EIE3105: Introduction to Computing and AVR/ARM
(Chapter 1 and 2)

Dr. Lawrence Cheung

Semester 1, 2021/22

Topics

- Internal organization of computers
- Memory
- CPU (Central Processing Unit)
- Connecting memory to CPU
- Connecting I/Os (Inputs/Outputs) to CPU
- How computers work
- How Instruction decoder works
- Von Neumann vs. Harvard architecture
- Microcontrollers vs. Microprocessors
- AVR features and members

Internal organization of computers

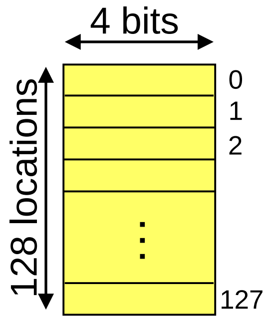
- Memory
- CPU
- I/O
  - Input
    - Example: Keyboard, Mouse, Sensor
  - Output
    - Example: LCD, printer, hands of a robot

Memory

- Everything that can store, retain, and recall information
  - Example: hard disk, a piece of paper, etc.

Memory characteristics

- Capacity
  - The number of bits that a memory can store.
    - Example: 128 Kbits, 256 Mbits
- Organization
  - How the locations are organized?
    - Example: a 128 x 4 memory has 128 locations, 4 bits each
- Access time
  - How long it takes to get data from memory?



Semiconductor memory

- ROM
  - Read-only memory
  - Do not lose its content when the power is turned off.
- Types
  - Mask ROM
  - PROM (Programmable ROM)
  - EPROM (Erasable PROM)
  - UV-EPROM (Ultra-Violet Erasable PROM)
  - EEPROM (Electronic Erasable PROM)
  - Flash EPROM

Semiconductor memory

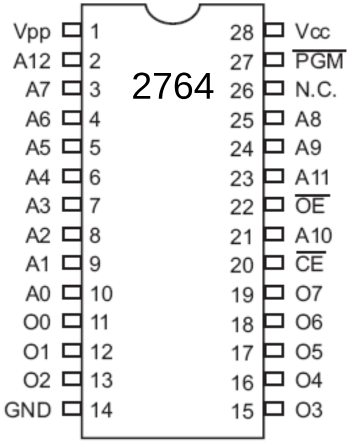
- Mask ROM
  - Programmed by the IC manufacturer
- PROM (Programmable ROM)
  - OTP (One-Time Programmable)
  - You can program it only once

Semiconductor memory

- UV-EPROM
  - You can shine ultraviolet (UV) radiation to erase it.
  - Erasing takes up to 20 minutes.
  - The entire contents of ROM are erased.

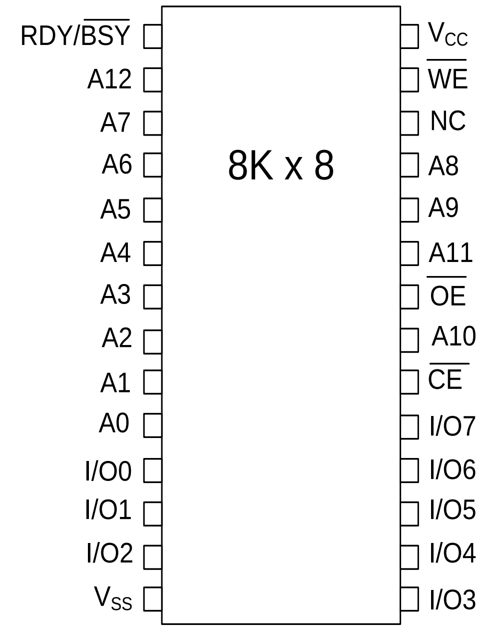
Table 0-5: Some UV-EPROM Chips

Part #	Capacity	Org.	Access	Pins	V_{PP}
2716	16K	2K × 8	450 ns	24	25 V
2732	32K	4K × 8	450 ns	24	25 V
2732A-20	32K	4K × 8	200 ns	24	21 V
27C32-1	32K	4K × 8	450 ns	24	12.5 V CMOS
2764-20	64K	8K × 8	200 ns	28	21 V
2764A-20	64K	8K × 8	200 ns	28	12.5 V
27C64-12	64K	8K × 8	120 ns	28	12.5 V CMOS



Semiconductor memory

- EEPROM
  - Erase electrically
  - Erase instantly
  - Each byte can be erased separately.



Part No.	Capacity	Org.	Speed	Pins	V_{PP}
2816A-25	16K	2K × 8	250 ns	24	5 V
2864A	64K	8K × 8	250 ns	28	5 V
28C64A-25	64K	8K × 8	250 ns	28	5 V CMOS
28C256-15	256K	32K × 8	150 ns	28	5 V
28C256-25	256K	32K × 8	250 ns	28	5 V CMOS

Semiconductor memory

- Flash EPROM
  - Flash memory
  - Erase in a flash
  - The entire device is erased at once.
    - Unlike EEPROM, there is no byte erasure option.

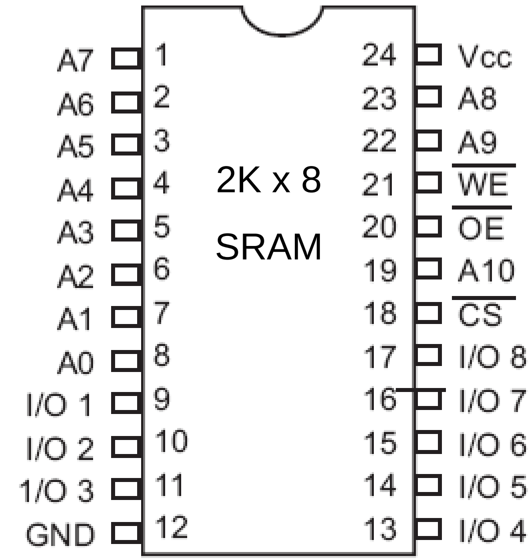
Part No.	Capacity	Org.	Speed	Pins	V_{PP}
28F256-20	256K	32K × 8	200 ns	32	12 V CMOS
28F010-15	1024K	128K × 8	150 ns	32	12 V CMOS
28F020-15	2048K	256K × 8	150 ns	32	12 V CMOS

Semiconductor memory

- RAM
  - Random access memory
  - Loss data when the power is turned off.
- Types
  - SRAM (Static RAM)
  - DRAM (Dynamic RAM)
  - NV-RAM (Non-volatile RAM)

Semiconductor memory

- SRAM
  - Made of flip-flops (Transistors)
  - Advantages:
    - Fast
    - No need for refreshing
  - Disadvantages:
    - High power consumption
    - Expensive



Semiconductor memory

- DRAM
  - Made of capacitors
  - Advantages:
    - Less power consumption
    - Cheap
    - High capacity
  - Disadvantages:
    - Slow
    - Refresh needed

Semiconductor memory

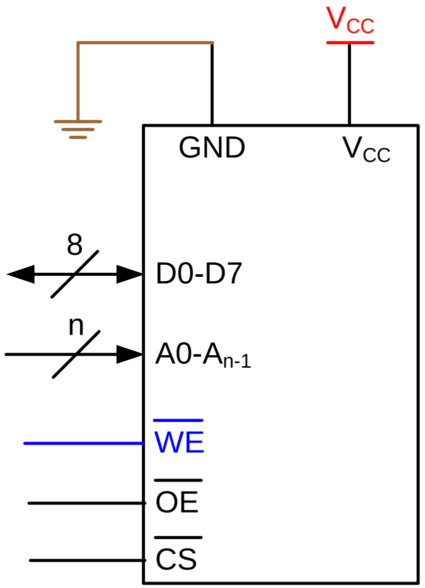
- NV-RAM
  - Made of SRAM, Battery, control circuitry
  - Advantages:
    - Very fast
    - Infinite program/erase cycle
    - Non-volatile
  - Disadvantage:
    - Expensive

CPU

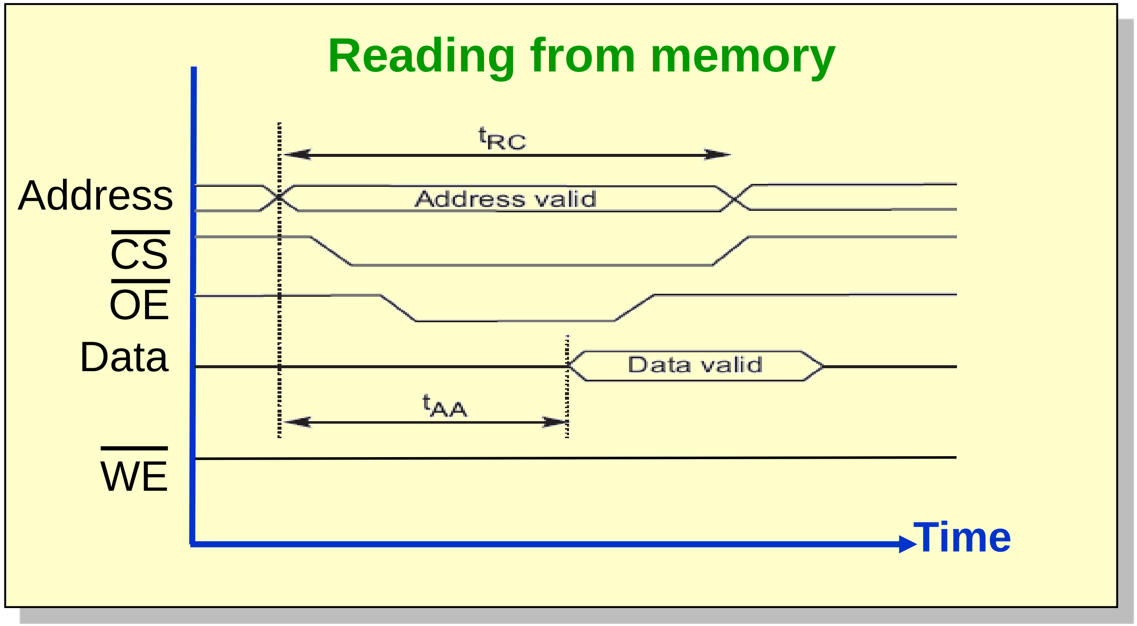
- Tasks:
  - It executes instructions.
    - It recalls the instructions one after another and execute them.

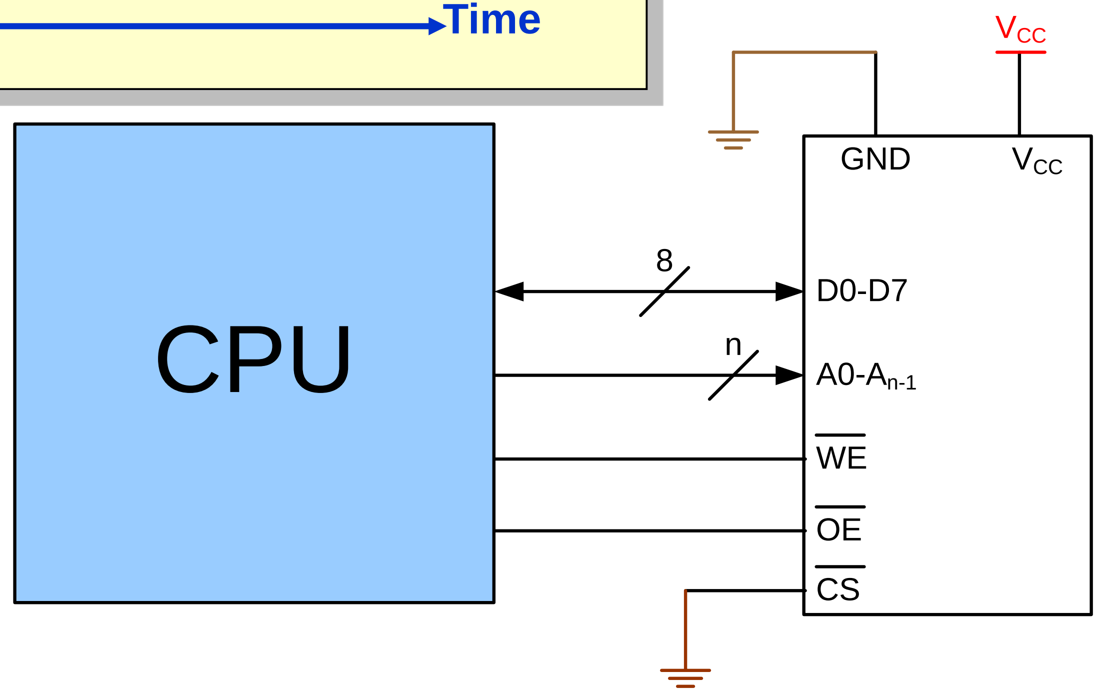
Connecting memory to CPU

- Memory pinout

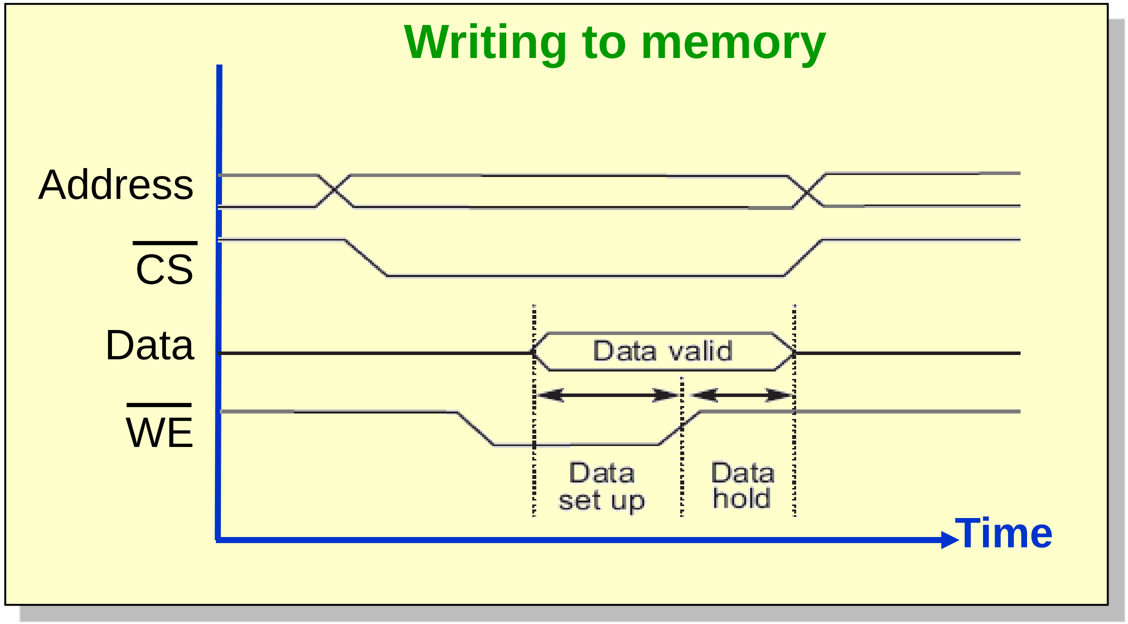


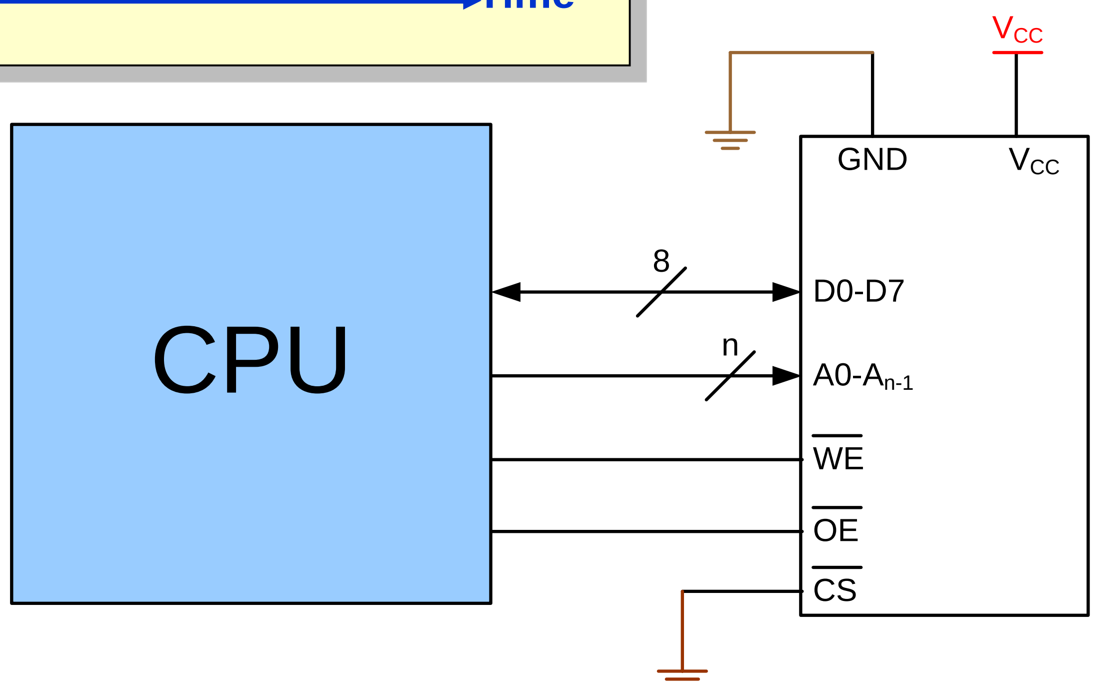
Connecting memory to CPU





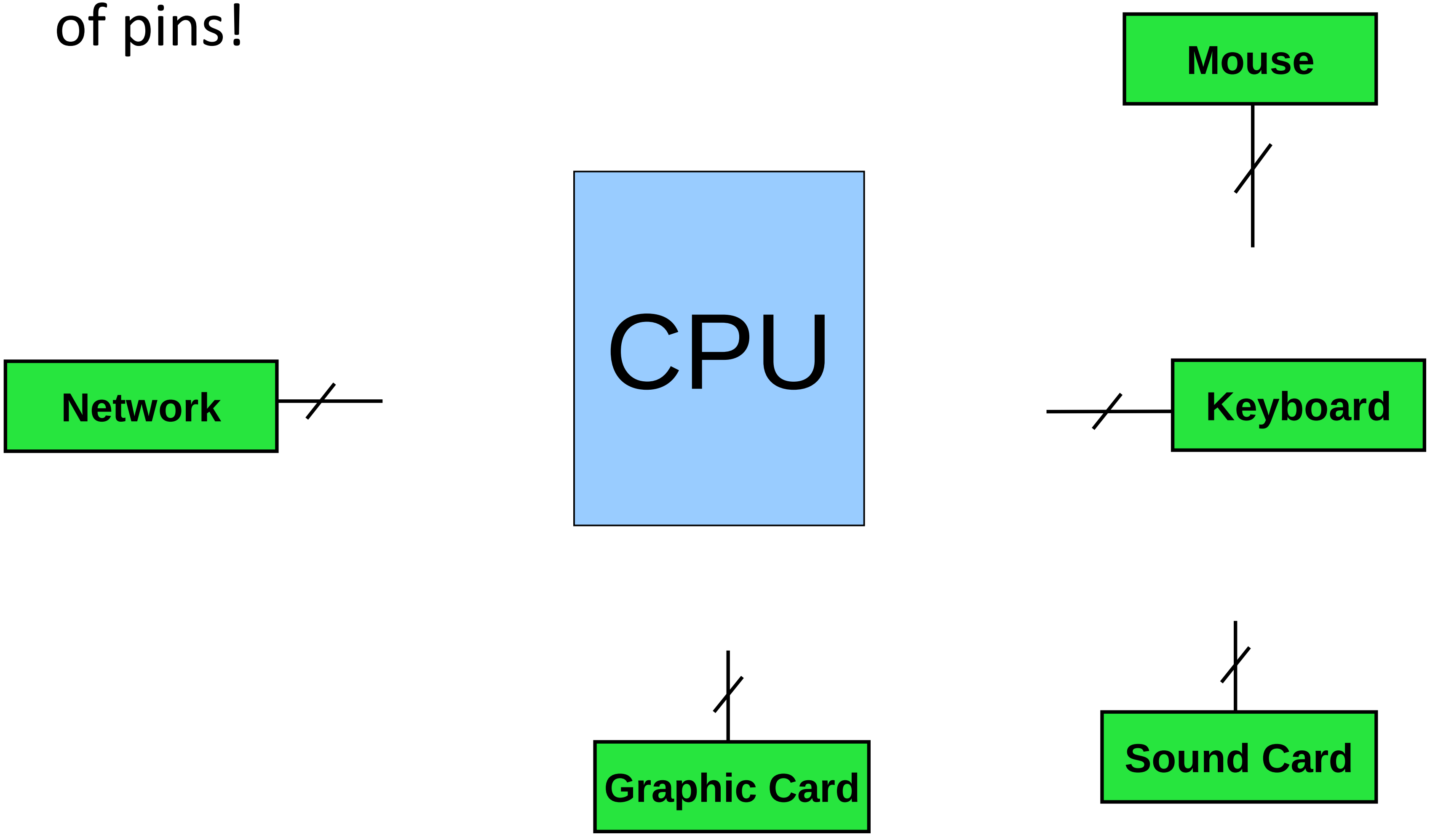
Connecting memory to CPU



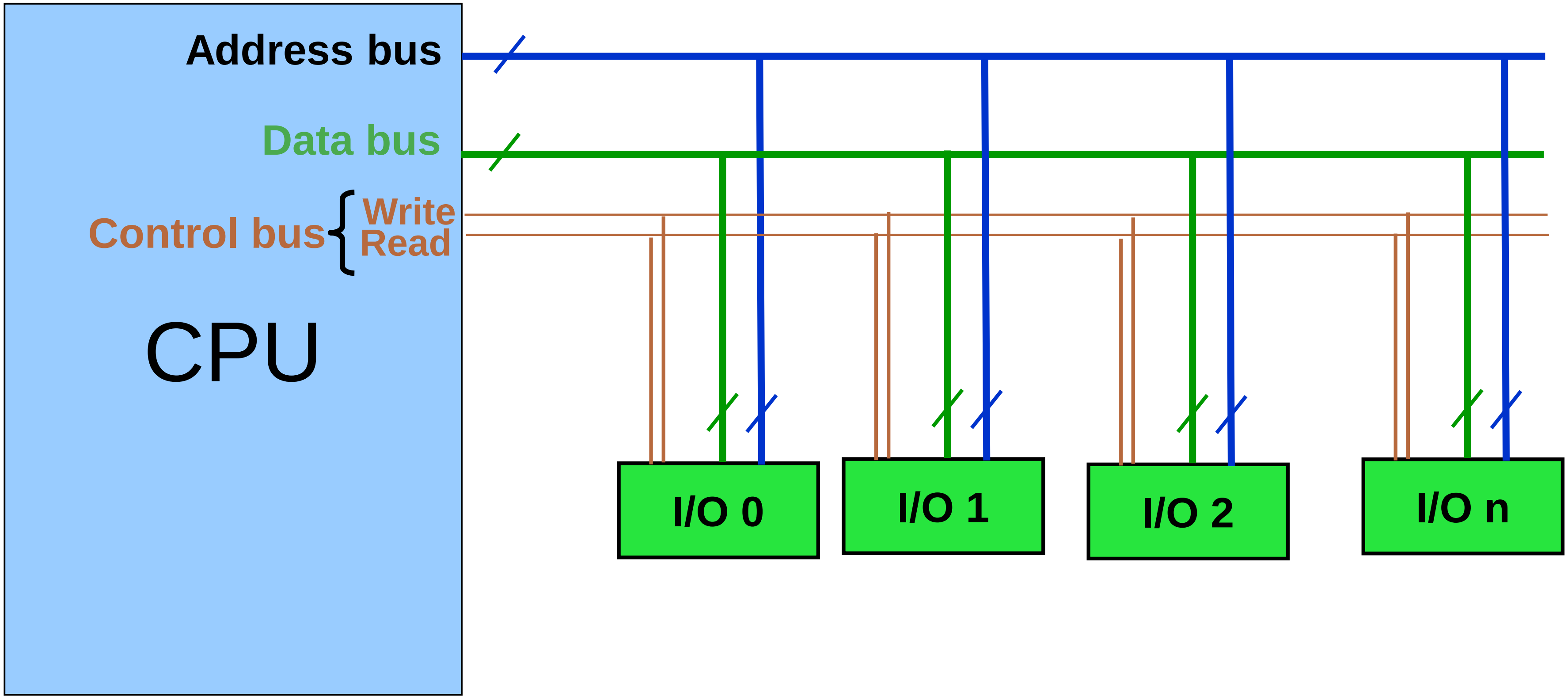


Connecting I/Os to CPU

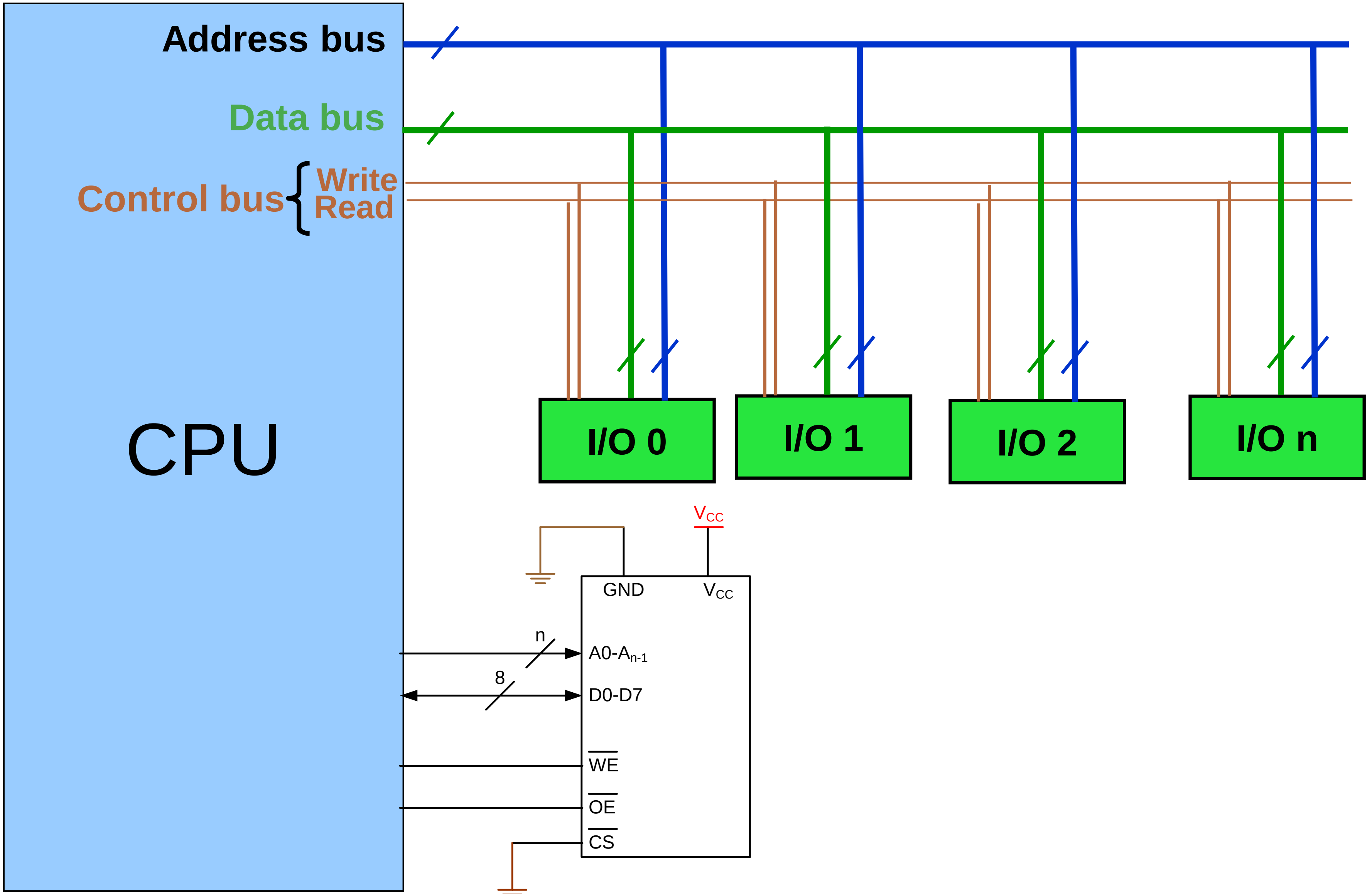
- CPU should have lots of pins!



Connecting I/Os to CPU using bus

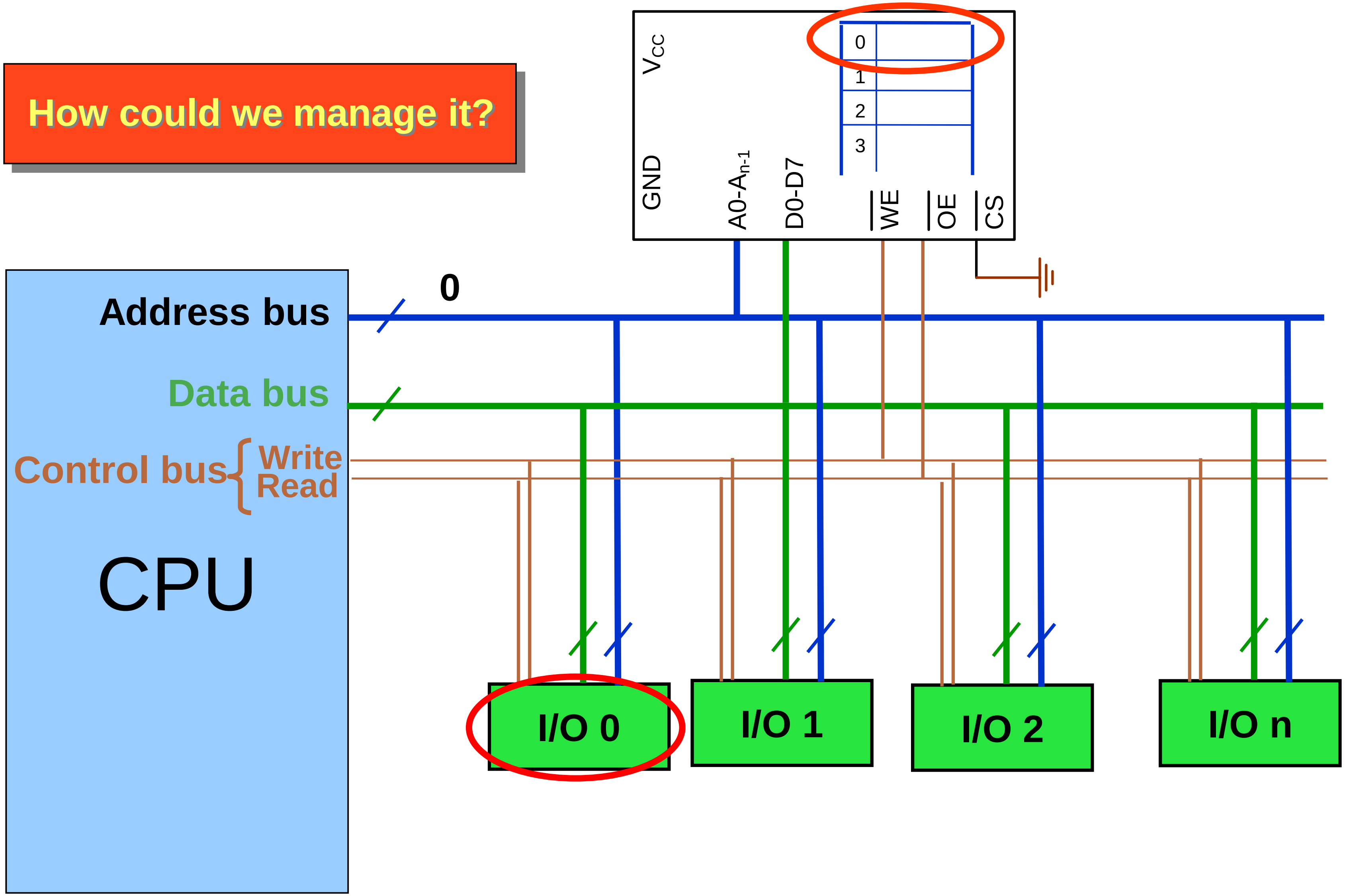


Connecting I/Os and memory to CPU



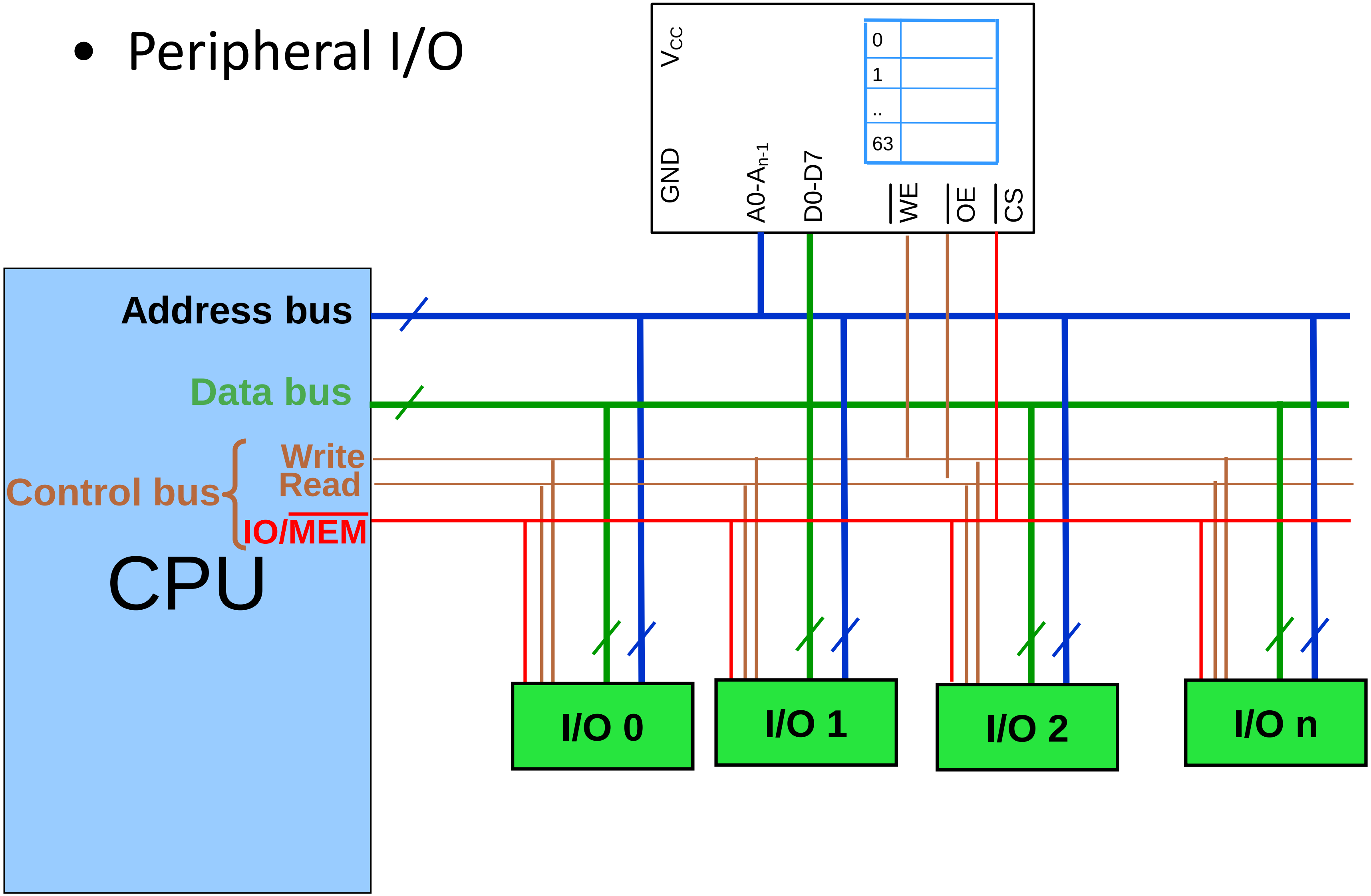
Connecting I/Os and memory to CPU using bus

How could we manage it?



Connecting I/Os and memory to CPU using bus

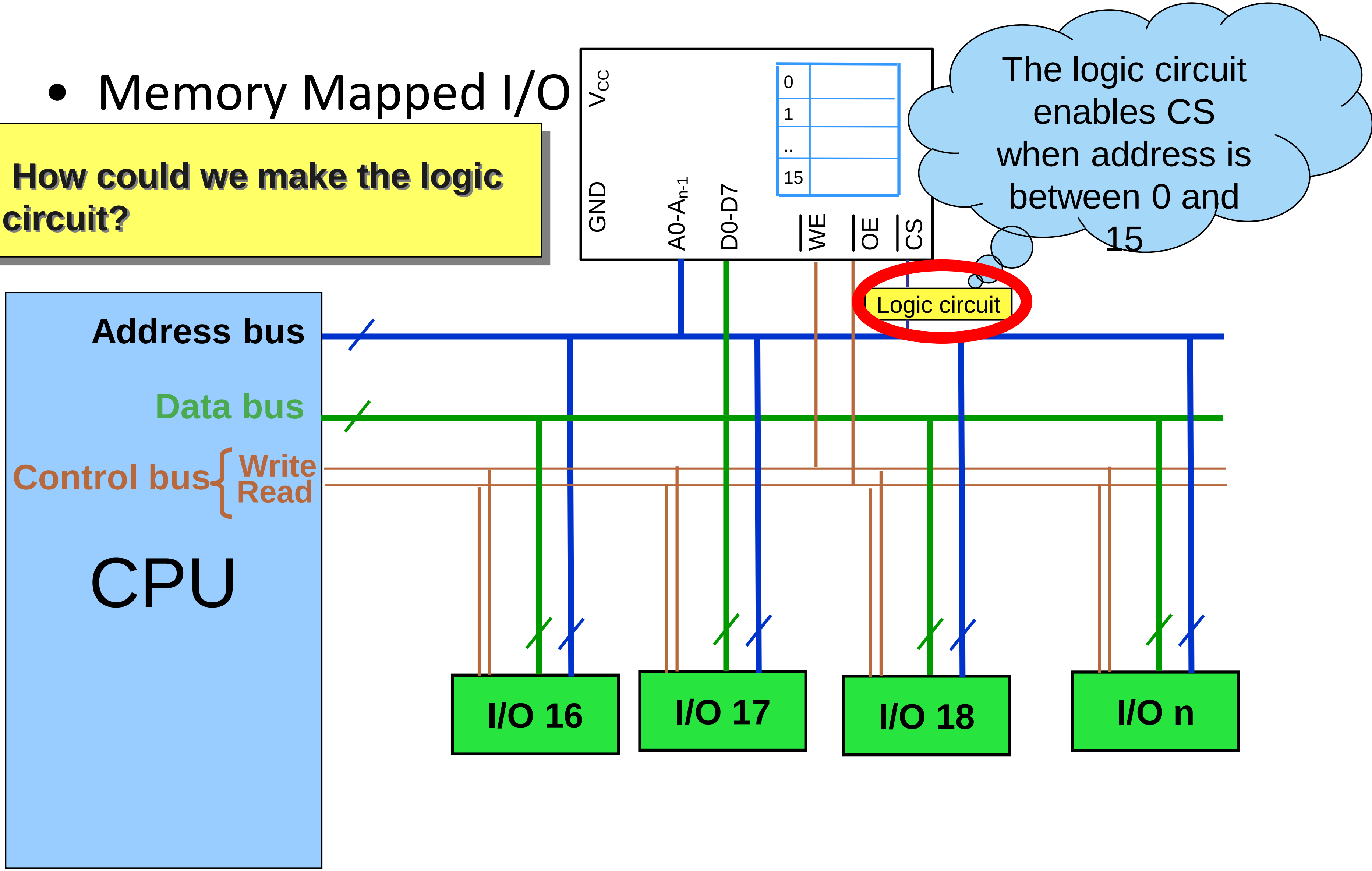
- Peripheral I/O



Connecting I/Os and memory to CPU using bus

- Memory Mapped I/O

How could we make the logic circuit?



Connecting I/Os and memory to CPU using bus

- Memory Mapped I/O

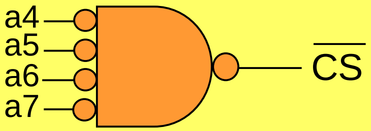
Solution

1. Write the address range in binary

3. Using a NAND, design a logic circuit whose output activates when the fixed address is given to it.

From address 0 ➔ 0 0 0 0 0 0 0 0

To address15 ➔ 0 0 0 0 1 1 1 1



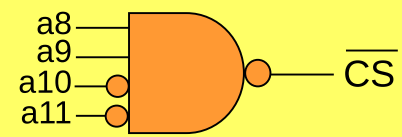
Another example for address decoder

- Design an address decoder for address of 300H to 3FFH.

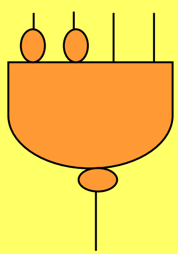
Solution

1. Write the address range in binary
2. Separate the fixed part of address
3. Design the logic circuit.

	a11	**a10**	**a9**	**a8**	a7	a6	a5	a4	a3	a2	a1	a0
From address 300H ➔	0	0	1	1	0	0	0	0	0	0	0	0
To address 3FFH ➔	**0**	**0**	**1**	**1**	1	1	1	1	1	1	1	1

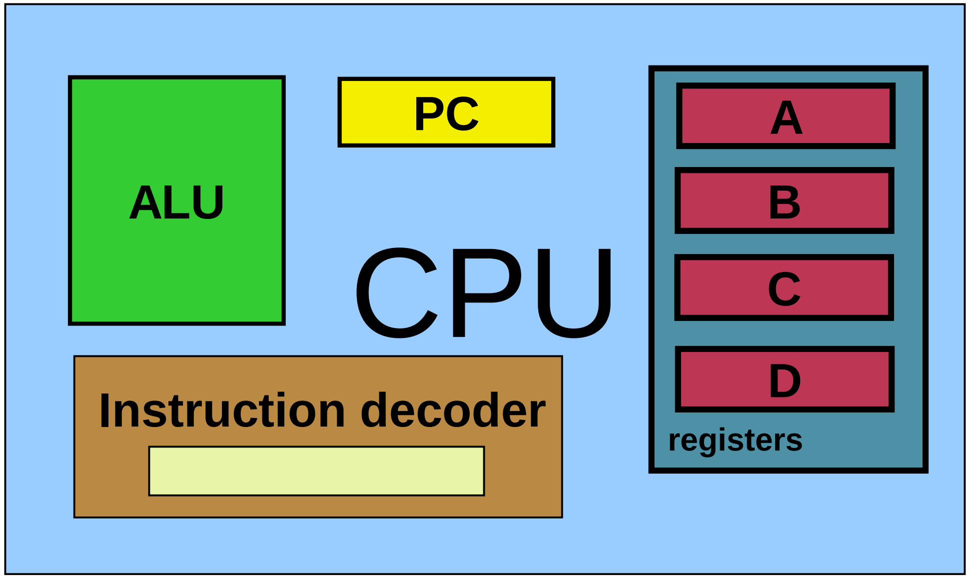


An easy way of designing



Inside the CPU

- PC (Program Counter)
- Instruction decoder
- ALU (Arithmetic Logic Unit)
- Registers



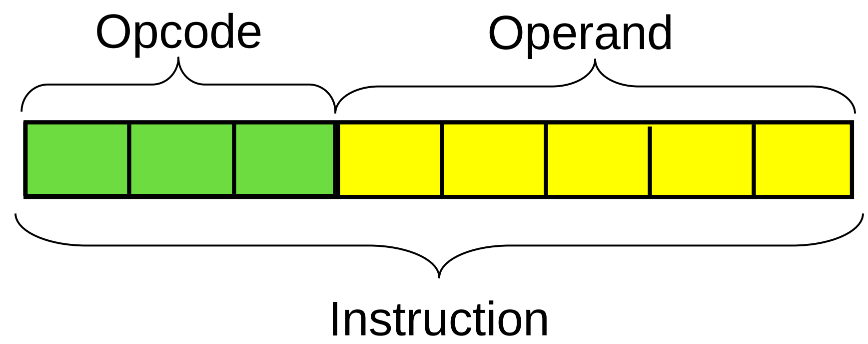
How Instruction decoder works

Opcode 100

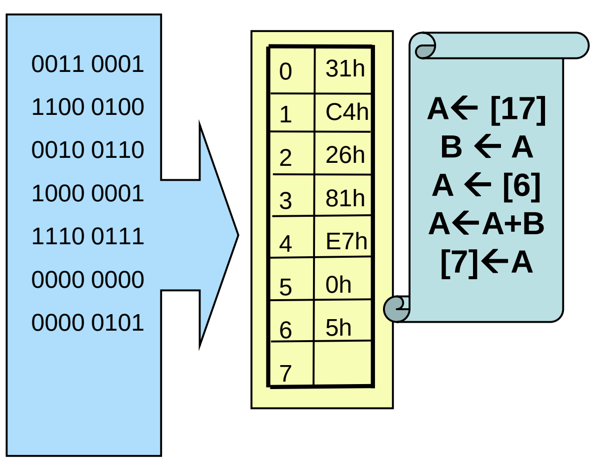
Operand	register (x)
00001	B
00010	C
00011	D

Opcode 110

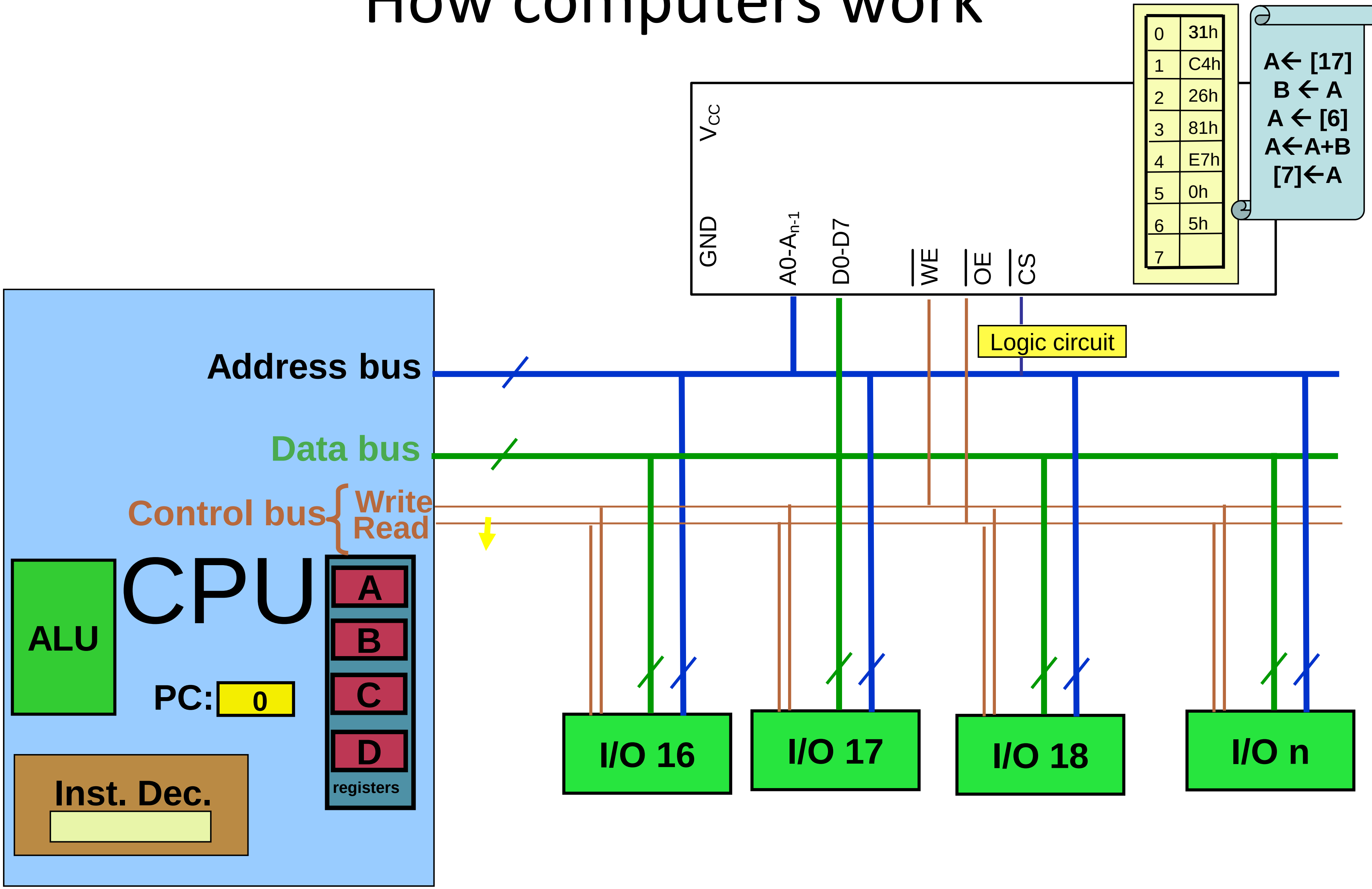
Operand	Register (x_H) ← Register (x_L)
00001	A ← B
00010	A ← C
00011	A ← D
00100	B ← A
00101	C ← A
00110	D ← A



Operation Code	Meaning
000	A ← x
001	A ← [x]
010	A ← A – register (x)
011	A ← A + x
100	A ← A + register (x)
101	A ← A – x
110	Register (x_H) ← Register (x_L)
111	[x] ← A

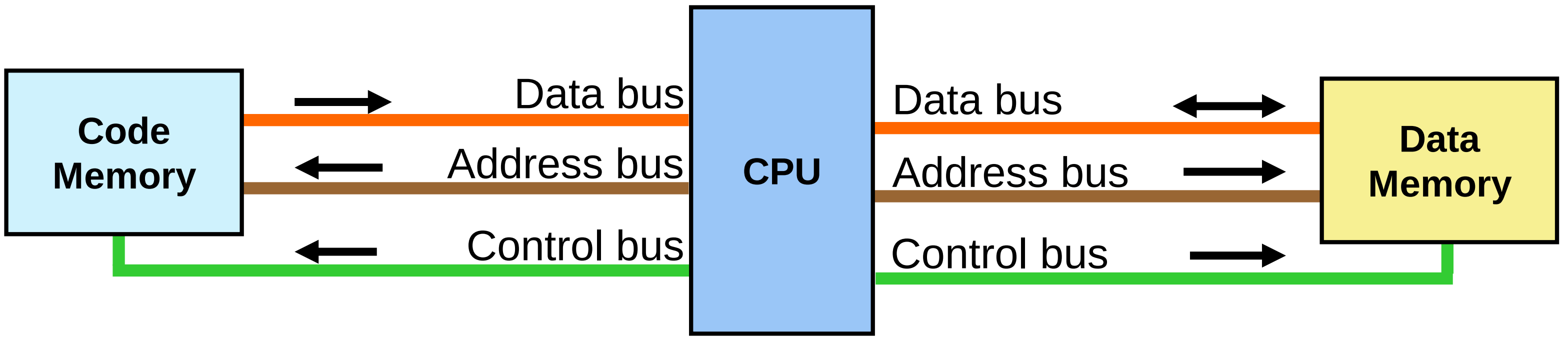


How computers work

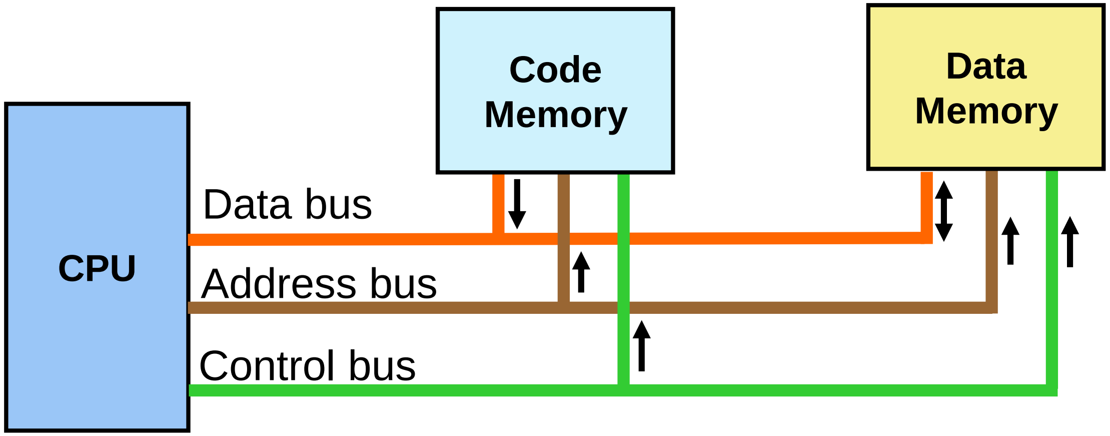


Von Neumann vs. Harvard architecture

- Harvard architecture

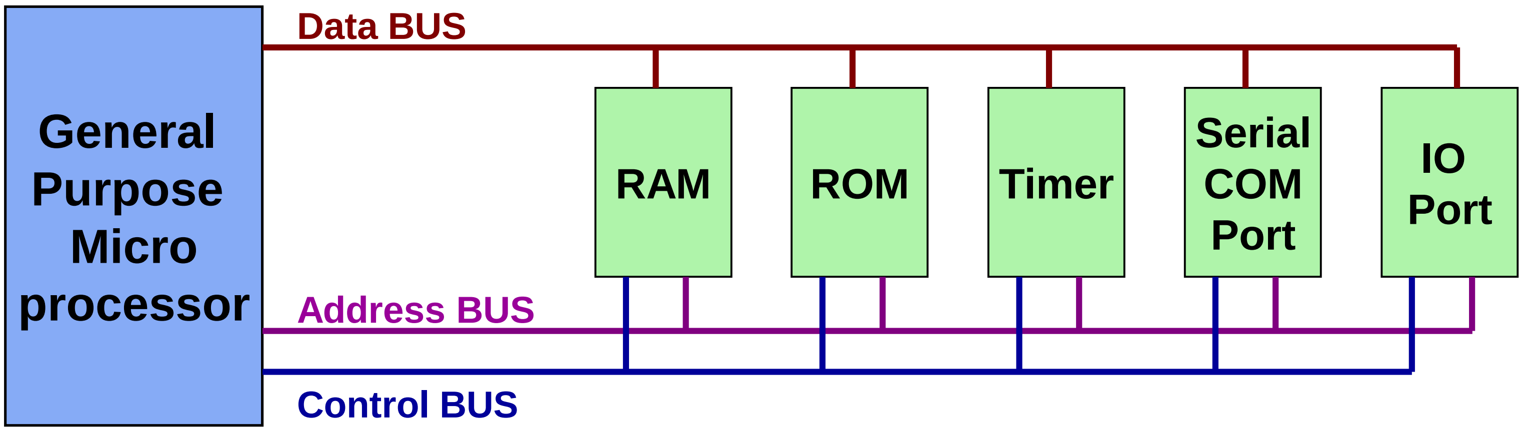


- Von Neumann architecture

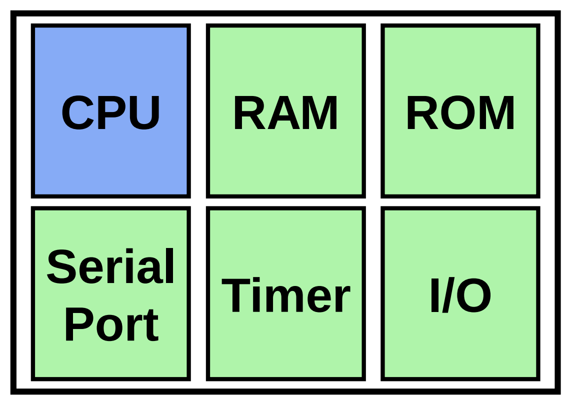


Microcontrollers vs. Microprocessors

- General purpose microprocessors



- Microcontrollers



Microcontrollers vs. Microprocessors

- General purpose microprocessors
  - CPU for Computers
  - No RAM, ROM, I/O on CPU chip itself
  - Example：Intel’s x86, Motorola’s 680
  - CPU is stand-alone, RAM, ROM, I/O, timer are separated.
  - Designer can decide on the amount of ROM, RAM and I/O ports.
  - General-purpose
  - Expensive

Microcontrollers vs. Microprocessors

- Microcontrollers
  - A smaller computer
  - On-chip RAM, ROM, I/O ports...
  - Example：Motorola’s 6811, Intel’s 8051, Zilog’s Z8 and PIC 16X
  - CPU, RAM, ROM, I/O and timer are all on a single chip.
  - Fix amount of on-chip ROM, RAM, I/O ports
  - Single-purpose
  - Cheap
  - For applications in which cost, power and space are critical.

Microcontrollers vs. Microprocessors

- Embedded System
  - An embedded system means the processor is embedded into that application.
  - An embedded product uses a microprocessor or microcontroller to do one task only.
  - In an embedded system, there is only one application software that is typically burned into ROM.

Microcontrollers vs. Microprocessors

- Three Criteria in Choosing a Microcontroller:

1. Meeting the computing needs of the task efficiently and cost effectively
   - Speed, the amount of ROM and RAM, the number of I/O ports and timers, size, packaging, power consumption
   - Easy to upgrade
   - Cost per unit

Microcontrollers vs. Microprocessors

2. Availability of software development tools
   – Assemblers, debuggers, C compilers, emulator, simulator, technical support
3. Wide availability and reliable sources of the microcontrollers

Microcontrollers vs. Microprocessors

- Some embedded products using microcontrollers

Home	Office
Appliances	Telephones
Intercom	Computers
Telephones	Security systems
Security systems	Fax machine
Garage door openers	Microwave
Answering machines	Copier
Fax machines	Laser printer
Home computers	Color printer
TVs	Paging
Cable TV tuner	**Auto**
VCR	Trip computer
Camcorder	Engine control
Remote controls	Air bag
Video games	ABS
Cellular phones	Instrumentation
Musical instruments	Security system
Sewing machines	Transmission control
Lighting control	Entertainment
Paging	Climate control
Camera	Cellular phone
Pinball machines	Keyless entry
Toys	
Exercise equipment	

Class Exercise 1

- In each of following embedded products, what is the most important factor (the power consumption, the ROM size and the number of interrupt service pins) in choosing a microcontroller?

Class Exercise 1 (Your work / Answer)

A mobile electronic device product uses to drive of a number of external devices

A mobile electronic device uses battery

A mobile electronic device uses a complicated real-time operating system

The power consumption

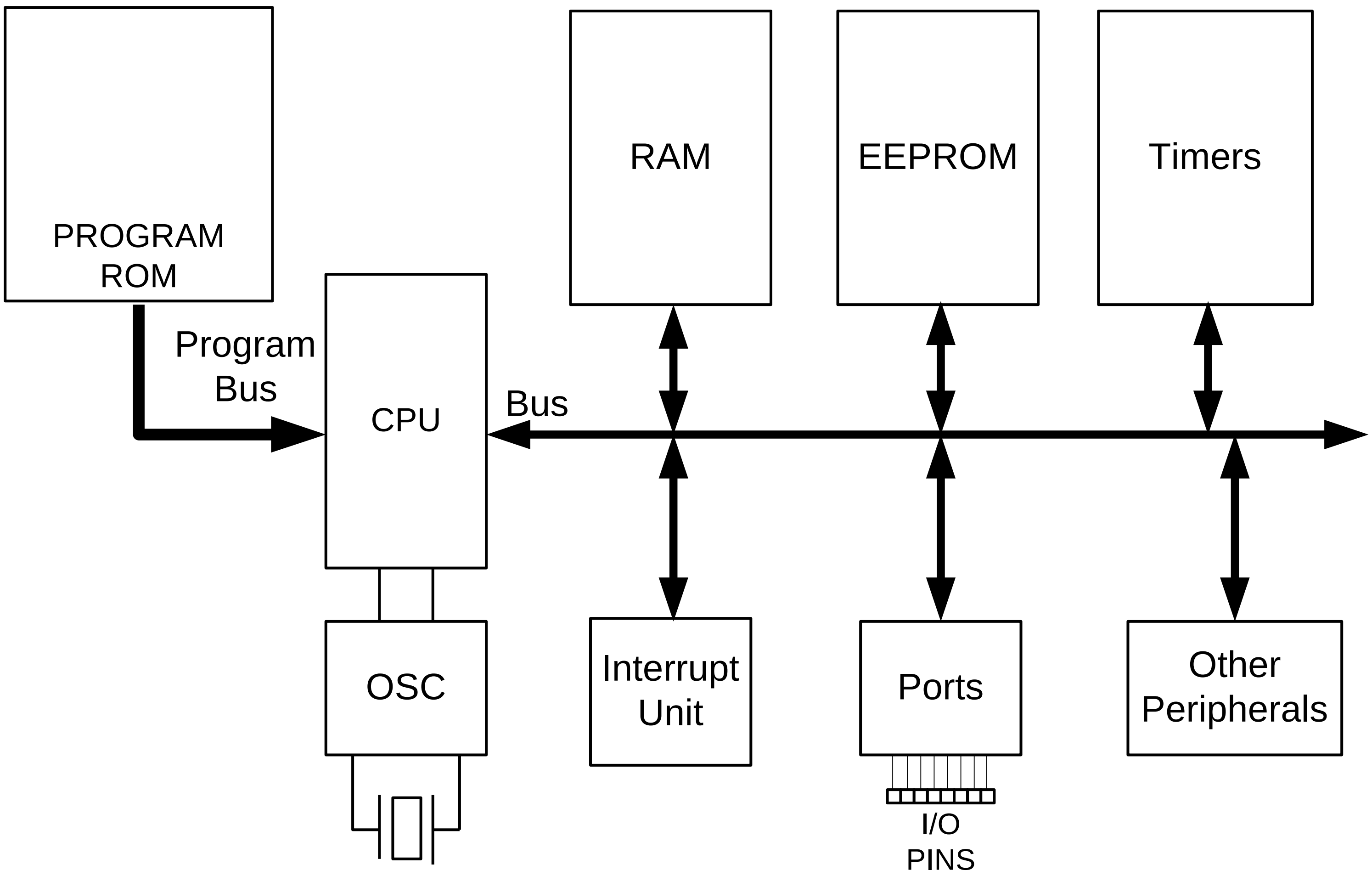
The ROM size

The number of interrupt service pins

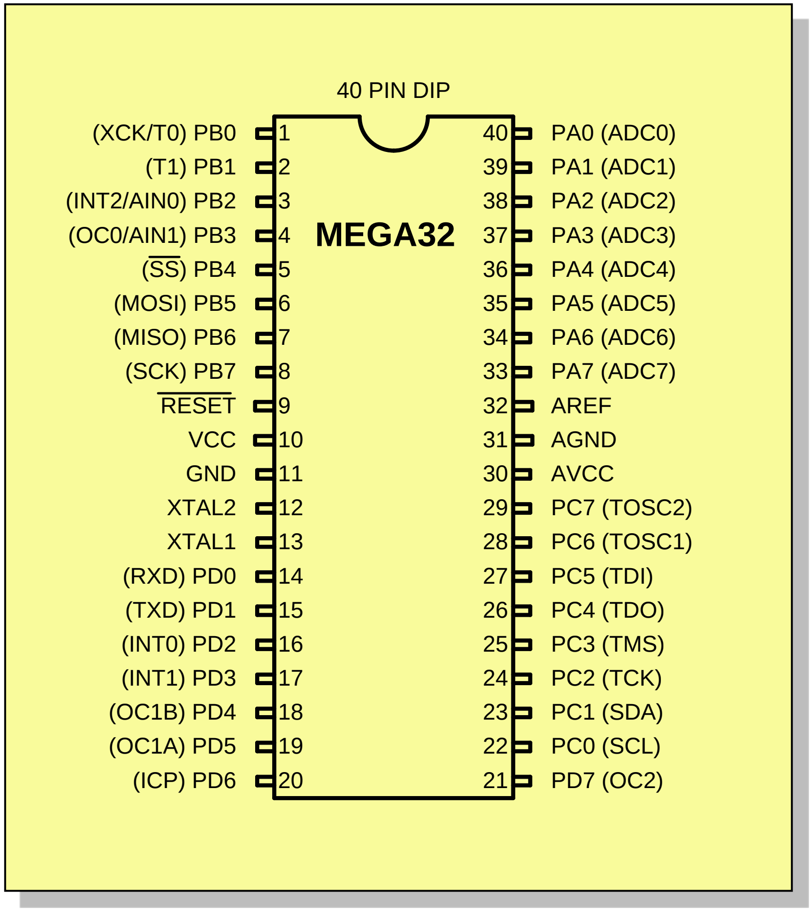
AVR features and members

- 8-bit microcontrollers
  - AVR (Advanced and Virtual RISC or Alf and Vegard RISC)
    - RISC = Reduced Instruction Set Computing
  - PIC (Peripheral Interface Controller)
  - HCS12 (Freescale semiconductor)
  - 8051 (Intel)
- 32-bit microcontrollers
  - ARM (Advanced RISC Machine)
  - PIC32

AVR features and members



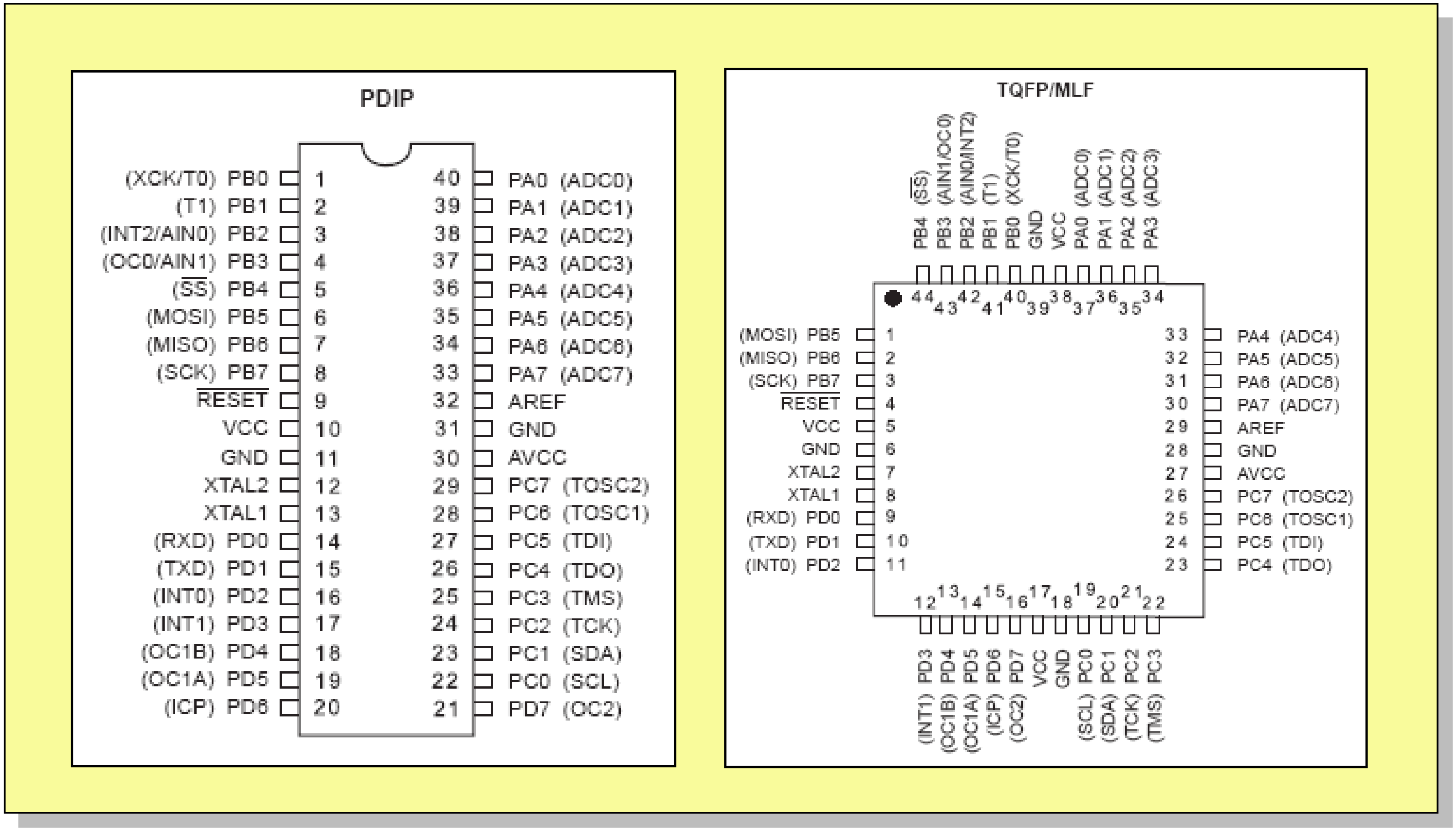
AVR features and members



AVR features and members

- Classic AVR
  - Example: AT90S2313, AT90S4433
- Mega
  - Example: ATmega8, ATmega32, ATmega128
- Tiny
  - Example: ATtiny13, ATtiny25
- Special Purpose AVR
  - Example: AT90PWM216,AT90USB1287

AVR features and members



AVR features and members

Table 1-3: Some Members of the Classic Family

Part Num	Code ROM	Data RAM	Data EEPROM	I/O pins pins	ADC	Timers	Pin numbers & Package
AT90S2313	2K	128	128	15	0	2	SOIC20,PDIP20
AT90S2323	2K	128	128	3	0	1	SOIC8,PDIP8
AT90S4433	4K	128	256	20	6	2	TQFP32,PDIP28

Notes:
1. All ROM, RAM, and EEPROM memories are in bytes.
2. Data RAM (General-Purpose RAM) is the amount of RAM available for data manipulation (scratch pad) in addition to the Registers space.

Table 1-4: Some Members of the Mega Family

Part Num	Code ROM	Data RAM	Data EEPROM	I/O pins pins	ADC	Timers	Pin numbers & Package
ATmega8	8K	1K	0.5K	23	8	3	TQFP32,PDIP28
ATmega16	16K	1K	0.5K	32	8	3	TQFP44,PDIP40
ATmega32	32K	2K	1K	32	8	3	TQFP44,PDIP40
ATmega64	64K	4K	2K	54	8	4	TQFP64,MLF64
ATmega1280	128K	8K	4K	86	16	6	TQFP100,CBGA

Notes:
1. All ROM, RAM, and EEPROM memories are in bytes.
2. Data RAM (General-Purpose RAM) is the amount of RAM available for data manipulation (scratch pad) in addition to the Registers space.
3. All the above chips have USART for serial data transfer.

AVR features and members

Table 1-5: Some Members of the Tiny Family

Part Num	Code ROM	Data RAM	Data EEPROM	I/O pins pins	ADC	Timers	Pin numbers & Package
ATtiny13	1K	64	64	6	4	1	SOIC8,PDIP8
ATtiny25	2K	128	128	6	4	2	SOIC8,PDIP8
ATtiny44	4K	256	256	12	8	2	SOIC14,PDIP14
ATtiny84	8K	512	512	12	8	2	SOIC14,PDIP14

Table 1-6: Some Members of the Special purpose Family

Part Num	Code ROM	Data RAM	Data EEPROM	Max I/O pins	Special Capabilities	Timers	Pin numbers & Package
AT90CAN128	128K	4K	4K	53	CAN	4	LQFP64
AT90USB1287	128K	8K	4K	48	USB Host	4	TQFP64
AT90PWM216	16K	1K	0.5K	19	Advanced PWM	2	SOIC24
ATmega169	16K	1K	0.5K	54	LCD	3	TQFP64,MLF64

AVR features and members

ATmega128

- AT → Atmel
- mega → group
- 128 → Flash =128K

ATtiny44

- AT → Atmel
- tiny → Tiny group
- 4 → Flash =4K

AT90S4433

- AT → Atmel
- 90S → Classic group
- 4 → Flash =4K

AVR in textbook and lab.

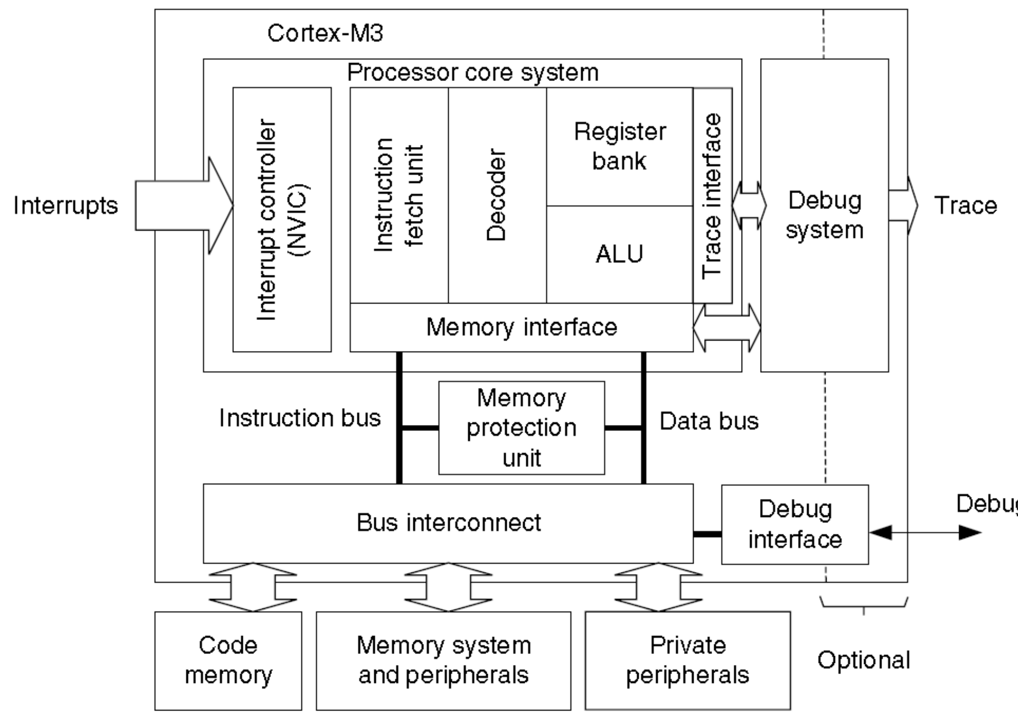
- AVR in textbook: ATmega32
  - 8-bit AVR
  - SRAM: 2 KB
  - EEPROM: 1 KB
  - Flash memory: 32 KB
  - Clock speed: 16 MHz
  - 4 PWM channels

AVR in textbook and lab.

- AVR in laboratory exercise: Atmega328p
  - 8-bit AVR (same)
  - SRAM: 2 KB (same)
  - EEPROM: 1 KB (same)
  - Flash memory: 32 KB (same)
  - Clock speed: 20 MHz (different)
  - 6 PWM channels (different)
  - P: picoPower (power saving)

ARM in textbook and lab.

- ARM Cortex-M3
- Model No.: STM32F103RB
  - STMicroeletronics
  - 32: 32 bits
  - F1: ATM Cortex-M3
  - 03: Performance
  - LQFP64: IC package



Reference Readings

- Chapter 1 and 2 – *The AVR Microcontroller and Embedded Systems : Using Assembly and C*, M. A. Mazidi, S. Naimi, and S. Naimi, Pearson, 2014.
- Chapter 2 – *The Definitive Guide To The ARM Cortex-M3*, Joseph Yiu, 2nd edition, Newnes, 2010.

End