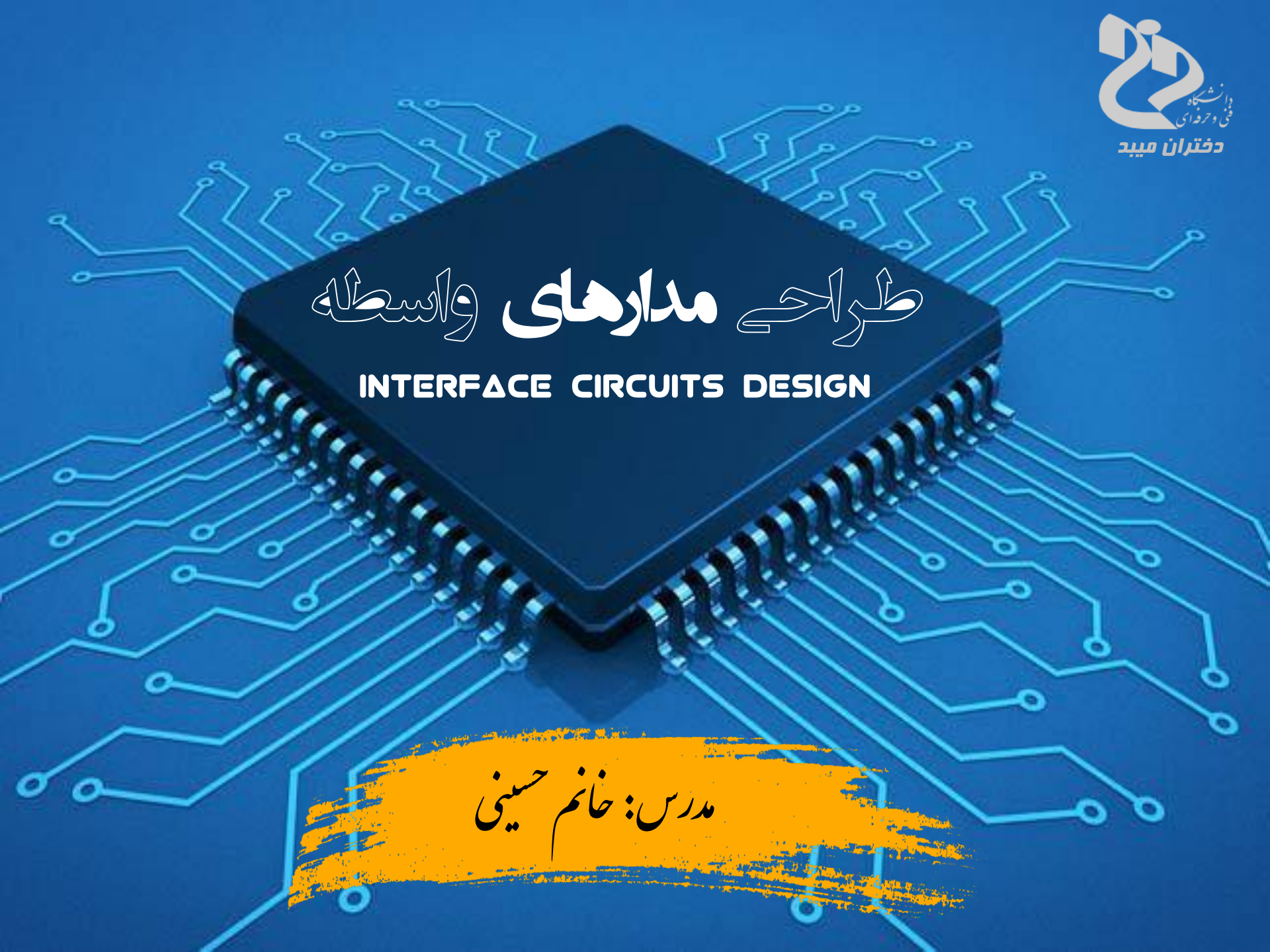
دانشگاه فنی و حرفه‌ای
دختران میبد

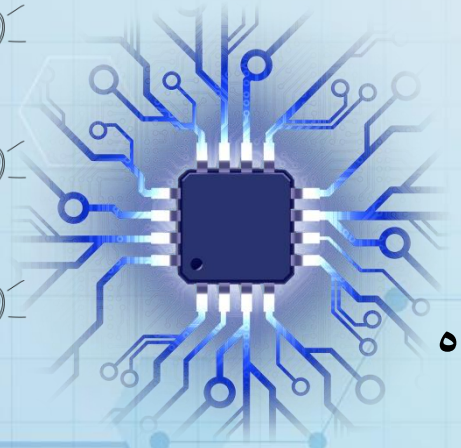
طراحی مدارهای واسطه

INTERFACE CIRCUITS DESIGN

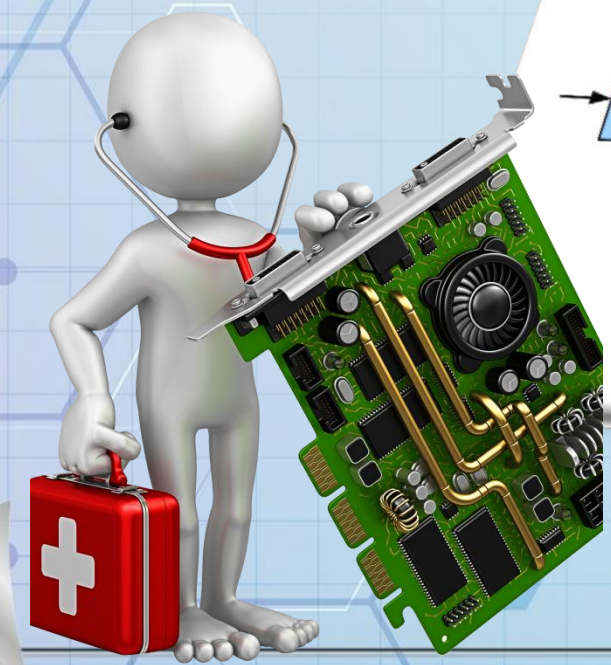


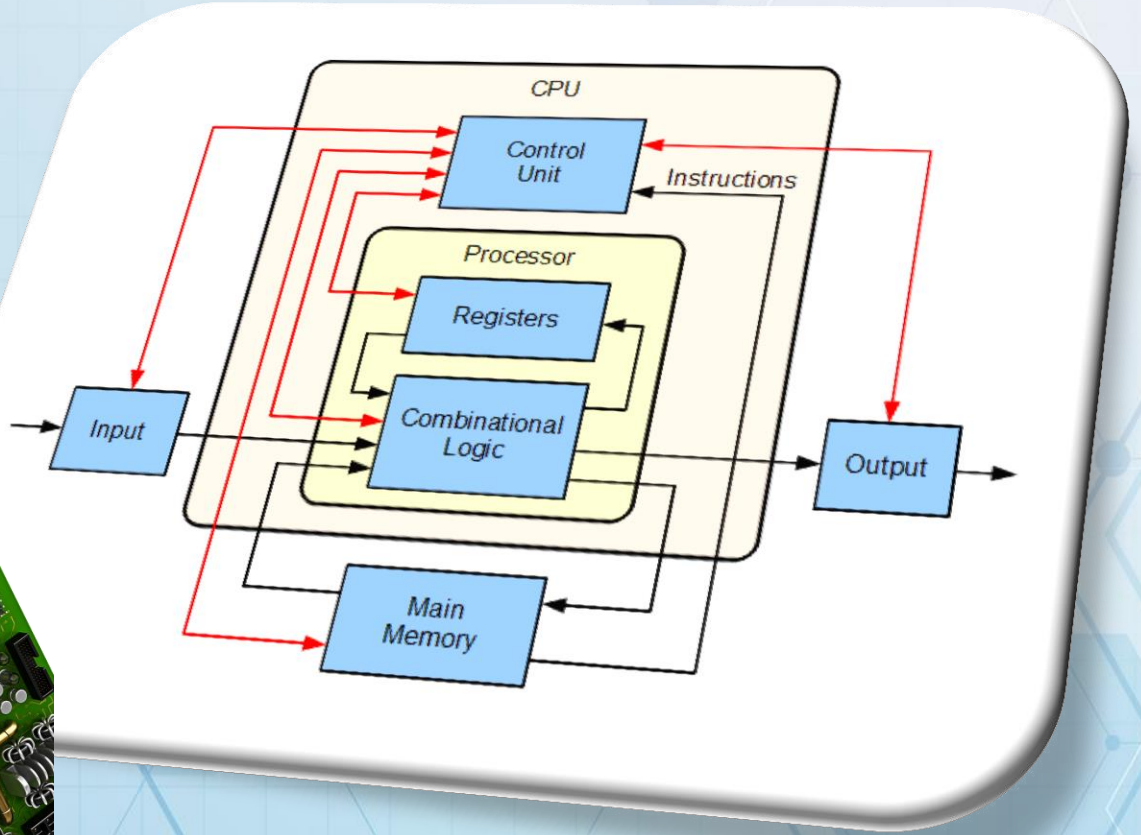
مدرس: خانم حسینی

فهرست مطالب



INTERFACE CIRCUITS DESIGN

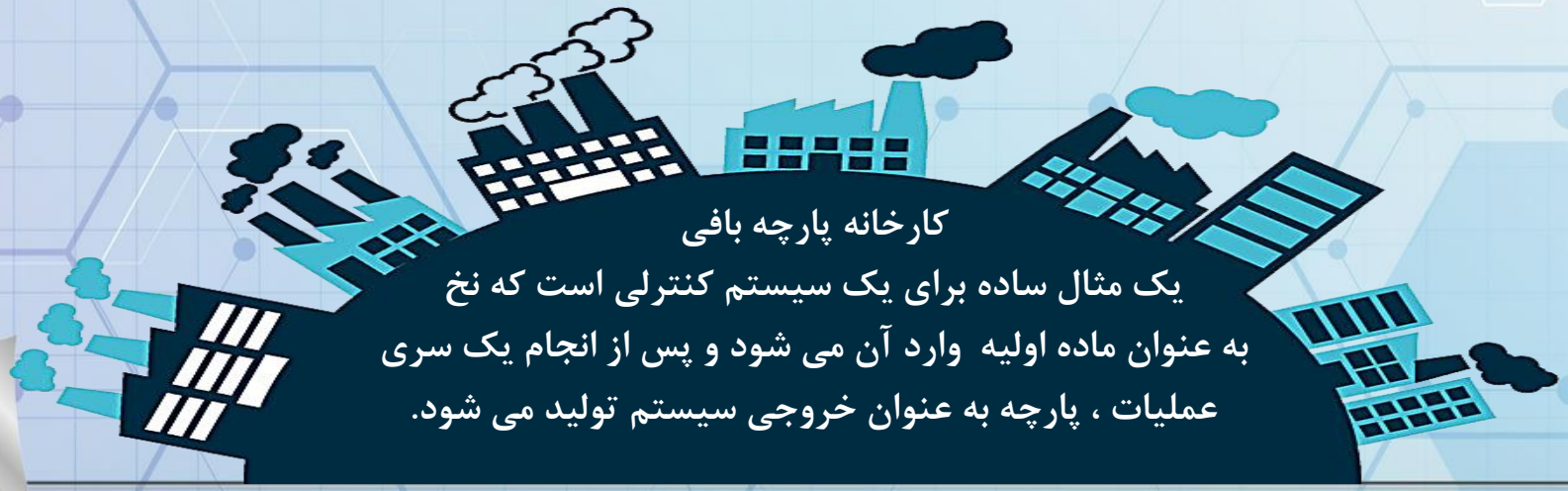




آشنایی با سیستم های مبتنی بر پردازنده

به طور کلی هر سیستم کنترلی دارای سه بخش ورودی ، پردازشگر و خروجی است . چنین سیستمی داده ها را از دنیای خارج دریافت می کند(بخش ورودی)، روی این داده ها عملیاتی انجام می دهد (بخش پردازش) و بر اساس این پردازش ها خروجی لازم را تولید می کند (بخش خروجی).

اجزای یک سیستم کنترلی



کارخانه پارچه بافی

یک مثال ساده برای یک سیستم کنترلی است که نخ به عنوان ماده اولیه وارد آن می شود و پس از انجام یک سری عملیات ، پارچه به عنوان خروجی سیستم تولید می شود.

تصمیم گیری های منطقی

بخش اساسی پردازش در سیستم های کنترلی هستند . در درس مدارهای منطقی با نحوه طراحی بخش پردازشگر سیستم های الکترونیکی با استفاده از مدارات منطقی ترتیبی و ترکیبی آشنا شده اید . عملکرد پردازنده ها از نظر کلی با مدارات منطقی یکسان هستند و تنها چگونگی ورودی و خروجی داده ها و نحوه انجام پردازش روی آنهاست که باعث متفاوت شدن ویژگی های آن می شود.

ایده اصلی ایجاد و گسترش پردازنده ها ، قابلیت برنامه پذیری آنهاست . یک سیستم کنترلی مبتنی بر پردازنده دارای تعدادی ورودی و خروجی است که تمام پردازش هایی که قرار است روی ورودی ها انجام شود به صورت مجموعه ای از دستورات نرم افزاری که برنامه نام دارد ، به سیستم داده می شود

ایده اصلی ایجاد و گسترش پردازنده ها ، قابلیت برنامه پذیری آنهاست . یک سیستم کنترلی مبتنی بر پردازنده دارای تعدادی ورودی و خروجی است که تمام پردازش هایی که قرار است روی ورودی ها انجام شود به صورت مجموعه ای از دستورات نرم افزاری که برنامه نام دارد ، به سیستم داده می شود

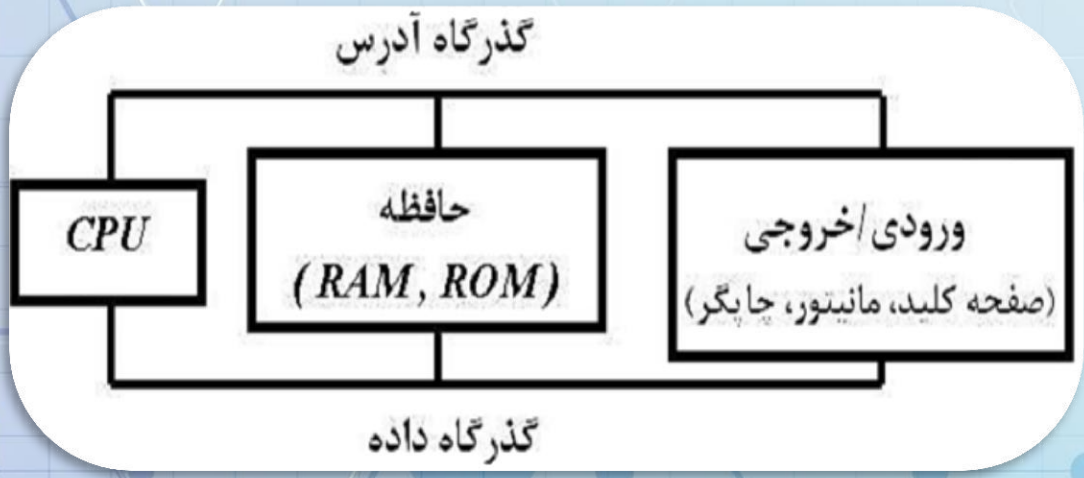
سیستمهای منطقی برنامه پذیر

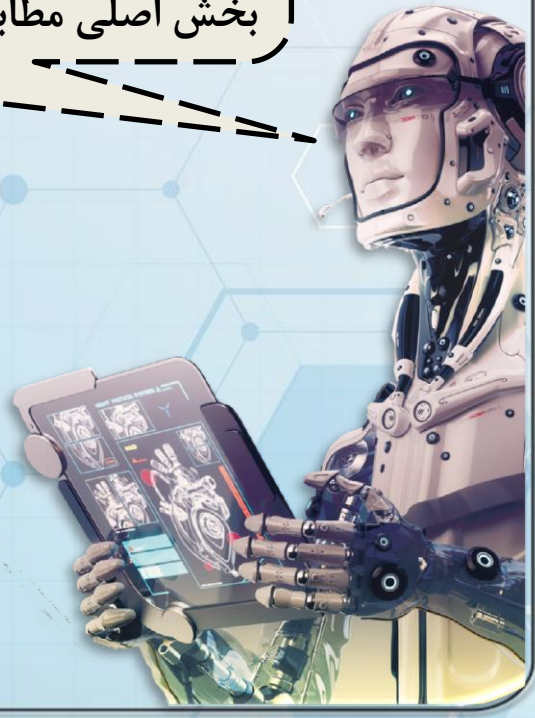
در این سیستم ها ، به جای طراحی بخش پردازشگر سیستم کنترلی به صورت سخت افزار سفارشی که مشکلات ذکرشده را به دنبال دارد ، پردازش را به یک برنامه (نرم افزار) می سپاریم که هم طراحی و پیاده سازی و اشکال زدایی و هم تغییر کارکرد آن نسبت به سخت افزار ، بسیار ساده تر و کم هزینه تر می باشد . وظیفه ی پردازنده اجرای این برنامه است .

اکنون که با ضرورت وجود پردازنده ها آشنا شدیم ، به بررسی چگونگی عملکرد آنها می پردازیم . در ادامه با ساختار کلی یک کامپیوتر به عنوان یک سیستم مبتنی بر پردازنده و سپس با چند مثال با نحوه اجرای برنامه ها آشنا خواهید شد .

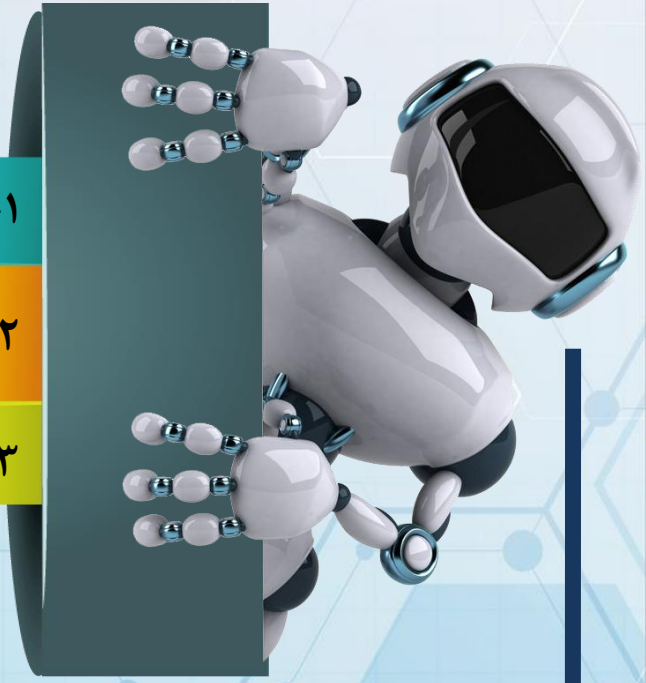
یک سیستم پردازنده ای چه اجزایی دارد ؟

شما با کامپیوتر به عنوان یک سیستم پردازنده ای آشنایی دارید . قطعه ی **CPU** داخل کامپیوتر همان پردازنده ی مورد نظر ماست ، اما کامپیوتر اجزای دیگری مانند دیسک سخت ، **RAM**، کارت گرافیکی ، صفحه کلید ، نمایشگر و ... دارد. تمام این اجزا را می توان در سه بخش اصلی مطابق شکل زیر طبقه بندی کرد .





اجزای یک سیستم مبتنی بر پردازنده



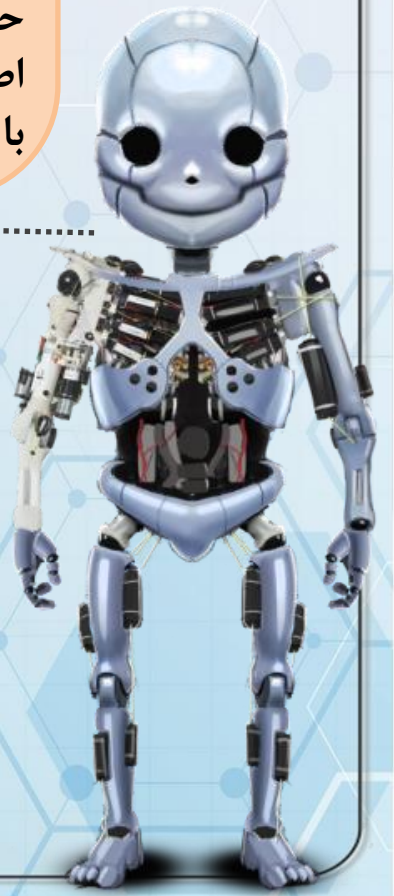
۱- واحد پردازنده مرکزی (CPU)

۲- حافظه

۳- وسایل ورودی و خروجی (I/O)

مجموعه ی دستورات یا برنامه ای که قرار است پردازش های سیستم ما را انجام دهد ، در واحد حافظه ذخیره می‌شود . کار واحد پردازش مرکزی اجرای خط به خط این برنامه است . کار واحدهای ورودی/خروجی ، تبادل اطلاعات با دنیای خارج است .

از نظر ساختاری می توان کامپیوتر را مانند یک انسان در نظر گرفت که داده هایی را از حواس خود (ورودی) مانند بویایی و بینایی می گیرد ، در مغز خود (پردازنده) با توجه به اطلاعاتی که در ذهن (حافظه) دارد ، پردازش هایی را روی آنها انجام می دهد و متناسب با آنها عکس العمل (خروجی) نشان می دهد .



مانند اجزای بدن که از طریق رگ ها و اعصاب با هم مرتبط هستند ، اجزای داخلی یک کامپیوتر از طریق مجموعه ای از سیم ها به نام باس یا گذرگاه با یکدیگر در ارتباط می باشند .

در اسلاید های بعد به شرح مختصری از بخش های ذکر شده می پردازیم .

کار پردازنده چیست؟

کار پردازنده اجرای مجموعه ای از دستورات یا برنامه ای است که در حافظه ذخیره شده است . برای این کار پردازنده باید :

- دستورات را به ترتیب از حافظه دریافت یا واکشی کند (Fetch)
- معنای آن را درک کند (Decode)
- آن را اجرا کند (Execute)

برای درک سه مرحله‌ی فوق ، کار یک نوازنده‌ی موسیقی را در نظر بگیرید . او برای اجرای یک آهنگ باید سمبل نت موسیقی را از روی یک صفحه که نت های یک آهنگ روی آن نوشته شده‌اند، بخواند (Fetch) ، این که آن سمبل نشان دهنده‌ی کدام نت است را درک کند (Decode) ، و سپس آن نت را بنوازد (Execute) .

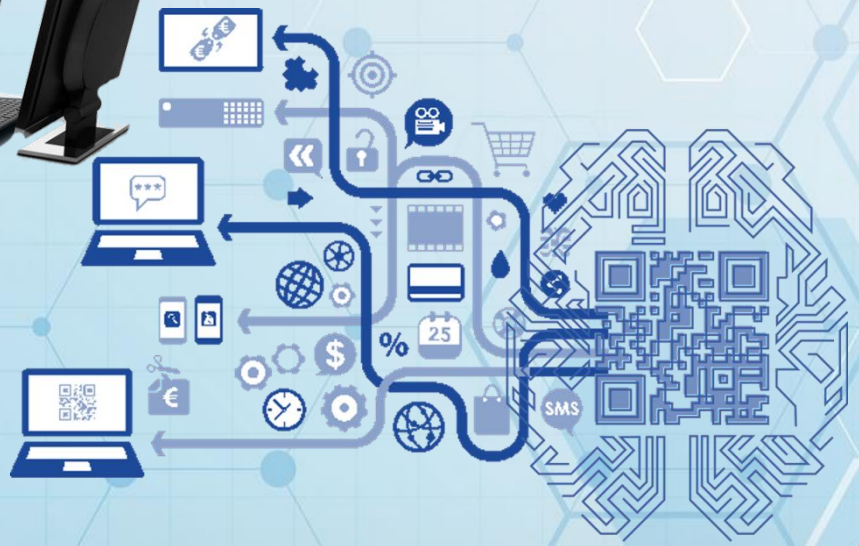
برنامه چگونه به پردازنده داده می شود ؟

کوچک ترین واحد اطلاعاتی برای یک پردازنده ، یک رقم دودویی با یک بیت است که حاوی یک سیگنال منطقی (صفر یا یک) می باشد . تمام اطلاعات کامپیوتری از مجموعه ای از بیت ها تشکیل می شوند . هر هشت بیت یک بایت نامیده می شود که یک واحد اطلاعاتی مهم در پردازنده ها ست . واحد مهم دیگر ، **Nibble** است که از چهار بیت تشکیل می‌شود .

اما پردازنده نه سمبل های موسیقی را می‌فهمد و نه با هیچکدام از زبان های محاوره‌ای ما آشنا است. تنها زبانی که می توان به وسیله ی آن با پردازنده ارتباط برقرار کرد ، زبان ارقام دودویی است . در واقع به دلیل سهولت استفاده از اجزای الکتریکی مانند لامپ خلاء ، رله و ترانزیستور که به صورت دو حالته نیز قابل استفاده هستند ، از ابتدا مدارات منطقی و پردازنده ها با دستگاه دودویی استاندارد شدند .

زبان ماشین

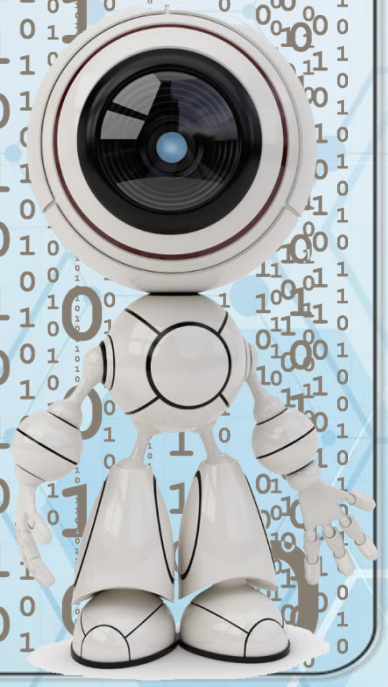
برای ارتباط با یک پردازنده باید از ارقام دودویی استفاده کنیم . تمام دستورات ، ورودی ها و خروجی ها در این قالب قرار می گیرند که آن را زبان ماشین یا زبان صفر و یک می نامیم . زبان ماشین ، پایین ترین سطح گفتگو با پردازنده است .



در ابتدا به دلیل عدم گستردگی کاربرد پردازنده ها ، از زبان صفر و یک برای برنامه نویسی پردازنده ها استفاده می‌شد . مثلا فرض کنید می‌خواهیم با استفاده از یک پردازنده ی فرضی ، دو عدد را بخوانیم و با هم جمع کنیم و نتیجه را نشان دهیم . با مطالعه ی برگه‌ی اطلاعات پردازنده‌ی مورد نظر متوجه می شویم که در این پردازنده رشته "10110001" به عنوان دستور خواندن عدد از ورودی در نظر گرفته شده است ، یعنی پردازنده با دریافت این رشته صفر و یک از حافظه (به عنوان دستور) ، یک عدد را از ورودی می‌خواند . به همین ترتیب دستور جمع دو عدد "00110011" و دستور ارسال نتیجه به خارج "00100010" است . بنابراین برنامه فوق به شکل زیر درمی‌آید

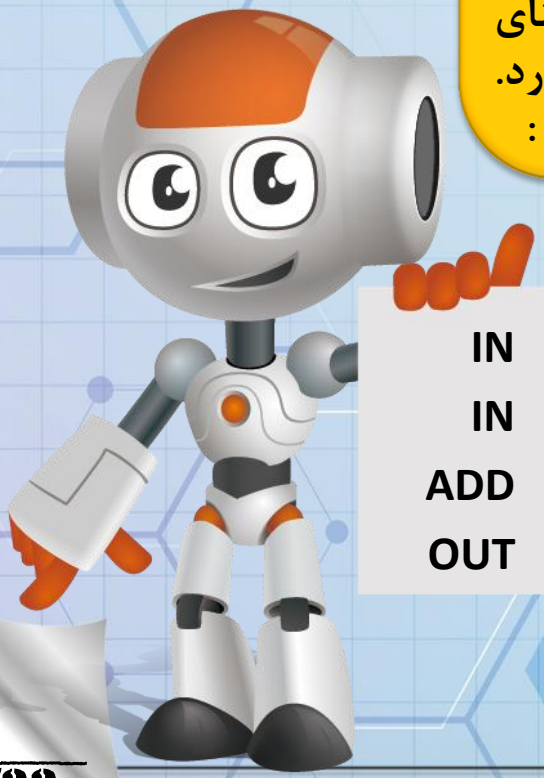
```
10110001
10110001
00110011
00100010
```

همانطور که می بینید برنامه نویسی و همچنین فهم این برنامه بسیار دشوار است. پس از گسترده شدن کاربرد پردازنده ها در سیستم های مختلف ، نیاز به زبانی که علی رغم نزدیکی کافی به سطح ماشین ، دارای دستورات ساده تر و قابل فهم تر باشد به شدت احساس می شد .



زبان اسمبلی

در زبان اسمبلی که به همین هدف عرضه شد ، برای هر دستور یک کلمه معادل که شبیه به کلمات انگلیسی است ، وجود دارد؛ مثلا در پردازنده فوق کلمه IN معادل رشته "10110001" در نظر گرفته شده است که با معنای این دستور که خواندن عدد از ورودی (INPUT) است نیز همخوانی دارد.
برنامه قبلی با استفاده از معادل اسمبلی دستورات به صورت زیر در می‌آید :



```
IN
IN
ADD
OUT
```

برنامه نویسی به زبان اسمبلی ،
کار را برای برنامه نویس راحت تر می کند ؛
اما پردازنده تنها زبان صفر و یک را متوجه می شود . نرم افزار اسمبلر کار ترجمه دستورات زبان اسمبلی به دستورات زبان ماشین را برعهده دارد .

زبان اسمبلی با ایجاد تحول در برنامه نویسی پردازنده ها سالها به عنوان زبان برنامه نویسی متداول پردازنده ها به کار می‌رفت . اما از دشواری نسبی برنامه نویسی به زبان اسمبلی که بگذریم ، برنامه نویس این زبان باید از ساختمان داخلی پردازنده ای که برای آن برنامه می نویسد ، کاملا آگاه باشد. بعلاوه برنامه ای که به زبان اسمبلی یک پردازنده نوشته می شود، روی پردازنده دیگری قابل اجرا نیست ، چون هر برنامه اسمبلی (یازبان ماشین) با توجه به ساختار داخلی یک پردازنده نوشته می شود که با پردازنده های دیگر متفاوت است. به همین دلیل زبان ماشین و زبان اسمبلی را زبانهای وابسته به ماشین می گویند

زبانهای سطح بالا

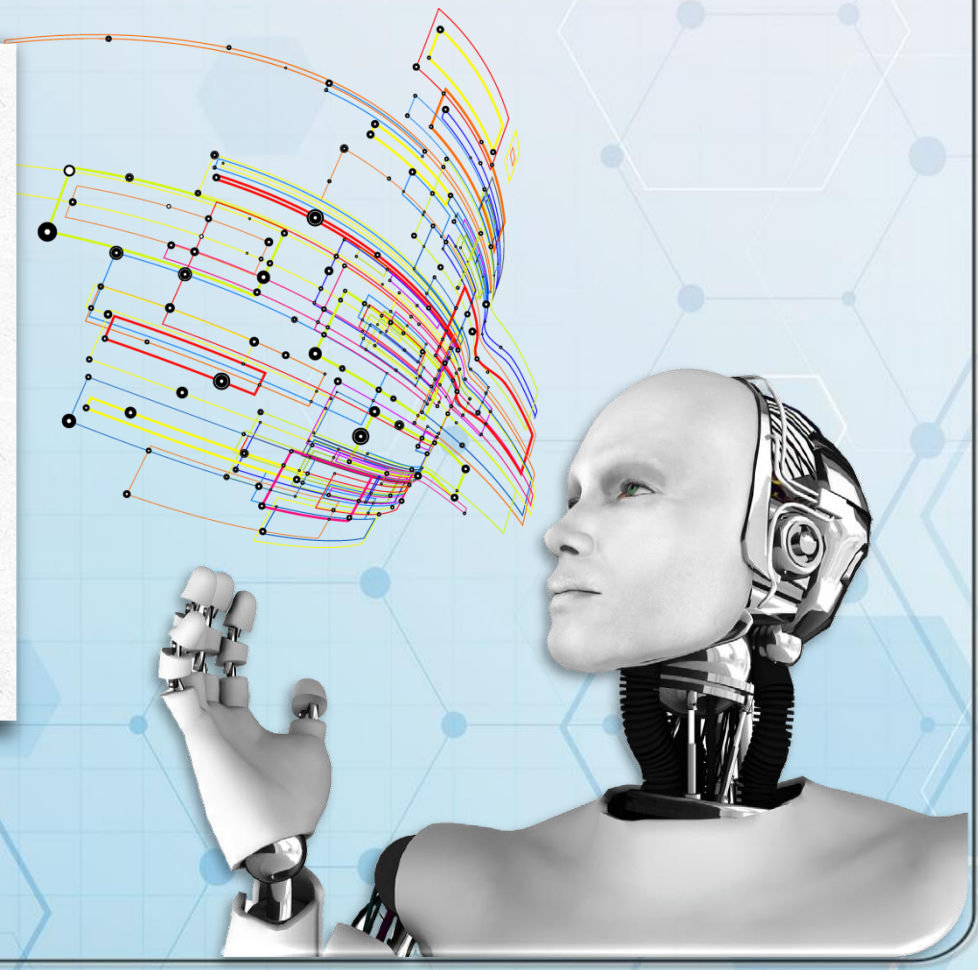
دستورات زبانهای برنامه نویسی سطح بالا مانند زبانهای پاسکال، C، فرترن و .. علاوه بر شباهت زیاد به زبان انگلیسی که باعث سادگی برنامه نویسی می شود، به کاربر اجازه می دهد بدون نیاز به اطلاع از ساختار داخلی پردازنده ، برنامه های خود را بنویسد، مثلا دستور پاک کردن صفحه نمایش کامپیوتر در زبانهای پاسکال و C، دستور clrscr (Clear Screen)است، اما برای انجام همین کار به زبان اسمبلی کامپیوتر این دستورات باید نوشته شوند:

```
MOV     AH,6H
MOV     AL,25
MOV     CX,0
MOV     DH,24
MOV     DL,79
MOV     BH,14H
INT     10H
```

این مثال به خوبی تفاوت برنامه نویسی به زبان سطح بالا و زبان اسمبلی را نشان می دهد .

چون پردازنده تنها دستورات زبان ماشین را درک می کند ، از نرم افزار مترجم (کامپایلر) برای تبدیل دستورات زبان سطح بالا به سطوح پایین تر استفاده می شود.

ذکر این نکته ضروریست که زبان اسمبلی هنوز در مواردی مانند وقتی که حجم و زمان اجرای برنامه ها مهم است ، مورد استفاده قرار می گیرد ، به طوری که حتی گاهی بخشی از یک برنامه سطح بالا را با استفاده از دستورات اسمبلی می نویسیم . این کار برای کاهش حجم کد ماشین نهایی و یا زمان اجرای آن و نیز استفاده موثر از سخت افزار انجام می شود .



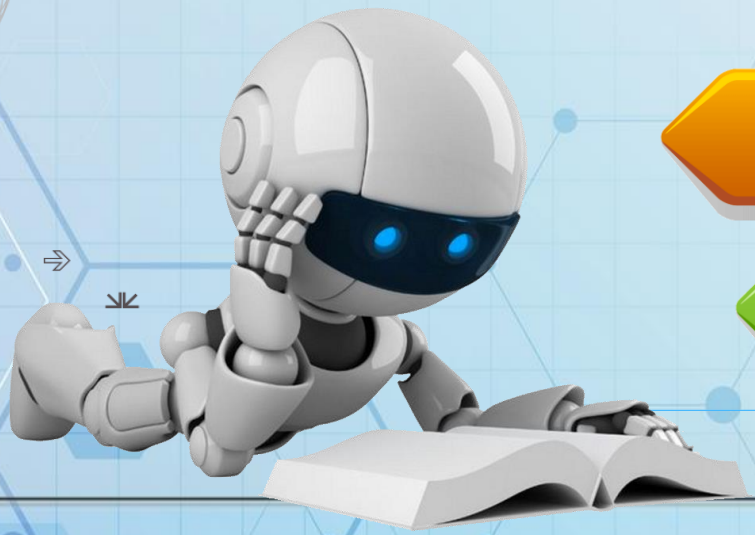
اجزاء داخلی پردازنده ها

پردازنده برای انجام سه وظیفه اصلی خود یعنی واکشی ، رمزگشایی و اجرای دستورات، به امکاناتی مجهز است که مهمترین آنها از این قرارند:

۱- ثبات ها

۲- واحد محاسبه و منطق

۳- واحد کنترل

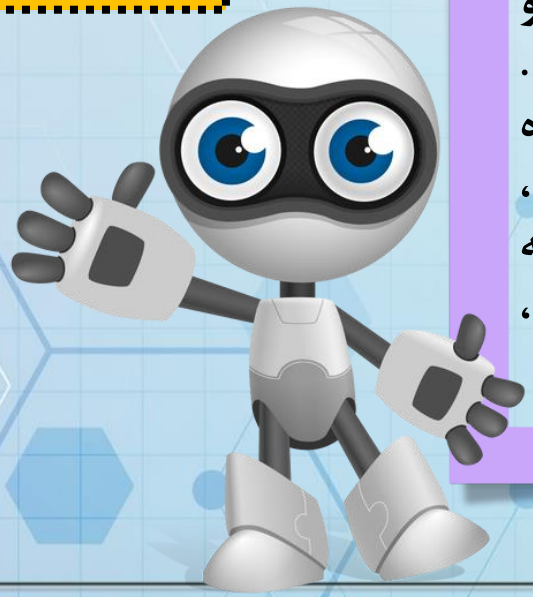


پردازنده ها دارای ثبات های متنوعی هستند که در زیر چند نوع از آن ها را معرفی می کنیم :

- ثبات انباره
- ثبات حالت(پرچم)
- ثبات شمارنده برنامه

۱- ثبات ها

ثباتها ، حافظه های کوچک و سریعی هستند که داخل پردازنده قرار دارند و برای ذخیره موقت داده ها و دستکاری آن ها به کار می روند . این ثبات ها بسته به نوع پردازنده می توانند ۸ بیتی ، ۱۶بیتی ، ۳۲بیتی ، ۶۴ بیتی و.. باشند. هر چه تعداد و اندازه ثبات ها بیشتر باشد ، کارایی پردازنده بالاتر است.



ثبات انباره:

یک ثبات همه منظوره است که در انواع دستورات حسابی و منطقی و انتقال داده ها به عنوان برگه کاری موقت به کار می رود .

ثبات حالت(پرچم):

بیت های این ثبات بیانگر حالت پردازنده بعد از اجرای دستورات است. مثلا یکی از بیت های این ثبات ، پرچم صفر است که اگر نتیجه یک عمل منطقی یا حسابی صفر باشد ، مقدار آن برابر یک می شود . یعنی مقدار ZF بعداز انجام یک عملیات منطقی یا حسابی ، بیانگر صفر بودن یا صفر نبودن نتیجه عملیات است.

از بیت های ثبات پرچم که در اکثر پردازنده ها وجود دارند می توان به پرچم صفر، پرچم سرریز ، پرچم علامت و پرچم رقم نقلی اشاره کرد. کاربرد اصلی ثبات پرچم در تصمیم گیری های لازم بعد از انجام یک عمل داخلی پردازنده است.

ثبات شمارنده برنامه :

چون دستورات یک برنامه باید به ترتیب اجرا شوند ، پردازنده باید به طریقی بداند دستور بعدی که باید اجرا کند کدام است . کار ثبات شمارنده برنامه ، نگهداری آدرس دستور بعدی است که قرار است توسط پزدازنده اجرا شود . با اجرای هر دستور پردازنده به صورت خودکار یک واحد به این ثبات اضافه می کند تا به دستور بعدی اشاره کند . با کمک این ثبات و گذرگاه های داده و آدرس ، پردازنده دستورات را از حافظه دریافت می کند.

گفتیم که پردازنده ، دستوراتی را که از حافظه دریافت می کند ، اجرا می نماید.

۲- واحد محاسبه و منطق

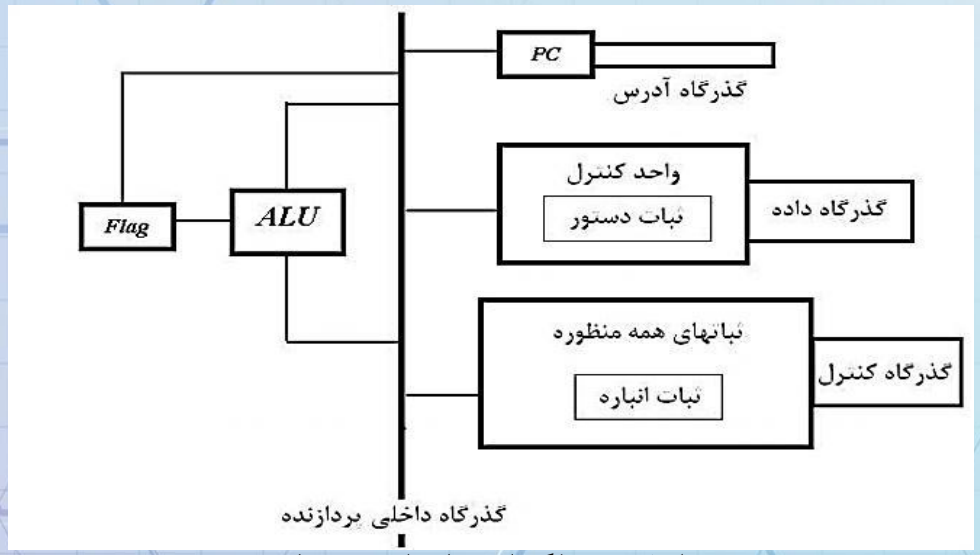
این واحد پردازنده که اصطلاحا به آن **ALU** گفته می شود، قسمتی از پردازنده است که مسئول انجام اعمال ریاضی مانند جمع ، تفریق، ضرب و تقسیم و اعمال منطقی مانند **AND** ، **OR** و **NOT** می باشد.

واحد **ALU** تنها یک حسابگر است و برای عملکرد درست باید کنترل شود.

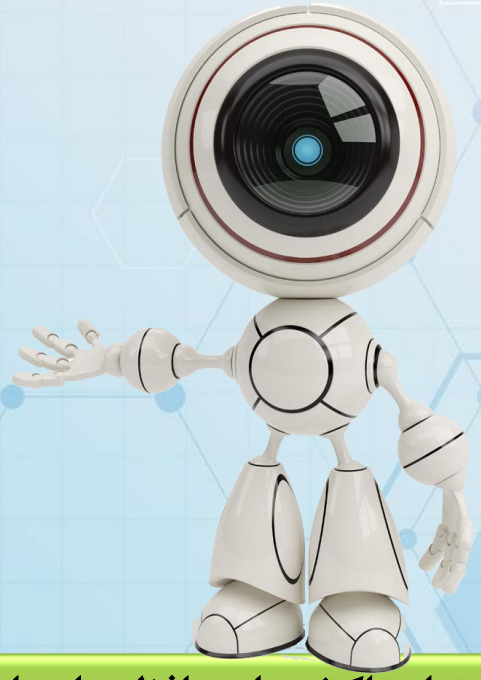
۳- واحد کنترل

کار این بخش ، کنترل تمام فعالیت های پردازنده است.

یکی از مهم ترین قسمت های واحد کنترل ، ثبات رمزگشای دستور است . می توان این قسمت را به عنوان یک فرهنگ لغت تصور کرد که معنای هر دستور و مراحلی را که پردازنده باید برای اجرای آن در پیش بگیرد ، مشخص می کند.



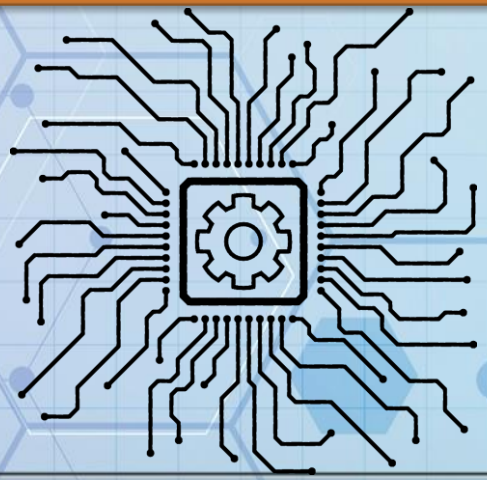
بلوک دیاگرام داخلی پردازنده

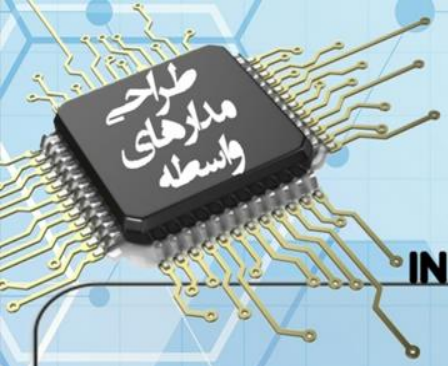


دستورات پس از واکشی از حافظه وارد این ثبات می شوند تا پس از رمزگشایی اجرا شوند.

سرعت پردازنده به چه معناست؟

در آخر به این نکته توجه کنید که پردازنده از نظر عملیاتی ، مانند یک مدار ترتیبی منطقی همزمان(سنکرون) عمل می کند و تمام اعمال آن با یک موج مربعی که پالس ساعت نام دارد ، هماهنگ می شود. فرکانس پالس ساعت ، یکی از معیارهای کارایی پردازنده و بیانگر سرعت آن خواهد بود . معیار مشابه دیگر به صورت تعداد دستورات قابل اجرا توسط پردازنده بر حسب میلیون دستور در ثانیه ، بیان می شود .





INTERFACE CIRCUITS DESIGN

ای بسا عالم ز دانش بی نصیب
حافظ علمست آنکس نه حبیب

مولانا

شاد و پیروز باشید