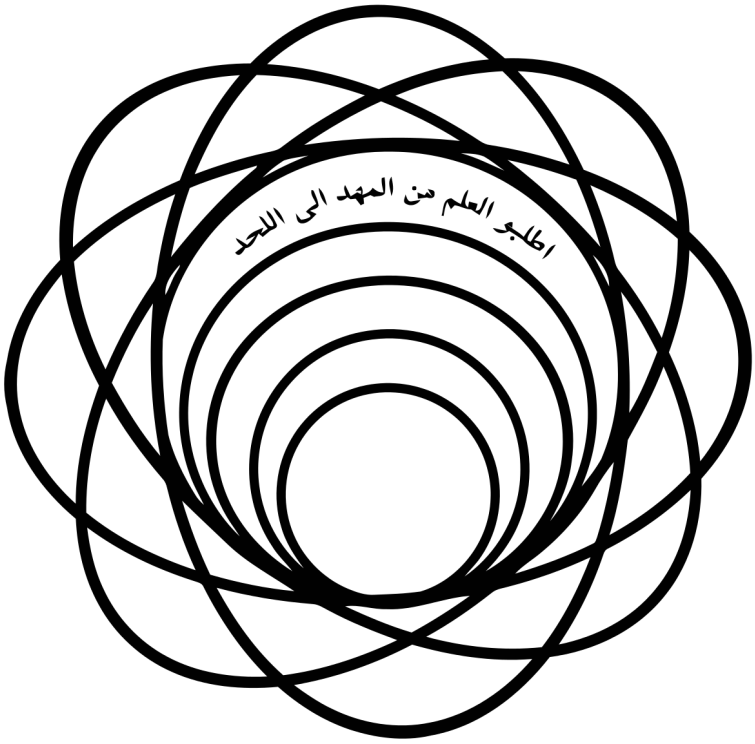


Shamsipour
Technical and Vocational
College

درس:
برنامه سازی سیستم

جمع آوری مطالب :
برهلیا

جلسه اول

فصل اول: مبانی برنامه سازی سیستم

نیمسال دوم ۹۹–۹۸

فصل ۱: مبانی برنامه‌سازی سیستم
(System Programming Basics)

۱-۱ برنامه‌سازی سیستم چیست؟

از دید افراد مختلفی که با کامپیوتر کار می‌کنند این پرسش می‌تواند پاسخهای متفاوتی داشته باشد. برخی به هر برنامه به دید یک "سیستم" نگاه می‌کنند و **برنامه‌سازی سیستم** را فرآیند تبدیل یک مسأله به یک برنامه‌ی قابل اجرا می‌دانند. برخی به نوشتن برنامه‌های خاص برای یک سیستم کامپیوتری خاص، برنامه‌سازی سیستم می‌گویند.

گرچه تعریف دوم تا حدی درست است، اما به‌طور دقیق‌تر از دید ما، برنامه‌سازی سیستم دارای تعریف مشخص و معینی است: **برنامه‌سازی سیستم نوع خاصی از برنامه‌سازی است که نیازمند اطلاع از جزئیات فنی سیستم مورد نظر و دسترسی و مدیریت منابع سخت‌افزاری سیستم به‌صورت دلخواه است.** این نوع برنامه‌سازی عموماً برای نوشتن **نرم‌افزارهای سیستمی**[۱] مورد استفاده قرار می‌گیرد.

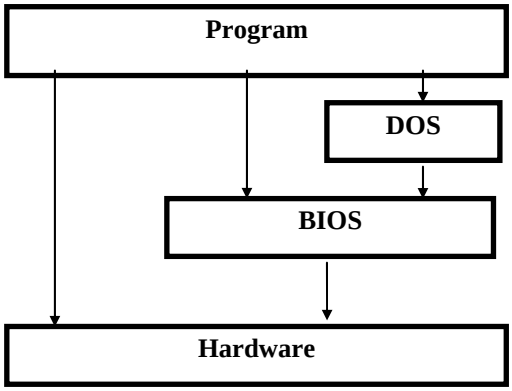
برنامه‌سازی کاربردی[۲] **در مقابل برنامه‌سازی سیستم:** برنامه‌سازی کاربردی شامل مدیریت و ارائه‌ی اطلاعات در قالب یک برنامه‌ی کامپیوتری بوده و نیازمند نگهداری اطلاعات در قالب ساختارهای داده‌ای (نظیر آرایه، رکورد و غیره) و پردازش آنها است. الگوریتم‌های مورد استفاده در برنامه‌سازی کاربردی **مستقل از سیستم**[۳] بوده و برای اغلب کامپیوترها قابل تعریف هستند. اما، روشی که اطلاعات به برنامه منتقل می‌شود و طریقی که اطلاعات نمایش داده می‌شوند یا چاپ می‌شوند، **وابسته به سیستم**[۴] است. در برنامه‌سازی سیستم، کنترل هرنوع سخت‌افزاری که اطلاعات را به کامپیوتر می‌فرستد یا از آن دریافت می‌کند، انجام می‌شود. برای پردازش این اطلاعات نیاز به برنامه‌های کاربردی هستیم. از این‌رو، برای بهره‌برداری از سخت‌افزار کامپیوتر، هر دو نوع برنامه‌سازی لازم هستند.

۱-۲ مدل سه لایه‌ای

مهمترین هدف برنامه‌سازی سیستم، دسترسی و مدیریت سخت‌افزار کامپیوتر است. نوعِ سیستمِ کامپیوتری که مورد نظر ما است، **کامپیوتر شخصی(PC)**[۵] است. این نوع کامپیوتر دارای معماری خاص بوده و اجزاء سخت‌افزاری آن با سایر معماری‌ها متفاوت است. نوعی از برنامه‌سازی سیستم که مورد نظر ما است، برنامه‌سازی همین نوع سیستم کامپیوتری بوده و هدف آن دسترسی و مدیریت سخت‌افزار PC است.

در محیط PC، دسترسی به سخت‌افزار از طریق ROM-BIOS یا DOS نیز امکان پذیر است. ROM-BIOS، مخفف Read Only Memory-Basic Input/output System است و DOS هم مخفف Disk Operating System است. BIOS و DOS واسطهای نرم‌افزاری[۶] هستند که به‌طور خاص برای مدیریت سخت‌افزار ایجاد شده‌اند. در شکل (۱-۱) مدلی سه‌لایه‌ای از ارتباط برنامه‌ها با BIOS و DOS ارائه شده است.

شکل ۱-۱: مدل سه‌لایه‌ای



مهمترین مزیت استفاده از DOS یا BIOS این است که یک برنامه نیازمند دسترسی مستقیم به سخت‌افزار نیست و در عوض یک روال[۷] BIOS یا DOS را فراخوانی می‌کند تا کار موردنظر را برایش انجام دهد. روالهای BIOS پس از انجام یک کار، اطلاعات وضعیت[۸] را به برنامه برمی‌گردانند تا برنامه از نتیجه‌ی عمل آگاه شود. این واسطها باعث می‌شوند که برای نوشتن برنامه‌ها، هزینه و زمان کمتری صرف شود.

---

۱ system softwares
۲ application programming
۳ system independent
٤ system dependent
٥ Personal Computer
٦ software interfaces
۷ routine
۸ status information

مزیت دیگر استفاده از واسطهای فوق، عدم وابستگی برنامه به مشخصات فیزیکی سخت‌افزار است. برای مثال، نمایش اطلاعات بر روی صفحه‌ی نمایش با استفاده از کارتهای ویدئویی[۱] تک رنگ[۲] و رنگی (نظیر EGA، VGA، SuperVGA و غیره) با هم دارای تفاوتهای اساسی است. اگر برنامه بخواهد خودش اطلاعات را با برنامه‌سازی کارت ویدئویی، نمایش دهد، باید روالهای مجزایی برای هر کدام از انواع کارت ویدئویی نوشته شود. اما اگر از روالهای BIOS استفاده نماید، برنامه مستقل از نوع کارت ویدئویی خواهد بود. چون انواع مختلف کارتهای ویدئویی بوسیله‌ی BIOS پشتیبانی می‌شوند.

ROM-BIOS ۳-۱

در شکل (۱-۱) یک مدل سه‌لایه‌ای را مشاهده می‌کنید که واسط BIOS از DOS به سخت‌افزار نزدیک‌تر است. BIOS توابعی را برای دسترسی و مدیریت لوازم[۳] زیر فراهم می‌کند:

۱. کارت ویدئویی،
۲. حافظه (RAM)،
۳. دیسکت،
۴. دیسک سخت[۴]،
۵. درگاههای سریال[۵] (Com1 و Com2)،
۶. درگاههای موازی[۶] (LPT1 و LPT2)،
۷. صفحه کلید،
۸. ساعت بی‌درنگ عمل‌کننده‌با باتری[۷].

گرچه می‌توان بدون استفاده از BIOS نیز با سخت‌افزار ارتباط برقرار نمود، اما عموماً بهتر است از طریق این توابع استاندارد به کامپیوتر دسترسی پیدا کرد، تا برنامه مستقل از سخت‌افزار باشد.

BIOS در یک تراشه‌ی حافظه[۸] از نوع ROM در برد اصلی[۹] سیستم PC وجود دارد . بلافاصله پس از روشن شدن کامپیوتر، قابل دسترسی است. توابع موجود در BIOS، وظایفی نظیر آزمون حافظه و بررسی لوازم جانبی را پس از روشن شدن کامپیوتر برعهده دارند. به همراه هر نوع جدید برد اصلی PC و مدل CPU، نگارشهای[۱۰] جدیدتری از BIOS معرفی می‌شوند. از معروفترین انواع BIOS می‌توان از Award و AMI نام برد.

DOS ۴-۱

DOS که در حقیقت وظیفه‌ی یک سیستم عامل را برعهده دارد، بوسیله‌ی BIOS از روی دیسک در حافظه‌ی کامپیوتر قرار داده شده و اجرا می‌شود. از جمله توابعی که در DOS فراهم شده است، توابعی هستند که امکان کار با دیسک را فراهم می‌نمایند و این امر دلیل نام‌گذاری DOS است. یعنی سیستم عاملی که توانایی کار با دیسک را دارد.

در کنار BIOS، DOS نیز توابعی را برای دسترسی به سخت‌افزار فراهم می‌کند. از آنجایی ک DOS به سخت‌افزار به عنوان **لوازم منطقی**[۱۱] نگاه می‌کند، نه مثل BIOS به عنوان لوازم فیزیکی، لذا توابع DOS سخت‌افزار را به‌طریقی متفاوت مدیریت می‌کنند. برای مثال، BIOS به گرداننده‌های دیسک[۱۲] به‌عنوان گروهی از شیارها[۱۳] و بخشها[۱۴] نگاه می‌کند، اما DOS آنها را به‌عنوان گروه‌هایی از فایلها و فهرست‌ها[۱۵] می‌بیند.

---

۱ video cards
۲ monochrome
۳ devices
۴ harddisk
۵ serial ports
۶ parallel ports
۷ battery-operated real-time clock
۸ memory chip
۹ mother board
۱۰ version
۱۱ logical devices
۱۲ disk drives
۱۳ tracks
۱۴ sectors
۱۵ directory

اگر بخواهید ۱۰۰۰ کاراکتر اول یک فایل را بر روی صفحه نمایش ببینید، از طریق BIOS اگر عمل کنید، باید مکان فایل بر روی درایو (شماره شیار و بخش) را به BIOS بگویید. اما با استفاده از توابع DOS، کافی است به DOS بگوییم که فایل با نام داده‌شده را در گرداننده‌ی A: یا C: باز نموده و نمایش دهد.

گرچه اغلب اوقات DOS از طریق BIOS به سخت‌افزار دسترسی دارد، ولی گاهی اوقات هم به‌صورت مستقیم با سخت‌افزار ارتباط برقرار می‌کند.

۱-۵ انتخاب روش دسترسی به سخت‌افزار

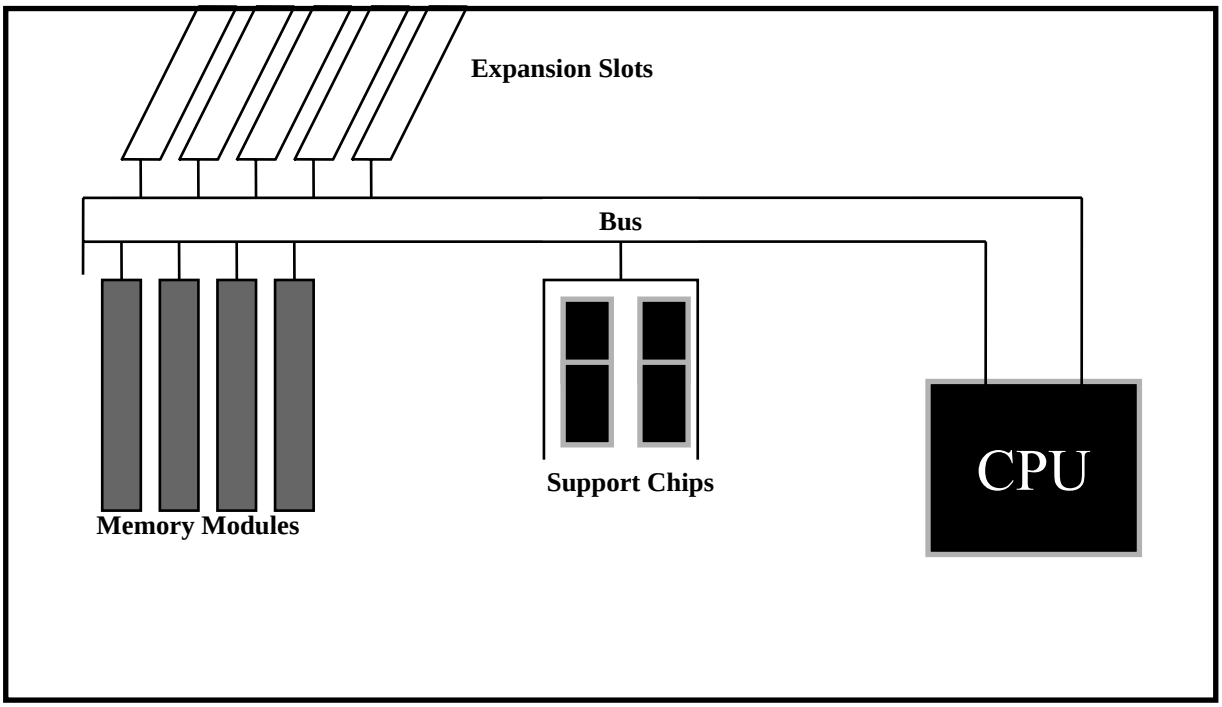
از کدام توابع استفاده کنیم؟ در ادامه‌ی درس در مورد اینکه DOS و BIOS چه توابعی دارند صحبت خواهیم نمود. اما ابتدا باید تصمیم بگیریم که کدام روش دسترسی را انتخاب نماییم: DOS یا BIOS یا دسترسی مستقیم به سخت‌افزار؟ گاهی اوقات مجبور به دسترسی مستقیم هستیم. زیرا ممکن است عمل موردنظر ما را هیچکدام از توابع DOS یا BIOS انجام ندهند. برای مثال اگر بخواهید به کارت ویدئویی بگویید که یک خط یا دایره رسم کند، هیچ تابعی در DOS یا BIOS برای این منظور پیدا نخواهید کرد. بنابر این، مجبورید که یک روال برای دسترسی مستقیم به سخت‌افزار و برنامه‌ریزی آن برای رسم خط یا دایره بنویسید، یا آنکه از یک تابع کتابخانه‌ای نرم‌افزاری[۱] تهیه شده بوسیله‌ی دیگران استفاده کنید.

انتخاب DOS یا BIOS ؟ این انتخاب به نوع کاربرد بستگی دارد. برای مثال اگر می‌خواهید در برنامه با فایل کار کنید، باید از توابع DOS استفاده کنید. ولی اگر می‌خواهید یک دیسکت را قالب‌بندی[۲] کنید، باید از توابع BIOS استفاده کنید. هم توابع DOS و هم توابع BIOS، در برخی اوقات **کند** هستند. زیرا مدل سه‌لایه‌ای شکل (۱-۱) و نیز عمومی بودن توابع، باعث عدم بهینه‌بودن توابع این واسطها می‌شود. حتی به این دلیل هم گاهی اوقات مجبور به برنامه‌سازی مستقیم سخت‌افزار هستیم.

۱-۶ سخت‌افزار PC

بلوک دیاگرام اجزاء سخت‌افزار PC در شکل (۱-۲) نمایش داده شده است:

شکل ۱-۲: بلوک دیاگرام سخت‌افزار PC



اجزاء سخت‌افزار PC شامل موارد زیر است:

- **پردازنده:** CPU در PC، یکی از ریزپردازنده‌های سری 80X86 ساخت شرکت Intel است. اولین PC ساخت شرکت IBM با نام System/23 DataMaster دارای ریزپردازنده‌ی ۸ بیتی ۸۰۸۵ بود. سپس به ترتیب ریزپردازنده‌های ۸۰۸۶، ۸۰۸۸، ۸۰۲۸۶، ۸۰۳۸۶، ۸۰۴۸۶ و Pentium از طرف شرکت Intel معرفی شدند و این پردازنده‌ها در مدلهای مختلف کامپیوترهای شخصی ساخت شرکت IBM یا شرکتهای دیگر مورد استفاده قرار گرفتند.

---

۱ software library function
۲ format

- **گذرگاه**[۱]: یکی از بخشهای اساسی سیستم PC است که وظیفه‌ی اتصال اجزاء مختلف آنرا دارد. گذرگاه در حقیقت یک کابل با ۶۲ خط اتصال است. پردازنده از طریق گذرگاه اطلاعات را از حافظه خوانده و پردازش می‌کند و دوباره در آن می‌نویسد.
- **تراشه‌های پشتیبانی**[۲]: پردازنده توانایی انجام همه‌ی اعمال موردنیاز در یک سیستم کامپیوتری را ندارد. از این‌رو تراشه‌های پشتیبانی مورد نیاز هستند تا اعمالی را که پردازنده قادر به انجام آن نیست، برعهده بگیرند. این تراشه‌ها که به آنها کنترلر[۳] نیز گفته می‌شود، بخشی از سخت‌افزار را کنترل می‌کنند و کارهایی را انجام می‌دهند و پردازنده را برای انجام کارهای مهمتر آزاد می‌گذارند. برخی از مهمترین این تراشه‌ها که در سیستم IBM PC وجود داشت عبارتند از:

- **DMA Controller (8237):** DMA مخفف Direct Memory Access است و این تکنیک به لوازمی نظیر دیسک سخت اجازه می‌دهد که داده‌ها را به‌طور مستقیم بر روی حافظه بنویسند. این تکنیک ضمن آنکه باعث نقل و انتقال سریع داده‌ها بین لوازم و حافظه می‌شود، باعث آزاد ماندن پردازنده برای انجام سایر کارها می‌شود.

- **Interrupt Controller (8259):** وقفه[۴] راهی برای آگاه کردن پردازنده نسبت به آمادگی لوازم (نظیر دیسک سخت، صحفه کلید و غیره) برای نقل و انتقال داده است. با استفاده از وقفه‌ها، نیاز نیست که پردازنده منتظر آماده‌شدن یک وسیله برای تبادل داده باشد(با روش **سرکشی**[۵])، بلکه هر گاه وسیله آماده شد، یک سیگنال کنترلی به پردازنده می‌فرستد و باعث توقف در عملیات آن شده و درخواست سرویسی از پردازنده می‌کند. مثلاً با زدن یک کلید، صفحه کلید یک سیگنال وقفه به پردازنده می‌فرستد و زده شدن کلید را به آن اطلاع می‌دهد. وقوع وقفه از طرف لوازم مختلف باید دارای نظم باشد تا تداخلی بین آنها پیش نیاید و متناسب با الویت لوازم به آنها سرویس داده شود. تراشه‌ی ۸۲۵۹ وظیفه‌ی مدیریت وقفه‌های سخت‌افزاری را در سیستم PC برعهده دارد.

- **Programmable Peripheral Interface (8255):** این تراشه پردازنده را به لوازم جانبی، نظیر صفحه کلید یا مولد صوت[۶] متصل می‌کند و نقش یک میانجی را بازی می‌کند.

- **The Clock (8248):** اگر پردازنده را مغز سیستم کامپیوتری فرض کنیم، ساعت قلب آن خواهد بود. این قلب چند میلیون بار در ثانیه ضربان دارد (مثلاً 233 MHZ) و وسیله‌ای برای همزمان‌سازی پردازنده و لوازم دیگر سیستم کامپیوتری است.

- **The Timer (8253):** این تراشه به‌عنوان یک شمارشگر و نگهدارنده‌ی زمان در سیستم مورد استفاده قرار می‌گیرد. این تراشه دارای خروجی‌هایی است که سیگنالهای الکتریکی منظمی را صادر می‌کند و تناوب آن قابل برنامه‌ریزی است، اینکه سیگنال تایمر در چه فواصل زمانی صادر شود.–

- **حفره‌های گسترش**[۷]: برخی از تراشه‌های دیگر مورد نیاز در سیستم PC در این قسمت قرار داده می‌شوند. دو مورد مهم عبارتند از:

- **CRT Controller (6845):** این تراشه وظیفه‌ی کنترل لوله‌ی اشعه‌ی کاتدی[۸] را بر عهده دارد و در کارت ویدئویی قرار دارد. وظیفه‌ی این کنترلر، مدیریت و کدبندی نحوه‌ی نمایش اطلاعات در صفحه‌ی نمایش است.

- **Disk Controller (765):** این کنترلر هم در قسمت حفره‌های گسترش نصب می‌شود و توسط سیستم عامل نشان‌دهی شده و گرداننده‌ی دیسک را مدیریت می‌کند. حرکت دادن هد دیسک برای خواندن و نوشتن اطلاعات در بخشهای مختلف دیسک، وظیفه‌ی این کنترلر است.

- **حافظه**: در سیستمهای PC دو نوع حافظه وجود دارد. حافظه‌های فقط خواندنی یا ROM[۹]، که داده‌ها و کدهای غیرقابل تغییر در آن نگهداری می‌شود و نمونه‌ی مشخص آن، ROM-BIOS است. نوع دوم حافظه، حافظه‌ی خواندنی–نوشتنی است که به‌صورت تصادفی قابل دستیابی است و به آن RAM[۱۰] گفته می‌شود. این نوع حافظه محل نگهداری داده‌ها و پردازش آنها بوسیله‌ی پردازنده است. اولین نوع PC دارای 16KB حافظه بود و در PCهای امروزی امکان قرار دادن چند صد MB حافظه وجود دارد. تراشه‌های حافظه به صورت ماجولهایی هستند که در مکانهای مشخص در برد اصلی سیستم نصب می‌شوند و قابل کم و زیاد شدن هستند.

سیستم‌های عامل مختلف توانایی استفاده از همه‌یا بخشهایی از حافظه را دارند. سیستم عامل DOS نهایتاً می‌تواند 640KB حافظه را مورد استفاده قرار دهد و اگر مقدار حافظه بیشتر باشد، مابقی بلااستفاده می‌ماند. در PC، برای مدیریت حافظه و تخصیص آن به برنامه‌ها، حافظه‌به قطعه‌های[۱۱] 64KB تقسیم می‌شود. بنابر این کل حافظه‌های موجود در سیستم PC دارای نمای زیر است:

---

۱ Bus
۲ support chips
۳ controler
۴ interrupt
۵ polling
۶ speaker
۷ expansion slots
۸ Cathode Ray Tube (CRT)
۹ Read Only Memory
۱۰ Random Access Memory
۱۱ segment

شکل ۳-۱: نمایی از قسمت‌بندی حافظه‌ی PC

Block	Address	Contents
15	F000:0000 - F000:FFFF	ROM-BIOS
14	E000:0000 - E000:FFFF	Free for ROM Cartridge
13	D000:0000 - D000:FFFF	Free for ROM Cartridge
12	C000:0000 - C000:FFFF	Additional ROM-BIOS
11	B000:0000 - B000:FFFF	Video RAM
10	A000:0000 - A000:FFFF	Additional Video RAM (EGA/ VGA)
9	9000:0000 - 9000:FFFF	RAM from 576K to 640K
8	8000:0000 - 8000:FFFF	RAM from 512K to 576K
7	7000:0000 - 7000:FFFF	RAM from 448K to 512K
6	6000:0000 - 6000:FFFF	RAM from 384K to 448K
5	5000:0000 - 5000:FFFF	RAM from 320K to 384K
4	4000:0000 - 4000:FFFF	RAM from 256K to 320K
3	3000:0000 - 3000:FFFF	RAM from 192K to 256K
2	2000:0000 - 2000:FFFF	RAM from 128K to 192K
1	1000:0000 - 1000:FFFF	RAM from 64K to 128K
0	0000:0000 - 0000:FFFF	RAM from 0K to 64K

۱-۷ ثباتهای پردازنده[۱]

نکته‌ی مهم در مورد انواع پردازنده‌ها، مدل برنامه‌سازی آن است و در این میان، **ثباتها** نقش مهمی در برنامه‌سازی دارند. ثباتها، مکانهای حافظه‌ای در درون پردازنده هستند، و به این دلیل به مراتب سریع‌تر از RAM قابل دستیابی هستند. همچنین، ثباتها مکانهای ویژه‌ای برای انجام اعمال حسابی و منطقی بوسیله‌ی پردازنده هستند. برای برنامه‌سازی سیستم، آشنایی با این ثباتها، بسیار مهم است. زیرا، جریان اطلاعات بین برنامه، DOS و BIOS، با استفاده از ثباتها برقرار می‌شود. برنامه‌ها پارامترها را از طریق ثباتها به توابع واسطهای DOS یا BIOS ارسال نموده و نتایج و اطلاعات وضعیت را به عنوان مقادیر بازگشتی از این توابع دریافت می‌کنند.

از دیدگاه برنامه‌سازی سیستم با وجود معرفی مدلهای جدیدتر پردازنده‌های 80X86، این ثباتها از 8086 به بعد تغییر نکرده‌اند. این امر بدان دلیل است که BIOS و DOS وابسته به این پردازنده ایجاد شده‌اند و به‌دلایل سازگاری، بعداً تغییر اساسی از دید برنامه‌سازی سیستم در آنها داده نشده است. چون ثباتهای این پردازنده، ۱۶ بیتی بود، با استفاده از DOS، تحت پردازنده‌های ۳۲ بیتی 80386 یا 80486 هم فقط نصف ثبات، یا ۱۶-بیت آن قابل دستیابی است.

حال به انواع ثباتهای 8088 نگاه می‌کنیم. ثباتها به چند دسته تقسیم می‌شوند:

1. Common Registers:
   - AX (AH, AL) : Accumulator
   - BX (BH, BL) : Base
   - CX (CH, CL) : Count
   - DX (DH, DL) : Data
   - DI : Destination Index
   - SI : Source Index
   - SP :Stack Pointer
   - BP : Base Pointer
2. Segment Registers:

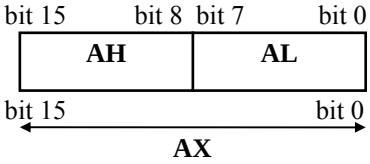
[۱] processor registers

- DS : Data Segment
- CS : Code Segment
- ES : Extra Segment
- SS :Stack Segment

3. Program Counter:
   - IP :Instruction Pointer

4. Flag Register
   - □□□□ODITSZ□A□P□C

ثباتهای مشترک: عموماً برای فراخوانی توابع DOS و BIOS مورد استفاده قرار می‌گیرند. همچنین این ثباتها بوسیله‌ی اعمال ریاضی (مثل جمع، تفریق و غیره) دستکاری می‌شوند. چهار ثبات ۱۶ بیتی AX، BX، CX و DX دارای ویژگی قابل قسمت‌شدن به دو بخش ۸ بیتی High و Low هستند. بنابر این، AX، AL و AH، هر سه قابل دستیابی هستند. مقدار AX با استفاده از مقادیر AL و AH قابل محاسبه است: AX = AH * 256 + AL



ثباتهای ۸ بیتی مثل CL یا CH برای خواندن و نوشتن داده‌های ۸ بیتی در حافظه و ثباتهای ۱۶ بیتی، مثل CX برای خواندن و نوشتن داده‌های ۱۶ بیتی قابل استفاده هستند.

ثبات پرچم[۱]**:** این ثبات برای برقراری ارتباط مابین دستورات پی‌درپی اسمبلی به‌کار می‌رود و برای این منظور وضعیت عملیات ریاضی و منطقی را نگهداری می‌کند. برای مثال با استفاده از Carry Flag یک برنامه می‌تواند تعیین کنید که آیا در هنگام جمع دو ثبات ۱۶ بیتی، حاصل بیش از ۶۵۵۳۵ شده است یا نه.

این ثبات هم ۱۶ بیتی است، ولی فقط ۹ بیت آن مورد استفاده قرار گرفته است که عبارتند از:

- Bit 0: CF (Carry Flag)
- Bit 2: PF (Parity Flag)
- Bit 4: AF (Auxiliary Flag)
- Bit 6: ZF (Zero Flag)
- Bit 7: SF (Sign Flag)
- Bit 8: TF (Trap Flag)
- Bit 9: IF (Interrupt Flag)
- Bit 10: DF (Direction Flag)
- Bit 11: OF (Overflow Flag)

در برنامه‌سازی سیستم با استفاده از زبانهای سطح بالا فقط دو پرچم ZF و CF مورد استفاده قرار می‌گیرند. زیرا اغلب توابع DOS و BIOS از آنها برای مشخص کردن رخداد خطا در هنگام عملیات استفاده می‌کنند.

ثبات نشانی[۲]**:** تعداد مکانهای حافظه که یک پردازنده می‌تواند به آنها دسترسی داشته باشد، به این ثبات بستگی دارد. هر قدر تعداد بیتهای این ثبات بیشتر باشد، حداکثر حافظه‌ی قابل دسترسی، بیشتر می‌شود. اگر این ثبات ۱۶ بیتی باشد، حداکثر ۶۵۵۳۵ مکان حافظه، قابل دستیابی است. به همین دلیل است که پردازنده‌های اولیه فقط قادر به دسترسی به 64K حافظه بودند. برای نشانی‌دهی 1MB حافظه، این ثبات باید دارای حداقل ۲۰ بیت باشد. اما در زمان طراحی ۸۰۸۸، امکان استفاده از ثبات نشانی ۲۰ بیتی وجود نداشت. از این‌رو، راه حل دیگری مورد استفاده قرار گرفت و دو عدد ۱۶ بیتی متفاوت برای تشکیل نشانی ۲۰ بیتی مورد استفاده قرا می‌گیرد. نخستین عدد در یک **ثبات قطعه**[۳] قرار می‌گیرد و دومین عدد در یک ثبات دیگر یا در یک مکان حافظه. به این ترتیب نشانی دارای دو بخش قطعه و مبدأ[۴] خواهد بود. نشانی قطعه، که در ثبات قطعه وجود دارد، شروع یک قطعه حافظه را نشان می دهد. نشانی مبدأ، شماره‌ی مکان حافظه را در درون قطعه مشخص می‌کند. چون نشانی مبدأ یک عدد ۱۶ بیتی است، پس یک قطعه نمی‌تواند دارای بیش از ۶۵۵۳۵ یا 64K مکان حافظه‌باشد.

---

۱ flag register
۲ address register
۳ segment register
٤ offset

چهار نوع ثبات قطعه در ۸۰۸۸ وجود دارد:

1. CS: Code Segment
2. DS: Data Segment
3. SS: Stack Segment
4. ES: Extra Segment

CS، از ثبات IP (Instruction Pointer) به عنوان نشانی مبدأ استفاده می‌کند. به IP، PC (Program Counter) هم گفته می‌شود و در حقیقت نشانی دستور بعدی برنامه را در حافظه نشان می‌دهد و به‌طور خودکار با اجرای یک دستور، یک واحد افزایش می‌یابد. **DS**، نشانی قطعه‌ای از حافظه را که شامل داده‌های مورد دستیابی برنامه است را نشان می‌دهد. **SS**، نشانی شروع پشته[۱] است و **ES**، بوسیله‌ی برخی از دستورات اسمبلی برای نشانی‌دهی بیش از 64K داده یا انتقال داده‌ها مابین قطعه‌های مختلف مورد استفاده قرار می‌گیرد.

۱-۸ درگاه‌ها[۲]

درگاه‌ها، واسطی بین پردازنده و سایر اجزاء سخت‌افزار سیستم هستند. هر درگاه مشابه یک دریچه‌ی ۸ بیتی ورودی یا خروجی است که به بخش خاصی از یک جزء سخت‌افزاری متصل است و با مقادیر بین ۰ تا ۶۵۵۳۵ قابل نشانی‌دهی است. درگاه‌ها در حقیقت، ثباتهایی در اجزاء سخت‌افزاری هستند که برای ارتباط با آن، مقداری در آن نوشته یا از آن خوانده می‌شود. پردازنده از گذرگاه برای دسترسی به درگاه‌ها استفاده می‌کنند و در پردازنده‌های 80x86 از دو دستورالعمل IN و OUT می‌توان برای دسترسی به درگاه‌ها در برنامه‌های اسمبلی استفاده نمود.

۱-۹ وقفه‌ها

وقفه مکانیسمی است که پردازنده را مجبور می‌کند تا اجرای برنامه‌ی جاری را متوقف نموده و یک روال خاصی را که وظیفه‌ی مدیریت وقفه[۳] را دارد را اجرا کند. وقفه‌ها هم برای کنترل سخت‌افزارها استفاده می‌شوند و هم برای برقرار ارتباط بین برنامه و توابع DOS و BIOS. به این ترتیب دو نوع وقفه داریم:

- **وقفه‌های نرم‌افزاری[۴]:** این نوع وقفه‌ها برای فراخوانی توابع DOS یا BIOS استفاده می‌شوند. در این حالت، توابع فراخوانی شده، همانند یک زیرروال[۵] برنامه عمل می‌کنند و پس از پایان اجرای تابع، به برنامه‌ی اصلی برگشت انجام می‌شود. توابع DOS و BIOS به ترتیب خاصی در حافظه قرار می‌گیرند و تشکیل یک جدول را می‌دهند که به آن **جدول بردار وقفه**[۶] می‌گویند. این جدول دارای دو ستون ورودی یا شماره تابع و نشانی شروع کد مربوطه در حافظه است. فراخوانی توابع با استفاده از ورودی مربوطه در این جدول انجام می‌شود. برای مثال، اگر تابع DOS 21H را فراخوانی کنیم، پردازنده نشانی شروع کد تابع 21H را از جدول بردار وقفه‌ها بدست آورده و به آن پرش می‌کند.
- **وقفه‌های سخت‌افزاری[۷]:** این وقفه‌ها بوسیله‌ی اجزاء سخت‌افزاری تولید می‌شوند و از طریق کنترلر وقفه‌ها به پردازنده منتقل می‌شوند. برخی از وقفه‌های سخت‌افزاری قابل ناتوان‌سازی[۸] هستند. یعنی باعث می‌شود که یک سخت‌افزار نتواند به پردازنده وقفه بدهد. اما برخی از وقفه‌ها غیرقابل ناتوان‌سازی هستند که به آنها NMI[۹] یا Trap گفته می‌شود.

۱-۷ محاوره‌ی سیستم[۱۰]

برای انجام کارها در سیستم PC، اجزاء مختلف سخت‌افزاری و نرم‌افزاری با هم کار می‌کنند. برای مثال برای خواندن یک کلید فشار داده شده بر روی صفحه کلید، موارد زیر مطرح هستند:

- **سخت‌افزار صفحه کلید:** وقتی یک کلید فشار داده می‌شود، سخت‌افزار صفحه کلید(در صورت توانا بودن وقفه‌های سخت‌افزاری)، وقفه‌ی شماره‌ی 09H را به پردازنده می‌فرستد. پردازنده پس از دریافت این وقفه، براساس الویت وقفه‌ها و اگر درحال اجرای یک وقفه با الویت بالاتر نباشد، روال مدیریت وقفه‌ی شماره‌ی 09H را اجرا خواهد کرد.
- **روال مدیریت صفحه‌کلید BIOS:** روالی که پردازنده برای مدیریت وقفه‌ی 09H فراخوانی می‌کند، مربوط به توابع BIOS است. این روال با توجه به اینکه چه کلیدی فشار داده شده است، کد مربوطه را بدست می‌آورد و معتبر بودن آن را بررسی می‌کند.

---

۱ stack
۲ ports
۳ interrupt handler
۴ software interrupts
۵ subroutine
۶ interrupt vector table
۷ hardare interrupt
۸ disable
۹ Non-Maskable Interrupt
۱۰ system interaction

- **بافر صفحه‌کلید**: پس از آنکه روال مدیریت صفحه‌کلید، کد کلید فشار داده شده را معتبر تشخیص داد، آنرا در یک بافر ۱۶ بایتی در RAM ذخیره می‌کند. این بافر مثل یک صف است و کلیدها به ترتیب فشار داده شدن در آن قرار می‌گیرند و چون ۱۶ بایتی است، در صورتی که بیش از ۱۶ کلید فشار داده شود، پر خواهد شد و در این صورت است که صدای بوق شنیده می‌شود.
- **وقفه‌ی صفحه‌کلید BIOS**: مرحله‌ی بعد، خواندن کاراکتر از بافر فوق و آماده‌سازی آن برای برنامه است. این کار بوسیله‌ی وقفه‌ی شماره‌ی 16H قابل انجام است. پس با فراخوانی این وقفه‌ی نرم‌افزاری، کاراکتر موجود در بافر و وضعیت خوانده می‌شوند. برای این منظور می‌توان از دستور INT اسمبلی استفاده کرد.
- **در سطح DOS:** همچنین در DOS نیز توابعی برای کار کردن با صفحه‌کلید وجود دارد که جزو توابع 21H هستند. این توابع، کارهای بیشتری را در مقایسه به تابع BIOS 16H انجام می‌دهند. برای نمونه کلید زده شده را بر روی صفحه نمایش، نشان می‌دهند.