

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## میکرو کنترلرهای AVR

دانشکده مهندسی برق

دانشگاه صنعتی شاهرود

حسین خسروی

۱۴۰۰

# میکروکنترلر چیست؟

◀ یک مدار مجتمع که برای انجام یک کار خاص، برنامه ریزی می شود

◀ میکروکنترلر در واقع یک مینی کامپیوتر است



Atmel AVR



AVR



ATX Mega



ATmega 328P



PIC 16F877A



8051



Arduino



ARM

# میکروکنترلرها در کجا استفاده می شوند

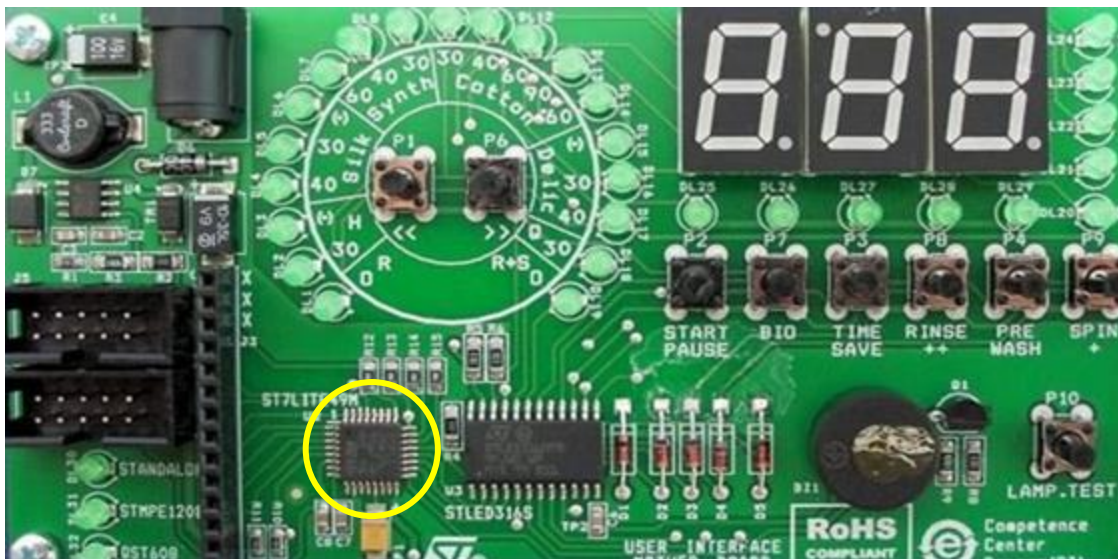
میکروکنترلرها در بسیاری از وسایل خانگی، گجت‌های هوشمند و سایر وسایل الکترونیکی وجود دارند



# نمونه ای از کاربردها

◀ برد ماشین لباسشویی

◀ میکروی هشت بیتی ST7LITE49M با تنها ۴ کیلوبایت حافظه کد





# نمونه ای از کاربردها - برد کنترلر سمند

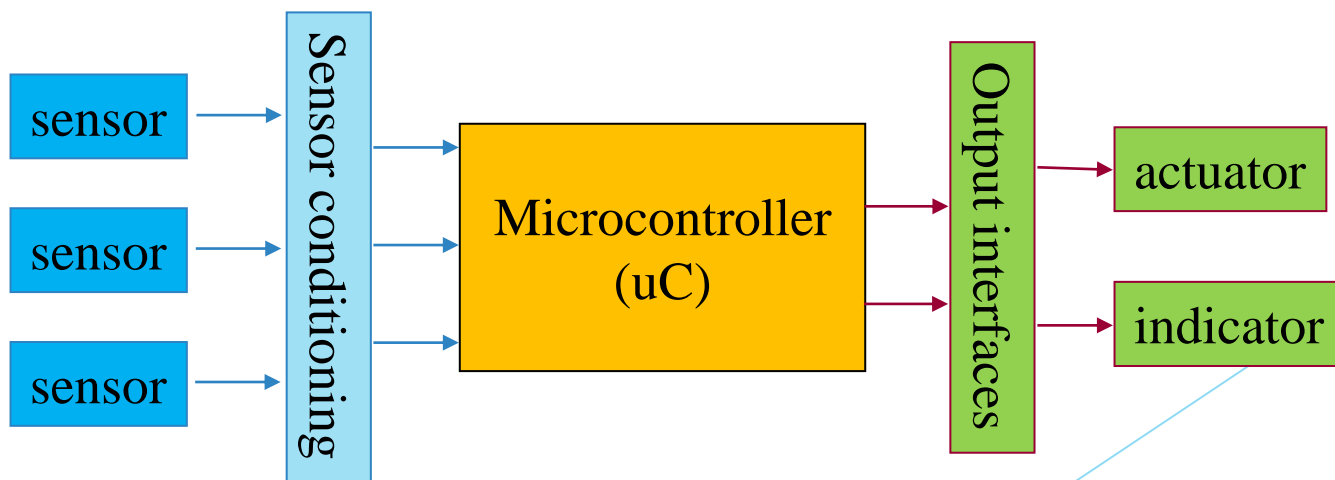
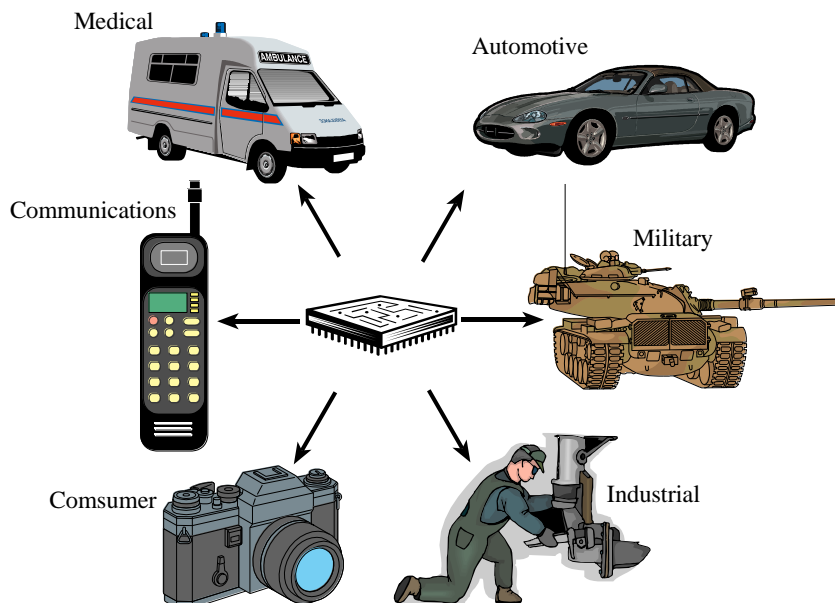


- |   |  |
|---|--|
| 1 - ST10F275-CGE - میکروکنترلر ۱۶ بیتی با حافظه فلش داخلی ۷۶۸ کیلوبایت      | 10 - 30536 - آیسی مولتی راه انداز انژکتورهای گاز و گیج گاز |
| 2 - کریستال ساعت داخلی  | 11 - خازن صافی برق ۱۲ ولت                                  |
| 3 - نوسان ساز کریستالی اصلی میکروکنترلر                                     | 12 - 30424 - آیسی پردازنده سیگنال سنسور ناک                |
| 4 - 74HC08D - دو گیت اند دو ورودی   | 13 - LM2903 - تقویت کننده سیگنال تفاضلی دور موتور          |
| 5 - خازن صافی برای ولتاژ تغذیه ۵ ولت  | 14 - TLE7209 - آیسی راه انداز دریچه گاز برقی               |
| 6 - 30522 - واسط سنسورهای اکسیژن  | 15 - DALE - نوسان گیر پرازیت کوئل ها                       |
| 7 - B82793 - بخشی از مدار محافظتی خطوط شبکه کن                              | 16 - 07096 - راه انداز کوئل ۳                              |
| 8 - 30639 - رگولاتور ولتی و محرک منفی رله های پمپ بنزین و اصلی و گیج بنزین  | 17 - 07096 - راه انداز کوئل ۲                              |
| 9 - 30536 - راه انداز انژکتورهای بنزین - گرم کن سنسور اکسیژن و منفی شیربرقی | 18 - 07096 - راه انداز کوئل ۴                              |
| CVVT  | 19 - 07096 - راه انداز کوئل ۱                              |

راهنمای تعمیرکار روی برد بوش ۷۴۹ سمند و دنا با  
موتور EF7

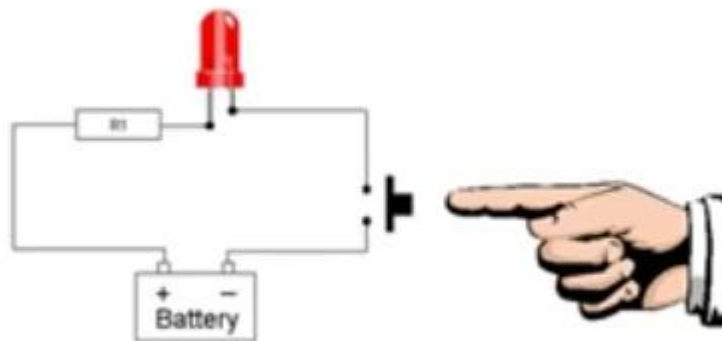
# سیستمهای نهفته - Embedded Systems

سیستمهایی که نیاز به  
رایانه جداگانه ندارند.  
به جای آن یک میکرو  
در درون خود دارند



# کجا باید از میکروکنترلر استفاده کنیم

◀ هر کجا که بخواهیم هوشمندی به سیستم اضافه کنیم





Vacuum Tube  
1939



Transistor  
1947



Logic Gate  
1960

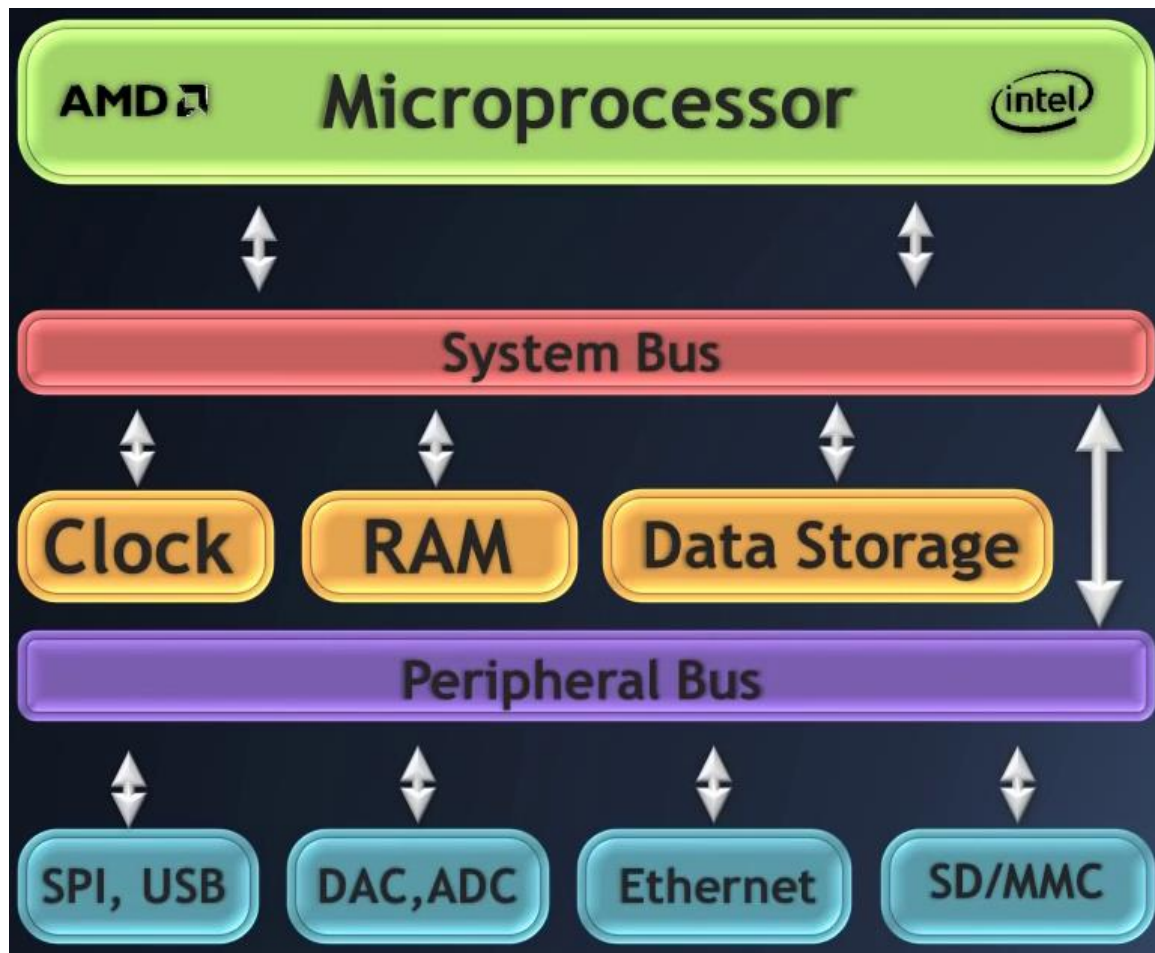


Microcontroller  
1971



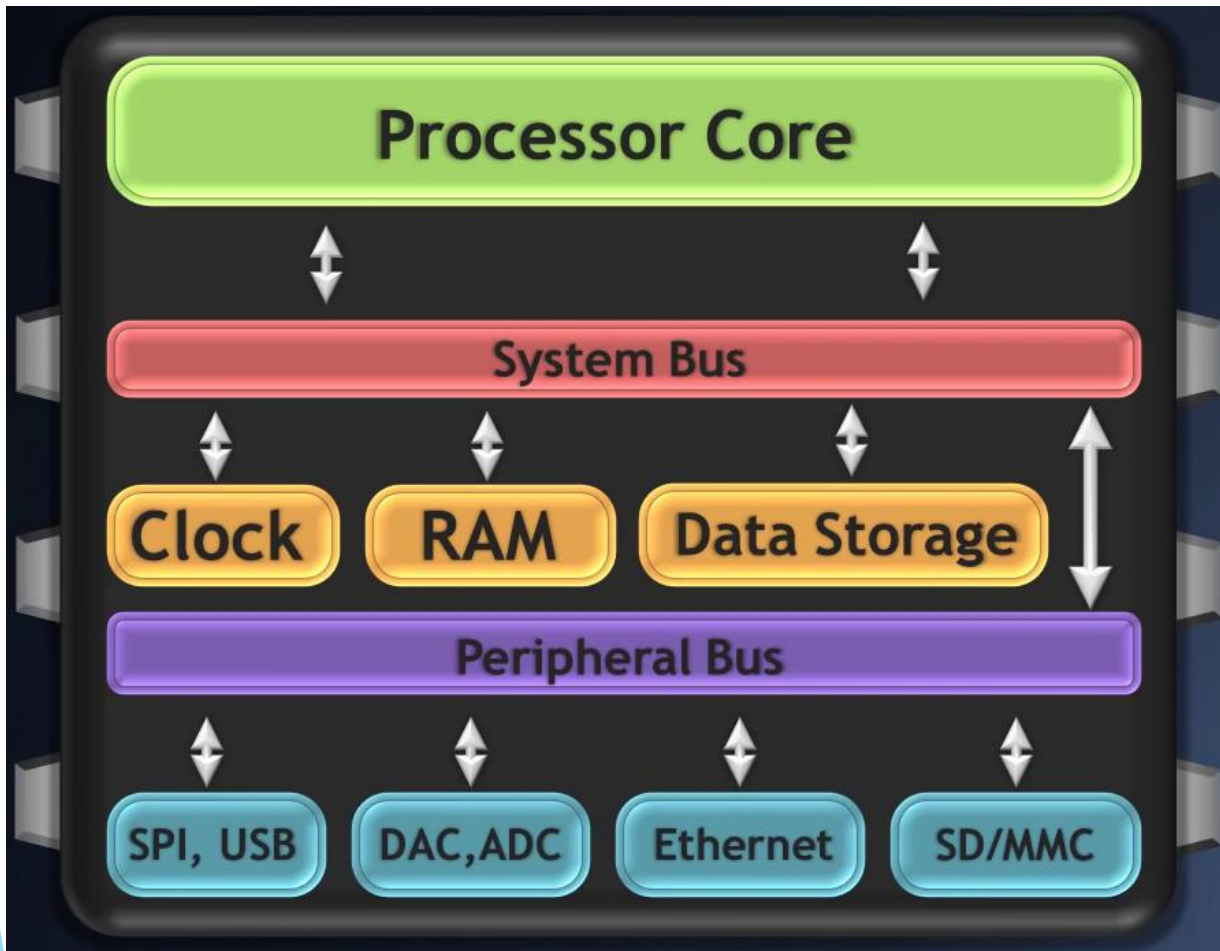
# ریزپردازنده

◀ ریزپردازنده بدون مادربرد و قطعات جانبی، قابل استفاده نیست



# میکروکنترلر

◀ میکروکنترلر، بسیاری از تراشه های مورد نیاز مثل RAM, ROM, Clock, ADC, I2C و مانند اینها را درون خود داشته و نیازی به مادربرد ندارد.

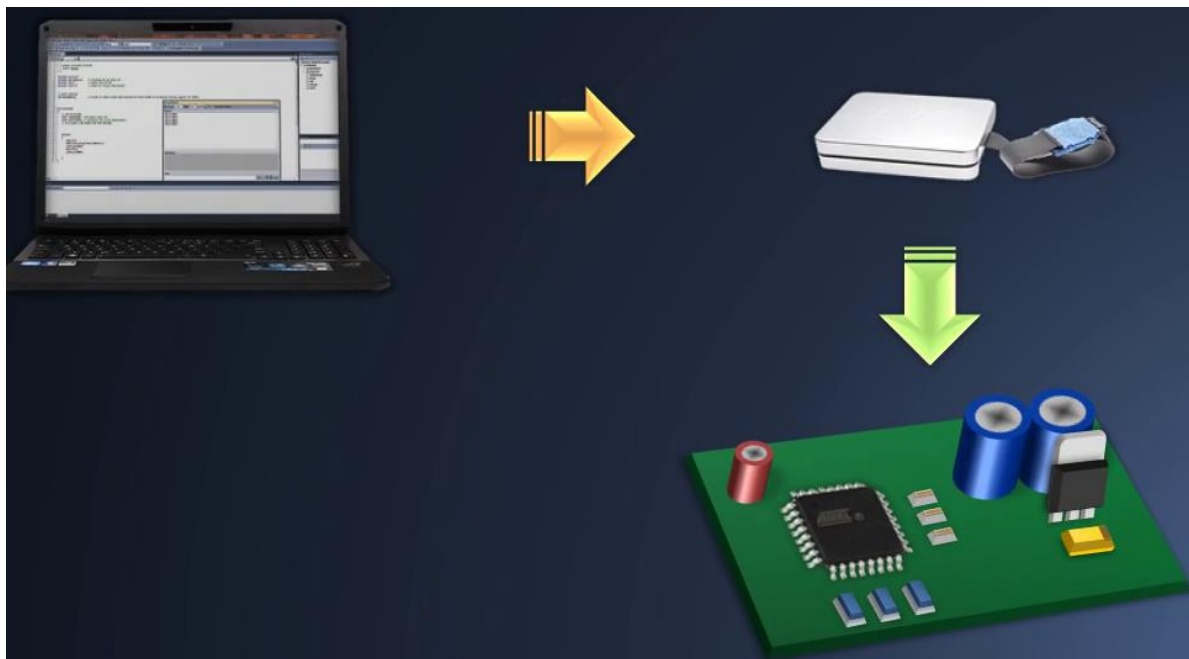


# اساس عملکرد میکروکنترلر

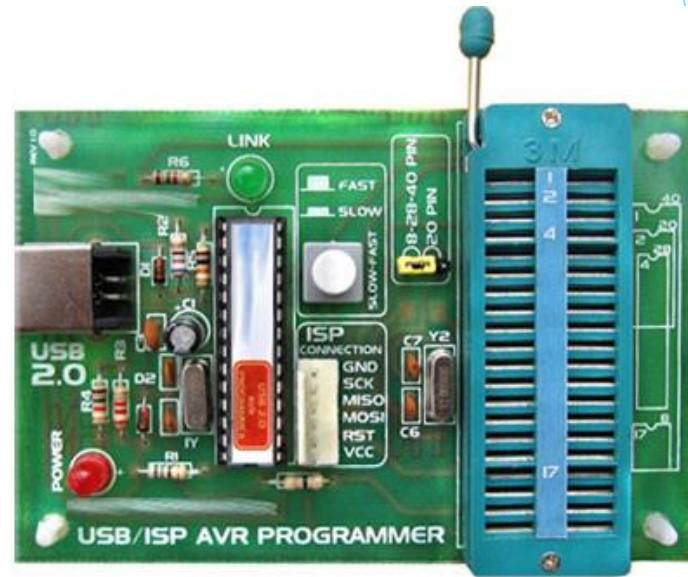
- ◀ میکروکنترلر برای کاربردهای خاص استفاده می شود
- ◀ نیاز نیست خیلی مشخصات قوی داشته باشند چرا که بسیاری از کاربردها سرعت و حافظه کمی نیاز دارند
- ◀ هوشمندی میکرو به کدی است که شما می نویسید
- ◀ شاید بتوان گفت ۸۰ درصد دانش میکروکنترلر کدنویسی است
- ◀ میکرو همانقدر مفید است که کد شما به آن دیکته می کند. اگر شما می خواهید با عبور دما از یک حد آستانه، چراغ قرمزی روشن شود، این کار باید از طریق کد بیان شود.

# برنامه نویسی میکرو

- ◀ کد لازم در یک محیط توسعه یکپارچه (IDE) به زبانهایی مثل C یا اسمبلی نوشته می شود
- ◀ بعد از خطایابی توسط کامپایلر و اصلاح خطاهای احتمالی، کد به فرمت باینری (کد ماشین یا کد هگز) مناسب برای میکرو تبدیل می شود
- ◀ در نهایت توسط دستگاه برنامه ریز (Programmer) کد هگز روی میکروکنترلر ریخته می شود. برخی بردها مثل آردینو برنامه ریز سرخود دارند و نیاز به دستگاه جانبی نیست



# دو نمونه برنامه ریز (AVR و آردینو)





# بسته بندی میکروکنترلر



## DIP

(Dual Inline Package)

Through hole

8 pins

9mm x 6mm

0.15pins/mm<sup>2</sup>



## SOIC

(Small Outline IC)

Surface Mount

18 pins

11mm x 7mm

0.23pins/mm<sup>2</sup>



## QFP

(Quad Flat Package)

Surface Mount

32 pins

7mm x 7mm

0.65pins/mm<sup>2</sup>



## BGA

(Ball Grid Array)

Surface Mount

100 pins

6mm x 6mm

2.78pins/mm<sup>2</sup>

# بسته بندی میکروکنترلر



## DIP

(Dual Inline Package)

Through hole

8 pins

9mm x 6mm

0.15pins/mm<sup>2</sup>



## SOIC

(Small Outline IC)

Surface Mount

18 pins

11mm x 7mm

0.23pins/mm<sup>2</sup>



## QFP

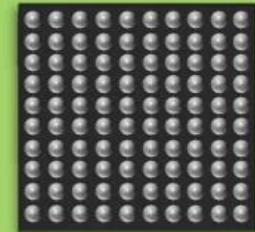
(Quad Flat Package)

Surface Mount

32 pins

7mm x 7mm

0.65pins/mm<sup>2</sup>



## BGA

(Ball Grid Array)

Surface Mount

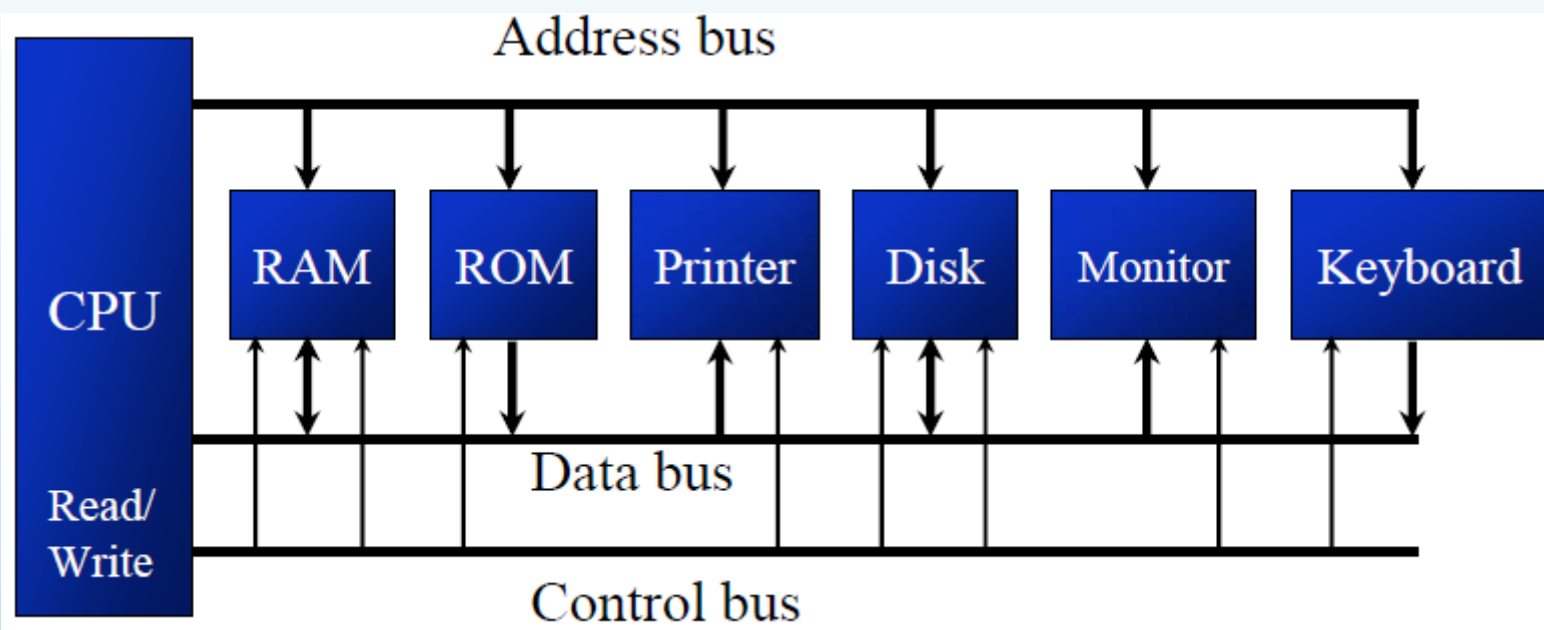
100 pins

6mm x 6mm

2.78pins/mm<sup>2</sup>

## □ ساختار ساده یک کامپیوتر

■ پردازنده مرکزی، حافظه‌های RAM و ROM، ابزارهای ورودی خروجی، گذرگاههای آدرس و داده



# انواع حافظه کد (ROM)

□ PROM یا OTP : تنها یک بار قابلیت نوشته شدن دارد.

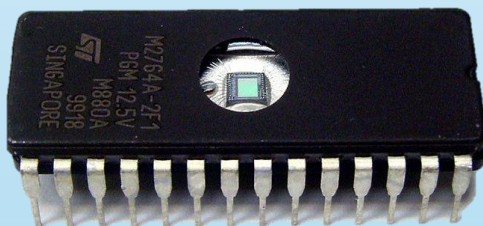
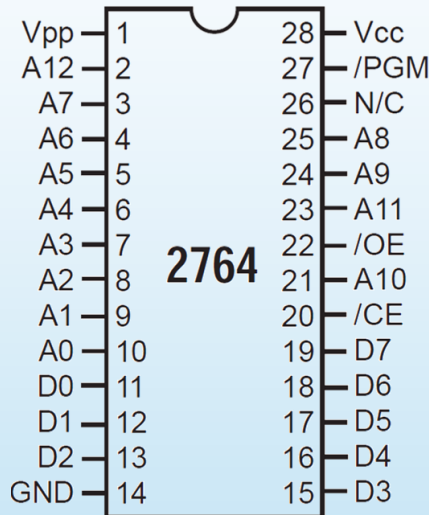
□ EPROM یا UV-EPROM

■ قابلیت پاک شدن دارد؛ ۲۰ دقیقه در مجاورت UV

■ نیاز به دو دستگاه programmer و پاک کننده

■ تعداد دفعات نوشتن و پاک کردن: هزار بار

■ مثال: ۲۷۶۴ (ظرفیت ۶۴ کیلوبایت)



□ EEPROM

■ قابلیت پاک شدن الکتریکی

■ پاک کردن بایتهای دلخواه

■ قابلیت برنامه ریزی روی برد (نیاز به مدار برنامه ریز)

■ تعداد دفعات نوشتن و پاک کردن: ۱۰۰ هزار بار

# انواع حافظه کد (ROM) - ادامه

## Flash

- EEPROM سریع
- پاک شدن کل حافظه در کمتر از یک ثانیه
- تعداد دفعات نوشتن و پاک کردن: ۱۰۰ هزار بار
- مورد استفاده در بایاس کامپیوتر و حافظه برنامه میکروهای AVR
- در آینده جایگزین دیسکهای سخت خواهد شد

## Mask ROM

- توسط تولید کننده IC برنامه ریزی می شود
- مناسب برای تولید انبوه
- پس از نهایی شدن کد، آن را به سازنده IC داده تا تولید کند.

 نکته: تمام حافظه‌های ROM، ۸ پایه برای داده دارند.



# انواع حافظه داده (RAM)

## حافظه ایستا یا SRAM

- از فلیپ فلاپ ساخته شده و نیازی به تازه سازی ندارد.
- حجم بیشتری روی تراشه اشغال می کند ( هر سلول ۴ تا ۶ ترانزیستور)

## حافظه پویا یا DRAM

- از خازن برای سلولهای حافظه استفاده می کند. **نیاز به تازه سازی دارد.**
- مزایا: ظرفیت بالا، هزینه کمتر و مصرف برق کمتر

## حافظه غیرفرار NV-RAM

- مزایای RAM و ROM را شامل می شود
- سلولهای SRAM با تکنولوژی CMOS (کم مصرف)
- باتری لیتیوم داخلی (قابلیت نگهداری اطلاعات تا ۱۰ سال)
- کنترل هوشمند جریان پین VCC جهت سوئیچ کردن بین منبع داخلی و خارجی

نکته: تعداد دفعات نوشتن و خواندن در RAM **بینهایت** است.

# نمونه ای از SRAM

## □ مثالی از حافظه ایستا: ۶۱۱۶

■ پینهای A0-A10 امکان آدرس دهی 2k خانه را فراهم می کنند.

■ پینهای D0-D7 بیانگر ۸ بیتی بودن خانه های حافظه است.

■ ساختار حافظه: 2kx8

■ (Write Enable) WE

□ برای نوشتن روی حافظه

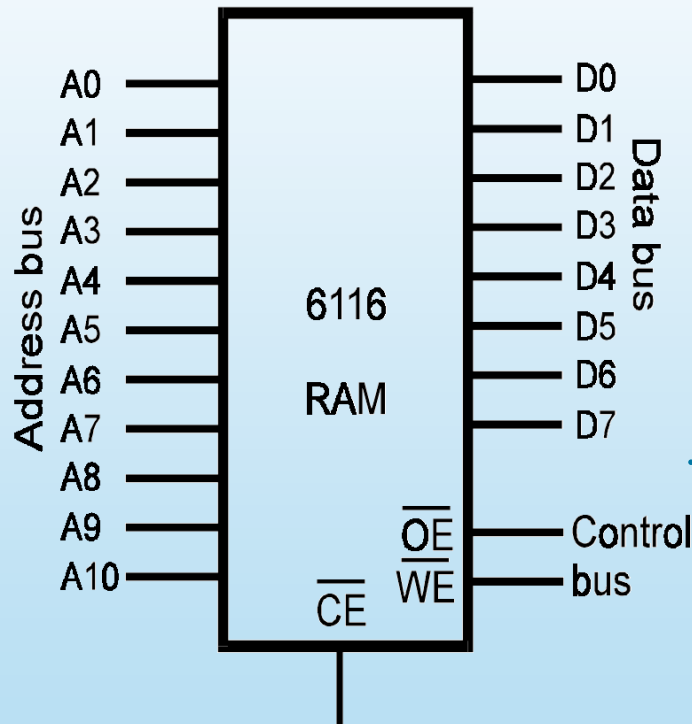
■ (Output Enable) OE

□ برای خواندن از حافظه

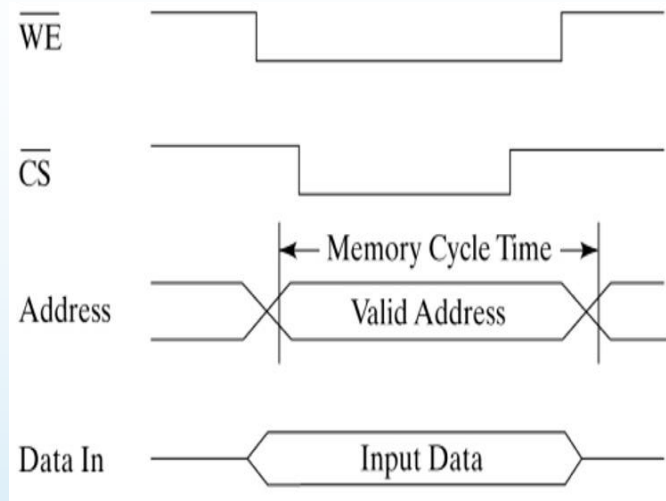
■ پینهای کنترلی WE و OE صفر-فعال هستند.

■ (Chip Select) CS

□ برای انتخاب تراشه



# نمودار زمانبندی نوشتن در SRAM - حالت ساده



نوشتن □

■ پایه های A0-A10 را فراهم کنید

■ پایه CS را فعال کنید

■ WE را فعال کنید

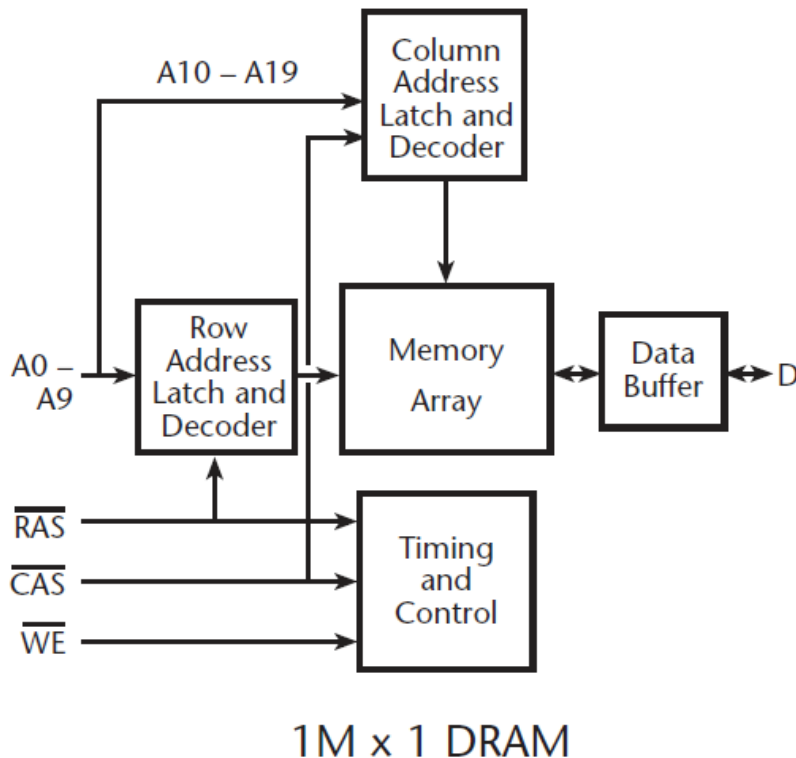
■ داده ها را در D0-D7 قرار دهید

## تعداد زیاد پینهای آدرس

■ استفاده از تکنیک چیدمان ماتریسی

■ دسترسی به خانه های حافظه با استفاده از RAS و CAS

■ تعداد پینهای لازم برابر است با نصف + ۲ پین برای RAS و CAS



# کد گشایی آدرس حافظه

- هر ابزار جانبی مانند حافظه‌های خارجی باید محدوده‌ی آدرس خاص خود را داشته باشند.
- مثلا اگر دو حافظه ROM خارجی با ساختار 2kx8 داریم می‌توانیم یکی را در محدوده‌ی آدرس 0X0000-0X07FF و دیگری را در محدوده‌ی آدرس 0X5000-0X57FF جاگذاری کنیم.
- به این عمل نگاشت حافظه (memory mapping) گویند.
- برای این کار پینهای کم ارزش گذرگاه آدرس را به خطوط آدرس حافظه ROM متصل کرده و پینهای باقیمانده را با استفاده از گیت‌های منطقی یا مالتی پلکسر به آدرس مورد علاقه نگاشت می‌دهیم.
- پینهای باقیمانده باید به CS یا CE حافظه وصل شوند.



□ پایه های کنترلی پردازنده، (RD و WR) به پایه های متناظر از حافظه متصل می شوند (OE و WE).

- RD: Read
- WR: Write
- OE: Output Enable
- WE: Write Enable

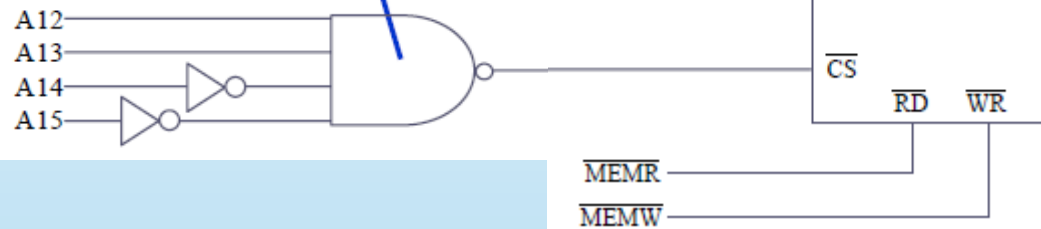
□ دو روش متداول برای کدگشایی:

- استفاده از گیت NAND
- استفاده از دیکدر (مثل 74LS138)

# کدگشایی با استفاده از NAND

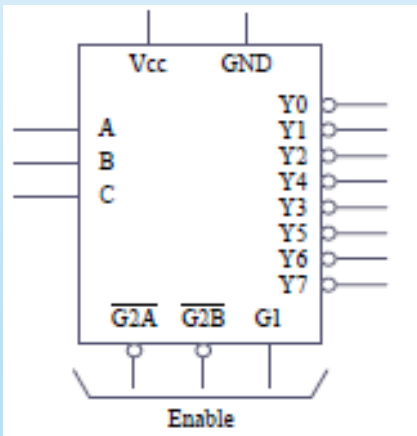
□ خروجی NAND صفر فعال است و پایه های CS یا CE هم غالباً صفر فعال هستند، لذا استفاده از NAND بسیار ساده است:

A15-A12 must be 0011 in order to select the chip  
This result in the assignment of address 3000H to 3FFFH to this memory chip



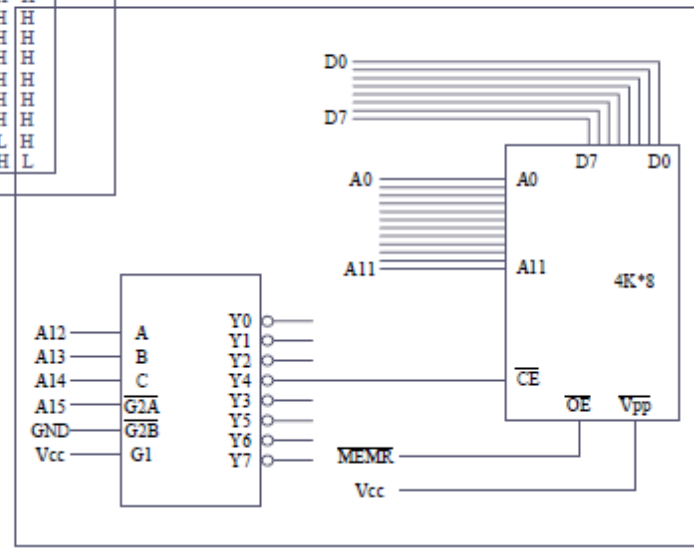
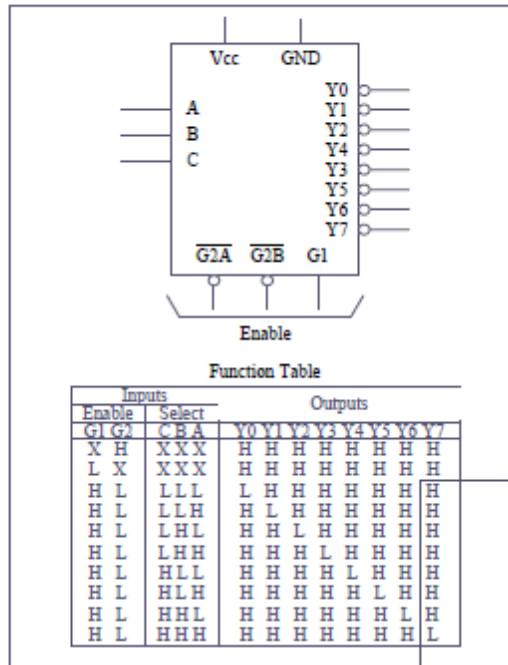
# کدگشایی با استفاده از 74LS138

- ☐ یک دیکدر ۳ به ۸
- ☐ ورودیها:  $A, B, C$  و خروجی  $Y_0 - Y_7$  صفر فعال
- ☐ هر خروجی را می توان به یک حافظه جداگانه متصل کرد و به این ترتیب به ۸ حافظه تخصیص آدرس داد.
- ☐ سه ورودی اضافی  $G_2A, G_2B$  و  $G_1$  هم، خود دیکدر را فعال یا غیرفعال می کنند.
- ☐ این سه می توانند همواره فعال باشند (با اتصال به زمین یا  $V_{CC}$ ) یا در صورت نیاز به خطوط آدرس وصل شوند.



# کدگشایی با 74LS138

## 74LS138 Decoder



□ در شکل قبل، فضای آدرس مربوط به  $Y_4, Y_2$  و  $Y_7$  را بیابید

(a) The address range for  $Y_4$  is calculated as follows.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1

The above shows that the range for  $Y_4$  is 4000H to 4FFFH. In Figure 14-6, notice that  $A_{15}$  must be 0 for the decoder to be activated.  $Y_4$  will be selected when  $A_{14} A_{13} A_{12} = 100$  (4 in binary). The remaining  $A_{11}-A_0$  will be 0 for the lowest address and 1 for the highest address.

(b) The address range for  $Y_2$  is 2000H to 2FFFH.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

(c) The address range for  $Y_7$  is 7000H to 7FFFH.

A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1



# طرز کار پردازنده

---

## INSIDE THE COMPUTER

### Internal Organization of Computers (cont')

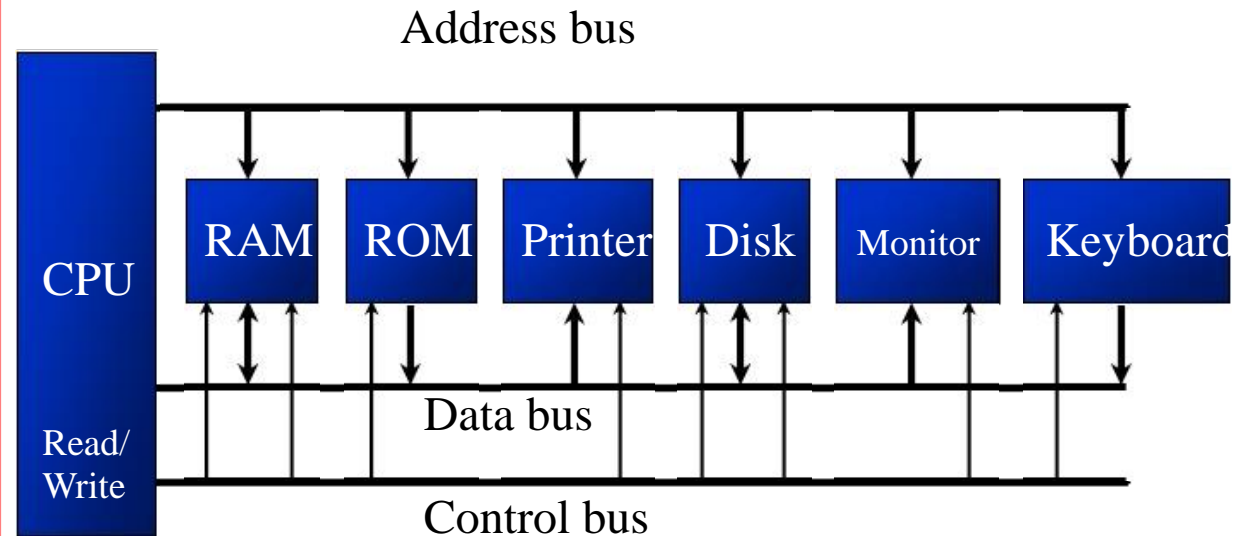
The CPU is connected to memory and I/O through strips of wire called a bus

Carries information from place to place

Address bus

Data bus

Control bus



## INSIDE THE COMPUTER

### Internal Organization of Computers (cont')

#### Address bus

For a device (memory or I/O) to be recognized by the CPU, it must be assigned an address

The address assigned to a given device must be unique

The CPU puts the address on the address bus, and the decoding circuitry finds the device

#### Data bus

The CPU either gets data from the device or sends data to it

#### Control bus

Provides read or write signals to the device to indicate if the CPU is asking for information or sending it information

## INSIDE THE COMPUTER

### More about Data Bus

The more data buses available, the better the CPU

Think of data buses as highway lanes

More data buses mean a more expensive CPU and computer

The average size of data buses in CPUs varies between 8 and 64

Data buses are bidirectional

To receive or send data

The processing power of a computer is related to the size of its buses

## INSIDE THE COMPUTER

### More about Address Bus

The more **address buses** available, the larger the number of devices that can be addressed

The number of locations with which a CPU can communicate is always equal to  $2^x$ , **where x is the address lines**, regardless of the size of the data bus

ex. a CPU with **24 address** lines and 8 data lines can provide a total of  $2^{24}$  or 16M byte of addressable memory

Each location can have a maximum of 1 byte of data, since all general-purpose CPUs are byte addressable

The address bus is unidirectional

## INSIDE THE COMPUTER

### CPU's Relation to RAM and ROM

For the CPU to process information, the data must be stored in RAM or ROM, which are referred to as primary memory

ROM provides information that is fixed and permanent

- Tables or initialization program

RAM stores information that is not permanent and can change with time

- Various versions of OS and application packages

- CPU gets information to be processed

  - first from RAM (or ROM)

  - if it is not there, then seeks it from a mass storage device, called secondary memory, and transfers the information to RAM



# INSIDE THE COMPUTER

## Inside CPUs

### Registers

The CPU uses registers to store information temporarily

- Values to be processed

- Address of value to be fetched from memory

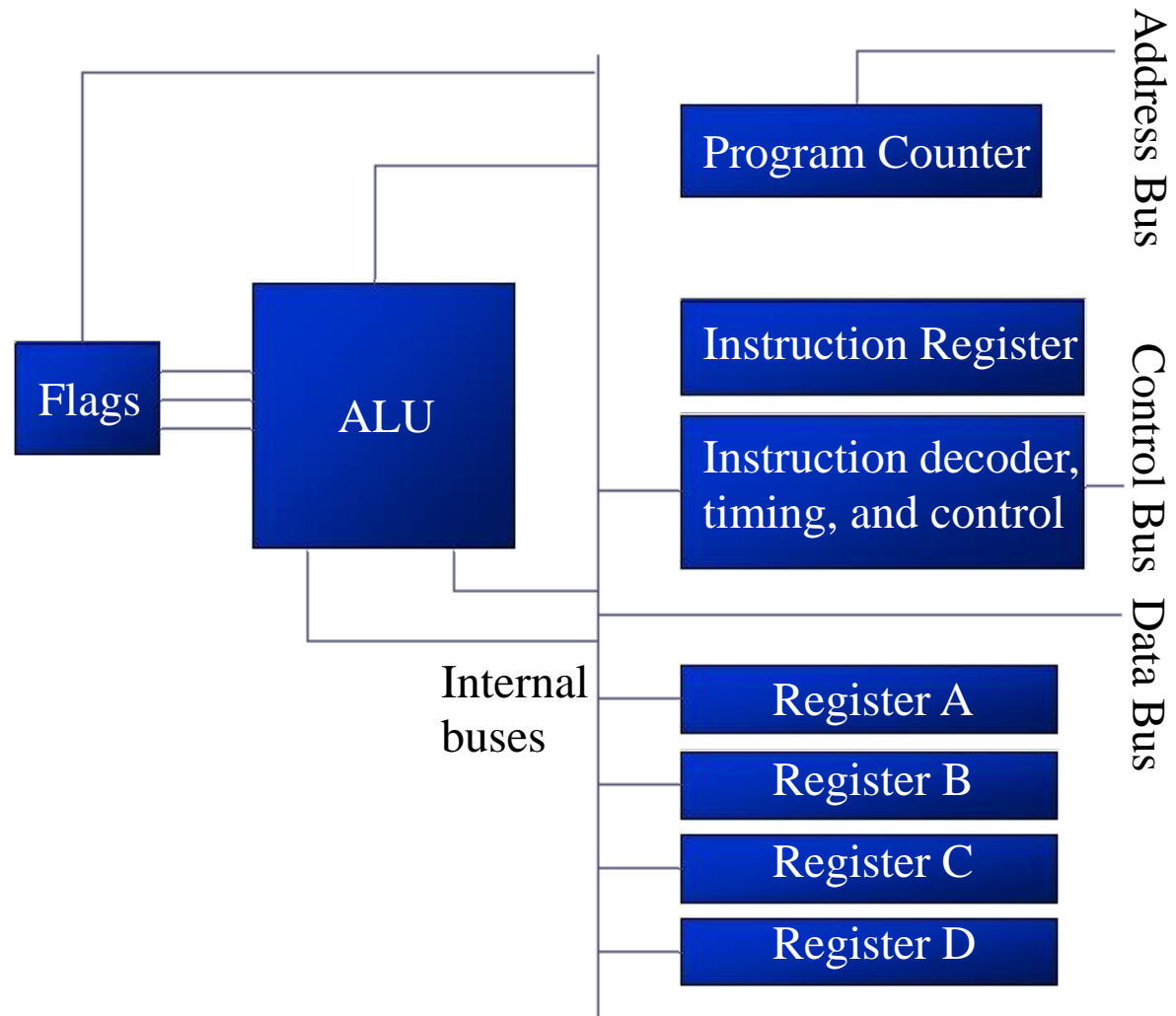
In general, the more and bigger the registers, the better the CPU

- Registers can be 8-, 16-, 32-, or 64-bit

- The disadvantage of more and bigger registers is the increased cost of such a CPU

# INSIDE THE COMPUTER

## Inside CPUs (cont')



## INSIDE THE COMPUTER

### Inside CPUs (cont')

#### ALU (arithmetic/logic unit)

Performs arithmetic functions such as add, subtract, multiply, and divide, and logic functions such as AND, OR, and NOT

#### Program counter

Points to the address of the next instruction to be executed

As each instruction is executed, the program counter is incremented to point to the address of the next instruction to be executed

#### Instruction decoder

Interprets the instruction fetched into the CPU

A CPU capable of understanding more instructions requires more transistors to design

## INSIDE THE COMPUTER

### Internal Working of Computers

Ex. A CPU has registers A, B, C, and D and it has an 8-bit data bus and a 16-bit address bus. The CPU can access memory from addresses 0000 to FFFFH

Assume that the code for the CPU to **move** a value to register A is **B0H** and the code for adding a value to register A is **04H**

The action to be performed by the CPU is to put 21H into register A, and then add to register A values 42H and 12H

...

## INSIDE THE COMPUTER

### Internal Working of Computers (cont')

Ex. (cont')

Action	Code	Data
Move value 21H into reg. A	B0H	21H
Add value 42H to reg. A	04H	42H
Add value 12H to reg. A	04H	12H

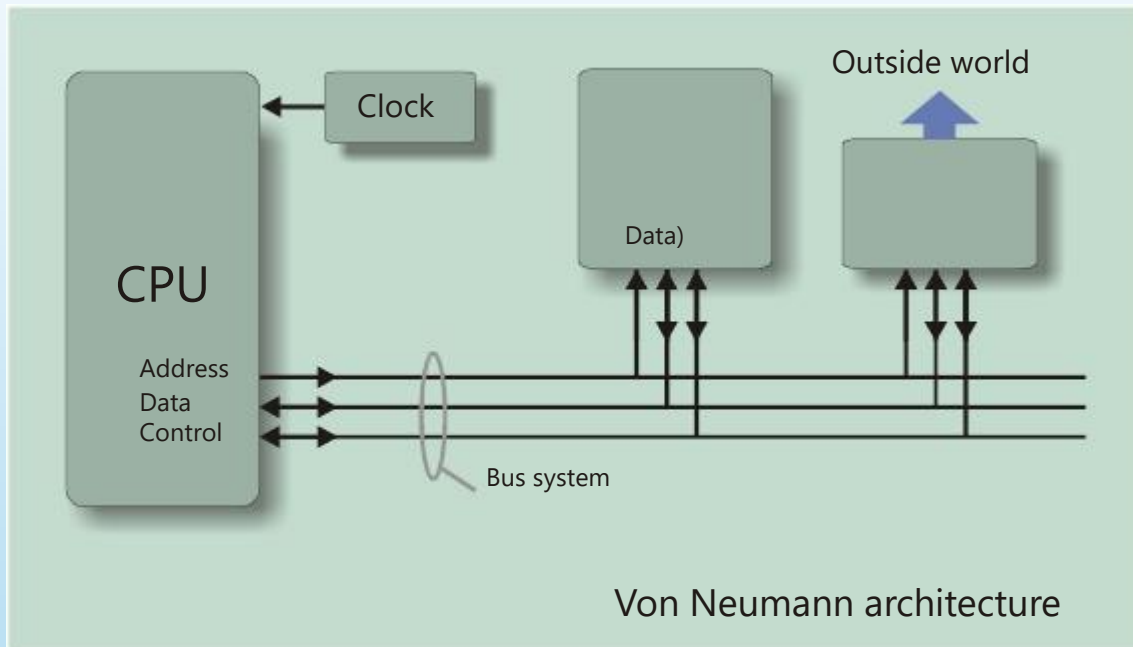
Mem. addr.	Contents of memory address
1400	(B0) code for moving a value to register A
1401	(21) value to be moved
1402	(04) code for adding a value to register A
1403	(42) value to be added
1404	(04) code for adding a value to register A
1405	(12) value to be added
1406	(F4) code for halt

...

# معماری ون نیومن و هاروارد

## معماری ون نیومن □

- گذرگاه داده و آدرس مشترک برای داده‌ها و کد
- مورد استفاده در اکثر کامپیوترهای امروزی

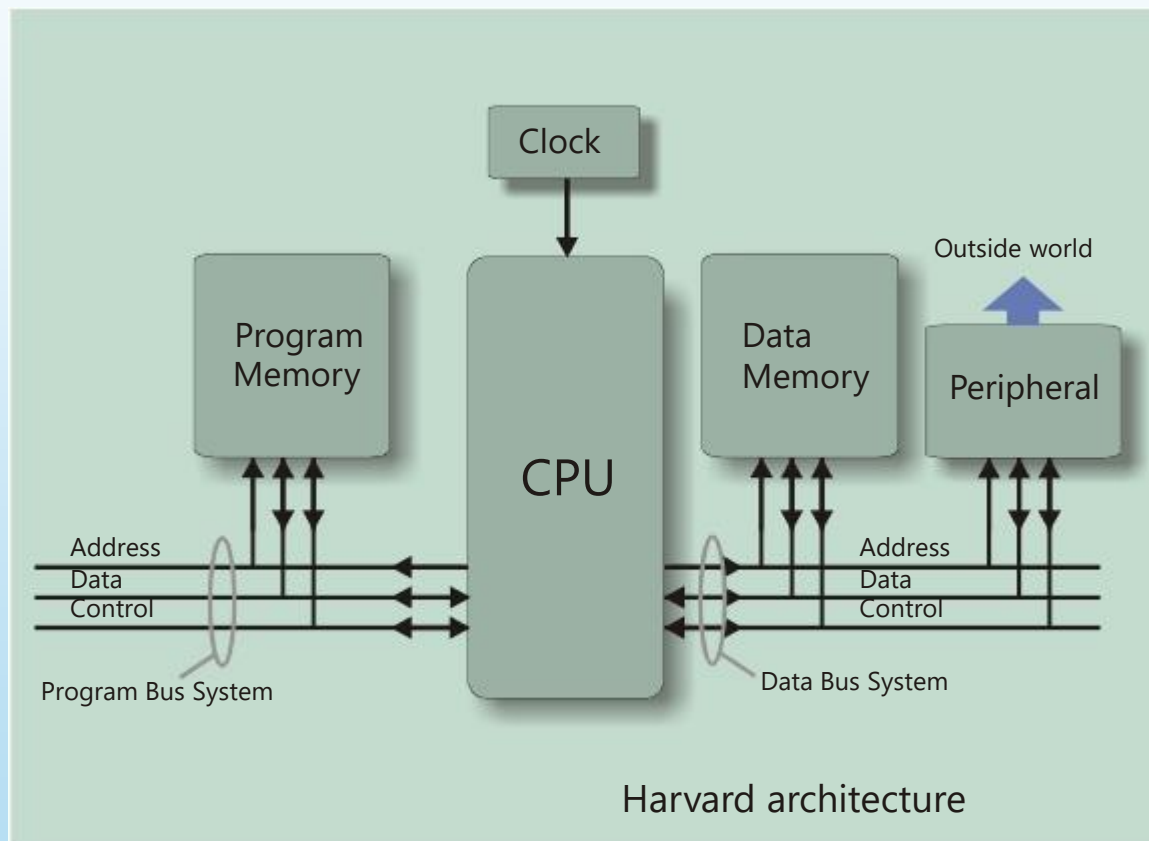




## □ دو گذرگاه داده و دو گذرگاه آدرس

■ یکی برای داده ها و ابزارهای جانبی

■ دیگری برای کد



## □ فصل صفرم

■ ۱، ۲، ۳، ۴، ۷، ۱۰، ۱۱، ۱۴، ۲۱، ۲۲، ۲۳، ۲۶، ۳۱، ۳۵، ۳۷، ۴۲، ۴۴، ۴۶، ۵۰

## □ فصل اول

■ ۱، ۹، ۱۱، ۱۲، ۱۵، ۱۶، ۱۷، ۱۸، ۲۶، ۲۹

## □ موعد تحویل: ۷ روز آینده