01.04.20 11:41

CSCI 150
Introduction to Digital and Computer System Design
Lecture 6: Memory I

Jetic Gū

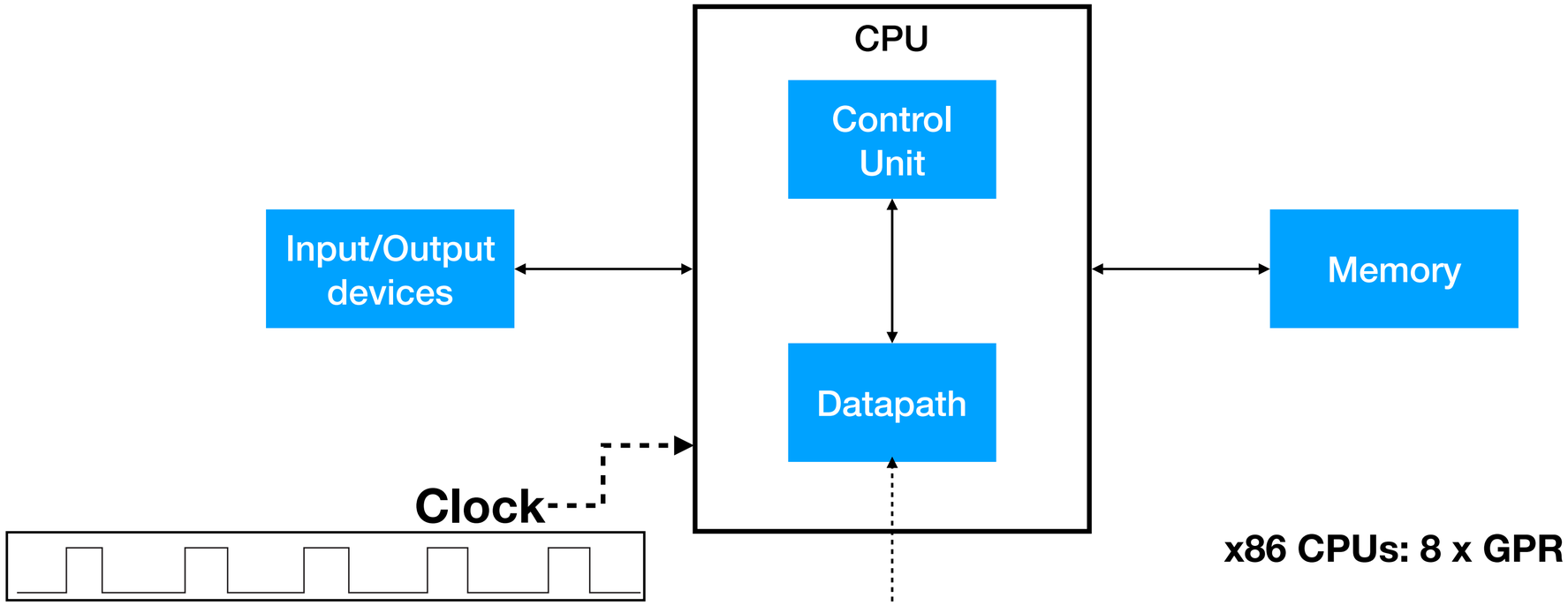
2020 Winter Semester (S1)

Overview

- Focus: Fundamentals of Complex Digital Circuit Design
- Architecture: von Neumann
- Textbook v4: Ch8 8.1, 8.2, 8.3; v5: Ch7 7.1, 7.2, 7.3
- Core Ideas:
    1. Memory Definition
    2. Read Only Memory
    3. Random Access Memory

P0
Review

Computer



x86 CPUs: 8 x GPR

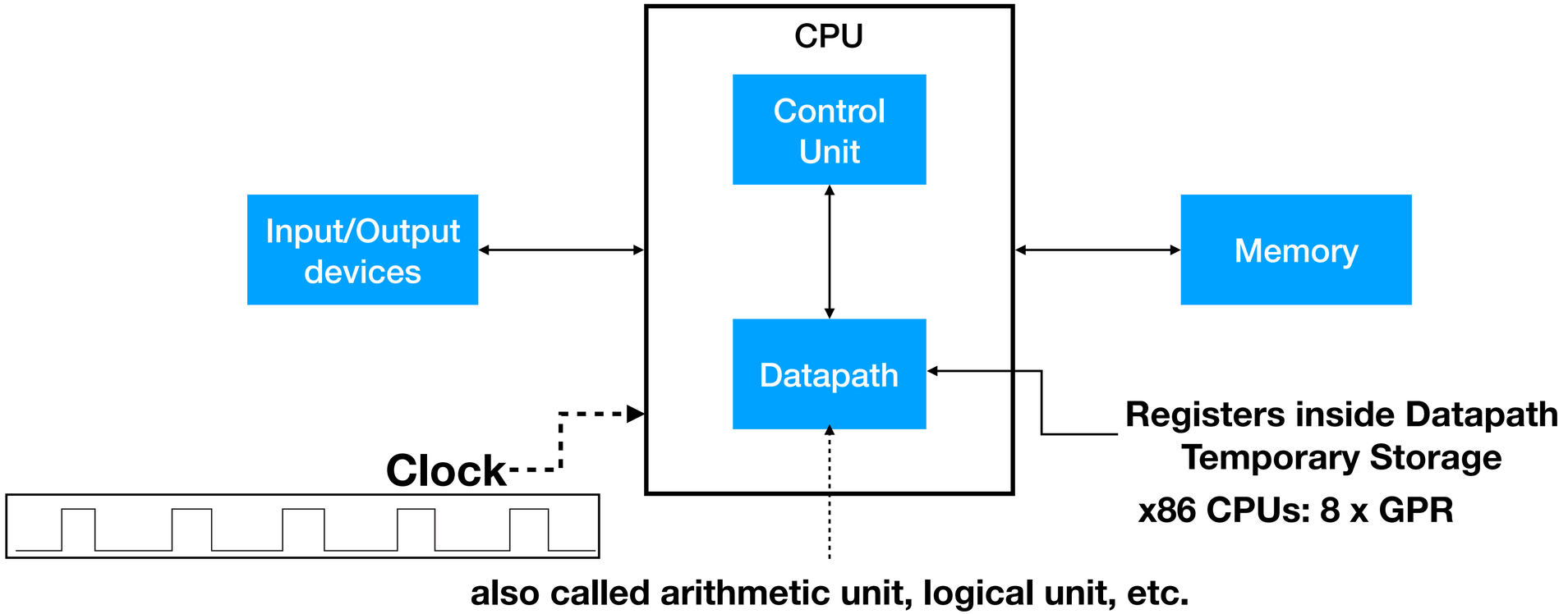
also called arithmetic unit, logical unit, etc.

1. Von Neumann Architecture

Review

P0
Review

Computer

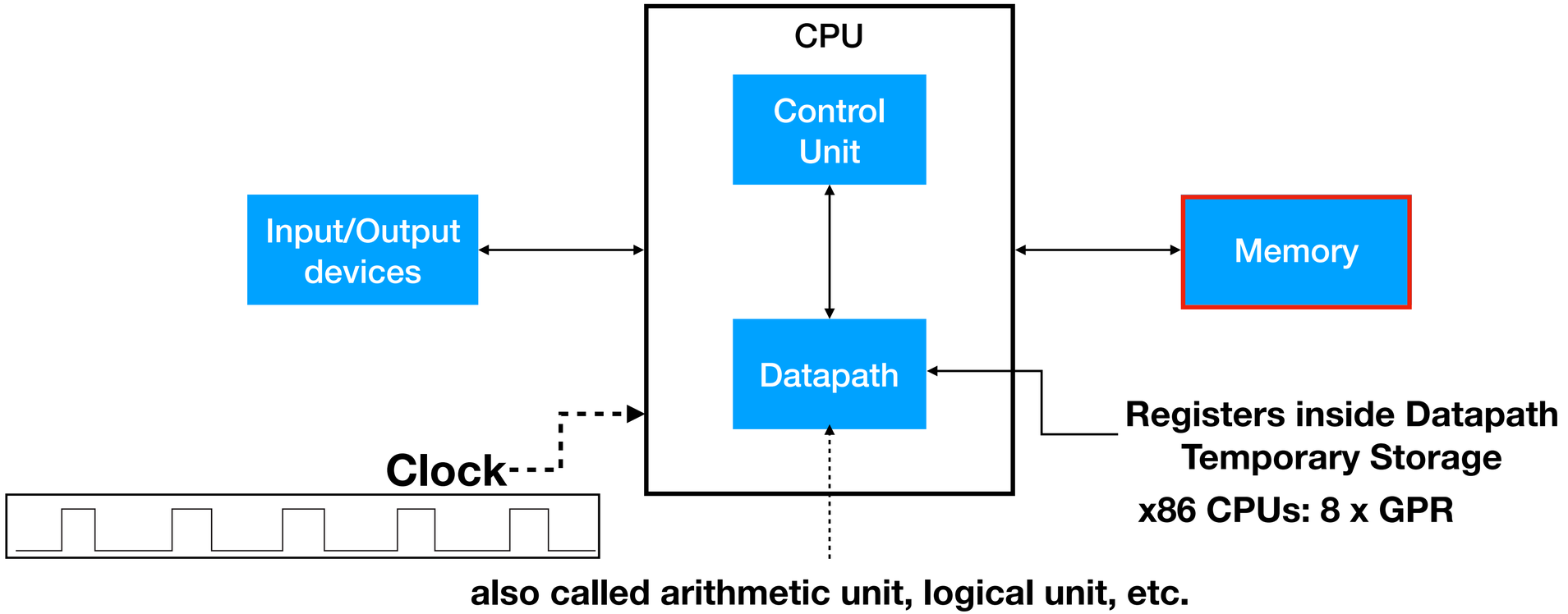


1. Von Neumann Architecture

Review

P0
Review

Computer

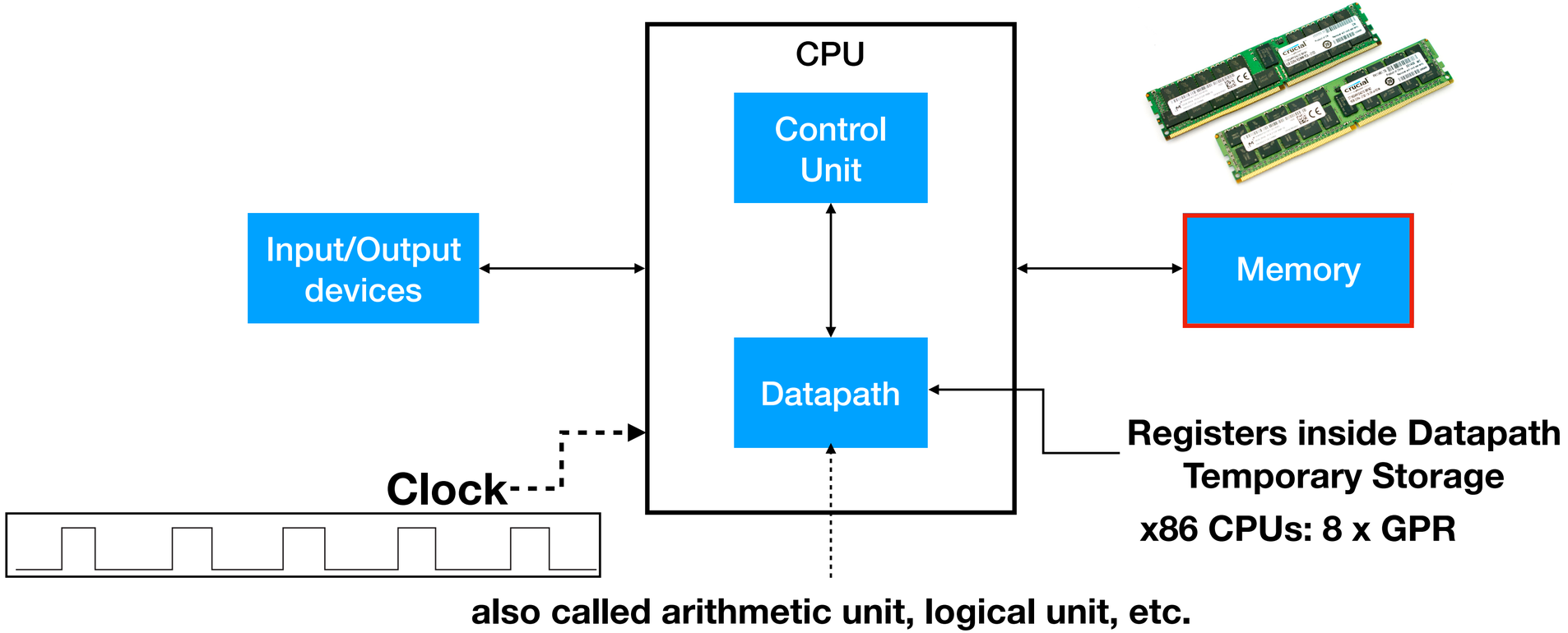


1. Von Neumann Architecture

Review

P0
Review

Computer



1. Von Neumann Architecture

Review

P1
Definition

Memory Definition

Wait… do I have to remember that?

Summary

P1
Definition

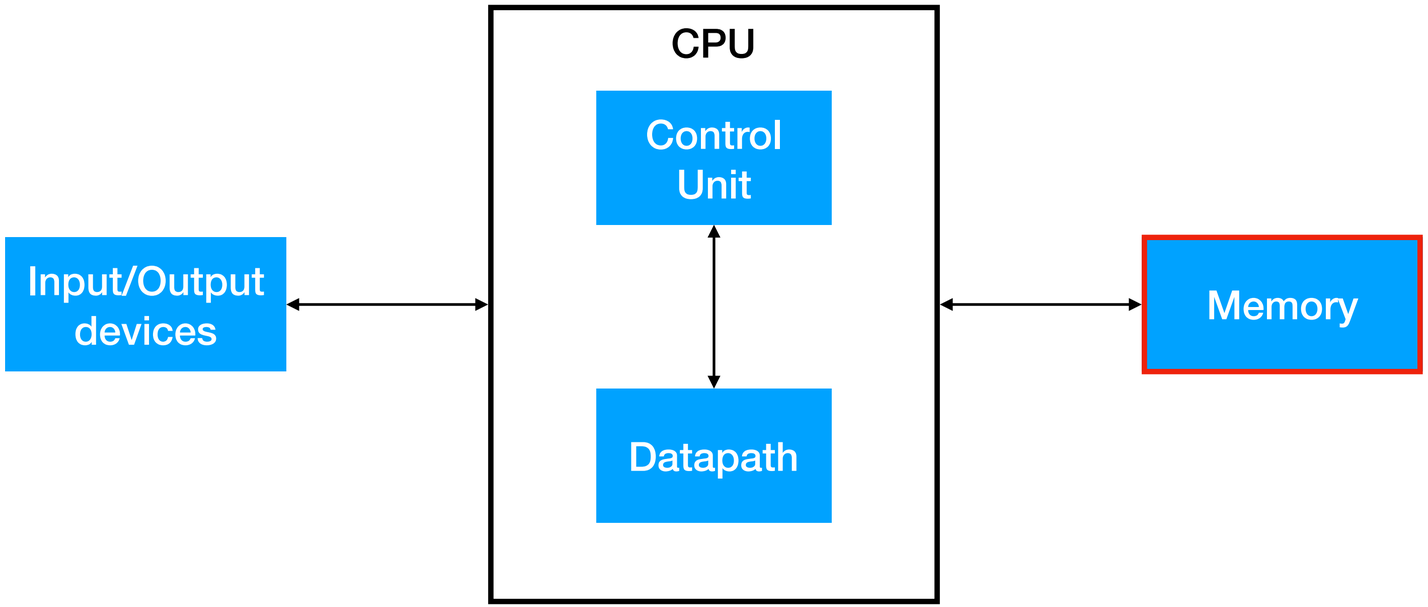
Memory Definition

- A **collection of cells** capable of storing binary information
- providing **temporary** or **permanent** storage for **substantial amounts** of binary information
  - **Substantial**: much much much more than registers in a CPU
  - Temporary: e.g. DDR4 memory sticks
  - Permanent: e.g. IO devices (SSD, HDD), ROM

Concept

P1
Definition

Computers In Theory



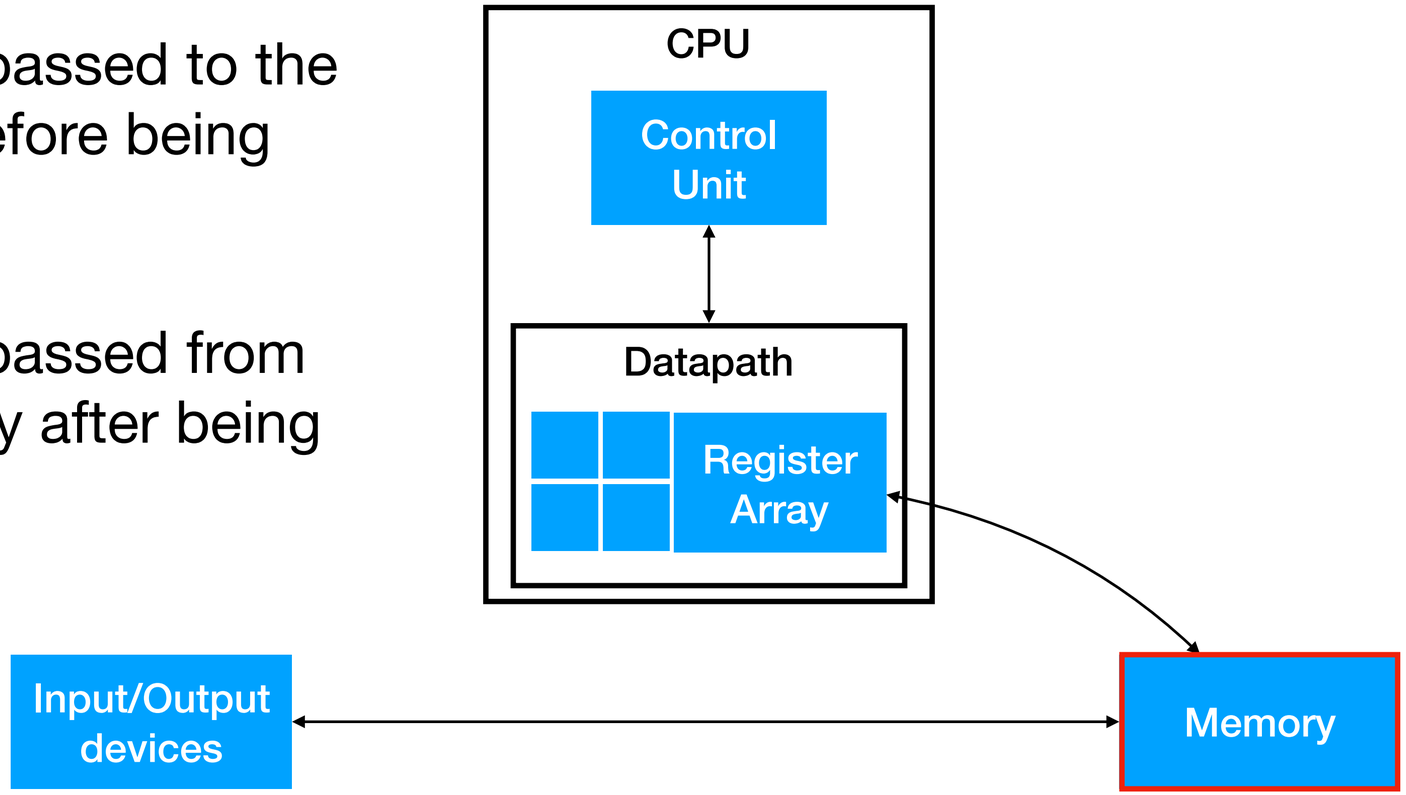
Concept

1. Von Neumann Architecture

P1
Definition

Computers In Reality

- IO data is passed to the memory before being processed
- IO data is passed from the memory after being processed

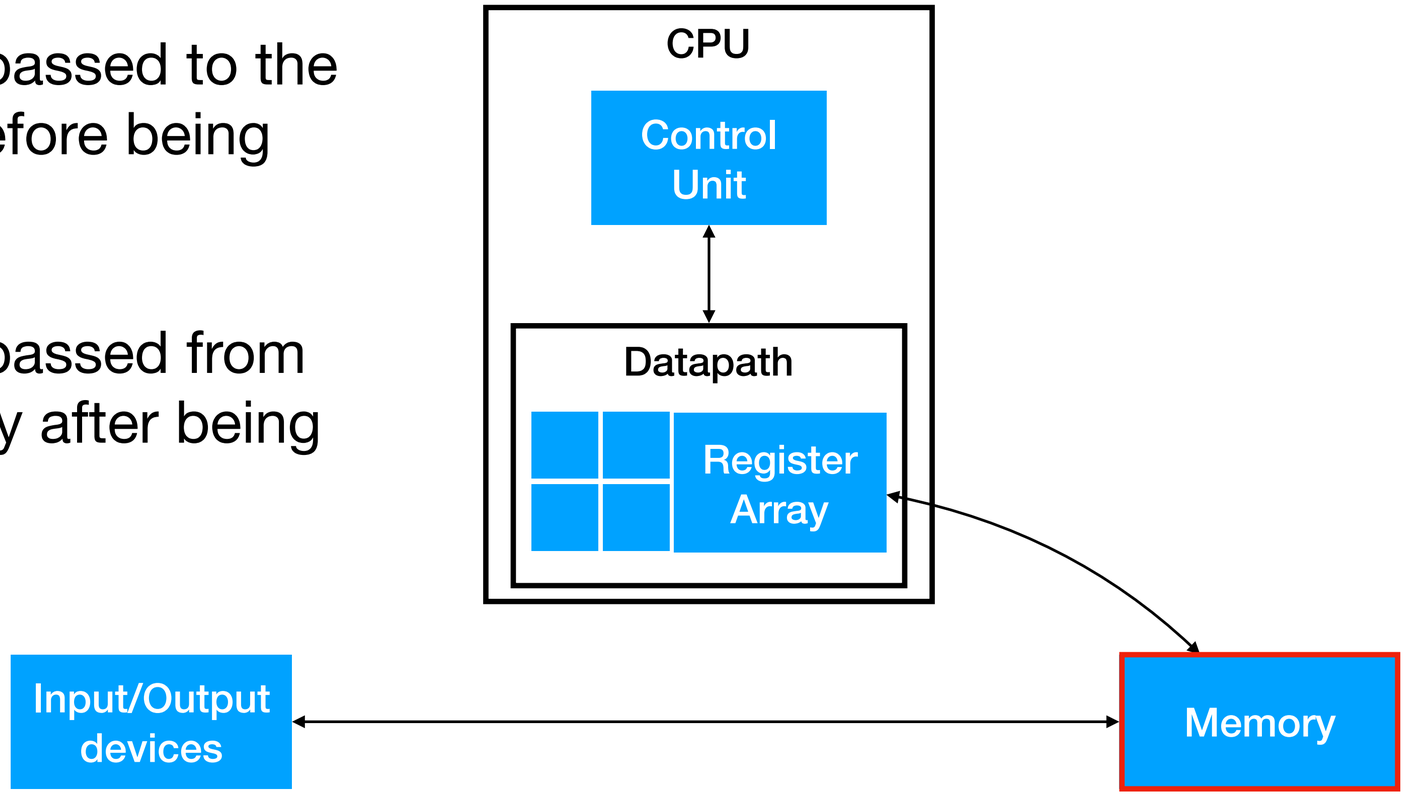


Concept

P1
Definition

Computers In Reality

- IO data is passed to the memory before being processed
- IO data is passed from the memory after being processed

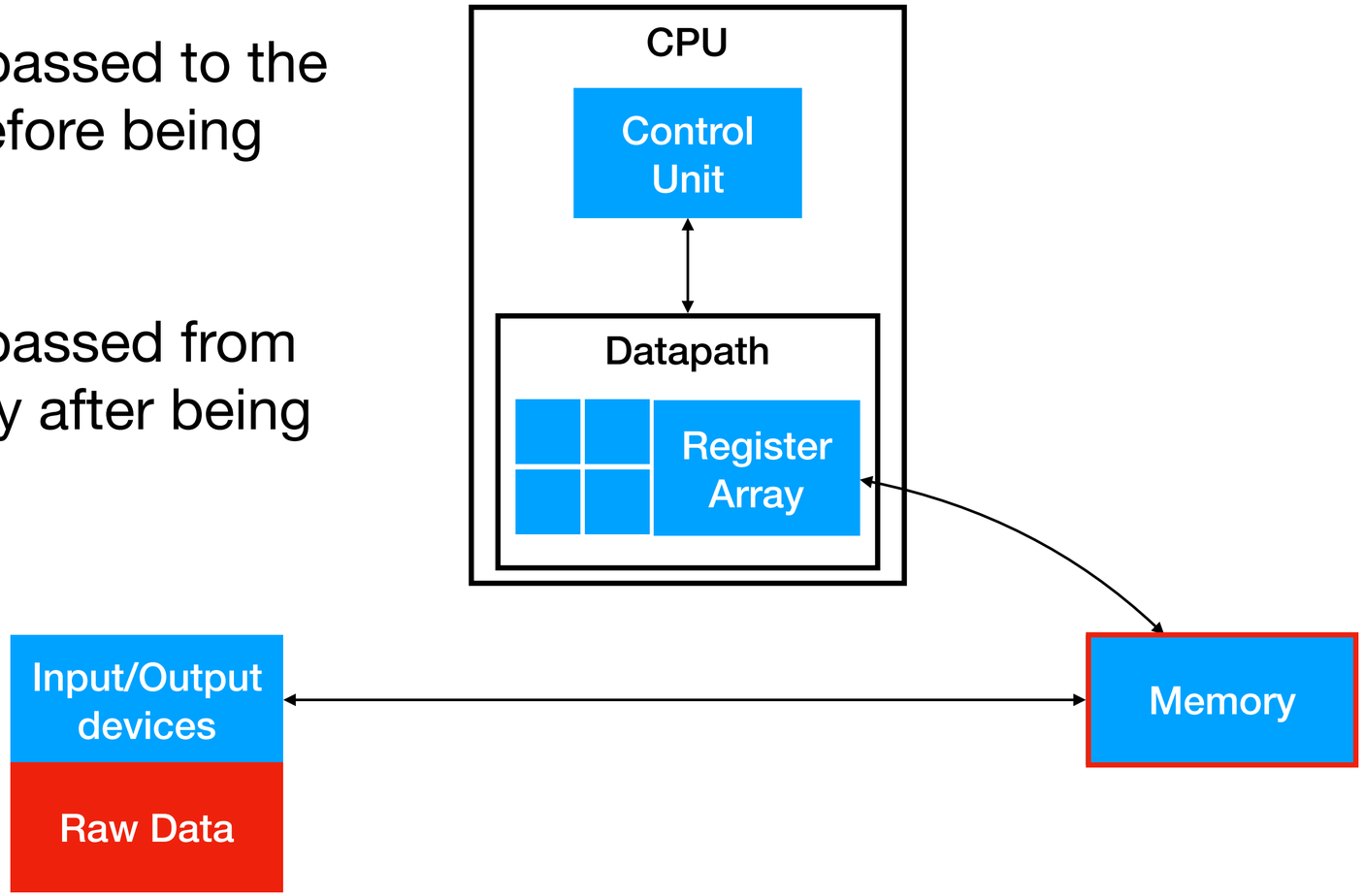


Concept

P1
Definition

Computers In Reality

- IO data is passed to the memory before being processed
- IO data is passed from the memory after being processed

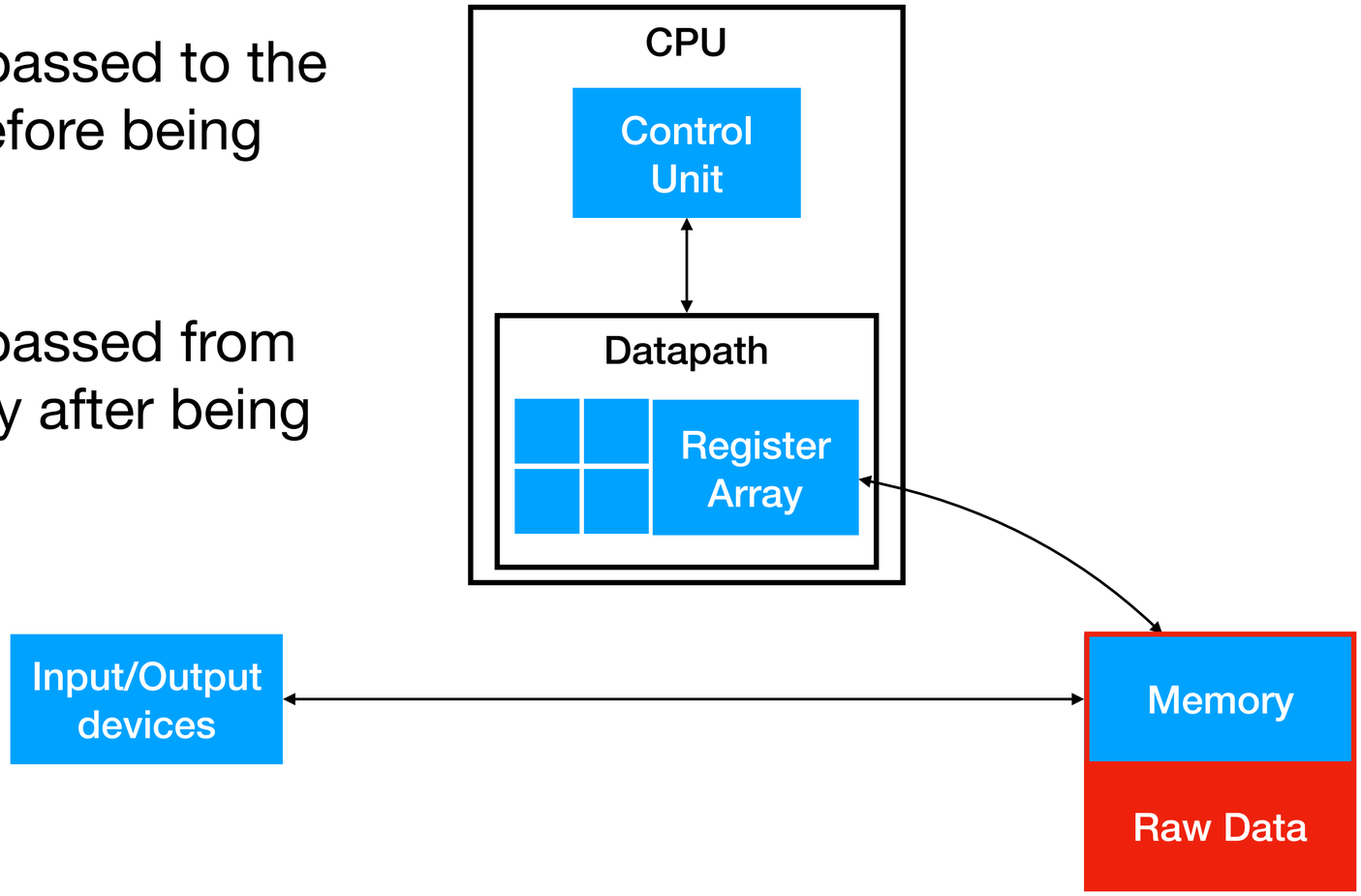


Concept

P1
Definition

Computers In Reality

- IO data is passed to the memory before being processed
- IO data is passed from the memory after being processed

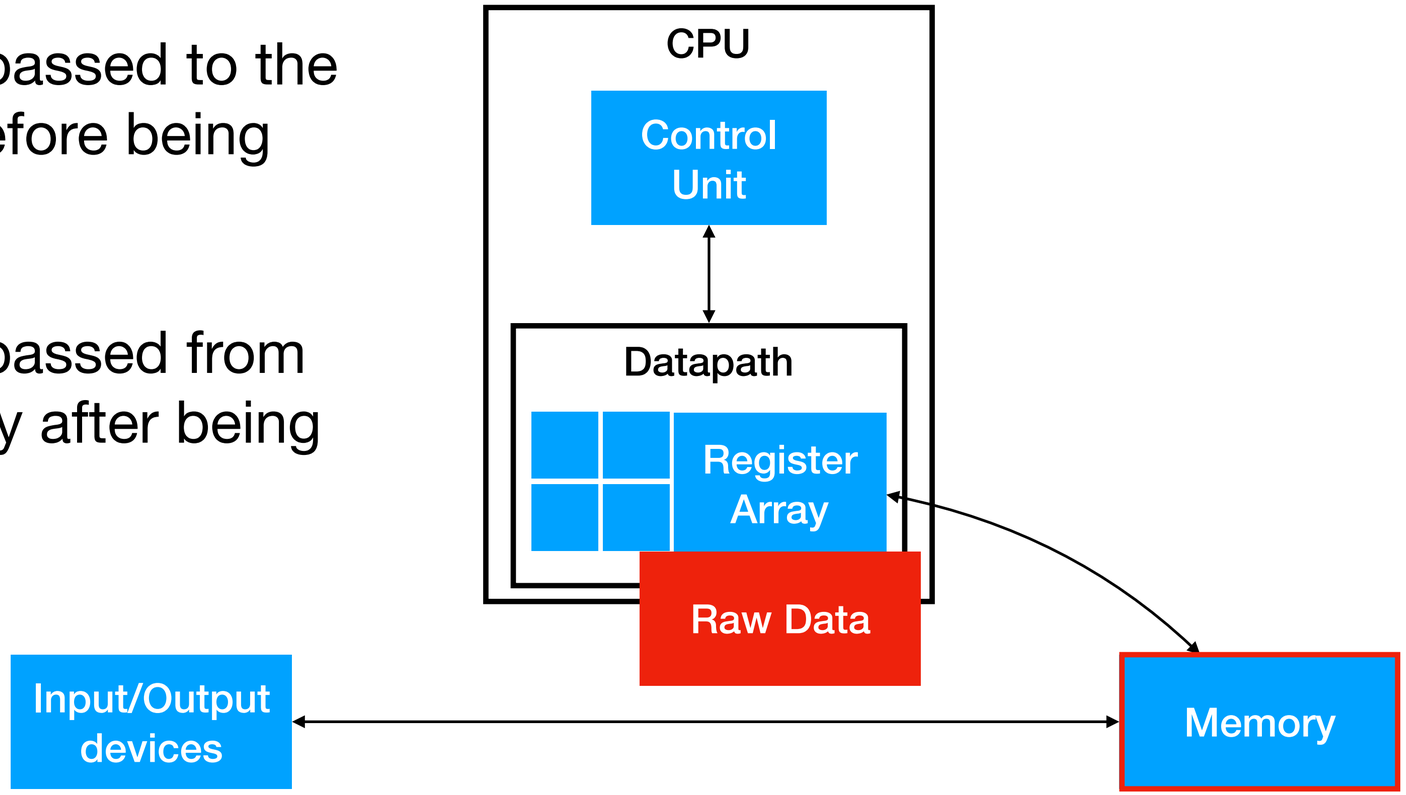


Concept

P1
Definition

Computers In Reality

- IO data is passed to the memory before being processed
- IO data is passed from the memory after being processed

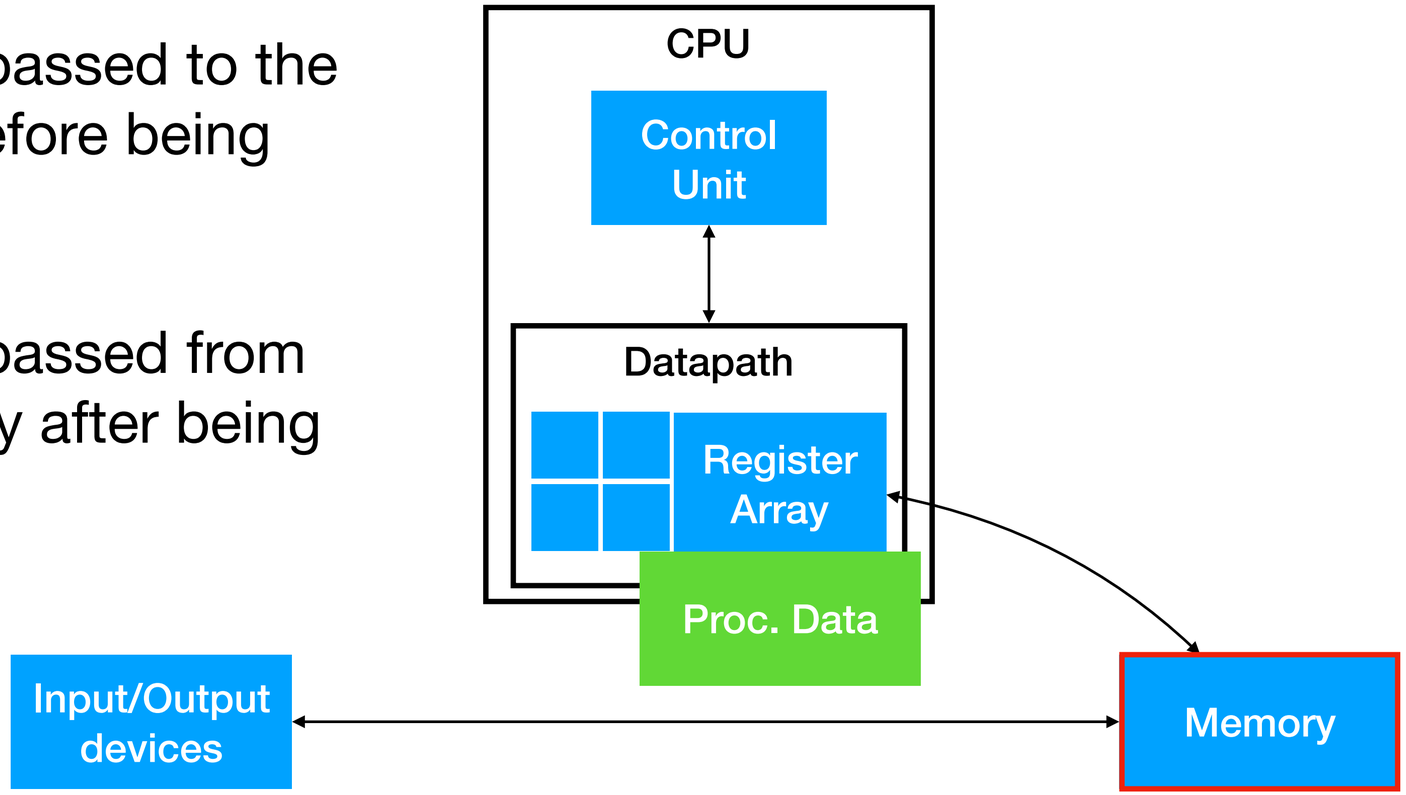


Concept

P1
Definition

Computers In Reality

- IO data is passed to the memory before being processed
- IO data is passed from the memory after being processed

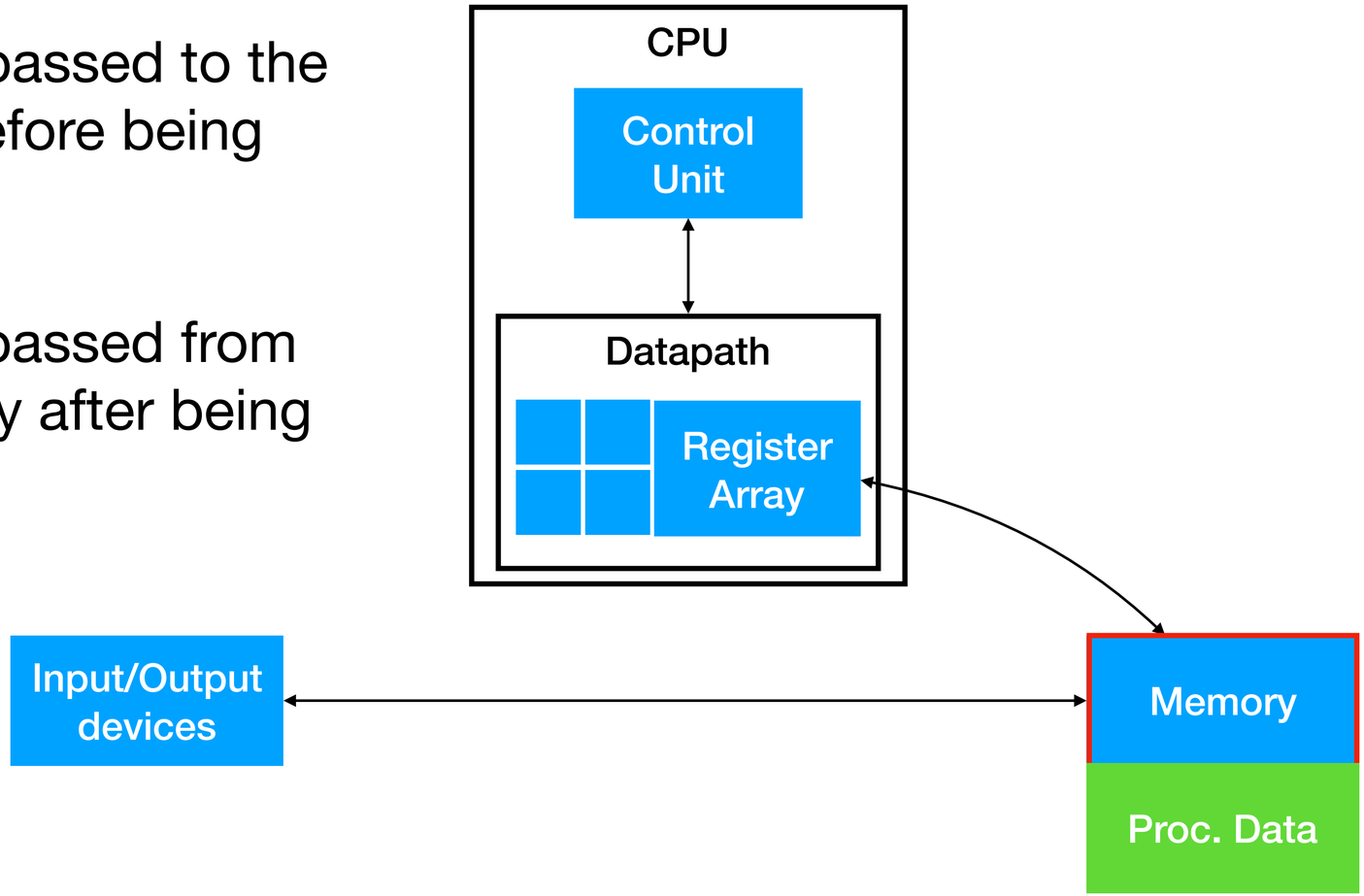


Concept

P1
Definition

Computers In Reality

- IO data is passed to the memory before being processed
- IO data is passed from the memory after being processed

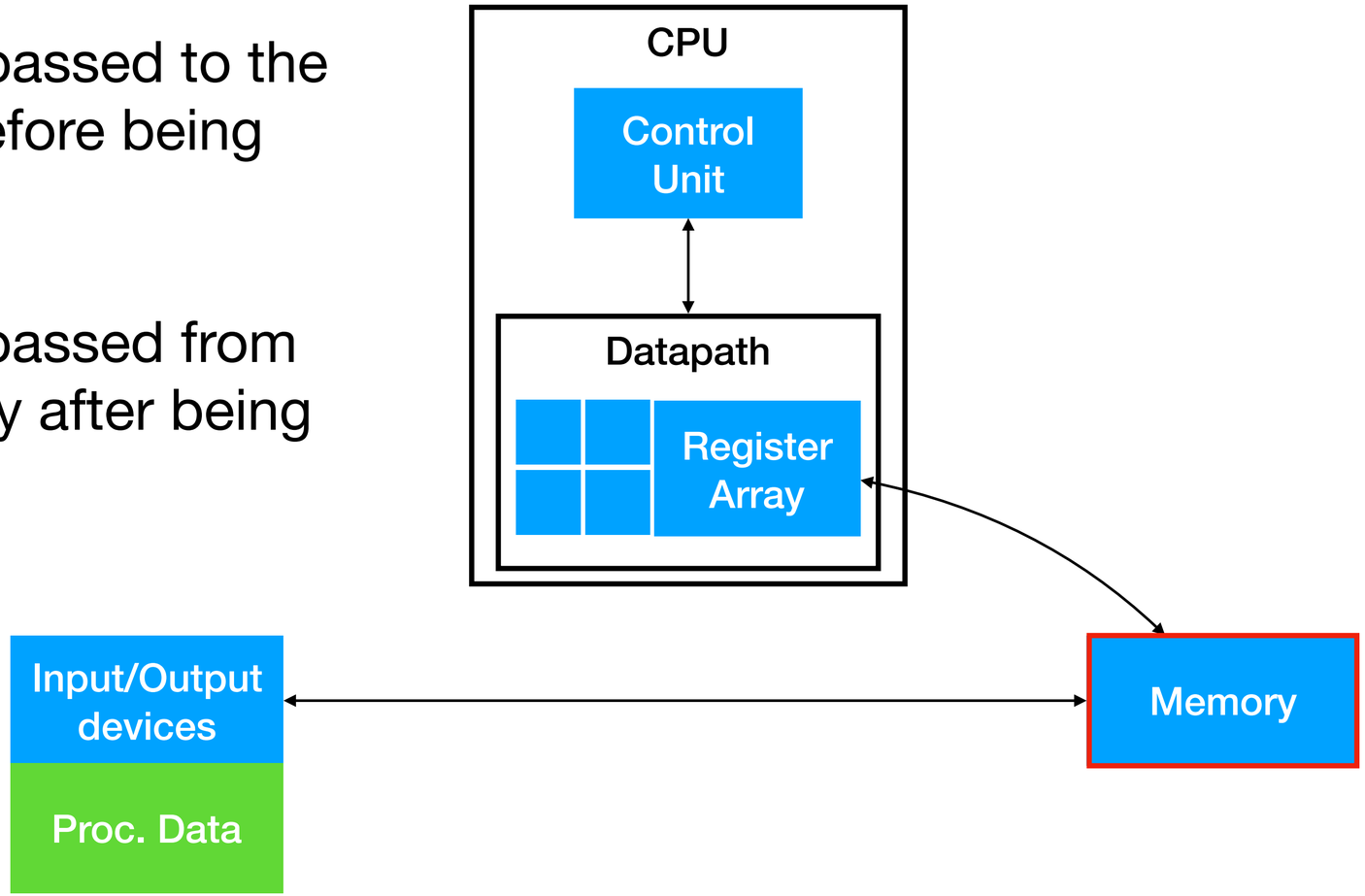


Concept

P1
Definition

Computers In Reality

- IO data is passed to the memory before being processed
- IO data is passed from the memory after being processed



Concept

P1
Definition

2 Types of Memories

- RAM (Random Access Memory)
  Memory Read Operation: retrieves information from memory
  Memory Write Operation: accepts new information for storage
  - Volatile: require power for keeping its content 
- ROM (Read Only Memory)
  Memory Read Operation: retrieves information from memory
  - Non-volatile: might not need constant power to keep its content 
- Why do we need ROMs?

Concept

P2
ROM

Read Only Memory

Remember CD-ROMs?

Summary

The Many Types of ROMs

- CD-ROM
- DVD-ROM
- Firmware ROM

- Essentially, devices that stores information that cannot be normally erased/ rewritten, and usually are non-volatile

Concept

ROM Abstraction

- Storage: $2^n \times m$, 2^n words, each m-bits
  (usually $m = 8$, which makes a word=a byte)
  - 32-bit processor, $n = 32$, supports 2^{32} bytes in memory ~ 4GB
- Input: address in n-bit binary
- Output: m-bits of stored information

Concept

ROM Abstraction

- 8-bit processor
  Each address is in 8-bit, total of 256 different addresses

Example

ROM Abstraction

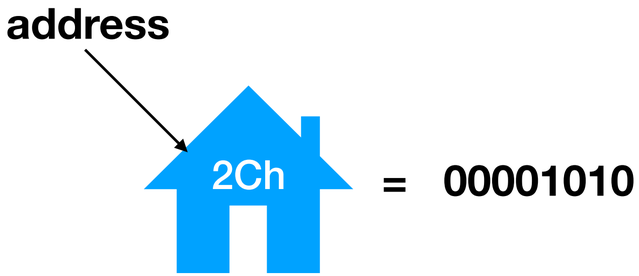
- 8-bit processor
  Each address is in 8-bit, total of 256 different addresses

= **00001010**

Example

ROM Abstraction

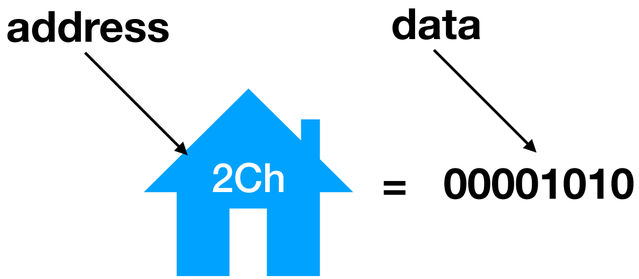
- 8-bit processor
  Each address is in 8-bit, total of 256 different addresses



Example

ROM Abstraction

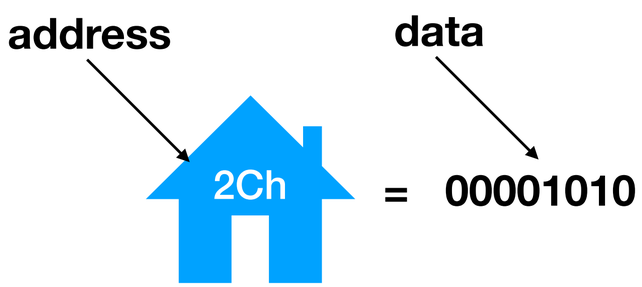
- 8-bit processor
  Each address is in 8-bit, total of 256 different addresses

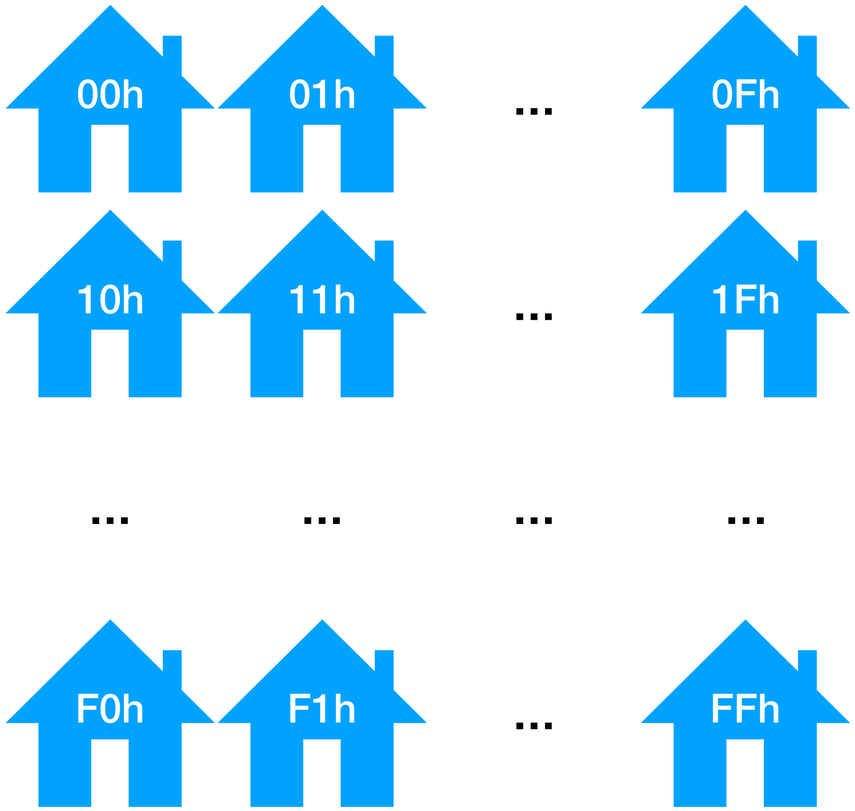


Example

ROM Abstraction

- 8-bit processor
  Each address is in 8-bit, total of 256 different addresses





Example

P2
ROM

Question

Question

Question

- What is the maximum supported memory size for a CPU with 8-bit address space, 1 byte per word?

Question

Question

- What is the maximum supported memory size for a CPU with 8-bit address space, 1 byte per word?
  - Ans: $2^8 \times 1 = 256$

Question

Question

- What is the maximum supported memory size for a CPU with 8-bit address space, 1 byte per word?
  - Ans: $2^8 \times 1 = 256$
- What is the maximum supported memory size for a CPU with 16-bit address space, 1 byte per word?

Question

Question

- What is the maximum supported memory size for a CPU with 8-bit address space, 1 byte per word?
  - Ans: $2^8 \times 1 = 256$
- What is the maximum supported memory size for a CPU with 16-bit address space, 1 byte per word?
  - Ans: $2^{16} \times 1 = 65536$

Question

- What is the maximum supported memory size for a CPU with 8-bit address space, 1 byte per word?
  - Ans: $2^8 \times 1 = 256$
- What is the maximum supported memory size for a CPU with 16-bit address space, 1 byte per word?
  - Ans: $2^{16} \times 1 = 65536$
- What is the maximum supported memory size for a CPU with 16-bit address space, 2 bytes per word?

Question

- What is the maximum supported memory size for a CPU with 8-bit address space, 1 byte per word?
  - Ans: $2^8 \times 1 = 256$
- What is the maximum supported memory size for a CPU with 16-bit address space, 1 byte per word?
  - Ans: $2^{16} \times 1 = 65536$
- What is the maximum supported memory size for a CPU with 16-bit address space, 2 bytes per word?
  - Ans: $2^{16} \times 2 = 131072$

P2
ROM

Question

Question

Question

- 64-bit CPUs have 64-bit memory space, each word 8-bits

Question

Question

- 64-bit CPUs have 64-bit memory space, each word 8-bits
- What is the maximum supported memory size for a 64-bit CPU?

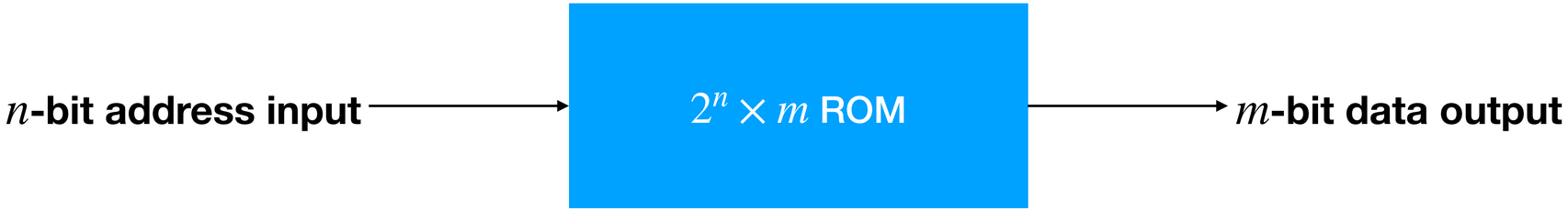
Question

Question

- 64-bit CPUs have 64-bit memory space, each word 8-bits
- What is the maximum supported memory size for a 64-bit CPU?
  - Ans: $2^{64} \times 1 = 18{,}446{,}744{,}073{,}709{,}551{,}616$ = 17 million Terabytes (16 exabytes)

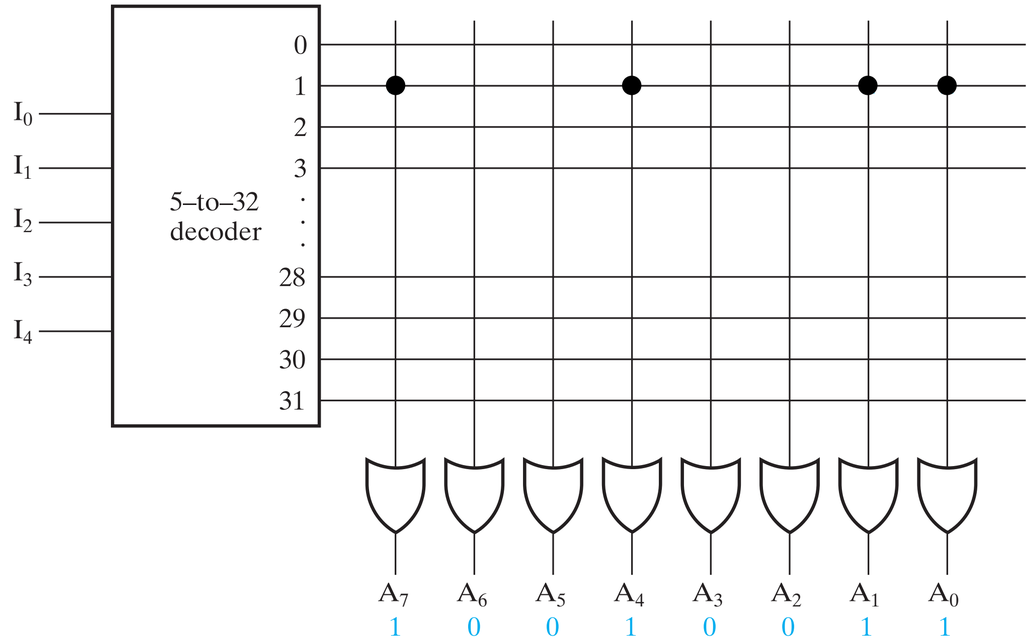
Question

ROM Abstraction

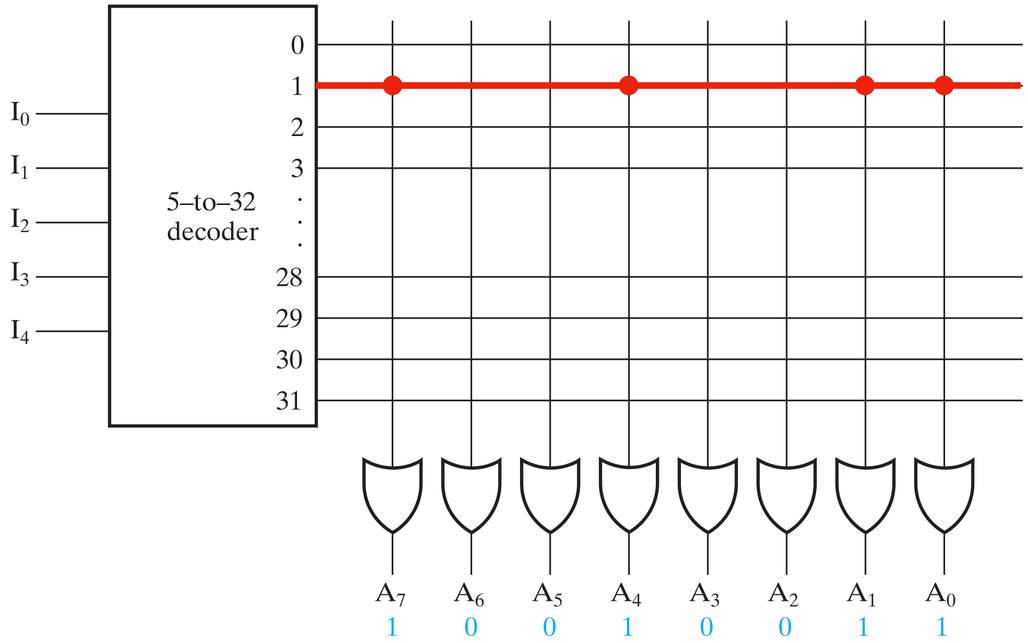


Concept

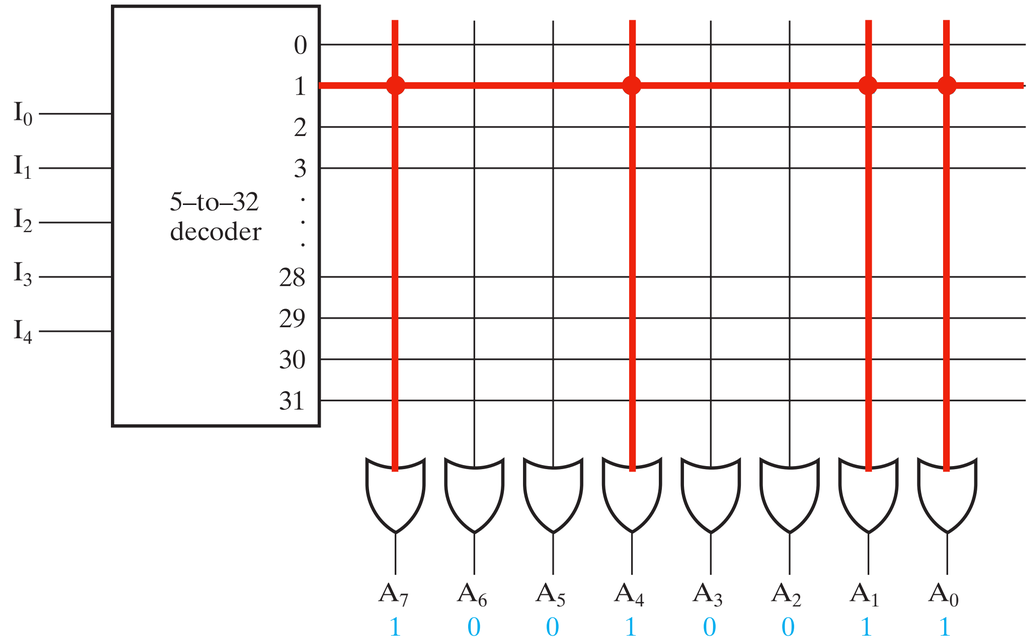
ROM Implementation Example



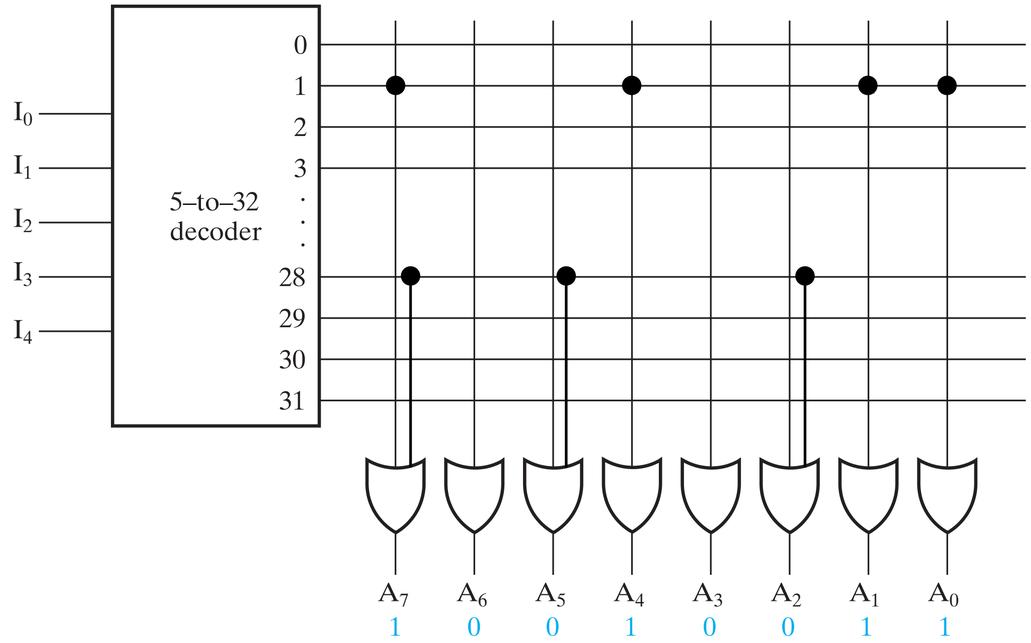
ROM Implementation Example



ROM Implementation Example

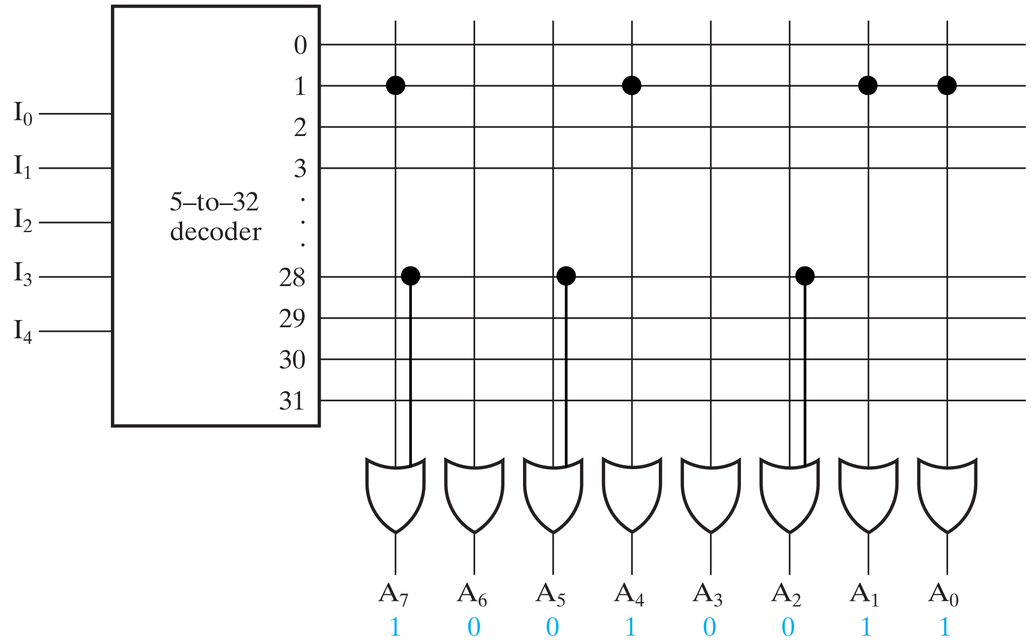


ROM Implementation Example



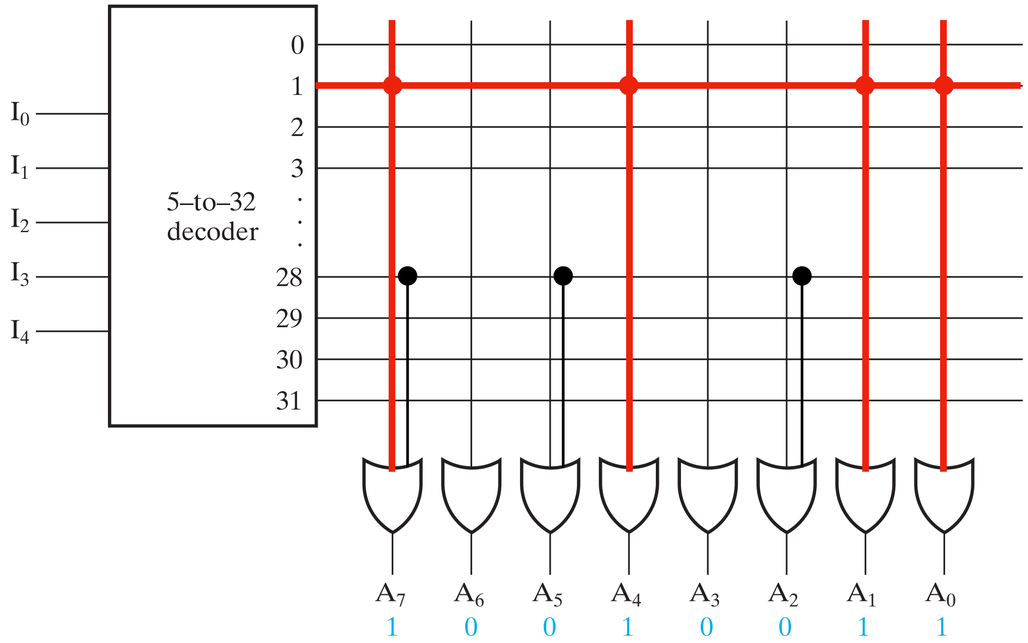
- This ROM has value `10010011` at address `01h`;

ROM Implementation Example



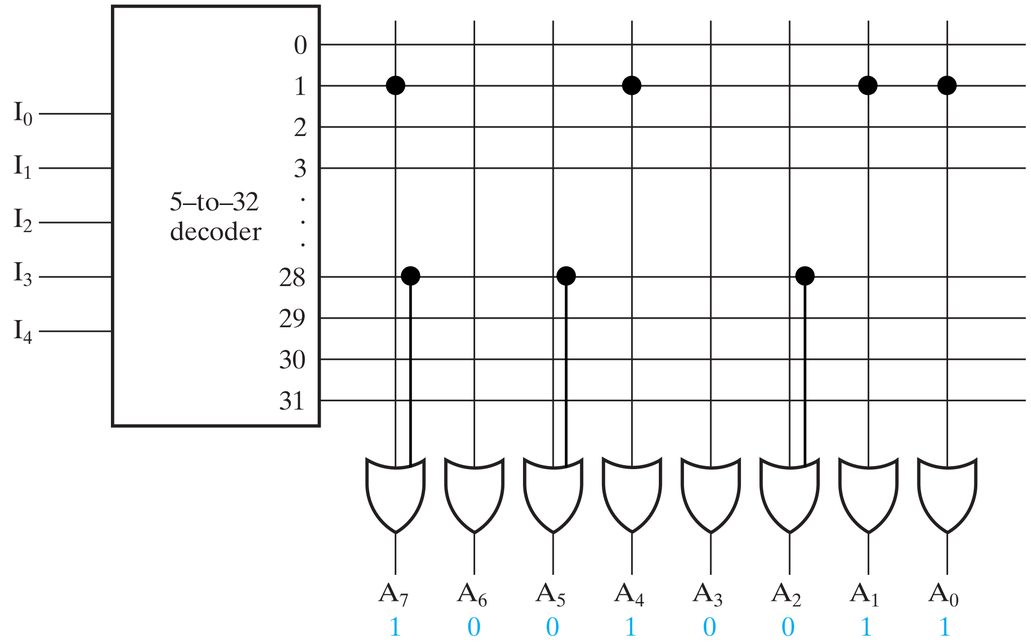
- This ROM has
  value `10010011` at address `01h`;
  value `10100100` at address `1Ch`;

ROM Implementation Example



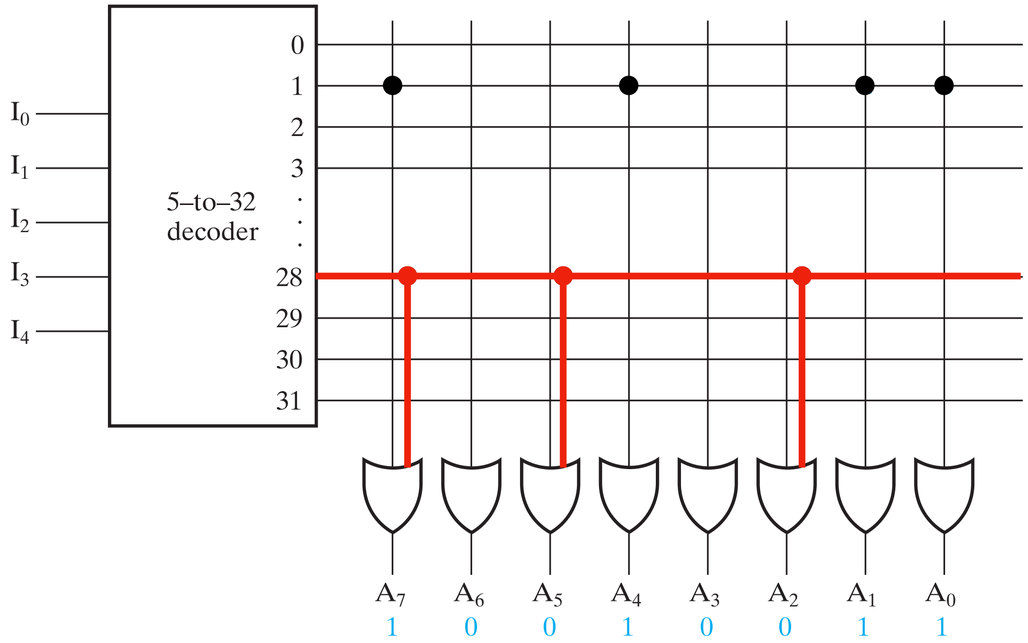
- This ROM has
  value `10010011` at address `01h`;
  value `10100100` at address `1Ch`;

ROM Implementation Example



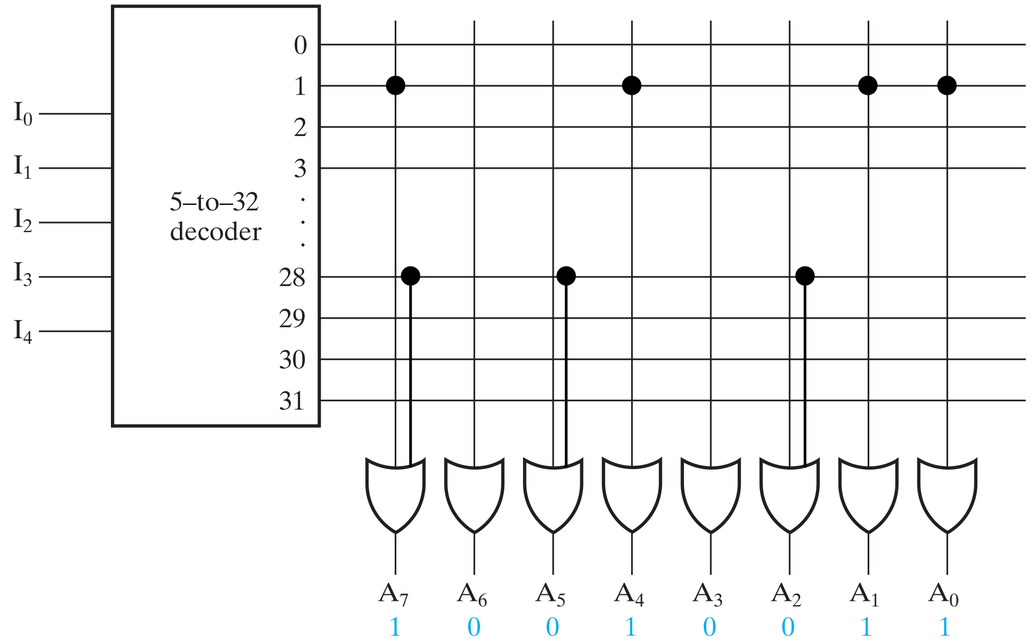
- This ROM has
  value `10010011` at address `01h`;
  value `10100100` at address `1Ch`;

ROM Implementation Example



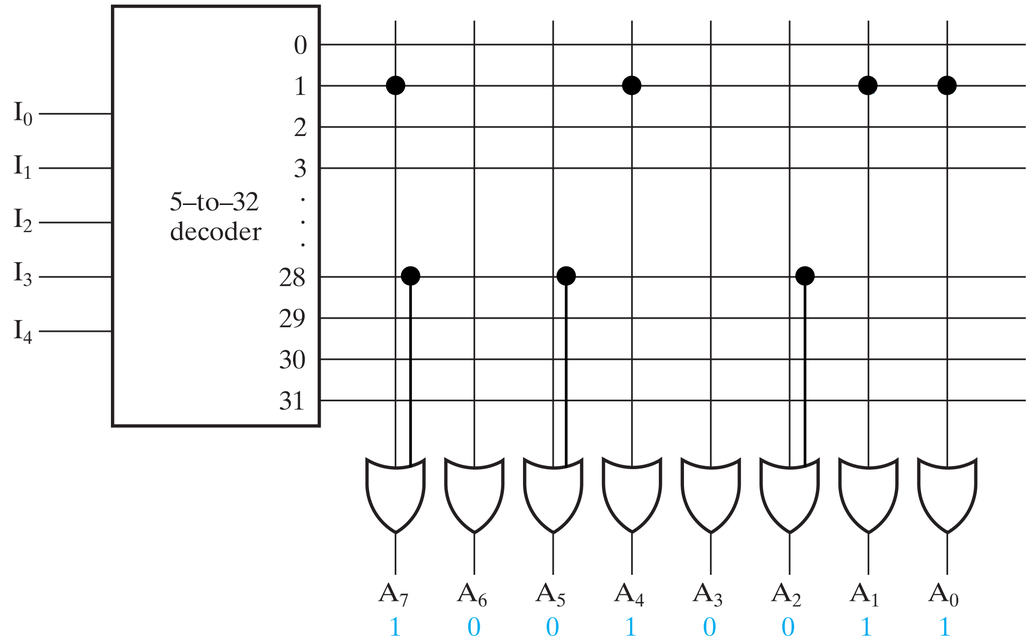
- This ROM has
  value `10010011` at address `01h`;
  value `10100100` at address `1Ch`;

ROM Implementation Example



- This ROM has
  value `10010011` at address `01h`;
  value `10100100` at address `1Ch`;

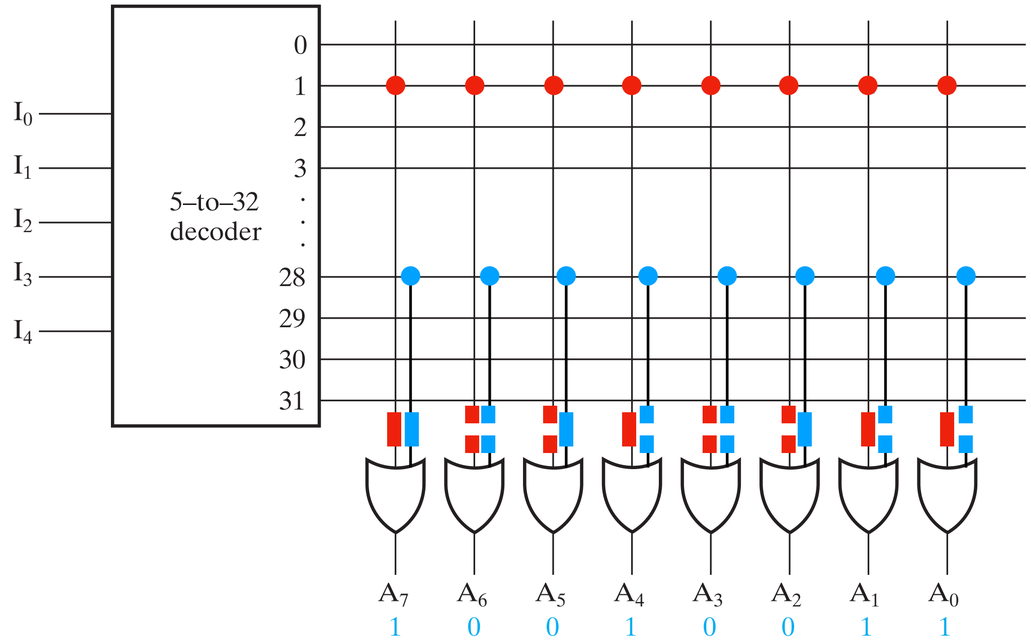
ROM Implementation Example



- This ROM has
  value `10010011` at address `01h`;
  value `10100100` at address `1Ch`;

- In reality, these wiring are done with programmable technology

ROM Implementation Example



- This ROM has
  value `10010011` at address **`01h`**;
  value `10100100` at address **`1Ch`**;

- In reality, these wiring are done with programmable technology

P2
ROM

Reasons for choosing ROM

Concept

P2
ROM

Reasons for choosing ROM

- Originally proposed as unhackable storage devices
  Fixed internal data, cannot be hijacked through software

Concept

Reasons for choosing ROM

- Originally proposed as unhackable storage devices
  Fixed internal data, cannot be hijacked through software

- Usually very fast readings (not as fast as registers but usually **much faster than e.g. HDD**)

Concept

Reasons for choosing ROM

- Originally proposed as unhackable storage devices
  Fixed internal data, cannot be hijacked through software

- Usually very fast readings (not as fast as registers but usually **much faster than e.g. HDD**)

- Fun fact: early **SSD**s were using Flash Memory technology, an **Electrically Erasable ROM** (like in the previous slide), which is why they were so expensive

Concept

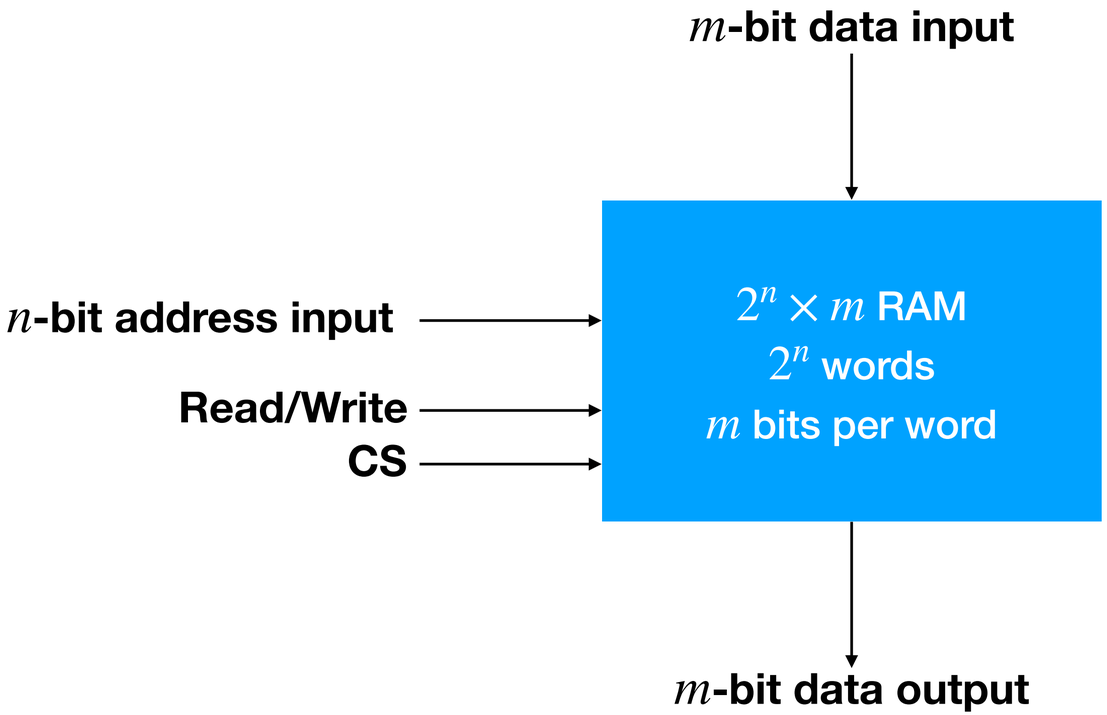
P3
RAM

Read Access Memory

Now supporting **`Write`** in all participating memory sticks

Summary

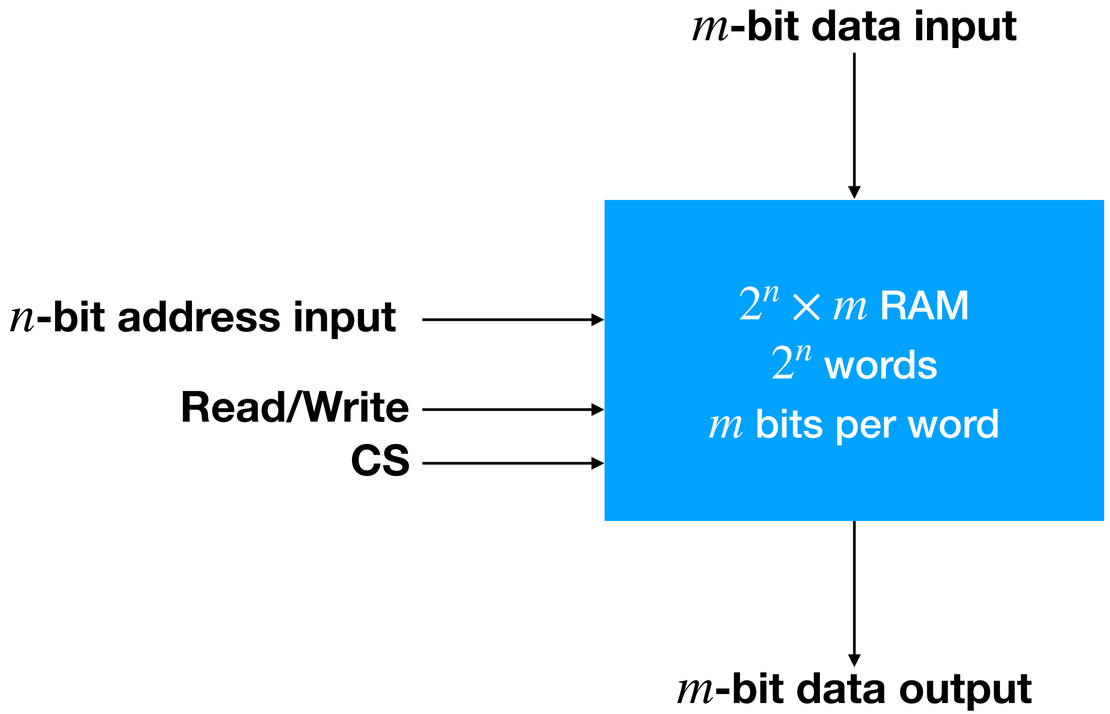
RAM Abstraction



Chip Select CS	Read/$\overline{\text{Write}}$ R/$\overline{\text{W}}$	Memory Operation
0	×	None
1	0	Write to selected word
1	1	Read from selected word

Concept

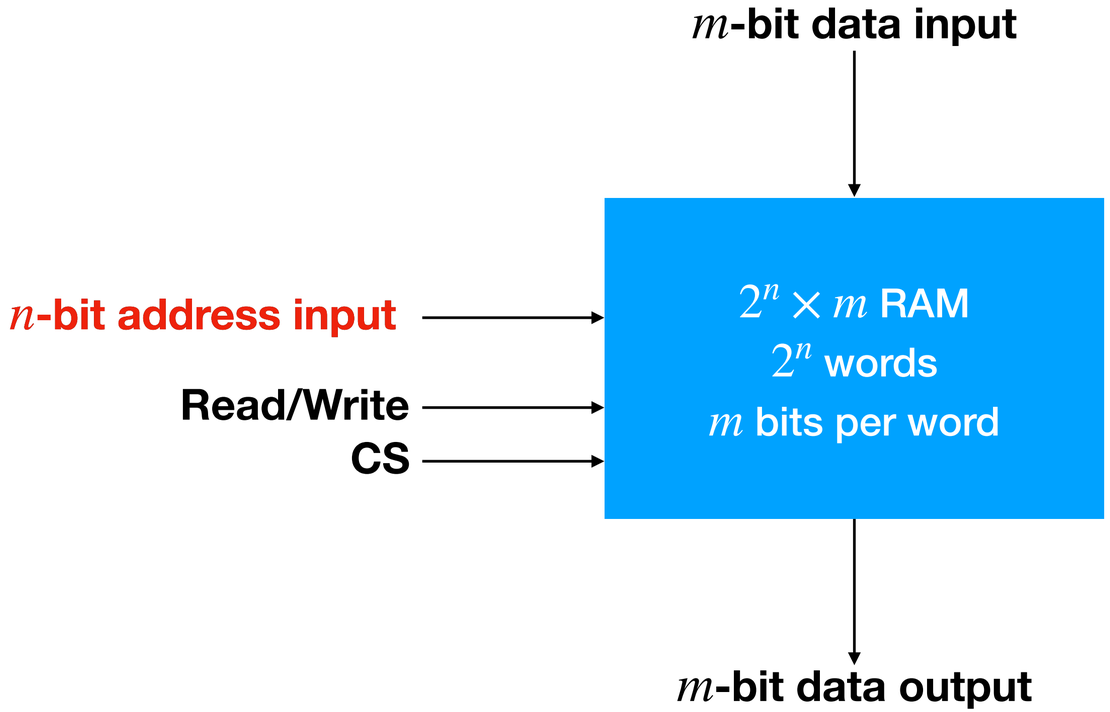
Steps for `Write`



Demo

Steps for `Write`

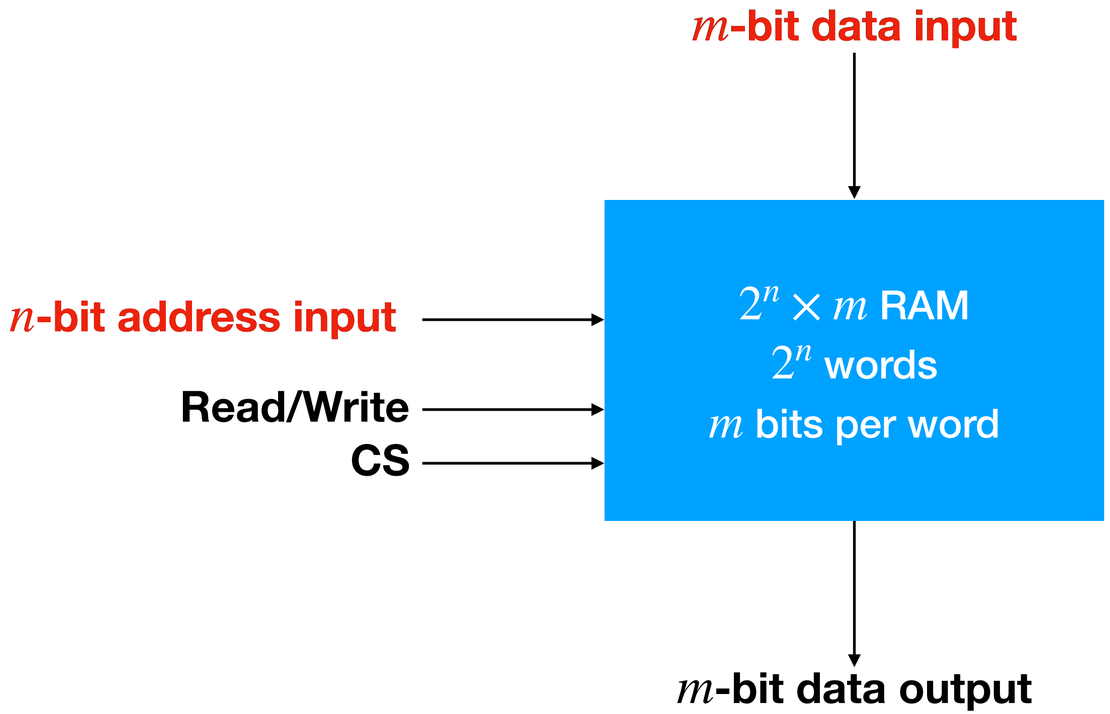
1. Apply the address of the desired word to the address lines



Demo

Steps for `Write`

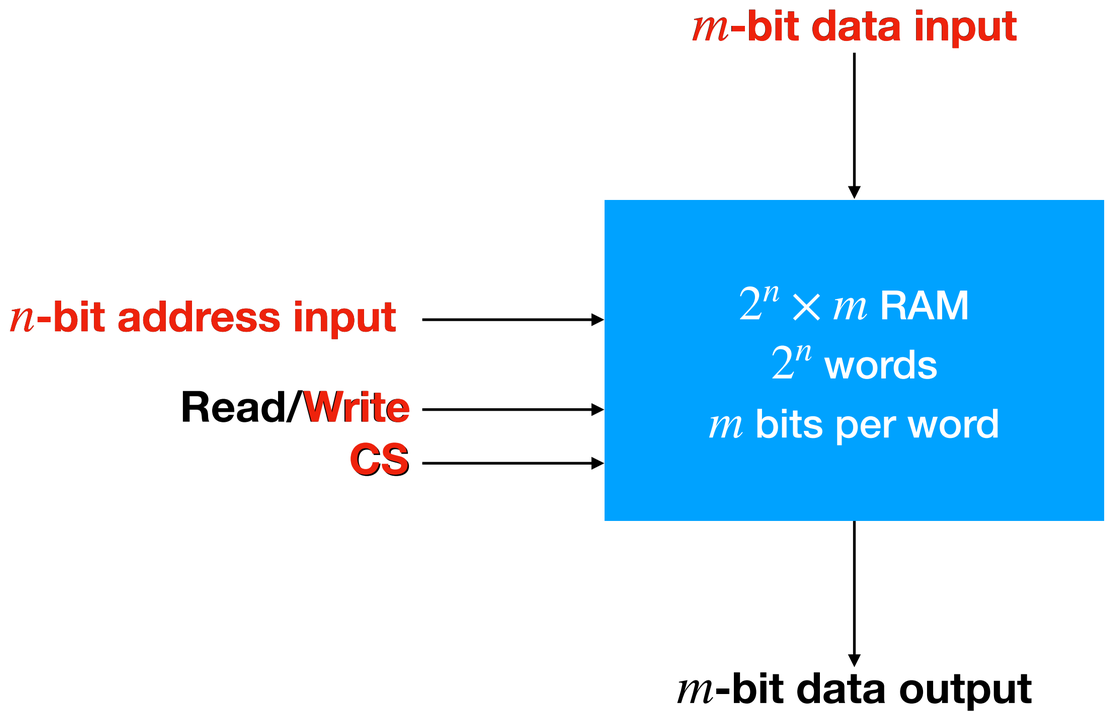
1. Apply the address of the desired word to the address lines
2. Apply the data bits that must be stored in memory to the data input lines



Demo

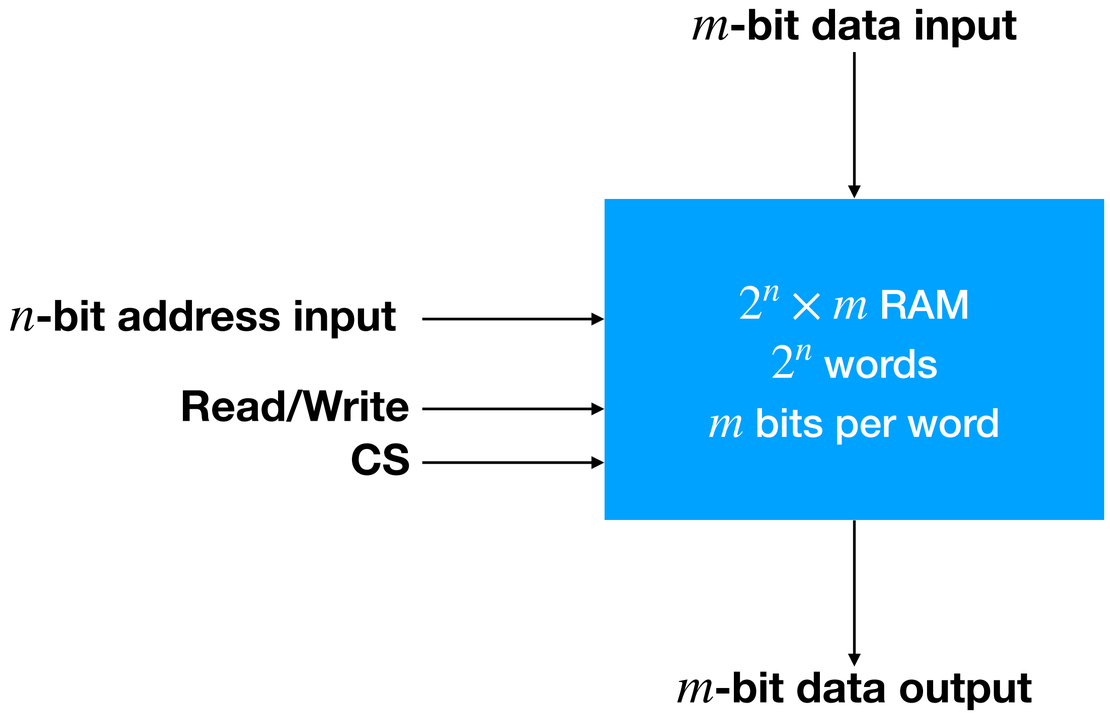
Steps for `Write`

1. Apply the address of the desired word to the address lines
2. Apply the data bits that must be stored in memory to the data input lines
3. Activate the Write input



Demo

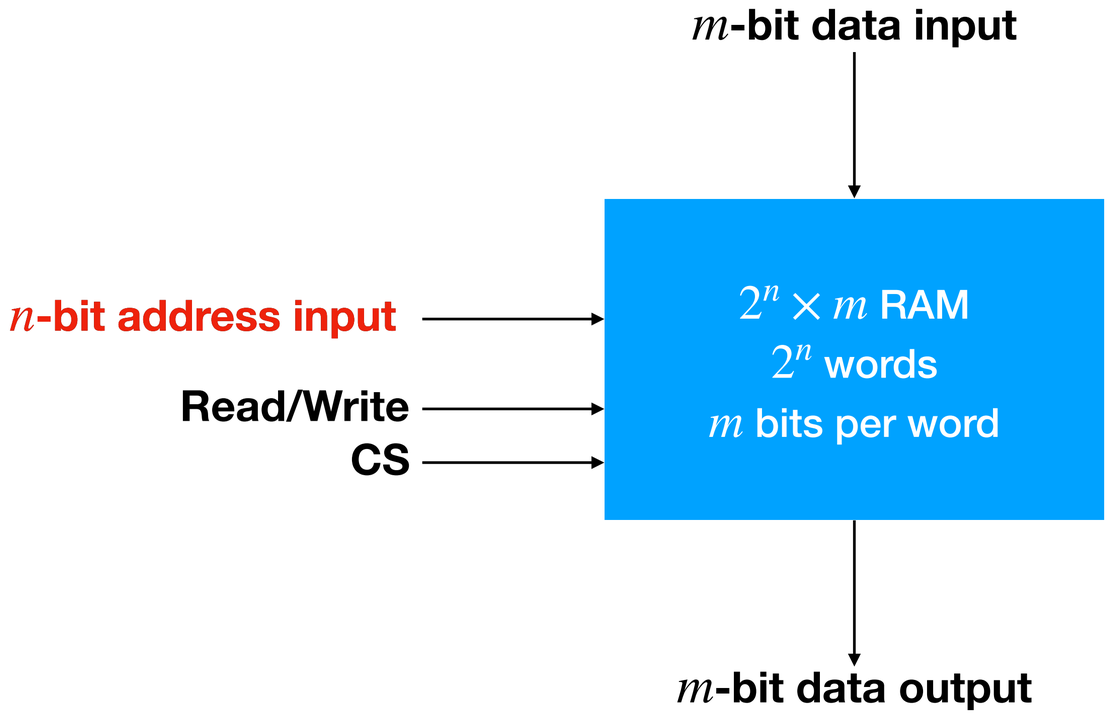
Steps for `Read`



Demo

Steps for `Read`

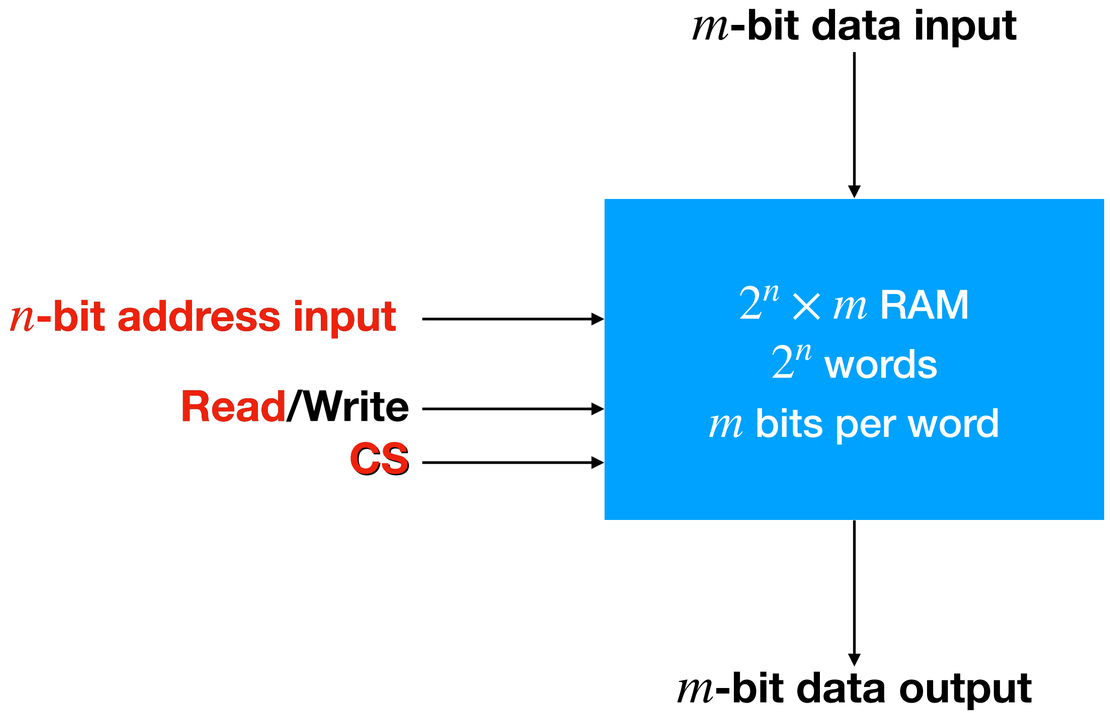
1. Apply the address of the desired word to the address lines



Demo

Steps for `Read`

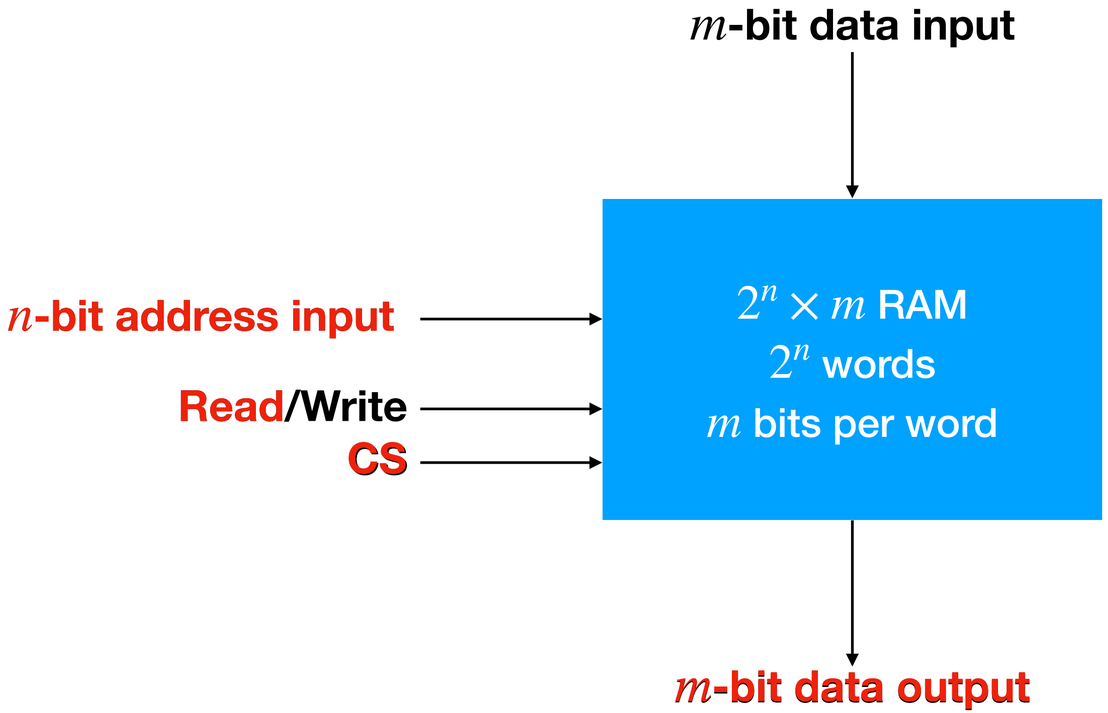
1. Apply the address of the desired word to the address lines
2. Activate the Read input



Demo

Steps for `Read`

1. Apply the address of the desired word to the address lines
2. Activate the Read input



Demo

P4
Summary

Summary Today

- Memory Definition: what are some memory devices? what are not memory devices?
- Read Only Memory
  - And its implementation using OR gate array and decoder
- Random Access Memory
  - What does the interface look like?

Review