

اتوماسیون، سیستم های کنترلی، پردازنده ها

اتوماسیون یا فرآیند خودکار کردن فعالیتهایی که در زندگی روزمره شهری یا محیطهای صنعتی صورت می گیرد، یکی از مهمترین پایه ها و انگیزه های پیشرفت علم پردازنده ها و کامپیوترها به شمار می رود.

در این فصل پس از آشنایی با مفاهیم کلی اتوماسیون و سیستم های کنترلی که انگیزه اصلی ما از روی آوردن به مهندسی سخت افزار است، با پردازنده ها بعنوان یکی از مهمترین تحولات عرصه طراحی سیستمهای کنترلی آشنا خواهیم شد.

اتوماسیون^۱

پس از ساخته شدن ابزارهای محاسباتی که هر یک کاربردی در علوم زمان خود (به ویژه علم نجوم) داشتند، ساخت ماشین حسابی که بتواند محاسبات عددی با حجم زیاد را در زمان کوتاه انجام دهد، یکی از ایده های مهم بشر بوده است.

پس از ساخته شدن ماشین حسابهایی با تواناییهای گوناگون، فکر ساختن ماشینهای خودکار برای زمینه های مختلف به ویژه صنایع گوناگون مطرح شد. یکی از نخستین تلاشها در این راه، چرخ بافندگی خودکار ژوزف ژاکار فرانسوی بود که در سال ۱۸۰۸ ساخته شد و می توانست به کمک یک دسته کارت مشبک، حرکت ماشین را کنترل کرده و نقشهای متنوع روی پارچه ایجاد کند.

ماشینهای خودکار به سرعت جای خود را در صنعت باز کردند؛ به نحوی که می توان ایده خودکار کردن فرآیندهای صنعتی که اصطلاحاً *اتوماسیون* نامیده می شود را یکی از مهمترین انگیزه های پیشرفت تکنولوژی صنعتی دانست. در حال حاضر بخش اتوماسیون یکی از مهمترین بخشهای کارخانه ها و محیط های صنعتی به شمار می رود و بسیاری از خطوط تولید در این محیطها، به صورت خودکار و بدون دخالت انسان عمل می کنند.

اتوماسیون در زندگی عادی و روزمره ما نیز جایگاه ویژه ای یافته است :

^۱ Automation

- می توانید بدون ایستادن در صفهای طولانی بانک و جابجا کردن پول نقد، از طریق یک کارت الکترونیکی یا تلفن همراه پولهای خود را از جایی به جای دیگر انتقال دهید یا قبضه‌های خود را پردازید.
- می توانید بدون همراه بردن پول نقد، از فروشگاهها خرید کنید.
- می توانید از طریق تلفن همراه، چراغهای منزل خود را خاموش یا روشن کنید
- ...

سرعت پیشرفت اتوماسیون شهری به جهت ایجاد راحتی در زندگی انسان حیرت انگیز است؛ به نحوی که هم اکنون در بعضی کشورها حتی پرداخت کرایه تاکسی نیز از طریق دستگاهی که درون تاکسی نصب می شود و با اثر انگشت مسافر، پول را از حساب وی به حساب راننده انتقال می دهد انجام می پذیرد! روزی که چنین امکاناتی در زندگی ما حالت عادی پیدا کند چندان دور نیست ...

مزایای اتوماسیون

چرا اتوماسیون به این سرعت پیشرفت کرده است؟ در زیر چند دلیل مهم را ذکر می کنیم. می توانید با فکر کردن راجع به مثالهایی از اتوماسیون که با آنها برخورد می کنید، دلایل دیگری را نیز بیابید.

(۱) کم خطر بودن

آسیبهایی که در محیطهای صنعتی به سلامت انسانها وارد می شود، یکی از بالاترین آمارهای مرگ و میر و نقص عضو در جهان به شمار می رود. مسلماً یکی از اهداف ابتدایی اتوماسیون، پایین آوردن خطرهای تهدید کننده سلامت نیروی انسانی بوده است. آسیبها گاهی ناگهانی مانند نشت گاز، انفجار، جرقه های الکتریکی و ... هستند و گاهی مانند گازهای سمی محیطهای شیمیایی یا تشعشعات رادیواکتیو در بعضی محیطهای صنعتی، در درازمدت باعث آسیب به بافتهای بدن انسان می شوند. ایجاد سیستمهایی که بتواند در این محیطهای خطرناک جای انسان را بگیرد، یکی از وظایف مهم مهندسين سخت افزار به شمار می رود.

(۲) کاهش محدودیتهای انسانی

یک سیستم خودکار خسته نمی شود، می تواند با سرعت بالا کار کند، دقت آن معمولاً بسیار بالاست و با کاهش ضریب اشتباه می تواند اطمینان کاربران سیستم را به خوبی جلب کند. این مشخصات معمولاً در نیروی انسانی که در صنعت یا خدمات شهری مشغول به کارند کمیاب است.

۳) ایجاد راحتی در زندگی بشر

با توجه به آنچه در مورد اتوماسیون شهری گفته شد، مزایایی که این طیف اتوماسیون در زندگی بشری ایجاد می کند نیاز به شرح و توضیح ندارد. در حال حاضر یکی از مهمترین زمینه های اتوماسیون، نفوذ به زندگی روزمره مردم در جهت ایجاد تسهیلات لازم برای راحتی و صرفه جویی در وقت و هزینه و ... می باشد. پس از آشنایی با اتوماسیون، به سیستم های کنترلی که بافت اصلی اتوماسیون را تشکیل می دهند می پردازیم.

سیستم های کنترلی

یک تعریف ساده از یک سیستم کنترلی، سیستمی است که بتواند جایگزین یک انسان شود و تمام یا بخشی از وظایف او را انجام دهد.

- یک سیستم خودپرداز بانک جایگزین یک کارمند بانک می شود و وظایف او را انجام می دهد.
- یک سیستم امنیتی که هویت افراد را از روی اثر انگشت یا نقش پرده شبکه چشم او تشخیص داده و به او اجازه عبور از یک محل را می دهد، مانند یک نگهبان عمل می کند.
- سیستم ناوبری خودکار قایق می تواند وظایف ناخدا را انجام دهد.
- ...

سیستمهای ذکر شده تمام خودکار هستند. سیستم کنترلی می تواند نیمه خودکار نیز باشد؛ مانند بعضی ماشینهای برش ورقهای فلزی که زاویه و سرعت و عمق برش باید توسط کاربر تعیین شود و عمل برش به طور خودکار انجام می شود.

اجزاء سیستمهای کنترلی

یک سیستم کنترلی دارای سه بخش اصلی است :

- ۱) بخش ورودی که اطلاعات لازم را جمع آوری و وارد سیستم می کند.
- ۲) بخش پردازشگر که اطلاعات ورودی را بررسی کرده و براساس ساختار طراحی خود، روی آنها پردازش و تصمیم گیری انجام می دهد.
- ۳) بخش خروجی که نتیجه بخش پردازشگر را به صورت مناسب در جهان خارج ارایه می کند.



اجزاء یک سیستم کنترلی

مثلاً نگهبان درب یک ساختمان را در نظر بگیرید؛ وظیفه این نگهبان دیدن کارت شناسایی افراد مراجعه کننده (ورودی)، جستجوی شماره شناسایی آنها در یک دفتر مخصوص (پردازش)، و باز کردن درب برای مراجعه کنندگان در صورت تأیید هویت آنها (خروجی) است.

حال می خواهیم یک سیستم کنترلی را جایگزین این نگهبان کنیم (نگهبان ممکن است پیر شده باشد، دقت و سرعت کافی نداشته باشد، گاهی به بعضی آشنایان بدون داشتن کارت اجازه عبور دهد و ...!). برای این کار :

الف) به یک بخش ورودی نیاز داریم؛ این بخش می تواند یک دستگاه کارت خوان، یک دستگاه ثبت اثر انگشت یا ... باشد. به هر حال اطلاعات مربوط به هویت افراد به نحوی وارد سیستم می شود.

ب) بخش پردازشگر سیستم ما می تواند مدار منطقی، پردازنده، میکروکنترلر یا کامپیوتر باشد. این سیستم باید اطلاعات هویتی وارد شده از بخش ورودی را دریافت و با محتویات یک پایگاه اطلاعاتی که افراد مجاز به عبور را نشان می دهد مقایسه کند و در صورت تأیید هویت، دستور باز کردن درب را صادر کند.

ج) بخش خروجی سیستم معمولاً یک رابط الکتریکی - مکانیکی است که دستور باز کردن درب را از بخش پردازشگر گرفته و درب را به طور مکانیکی باز می کند.

فرآیند کلی طراحی سیستم های کنترلی کم و بیش به صورت ذکر شده انجام می شود.

بخش اصلی بحث ما راجع به بخش پردازشگر سیستم های کنترلی است. بخش ورودی و خروجی عموماً شامل دستگاههای الکترو - مکانیکی است که نحوه طراحی و استفاده از آنها خود بحثهای مفصلی را می طلبد که از حوصله بحث ما خارج است.

بنابراین از این پس روی بخش پردازشگر تمرکز می کنیم. برای راحتی فرض می کنیم بخش ورودی و خروجی با سیگنالهای منطقی که در فصل قبل به آنها پرداختیم آشنا هستند و در حقیقت با بخش پردازشگر به زبان صفر و یک منطقی صحبت می کنند.

طراحی بخش پردازشگر

(۱) طراحی با مدارات منطقی

در ابتدا فرض می‌کنیم بخش پردازشگر سیستم کنترلی که در واقع به منزله مغز تصمیم‌گیرنده سیستم عمل می‌کند، براساس مدارات منطقی طراحی شده است. این کار در گذشته در مقیاس وسیعی انجام می‌شد و هم‌اکنون نیز در بعضی موارد کاربردهای محدود دارد. اما به دلایلی که در ذیل بررسی می‌کنیم، اهمیت خود را از دست داده است:

الف) پیچیدگی

طراحی بخش پردازشگر سیستم با مدارات منطقی، برای سیستم‌هایی که تعداد ورودی و خروجی‌های آنها کم و پردازش آنها ساده است، مناسب می‌باشد؛ اما با بالا رفتن تعداد ورودیها و خروجیها و نیز با پیچیده شدن پردازشهای لازم سیستم، طراحی، پیاده‌سازی، اشکالزدایی و نگهداری سیستم بسیار پیچیده، هزینه‌بر و گاهی غیرممکن می‌شود. مثلاً شاید استفاده از یک مدار منطقی برای طراحی یک شمارنده ۸ بیتی معقول باشد، اما برای طراحی سیستم کنترل از راه دور یک ماشین اصلاً منطقی نیست!

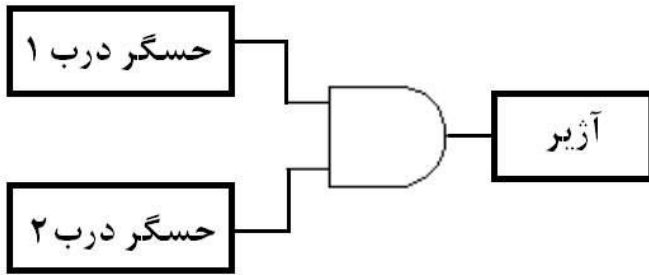
ب) انعطاف ناپذیری

اما مشکل عمده مدارات منطقی به عنوان بخش پردازشگر سیستم‌های کنترلی، مشکل انعطاف ناپذیری آنهاست.

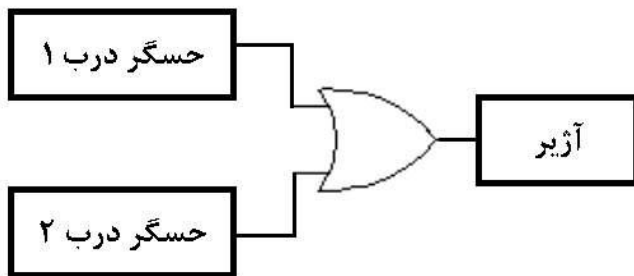
برای فهم این موضوع به یک مثال توجه کنید:

فرض کنید یک سیستم امنیتی باید حالت دو درب یک ساختمان را کنترل کرده و در صورت ایجاد خطر، آژیری را به صدا درآورد. یک حسگر^۱، باز و بسته بودن درب را به صورت یک سیگنال منطقی ("صفر" و "یک") گزارش می‌کند؛ به این صورت که اگر درب باز باشد مقدار سیگنال حسگر "یک" و در غیراین صورت "صفر" است. خروجی سیستم که باید آژیر را فعال کند نیز یک سیگنال منطقی است که وقتی "یک" شود آژیر فعال می‌شود.

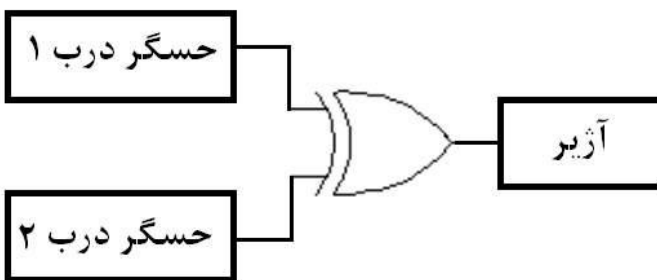
¹Sensor



می‌خواهیم سیستم را به گونه‌ای طراحی کنیم که فقط اگر هر دو درب باز بودند، آژیر به صدا درآید. مدار ساده شده این سیستم به شکل روبرو است. وقتی هر دو درب باز باشند، هر دو ورودی گیت AND "یک" است؛ پس خروجی آن نیز یک می‌شود و آژیر را به صدا درمی‌آورد. در غیر این صورت آژیر غیرفعال است.



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 روبرو این سیستم را نشان می‌دهد. اکنون سیاست امنیتی به گونه‌ای تغییر می‌یابد که فقط هنگامی که یکی از دربها باز بود آژیر باید به صدا درآید. شکل زیر را ببینید.



همانطور که مشاهده می‌کنید هر بار که نظر کارفرما تغییر می‌کند، ساختار سیستم نیز باید عوض شود؛^۱ البته در این مثال ما تنها دو ورودی و یک خروجی داریم؛ در صورتی که همانند سیستمهای بزرگ و صنعتی دهها ورودی و خروجی داشته باشیم، تغییر کلی سیستم، هزینه‌های سرسام‌آوری در پی دارد. راه حل چیست؟

۲) طراحی با پردازنده‌ها

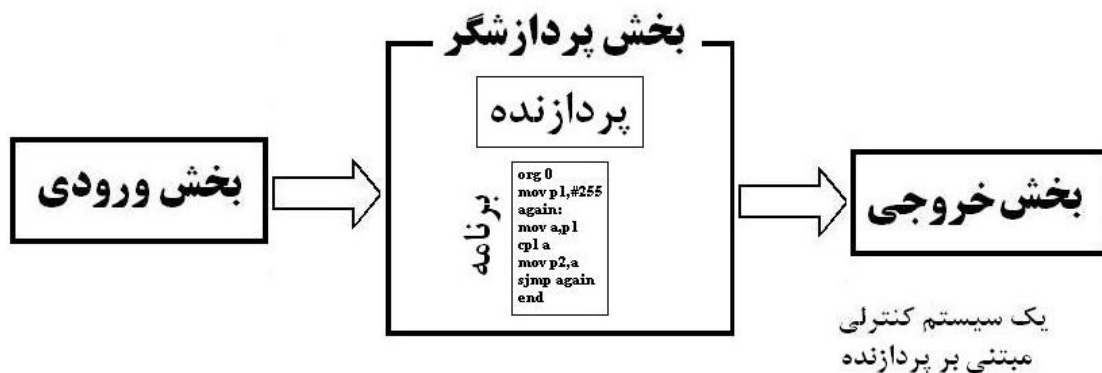
پردازنده‌ها، قطعات الکترونیکی هستند که به عنوان بخش پردازشگر در سیستمهای کنترلی به خصوص کامپیوترها به کار می‌روند. ایده اصلی ایجاد و گسترش پردازنده‌ها، قابلیت برنامه‌پذیری آنهاست.

یک سیستم کنترلی مبتنی بر پردازنده دارای تعدادی ورودی و خروجی است که تمام پردازشهایی که قرار است روی ورودیها انجام شود به صورت مجموعه‌ای از دستورات نرم افزاری که برنامه^۲ نام

^۱ به این سیستمها اصطلاحاً طراحی شده به صورت سفارشی (Custom Designed) گفته می‌شود.

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. این موضوع اساس کار **کنترل‌کننده‌های منطقی برنامه‌پذیر**^۱ که سیستم‌هایی مبتنی بر پردازنده هستند، را تشکیل می‌دهد.

در این سیستمها، به جای طراحی بخش پردازشگر سیستم کنترلی به صورت *سخت افزار سفارشی* که مشکلات ذکر شده را به دنبال دارد، پردازش را به یک **برنامه (نرم افزار)** می‌سپاریم که هم طراحی و هم تغییر کارکرد آن نسبت به سخت افزار، بسیار ساده تر و کم هزینه تر می‌باشد. وظیفه پردازنده اجرای این برنامه است.



پس استفاده از یک پردازنده به جای یک مدار منطقی به عنوان مغز یک سیستم کنترلی، علاوه بر ساده کردن کار طراحی، پیاده سازی، اشکالزدایی و نگهداری واحد پردازشگر سیستم، به طراح سیستم توانایی تغییر عملکرد سیستم با تغییر برنامه که بسیار ساده و کم هزینه است را می‌دهد.

اکنون که با ضرورت وجود پردازنده‌ها به عنوان پردازشگر سیستم‌های کنترلی آشنا شدیم، به بررسی چگونگی عملکرد سیستم‌های مبتنی بر پردازنده ها می‌پردازیم.

اجزای یک سیستم مبتنی بر پردازنده

دانستیم که یک سیستم مبتنی بر پردازنده باید برنامه کاربر را برای پردازش داده های ورودی و ایجاد داده های خروجی اجرا کند. برای این کار سیستم فوق باید مجهز به ابزار مختلفی باشد؛ تمام این اجزاء را می‌توان در سه بخش اصلی طبقه‌بندی کرد :

(۱) واحد پردازش مرکزی^۲ (CPU)

(۲) حافظه^۱

^۱ Programmable Logic Controller : PLC

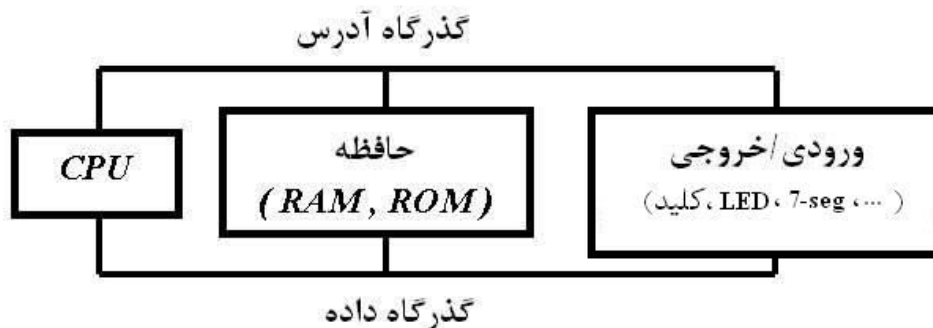
^۲ Central Processing Unit : CPU

۳) وسایل ورودی و خروجی^۲ (I/O)

مجموعه دستورات یا برنامه ای که قرار است پردازشهای سیستم را انجام دهد، در واحد حافظه ذخیره می شود.

کار واحد پردازش مرکزی (که آن را به اختصار پردازنده یا ریزپردازنده می نامیم) اجرای خط به خط این برنامه است.

پردازنده می تواند از طریق وسایل ورودی/خروجی، در مواقع لزوم داده هایی را با دنیای خارج مبادله کند (مثلاً دمای اتاق را از دماسنج الکترونیکی دریافت و پس از مقایسه آن با دمای مطلوب، دستور مناسب را برای تنظیم دما به بخاری یا کولر بدهد).



ساختار داخلی یک سیستم کنترلی مبتنی بر پردازنده

از نظر ساختاری می توان یک سیستم مبتنی بر پردازنده 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، پردازشهایی (شاید ناخودآگاه) را روی آنها انجام می دهد و متناسب با آنها عکس العمل (خروجی) نشان می دهد.

مانند اجزای بدن که از طریق رگها و اعصاب با هم مرتبط هستند، اجزاء داخلی یک کامپیوتر از طریق مجموعه ای از سیمها به نام گذرگاه یا باس^۳ با یکدیگر در ارتباط می باشند. اکنون به شرح مختصری راجع به بخشهای ذکر شده می پردازیم.

¹Memory

²Input / Output Devices : I/O

³Bus

پردازنده

کار پردازنده اجرای مجموعه 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 مرحله فوق، کار یک نوازنده موسیقی را در نظر بگیرید؛ او برای اجرای یک آهنگ باید سمبل نت موسیقی را از روی یک صفحه کاغذ 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).

پرسش اول این است که برنامه چگونه به پردازنده داده می‌شود؟

زبانهای برنامه‌سازی

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. همانطور که گفتیم، به دلیل سهولت استفاده از اجزاء الکتریکی دو حالته مانند لامپ خلاً، رله، ترانزیستور و ... از ابتدا مدارات منطقی و پردازنده‌ها با دستگاه دودویی استاندارد شدند.

کوچکترین واحد اطلاعاتی در دستگاه دودویی، یک رقم دودویی یا یک **بیت** است که حاوی یک سیگنال منطقی ("صفر" یا "یک") می‌باشد. تمام اطلاعات در یک سیستم مبتنی بر پردازنده از مجموعه‌ای از بیتها تشکیل می‌شوند. هر ۸ بیت یک **بایت** نامیده می‌شود که یک واحد اطلاعاتی مهم در پردازنده‌هاست. واحد مهم دیگر، *Nibble* است که از ۴ بیت تشکیل می‌شود.

زبان ماشین^۱

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

¹ Machine Language

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 استفاده از 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 (دستور) از حافظه، 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 در می‌آید:

```
10110001
10110001
00110011
00100010
```

همانطور که می‌بینید نوشتن 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 می‌شد.

زبان اسمبلی^۱

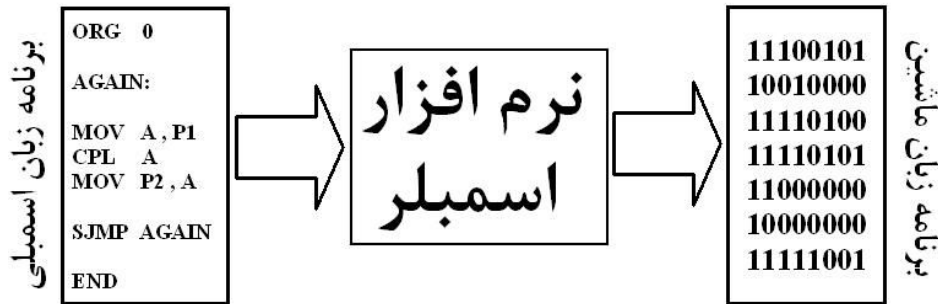
در زبان اسمبلی که به همین هدف عرضه شد، برای هر دستور یک کلمه معادل که شبیه به کلمات انگلیسی است، وجود دارد؛ مثلاً در پردازنده فوق کلمه IN معادل رشته "10110001" در نظر گرفته شده است که با معنای این دستور که خواندن عدد از ورودی (INput) است نیز همخوانی دارد. برنامه قبلی با استفاده از معادل اسمبلی دستورات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

```
IN
IN
ADD
OUT
```

برنامه‌نویسی به زبان اسمبلی، کار را برای برنامه‌نویس راحت‌تر می‌کند؛ اما فراموش نکنیم که پردازنده تنها زبان صفر و یک را متوجه می‌شود؛ نرم افزار اسمبلر^۲ کار ترجمه دستورات زبان اسمبلی به دستورات زبان ماشین را بر عهده دارد. در واقع اسمبلر به جای هر دستور اسمبلی، معادل زبان ماشین آن را جایگزین می‌کند.

^۱ Assembly Language

^۲ Assembler



زبان اسمبلی با ایجاد تحول در برنامه نویسی پردازنده ها، سالها بعنوان زبان برنامه نویسی متداول پردازنده ها به کار می رفت. اما از دشواری نسبی برنامه نویسی به زبان اسمبلی که بگذریم، برنامه نویس به این زبان باید از ساختمان داخلی پردازنده ای 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 پردازنده های دیگر متفاوت است. به همین دلیل زبان ماشین و زبان اسمبلی را **زبانهای وابسته به ماشین^۱** می گویند.

زبانهای سطح بالا^۲

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؛ 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 (که زبان اسمبلی پردازنده 8086 است)، 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)^۱ برای تبدیل دستورات زبان سطح بالا به سطوح پایینتر استفاده می شود.

¹ Machine Dependent

² High Level Languages : HLL

در این کتاب برای برنامه نویسی میکروکنترلر ۸۰۵۱ از زبان اسمبلی استفاده می کنیم؛ زبان اسمبلی با وجود دشواری نسبی برنامه نویسی باعث می شود شما به خوبی با ساختار داخلی و معماری میکروکنترلر ۸۰۵۱ آشنا شوید. در فصل آخر کتاب با زبان C-51 که زبان سطح بالا برای برنامه نویسی میکروکنترلر ۸۰۵۱ است، آشنا خواهید شد.

اجزاء داخلی پردازنده ها

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

۱ - **ثبات ها^۲**: ثبات ها، حافظه های کوچک و سریعی هستند که داخل پردازنده قرار دارند و برای ذخیره موقت داده ها و دستکاری آنها به کار می روند. این ثباتها بسته به نوع پردازنده می توانند ۸ بیتی، ۱۶ بیتی، ۳۲ بیتی، ۶۴ بیتی و ... باشند. هرچه تعداد و اندازه ثباتها بیشتر باشد، کارایی پردازنده بالاتر است.

پردازنده ها دارای ثباتهای متنوعی هستند که در زیر چند نوع از معمولترین آنها را معرفی می کنیم:

- **ثبات انباره^۳**: یک ثبات همه منظوره^۴ است که در انواع دستورات حسابی و منطقی و انتقال داده ها بعنوان برگه یادداشت موقت به کار می رود.
- **ثبات حالت (پرچم)^۵**: بیت های این ثبات بیانگر حالت پردازنده بعد از اجرای دستورات است. مثلاً یکی از بیت های این ثبات، پرچم صفر^۶ است که اگر نتیجه یک عمل منطقی یا حسابی برابر صفر باشد، مقدار آن برابر یک می شود؛ یعنی مقدار پرچم صفر بعد از 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^۷، پرچم علامت^۸ و پرچم رقم نقلی^۹ اشاره کرد.

¹ Compiler

² Registers

³ Accumulator Register

⁴ ثباتهای همه منظوره (General Function Registers) ثباتهایی هستند که برای انجام هر عملیاتی داخل پردازنده می توانند مورد استفاده قرار گیرند.

⁵ Flag Register

⁶ Zero Flag : ZF

⁷ Overflow Flag : OF

⁸ Sign Flag : SF

⁹ Carry Flag : CF

کاربرد اصلی ثبات پرچم در تصمیم گیریهای لازم بعد از انجام یک عمل داخلی پردازنده است؛ مثلاً بعد از انجام یک عملیات حسابی، اگر بخواهیم روی صفر بودن یا نبودن نتیجه عملیات تصمیم بگیریم، باید از پرچم صفر استفاده کنیم.

• **ثبات شمارنده برنامه^۱**: 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 می کند تا به دستور بعدی اشاره کند. با کمک این ثبات و گذرگاههای 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. این دستورات چگونه اجرا می شوند؟

۲- **واحد محاسبه و منطق^۲**: این واحد پردازنده که اصطلاحاً به آن ALU گفته می شود، قسمتی از پردازنده است که مسوول انجام اعمال ریاضی مانند جمع، تفریق، ضرب و تقسیم و اعمال منطقی مانند AND، OR و NOT می باشد.

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

۳- **واحد کنترل^۳**: کار این بخش، کنترل تمام فعالیتهای پردازنده است.

یکی از مهمترین قسمتهای واحد کنترل، ثبات رمزگشای دستور^۴ است. می توان این قسمت را به عنوان یک فرهنگ لغت تصور کرد که معنای هر دستور و مراحل را که پردازنده باید برای اجرای آن در پیش بگیرد، مشخص می کند. دستورات پس از واکنشی از حافظه وارد این ثبات می شوند تا پس از رمزگشایی اجرا شوند.

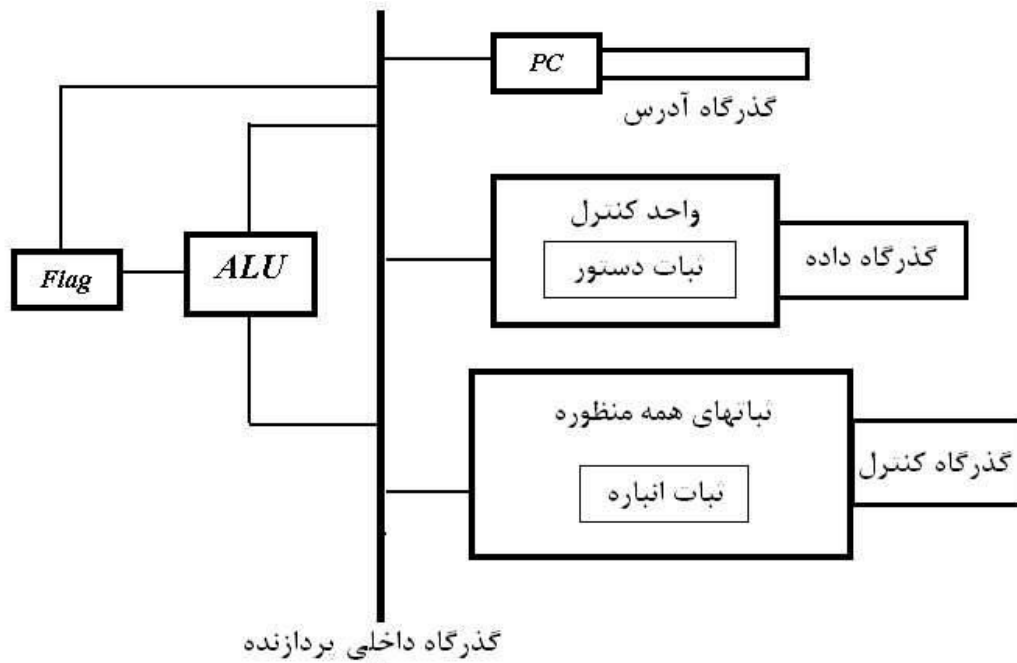
بلوک دیاگرام داخلی پردازنده در شکل زیر آمده است.

^۱Program Counter : PC

^۲Arithmetic & Logic Unit : ALU

^۳Control Unit : CU

^۴Instruction (Decoder) Register : IR



بلوک دیاگرام داخلی یک پردازنده

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 این پالس ساعت، یکی از معیارهای کارایی پردازنده و بیانگر سرعت آن خواهد بود. معیار مشابه دیگر به صورت MIPS^۱ 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 مراجعه کنید.

حافظه

دستوراتی که باید توسط پردازنده اجرا شود بعلاوه بعضی اطلاعات دیگر، باید در حافظه ذخیره شود. در سیستم‌های پردازنده‌ای معمولاً از دو نوع حافظه استفاده می‌شود :

- **حافظه خواندنی و نوشتنی (RAM)**^۳ که برای ذخیره داده‌های موقت مانند متغیرهای یک برنامه به کار می‌رود.

^۱Milion Instruction Per Second : MIPS

^۲ Mano Moris, Computer System Architecture, Prentice Hall, 1992.

^۳ Random Access Memory : RAM

- **حافظه فقط خواندنی (ROM)**^۱ که همانطور که از نام آن برمی آید، محل ذخیره اطلاعات دائمی^۲ و ثابت است؛ مثلاً برنامه سیستم که باید همیشه اجرا شود، طبیعتاً باید در حافظه ROM سیستم ذخیره شود. اطلاعات ROM بر خلاف RAM دائمی است و با قطع برق از بین نمی رود؛ به همین دلیل، گاهی ROM را حافظه غیرفرآر^۳ هم می نامند.

برای آگاهی از جزئیات انواع حافظه های RAM و ROM به ضمیمه ب.۱ مراجعه کنید.

وسایل ورودی/خروجی

در حقیقت پردازنده ها برای رفع نیازهای کاربران طراحی شده اند؛ به این منظور باید داده هایی که باید پردازش شوند به پردازنده وارد و پس از انجام عملیات مورد نظر، نتایج از پردازنده خارج شوند. مجموعه ای شامل پردازنده، گذرگاهها و وسایل ورودی/خروجی، این وظیفه را در سیستمهای مبتنی بر پردازنده به عهده دارند.

وسایل ورودی می توانند بسیار ساده مانند یک کلید دو حالتی و یا پیچیده مانند صفحه کلید یا قلم نوری در کامپیوتر باشند. به طور مشابه، وسایل خروجی نیز انواع ساده مانند دیود نوری تا انواع پیچیده مانند نمایشگر کامپیوتر را دربرمی گیرند.

گذرگاهها در سیستمهای کامپیوتری

در هر سیستم مبتنی بر پردازنده به طور معمول سه نوع گذرگاه وجود دارد: گذرگاه داده^۴، گذرگاه آدرس^۵ و گذرگاه کنترلی^۶. تمام اطلاعاتی که باید در یک سیستم کامپیوتری جابجا شوند، از گذرگاه داده عبور می کنند. گذرگاه آدرس مشخص می کند که گذرگاه داده در هر لحظه باید در اختیار چه وسیله ای باشد و گذرگاه کنترلی این ارتباط را نظم می بخشد.

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

¹ Read Only Memory : ROM

² Permanent

³ Non Volatile Memory : NV-Memory

⁴ Data Bus

⁵ Address Bus

⁶ Control Bus

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. پهنای گذرگاه داده (یعنی اینکه گذرگاه داده یک پردازنده، چند بیتی است یا به بیان دیگر چند پهنای تراشه پردازنده به تبادل داده ها اختصاص یافته است؟) تا آنجا مهم است که یکی از معیارهای نامگذاری پردازنده ها نیز به شمار می رود؛ مثلاً وقتی گفته می شود پردازنده ۸۰۸۵ یا میکروکنترلر ۸۰۵۱، ۸ بیتی هستند، یعنی می توانند هر بار (در هر سیکل کاری) ۸ بیت را پردازش کنند؛ به بیان دیگر ورود و خروج اطلاعات در آنها باید در قالبهای ۸ بیتی صورت گیرد.

پرسش) بر اساس جهت عبور داده ها، گذرگاههای سیستم به دو دسته یک جهته و دوجته تقسیم می شوند. آیا گذرگاه داده یک جهته است یا دو جهته؟ چرا؟

خلاصه

در این فصل با سیستم های کنترلی به عنوان قلب اتوماسیون آشنا شدیم. پس از بررسی مشکلات طراحی سیستم های کنترلی با مدارات منطقی، به پردازنده ها روی آوردیم و با بررسی ساختار داخلی آنها، اجزاء سیستم های مبتنی بر پردازنده را شناختیم.

در فصل آینده با میکروکنترلرها به عنوان گزینه ای دیگر برای طراحی بخش پردازشگر سیستم های کنترلی آشنا خواهیم شد.

پرسشهای دوره ای

(۱) الف) اتوماسیون چیست؟

ب) مواردی از اتوماسیون شهری که در زندگی روزمره خود با آن سروکار دارید را نام ببرید.

(۲) در هر کدام از موارد زیر از کدام مزایای اتوماسیون استفاده شده است؟

الف) سیستم خودپرداز بانک

ب) سیستمی که به طور خودکار سطح ماده مذاب در یک دیگ را کنترل می کند.

ج) سیستمی که در صورت تلاش فردی ناشناس برای ورود به منزل، با پلیس تماس می گیرد.

- د) سیستم کنترل خودکار روشنایی منزل در ساعات مختلف شبانه روز
- ه) سیستم تشخیص خودکار نقص روی ورقهای فولادی و از رده خارج کردن ورق معیوب
- ۳) سیستم کنترلی را تعریف کنید و چند مثال از آن ارائه کنید.
- ۴) بلوکهای یک سیستم کنترلی را نام ببرید و وظیفه هریک را شرح دهید.
- ۵) یک کارمند بانک را در نظر بگیرید که با دریافت چک و کارت شناسایی فرد، هویت او را تشخیص می دهد و مبلغ چک را به او پرداخت می کند. می خواهیم یک سیستم کنترلی را جایگزین این کارمند بانک کنیم. اجزاء تشکیل دهنده این سیستم را توضیح دهید.
- (این موضوع یکی از ایده های **دولت الکترونیک^۱** است)
- ۶) کارگری را در نظر بگیرید که باید سطح آب را در یک تانکر کنترل کند. او سطح آب را مرتباً می بیند و اگر سطح آب از حد مجاز بالاتر آمده بود، شیر ورودی آب را می بندد و در غیراین صورت آن را باز می کند. اجزاء سیستم کنترلی خودکار که می تواند جایگزین این کارگر شود را شرح دهید.
- ۷) با وجودی که انجام هرگونه پردازشی با استفاده از مدارات ترتیبی و ترکیبی منطقی ممکن است، دلایل ایجاد و گسترش پردازنده ها را در چه می بینید؟
- ۸) مزیت برنامه پذیری سیستم های کنترلی مبتنی بر پردازنده را توضیح دهید.
- ۹) اجزای یک سیستم کنترلی مبتنی بر پردازنده را نام ببرید و وظیفه هرکدام را شرح دهید.
- ۱۰) تفاوت زبان ماشین و زبان اسمبلی و نقش نرم افزار اسمبلر را توضیح دهید.
- ۱۱) وظیفه ثبات شمارنده (PC) در پردازنده چیست؟
- ۱۲) تفاوت حافظه های RAM و ROM و کاربرد هرکدام را توضیح دهید.
- ۱۳) جمله "1051 یک میکروکنترلر 1 بیتی است" یعنی چه؟

¹ Electronic Government : E-Gov.