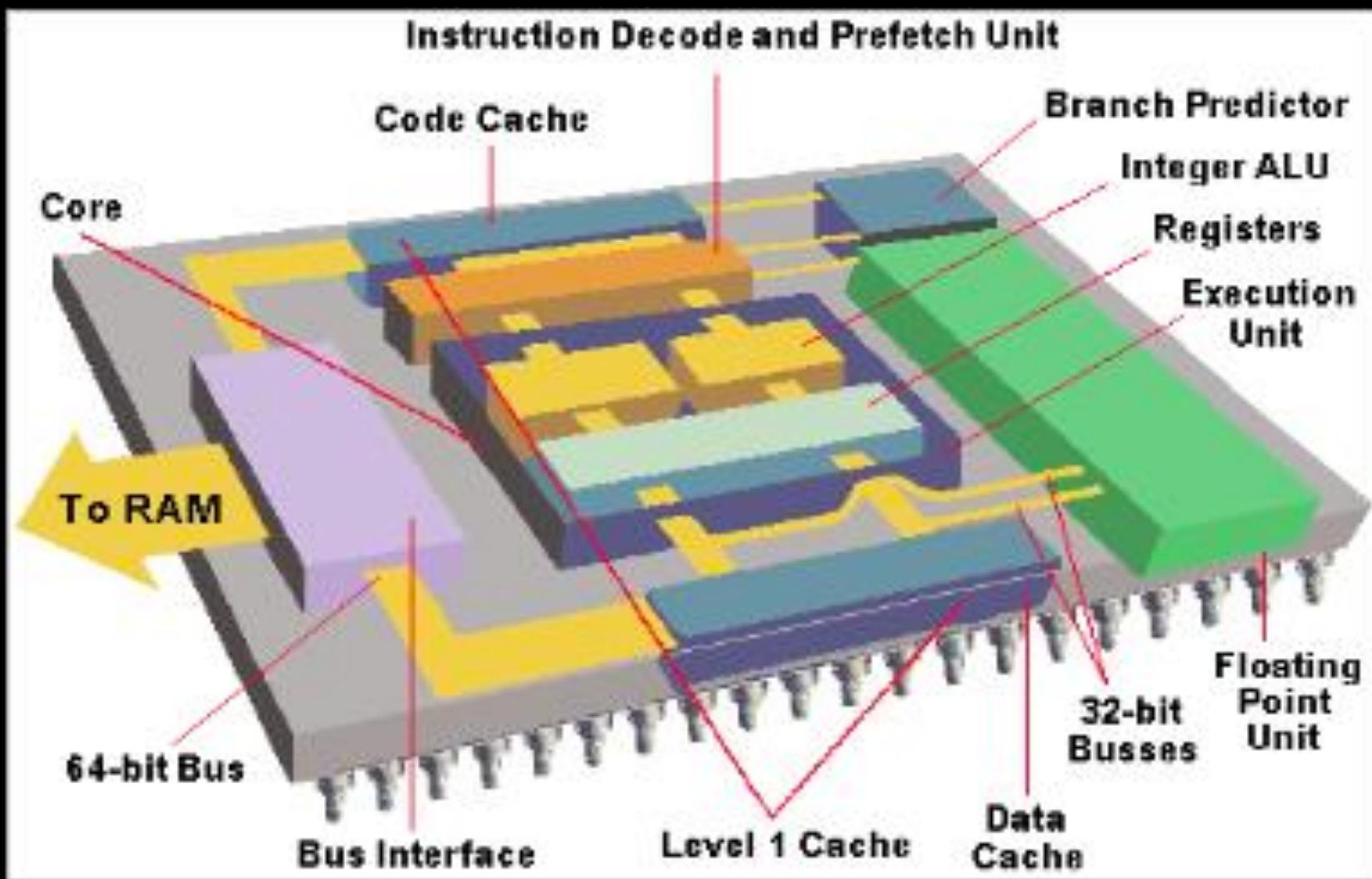
Memory is in here

Registers are in here  
in the CPU



Memory chips are  
plugged in here

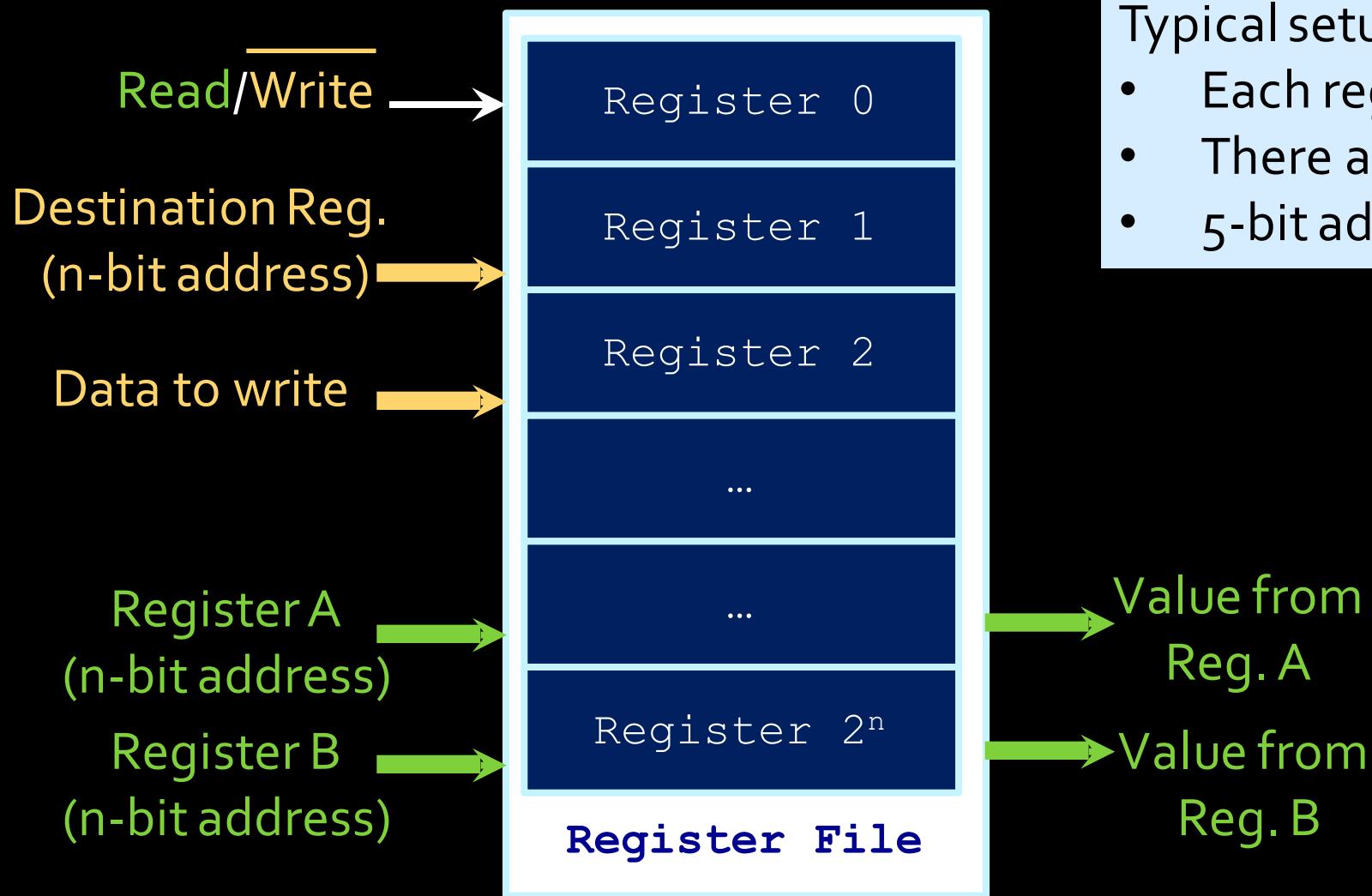




# Register file

An array of registers in the CPU

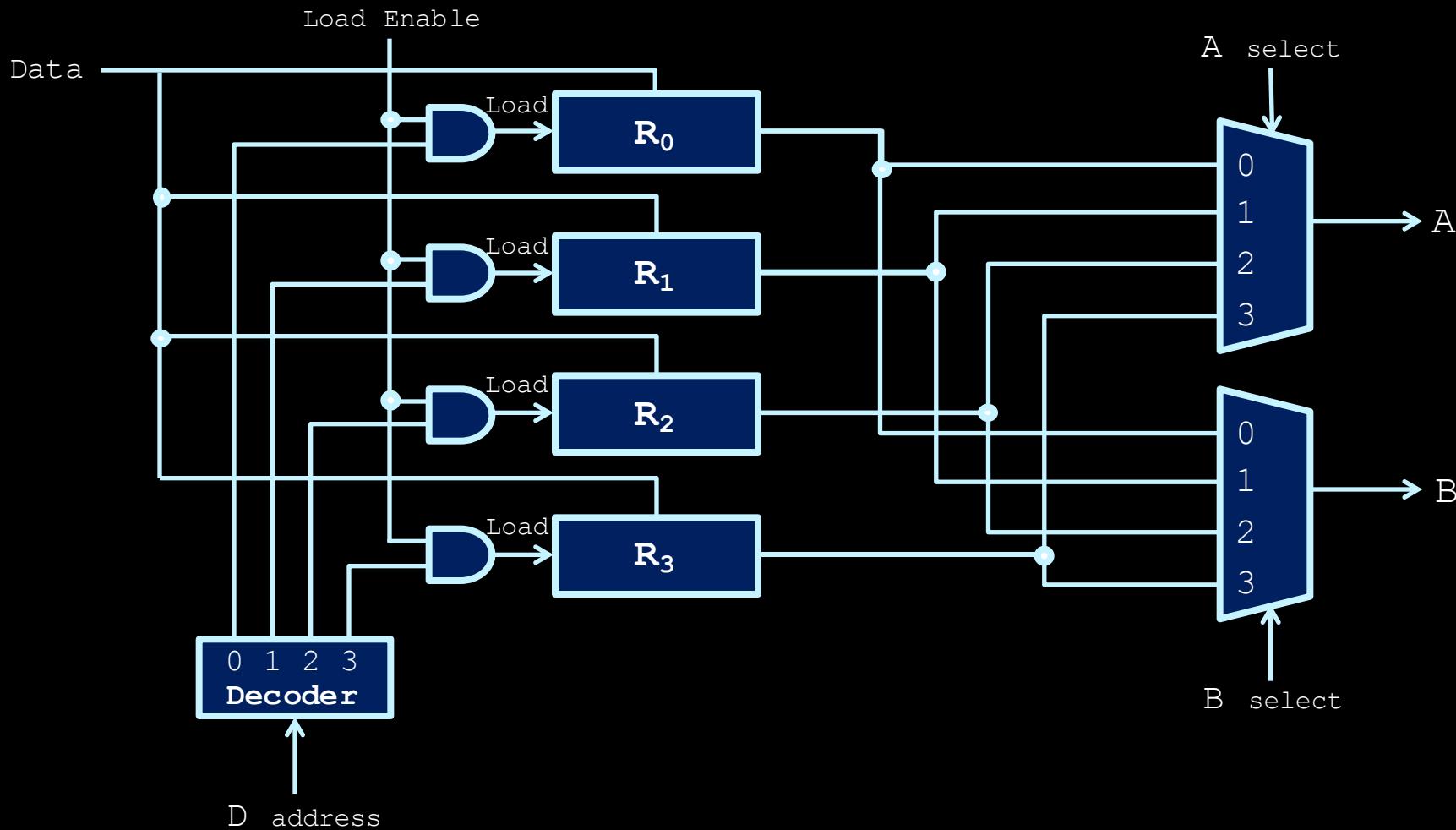
# Register File Functionality



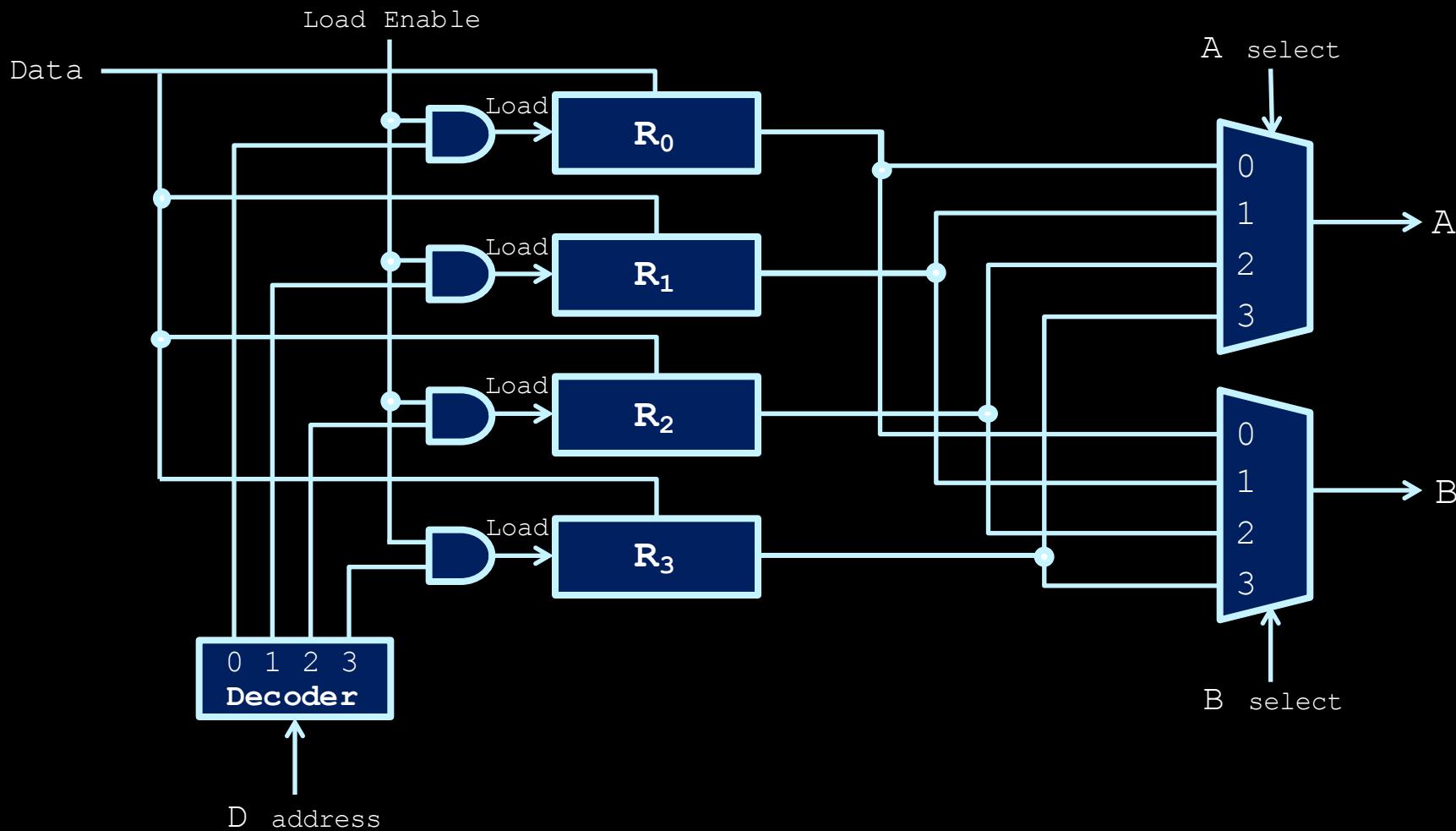
Typical setup (MIPS):

- Each register is 32-bit
- There are 32 registers.
- 5-bit address

# Register File - Write Operation



# Register File - Read Operation

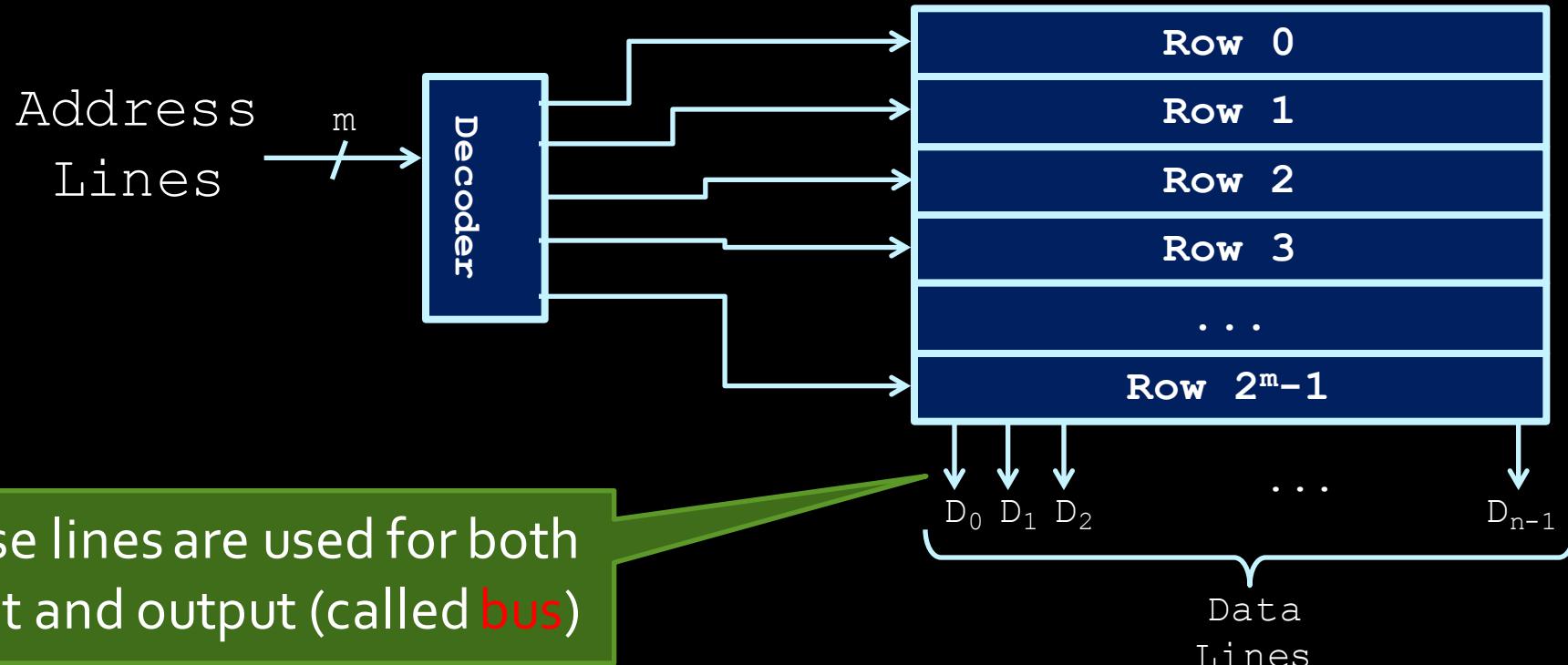


# The main memory

An array of memory units

# Electronic Memory

- Like register files, main memory is made up of a decoder and rows of memory units.



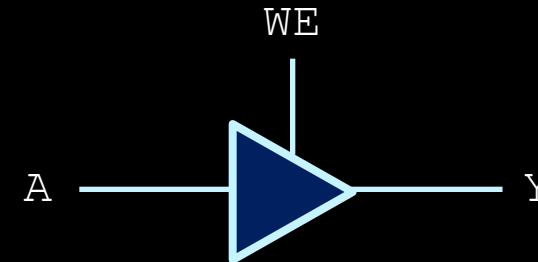
# One-hot decoder

- The decoder takes in the m-bit binary address, and activates a single row in the memory array.

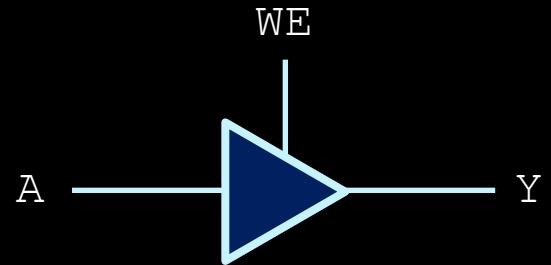
A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	O <sub>7</sub>	O <sub>6</sub>	O <sub>5</sub>	O <sub>4</sub>	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
...										
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

# Controlling the flow

- Since some lines (buses) will now be used for both input and output, we introduce a (sort of) new gate called the **tri-state buffer**.
- When WE (write enable) signal is low, buffer output is a **high impedance** signal.
  - The output is neither connected to high voltage or to the ground.



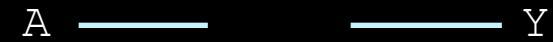
WE	A	Y
0	x	z
1	0	0
1	1	1



$WE = 1$

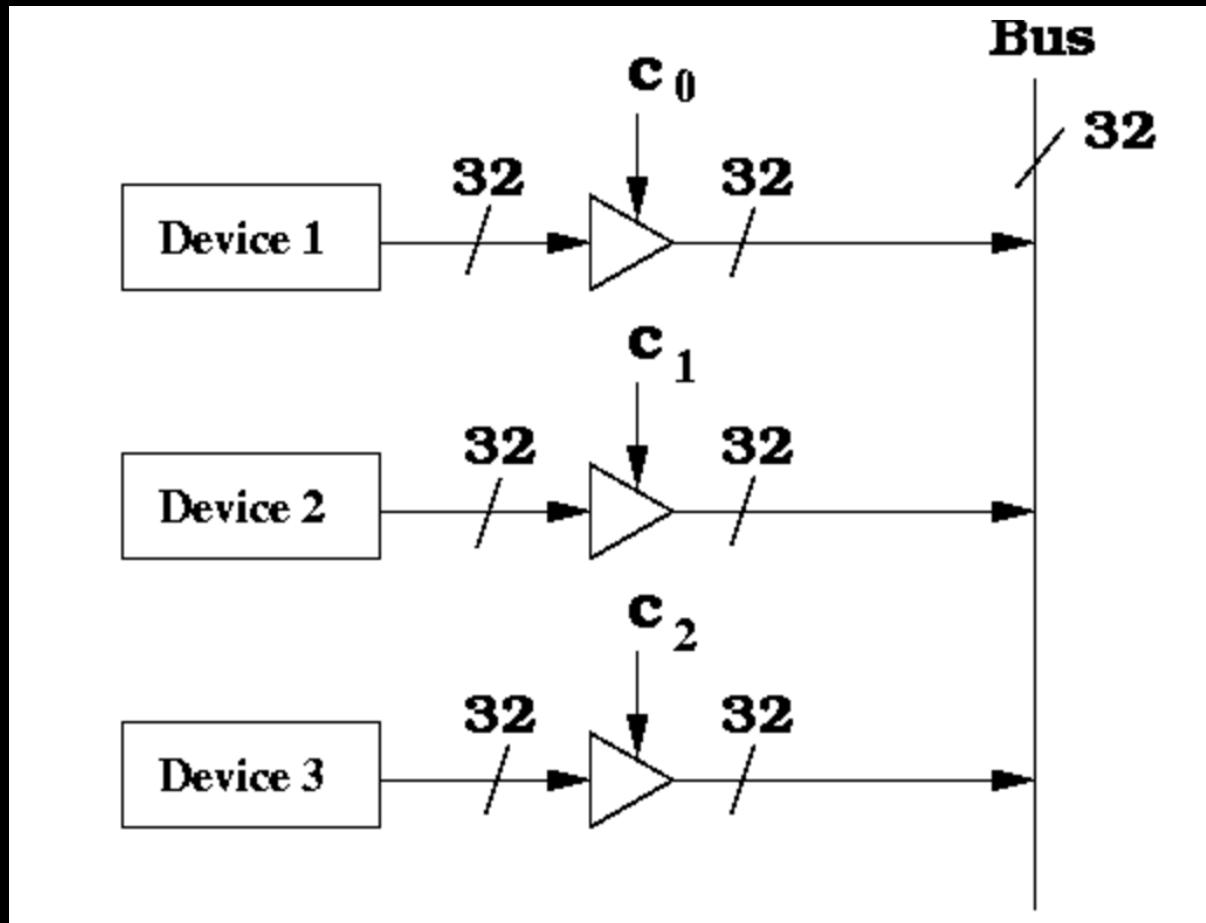


$WE = 0$



WE	A	Y
0	x	z
1	0	0
1	1	1

# Control the flow using tri-state buffer



Control  $c_0$   $c_1$  and  $c_2$  so that only one of the devices output is written to the bus.

In general, the bus can be read by multiple devices at the same time but can only be written by one device at a time.

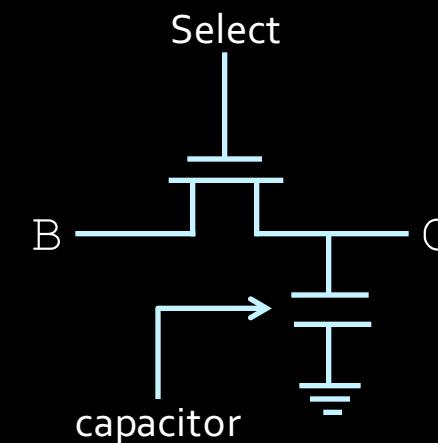
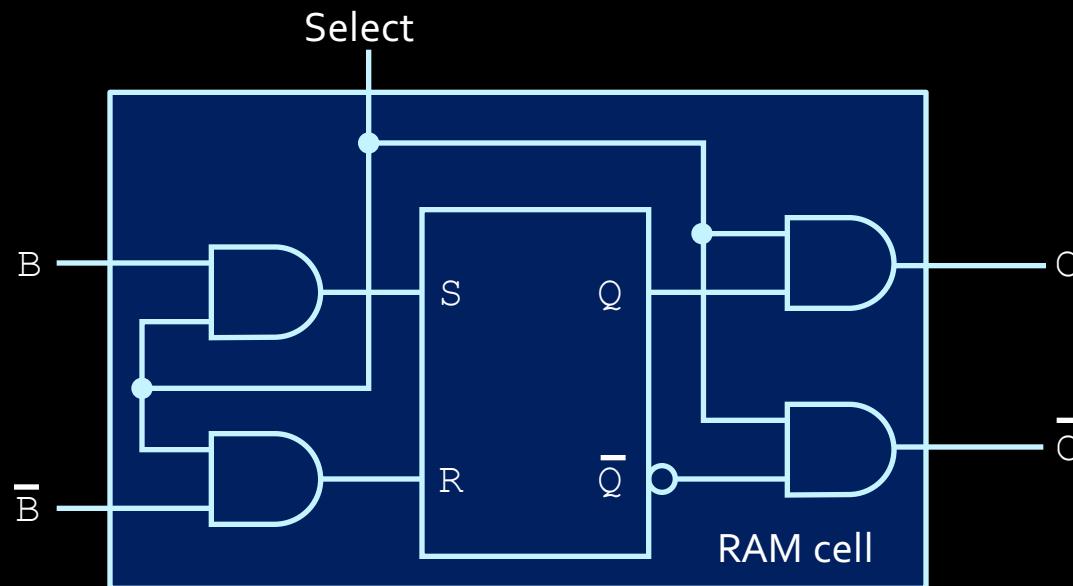
# RAM Storage Cell

what stores each bit of the memory

# Storage cells

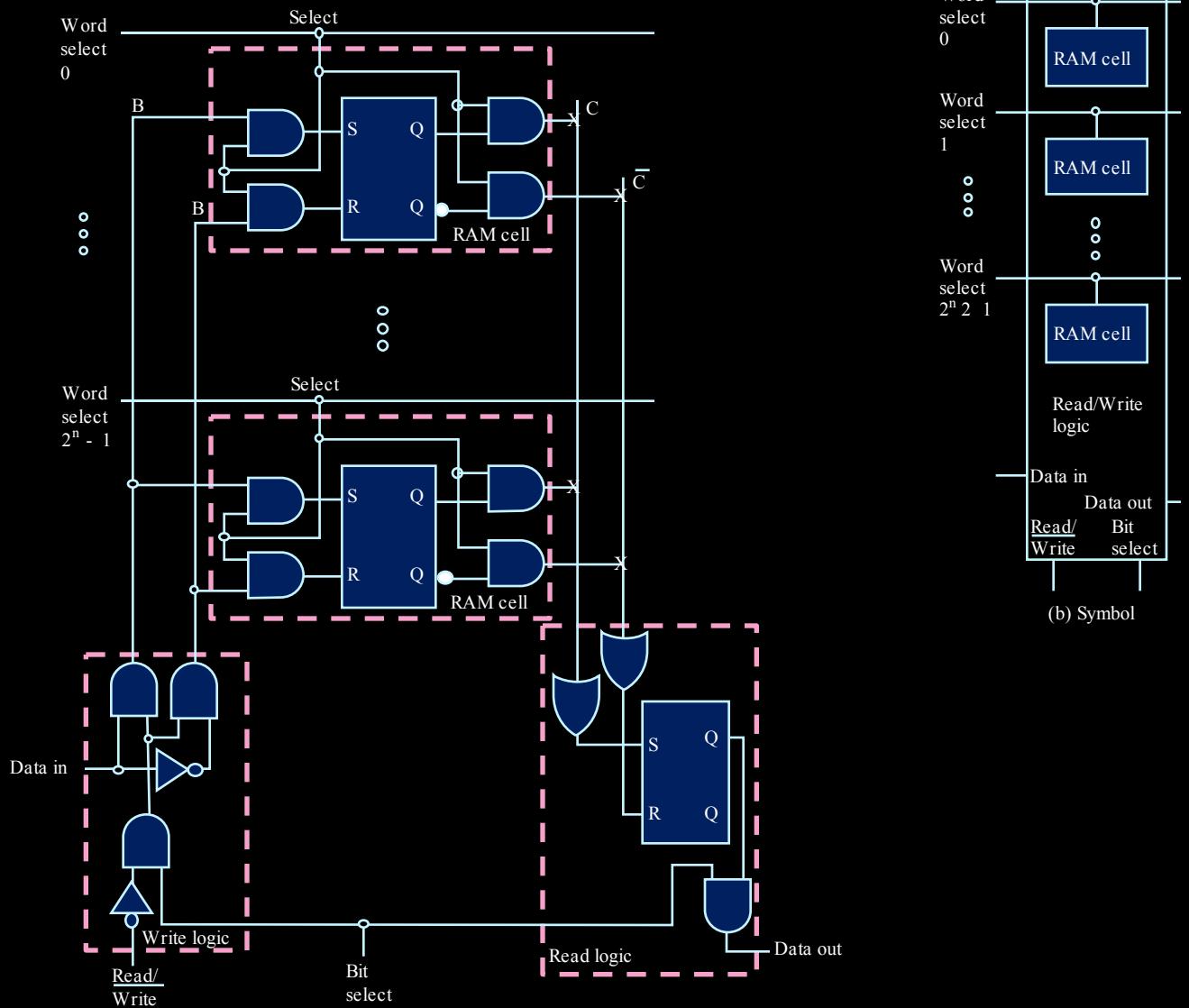
- For storing a single bit
- Each row is made of an array of storage cells.
- Multiple ways of representing these cells.
  - e.g. RAM cell (basically a D-latch):

DRAM IC cell:



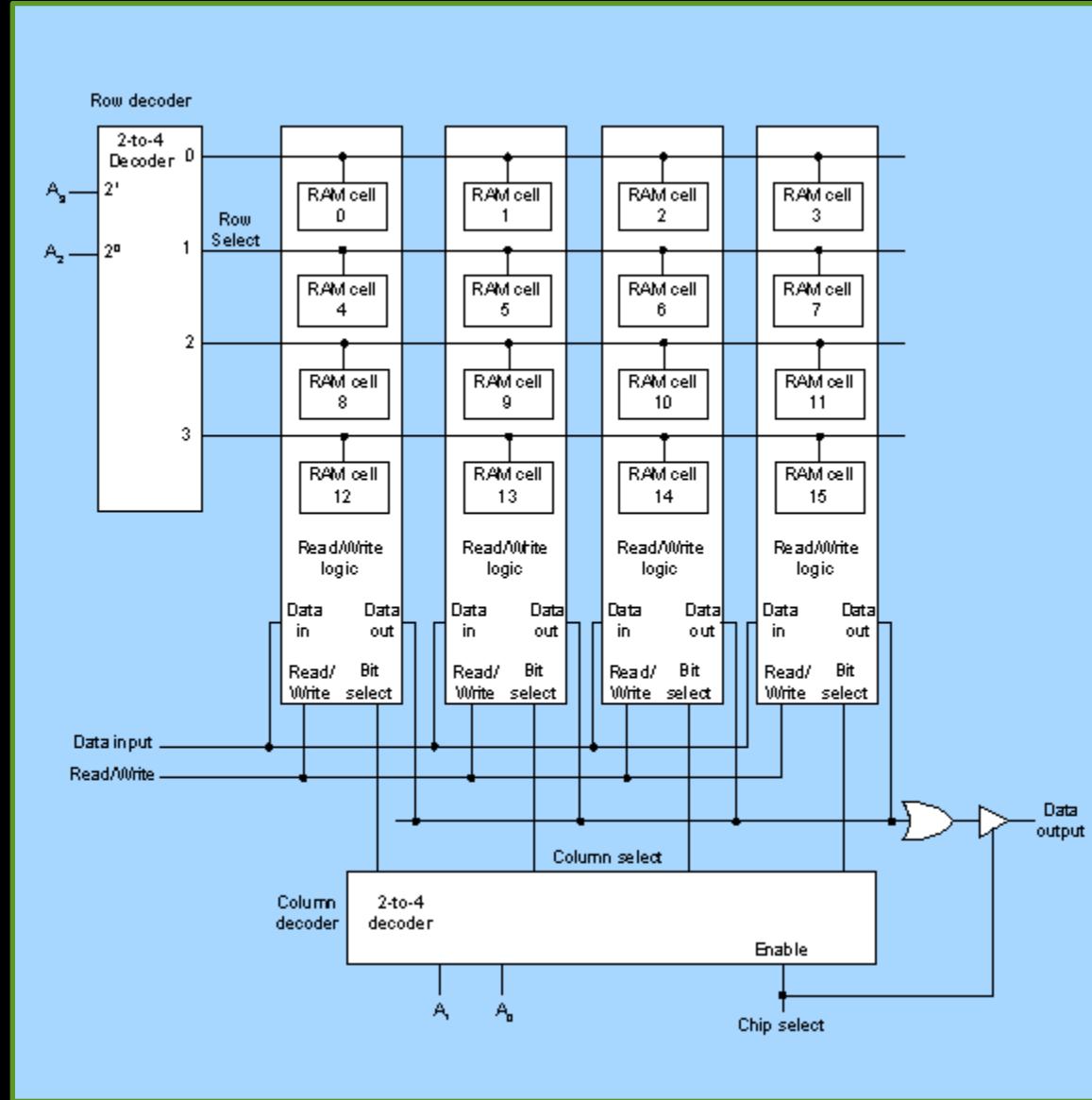
# RAM slice model

- Word select signals (via a one-hot decoder) determine which row to send out on the C lines.



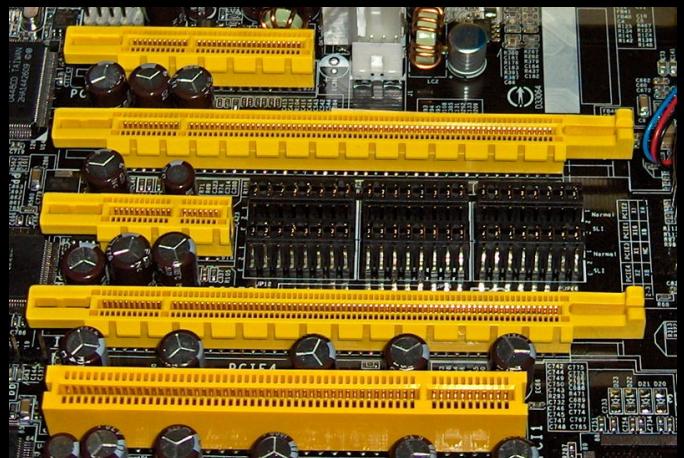
# RAM slice model

- Or, use word select to choose row, and use bit-select to choose column.



# Data Bus

- Communication between components takes place through groups of wires called a **bus** (or **data bus**).
  - Multiple components can read from a bus, but only one can write to a bus at a time.
    - Also called a **bus driver**.
  - Each component has a tri-state buffer that feeds into the bus. When not reading or writing, the tri-state buffer drives high impedance onto the bus.



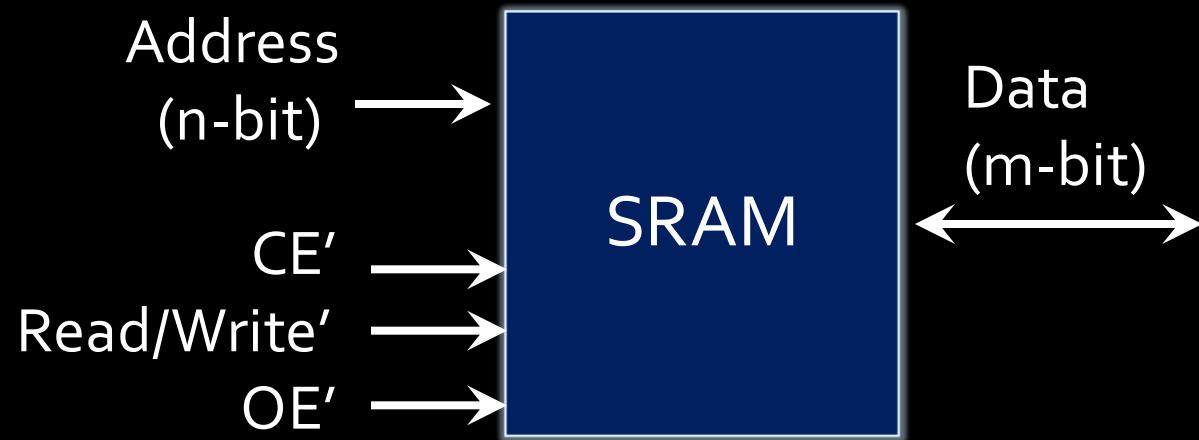
Example:

## SRAM

Static Random Access Memory

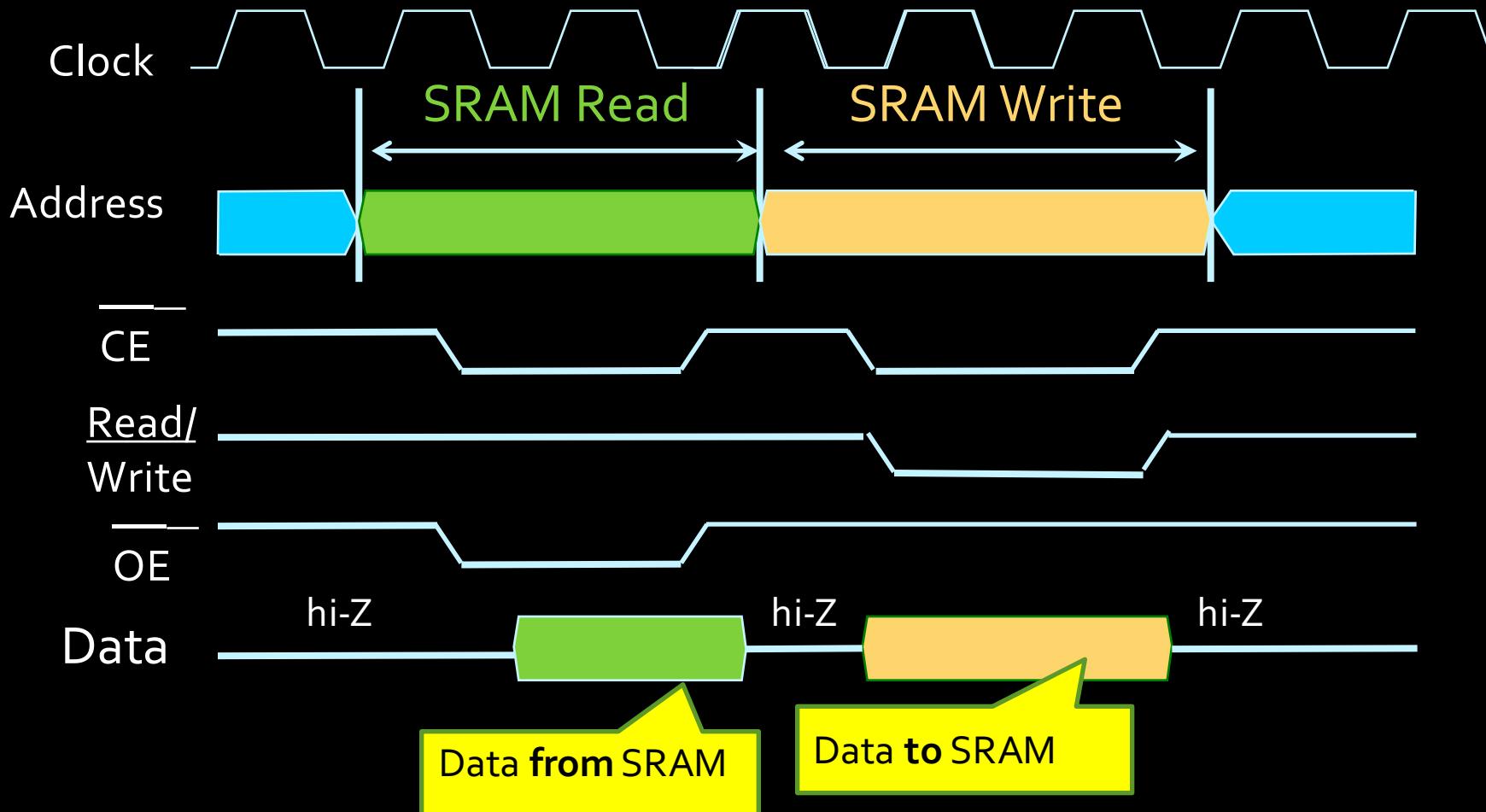
There are other types of RAMs such as DRAM, SDRAM, DDR SDRSM, RDRAM, VRAM, etc.

# Asynchronous SRAM Interface - An example



Chip Enable' (CE')	Read/Write'	Output Enable' (OE')	Access Type
0	0	X	SRAM Write
0	1	0	SRAM Read
1	X	X	SRAM not enabled

# Read/Write SRAM - Timing waveforms



- Reading and writing of signals takes time.

# Memory vs registers

- Memory houses most of the data values being used by a program.
- Registers are more local data stores, meant to be used to execute an instruction.
  - Registers are not meant to host memory between instructions (like scrap paper for a calculation).
  - Exception is the stack pointer register, which is sometimes in the same register file as the others.

# Next...

## The Controller Thing

