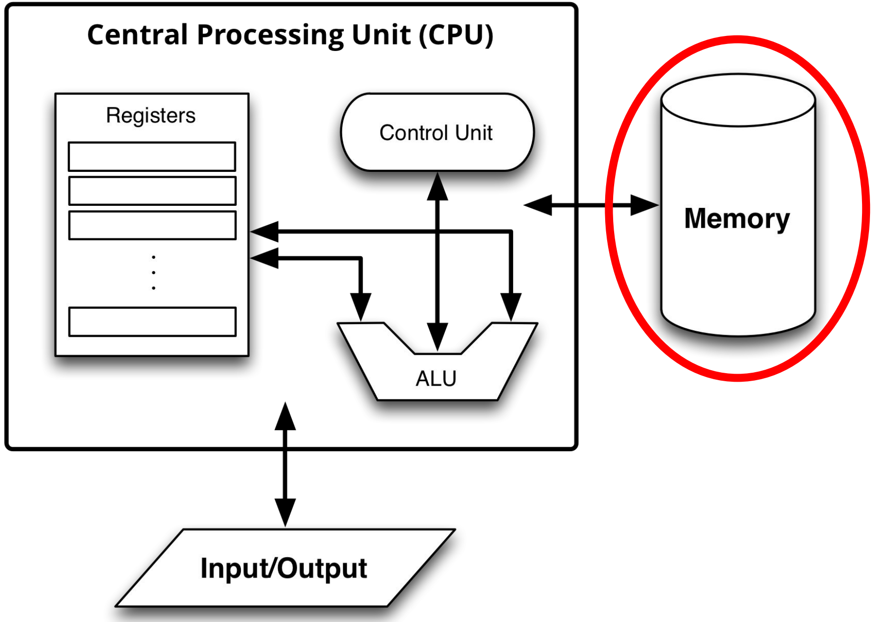
Memory

Addressing, Storing, Loading

Recap: The Von Neumann Architecture

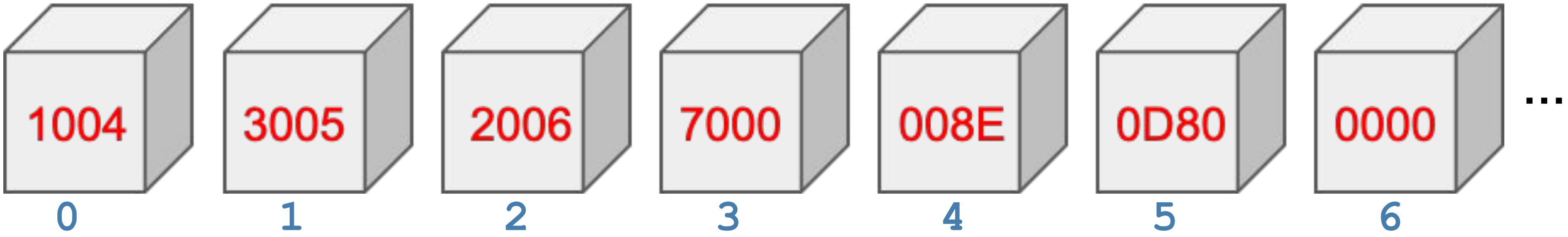
Memory

- Stores both **data** *and* program **code**
- Connected to CPU via a **bus**
- CPU can read and write words from/to memory



Recap: Memory model

Memory is **external to the CPU**. Think of it as a sequence of boxes:



Each box contains a **value** (here: a 16-bit number).

This could be a machine code instruction, or data.

We give each box an **address**: the number of the box, starting from 0.

What do we store in memory?

Address	Hex Value	Integer	Bit pattern	Instruction
000	1004	4100	0001000000010100	Load 004
001	3005	12293	0011000000000101	Add 005
002	2006	8198	0010000000000110	Store 006
003	7000	28672	0111000000000000	Halt
004	008E	142	0000000010001110	JnS 08E
005	0D80	3456	0000110110000000	JnS D80
006	0000	0	0000000000000000	JnS 000

What do we store in memory?

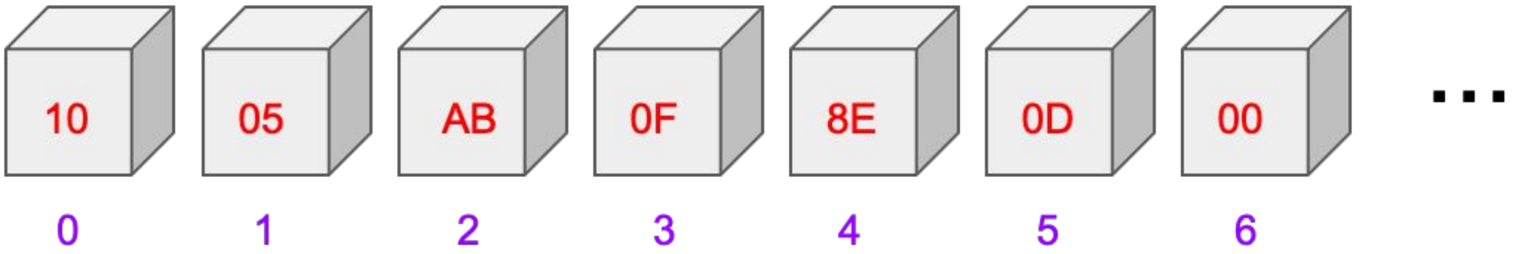
Address	Hex Value	Integer	Bit pattern	Instruction
000	1004	4100	0001000000010100	Load 004
001	3005	12293	0011000000000101	Add 005
002	200[illegible]	[illegible]	[illegible]	Store 006
003	70[illegible]	[illegible]	[illegible]	Halt
004	00[illegible]	[illegible]	[illegible]	JnS 08E
005	0D[illegible]	[illegible]	[illegible]	JnS D80
006	00[illegible]	[illegible]	[illegible]	JnS 000

The memory doesn't know!
The CPU doesn't know!

The program **interprets** the memory in a certain way.

Addressing

Most architectures store **one byte per memory location**:

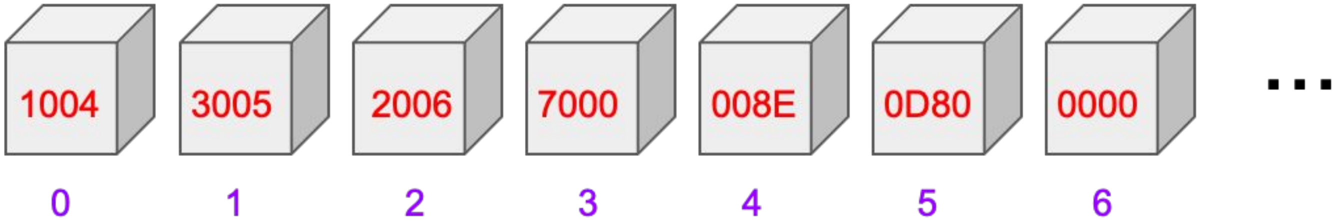


Each byte has its own address.

These architectures are called **byte-addressable**.

Addressing in MARIE

Some architectures (including MARIE) store **one word per location**:



Each word has its own address.

This is called **word-addressable**.

Remember: In MARIE, one word is 16 bits.

How much memory can we address?

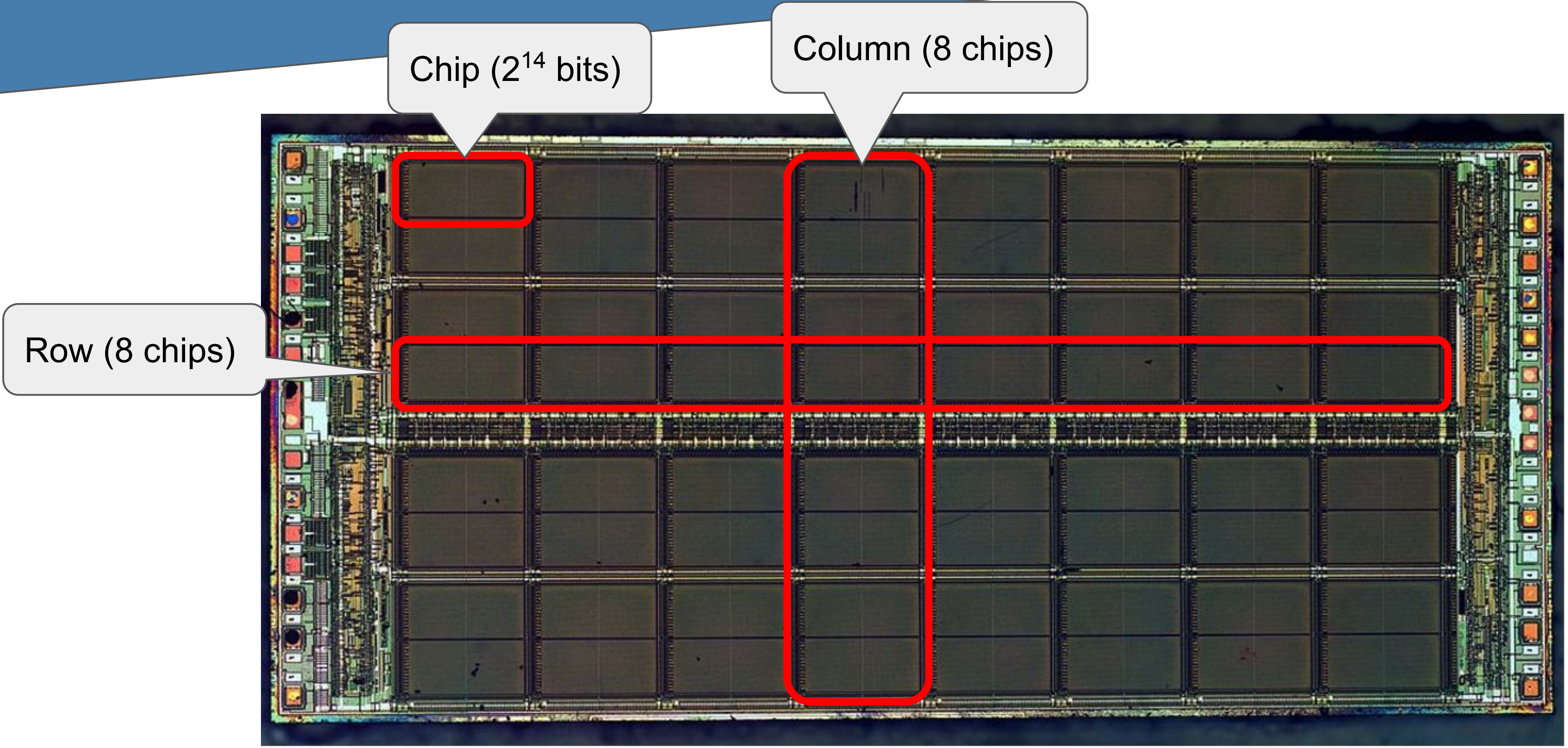
Easy: *n* bits can represent 2^n different addresses!

- MARIE addresses are 12 bits long
- That's 2^{12} different addresses.
- MARIE is word-addressable, each address contains a 16-bit (2-byte) value.
- So MARIE can address
  2×2^{12} bytes = 8,192 bytes = 8 kibibyte of memory.

Random Access Memory (RAM)

- Computer main memory is often called Random Access Memory (RAM)
- Random Access means we can read and write data at any address, in any order
- In contrast to storage such as magnetic tapes, disks, DVDs etc.
  - disk read/write heads need to move to the right place

RAM modules



Total size: 64×2^{14} bits = 2^{20} bits = 1 Mbit

Source: Wikipedia.

RAM module addressing

8 rows

2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8

8 columns = 8×2^{11} bytes per row

$8 \times 8 \times 2^{11} =$
2^{17} locations

We need **17 bits** to address each byte in this RAM module!

RAM module addressing

Example address (17 bit):

010110000010010011

Row 2
(3 bits)

Column 6
(3 bits)

Byte 147
(11 bits)

2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8
2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8	2^{11}×8

We can implement this using MUXes!

- One 11-bit MUX per chip: select byte
- One 3-bit MUX per row: select column
- One 3-bit MUX per module: select row

Summary

- Memory holds **data** – **interpretation** (Number? Text? Image? Code?) is up to the program
- Byte-addressable: one address per byte
- Word-addressable: one address per word
- **RAM** (Random Access Memory) is the main memory in a computer

EOF