



دانشکده مهندسی کامپیوتر  
دانشگاه صنعتی امیرکبیر

## فصل دوم

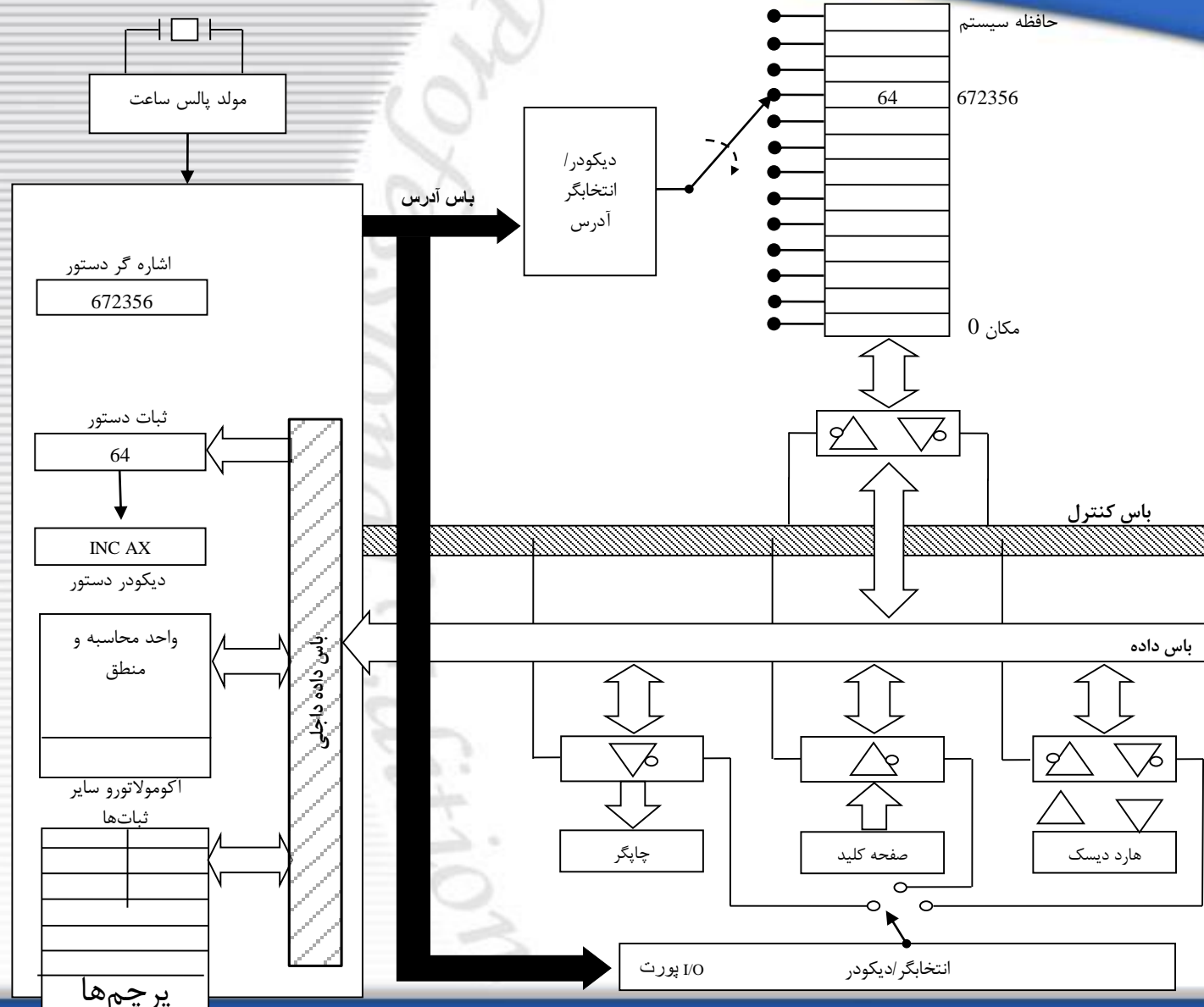
معماری اجمالی یک سیستم فرضی مبتنی بر ریزپردازنده،

## سرفصل مطالب

---

- بلوک دیاگرام یک ریزپردازنده فرضی
- واکنشی و اجرای دستورالعمل
- پرچم‌ها
- انواع سیکل‌های باس (سیکل ماشین)

# بلوک دیاگرام یک ریزپردازنده فرضی



شکل ۱

## بعضی از پرچم‌ها در ۸۰۸۶

نام پرچم	عملکرد
CF	پرچم <b>Carry</b> : اگر بر بیت پر ارزش نتیجه، <b>Carry</b> یا <b>Borrow</b> اتفاق افتد، این پرچم 1 شده و در غیر اینصورت 0 خواهد بود.
PF	پرچم <b>Parity</b> : این پرچم 1 می شود اگر تعداد بیت‌های 1 در بیت‌های نتیجه، زوج باشد. در غیر اینصورت 0 می شود.
AF	اگر از چهار بیت کم ارزش <b>AL</b> ، <b>Carry</b> یا <b>Borrow</b> اتفاق افتد، مقدار این پرچم برابر 1 و گرنه 0 می شود.
ZF	پرچم <b>Zero</b> : اگر نتیجه صفر باشد، این پرچم 1 و گرنه 0 می شود
SF	پرچم <b>Sign</b> : این پرچم مقدار پرارزش ترین بیت نتیجه را می گیرد. (بیت علامت)
IF	پرچم <b>Interrupt-enable</b> : وقتی این پرچم 1 شود، وقفه های قابل <b>mask</b> شدن باعث می شوند که <b>CPU</b> ، کنترل برنامه را به مکان بردار وقفه منتقل کند.

## انواع سیکل‌های باس (سیکل ماشین)

اجرای یک دستورالعمل با اجرای یک یا چند سیکل باس صورت می‌گیرد.

### انواع سیکل‌های باس

- سیکل خواندن از حافظه (MR)
- سیکل نوشتن در حافظه (MW)
- سیکل خواندن از I/O (IOR)
- سیکل نوشتن در I/O (IOW)
- سیکل باس بیکار (Idle) (عملیات داخل CPU انجام می‌شود که نیازی به دسترسی به باس ندارد)

INC AX

Bus cycles: MR (Opcode read)

MOV AX, [SI] ;  $AX \leftarrow [SI]$  Machine code: 8B 04

Bus cycles: MR (Opcode read), MR (Reading the content of SI location in memory)

MOV [SI], AX ;  $[SI] \leftarrow AX$

Bus cycles: MR (Opcode read), MW (Writing AX in the SI memory location)

IN AH, 50H ;  $AH \leftarrow \text{Content of Port 50H}$

Bus cycles: MR (Opcode read), IOR (Reading the content of port No. 50H)

OUT 50H, AH ;  $\text{Port 50H} \leftarrow AH$

Bus cycles: MR (Opcode read), IOW (Writing AH in the port No. 50H)

INC BYTE PTR[SI]

# فرمت یک دستورالعمل در زبان اسمبلی

Label:                      Instruction operands    ;Comments

Instruction: کد حفظی (mnemonic) دستورالعمل مورد نظر. مثل ADD, INC, NOP و مانند آن

Operands: عملوندها که می‌تواند هیچ، یک یا دو عملوند که می‌تواند یک داده بلافاصله، یک ثبات، آدرس داده مورد نظر در حافظه، یک آدرس در حافظه، شماره یک پورت و مانند آن باشد.

Comments: توضیحات دستورالعمل که باید بعد از ; بیاید.

شبه دستورالعمل‌ها مانند DB, EQU, ORG و ....

مثال:

```
ORG 1000H
EQU Value 10H
MOV AH, 23H; ; Fill AX with and Immediate value;
MOV CX, Value; ; CX is used as a counter
LoopBegin: NOP; ; No operation, An Instruction with no operand
MOV AX, [SI]; ; Transfer data from address in memory indicated by [SI] to AX
INC SI; ; Increment SI
OUT 50H, AX; ; Transfer AX to output port (port address: 50H)
DEC CX; ; Decrement CX
JNZ LoopBegin; ; Jump if Z flag is not set
```

## معرفی چند شبه دستورالعمل

توضیحات	شبه دستورالعمل
مشخص کننده آدرس قرارگیری کد در حافظه	ORG Address
یک شناسه را به یک مقدار ثابت منتسب می کند	EQU
رزرو فضا در حافظه برای متغیرها و مقداردهی آنها	DB, DW, DD, DQ, DT
مدل حافظه: اندازه حافظه برنامه و داده	.model
مشخص کردن سگمنت مورد نظر	.data .code .extra .stack
مشخص کردن حجم پشته	.stack size

مثال:

.model memory\_model

انواع memory\_model

tiny: code+data <= 64K (.com program)

small: code<=64K, data<=64K, one of each

medium: data<=64K, one data segment

compact: code<=64K, one code segment

large: multiple code and data segments

huge: allows individual arrays to exceed 64K

flat: no segments, 32-bit addresses, protected mode only (80386 and higher)

## معرفی چند شبه دستورالعمل (ادامه)

مثال:

;Note: EQUated symbols are not variables

Count EQU 10

Element EQU 5

MyString EQU "Maze of twisty passages"

مثال

; Names can be associated with storage locations, These names are called variables.

; DB, DW, DD, DQ and DT are used for reserving space in memory for variables

; Word, doubleword, and quadword data are stored in reverse byte order (in memory)

.data

DB 255,?, -128,'X' ; ? represents an uninitialized storage location

numRows DB 25

videoBase DW 0800h

MyWord DW 256 ; 00 01

MyDoubleWord DD 1234567h ; 67 45 23 01

MyQuadWord DQ 10 ; 0A 00 00 00 00 00 00 00

aTOM DB "ABCDEFGHJKLMN" ; Ascii of characters are used



## معرفی چند شبه دستورالعمل (ادامه)

---

مثال:

```
A Num DB -4  
      DW 17  
ONE  
UNO DW 1  
X    DD ?
```

The above names are variables:

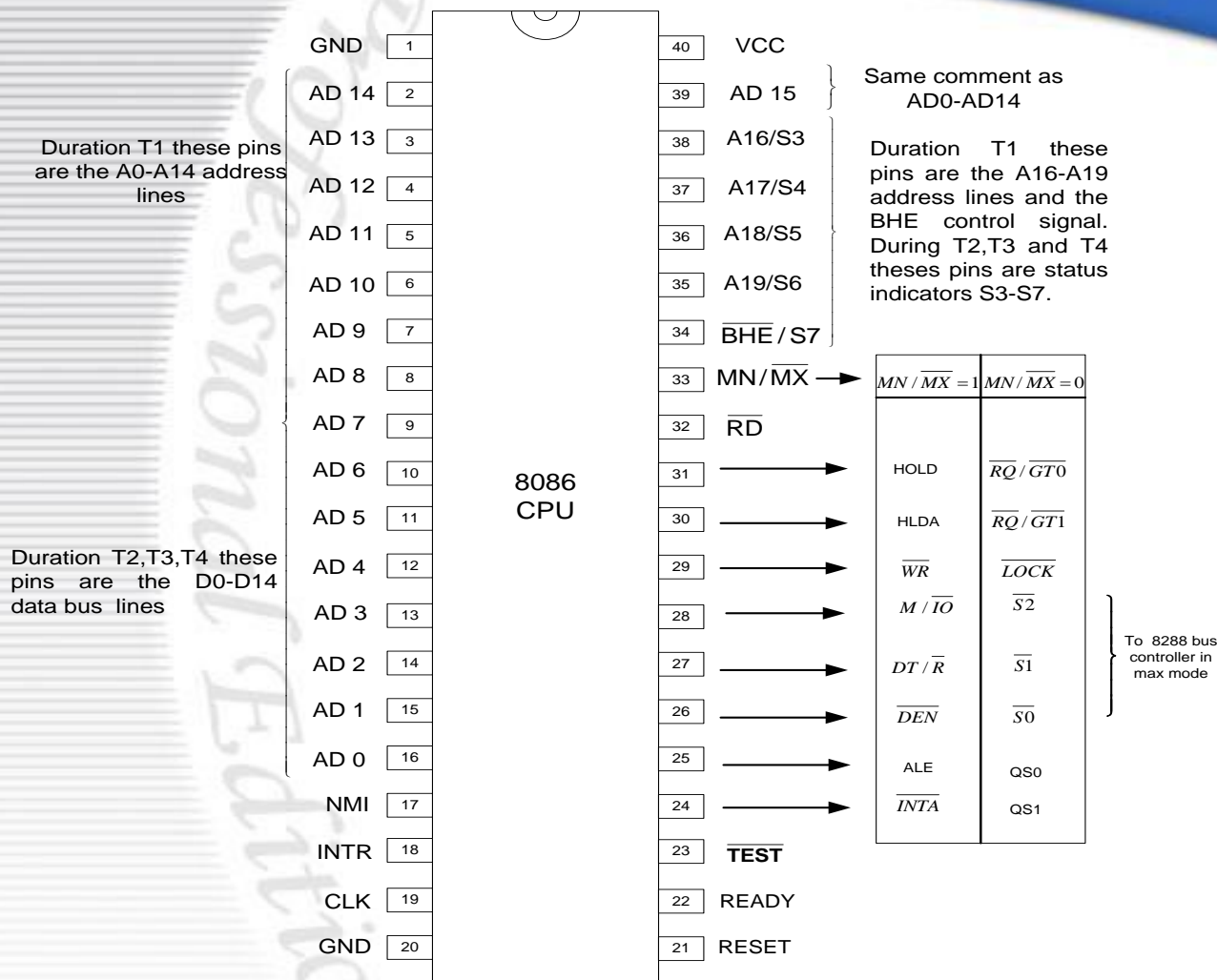
A Num refers to a byte storage location, initialized to FCh=-4D

The next word has no associated name

ONE and UNO refer to the same word

X is an uninitialized doubleword

# نمونه‌ای از پایه‌های یک ریزپردازنده (۸۰۸۶)



شکل ۴