

## ریزپردازنده های اینتل

با توجه به اینکه دستورات اسمبلی ۸۰۸۶ در قسمت ها آینده بررسی می شوند، در این صفحه با نحوه آدرس دهی حافظه و ثبات های این پردازنده آشنا خواهید شد.

[خانواده 80x86](#)  
[آدرس دهی سگمنتی](#)  
[مدهای اجرا](#)  
[مجموعه ثبات ها](#)

### خانواده 80x86

کلیه کامپیوترهای شخصی IBM پردازنده ای از خانواده 80x86 دارند. پردازنده های این خانواده همگی دارای ویژگی های مشترکی از جمله زبان ماشین پایه یکسان هستند. البته اعضای جدید ویژگی های خود را به میزان زیادی افزایش داده اند.

تعدادی از پردازنده های این خانواده بدین شرح می باشند:

**(1978)8086, (1979)8088**

• این CPU ها، که از دیدگاه برنامه نویسی برابر هستند، پردازنده هایی بودند که روی اولین کامپیوترهای شخصی به کار رفته اند. دارای ثبات های ۱۶ بیتی (AX, BX, CX, DX, SI, DI, BP, SP, CS, DS, SS, ES, IP) و (FLAGS) هستند و تنها در مد حقیقی عمل می کردند. ۸۰۸۶ دارای گذرگاه داده ۱۶ بیتی و گذرگاه آدرس ۲۰ بیتی بود و بنا براین قابلیت آدرس دهی تا ۱ مگابایت حافظه را داشت و می توانست با داده های ۸ یا ۱۶ بیتی همزمان کار کند. ۸۰۸۸ با گذرگاه داده ۸ بیتی به طراحان اجازه پیچیدگی کمتر و ارزانتر سیستم های کامپیوتری را می داد.

**(1983)80286**

• این پردازنده، که در کامپیوترهای شخصی کلاس AT استفاده شد، دستورالعمل های جدیدی را به زبان ماشین ۸۸/۸۰۸۶ اضافه کرد. اما ویژگی اصلی آن مد محافظت شده ۱۶ بیتی بود که در این حالت می توانست تا ۱۶ مگابایت حافظه را دسترسی پیدا کند. البته برنامه ها همچنان به سگمنت هائی تقسیم بندی می شدند که نمی توانستند بیشتر از ۶۴K باشند.

**(1986)80386**

• اولین پردازنده ۳۲ بیتی که توسط اینتل معرفی شد ۸۰۳۸۶ DX بود که علاوه بر حفظ سازگاری با پردازنده های قبلی اجرای عالی داشت. این پردازنده چند ثبات را به ۳۲ بیتی گسترش داد (EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP, EIP) و دو ثبات جدید ۱۶ بیتی FS و GS را اضافه کرد. دارای گذرگاه های آدرس ۳۲ بیتی بود و در مد محافظت شده ۳۲ بیتی می توانست تا ۴ گیگابایت حافظه فیزیکی را آدرس دهی کند. برنامه ها دوباره به سگمنت ها تقسیم می شدند اما اندازه هر سگمنت می توانست تا ۴ گیگا بایت باشد. نسخه ۱۶ بیتی آن 80386 SX با گذرگاه آدرس ۲۴ و داده ۱۶ بیتی در ۱۹۸۸ بیرون آمد که تنها تا 16 مگابایت را دسترسی داشت.

**(1989)80486**

• 80486 DX دارای حافظه نهان و کمک پردازنده ریاضی در یک تراشه بود که حدود ۵۰٪ سریع تر از ۸۰۳۸۶ بود. ۸۰۴۸۶ SX را هم معرفی شد که تنها پیوند آن با میکروپروسور ریاضی وجود نداشت.

**(1993)Pentium/Pentium Pro**

• پردازنده های ۶۴ بیتی پنتیوم، که چند دستورالعمل را در یک زمان اجرا می کنند، سرعت اجرای دستورالعمل ها را بالا بردند. این پردازنده ها دارای گذرگاه داده ۶۴ بیتی و گذرگاه آدرس ۳۲ بیتی هستند. پنتیوم از نظر کارایی دوبار سریع تر از ۸۰۴۸۶ است و عملیات ممیزشناور را سریع تر انجام می دهد در عین حال که کاملاً با قبلی ها سازگاری دارد.

**Pentium MMX**

• این پردازنده دستورات (MultiMedia eXtensions) MMX را به پنتیوم اضافه کرد. این دستورالعمل ها می توانند عملیات گرافیکی معمول را سرعت ببخشند.

**(1997)Pentium II**

• این پردازنده توسعه یافته پنتیوم است که قادر است ۴ پردازنده را همزمان پشتیبانی کند و به ۶۴ گیگابایت حافظه دسترسی دارد. در واقع یک پردازنده پنتیوم پرو همراه با دستورالعمل های MMX است.

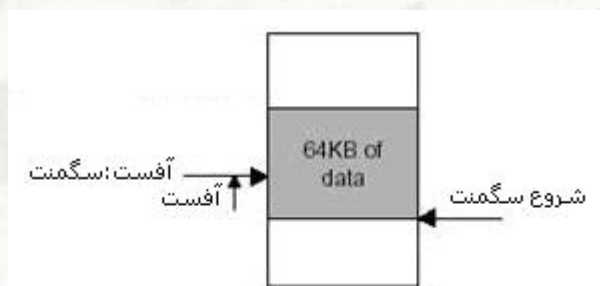
**(1999)Pentium III/(2002)Pentium IV**

• این پردازنده ها تنها سرعت اجرای دستورالعمل ها را بالا بردند.

**آدرس دهی سگمنتی**

پردازنده های ۸۰۸۶ دارای گذرگاه ۲۰ بیتی هستند، بنابراین می تواند تا ۱ مگابایت حافظه را آدرس دهد (از آدرس 00000 تا 1MB=1048575=FFFFF). این آدرس ها به یک عدد ۲۰ بیتی احتیاج دارند. روشن است که یک عدد ۲۰ بیتی را نمی توان در ثبات های ۱۶ بیتی ۸۰۸۶ جا داد. اینتل این مشکل را با آدرس دهی سگمنتی (segment addressing) حل کرد. سگمنت یک تکه از حافظه با اندازه ۶۴ کیلوبایت است. یک محل از حافظه با یک آدرس سگمنت و یک آفست (offset) مشخص می شود که به صورت دو عدد ۱۶ بیتی نشان داده می شوند. آدرس سگمنت به سگمنتی در حافظه اشاره می کند که حاوی محل مورد نظر است. هر سگمنت از یک پاراگراف می تواند شروع شود. هر پاراگراف ۱۶ بایت دارد، بنابراین سگمنت از آدرسی که مضربی از ۱۶ است شروع می شود. یعنی سگمنت اول از آدرس 00000، سگمنت دوم از آدرس 00010، بعدی از آدرس 00020 و الی آخر شروع می شود. در نتیجه آدرس شروع هر سگمنت از سمت راست به صفر ختم می شود که از آن صرف نظر می شود. بنابراین آدرس سگمنت همیشه به صورت یک عدد ۴ رقمی هگز نوشته می شود.

آفست فاصله بایت مورد نظر از ابتدای سگمنت را مشخص می کند. با توجه به اینکه هر سگمنت 64KB حافظه دارد، آفست می تواند بین 0000 تا ffff باشد. بنابراین آفست نیز همیشه یک عدد ۴ رقمی هگز است.



آدرس سگمنتی به صورت آفست: سگمنت نوشته می شود.

مثال ۱. آدرس فیزیکی 18A3:5B27 به بایستی در سگمنت 18A30 اشاره دارد که از ابتدای این سگمنت 5B27 بایت فاصله دارد.

مثال ۲. آدرس فیزیکی 04808 می تواند توسط 047C:0048 رجوع شود.

آدرس های سگمنت-آفست یک آدرس منطقی را تعیین می کنند. برای ساختن آدرس فیزیکی ۲۰ بیتی طبق فرمول زیر محتوای آدرس سگمنت را در ۱۶ ضرب کرده با آدرس آفست جمع می کنیم:

$$16 \times \text{segment} + \text{offset}$$

ضرب در ۱۶ آسان است کافی است یک صفر در سمت راست عدد گذاشته شود.

مثال ۳. آدرس فیزیکی رجوع شده توسط 047C:0048 برابر است با:

$$047C0 + 0048 = 04808$$

سگمنت ها روی هم می توانند قرار بگیرند بنابراین یک بایت، با داشتن آدرس فیزیکی منحصر بفرد در حافظه، می تواند از طریق چندین ترکیب سگمنت-آفست بدست می آید.

مثال ۴. آدرس فیزیکی 04808 می تواند توسط 047C:0048، 047D:0038، 047E:0028 یا 047B:0058 رجوع شود.

## سگمنت های برنامه

سه ناحیه از سگمنت های حافظه که هر یک می توانند 64KB باشند برای یک برنامه در نظر گرفته می شوند :

۱. سگمنت کد
  - شامل دستورالعمل های زبان ماشین برنامه ای که دارد اجرا می شود. اولین دستور اجرایی برنامه در ابتدای این سگمنت قرار دارد و سیستم عامل CPU را برای اجرای برنامه به این محل ارجاع می دهد.
۲. سگمنت داده
  - شامل داده های تعریف شده و ناحیه کاری که برنامه نیاز دارد.
۳. سگمنت پشته
  - شامل آدرس های برگشتی از زیربرنامه ها و داده های محلی است.

نکته ۱. برنامه و داده در هر سگمنتی از حافظه می توانند قرار گیرند، فقط آدرس شروع سگمنت باید برای CPU تعریف شده باشد. این آدرس ها در ثبات های سگمنت ذخیره می شوند و اغلب در طول اجرای برنامه ثبت باقی می مانند. در عمل هنگام برنامه نویسی تنها از آدرس ۴ رقمی آفست استفاده می شود.

نکته ۲. سگمنت ها می توانند روی همدیگر بیافتند. در بعضی مواقع که برنامه کوتاه است سگمنت داده می تواند از داخل سگمنت کد شروع شود به شرط اینکه تداخل رخ ندهد.

## مدهای اجرا

پردازنده های ۸۰۲۸۶ به بعد دارای دو مد حقیقی (real mode) و محافظت شده (protected mode) برای اجرا هستند. تعاریف سگمنت داده شده در قسمت بالا بر اساس مد حقیقی است.

در مد حقیقی پردازنده مانند ۸۰۸۶ عمل می کند. ارجاع به حافظه توسط یک آفست ۱۶ بیتی درون یک سگمنت تعیین می شود. آدرس فیزیکی ۲۰ بیتی طبق فرمول آفست + ۱۶ × سگمنت بدست می آید. به این طریق تا یک 1MB حافظه قابل آدرس دهی است. اما در هر لحظه فقط تا 64KB را می توان آدرس داد. در این حالت یک برنامه به هر آدرسی از حافظه دسترسی دارد حتی حافظه برنامه های دیگر که باعث می شود اشکالزدایی و امنیت بسیار دشوار بشود.

کلید برنامه های تحت DOS در مد مجازی اجرا می شوند.

در مد محافظت شده پردازنده می تواند از قابلیت های خود در گذرگاه های آدرس و داده به طور کامل استفاده کنند. در این مد می تواند حافظه بیشتری را آدرس دهی کند و برنامه ها را از دسترسی حافظه های یکدیگر محافظت می کرد.

مد محافظت شده تکنیکی به نام حافظه مجازی را استفاده می کند که برپایه نگهداری قسمتی از داده و کد در حافظه است که برنامه دارد اجرا می کند. بقیه داده و کد تا زمانی که مورد نیاز باشند روی دیسک نگهداری می شوند. سگمنت ها بین حافظه و دیسک در صورت نیاز منتقل می شوند و برخلاف مد حقیقی مکان های ثابتی در حافظه ندارند. اطلاعات سگمنت ها درون جدولی ذخیره می شود. ایندکس جدول هنگام آدرس دهی در ثبات سگمنت قرار می گیرد.

## مجموعه ثبات ها

پردازنده ۸۰۸۶ دارای ۱۴ ثبات ۱۶ بیتی با کاربردهای متفاوت است. این ثبات ها را می توان به صورت زیر گروه بندی کرد :

۱. ثبات های همه منظوره: AX، BX، CX و DX
۲. ثبات های ایندکس: SI و DI
۳. ثبات های آدرسی: BP، SP و IP
۴. ثبات های سگمنت: CS، DS، SS و ES
۵. ثبات های وضعیتی: Flag

## ثبات های همه منظوره

CPU اولیه ۸۰۸۶ با چهار ثبات همه منظوره طراحی شد که در دستورات محاسباتی و ورودی/خروجی استفاده می شوند. هر کدام از این ثبات ها یک یا چند وظیفه خاص هم دارند .

ثبات های همه منظوره می توانند به صورت ۸ یا ۱۶ بیتی استفاده شوند. هر کدام از آنها از دو بایت تشکیل شده اند؛ بایت سمت راست را Low Order و سمت چپ را High Order می نامند .

همه منظوره ترین ثبات است و معمولاً برای هر کاری از جمله عملیات ورودی/خروجی، رشته ای و محاسباتی به کار می رود از دو جزء AL و AH تشکیل شده است	Accumulator Register	AX
تنها ثباتی که می تواند بعنوان ایندکس در آدرس دهی مورد استفاده قرار می گیرد. شامل دو قسمت BH و BL است	Base Register	BX
بعنوان شمارنده در کنترل تعداد دفعات تکرار در دستور حلقه استفاده می شود. دارای دو قسمت CL و CH است	Count Register	CX
در اعمال ورودی/خروجی و عملیات ضرب و تقسیم استفاده می شود. دارای دو بخش DL و DH است	Data Register	DX

## ثبات های سگمنت

برای آدرس دهی به هر یک از این سگمنت های برنامه یک ثبات وجود دارد که مشخص می کند کدام بخش حافظه برای قسمت های مختلف برنامه به کار رفته است .

شامل آدرس شروع سگمنت کد، که به CPU می فهماند دستورالعمل های برنامه در کجا قرار دارند	Code Segment	CS
شامل آدرس شروع سگمنت داده که به پردازنده می فهماند داده ها و فضای کاری در کجا قرار دارد	Data Segment	DS
آدرس شروع سگمنت پشته را در خود ذخیره می کند	Stack Segment	SS
آدرس شروع سگمنت اضافی، سگمنت داده دوم	Extra Segment	ES

## ثبات های ایندکس

دو ثبات ۱۶ بیتی ایندکس وجود دارد که اغلب به عنوان اشاره گر به همراه DS به کار می روند تا به داده های موجود در سگمنت داده دسترسی شود. اما می توانند به همان منظورهای دیگر، مانند ثبات های همه منظوره، هم استفاده شود؛ گرچه نمی توانند به دو بخش ۸ بیتی تجزیه شوند .

برای آدرس دهی و در عملیات رشته ای بعنوان مبدا استفاده می شود	Source Index	SI
برای آدرس دهی و در عملیات رشته ای بعنوان مقصد استفاده می شود	Destination Index	DI

## ثبات های اشاره گر

ثبات های اشاره گر نگهدارنده بخش آفست در آدرس دهی هستند و همراه با یکی از ثبات های سگمنت به محلی از حافظه اشاره دارند. طبق پیش فرض ثبات های همه منظوره و ایندکس همراه با DS و ثبات های پشته همراه با SS و IP همراه با CS استفاده می شوند .

همراه با ثابت CS به دستورالعمل بعدی که باید توسط CPU اجرا شود اشاره می کند	Instruction Pointer	IP
آهست مکانی از سگمنت پشته که عمل قرار گرفتن داده در پشته صورت می گیرد. به عبارت دیگر SS:SP به بالای پشته اشاره دارد	Stack Pointer	SP
برای دسترسی به متغیرهای محلی که در پشته قرار دارند استفاده می شود	Base Pointer	BP

### ثبات FLAGS

فلگ ها اطلاعاتی درباره نتایج اجرای دستورالعمل قبلی را نگه می دارند. این نتایج به صورت بیت های مجزا در ثبات وضعیت FLAGS ذخیره می شوند. ۹ بیت از ۱۶ بیت این ثبات برای تعیین وضعیت فعلی ماشین و نتیجه اجرای دستورالعمل به کلر می روند. هر کدام از این بیت ها هم فلگ نامیده می شوند زیرا می توانند 1 (Set) یا صفر (Set Not) باشند. بسیاری از دستورالعمل ها وضعیت این بیت ها را تغییر می دهند. گرچه کلیه دستورات روی فلگ تاثیر نمی گذارند. این ثبات فاقد آدرس است و به طور مستقیم توسط برنامه نویس قابل دسترس نمی باشد.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				O	D	I	T	S	Z		A		P		C

محتوی رقم نقلی بوجود آمده از باارزش ترین بیت در عملیات محاسباتی یا چرخش	Carry Flag	CF
برای کنترل صحت انتقال داده. اگر صفر باشد تعداد بیتهای انتقالی فرد است و اگر یک باشد زوج است	Parity Flag	PF
محتوی رقم نقلی از بیت سوم به چهارم در یک بایت است. در عملیات BCD کاربرد دارد	Auxiliary Carry	AF
اگر نتیجه عملیات محاسباتی صفر باشد این بیت ۱ است در غیر اینصورت صفر است	Zero Flag	ZF
در صورت منفی بودن نتیجه عملیات این بیت ۱ است در غیر اینصورت صفر است	Sign Flag	SF
برای اجرای دستورالعمل به صورت دستور به دستور این بیت باید ۱ باشد	Trap Flag	TF
اگر ۱ باشد وقفه فعال است و اگر صفر باشد وقفه غیر فعال است یعنی سیستم وقوع وقفه را نادیده می گیرد	Interrupt Flag	IF
اگر ۱ باشد عمل مقایسه یا انتقال داده از سمت راست به چپ صورت می گیرد در غیر اینصورت از چپ به راست	Direction Flag	DF
اگر در باارزش ترین بیت سرریزی وجود داشته باشد این بیت یک می شود	Overflow Flag	OF