

# زبان ماشین و برنامه سازی سیستم

## فصل چهارم : ساختار اصولی کامپیوترهای شخصی

مدرس:

سید محمد بیدکی



دانشگاه خلیج فارس

دانشکده فنی و مهندسی جم

تابستان ۱۳۹۳

# عناوین

---

- I. ساختار داخلی پروسور
- II. حافظه
- III. ثبات های پروسور
- IV. نحوه قرار گرفتن سگمنت های برنامه در حافظه
- V. افست سگمنت
- VI. حافظه پشته

## مقدمه

- کامپیوتر های شخصی دارای پروسسور از خانواده ۸۶×۸۰ شرکت اینتل هستند.
- پروسسورهای اولیه دارای ثبات ها و قابلیت آدرس دهی ۱۶ بیتی بودند.
- ولی پروسسورهای جدیدتر پنتیوم ۸۰۵۸۶ و پنتیوم پرو دارای ثبات های ۳۲ بیتی هستند. که می توان از آن به صورت ۸ بیتی، ۱۶ بیتی یا ۳۲ بیتی متناسب با نیاز استفاده کرد.
- روز به روز به سرعت، ظرفیت حافظه، ظرفیت ثبات ها و امکانات سخت افزاری و نرم افزاری پروسسورها اضافه می شود و در نتیجه کار آیی پروسسورها بالاتر می رود.

# ساختار داخلی پروسسور

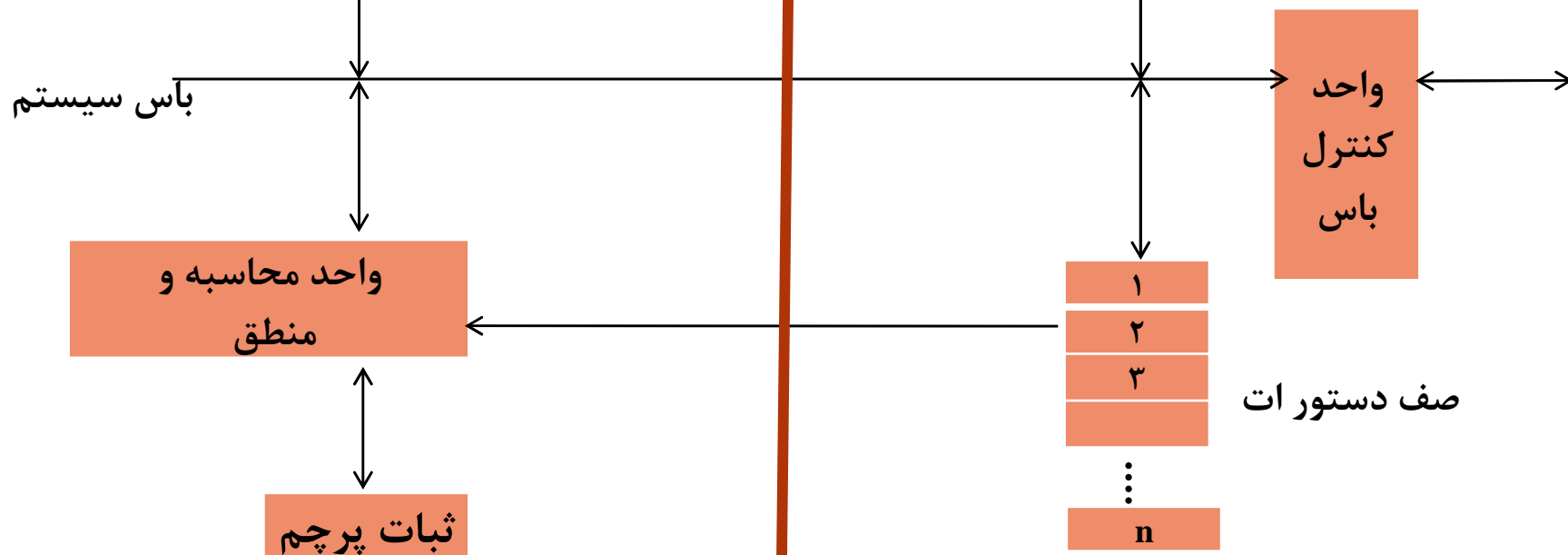
- پروسسور یا CPU کامپیوترهای شخصی از دو قسمت واحد اجرایی EU و واحد واسط باس BIU تشکیل شده است.
- واحد اجرایی EU مسئول اجرای دستورات است که از یک واحد محاسبه و منطقی ALU و تعدادی ثبات تشکیل شده است.
- واحد واسط باس BIU شامل واحد مدیریت کنترل باس، ثبات های سگمنت و صف دستورات می باشد.
- یکی از مهمترین کارهای واحد واسط باس BIU، پیش خوانی (prefetch) دستورات از حافظه و قرار دادن آنها در صف دستورات است.
- زمانی که واحد اجرایی یک دستور را اجرا می کند، واحد واسط باس BIU دستور بعدی را از حافظه می خواند و در صف دستورات قرار می دهد.
- هر لحظه واحد اجرایی بخواهد دستور بعدی را اجرا کند بلافاصله از صف دستورات، دستور را می گیرد و منتظر خواندن دستور از حافظه نمی شود.
- واحد اجرایی و واسط باس به صورت موازی با هم کار می کنند در نتیجه سرعت CPU بالا می رود.

## واحد EU

AH	AL
BH	BL
CH	BL
DH	DL
SP	
BP	
SI	
DI	

## واحد واسطه باس BIU

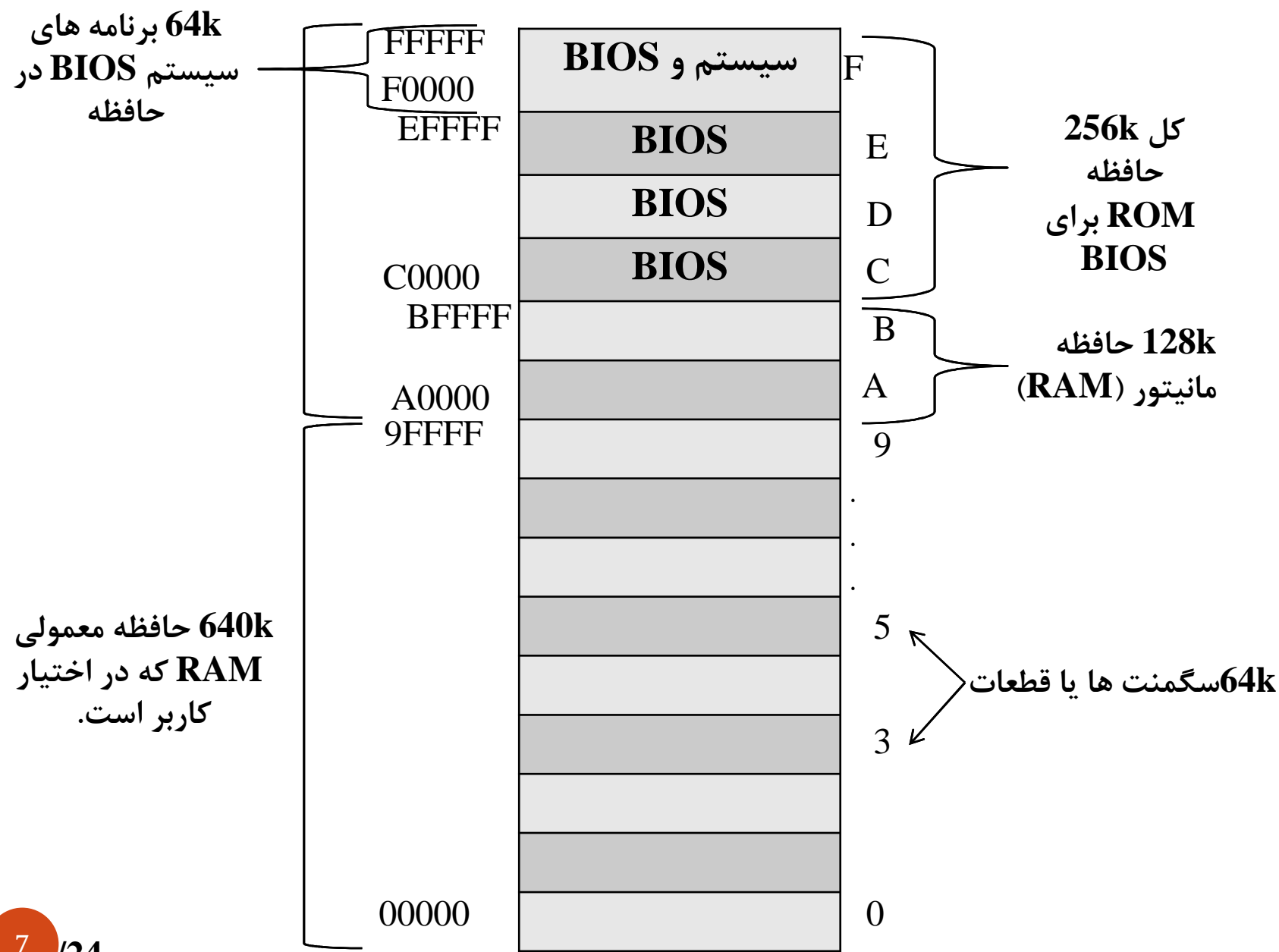
IP
CS
DS
SS
ES



# حافظه

- در حافظه کامپیوترهای شخصی، ۶۴۰ کیلو بایت اول، یعنی از آدرس 00000 تا آدرس 9FFFFH، حافظه RAM می باشد که در اختیار برنامه کاربر است.
- از آدرس A0000H تا BFFFFH (128 K) برای عملیات مانیتور می باشد.
- از آدرس C0000H تا EFFFFH (192 K) به حافظه ROM اختصاص دارد و برای روتین های ورودی خروجی پایه BIOS بکار می رود. (جهت سرویس دستگاه های ورودی خروجی)
- از آدرس F0000H تا FFFFFH (64 K) برای برنامه های سیستم، تست کردن قسمت های مختلف کامپیوتر، روتین های BIOS و برنامه راه اندازی و ... از آن استفاده می شود.

# نقشه یک مگابایت اول حافظه



# نوشتن اطلاعات در حافظه

- اگر اطلاعاتی که در داخل حافظه باید ذخیره شود ۱۶ بیتی باشد، در این صورت بایت کم ارزش تر در یک خانه حافظه ذخیره می شود و بایت پر ارزش تر عدد در خانه حافظه بعدی ذخیره می گردد.
- به عبارت دیگر بایت کم ارزش تر در آدرس کوچکتر و بایت پر ارزش تر در آدرس بزرگتر ذخیره می شود. (Little-Endian)
- مثال : عدد 0538H چگونه در حافظه ذخیره می گردد:

9612	38
9613	05



# خواندن اطلاعات از حافظه

- در هنگام خواندن اطلاعات از حافظه (خارج کردن این اطلاعات از حافظه و وارد کردن آنها در ثبات های پروسسور) ابتدا بایت با آدرس کوچکتر در بایت کم ارزش ثبات قرار می گیرد و بایت با آدرس بزرگتر در بایت پرارزشتر ثبات قرار می گیرد.

9612	38
9613	05

<b>AH = 05</b>	<b>AL = 38</b>
----------------	----------------

# انواع ثبات های پروسور

- کامپیوتر های شخصی دارای تعدادی ثبات می باشند که کاربر می تواند روی آنها عملیات انجام دهد.
- ثبات ها به ۴ دسته تقسیم بندی می شوند:
  1. ثبات های عمومی
  2. ثبات های شاخص یا اشاره گر
  3. ثبات های سگمنت
  4. ثبات وضعیت یا پرچم

# ثبات های عمومی

- چهار ثبات  $AX, BX, CX, DX$  که ۱۶ بیتی هستند.
  - هر کدام از این ثباتها به یک بایت پر ارزش و یک بایت کم ارزش تقسیم میشوند.
  - می توان از این ثباتها، بسته به نیاز به صورت ۸بیتی یا ۱۶ بیتی استفاده نمود.
  - در صورتی که در یک دستور از ثبات ۸بیتی استفاده شود، آن دستور را ۸بیتی نامند و اگر از ثبات ۱۶ بیتی استفاده شود، آن را دستور ۱۶ بیتی گویند.
  - کاربرد معمول هر کدام از این ثباتها در ادامه شرح داده می شود.
- ❖ ثبات :  $AX$  این ثبات برای ذخیره کردن و انجام عملیات محاسباتی استفاده می شود.

AH	AL
----	----

## ثبات های عمومی ...

- ثبات BX: این ثبات برای محاسبات ریاضی و آدرس دهی استفاده می شود.

BH	BL
----	----

- ثبات CX: این ثبات جهت شمارش دفعات حلقه تکرار و شمارش تعداد شیفت ها و در کارهای محاسباتی مورد استفاده قرار می گیرد.

CH	CL
----	----

- ثبات DX: از این ثبات در عملیات ورودی خروجی و همچنین عملیات ضرب و تقسیم استفاده می شود.

DH	DL
----	----

# ثبات های شاخص یا اشاره گر

1. ثبات اشاره گر پشته SP (Stack Pointer)
2. ثبات اشاره گر پایه BP (Base Pointer)
3. ثبات اشاره گر دستور IP (Instruction Pointer)
4. ثبات ایندکس SI (Source Index)
5. ثبات ایندکس DI (Destination Index)

# ثبات های سگمنت

- یک برنامه اسمبلی از چند بخش تشکیل میشود که هر بخش اصطلاحاً یک سگمنت گفته می شود.
  - هر سگمنت برنامه اسمبلی، در یک سگمنت از حافظه قرار می گیرد.
1. ثبات سگمنت کد CS: دستورات برنامه اسمبلی در سگمنت کد قرار می گیرند و آدرس شروع این سگمنت درون این ثبات قرار می گیرد.
  2. ثبات سگمنت داده DS: در این سگمنت داده های مورد نیاز برنامه قرار می گیرند و آدرس شروع آن در ثبات DS قرار می گیرد.
  3. ثبات سگمنت پشته SS: این سگمنت جهت فراخوانی و آدرس دهی بازگشت از روتینها مورد استفاده قرار می گیرد و آدرس شروع ناحیه Stack در ثبات SS قرار می گیرد.
  4. ثبات سگمنت داده اضافی ES: این ثبات برای انجام عملیات رشته ای و کاربرد های خاص مورد استفاده قرار می گیرد، آدرس شروع این سگمنت در ثبات ES است.

# نمونه برنامه اسمبلی

```
;-----
; 1- Define stack segment
;-----
STACKSG SEGMENT STACK 'STACK'           سگمنت پشته
        DW 32H DUP(0)      ; 32H word for stack
STACKSG ENDS                          ; End of segment
;-----
; 2- Define data segment
;-----
DATASG  SEGMENT 'DATA'                   متغیرها در سگمنت داده
DATA1 DB 35H
DATA2 DB 32H
DATA3 DB ?
DATASG  ENDS                          ; End of segment
;-----
; 3- Define code segment
;-----
;-----
; دستورات برنامه در سگمنت کد
CODESG  SEGMENT 'CODE'
        ASSUME SS:STACKSG, DS:DATASG, CS:CODESG
MAIN    PROC FAR
        MOV AX,DATASG      ;1-Initialize DS
        MOV DS,AX         ;2- register
;-----
; Main program
;-----
        MOV BL,DATA1      ;3- Move 35h to BL
        SUB BL,30H        ;4- Change to binary
        MOV AL,DATA2      ;5- Move 32H to AL
        SUB AL,30H        ;6- Change to binary
        ADD AL,BL         ;7- Add BL to AL
        MOV DATA3,AL     ;8- Store sum in DATA3
;-----
; Come back to operating system
;-----
        MOV AX,4C00H      ;9- End of
        INT 21H          ;10- processing
MAIN    ENDP              ;11- End of procedure
CODESG  ENDS              ;12- End of segment
        END MAIN         ;13- End of program
```

# ثبات وضعیت یا پرچم (PSW)

این ثبات ۱۶ بیتی بوده و وضعیت فعلی پروسسور را مشخص می کند. دستورات محاسباتی و منطقی مانند جمع و تفریق و ... روی بیت های پرچم اثر می گذارند. شکل کلی ثبات PSW:

R	R	R	R	OF	DF	IF	TF	SF	ZF	U	AF	U	PF	U	CF
---	---	---	---	----	----	----	----	----	----	---	----	---	----	---	----

R: بیت رزرو شده

U: بیت رزرو شده

بیت پرچم نقلی CF: برای اعمال جمع و تفریق از این بیت استفاده می گردد.

بیت توازن PF: بعد از عملیات محاسباتی یا منطقی بایت کوچکتر بررسی می شود اگر تعداد بیت های یک این بایت زوج باشد بیت توازن یک می شود در غیر اینصورت صفر (بیانگر توازن فرد).

بیت نقلی کمکی AF: در محاسبات با کد های BCD اگر از بیت شماره ۳ بیت نقلی به بیت

بعدی ایجاد شود بیت پرچم نقلی کمکی AF یک می گردد در غیر این صورت صفر.  
زبان ماشین و برنامه سازی سیستم - فصل چهارم - ساختار اصولی کامپیوترهای شخصی - مدرس: بیدکی



# ثبات وضعیت یا پرچم ...

- بیت تشخیص صفر  $ZF$ : این بیت زمانی یک می شود که نتیجه عملیات محاسباتی یا منطقی صفر شود در غیر این صورت صفر.
- بیت علامت  $SF$ : در نمایش اعداد باینری ، بزرگترین یا پر ارزش ترین بیت ، بیت علامت است. بیت پرچم نتیجه آخرین محاسبات را نشان می دهد.
- بیت پرچم  $TF$ : زمانی که این بیت یک باشد یک دستور اجرا می شود که بعد از بررسی نتیجه محاسبات توسط کاربر دستور بعدی اجرا می گردد این عمل برای پیدا کردن اشتباه در برنامه بسیار مناسب است.
- بیت فعال کردن وقفه  $IF$ : این بیت برای فعال کردن (1) یا غیر فعال کردن (0) وقفه های خارجی است.
- بیت پرچم  $DF$ : این بیت برای کنترل جهت عملیات دستورات رشته بکار می رود. (1: راست به چپ و 0: چپ به راست)
- بیت سر ریز  $OF$ : این بیت موقعی یک می شود که نتیجه عملیات اعداد جبری بزرگتر از مقدار مجاز باشد.

# ثبات وضعیت یا پرچم (مثال)

38=00111000

+

2F=00101111

=

67=01100111

CF=0,AF=1,ZF=0,PF=0,SF=0, OF=0

## نحوه قرار گرفتن سگمنت های برنامه در حافظه

- سگمنت های برنامه می توانند در هر محل حافظه قرار گیرند ولی معمولاً از آدرس های حافظه که سمت راست آن ها صفر هستند شروع می شوند مانند  $00000H$  و  $00010H$  و  $00020H$  و ... .
- یعنی آدرس شروع هر سگمنت در حافظه نسبت به سگمنت بعدی حداقل به اندازه  $10H$  یا  $16$  بایت فاصله دارد.
- پس آدرس شروع هر سگمنت مضربی از  $16$  است و لذا میتوان اولین رقم صفر سمت راست آدرس سگمنت را در ثباتهای سگمنت ذخیره نکرد و فقط  $4$  رقم پر ارزشتر را بیان نمود.
- در هنگام مراجعه به حافظه، خود سخت افزار، صفر را جلوی آدرس ابتدای سگمنت قرار می دهد.
- به عنوان مثال اگر سگمنت داده در آدرس  $303E0H$  باشد، در ثبات سگمنت داده داریم:  $DS=303EH$

# افست سگمنت

- در یک برنامه اسمبلی تمام محل های حافظه نسبت به آدرس ابتدای هر سگمنت مشخص می شوند و فاصله یک اطلاعات یا دستور از ابتدای سگمنت، افست آدرس یا جابجایی نامیده می شود که این افست آدرس از 0000H تا FFFFH (۰ تا ۶۵۵۳۵) می باشد.
- بدیهی است اولین بایت سگمنت کد یا داده دارای افست آدرس 0000 و دومین بایت دارای افست آدرس 0001 و ... تا افست آدرس FFFFH می باشد.
- به عنوان مثال اگر  $DS=303EH$  بوده و اطلاعاتی با افست آدرس 16H مورد نیاز باشد ( `MOV AL, [0016H]` )
- آدرس منطقی داده:  $DS:Offset \rightarrow 303E:0016$
- آدرس فیزیکی (واقعی) داده:  $DS*16+Offset \rightarrow 303E0+0016= 303F6H$

## نحوه قرار گرفتن سگمنت پشته، داده و کد در حافظه

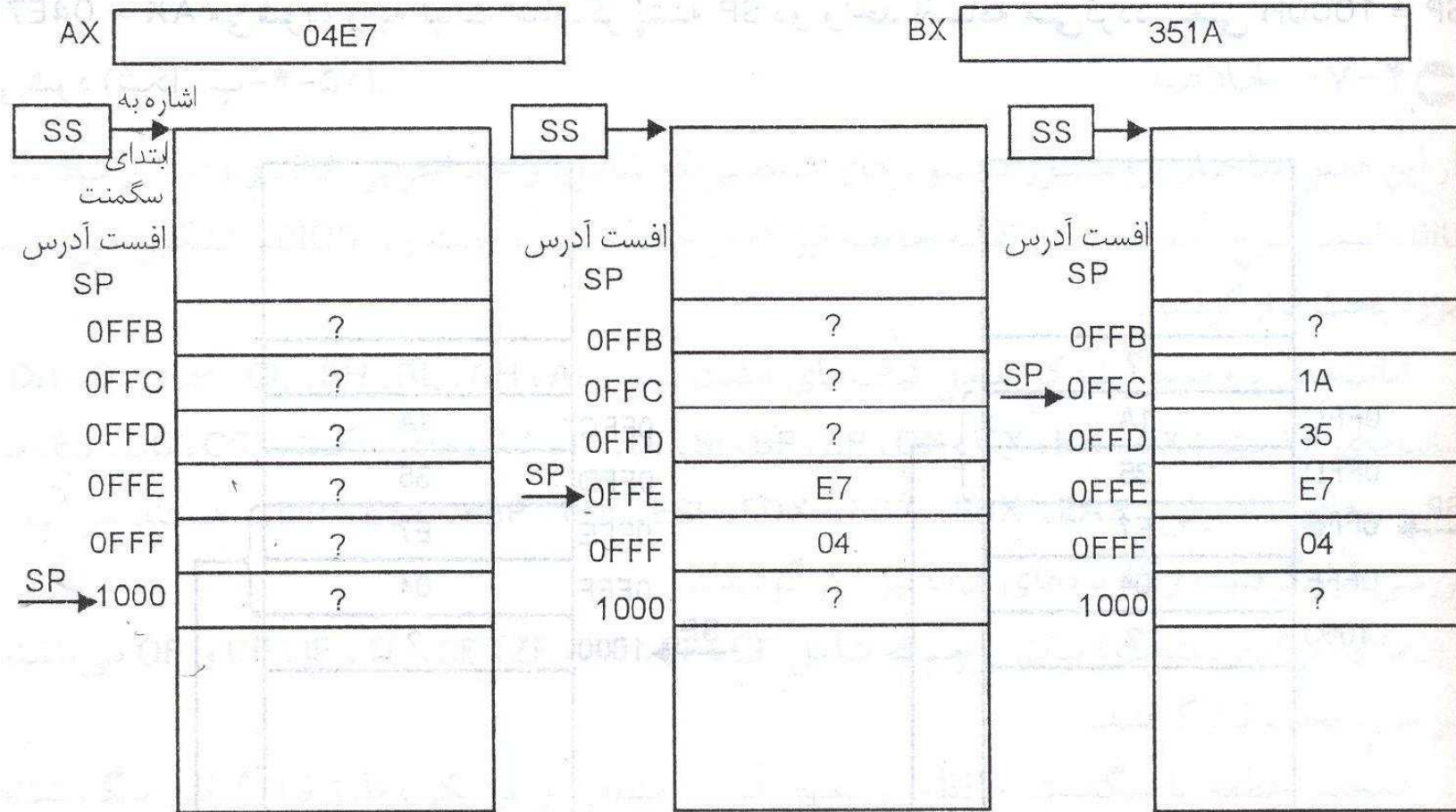
سیستم عامل :

1. از مقداری که در برنامه کاربر برای سگمنت پشته رزرو شده، آدرس شروع سگمنت بعدی یعنی سگمنت داده را تعیین می کند.
2. از داده های تعریف شده در سگمنت داده، مقداری حافظه مورد لزوم را تعیین می نماید و آدرس سگمنت بعدی یعنی سگمنت کد را تعیین می کند.
3. در سگمنت کد نیز می توان از ۰ تا ۶۵۵۳۵ بایت حافظه برای دستورات استفاده کرد.

# حافظه پشته

- هر برنامه اسمبلی به حافظه پشته نیاز دارد و اطلاعات در حافظه پشته به صورت یک کلمه، یک کلمه (۲بایت، ۲بایت) نوشته یا از آن خوانده می شوند.
- ثبات سگمنت پشته SS، آدرس ابتدای سگمنت پشته را مشخص می نماید و ثبات اشاره گر پشته SP، افسر آدرس در داخل سگمنت پشته را نشان می دهد.
- دستور PUSH برای ذخیره اطلاعات در حافظه پشته است و دستور POP جهت بازیابی اطلاعات از حافظه پشته می باشد.
- دستور PUSHF برای ذخیره اطلاعات ثبات پرچم در حافظه پشته و دستور POPF نیز برای بازیابی پرچم ها از حافظه پشته و قراردادن آن ها در ثبات پرچم بکار می روند.
- دستور PUSHA تمام ثبات های پروسور را در حافظه پشته ذخیره و دستور POPA آن ها را بازیابی می نمایند.

# نوشتن در پشته

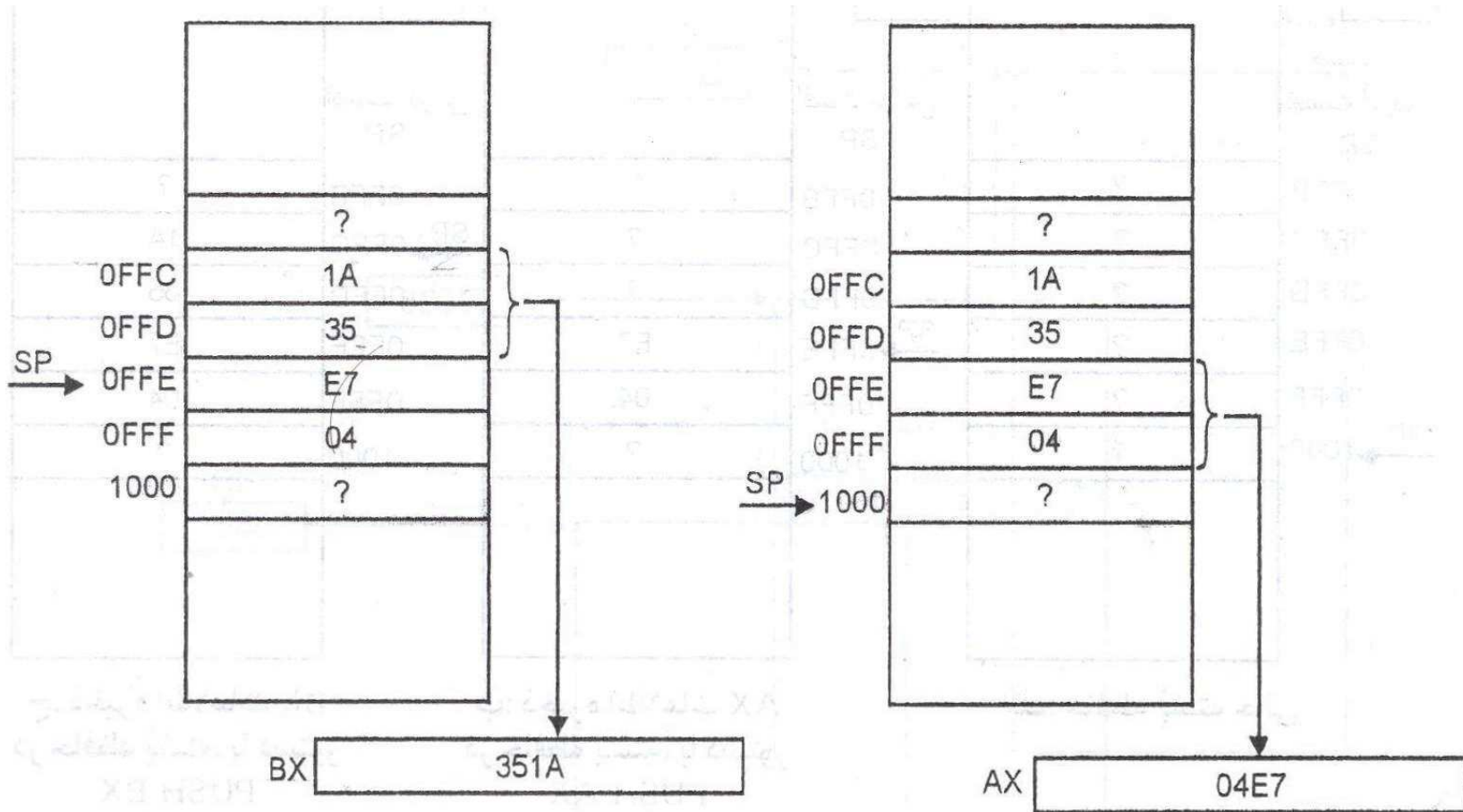


الف: حافظه پشته خالی

ب: ذخیره اطلاعات AX  
در حافظه پشته، با دستور  
PUSH AX

ج: ذخیره اطلاعات BX  
در حافظه پشته، با دستور  
PUSH BX

# خواندن از پشته



الف: بازیابی اطلاعات حافظه پشته، با دستور  
POP BX، و قرار دادن آن در ثبات BX

ب: بازیابی اطلاعات حافظه پشته، با دستور  
POP AX، و قرار دادن آن در ثبات AX

شکل (۴-۱۵) نمایش اطلاعات حافظه پشته و اجرای دستور POP