

زبان ماشین و برنامه سازی سیستم

فصل اول : اصول زبان ماشین و ساختار کامپیوتر

مدرس:

سید محمد بیدکی



دانشگاه خلیج فارس

دانشکده فنی و مهندسی جم

تابستان ۱۳۹۳

مقدمه

- زبانی که کامپیوتر با آن کار می کند، **زبان ماشین** نامیده می شود که از یک سری 0 و 1 تشکیل می گردد.
- چون کار با کدهای زبان ماشین دشوار است، زبانی سمبولیک به نام **اسمبلی (Assembly)** ایجاد گردید که بسیار به زبان ماشین نزدیک بوده ولی برای بیان دستورات آن از حروف انگلیسی استفاده شده و لذا کار کردن با آن ساده تر می باشد.
- برای ترجمه دستورات زبان اسمبلی که شبیه کلمات انگلیسی هستند به معادل 0 و 1 دستورات زبان ماشین، یک نرم افزار **اسمبلر (Assembler)** نیاز است.
- زبان اسمبلی هر کامپیوتر تابع ساختار سخت افزاری و پروسسور آن کامپیوتر می باشد در نتیجه اسمبلی کامپیوترهای مختلف با یکدیگر متفاوت می باشد.
- لذا زبان ماشین و زبان اسمبلی را **زبان سطح پایین (Low-level)** گویند.
- زبان های برنامه نویسی که به سخت افزار خاصی وابسته نیستند، زبان سطح بالا **(High-level)** نامیده می شوند.

ویژگی برنامه نویسی اسمبلی

- نوشتن برنامه به زبان ماشین یا زبان اسمبلی باعث می شود که:
 1. از امکانات سخت افزاری کامپیوتر بهتر استفاده شود.
 2. بدلیل استفاده از امکانات داخلی کامپیوتر، برنامه اسمبلی نسبت به برنامه با زبان های سطح بالا سریع تر اجرا شود و جای کمتری نیز در حافظه بگیرد.
 3. برای کار با زبان ماشین یا زبان اسمبلی، کاربر می باید اطلاعات کلی راجع به معماری کامپیوتر و پروسسور آن داشته باشد. (در صورتی که در زبان های سطح بالا چنین چیزی لازم نیست).
- برخی از برنامه ها مثل برنامه های سیستم عامل، برنامه های مقیم در حافظه و سرویس های وقفه تقریباً همه به زبان اسمبلی نوشته می شوند.

معرفی اصول زبان ماشین و زبان اسمبلی

- تمام کامپیوترها برنامه را به زبان ماشین اجرا می نمایند.
- در حقیقت دستورات زبان ماشین یک سری بیت های ۰ یا ۱ است که کامپیوتر به وسیله ی آن می فهمد چه باید انجام دهد.
- لذا یک برنامه از دید کامپیوتر ترکیبی از دستورات زبان ماشین به فرم بیت های ۰ و ۱ است.
- به عنوان مثال، جدول صفحه بعد، یک سری دستورات به زبان اسمبلی و معادل آن ها به زبان ماشین در کد باینری ۰ و ۱ یا معادل آن ها در کد هگزادسیمال را نشان می دهد.

نمونه ای از دستورات زبان اسمبلی و ترجمه آن

دستورات زبان اسمبلی (برنامه اسمبلی)	دستورات زبان ماشین	
	کد باینری	کد هگزادسیمال
MOV AL, ADDRES-2	1010 0000 0000 0001 0000 0100	A0 01 04
SUB AL, 30H	0010 1100 0011 0000	2C 30
ADD AL, BL	0000 0010 1100 0011	02 C3
HLT	1111 0101	F5

- همان طوری که در جدول فوق ملاحظه می شود، زبان ماشین از نظر ما بسیار نامفهوم و پیچیده است، لذا به جای زبان ماشین، ما از زبان اسمبلی یا زبان سمبولیک استفاده می کنیم.

ساختار یک دستور اسمبلی

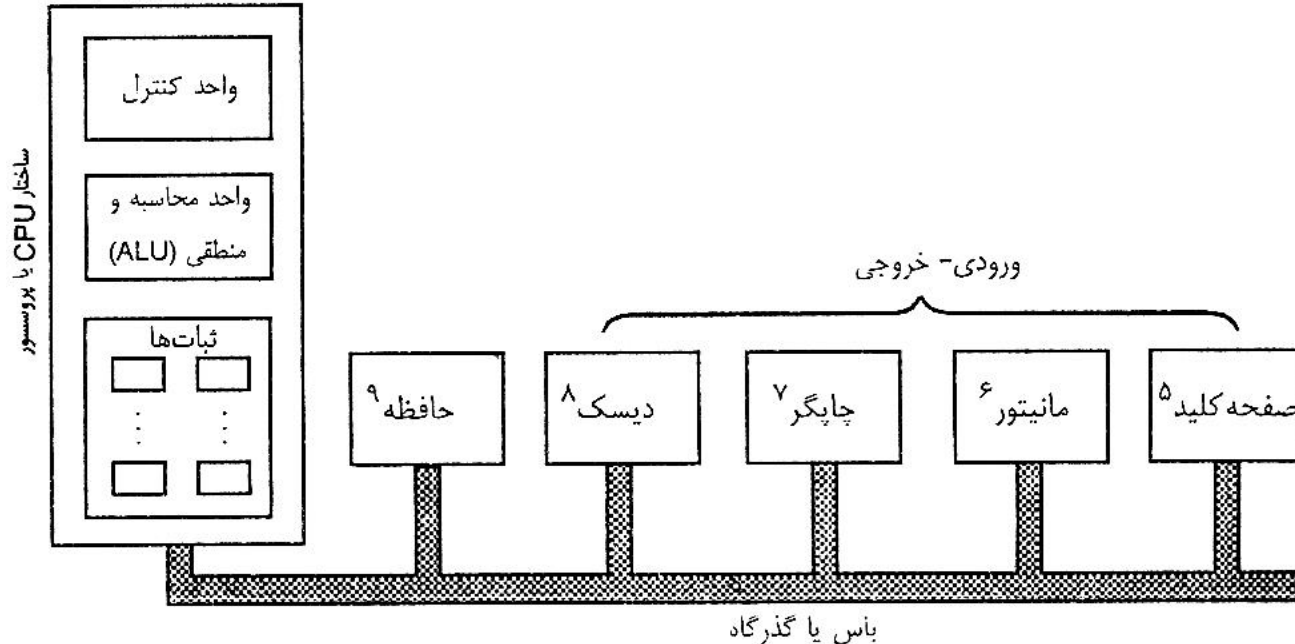
- هر دستور زبان اسمبلی، معادل یک سری ۰ و ۱ در کد باینری یا زبان ماشین است.
 - هر برنامه به زبان اسمبلی شامل تعدادی دستور می باشد.
 - هر دستور ماشین دارای دو قسمت است :
1. آدرس اطلاعات و یا داده ها که اصطلاحاً عملوند یا اپراند (Operand) گویند،
 2. کد اجرا (Operational Code).

ساختار کامپیوتر

- کامپیوتر به ۲ بخش تقسیم می شود:
- سخت افزار، شامل مدارهای الکترونیکی و الکترومکانیکی
- نرم افزار، شامل برنامه های سیستم عامل، کامپایلر، اسمبلر و...
- لیست دستورات را برنامه کامپیوتر و محلی که این دستورات ذخیره شده اند، حافظه می نامند.
- کامپیوتر دستگاهی است که اطلاعات 0 و 1 را به عنوان ورودی گرفته و طبق دستوراتی که در حافظه آن قرار دارد، یکسری عملیات بر روی اطلاعات مذکور انجام می دهد و خروجی را تولید می کند.

ساختار کامپیوتر ...

- هر کامپیوتری از واحدهای ورودی-خروجی، حافظه، محاسبه و منطقی ALU (Arithmetic & Logic Unit)، باس یا گذرگاه، و واحد کنترل تشکیل شده است. (شکل ۱-۲)
- مجموعه واحدهای محاسبه و منطقی، واحد کنترل و ثبات ها را، پروسسور یا CPU (Central Processing Unit) می نامند که وظیفه اجرای دستورات را به عهده دارد.

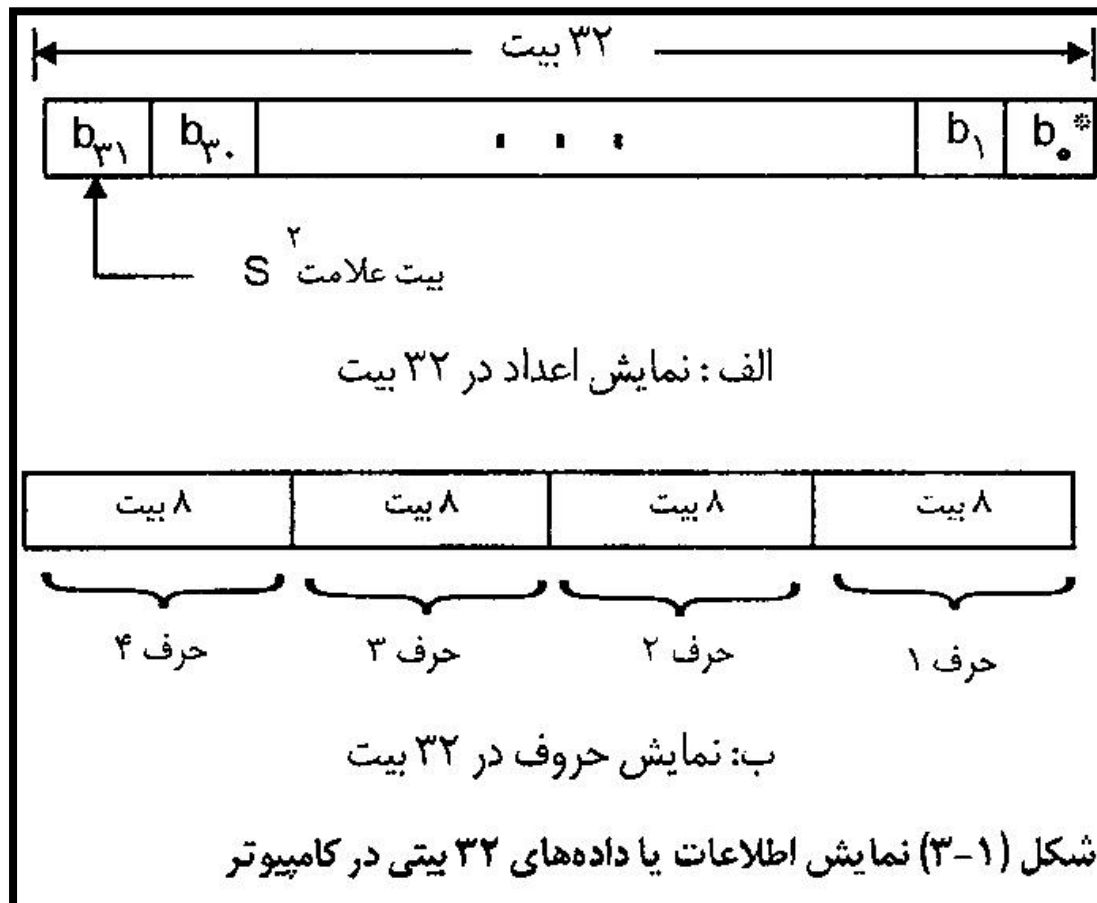


واحد پروسسور یا CPU

- وظیفه پروسسور یا CPU، اجرای دستورات برنامه ی کامپیوتر می باشد.
- پروسسور از واحدهای محاسبه و منطقی (ALU)، کنترل و یک سری ثبات (Register) تشکیل شده است.
- ثبات ها یک سری حافظه های بسیار کوچک در ابعاد یک یا چند بایت هستند که جهت ذخیره موقت اطلاعات یا اعداد و انجام عملیات بر آنها استفاده می شوند.
- به عنوان مثال فرض کنید می خواهیم دو عدد که در دو ثبات، قرار دارند را با هم جمع کنیم. برای این کار می باید این دو عدد به واحد محاسبه و منطقی ALU آورده شوند، که در آن جا با هم جمع گردند و نتیجه جمع ممکن است به حافظه برگردد و یا برای استفاده بعدی در ثبات های پروسسور ذخیره گردد. بدیهی است عملیات ریاضی و منطقی دیگر مانند ضرب، تقسیم، مقایسه و ... نیز در واحد محاسبه منطقی ALU، پروسسور انجام می شود.

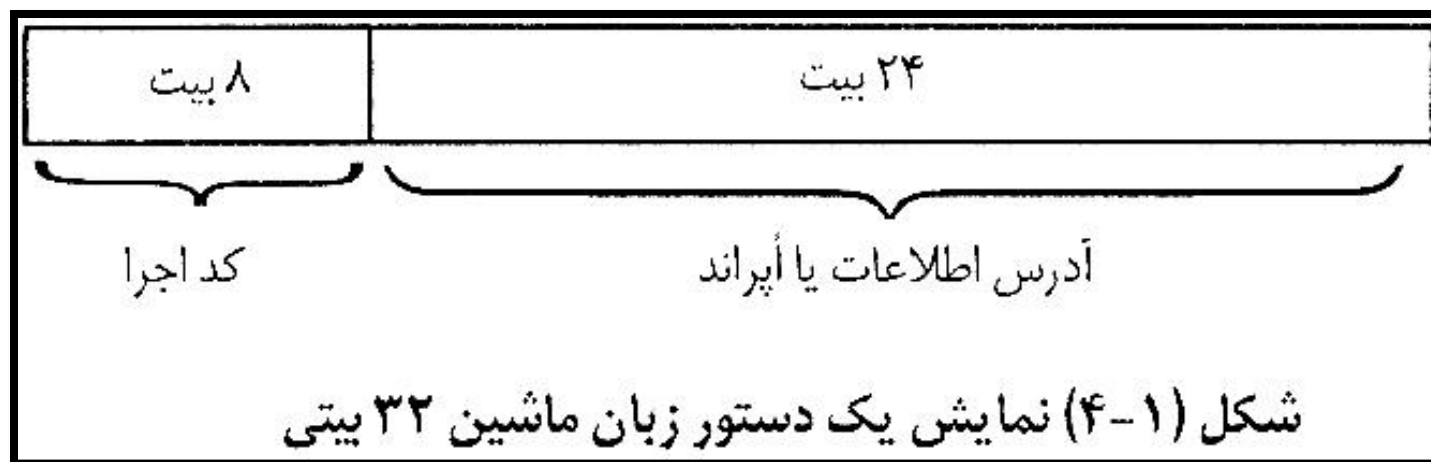
واحد ورودی

- اطلاعات را وارد کامپیوتر می کند.
- الف: داده ها که شامل اعداد، حروف و علامات می باشند. شکل (۱-۳)

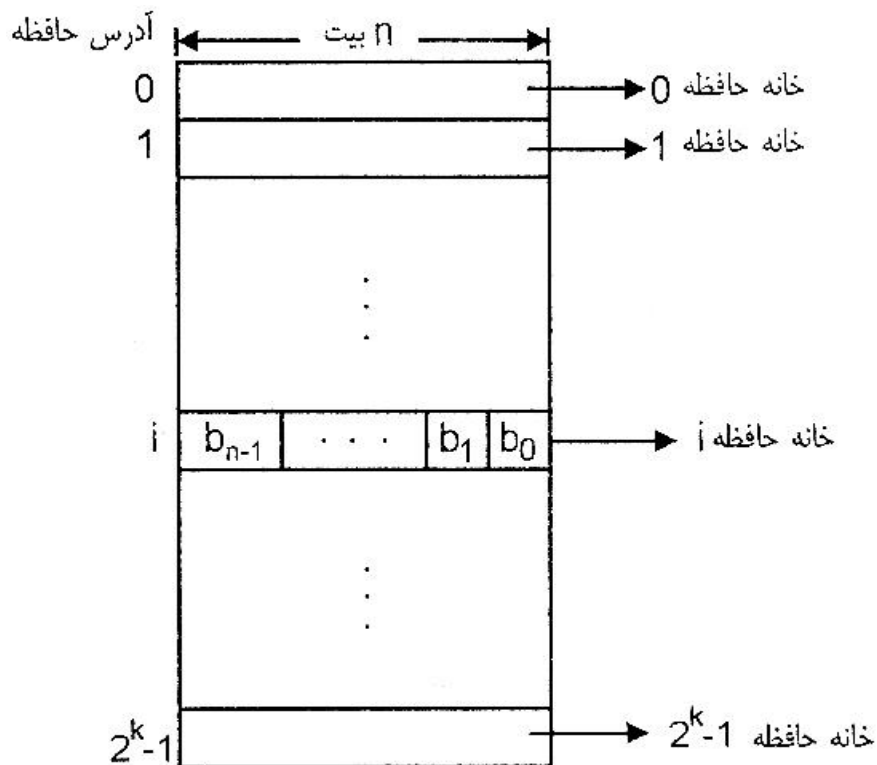


واحد ورودی ...

- ب: دستورات ماشین یا دستورات برنامه کامپیوتر که مشخص می نمایند که کامپیوتر چه باید بکند. همانطوری که از شکل (۱-۴) ملاحظه می شود، هر دستور دارای قسمتی است به نام کد اجرا و قسمت دیگر که آدرس اطلاعات یا عملوند (یا اپراند) می باشد.



واحد حافظه



- حافظه دارای یک سری سلول های حافظه از جنس نیمه هادی، می باشد که هر کدام یک بیت اطلاعات را ذخیره می کند.
- مجموعه ای از ۸ بیت سلول های حافظه که بطور همزمان خوانده یا نوشته می شوند، یک بایت اطلاعات یا یک خانه (یک بایتی) می نامند.
- برای دستیابی به خانه های حافظه به هر بایت از حافظه، یک آدرس تخصیص داده می شود.

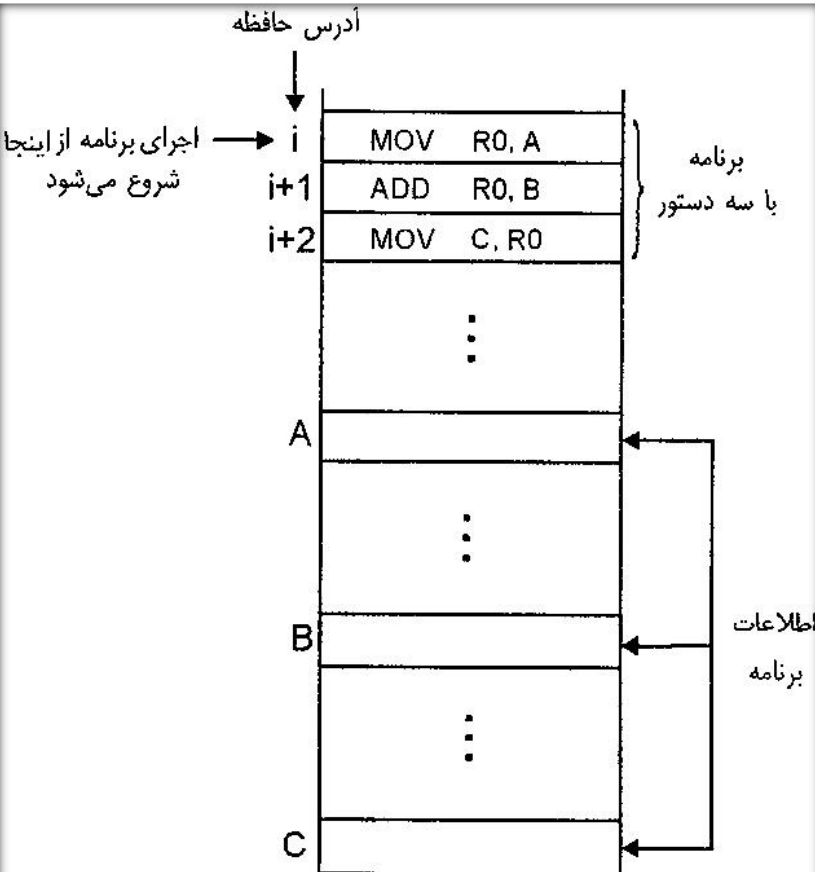
شکل (۱-۵) خانه های حافظه و آدرس آنها

واحد حافظه ...

- برنامه در خانه های حافظه به آدرس های i , $i+1$, $i+2$ و... ذخیره شده و اطلاعات در خانه های حافظه به آدرس های A , B , C قرار دارند.

- اگر بخواهیم اطلاعات یک خانه حافظه را بخوانیم، می بایست آدرس آن خانه به حافظه داده شود و فرمان خواندن از واحد کنترل برای حافظه ارسال گردد.

- جهت نوشتن اطلاعات در یک خانه حافظه کافیست آدرس آن خانه به حافظه داده شود و اطلاعات مذکور به حافظه ارسال گردد و فرمان نوشتن نیز از طرف واحد کنترل به حافظه فرستاده شود.



شکل (۱-۶) نمایش چگونگی ذخیره اطلاعات و دستورات در حافظه

واحد خروجی

- نتیجه محاسبات کامپیوتر می تواند به دستگاه خروجی ارسال گردد، که در اختیار کاربر قرار گیرد.
- مثالی از دستگاه خروجی: مانیتور، چاپگرهای مکانیکی، لیزری و ...

باس یا گذرگاه

- همواره ارتباط مداومی بین پروسسور و هر یک از دستگاه های ورودی-خروجی و حافظه برقرار می باشد. برای این ارتباط راحت ترین وسیله ارتباط مستقیم از طریق سیم های بین آن هاست.
- برخی از دستگاه ها، از یک تعداد سیم ارتباطی، بطور مشترک استفاده می نمایند که این سیم های مشترک را باس یا گذرگاه می نامند.
- تعداد سیم های باس تابع نوع دستگاه هاست؛ در کامپیوترهای معمولی: بین ۴۰ تا ۲۰۰ سیم
- پروسسور با هر واحد کامپیوتر از طریق باس ارتباط برقرار می کند.
- قسمتی از باس به نام **باس آدرس (Address Bus)** معروف است که پروسسور آدرس دستگاههای ورودی خروجی و یا حافظه را روی آن قرار می دهد.
- قسمتی از باس نیز، به نام **باس داده، یا دیتا (Data Bus)** معروف می باشد، که اطلاعات از طریق آن بین حافظه و دستگاه های ورودی خروجی و CPU انتقال می یابد.

باس یا گذرگاه ...

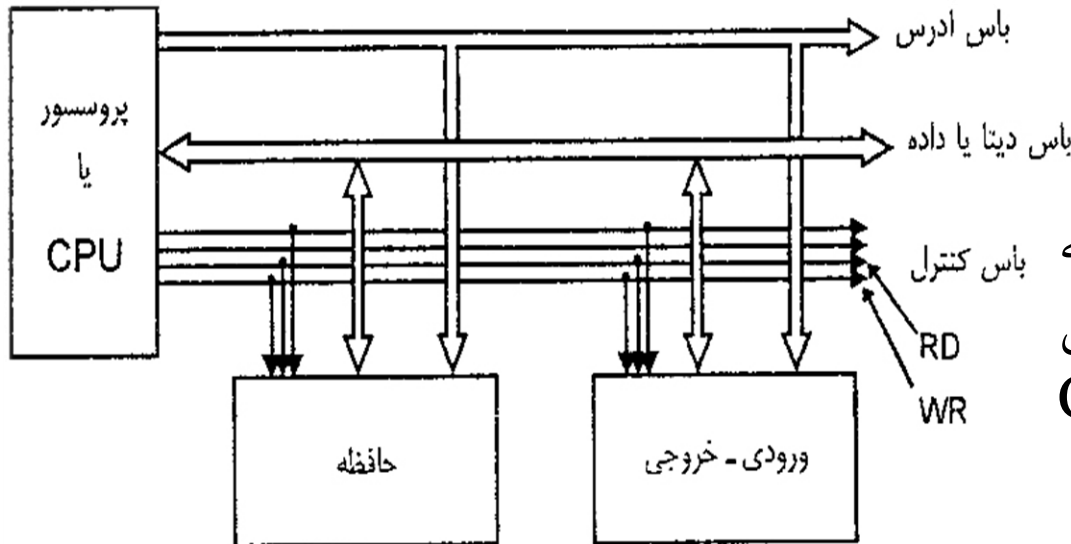
- علاوه بر این پروسوسور، فرمان هایی برای حافظه و واحدهای ورودی خروجی، می فرستد که عبارتند از:

1. فرمان خواندن RD، که اطلاعات از حافظه یا واحد ورودی، خوانده می شود و به CPU می رود.

2. فرمان نوشتن WR، که اطلاعات از پروسوسور بر واحد خروجی یا حافظه، نوشته می شود.

3. فرمان های کنترل دیگر

- پس قسمتی از باس کامپیوتر که برای انتقال فرمان های کنترلی است، **باس کنترل** (Control Bus) نامیده می شود.



شکل (۱-۷) قسمت های مختلف باس کامپیوتر

طرز اجرای دستورات برنامه در پروسسور

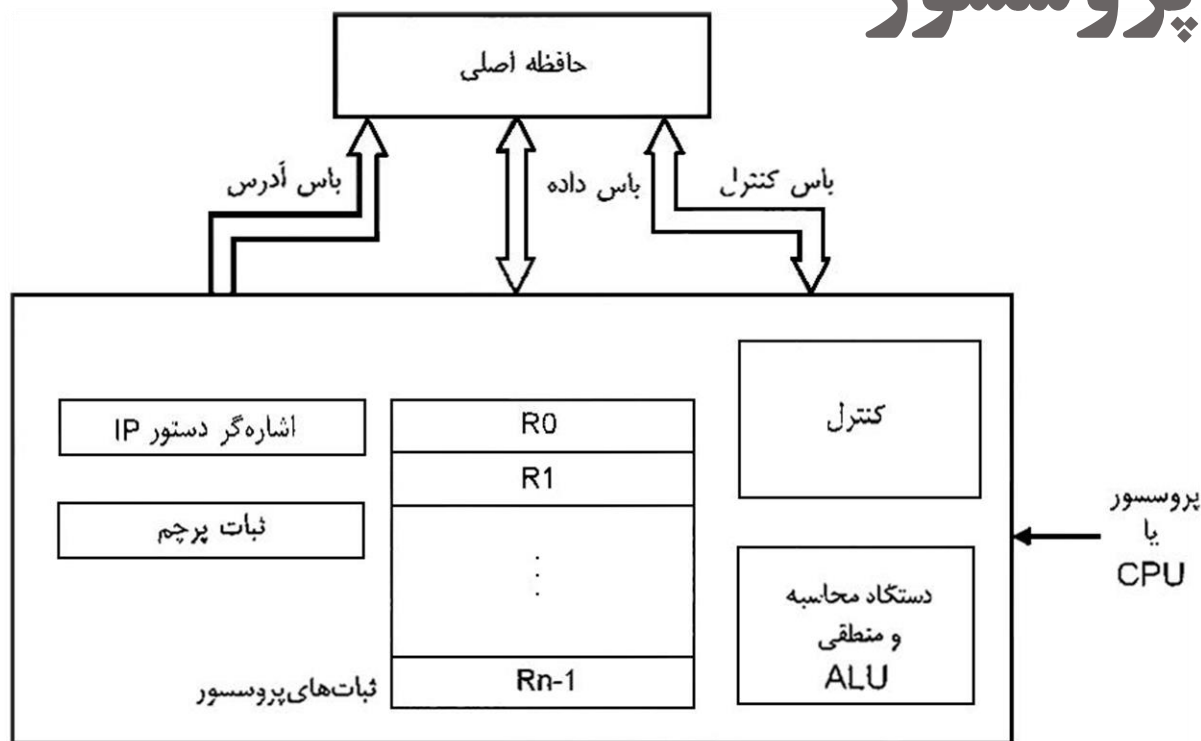
- کاربر برنامه را می نویسد و در حافظه بار می کند و سپس پروسسور یکی یکی دستورات را می خواند و از کد اجرای آن متوجه می شود که چه باید بکند و سپس دستور را اجرا می نماید. و به همین ترتیب دستور بعدی... .

• مثال: `Add R0,[5000]`

مراحل:

1. واکنشی دستور از حافظه، قرار دادن در CPU و بررسی اینکه پروسسور چکار کند.
2. دادن آدرس به حافظه، صدور فرمان خواندن به حافظه، خواندن محتوای بایت ۵۰۰۰ و انتقال به پروسسور
3. محتوای خانه ۵۰۰۰ با محتوای ثابت R0 جمع می شود و در R0 قرار می گیرد.

ساختار پروسور



شکل (۸-۱) ساختار اصولی پروسورها یا CPU، و ارتباط آن با حافظه

- ثبات‌ها در کامپیوترهای شخصی به نام AX, BX, CX, DX, SI, DI, ... خوانده می‌شوند و در میکروکنترلرها A, B, R0, R1, R2, ..., R7 می‌باشند که دستورات بر روی محتوای آن‌ها اجرا می‌شوند.

اشاره گر دستور IP (Instruction Pointer)

- در پروسسور یک اشاره گر دستور IP نیز وجود دارد، که آدرس دستور بعدی که باید از حافظه خوانده شود را می دهد.
- شمارنده IP یک شماره می اندازد که آدرس دستور بعدی در حافظه را مشخص می نماید و در نتیجه دستور بعدی به پروسسور آورده شده و اجرا می گردد.
- ثبات IP در بعضی کامپیوترها PC (Program Counter) نامیده می شود.

ثبات پرچم FR (Flag Register)

- ثبات دیگری به نام ثبات پرچم FR یا ثبات وضعیت برنامه (PSW Program Status Word) در پروسسور وجود دارد، که بیت های آن برای تشخیص وضعیت نتیجه محاسبات به کار می روند.
- تعدادی از بیت های پرچم عبارتند از:
 1. بیت پرچم تشخیص صفر ZF: اگر نتیجه آخرین محاسبه صفر شود، بیت پرچم تشخیص صفر ZF برابر یک می گردد یعنی $ZF=1$ ، در غیر این صورت $ZF=0$ می شود.
 2. بیت پرچم نقلی CF: اگر نتیجه آخرین محاسبه دارای بیت نقلی باشد، مقدرا $CF=1$ می گردد، در غیر این صورت $CF=0$ می شود.
 3. بیت پرچم علامت SF: اگر نتیجه آخرین محاسبه مثبت باشد $SF=0$ و اگر نتیجه منفی شود $SF=1$ می گردد.
 4. بیت های پرچم توازن PF برای تشخیص اشتباه، و بیت نقلی کمکی AC جهت عملیات جمع BCD.... نیز وجود دارند.

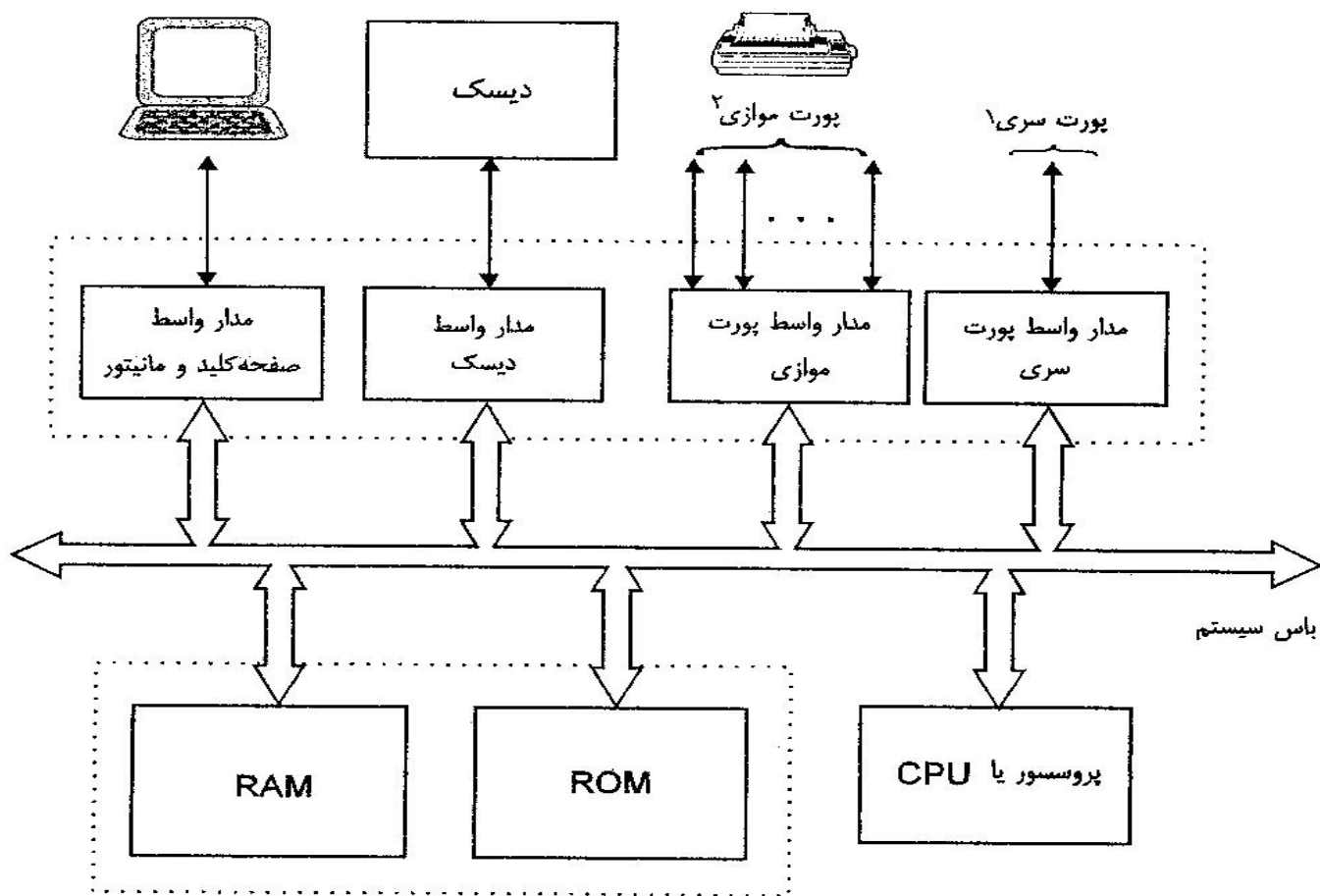
سیکل دستور یا زمان اجرای دستور

- پروسسور دستورات را یکی یکی از حافظه خوانده و اجرا می کند.
- نقطه شروع اجرای برنامه زمانیست که آدرس اولین دستور برنامه توسط اشاره گر دستور IP، به حافظه داده شود و فرمان خواندن نیز از طرف واحد کنترل به حافظه ارسال گردد.
- بعد از مدتی که برابر زمان دستیابی حافظه است، محتوای خانه ی حافظه ای که آدرس آن داده شده، از حافظه خوانده شده و وارد پروسسور می گردد، که این عملیات را **سیکل واكشی (Fetch Cycle)** دستور نامند.
- اکنون که دستور داخل پروسسور قرار دارد، در واحد محاسبه و منطقی ALU اجرا می گردد، که عملیات اجرای دستور را **سیکل اجرا (Execution Cycle)** نامند.
- مجموع سیکل واكشی و سیکل اجرا را، سیکل دستور (Instruction Cycle) یا زمان اجرای دستور نامند.

ارتباط کامپیوتر با دستگاه های ورودی-خروجی

- برای ارتباط کامپیوتر با دستگاه های ورودی خروجی یا دستگاه های جانبی، به مدارهای واسط نیاز است.
- مدار واسط اطلاعات را به فرمت قابل استفاده برای CPU و دستگاه های ورودی خروجی تبدیل می نماید.
- به عنوان مثال صفحه کلید کامپیوتر، یک دستگاه ورودیست که کار مدار واسط آن این است که تشخیص دهد کدام کلید فشار داده شده و در نتیجه کد آسکی و کد اسکن، نظیر کلید فشار داده شده را تولید نماید.
- در شکل (۱-۹) پروسوسور از طریق مدارهای واسط با دستگاه های ورودی-خروجی و پورت های سری و موازی ارتباط دارد.

ارتباط کامپیوتر با دستگاه های ورودی-خروجی



حافظه اصلی

شکل (۱-۹) مثالی از یک سیستم کامپیوتری

حافظه اصلی

- حافظه اصلی کامپیوتر دارای دو قسمت به نام های حافظه با دسترسی تصادفی (RAM) و حافظه فقط خواندنی (ROM) می باشد.
- حافظه RAM (Random Access Memory) قابل خواندن و نوشتن است و در هر لحظه میتوان اطلاعات آن را با بیان آدرس بایت مورد نظر خواند یا نوشت.
- حافظه ROM (Read-Only Memory) فقط قابل خواندن است و برای ذخیره سازی اطلاعاتی مفید است که نباید در زمان انجام عملیات پروسسور تغییر بیابد.
- اطلاعات موجود بر حافظه RAM ناپایدار است ولی اطلاعات حافظه ROM پایدار یا ماناست.
- یعنی با قطع برق، حافظه RAM پاک می شود ولی ROM نه!

سیستم عامل (Operating System)

- سیستم عامل مجموعه ای از برنامه های کامپیوتری به زبان ماشین (یا معادل آن زبان اسمبلی) است که وظیفه نظارت و هدایت عملیات اساسی کامپیوتر را به عهده دارد.
- سیستم عامل، سرویسهای اولیه و سطح پایین را پیاده سازی نموده و در اختیار کاربر قرار می دهد. و کاربر برای انجام این امور تنها سرویس های مربوطه را فراخوانی می نماید.
- سیستم عامل نقش گرفتن فرمان از کاربر، سازمان دهی و مدیریت حافظه، کنترل برنامه ها و همچنین ارائه سرویس برای برنامه های کاربر را به عهده دارد.
- به عبارت دیگر، سیستم عامل رابط بین برنامه کاربر و سخت افزار کامپیوتر است.
- به عنوان مثال: اگر بخواهیم برنامه ای را بر روی دیسک ذخیره کنیم، اولین مسئله ای که مطرح می شود این است که آیا دیسک جا دارد یا نه. اگر جا نباشد باید بعضی فایل ها حذف شوند که این کار توسط سیستم عامل مدیریت می شود ...