



دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیرکبیر

فصل دوم

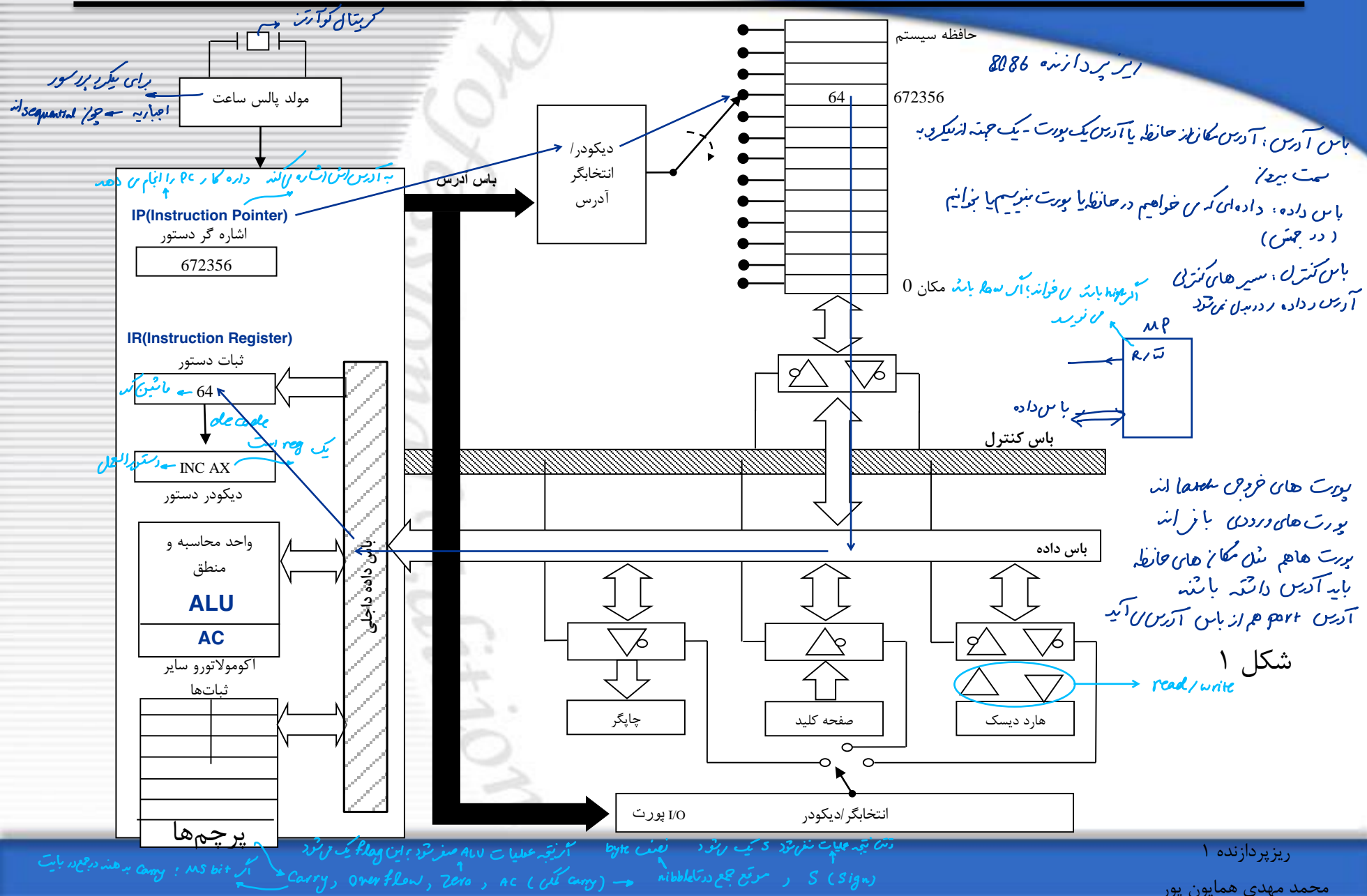
معماری اجمالی یک سیستم فرضی مبتنی بر ریزپردازنده،

ریزپردازنده ۱
محمد مهدی همایون پور

سرفصل مطالب

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

بلوک دیاگرام یک ریزپردازنده فرضی



بعضی از پرچم‌ها در ۸۰۸۶

نام پرچم	عملکرد
CF	پرچم Carry : اگر بر بیت پر ارزش نتیجه، Carry یا Borrow اتفاق افتد، این پرچم 1 شده و در غیر اینصورت 0 خواهد بود.
PF	پرچم Parity : 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 AL ، Carry یا Borrow اتفاق افتد، مقدار این پرچم برابر 1 و گرنه 0 می شود.
ZF	پرچم Zero : اگر نتیجه صفر باشد، این پرچم 1 و گرنه 0 می شود
SF	پرچم Sign : این پرچم مقدار پرارزش ترین بیت نتیجه را می گیرد. (بیت علامت)
IF	پرچم Interrupt-enable : وقتی این پرچم 1 شود، وقفه های قابل mask شدن باعث می شوند که CPU ، 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)

MOV AX, [SI]



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^{Hex} ; $AH \leftarrow \text{Content of Port 50H}$

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

OUT 50H, AH ; Port 50H $\leftarrow AH$
MR (Reading 50H as an address)

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

INC BYTE PTR[SI]
MR (Reading 50H as an address)

تمام دستورات یک سیکل memory Read برای خواندن Op code دارند

فرمت یک دستورالعمل در زبان اسمبلی

Label: Instruction operands ;Comments

Instruction: کد حفظی (mnemonic) دستورالعمل مورد نظر. مثل ADD, INC, NOP و مانند آن *برای این ترانس دهیم که بتوانیم به آن jump کنیم*

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      تعریف یک مقدار Constant بنام Value به مقدار 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 / Register indirect addressing mode
INC SI;              ; Increment SI
OUT 50H, AX          ; Transfer AX to output port (port address: 50H) / Direct addressing
DEC CX               ; Decrement CX
JNZ LoopBegin        ; Jump if Z flag is not set
که تولید تأخیر
(هیچ کار دیگری انجام نمی‌دهد)
دستور تبدیل که عملگر در تغییر داده
flag Zero در تغییر می‌دهد
Jump non zero
```

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 <i>Define Double word</i>
رزرو فضا در حافظه برای متغیرها و مقداردهی آنها	DB, DW, <i>byte</i> DD, <i>word</i> 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

(در درس کاری با هانس نداریم)

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 دستور العمل (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)

یک word؛ دو byte است

.data

DB 255,?, -128,'X'

; ? represents an uninitialized storage location

که ASCII را در آن قرار گیرد

Constant نیست در همین برنامه عوض لا شود (متغیر) اسم داریم پس

numRows DB 25

videoBase DW 0800h

MyWord DW 256

MyDoubleWord DD 1234567h

MyQuadWord DQ 10

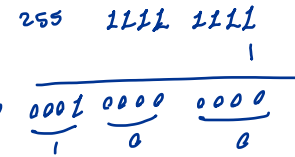
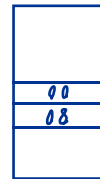
aTOM DB "ABCDEFGHJKLM"

; 00 01

; 67 45 23 01

; 0A 00 00 00 00 00 00 00

; Ascii of characters are used



00 00 00 00 00 00 00 0A

ترتیب ذخیره شدن حرف از چپ به راست

A
B
...

یعنی 13 حرف ← 13 بایت و 13 ASCII ذخیره می شود کاراکترها در یک بایت ذخیره می شود

معرفی چند شبه دستورالعمل (ادامه)

مثال:

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

باینام اختصاص نداریم
دو اسم به آن اختصاص داریم
مقداری باین اختصاص نداریم

The above names are variables:

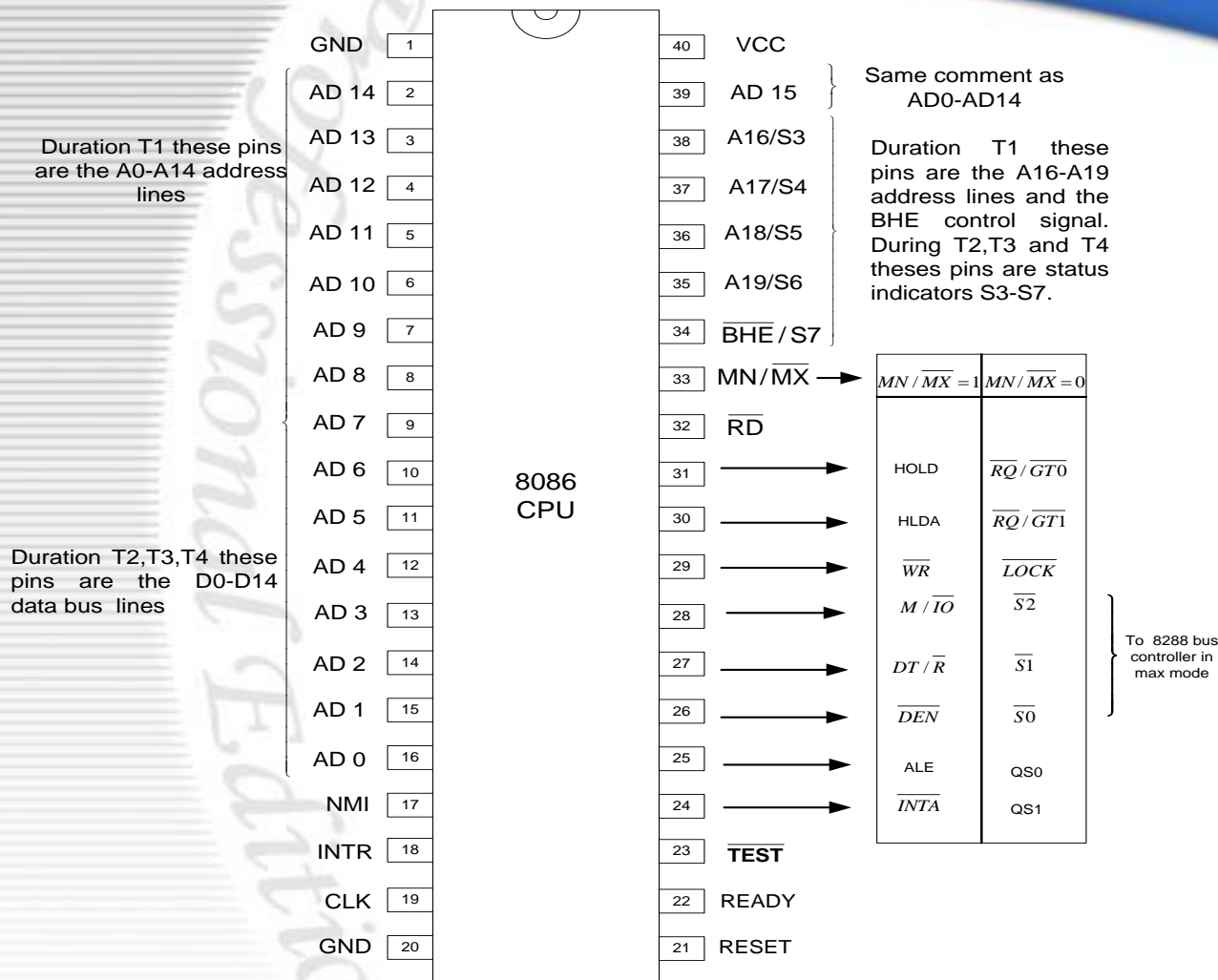
A Num refers to a byte storage location, initialized to FCh=-4D

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