

به نام خداوند بخشنده و مهربان

زبان ماشین و اسمبلی

جلسه سوم آموزشی

جلسه اول مجازی

رئوس مطالب:

- انواع زبان های برنامه نویسی: مدل مرجع OSI
- ابزارهای اجرای برنامه های نوشته شده به زبان های سطح بالا
- ساختمان داخلی کامپیوتر
- پردازنده های 8086
- مفهوم خط لولگی (Pipelining)

انواع زبان های برنامه نویسی:

برنامه های مختلف سیستم های کامپیوتری برای برقراری ارتباط بین سخت افزار یک سیستم و انسان با هدف اجرای فرامین مورد نظر انسان بوجود آمدند.

* زبان های برنامه نویسی، عاملی برای دمیدن روح در کالبد بی جان سیستم های کامپیوتری و برنامه ریزی آنها هستند.

هر زبان برنامه نویسی دارای یک محیط نرم افزاری برای وارد کردن متن برنامه، اجرا و رفع اشکال آن است.

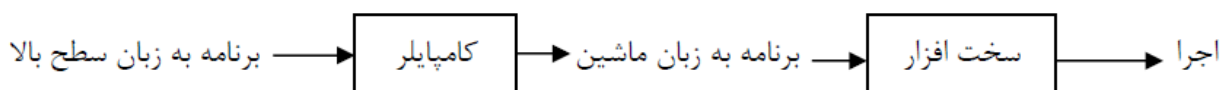
- عموماً زبانهای برنامه نویسی را به پنج نسل تقسیم می کنند:

- نسل اول زبان ماشین ☛ زبان صفرو یک
 - نسل دوم زبانهایی مانند اسمبلی ☛ قابل فهم تر برای انسان
 - نسل سوم زبانهایی مانند PASCAL (پاسکال)، BASIC، C، FORTRAN (فورترن) ☛ دستورات قابل فهم تر برای انسان و نیاز به کامپایلرها
 - نسل چهارم مثل زبانهای Oracle (اوراکل) و فاکس پرو ☛ زبانهای این نسل برنامه نویس را قادر می سازند تا کارهای سطح بالاتر و بیشتری را توسط کد کمتری انجام دهد. (نزدیک به محاوره انسان)
 - نسل پنج زبانهایی مانند prolog و ویژوال بیسیک ☛ استفاده از شبکه های عصبی و هوش مصنوعی و همچنین استفاده از Agent ها به منظور انجام کارها در کامپیوتر از دیگر ویژگی های این نسل از زبان ها هستند.
- * از نسل دوم زبان های برنامه نویسی، کامپیوترها برای فهمیدن فرامین برنامه، نیاز به یک مترجم دارند. (دستورات باید به زبان ماشین تبدیل شوند).

ابزارهای اجرای برنامه های نوشته شده به زبان های سطح بالا:

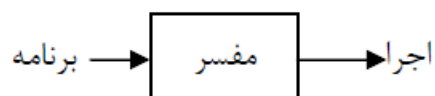
۱- کامپایلر (Compiler) :

در زبان های کامپایلری در ابتدا تنها یکبار کد ها به صورت کامل ترجمه شده و سپس برنامه از روی فایل کامپایل شده اجرا می شود.



۲- مفسر (Interpreter):

در زبان های مفسری کد ها به صورت خط به خط تفسیر و اجرا می شوند.



*تفاوت های زبان های برنامه نویسی کامپایلری و مفسری:

- ✓ زبان های کامپایلری وابسته به سیستم عامل هستند یعنی روی همان سیستم عاملی که کامپایل شده اند قابل اجرا هستند. اما زبان های مفسری وابسته به سیستم عامل نیستند.
- ✓ از آنجا که زبانهای کامپایلری به یکبار ترجمه شده و چندین بار اجرا میشوند سریعترند. زبان های مفسری در هر بار اجرا باید خط به خط ترجمه شوند.
- ✓ اشکال زدایی در زبان های مفسری بسیار ساده تر است. زیرا در زبان های مفسری هنگامی که برنامه خط به خط اجرا می شود، در هر بخشی از برنامه که مشکلی وجود داشته باشد قابل مشاهده و رفع می باشد اما در زبان های کامپایلری، باید کل برنامه بررسی شود که بسیار دشوارتر از زبان های مفسری است.

☛ مثال هایی از زبان های کامپایلری: Cobol, C#, C++, Visual Basic

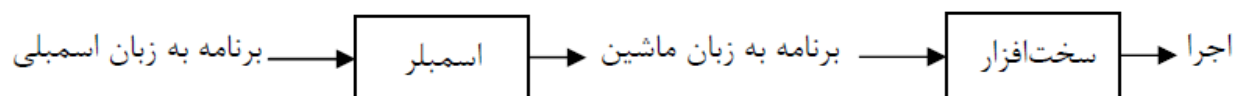
☛ مثال هایی از زبان های مفسری: Matlab و Python, Perl, PHP, JavaScript, VBScript

**** نکته ****

- زبان اسمبلی بسیار نزدیک به زبان ماشین است. با این تفاوت که برای نوشتن دستورات به جای کدهای باینری (0 و 1) ، از یکسری کلمات به زبان انگلیسی استفاده می شود.

اجرای برنامه های اسمبلی:

نرم افزاری به نام اسمبلر (مثل کامپایلر) ، برنامه را به طور کامل به زبان ماشین ترجمه کرده و سپس برنامه اجرا می شود.



**** نکته ****

چند اصطلاح مهم:

یکی از ویژگی های مهم در یک سیستم کامپیوتری ظرفیت حافظه (Memory) آن است. به خاطر داریم که یک بیت، یک عدد باینری است که شامل مقادیر 0 و 1 می باشد. از آنجه که هر بیت یک فضای حافظه را ایجاد می کند می توان گفت:

Bit	0	یک بیت
Nibble	0000	نصف بایت یا 4 بیت
Byte	0000 0000	یک بایت یا 8 بیت
Word	0000 0000 0000 0000	2 بایت یا 16 بیت

ظرفیت حافظه در کامپیوتر با مفاهیمی همچون Kilobyte ، Megabyte ، Gigabyte ، Terabyte بیان می‌شود.

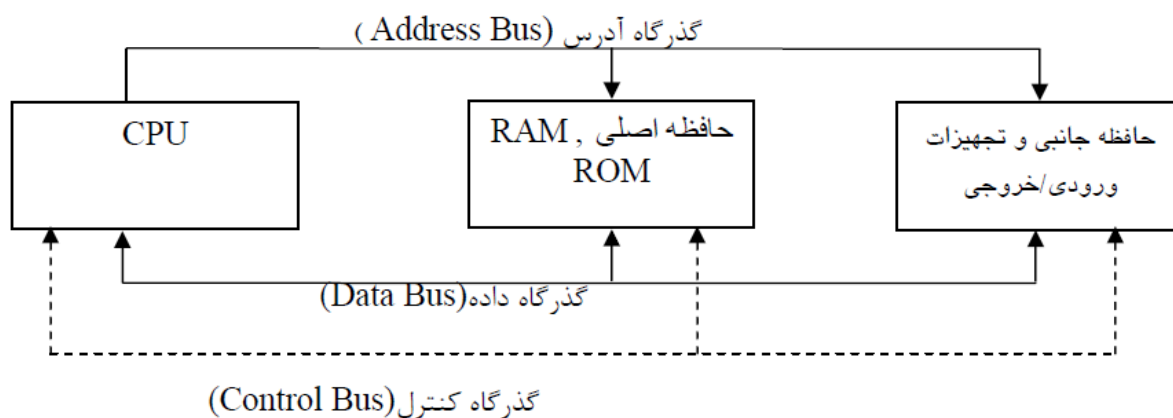
Kilobyte	2^{10} byte
Megabyte	2^{20} byte
Gigabyte	2^{30} byte
Terabyte	2^{40} byte

به این معنا که :

اگر یک کامپیوتر دارای 16 Megabyte حافظه باشد، یعنی دارای 2^{24} byte خانه حافظه است.

$$2^{24} \rightarrow 16 \text{ Megabyte} \times 2^{20} = 2^4 \times 2^{20} = 2^{24} \text{ byte}$$

ساختمان داخلی کامپیوتر:



ساختمان داخلی کامپیوتر به سه بخش اصلی تقسیم می شود:

۱- CPU (واحد پردازشگر مرکزی)

۲- Memory (حافظه)

۳- I/O devices (تجهیزات ورودی / خروجی)

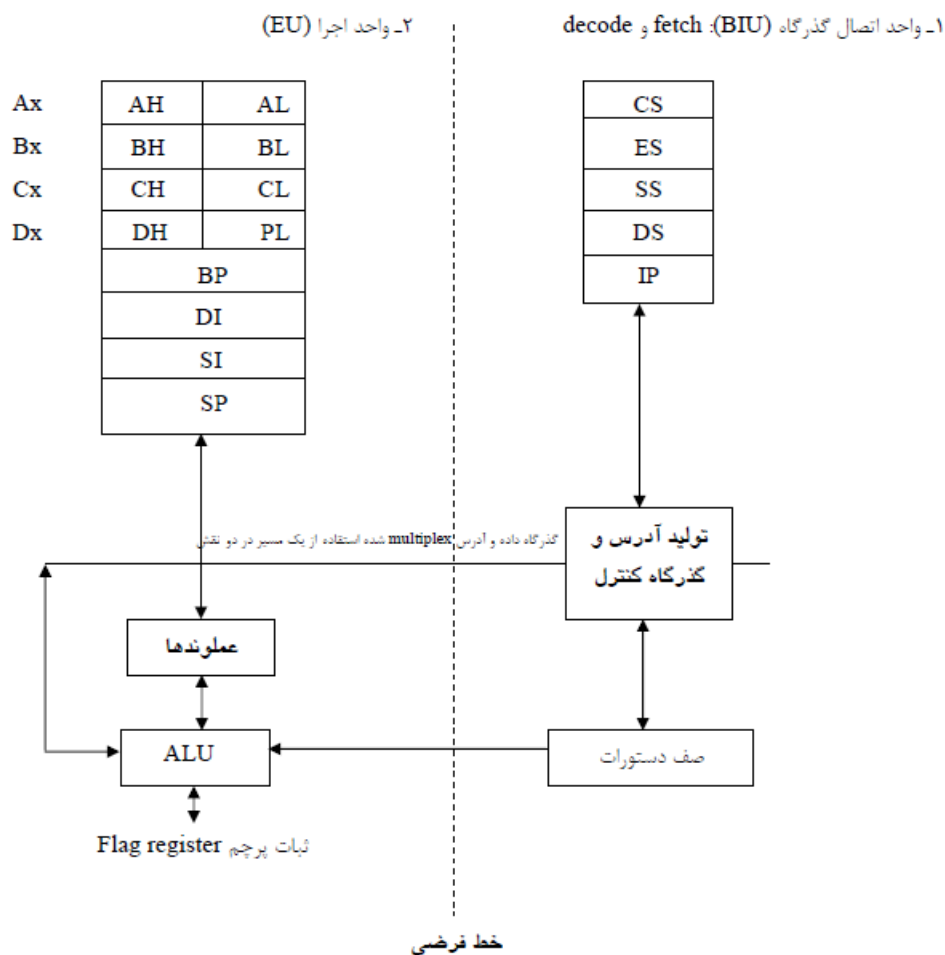
CPU اطلاعات ذخیره شده درون حافظه را اجرا می کند. دستگاه های ورودی و خروجی نیز با CPU در ارتباطند. در واقع، CPU از طریق خطوطی به نام BUS (گذرگاه) با حافظه و I/O متصل می شود. درون گذرگاه یک کامپیوتر، اطلاعات از مکانی به مکان دیگر منتقل می شوند. در هر کامپیوتر سه نوع گذرگاه وجود دارد:

۱- گذرگاه آدرس: برای آن که CPU به اطلاعات درون خانه های حافظه دسترسی پیدا کند، باید آدرس آن خانه ها را مشخص کند و برای این منظور، این آدرس ها را در گذرگاه آدرس قرار می دهد. ریزپردازنده به کمک خطوط آدرس، آدرس خاصی از حافظه را جهت خواندن یا نوشتن مشخص می سازد.

۲- گذرگاه داده: گذرگاه هایی که داده های بازیابی شده از حافظه را در اختیار CPU قرار می دهند و داده های پردازش شده را به کامپیوتر منتقل می کنند، گذرگاه داده نامیده می شوند. به کمک گذرگاه داده ها اطلاعات بین CPU و RAM و وسایل I/O رد و بدل می شود .

۳- گذرگاه کنترل: مسیرهایی هستند که سیگنال های کنترلی برای نظارت بر کلیه عملیات کامپیوتر از طریق آن ها ارسال می شود. واحد کنترل برای اعمال نظارت بر بخش های مختلف کامپیوتر، از این خطوط استفاده می کند.

پردازنده های 8086:



CPU از نظر منطقی به دو بخش تقسیم می شود:

۱- واحد اجرایی (EU): وظیفه EU، اجرای دستورات در یافتی از واحد BIU است. EU، شامل قسمت های محاسباتی و منطقی (ALU) و واحد کنترل (CU) و تعدادی ثبات است. این قسمت های اساس به منظور اجرای دستورات و عملیات محاسباتی و منطقی فراهم شده اند.

۲- واحد رابط گذرگاه (BIU): وظیفه واکشی و رمزگشایی دستورات و عملوندها در حافظه را بر عهده دارد.

مفهوم خط لولگی (Pipelining)

* اگر چه CU و BIU با هم به طور موازی کار می کنند، همواره BIU یک قدم از CU جلوتر است.

بالاترین دستورالعمل صف، دستورالعمل جاری می باشد. مادامیکه EU در حال اجرای یک دستور است BIU دستور بعدی را از حافظه واکشی می کند. این واکشی ها همراه با اجرای دستورات در یک زمان باعث افزایش سرعت پردازش می شود. که به این مفهوم خط لولگی می گویند.

Nonpipelined: (8085)

Fetch1	Exe1	Fetch2	Exe2	Fetch3	Exe3
--------	------	--------	------	--------	------

Pipelined: (8086)

F1	E1			
	F2	E2		
		F3	E3	
			F4	E4

** دلیل اینکه CPU به دو بخش منطقی تقسیم شده است، ایجاد قابلیت خط لولگی است.

** عمل Fetch (واکشی) را BIU و عمل exe (اجرا) را EU انجام می دهد.

** هدف ← افزایش سرعت و کارایی CPU