شاید تاکنون با واژه‌هایی مانند پردازنده‌های ۸ بیتی یا ۱۶ بیتی، ثبات یا رجیستر، ***ALU***، حافظه‌های ***RAM*** و ***ROM***، فضای آدرس‌دهی، گذرگاه یا باس داده و آدرس و... برخورد کرده باشید. این کلمات، واژه‌های معمول دنیای پردازنده‌ها هستند و هدف ما در این نوشتار، آشنا کردن شما با این مفاهیم می‌باشد.

۱–۱– پردازنده‌ها چگونه کار می‌کنند ؟

سیستمهای کنترلی

به طور کلی هر سیستم کنترلی دارای سه بخش ورودی، پردازشگر و خروجی است (شکل ۱). چنین سیستمی داده‌ها را از دنیای خارج **دریافت** می کند (بخش **ورودی**)، روی این داده‌ها **عملیاتی** انجام می دهد (بخش **پردازش**) و بر اساس این پردازشها **خروجی** لازم را تولید می کند (بخش **خروجی**). کارخانه پارچه بافی یک مثال ساده برای یک سیستم کنترلی است که نخ بعنوان ماده اولیه وارد آن می شود و پس از انجام یک سری عملیات، پارچه بعنوان خروجی سیستم تولید می شود.

شکل ۱–۱– اجزاء یک سیستم کنترلی

تصمیم‌گیریهای منطقی، بخش اساسی پردازش در سیستمهای کنترلی هستند. در درس مدارهای منطقی با نحوه طراحی بخش پردازشگر سیستمهای الکترونیکی با استفاده از مدارات منطقی ترکیبی و ترتیبی آشنا شده اید.

عملکرد پردازنده‌ها از نظر کلی با مدارهای منطقی یکسان است و تنها چگونگی ورود و خروج داده‌ها و نحوه انجام پردازش روی آنهاست که باعث متفاوت شدن ویژگیهای آنها می‌شود.

چرا پردازنده‌ها به وجود آمدند ؟

شاید این سؤال پیش بیاید که با وجود مدارهای ترکیبی و ترتیبی منطقی که انجام هر پردازشی را ممکن می‌سازند، چه نیازی موجب ایجاد و گسترش پردازنده‌ها شد ؟

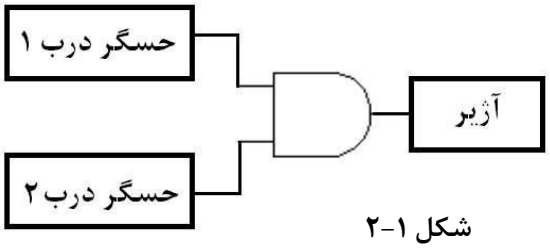
سادگی :

ساده‌ترین دلیل این است که حجم و پیچیدگی مدارهای منطقی با زیاد شدن تعداد ورودیها و خروجیهای سیستم و نیز پردازشهای آن به شدت افزایش می‌یابد که باعث دشواری طراحی، نگهداری، پیاده‌سازی، اشکالزدایی و... سیستم می شود و استفاده از مدارهای منطقی به عنوان بخش پردازشگر سیستمهای پیچیده و بزرگ را عملاً غیرممکن می‌سازد.

برنامه پذیری :

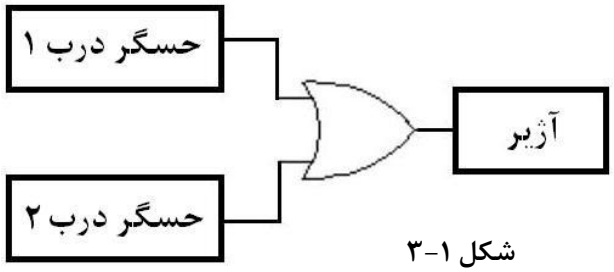
اما دلیل مهمتر، دشوار بودن پیکربندی مجدد و به عبارت دیگر تغییر ساختار مدارهای منطقی سیم‌بندی شده است. برای فهم این موضوع به مثال زیر توجه کنید :

فرض کنید یک سیستم امنیتی باید حالت دو درب را کنترل کرده و در صورت ایجاد خطر، آژیری را به صدا درآورد. یک حسگر[1]، حالت درب را به صورت یک سیگنال منطقی (صفر و یک) گزارش می‌کند؛ به این صورت که اگر درب باز باشد مقدار سیگنال یک و درغیراین‌صورت صفر است. خروجی سیستم که باید آژیر را فعال کند نیز یک سیگنال منطقی است که وقتی یک شود آژیر فعال می‌شود.



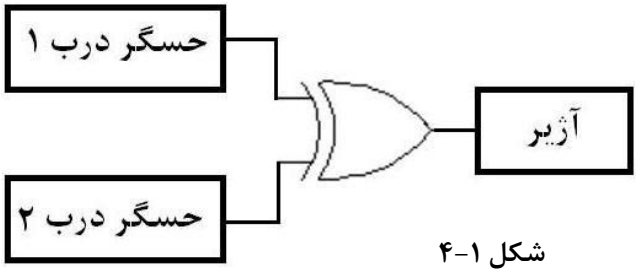
شکل ۱-۲

می‌خواهیم سیستم را به‌گونه‌ای طراحی کنیم که فقط اگر هر دو درب باز بودند، آژیر به صدا درآید. مدار ساده شده این سیستم در شکل ۱-۲ نشان داده شده است.



شکل ۱-۳

حال فرض کنید که از نظر امنیتی مدار باید به گونه‌ای تغییر یابد که اگر یکی از دربها هم باز شد، آژیر به صدا درآید. شکل ۱-۳ این سیستم را نشان می‌دهد.



شکل ۱-۴

اکنون سیاست امنیتی به گونه‌ای تغییر می‌یابد که فقط هنگامی که یکی از دربها باز بود، آژیر باید به صدا درآید. شکل ۱-۴ را ببینید.

[1] ***Sensor***

همانطور که مشاهده می‌کنید هر بار که نظر کارفرما تغییر می‌کند، سیستم نیز باید عوض شود[۱]؛ البته در این مثال ما تنها دو ورودی و یک خروجی داریم؛ در صورتی‌که همانند سیستمهای بزرگ و صنعتی دهها ورودی و خروجی داشته باشیم، تغییر کلی سیستم، هزینه‌های سرسام‌آوری درپی دارد.

ایده اصلی ایجاد و گسترش پردازنده‌ها، **قابلیت برنامه‌پذیری** آنهاست. یک سیستم کنترلی مبتنی بر پردازنده دارای تعدادی ورودی و خروجی است که تمام **پردازشهایی** که قرار است روی ورودیها انجام شود به صورت مجموعه ای از دستورات نرم افزاری که **برنامه**[۲] نام دارد، به سیستم داده می‌شود. تغییر نظر کارفرما، تنها منجر به **تغییر برنامه** سیستم می‌شود که هزینه آن در مقابل هزینه تغییر کلی سیستم بسیار ناچیز است. به بیان دیگر، پیچیدگی طراحی یک سیستم سخت‌افزاری، به طراحی یک برنامه مبدل می‌وشد که بسیار ساده‌تر است. این موضوع اساس کار کنترل‌کننده‌های منطقی برنامه‌پذیر[۳] که سیستمهایی مبتنی بر پردازنده (شکل ۱-۵) هستند، می‌باشد.

در این سیستمها، به جای طراحی بخش پردازشگر سیستم کنترلی به صورت سخت‌افزار سفارشی که مشکلات ذکر شده را به دنبال دارد، پردازش را به یک برنامه (نرم‌افزار) می سپاریم که هم طراحی و پیاده‌سازی و اشکالزدایی و هم تغییر کارکرد آن نسبت به سخت افزار، بسیار ساده‌تر و کم هزینه‌تر می باشد. وظیفه پردازنده **اجرای این برنامه** است.

شکل ۱-۵ – یک سیستم کنترلی مبتنی بر پردازنده

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

---

[۱] به این سیستمها اصطلاحاً **طراحی شده به صورت سفارشی(Custom Designed)** گفته می شود.

[2] ***Program***

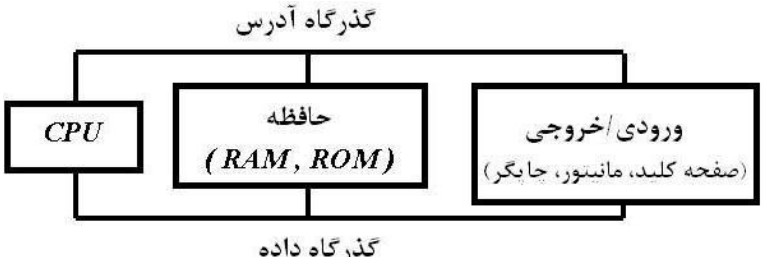
[3] ***Programmable Logic Controller : PLC***

یک سیستم پردازنده ای چه اجزایی دارد ؟

شما با کامپیوتر به عنوان یک سیستم پردازنده ای آشنایی دارید. قطعه CPU داخل کامپیوتر همان پردازنده مورد نظر ماست؛ اما کامپیوتر اجزاء دیگری مانند دیسک سخت، RAM، کارت گرافیکی، صفحه کلید، نمایشگر[1] و ... دارد. تمام این اجزاء را می‌توان در سه بخش اصلی (که اجزاء اساسی هر سیستم مبتنی بر پردازنده نیز هستند)، طبقه‌بندی کرد (شکل ۱-۶):

۱- *واحد پردازش مرکزی*[2] *(CPU)*
۲- *حافظه*[3]
۳- *وسایل ورودی و خروجی*[4] *(I/O)*

مجموعه دستورات یا برنامه ای که قرار است پردازشهای سیستم ما را انجام دهد، در واحد **حافظه** ذخیره می شود. کار **واحد پردازش مرکزی** (که از این به بعد آن را به اختصار **پردازنده** می نامیم) اجرای خط به خط این برنامه است. کار **واحدهای ورودی/خروجی**، تبادل اطلاعات با دنیای خارج است.



شکل ۱-۶ – اجزاء یک سیستم کنترلی مبتنی بر پردازنده

از نظر ساختاری می‌توان کامپیوتر را مانند یک انسان در نظرگرفت که داده‌هایی را از حواس خود (**ورودی**) مانند بویایی و بینایی می‌گیرد، در مغز خود (**پردازنده**) با توجه به اطلاعاتی که در ذهن (**حافظه**) دارد، پردازشهایی (شاید ناخودآگاه) را روی آنها انجام می‌دهد و متناسب با آنها عکس‌العمل (**خروجی**) نشان می‌دهد.

---
[1] ***Monitor***
[2] ***Central Processing Unit : CPU***
[3] ***Memory***
[4] ***Input / Output Devices : I/O***

مانند اجزای بدن که از طریق رگها و اعصاب با هم مرتبط هستند، اجزاء داخلی یک کامپیوتر از طریق مجموعه‌ای از سیمها به نام گذرگاه یا **باس**[۱] با یکدیگر در ارتباط می‌باشند.

اکنون به شرح مختصری راجع به بخشهای ذکر شده می‌پردازیم.

کار پردازنده چیست ؟

کار پردازنده اجرای مجموعه دستورات یا برنامه‌ای 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 مرحله فوق، کار یک نوازنده موسیقی را در نظر بگیرید. او برای اجرای یک آهنگ باید سمبل نت موسیقی را از روی یک‌صفحه که نتهای یک آهنگ روی آن نوشته شده‌اند، بخواند (Fetch)، اینکه آن سمبل نشان‌دهنده کدام نت است را درک کند (Decode)، و سپس آن نت را بنوازد (Execute).

برنامه چگونه به پردازنده داده می شود ؟

اما پردازنده نه سمبلهای موسیقی را می‌فهمد و نه با هیچکدام از زبانهای محاوره‌ای ما آشناست! تنها زبانی که می‌توان به وسیله آن با پردازنده ارتباط برقرار کرد، زبان ارقام دودویی است. در واقع به دلیل سهولت استفاده از اجزاء الکترونیکی مانند لامپ خلأ، رله و ترانزیستور که به صورت دوحالته نیز قابل استفاده هستند، از ابتدا مدارات منطقی و پردازنده‌ها با دستگاه دودویی استاندارد شدند.

کوچکترین واحد اطلاعاتی برای یک پردازنده، یک رقم دودویی یا یک **بیت**[۲] است که حاوی یک سیگنال منطقی (صفر یا یک) می باشد[۳]. تمام اطلاعات کامپیوتری از مجموعه‌ای از بیتها تشکیل می‌شوند. هر ۸ بیت یک **بایت**[۴] نامیده می‌شود که یک واحد اطلاعاتی مهم در پردازنده‌هاست. واحد مهم دیگر، ***Nibble*** است که از ۴ بیت تشکیل می‌شود.

---

[1] ***Bus***

[2] ***Binary digIT : BIT***

[۳] در تراشه‌های با استاندارد TTL، ولتاژ حدود صفر تا ۰/۸ ولت برای بیان **صفر منطقی** و ولتاژ حدود ۳ تا ۵ ولت برای بیان **یک منطقی** به کار می رود. صفر و یک منطقی می تواند با دو مقدار متفاوت جریان (مثلاً ۴ و ۲۰ میلی آمپر) نیز نمایش داده شود.

[4] ***Byte***

زبان ماشین[1]

برای ارتباط با یک پردازنده باید از ارقام دودویی استفاده کنیم. تمام دستورات، ورودیها و خروجیها در این قالب قرار می‌گیرند که آن را **زبان ماشین** یا **زبان صفر و یک** می‌نامیم. زبان ماشین، پایینترین سطح گفتگو با پردازنده است.

در ابتدا به دلیل عدم گستردگی کاربرد پردازنده‌ها، از زبان صفر و یک برای برنامه‌نویسی پردازنده‌ها استفاده می‌شد. مثلاً فرض کنید می‌خواهیم با استفاده از یک پردازنده فرضی، دو عدد را بخوانیم و با هم جمع کنیم و نتیجه را نشان دهیم. با مطالعه برگه اطلاعات پردازنده مورد نظر، متوجه می‌شویم 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 می خواند. به همین ترتیب دستور جمع دو عدد "00110011" و دستور ارسال نتیجه به خارج "00100010" است. بنابراین برنامه فوق به شکل زیر در می آید :

```
10110001
10110001
00110011
00100010
```

همانطور که می بینید برنامه نویسی و همچنین فهم این برنامه بسیار دشوار است.

پس از گسترده شدن کاربرد پردازنده‌ها در سیستمهای مختلف، نیاز به زبانی که علیرغم نزدیکی کافی به سطح ماشین، دارای دستورات ساده تر و قابل فهم تر باشد به شدت احساس می شد.

زبان اسمبلی[2]

در زبان اسمبلی که به همین هدف عرضه شد، برای هر دستور یک کلمه معادل که شبیه به کلمات انگلیسی است، وجود دارد؛ مثلاً 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 می آید:

```
IN
IN
ADD
OUT
```

[1] ***Machine Language***
[2] ***Assembly Language***

برنامه نویسی به زبان اسمبلی، کار را برای برنامه نویس راحت‌تر می کند؛ اما پردازنده تنها زبان صفر و یک را متوجه می شود. نرم افزار اسمبلر[۱] کار ترجمه دستورات زبان اسمبلی به دستورات زبان ماشین را بر عهده دارد.

زبان اسمبلی با ایجاد تحول در برنامه نویسی پردازنده‌ها، سالها بعنوان زبان برنامه‌نویسی متداول پردازنده‌ها به کار می رفت. اما از دشواری نسبی برنامه نویسی به زبان اسمبلی که بگذریم، برنامه‌نویس این زبان باید از ساختمان داخلی پردازنده‌ای که برای آن برنامه می نویسد، کاملاً آگاه باشد. بعلاوه برنامه ای که به زبان اسمبلی یک پردازنده نوشته می شود، روی پردازنده دیگری قابل اجرا نیست؛ چون هر برنامه اسمبلی (یا زبان ماشین) با توجه به ساختار داخلی یک پردازنده نوشته می شود که با پردازنده‌های دیگر متفاوت است. به همین دلیل زبان ماشین و زبان اسمبلی را **زبانهای وابسته به ماشین**[۲] می گویند.

زبانهای سطح بالا[۳]

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، برنامه‌های خود را بنویسد؛ مثلاً 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 می‌دهد.

چون پردازنده تنها دستورات زبان ماشین را درک می کند، از نرم افزار مترجم (**کامپایلر**)[۴] برای تبدیل دستورات زبان سطح بالا به سطوح پایین‌تر استفاده می شود.

---

[1] ***Assembler***
[2] ***Machine Dependent***
[3] ***High Level Languages : HLL***
[4] ***Compiler***

ذکر این نکته ضروریست که زبان اسمبلی هنوز در مواردی مانند وقتی که حجم و زمان اجرای برنامه‌ها 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 مؤثر از سخت‌افزار انجام می‌شود.

اجزاء داخلی پردازنده‌ها

پردازنده برای انجام سه وظیفه اصلی خود یعنی واکشی، رمزگشایی و اجرای دستورات، به امکاناتی مجهز است که در زیر به مهمترین آنها می‌پردازیم :

۱ – ثبات‌ها[1] : ثبات‌ها، حافظه‌های کوچک و سریعی هستند که داخل پردازنده قرار دارند و برای ذخیره موقت داده‌ها و دستکاری آنها به کار می‌روند. این ثباتها بسته به نوع پردازنده می‌توانند ۸ بیتی، ۱۶بیتی، ۳۲ بیتی، ۶۴ بیتی و ... باشند. هرچه تعداد و اندازه ثباتها بیشتر باشد، کارایی پردازنده بالاتر است.

پردازنده‌ها دارای ثباتهای متنوعی هستند که در زیر چند نوع از آنها را معرفی می‌کنیم:

- ***ثبات انباره[2] :*** یک ثبات همه‌منظوره[3] است که در انواع دستورات حسابی و منطقی و انتقال داده‌ها بعنوان برگه کاری موقت به‌کار می‌رود.
- ***ثبات حالت (پرچم)[4] :*** بیتهای این ثبات بیانگر حالت پردازنده بعد از اجرای دستورات است. مثلاً یکی از بیتهای این ثبات، پرچم صفر[5] است که اگر نتیجه یک عمل منطقی یا حسابی برابرصفر باشد، مقدار آن برابر یک می‌شود؛ یعنی مقدارZF بعد از انجام یک عملیات منطقی یا حسابی، بیانگر صفر بودن یا صفر نبودن نتیجه عملیات است.

  از بیتهای ثبات پرچم که در اکثر پردازنده‌ها وجود دارند می‌توان به پرچم صفر، پرچم سرریز[6]، پرچم علامت[7] و پرچم رقم نقلی[8] اشاره کرد. کاربرد اصلی ثبات پرچم در تصمیم تصمیم گیریهای لازم بعد از انجام یک عمل داخلی پردازنده است.

---

[1] ***Registers***
[2] ***Accumulator Register***
[3] ثباتهای همه منظوره (General Purpose Registers) ثباتهایی هستند که برای انجام هر عملیاتی داخل پردازنده می توانند مورد استفاده قرار گیرند.
[4] ***Flag Register***
[5] ***Zero Flag : ZF***
[6] ***Oveflow Flag : OF***
[7] ***Sign Flag : SF***
[8] ***Carry Flag : CF***

- ***ثبات شمارنده برنامه[1] :*** 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 پردازنده اجرا شود. با اجرای هر دستور، پردازنده به طور خودکار یک واحد به این ثبات اضافه می‌کند تا به دستور بعدی اشاره کند. با کمک این ثبات و گذرگاههای داده و آدرس، پردازنده دستورات را از حافظه دریافت می‌کند.

گفتیم که پردازنده، دستوراتی را که از حافظه دریافت می کند، اجرا می نماید. این دستورات چگونه اجرا می شوند؟

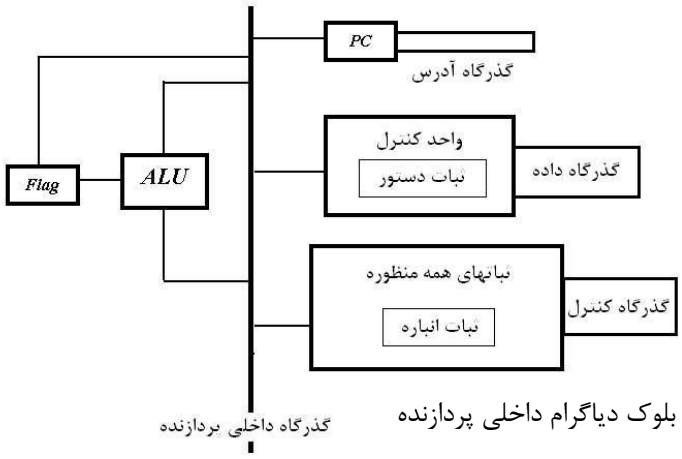
۲- *واحد محاسبه و منطق[2] :* این واحد پردازنده که اصطلاحاً به آن ALU گفته می‌شود، می‌شود، قسمتی از پردازنده است که مسؤول انجام اعمال ریاضی مانند جمع، تفریق، ضرب و تقسیم و اعمال منطقی مانند AND ، OR و NOT می‌باشد.

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

۳ - *واحد کنترل[3] :* کار این بخش، کنترل تمام فعالیتهای پردازنده است.

یکی از مهمترین قسمتهای واحد کنترل، ثبات رمزگشای دستور[4] است. می‌توان این قسمت قسمت را به‌عنوان یک فرهنگ لغت تصور کرد که معنای هر دستور و مراحلی را که پردازنده باید برای اجرای آن در پیش بگیرد، مشخص می‌کند. دستورات پس از واکشی از حافظه وارد این ثبات می‌شوند تا پس از رمزگشایی اجرا شوند.

بلوک دیاگرام داخلی پردازنده در شکل ۱-۷ آمده است.



شکل ۱-۷- بلوک دیاگرام داخلی پردازنده

[1] ***Program Counter : PC*** یا ***Instruction Pointer : IP***
[2] ***Arithmetic & Logic Unit : ALU***
[3] ***Control Unit : CU***
[4] ***Instruction (Decoder) Register : IR***

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 می‌کند و تمام اعمال آن با یک موج مربعی که پالس ساعت نام دارد، هماهنگ می‌شود. فرکانس پالس ساعت، یکی از معیارهای کارایی پردازنده و بیانگر سرعت آن خواهد بود. معیار مشابه دیگر به صورت MIPS[1] یعنی تعداد دستورات قابل اجرا توسط پردازنده بر حسب میلیون دستور در ثانیه، بیان می‌شود.

برای بحث دقیقتر در مورد چگونگی اجرای یک دستورالعمل در پردازنده، به مراجع معماری کامپیوتر مراجعه کنید.

نقش حافظه در سیستمهای کامپیوتری

اطلاعاتی که باید توسط پردازنده پردازش شود، اعم از دستورات و بعضی داده‌های ورودی/خروجی، باید در حافظه ذخیره شود. در سیستمهای پردازنده‌ای معمولاً از دو نوع حافظه استفاده می‌شود :

- ***حافظه فقط خواندنی (ROM)***[2]: اطلاعات این حافظه با قطع برق از میان نمی‌رود. بنابراین معمولاً (همانطور که از نام آن برمی‌آید) برای ذخیره اطلاعات دایمی و ثابت به کار می‌رود. مثلاً برنامه‌ای که حافظه کامپیوتر را تست می‌کند و حتماً اجرای آن را هنگام روشن کردن کامپیوتر خود دیده‌اید، باید پس از هر بار روشن شدن کامپیوتر اجرا شود؛ بنابراین باید در ROM کامپیوتر ذخیره شود[3]. معمولاً برای نوشتن در حافظه ROM به دستگاه مخصوصی به نام برنامه‌ریز[4] نیاز است.
- ***حافظه خواندنی و نوشتنی (RAM[5] یا RWM[6])***: اطلاعات این نوع حافظه با قطع برق از میان می‌رود. به علاوه برای نوشتن در آن نیازی به دستگاه برنامه‌ریز نیست. بنابراین برای ذخیره داده‌های موقت مانند متغیرهای یک برنامه یا در سطوح بالاتر برای نگهداری سیستم عامل یا نرم‌افزارهای مختلف مانند انواع بازیها یا برنامه‌های کاربردی دیگر به کار می‌رود.

  اطلاعات ROM بر خلاف RAM دایمی است و با قطع برق از بین نمی‌رود؛ به همین دلیل، گاهی ROM را حافظه غیرفرّار[7] هم می‌نامند.

---

[1] ***Milion Instruction Per Second : MIPS***
[2] ***Read Only Memory : ROM***
[3] به برنامه فوق که عملیات ابتدایی سیستم از جمله تست حافظه کامپیوتر را انجام می‌دهد، اصطلاحاً بایوس (BIOS) گفته می‌شود.
[4] ***Programmer***
[5] ***Random Access Memory : RAM***
[6] ***Read/Write Memory***
[7] ***Non Volatile Memory : NV-Memory***

در بخشهای بعدی با عملکرد و نحوه نوشتن و خواندن اطلاعات از حافظه بیشتر آشنا خواهیم شد.

نقش وسایل ورودی/خروجی در سیستمهای کامپیوتری

در حقیقت پردازنده‌ها برای رفع نیازهای کاربران طراحی شده‌اند؛ به این منظور باید داده‌هایی به پردازنده وارد شده و پس از انجام عملیات مورد نظر، نتایج از پردازنده خارج شوند. مجموعه‌ای شامل پردازنده، گذرگاهها و وسایل ورودی/خروجی، این وظیفه را در سیستمهای کامپیوتری به عهده دارند.

وسایل ورودی می‌توانند بسیار ساده مانند یک کلید دوحالته و یا پیچیده مانند صفحه کلید یا ماوس باشند. به طور مشابه، وسایل خروجی نیز انواع ساده مانند دیود نوری تا انواع پیچیده مانند نمایشگر را دربرمی‌گیرند. در ادامه این فصل با چگونگی استفاده از وسایل ورودی/خروجی ساده در سیستمهای پردازنده‌ای آشنا می‌شوید.

گذرگاهها در سیستمهای کامپیوتری

در هر کامپیوتر به طور معمول سه نوع گذرگاه وجود دارد : گذرگاه داده[1]، گذرگاه آدرس[2] و گذرگاه کنترلی[3].

تمام اطلاعاتی که باید در یک سیستم کامپیوتری جابجا شوند، از **گذرگاه داده** عبور می‌کنند. **گذرگاه آدرس** مشخص می‌کند که گذرگاه داده در هر لحظه باید در اختیار چه وسیله‌ای باشد و **گذرگاه کنترلی** این ارتباط را نظم می‌بخشد.

اهمیت گذرگاهها به حدی است که یکی از مهمترین معیارهای کارایی پردازنده‌ها، پهنای (تعداد خطوط) گذرگاه آدرس و گذرگاه داده آنها و تنوع سیگنالهای گذرگاه کنترلی آنهاست. کمی بیشتر با این گذرگاهها آشنا می‌شویم :

- ***گذرگاه داده :*** همانطور که گفته شد، گذرگاه داده محل عبور تمام اطلاعاتی است که باید داخل سیستم جابجا شوند؛ بعنوان مثال برای انتقال دستورات از حافظه به پردازنده، برای خواندن اطلاعات از ورودی و نیز برای ارسال اطلاعات به خروجی باید از گذرگاه داده استفاده شود.

[1] ***Data Bus***
[2] ***Address Bus***
[3] ***Control Bus***

پهنای گذرگاه داده (یعنی اینکه گذرگاه دادهٔ یک پردازنده چند بیتی است) تا آنجا مهم است که یکی از مهمترین معیارهای طبقه‌بندی پردازنده‌ها به شمار می‌رود که ابتدایی‌ترین نوع آن ۴ بیتی (مانند پردازنده‌های ۴۰۰۴ و ۴۰۴۰) بود و سپس انواع کاملتر ۸ بیتی (Z80، ۶۸۰۰، ۸۰۸۵)، ۱۶ بیتی (۸۰۸۶، ۸۰۸۸، ۸۰۲۸۶) و ۳۲ بیتی (۸۰۳۸۶، ۸۰۴۸۶، پنتیوم) به بازار آمد. وقتی گفته می‌شود پردازنده ۸۰۸۵، ۸ بیتی است یعنی می تواند هر بار (در هر سیکل کاری) ۸ بیت را پردازش کند؛ به بیان دیگر ورود و خروج اطلاعات در آن باید در قالبهای ۸ بیتی صورت گیرد.

واضح است که قدرت پردازش یک پردازنده به طور مستقیم با اندازه گذرگاه داده آن در ارتباط است؛ به عنوان مثال، یک پردازنده ۸ بیتی در هر سیکل کاری ۸ بیت را می‌تواند پردازش کند؛ اما یک پردازنده ۱۶ بیتی در همین مدت زمان می‌تواند ۱۶ بیت را مورد پردازش قرار دهد، به تعبیری سرعت پردازش ۲ برابر حالت قبل است.

گذرگاه داده یک گذرگاه دوجهته[1] است؛ یعنی داده‌ها از طریق آن هم می‌توانند به پردازنده وارد و هم از آن خارج شوند.

در بعضی پردازنده‌ها، پهنای گذرگاه داده خارجی (پینهای داده پردازنده) با پهنای گذرگاه داده داخلی (مسیر انتقال داده‌ها درون پردازنده) متفاوت است؛ مثلاً پردازنده ۸۰۸۸ دارای گذرگاه داده داخلی ۱۶ بیتی و گذرگاه داده داخلی ۸ بیتی است، در حالی که گذرگاههای داده داخلی و خارجی پردازنده هر دو ۱۶ بیتی‌اند.

- ***گذرگاه آدرس :*** همانطور که گفته شد، گذرگاه داده محل مشترک عبور اطلاعات سیستم است. پرسشی که بلافاصله به ذهن می آید این است که در هر لحظه چه کسی حق استفاده از این گذرگاه مشترک را دارد ؟

برای حل این مشکل، هر پردازنده دارای تعدادی پین آدرس است که با گذاشتن یک شماره روی آن، هویت وسیله ای را می تواند از گذرگاه داده استفاده کند را معلوم می کند. به مجموعه این پینها **گذرگاه آدرس پردازنده** گفته می‌شود.

نحوه استفاده از گذرگاه آدرس به این صورت است که طراح سیستم به همه **واحدهایی** که می خواهند از گذرگاه داده استفاده کنند (خانه‌های حافظه و تمام ورودیها و خروجیها)، یک شماره (**آدرس**) می دهد. در هنگام عملکرد سیستم، شماره یا آدرسی که پردازنده روی گذرگاه آدرسش قرار می دهد نشان دهنده

[1] ***Bi-directional***

شماره واحدی است که حق استفاده از گذرگاه داده را دارد. مثلاً اگر پردازنده‌ای ۸ خط آدرس (۸ پین آدرس) داشته باشد، با قرار گرفتن عدد 00110010 روی این پینها توسط پردازنده، به کل سیستم اعلام می شود که در این لحظه واحدی به شماره 00110010 (که ممکن است یک خانه حافظه، یک ورودی یا یک خروجی باشد) حق استفاده از گذرگاه داده (خواندن یا نوشتن) را دارد. این شماره در هنگام طراحی سیستم توسط طراح به این واحد داده شده است.

از آنجا که گذرگاه آدرس برای اعلان شماره واحدی که می تواند از گذرگاه داده استفاده کند به کار می‌رود، هر چه تعداد خطوط آن بیشتر باشد، پردازنده می‌تواند مقدار بیشتری حافظه یا تعداد بیشتری ورودی/خروجی را شناخته و از آنها استفاده کند. به‌عبارت دیگر، تعداد خطوط آدرس یک پردازنده، معیاری از تعداد واحدهایی است که می‌تواند آدرس‌دهی کند.

این تعداد واحدها همیشه توانی از ۲ است؛ به این ترتیب که یک پردازنده با n خط آدرس، می‌تواند به ۲^n واحد مختلف آدرس بدهد[۱]. مثلاً پردازنده Z80، ۱۶ خط آدرس دارد، این ۱۶ خط یا ۱۶ بیت می‌توانند $۲^{۱۶}$ یا ۶۵۵۳۶ حالت (آدرس دودویی) را بسازند، بنابراین Z80 می‌تواند تا ۶۵۵۳۶ واحد مختلف (حافظه یا ورودی/خروجی) (که هر کدام حاوی یک بایت می‌باشد) را آدرس‌دهی کند. اصطلاحاً گفته می‌شود که فضای آدرس‌دهی Z80 برابر ۶۵۵۳۶ بایت یا ۶۴ کیلوبایت است. توجه کنید که در اندازه‌گیری ظرفیت حافظه و فضای آدرس‌دهی، معنای کلمه کیلو، با سایر علوم متفاوت می باشد؛ به طوری‌که هر کیلوبایت شامل ۱۰۲۴ بایت است؛ به طور مشابه هر مگابایت برابر ۱۰۲۴ کیلو بایت، هرگیگابایت برابر ۱۰۲۴ مگابایت و هر ترابایت برابر ۱۰۲۴ گیگابایت است.

پردازنده 8086/88 دارای ۲۰ خط آدرس و فضای آدرس‌دهی یک مگابایت است؛ بنابراین آدرسهای حافظه در آن ۲۰ بیتی هستند. در آینده خواهیم دید چگونه با ثباتهای آدرس‌دهی ۱۶ بیتی پردازنده و سیستم سگمنت/آفست، این آدرس ۲۰ بیتی تولید می‌شود.

چون گذرگاه آدرس فقط برای آدرس‌دهی و توسط پردازنده به کار می‌رود، بنابراین یک گذرگاه یک‌جهته[۲] و سمت آن به سوی خارج پردازنده است.

- **گذرگاه کنترل** : این گذرگاه شامل پینهایی از پردازنده است که برای کنترل ارتباطات داخل سیستم مورد استفاده قرار می‌گیرد و در پردازنده‌های مختلف، نوع و

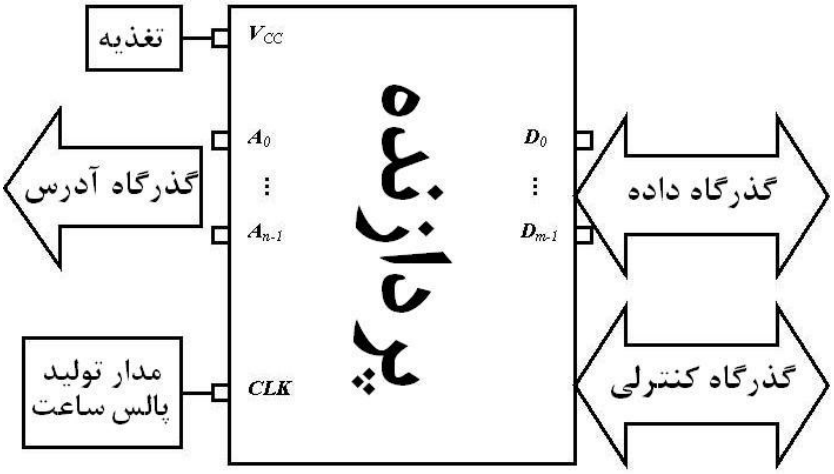
---

[۱] چون یک عدد دودویی با n رقم، می‌تواند ۲^n مقدار مختلف داشته باشد

[2] ***Uni-directional***

تعداد آنها متفاوت است. مثلاً تعدادی از این سیگنالها مشخص می‌کنند آیا پردازنده مشغول دریافت اطلاعات است یا ارسال اطلاعات ؟ دسته دیگر مشخص می‌کنند وسیله‌ای که آدرس آن توسط گذرگاه آدرس مشخص گردیده، از نوع حافظه است یا ورودی/خروجی ؟ و . . . .

شکل عمومی یک پردازنده با n خط آدرس و m خط داده را در شکل ۱-۸ مشاهده می‌کنید.



شکل ۱-۸ – شکل عمومی پینهای یک پردازنده

اکنون که با اجزاء یک سیستم پردازنده‌ای آشنا شده‌اید، وقت آن رسیده که با چند مثال ساده، چگونگی عملکرد پردازنده در اجرای برنامه‌ها را مورد بررسی قرار دهیم.