

مهندس | هر آنچه یک دانشجوی مهندس لازم دارد

دانلود رایگان : کتاب، جزوه، مقاله، پروژه، گزارشکار و ...

WWW.MOHANDES.ORG



فصل اول

مفاهیم منطقی

همان گونه که می دانید در سیستم های دیجیتال، اطلاعات ورودی به صورت گسسته (دیجیتال) وارد و به شکل دودویی (binary) نمایش داده می شوند. عملوندهای (Operand) مورد استفاده در محاسبات ممکن است در دستگاه اعداد دودویی و اجزای گسسته دیگر مثل ارقام دهدهی به کدهای دودویی بیان شوند. البته در کامپیوترها مبنای دیگری نظیر مبنای هشت و شانزده نیز مورد استفاده قرار می گیرند.

هدف از ارائه این فصل، معرفی مبنای عددی گوناگون و یادآوری مفاهیم و عملوندهای منطقی، همچنین ارائه توضیح در مورد برخی اصطلاحات استفاده شده در فصول آینده است که به عنوان یک چهارچوب به منظور مطالعه جزئیات بیشتر فصول بعدی استفاده می شوند.

به دلیل گستردگی دامنه استفاده از PLC در صنایع مختلف، این مجموعه جهت استفاده متخصصان و کارشناسان رشته های مختلف تدوین شده است. در این فصل سعی شده تا تمام مطالب و مفاهیم اولیه مورد نیاز در مورد سیستم های کنترل، عناصر و المان های کنترلی، مفاهیم خاص در مورد ریزپردازنده ها و ... توضیح داده شود. بنابراین جهت مطالعه این کتاب نیاز به مراجعه به مراجع دیگر وجود ندارد و مطالب مورد نیاز در ضمیمه آمده است.

۱-۱- مبنایهای عددی

۱-۱-۱- مبنای دو یا باینری (Binary)

مبنای عددی مورد استفاده در سیستم‌های دیجیتال، مبنای دو است. پیشوند bi به معنی "دو" بوده، این اصطلاح به طور کلی سیستم، مقیاس و یا شرطی را تعریف می‌کند که دو جزء یا حالت دارد. در ریاضیات این اصطلاح نشان‌دهنده سیستم عددی مبنای "دو" می‌باشد که در آن، مقادیر به شکل ترکیب‌هایی از دو رقم صفر و یک بیان می‌شوند.

با استفاده از این دو رقم می‌توان دو حالت (خاموش و روشن یا درست و نادرست) را نشان داد که آن‌ها به نوبه خود با سطح ولتاژ در مدارهای الکترونیکی قابل ارائه است. در حقیقت سیستم عددی دودویی را می‌توان قلب محاسبات رقمی در سیستم‌های دیجیتال دانست.

یک بیت، طبق تعریف یک رقم دودویی است، و هنگامی که به همراه یک کد به کار می‌رود بهتر است که آن را یک کمیت دودویی برابر با "۰" یا "۱" تصور کنیم. نمایش یک گروه از 2^n عنصر به صورت کد به حداقل n بیت نیاز دارد. زیرا n بیت را می‌توان به 2^n طریق مجزا در کنار هم قرار داد. به عنوان مثال گروهی مشتمل بر چهار ($2^2 = 4$) مقدار مجزا را می‌توان با استفاده از یک کد دو بیتی نمایش و هر مقدار مجزا را به یکی از ترکیبات بیتی ۱۰، ۰۱، ۱۱ و ۰۰ نسبت داد. یک گروه با هشت ($2^3 = 8$) جزء، نیازمند یک کد سه بیتی است که هر جزء آن فقط و فقط به یکی از ترکیبات ۰۰۰، ۰۰۱، ۰۱۰، ۰۱۱، ۱۰۰، ۱۰۱، ۱۱۰ و ۱۱۱ نسبت داده می‌شود.

مثالهای فوق نشان می‌دهد که ترکیبات یک کد n بیتی را می‌توان با شمارش دودویی از صفر تا $2^n - 1$ به دست آورد. اعداد دودویی معمولاً به شکل ترکیبات چهار رقمی نوشته می‌شوند و برای اینکه با اعداد دهدهی اشتباه نشوند بعد از آنها حرف b و یا عدد ۲ به صورت زیرنویس ذکر می‌گردد. بنابراین عدد دهدهی ۲ به شکل دودویی 10_2 و یا $10_{(2)}$ نوشته می‌شود تا با عدد دهدهی ۱۰ اشتباه نگردد.

همان گونه که ذکر شد گاهی برای جلوگیری از بروز اشتباه، مبنای اعداد را در داخل پرانتزی جلوی عدد و به صورت زیرنویس اضافه می‌کنند.

$$10_{(2)} = 10_{(10)} = 2_{(10)} \text{ (عدد دهدهی ۲)}$$

در جدول ۱-۱ معادل اعداد ۰ تا ۹ دهدهی به صورت دودویی نمایش داده شده‌اند.

جدول ۱-۱: معادل دودویی ارقام دهدهی که با چهار بیت نمایش داده شده‌اند.

دهدهی	دودویی
۰	۰۰۰۰
۱	۰۰۰۱
۲	۰۰۱۰
۳	۰۰۱۱
۴	۰۱۰۰
۵	۰۱۰۱
۶	۰۱۱۰
۷	۰۱۱۱
۸	۱۰۰۰
۹	۱۰۰۱

اعداد دودویی نیز به شکل اعداد دهدهی به توان می‌رسند.

$$2^1 = 10_{(2)} \text{ (عدد دهدهی ۲)}$$

$$10^1 = 10_{(10)}$$

$$2^2 = 100_{(2)} \text{ (عدد دهدهی ۴)}$$

$$10^2 = 100_{(10)} \text{ : در مقایسه با}$$

$$2^3 = 1000_{(2)} \text{ (عدد دهدهی ۸)}$$

$$10^3 = 1000_{(10)}$$

در سیستم‌های دیجیتال گاهی لازم است تا مبناهای عددی به یکدیگر تبدیل شوند. این تبدیل

مبنا با مثال زیر روشن می‌گردد.

مثال ۱-۱: عدد $41_{(10)}$ را به مبنا ۲ تبدیل کنید.

روند اجرای عملیات تبدیل به صورت زیر می‌باشد:

ابتدا 41 بر 2 تقسیم شده تا خارج قسمت 20 و باقیمانده 1 به دست آید. خارج قسمت مجدداً بر 2 تقسیم شده تا خارج قسمت و باقیمانده جدیدی حاصل گردد. این روال به همین ترتیب تا زمانی ادامه می‌یابد که خارج قسمت صحیح به دست آمده صفر شود. ضرایب عدد دودویی مطلوب به صورت زیر از باقیمانده‌ها به دست می‌آیند.

خارج قسمت صحیح		باقیمانده	ضریب عدد دودویی
$\frac{41}{2} = 20$	+	۱	$a_0 = 1$
$\frac{20}{2} = 10$	+	۰	$a_1 = 0$
$\frac{10}{2} = 5$	+	۰	$a_2 = 0$
$\frac{5}{2} = 2$	+	۱	$a_3 = 1$
$\frac{2}{2} = 1$	+	۰	$a_4 = 0$
$\frac{1}{2} = 0$	+	۱	$a_5 = 1$

$$\text{جواب: } 41_{(10)} = (a_5 a_4 a_3 a_2 a_1 a_0)_2 = (101001)_2$$

روال ریاضی فوق می تواند به صورت مناسب تری به شکل زیر نمایش داده شود.

عدد صحیح	باقیمانده
41	
20	۱
10	۰
5	۰
2	۱
1	۰
0	۱
جواب = 101001	

تبدیل اعداد صحیح دهدهی به مبنای ۲ شبیه مثال مذکور است بجز اینکه تقسیم می بایست به جای ۲ بر ۲ صورت گیرد.

برای درک چگونگی تبدیل مبنای دودویی به دهدهی مثال ۱-۱ را به صورت عکس بیان می کنیم.
مثال ۱-۲: عدد $(101001)_2$ را به مبنای ۱۰ تبدیل کنید.

$$\begin{aligned} 101001_{(2)} &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ &= 32 + 0 + 8 + 0 + 0 + 1 = 41_{(10)} \end{aligned}$$

پس می توان گفت که هر عدد در مبنای ۲ به شکل $(a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0)_2$ به صورت

حاصلضرب توانهای r در ضرایب مربوطه اش بیان می شود.

$$a_n \times r^n + a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \dots + a_2 \times r^2 + a_1 \times r^1 + a_0$$

اعداد دودویی گرچه برای کامپیوترها استفاده می شوند اما چون رشته های تکراری از صفرها و یک ها هستند تفسیر آنها برای مردم عادی مشکل است. برنامه نویسان و کسانی که با قابلیت های پردازش داخلی کامپیوترها سر و کار دارند جهت سهولت در تفسیر این نوع اعداد از سیستم های عددی اکتال (مبنای ۸) و یا هگزا دسیمال (مبنای ۱۶) استفاده می کنند.

۱-۱-۲- مبنای هشت یا اکتال (Octal)

واژه Octal از کلمه لاتین "Octo" به معنی هشت گرفته شده و به مفهوم سیستم عددی مبنای هشت است که شامل ارقام ۰ تا ۷ می باشد. سیستم مبنای هشت برای ارائه اعداد دودویی به شکل فشرده تر در برنامه نویسی مورد استفاده قرار می گیرد. از آنجایی که این سیستم عددی از هشت رقم تشکیل یافته است و همچنین با استفاده از ۳ بیت می توان هشت ترکیب مختلف را نشان داد، اعداد دودویی برای تبدیل به مبنای ۸ معمولاً به گروه های ۳ بیتی تقسیم می شوند.

به عنوان مثال معادل های دودویی هشت رقم (ارقام ۰ تا ۷) در مبنای هشت در جدول ۱-۲ نشان داده شده است.

جدول ۱-۲: معادل اعداد دودویی در مبنای هشت

مبنای دو	مبنای هشت
۰۰۰	۰
۰۰۱	۱
۰۱۰	۲
۰۱۱	۳
۱۰۰	۴
۱۰۱	۵
۱۱۰	۶
۱۱۱	۷

همان گونه که ذکر شد برای تبدیل اعداد دودویی به مبنای هشت، عدد مورد نظر را به دسته های

۳ بیتی تقسیم می‌کنیم. برای مثال عدد دودویی $(1010011)_2$ را در نظر بگیرید. جهت تبدیل این عدد به مبنای هشت از سمت راست شروع کرده، دسته‌های ۳ بیتی جدا می‌کنیم و در مورد آخرین دسته در صورت کمبود بیت به تعداد تفاضل بیت‌های موجود در دسته و عدد ۳، صفر اضافه می‌نمائیم. با این عمل دسته‌های ۰۱۱، ۰۱۰ و ۰۰۱ به دست می‌آید که با تبدیل این دسته‌ها به مبنای هشت عدد $(123)_8$ حاصل می‌گردد.

$$(1010011)_2 = (001 \ 010 \ 011)_2 \\ = (1 \ 2 \ 3)_8$$

گرچه اعداد در مبنای هشت از نظر ظاهر شبیه اعداد دهدهی هستند اما مفهوم متفاوتی دارند. به عنوان مثال عدد $(123)_8$ معادل با عدد $1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0 = 123$ است در حالی که این عدد در سیستم مبنای هشت به مفهوم $1 \times 8^2 + 2 \times 8^1 + 3 \times 8^0$ می‌باشد که معادل عدد دهدهی ۸۳ است. از آنجایی که سیستم مبنای ۸ با مضرب‌هایی از ۳ بیت کار می‌کند و میکرو کامپیوترها معمولاً بر اساس واحدهای ۴، ۸، ۱۶ و ۳۲ ... هستند این سیستم عددی بیشتر در مینی کامپیوترها و کامپیوترهای بزرگ مورد استفاده قرار می‌گیرد. در مقابل، سیستم عددی مبنای ۱۶ کاربرد بیشتری در میکرو کامپیوترها دارد.

۱-۱-۳- مبنای شانزده یا هگزادسیمال (Hexadecimal)

واژه hexadecimal از ترکیب دو کلمه یونانی hex به معنی ۶ و کلمه لاتین decim به معنی ۱۰ گرفته شده است. سیستم عددی مبنای ۱۶ از ارقام صفر تا ۹ و حروف بزرگ A (معادل اعشاری یا دهدهی عدد ۱۰) تا F (معادل دهدهی عدد ۱۵) تشکیل می‌گردد. هگزا دسیمال که به اختصار "هگز" گفته می‌شود در برنامه‌نویسی برای نشان دادن اعداد دودویی مورد استفاده کامپیوتر در یک قالب فشرده‌تر به کار می‌رود.

اعداد هگزا دسیمال در گروه‌های ۸ بیتی که اساس حافظه و منبع ذخیره اطلاعات در کامپیوتر می‌باشد، جای می‌گیرند. از آنجایی که در چهار بیت می‌توان هر یک از ۱۶ رقم را نشان داد، یک عدد دو رقمی هگز در یک گروه هشت بیتی گنجانده می‌شود. به مثال زیر توجه کنید:

مثال ۱-۳: عدد دهدهی ۸۳ را به مبنای ۱۶ تبدیل کنید.

طبق روش گفته شده در مثال ۱-۱ ابتدا عدد ۸۳ را به ۱۶ تقسیم نموده، خارج قسمت و باقیمانده را به دست می‌آوریم.

خارج قسمت صحیح	باقیمانده	ضریب عدد هگز
$\frac{۸۳}{۱۶} = ۵$	+	$a_0 = ۳$
$\frac{۵}{۱۶} = ۰$	+	$a_1 = ۵$
$۸۳_{(۱۰)} = a_1 a_0_{(۱۶)} = ۵۳_{(۱۶)}$		

برای تبدیل عددی از مبنای ۲ به مبنای ۱۶ باید از سمت چپ بیت‌ها را به دسته‌های ۴ تایی تقسیم نمود و در دسته آخر به تعداد تفاضل بیت‌های موجود و عدد ۴، صفر قرار داد. مثلاً برای تبدیل عدد $(۱۱۱۰۰۱۰۱۰)_۲$ به مبنای ۱۶، دسته‌های ۴ بیتی ۱۰۱۰ ، ۱۱۰۰ و ۰۰۰۱ را خواهیم داشت و با قرار دادن معادل آنها عدد $(۱CA)_{۱۶}$ حاصل می‌گردد.

$$(۰۰۰۱ \ ۱۱۰۰ \ ۱۰۱۰)_۲$$

$$(\ ۱ \ C \ A)_{۱۶}$$

در جدول ۱-۳ معادل اعداد دودویی و هگز نشان داده شده است.

جدول ۱-۳: معادل دودویی و هگز اعداد در مبنای ۱۰

مبنای ۱۰	مبنای ۱۶	مبنای ۲
۰	۰	۰۰۰۰
۱	۱	۰۰۰۱
۲	۲	۰۰۱۰
۳	۳	۰۰۱۱
۴	۴	۰۱۰۰
۵	۵	۰۱۰۱
۶	۶	۰۱۱۰
۷	۷	۰۱۱۱
۸	۸	۱۰۰۰
۹	۹	۱۰۰۱
۱۰	A	۱۰۱۰
۱۱	B	۱۰۱۱
۱۲	C	۱۱۰۰
۱۳	D	۱۱۰۱
۱۴	E	۱۱۱۰
۱۵	F	۱۱۱۱

بنابر آنچه گفته شد هر گروه ۸ بیتی می تواند هر یک از $2^8 = 256$ عدد مختلف هگزا دسیمال (از 00_{16} تا FF_{16}) را در خود جای دهد.

۱-۴-۱- مبنای BCD (Binary Coded Decimal)

اعداد دهدهی کد شده در مبنای دو (دهدهی کد شده به دودویی) با حروف اختصاری BCD نشان داده شده و مبین سیستمی است که در آن، جهت جلوگیری از خطاها و اشتباهات مربوط به گرد کردن ها و تبدیل ها، اعداد دهدهی در مبنای دو کددهی می گردند. در کد BCD، هر رقم در مبنای دهدهی به طور جداگانه به شکل یک عدد دودویی کد می شود. هر یک از ارقام دهدهی صفر تا ۹ در چهار بیت کددهی شده، هر گروه چهار بیتی جهت سهولت در خواندن با یک فاصله از گروه دیگر جدا می گردد. در این روش که ۱-۲-۴-۸ نیز نامیده می شود مطابق جدول ۱-۴ کدهای زیر مورد استفاده قرار می گیرند.

جدول ۱-۴: نمایش اعداد در مبنای دهدهی و معادل BCD آنها

مبنای دهدهی	BCD
۰	۰۰۰۰
۱	۰۰۰۱
۲	۰۰۱۰
۳	۰۰۱۱
۴	۰۱۰۰
۵	۰۱۰۱
۶	۰۱۱۰
۷	۰۱۱۱
۸	۱۰۰۰
۹	۱۰۰۱

به عنوان مثال در کد BCD، عدد $12_{(10)}$ به شکل ۰۰۱۰ ۰۰۰۱ و عدد $96_{(10)}$ به شکل ۱۰۰۱ ۰۱۱۰ نشان داده می شود. در جدول ۱-۵ سیستم های اعداد به صورت یک جا نمایش داده شده اند.

جدول ۱-۵: نمایش اعداد در مبناهای عددی مختلف

دهدهی	باینری	اکتال	هگزادسیمال	BCD
۰	۰۰۰۰	۰	۰	۰۰۰۰
۱	۰۰۰۱	۱	۱	۰۰۰۱
۲	۰۰۱۰	۲	۲	۰۰۱۰
۳	۰۰۱۱	۳	۳	۰۰۱۱
۴	۰۱۰۰	۴	۴	۰۱۰۰
۵	۰۱۰۱	۵	۵	۰۱۰۱
۶	۰۱۱۰	۶	۶	۰۱۱۰
۷	۰۱۱۱	۷	۷	۰۱۱۱
۸	۱۰۰۰	۱۰	۸	۱۰۰۰
۹	۱۰۰۱	۱۱	۹	۱۰۰۱
۱۰	۱۰۱۰	۱۲	A	
۱۱	۱۰۱۱	۱۳	B	
۱۲	۱۱۰۰	۱۴	C	
۱۳	۱۱۰۱	۱۵	D	
۱۴	۱۱۱۰	۱۶	E	
۱۵	۱۱۱۱	۱۷	F	

۱-۲- منطق دودویی

منطق دودویی با متغیرهایی که دو ارزش جدا از هم می‌گیرند و با عملیاتی که مفهوم منطقی دارند سروکار دارد. دو ارزشی که متغیرها می‌گیرند ممکن است با اسامی مختلف نام برده شوند (مثلاً: درست و نادرست یا بلی و خیر و ...) اما برای مقصود ما مناسب است آنها را به صورت بیت در نظر گرفته، مقادیر "۰" و "۱" را به آنها اختصاص دهیم. منطق دودویی به منظور پردازش اطلاعات دودویی به فرم ریاضی به کار می‌رود.

این روش، به خصوص برای تحلیل و طراحی سیستم‌های دیجیتال مناسب است. مدارهای منطقی دیجیتال که محاسبات دودویی را انجام می‌دهند مداراتی هستند که طرز کارشان با توجه به

متغیرهای دودویی و عملیات منطقی به بهترین وجه بیان می‌شود. به این منطق، جبر بول^۱ نیز گفته می‌شود.

۱-۲-۱- تعریف منطق دودویی

منطق دودویی شامل متغیرهای دودویی و عملیات منطقی است. هر متغیر دودویی فقط و فقط می‌تواند دو ارزش "۰" یا "۱" را داشته باشد. در این منطق سه نوع عملیات منطقی اصلی به ترتیب زیر وجود دارد:

۱- AND: این عمل به وسیله یک نقطه و یا عدم وجود عملگر بین دو عملوند (متغیر) نمایش داده می‌شود. مثلاً: $x \cdot y = z$ یا $xy = z$ و بدین صورت تفسیر می‌شود که از نظر منطقی $z = 1$ خواهد بود اگر و تنها اگر $x = 1$ و $y = 1$ باشند، در غیر این صورت $z = 0$ است. (توجه کنید که x و y و z متغیرهای دودویی بوده، تنها می‌توانند مقادیر "۰" یا "۱" را اختیار کنند)

۲- OR: این عمل با علامت "+" نمایش داده می‌شود. برای مثال $x + y = z$ و مفهوم آن این است که $z = 1$ خواهد بود اگر و تنها اگر $x = 1$ یا $y = 1$ باشند و یا اینکه هر دو، مقدار "۱" داشته باشند. در صورتی که $x = 0$ و $y = 0$ باشد آن‌گاه $z = 0$ خواهد بود. به عبارت دیگر $z = 1$ خواهد بود اگر و تنها اگر حداقل یکی از دو متغیر x یا y مقدار "۱" داشته باشند.

۳- NOT: این عمل با علامت پریم (') و یا یک خط در بالای متغیر نشان داده می‌شود. مثلاً $x' = z$ یا $\bar{x} = z$ ، مفهوم آن این است که اگر $x = 1$ باشد آن‌گاه $z = 0$ است و بالعکس اگر $x = 0$ باشد آن‌گاه $z = 1$ خواهد بود.

این تعاریف را می‌توان با استفاده از جداول درستی فهرست نمود. یک جدول درستی، متشکل از تمام ترکیبات ممکن متغیرها و بیانگر ارتباط بین مقادیر آنها و نتایج حاصل از عملیات مربوطه بر روی آنها می‌باشد. جداول درستی AND و OR و NOT در جدول ۱-۶ نشان داده شده‌اند.

جدول ۱-۶: جدول درستی عملیات منطقی AND، OR و NOT

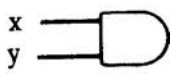
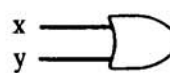
AND			OR			NOT	
x	y	xy	x	y	x + y	x	x'
۰	۰	۰	۰	۰	۰	۰	۱
۰	۱	۰	۰	۱	۱	۱	۰
۱	۰	۰	۱	۰	۱		
۱	۱	۱	۱	۱	۱		

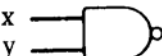
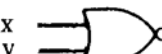
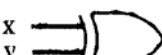
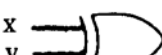
۱-۲-۲- سایر عملگرهای منطقی

هنگامی که عملگرهای منطقی AND و OR بین دو متغیر x و y قرار گیرند به ترتیب توابع بولی $x.y$ و $x + y$ حاصل خواهد شد.

عملگرهای دیگری نیز وجود دارند که توابع جدیدی معرفی می‌نمایند. اگر چه هر تابع را می‌توان بر حسب عملگرهای منطقی AND، OR و NOT بیان کرد اما دلیلی وجود ندارد که عملگر خاصی برای بیان سایر توابع تعیین نگردد. در جدول ۱-۷ کلیه عملگرهای منطقی دیجیتال و توابع متناظر آنها آمده است.

جدول ۱-۷: عملگرهای منطقی و توابع متناظر با آنها

نام	نماد ترسیمی	تابع جبری	جدول درستی		
			x	y	F
AND		$F = xy$	۰	۰	۰
			۰	۱	۰
			۱	۰	۰
			۱	۱	۱
OR		$F = x + y$	۰	۰	۰
			۰	۱	۱
			۱	۰	۱
			۱	۱	۱

			<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	x	F	0	1	1	0									
x	F																	
0	1																	
1	0																	
Inverter		$F = x'$																
			<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
Buffer		$F = x$																
			<table><tr><th>x</th><th>F</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	x	F	0	0	1	1									
x	F																	
0	0																	
1	1																	
			<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	1	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	1																
0	1	1																
1	0	1																
1	1	0																
NAND		$F = (xy)'$																
			<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																
			<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																
NOR		$F = (x + y)'$																
			<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																
			<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	x	y	F	0	0	0	0	1	1	1	0	1	1	1	0
x	y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	0																
Exclusive-OR (XOR)		$F = x \oplus y$ $= xy' + x'y$																
			<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																
			<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																
Exclusive-NOR (XNOR)		$F = x \odot y$ $= xy + x'y'$																
			<table><tr><th>x</th><th>y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	x	y	F	0	0	1	0	1	0	1	0	0	1	1	1
x	y	F																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

۱-۳- مدارهای ترتیبی و فلیپ فلاپ‌ها

مدارهای دیجیتالی که با استفاده از توابع ذکر شده مورد توجه قرار می‌گیرند همگی ترکیبی‌اند. خروجی این گونه مدارها در هر لحظه از زمان کاملاً وابسته به ورودی‌های همان زمان است. اگرچه احتمالاً هر سیستم دیجیتال دارای مدارهای ترکیبی است اما در عمل، بیشتر سیستم‌ها شامل عناصر حافظه نیز می‌باشند. این گونه سیستم‌ها باید ضرورتاً بر حسب منطق ترتیبی بررسی شوند. یکی از مهمترین عناصر استفاده شده در مدارهای ترتیبی فلیپ فلاپ‌ها هستند.

فلیپ فلاپ‌ها سلولهای دودویی هستند که قادر به ذخیره یک بیت اطلاعات می‌باشند. مدار یک فلیپ فلاپ دارای دو خروجی است، یکی برای مقدار طبیعی بیت ذخیره شده در آن و دیگری برای متمم آن (Q و Q').

یک فلیپ فلاپ قادر است تا وقتی که تحت تأثیر سیگنال ورودی برای تغییر حالت قرار نگرفته، یک حالت دودویی را به طور نامحدود در خود نگهداری کند. (البته تا زمانی که جریان الکتریکی لازم برای فلیپ فلاپ تأمین شده باشد).

همان گونه که ذکر شد هر فلیپ فلاپ دارای دو ورودی S (ست) و R (ری ست) و دو خروجی Q و Q' می‌باشد. این نوع فلیپ فلاپ را گاهی فلیپ فلاپ RS و یا لچ RS^1 می‌نامند.

۱-۴- اجزای حافظه و انواع آن

حال که تا حدودی با مفاهیم منطقی آشنا شدیم به ذکر مواردی در زمینه ریز پردازنده‌ها، حافظه‌ها و ... می‌پردازیم.

بیت (bit): این کلمه شکل خلاصه شده $binary\ digit$ بوده، مقدار آن در سیستم‌های عددی دودویی "۰" یا "۱" می‌باشد. در پردازش و ذخیره‌سازی، بیت کوچکترین واحد اطلاعات است که کامپیوتر مورد استفاده قرار می‌دهد و به طور فیزیکی به وسیله پالسی که به یک مدار ارسال می‌گردد و یا به شکل نقطه کوچکی روی دیسک مغناطیسی که قابلیت ذخیره‌سازی "۰" و "۱" را دارد، مشخص می‌گردد.

بیت‌ها کمترین اطلاعات قابل فهم برای انسان را ارائه می‌کنند. (مانند "۰" یا "۱"، درست یا نادرست، ۰ ولت یا ۵ ولت، ۰ ولت یا ۲۴ ولت و ...)

بیت‌ها در گروه‌های هشت‌تایی، بایت‌ها را تشکیل می‌دهند که جهت ارائه انواع اطلاعات از جمله حروف الفباء، ارقام صفر تا ۹ و ... مورد استفاده قرار می‌گیرند.

بایت (byte): شکل خلاصه شده binary term بوده و یک واحد اطلاعاتی است که از ۸ بیت تشکیل می‌شود. در پردازش و ذخیره‌سازی اطلاعات کامپیوتر، یک بایت معادل یک کاراکتر مثل یک حرف، عدد یا علامت است. چون بایت نشانگر مقدار اطلاعات بسیار کمی می‌باشد مقادیر حافظه و منابع ذخیره اطلاعات کامپیوتر معمولاً به کیلوبایت (۱۰۲۴ بایت) یا مگابایت (۱۰۴۸۵۷۶ بایت) اندازه‌گیری می‌شوند.

از آنجایی که یک بایت از ۸ بیت تشکیل می‌شود و هر بیت می‌تواند یکی از دو مقدار "۰" یا "۱" را داشته باشد پس در یک بایت $2^8 = 256$ حالت مختلف (از ۰۰۰۰۰۰۰۰ تا ۱۱۱۱۱۱۱۱) قابل نمایش است.

کلمه (word): هر کلمه از دو بایت یعنی ۱۶ بیت تشکیل می‌گردد پس در یک کلمه $2^{16} = 65536$ حالت مختلف قابل نمایش است.

۱-۴-۱- گذرگاه یا مسیر عمومی (Bus)

Bus در لغت به معنی اتوبوس یا وسیله حمل و نقل عمومی بوده، در اصطلاح کامپیوتری وسیله‌ای است که حمل و نقل عمومی داده‌ها را بر عهده دارد.

در این گذرگاه، قسمتی که حمل و نقل و جابجایی اطلاعات را بر عهده دارد دیتاباس (data bus) می‌نامند و به قسمتی از مسیر عمومی که جابجایی آدرس‌ها را بر عهده دارد آدرس باس (address bus) گفته می‌شود.

این گذرگاه مجموعه‌ای از خطوط سخت‌افزاری است که جهت انتقال داده‌ها بین اجزای یک سیستم کامپیوتری مورد استفاده قرار می‌گیرد.

به عبارت دیگر، گذرگاه، یک مسیر مشترک است که بین بخشهای مختلف سیستم از جمله ریزپردازنده، حافظه و درگاههای ورودی خروجی (I/O) و دیگر قسمت‌ها ارتباط برقرار می‌نماید.

در سیستم‌های کامپیوتری، گذرگاه توسط ریزپردازنده کنترل شده، به انتقال انواع مختلفی از اطلاعات اختصاص می‌یابد. به عنوان مثال، گروهی از خطوط، داده‌ها را انتقال داده، گروه دیگر آدرس‌های محل‌های استقرار اطلاعات را منتقل ساخته، یک گروه دیگر سیگنال‌های کنترل را جهت حصول اطمینان از اینکه بخشهای مختلف سیستم از مسیر مشترک خود بدون ایجاد تداخل استفاده می‌کنند، عبور می‌دهند که به این بخش از گذرگاه control bus گویند. گذرگاهها با تعداد بیت‌هایی که در هر لحظه می‌توانند انتقال دهند مشخص می‌شوند به عنوان مثال یک کامپیوتر دارای گذرگاه ۸ بیتی در هر لحظه ۸ بیت از داده‌ها و یک کامپیوتر دارای گذرگاه ۱۶ بیتی در هر لحظه ۱۶ بیت از داده‌ها را انتقال می‌دهند.

۱-۴-۲- حافظه (Memory)

حافظه یکی از قسمت‌های اساسی در سیستم‌های کامپیوتری می‌باشد. از حافظه جهت ذخیره نمودن دستورالعمل‌ها، اطلاعات و نتایج به دست آمده استفاده می‌شود. همان گونه که قبلاً ذکر شد اطلاعات ذخیره شده در حافظه به صورت باینری ("۰" یا "۱") می‌باشد. برای ساختن حافظه معمولاً از نیمه هادی‌ها استفاده می‌شود. در ادامه بحث به ذکر انواع مختلف حافظه در سیستم‌های دیجیتالی و کامپیوتری خواهیم پرداخت.

۱-۴-۳- حافظه‌های با دسترسی تصادفی (RAM)^۱

این حافظه‌ها به گونه‌ای طراحی شده‌اند که در هر لحظه به هر سلول آن می‌توان دست یافت و اطلاعات موجود در آن بخش را خواند.

این نوع حافظه به سه دسته زیر تقسیم می‌گردد:

(الف) حافظه‌های فقط خواندنی (ROM)

(ب) حافظه‌های اغلب خواندنی (RMM)

(ج) حافظه‌های خواندنی نوشتنی (RWM)

در ادامه به شرح این تقسیم‌بندی می‌پردازیم.

1 - Random Access Memory

الف) حافظه‌های فقط خواندنی (ROM)^۱

حافظه‌های فقط خواندنی امروزه با استفاده از تکنولوژی LSI^۲ ها و VLSI^۳ ها در حد وسیع و انواع مختلف تولید می‌شوند. محتوای این حافظه‌ها غیر قابل تغییر می‌باشد. بدین معنی که وقتی بر روی این حافظه‌ها اطلاعاتی نوشته شود، در حالت عادی نمی‌توان آن را تغییر داد. همچنین با قطع و وصل منبع تغذیه محتوای آنها تغییر نخواهد کرد. به عبارت دیگر این گونه حافظه‌ها پاک نشدنی هستند. به همین جهت این نوع حافظه‌ها برای نگهداری برنامه و یا جداول اطلاعاتی که همیشه ثابت و بدون تغییرند بسیار مناسب می‌باشد.

حافظه‌های فقط خواندنی به دو گروه تقسیم می‌شوند:

۱- ROM (Read Only Memory)

۲- PROM (Programmed Read Only Memory)

در صورتی که محتوای این نوع حافظه در موقع ساخت توسط سازنده برنامه‌ریزی شود به آن ROM گفته می‌شود ولی اگر به گونه‌ای باشد که توسط مصرف کننده و تنها برای یک بار قابل برنامه‌ریزی باشد به آن PROM می‌گویند.

ب) حافظه‌های اغلب خواندنی (RMM)^۳

این نوع حافظه نیز مانند ROM بوده، از آن جهت نگهداری اطلاعات مختلف استفاده می‌شود. اگر در ثبت بیت‌های اطلاعاتی حافظه‌های ROM و PROM که فقط برای یک بار قابل برنامه‌ریزی هستند اشتباهی رخ دهد راهی جز دور انداختن حافظه وجود ندارد. اما این گروه از حافظه‌ها که می‌توان محتویات آنها را پاک کرد این ضعف را برطرف می‌کند و می‌توان از آنها چندین بار استفاده نمود و برنامه‌های مختلف را در آنها ضبط و پس از اتمام کار آنها را پاک کرد. این نوع حافظه‌ها بر اساس نوع پاک شدن اطلاعات به دو گروه تقسیم‌بندی می‌شوند. (البته خاصیت پاک شدن آنها مربوط به تکنولوژی ساخت آنها است).

۱- EPROM (Erasable Programmed Read Only Memory)

۲- EEPROM (Electrically Erasable Programmed ROM)^۴

1 - Read Only Memory

2 - Large Scale Integration

3 - Read Mostly Memory

۴ - این نوع حافظه را با E² PROM نیز نمایش می‌دهند.

ج) حافظه‌های خواندنی نوشتنی (RWM)^۱

این نوع حافظه اغلب در سیستم‌های میکروپروسسوری برای نگهداری نتایج موقت و میانی در هنگام اجرای یک برنامه به کار می‌روند. ساختمان داخلی هر سلول حافظه RWM اساساً یک فلیپ‌فلاپ می‌باشد که دو حالت پایدار برای ذخیره‌سازی منطق "۱" یا "۰" داشته، پیش‌بینی‌های لازم برای تغییر حالت سریع سلول حافظه در آن به عمل آمده است و سیگنال‌های کنترل تعیین می‌کنند که اطلاعات در محل آدرس داده شده نوشته یا از آن خوانده شود. این گروه به دو بخش استاتیک و دینامیک تقسیم‌بندی می‌شوند.

در جدول ۸-۱ انواع حافظه‌ها و اطلاعات دیگر در رابطه با نحوه پاک شدن و ... آمده است.

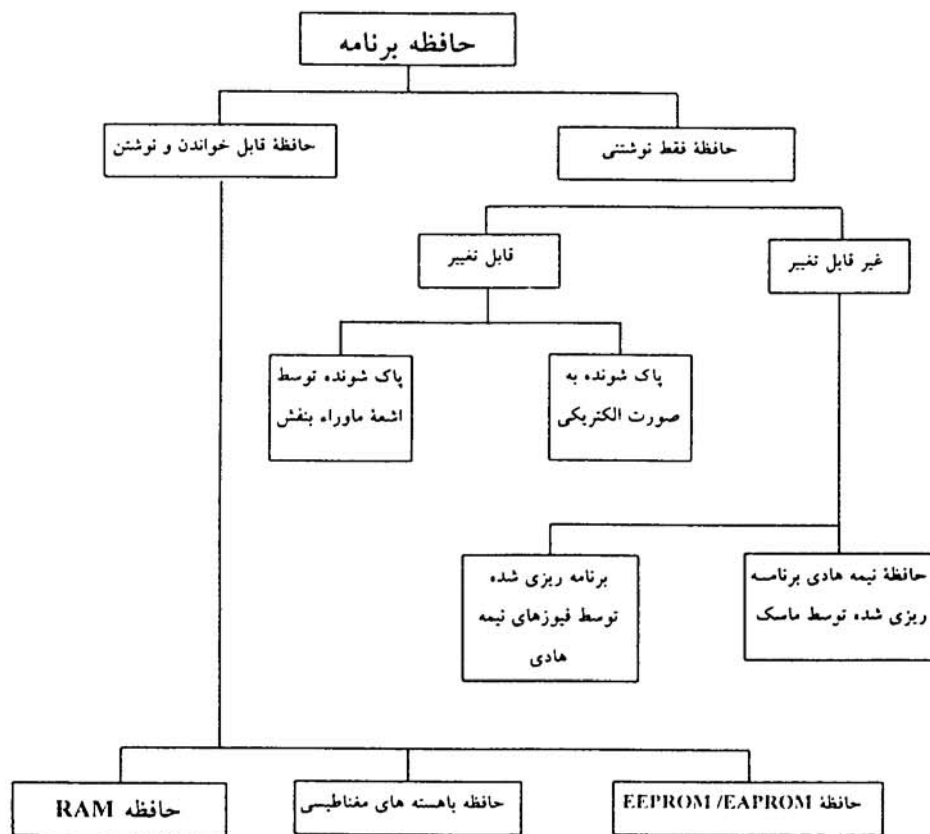
جدول ۸-۱: انواع حافظه‌ها و سایر اطلاعات در مورد آنها

محتوای حافظه پس از قطع تغذیه	برنامه‌ریزی	طریقه پاک شدن	نوع حافظه	ساختمان
ناپایدار	الکتريکی	الکتريکی	حافظه با دسترسی اتفاقی	RAM
پایدار	توسط کارخانه‌سازنده	غیرممکن	حافظه فقط خواندنی	ROM
			ROM قابل برنامه‌ریزی	PROM
	الکتريکی	توسط اشعه ماوراء بنفش	PROM قابل پاک شدن	EPROM
			PROM قابل برنامه‌ریزی مجدد	REEPROM
		الکتريکی	ROM با قابلیت پاک شدن الکتريکی	EEPROM
			ROM با قابلیت پاک شدن الکتريکی	EAPROM ^۲

در شکل ۱-۱ چگونگی تقسیم‌بندی حافظه‌ها به صورتی دیگر آورده شده است.

1 - Read Write Memory

2 - Electrical Alterable PROM



شکل ۱-۱: نحوه تقسیم بندی حافظه ها

حال که با اجزای تشکیل دهنده حافظه و انواع حافظه ها آشنا شدیم به بحث پیرامون سیستم های کنترل می پردازیم.

۵-۱- سیستم های کنترل

در سالهای اخیر، سیستم های کنترل اهمیت فزاینده ای در توسعه و پیشرفت تکنولوژی جدید

یافته‌اند. هر یک از جنبه‌های فعالیت روزمره ما عملاً تحت تأثیر نوعی سیستم کنترل قرار دارد. مثلاً در محدوده زندگی فردی، کنترل‌کننده‌های خودکار در سیستم‌های تهویه مطبوع، دما و رطوبت هوای خانه‌ها و ساختمان‌ها را در حد مطلوب نگاه می‌دارند.

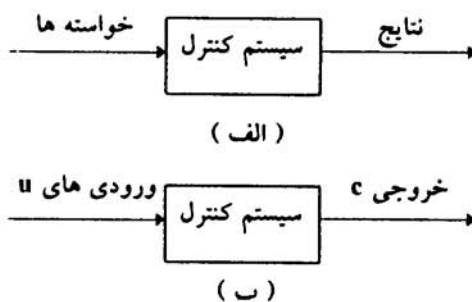
سیستم‌های کنترل در تمام بخشهای صنعت نظیر کنترل کیفیت محصولات، خط مونتاژ خودکار، کنترل ماشین‌ابزار، تکنولوژی فضایی و سیستم‌های نظامی، کنترل کامپیوتری، سیستم‌های حمل و نقل، سیستم‌های قدرت، آذمهای ماشینی و در موارد بسیار دیگری، به فراوانی یافت می‌شوند. صرف‌نظر از اینکه چه نوع سیستم کنترلی در اختیار داریم، سه بخش اساسی را می‌توان در آن مشخص کرد:

۱- خواسته‌های ما از سیستم کنترل

۲- اجزای سیستم کنترل

۳- نتایج

در شکل ۱-۲ (الف) ارتباط اساسی میان این سه بخش به شکل نمودار بلوکی نمایش داده شده است. همان‌طور که در شکل ۱-۲ (ب) دیده می‌شود این سه بخش اساسی به ترتیب با عناوین ورودی‌ها، اجزای سیستم و خروجی‌ها که اصطلاحات علمی تری هستند نیز شناخته می‌شوند.

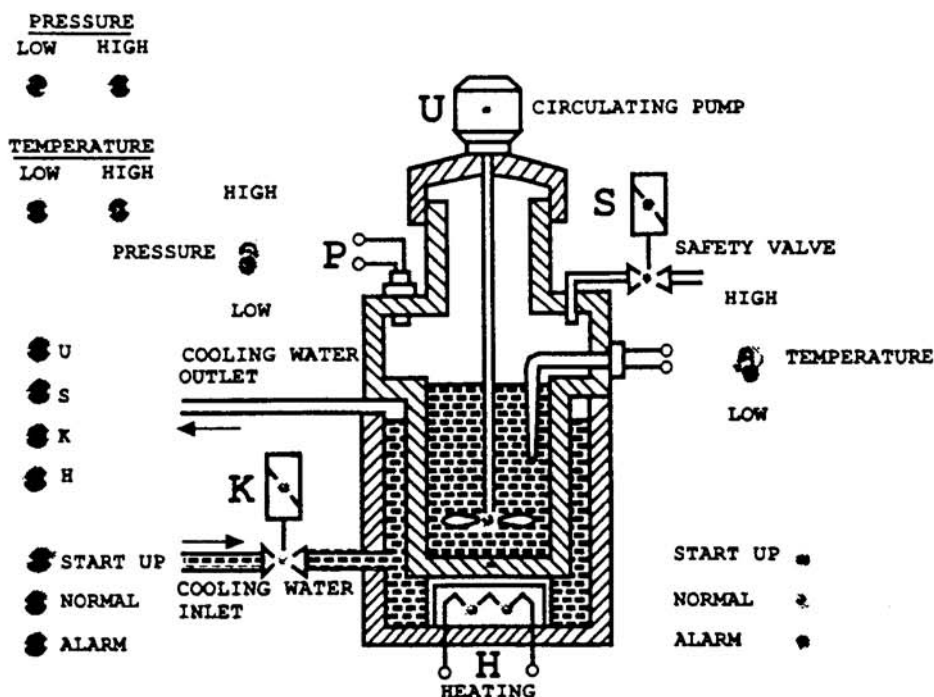


شکل ۱-۲: بخش‌های یک سیستم کنترل

به‌طور کلی، هدف سیستم‌های کنترل این است که خروجی‌های c را به شیوه‌ای از پیش تعیین شده‌ای به وسیله ورودی‌های u از طریق اجزای سیستم، کنترل کند. ورودی‌های سیستم کنترل، سیگنال‌های تحریک و خروجی‌های آن، متغیرهای تحت کنترل نیز نامیده می‌شوند.

به عنوان مثال با ثابت نگه داشتن متغیرهای زیر می توان فرآیند شیمیایی موجود در شکل ۳-۱ را کنترل نمود.

- موتور همزن الکتریکی (CIRCULATING PUMP)
- فشار هوای درون مخزن و شیر اطمینان (SAFETY VALVE و PRESSURE)
- درجه حرارت سیال داخل مخزن (TEMPERATURE)
- مقدار آب خنک کننده (COOLING WATER)
- زمان (TIME)



شکل ۳-۱: یک فرآیند شیمیایی تحت کنترل و موارد مهم در کنترل آن

۱-۵-۱- ساختار سیستم‌های کنترل

سیستم‌های کنترل از لحاظ ساختاری به دو بخش زیر تقسیم می‌شوند.

الف) سیستم‌های کنترل حلقه باز (Open Loop)

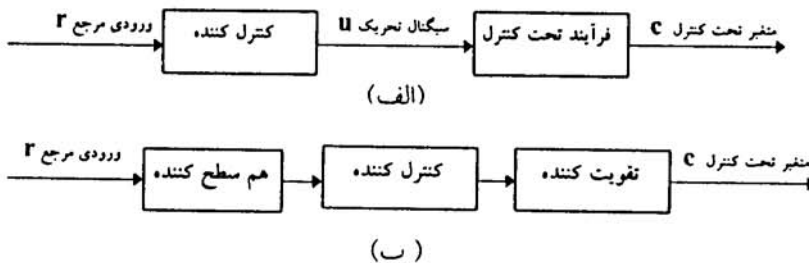
ب) سیستم‌های کنترل حلقه بسته (Closed Loop)

الف) سیستم‌های کنترل مدار باز (Open Loop)

در این گونه سیستم‌ها که در شکل ۴-۱ الف) نشان داده شده است خواسته‌های ما از عملکرد آن به خوبی برآورده نمی‌شود و تنها به دلیل سادگی و اقتصادی بودن سیستم‌های کنترل مدار باز، در بسیاری موارد می‌توان آنها را در حال کار یافت.

ماشین لباسشویی مثال بارزی از یک سیستم کنترل مدار باز است. زیرا عموماً مدت زمان شستشو از طریق قضاوت و تخمین فرد استفاده‌کننده تعیین می‌شود. یک ماشین لباسشویی خودکار باید بتواند دائماً میزان تمیزی لباس‌های در حال شستشو را بررسی کند تا هر زمان که به اندازه کافی تمیز شدند، بطور خودکار خاموش شود.

همان‌طور که در شکل ۴-۱ الف) نشان داده شده است، اجزای یک سیستم کنترل مدار باز را معمولاً می‌توان به دو دسته تقسیم نمود. کنترل‌کننده و فرآیند تحت کنترل. یک فرمان یا سیگنال ورودی r به کنترل‌کننده اعمال می‌شود، خروجی کنترل‌کننده به عنوان سیگنال تحریک u ، فرآیند را کنترل می‌کند به نحوی که متغیر تحت کنترل c بر اساس استانداردهای از پیش تعیین شده‌ای عمل کند.



شکل ۴-۱: قسمت‌های مختلف یک سیستم کنترل مدار باز

در موارد ساده، کنترل‌کننده می‌تواند بر حسب طبیعت سیستم، یک تقویت‌کننده، اتصالات

مکانیکی و یا وسایل کنترلی دیگر باشد. در موارد پیچیده تر کنترل الکترونیکی، کنترل کننده ممکن است یک حسابگر الکترونیکی نظیر یک ریزپردازنده باشد.

همان گونه که در شکل ۱-۴ (ب) نشان داده شده، گاهی ممکن است علاوه بر واحد کنترل کننده قسمتهای دیگری در سیستم کنترل مدار باز وجود داشته باشد. در ادامه بحث به توضیح در مورد اجزای یک سیستم کنترل مدار باز (اجزای نشان داده شده در شکل ۱-۴ (ب)) می پردازیم.

۱- واحد ورودی (ورودی مرجع r): اطلاعات از طریق کلیدها و حس کنندها (سنسورها) به ورودی ها منتقل می گردد. این اطلاعات معمولاً به شکل سیگنال های الکتریکی است که می تواند به صورت دیجیتال یا آنالوگ باشد.

۲- واحد متناسب کننده یا هم سطح کننده (Conditioning Unit): این قسمت در صورتی مورد نیاز است که میزان ولتاژ یا جریان سیگنال های ورودی، برای واحد کنترل کننده مناسب نباشد. بنابراین باید مقدار ولتاژ یا جریان به میزان معینی تنظیم گردد.

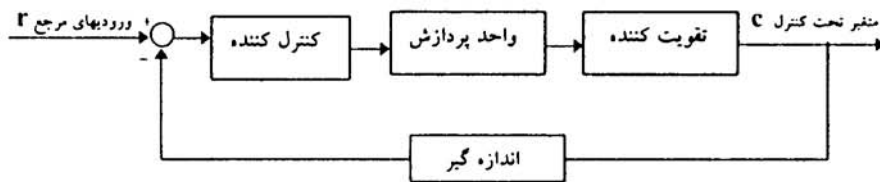
۳- واحد کنترل کننده (واحد پردازش): این واحد، هسته سیستم کنترل را تشکیل داده، اعمال منطقی و محاسباتی در این قسمت صورت می گیرد.

۴- واحد تقویت کننده: در این قسمت سیگنال های ضعیف دریافت شده از واحد پردازش، تقویت و ارسال شده تا وسایل کنترل کننده مانند شیرهای کنترل شونده توسط جریان الکتریکی (Solenoid Valves) یا کنتاکتورها را کنترل، یا لامپ های نشان دهنده را روشن نماید.

۵- واحد خروجی (متغیر تحت کنترل c): از این قسمت فرمانهای سیستم کنترل به اجرا کننده های فرمان یعنی محرک ها (Actuators) و ... ارسال می شود.

ب) سیستم های کنترل مدار بسته (Closed Loop)

آنچه برای کنترل دقیق تر و قابل انعطاف تر لازم بوده و در سیستم کنترل مدار باز وجود ندارد یک اتصال یا فیدبک از خروجی، به ورودی سیستم است. برای دستیابی به کنترل دقیق تر، سیگنال تحت کنترل $c(t)$ باید فیدبک شده، با ورودی مرجع r مقایسه شود و سیگنال تحریکی متناسب با تفاضل ورودی و خروجی به سیستم اعمال و در نتیجه، خطا تصحیح شود. سیستمی با یک یا چند مسیر فیدبک، نظیر آنچه که هم اکنون تشریح شد یک سیستم مدار بسته نامیده می شود. در شکل ۱-۵ طرز کار این گونه سیستم ها نشان داده شده است.



شکل ۱-۵: قسمت‌های مختلف در سیستم‌های کنترل مدار بسته

همان گونه که ذکر شد در این سیستم‌ها، مقدار خروجی توسط عنصری اندازه‌گیر (سنسور) اندازه‌گیری می‌شود و در مقایسه‌ای با مقدار مطلوب، اختلاف بین خروجی و مقدار مطلوب محاسبه شده، به عنوان سیگنال خطا معرفی می‌گردد. این سیگنال خطا به واحد کنترل کننده ارسال و سپس کنترل کننده در حلقه کنترل، به عنوان تصمیم گیرنده عمل می‌نماید. به این ترتیب که با توجه به تنظیم‌های از پیش تعیین شده، فرمان لازم را برای تصحیح خطا صادر می‌نماید.

۱-۶- انواع سیستم‌های کنترل

سیستم‌های کنترل را می‌توان بنا به روش کنترل آنها به دو دسته زیر تقسیم نمود:

۱- سیستم‌های کنترل سخت‌افزاری

۲- سیستم‌های کنترل نرم‌افزاری

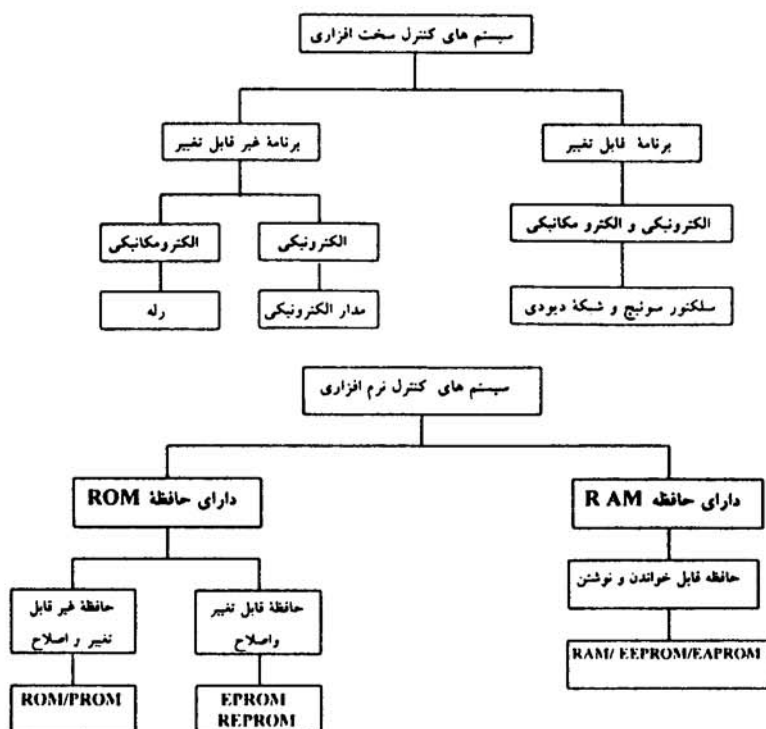
۱-۶-۱- سیستم‌های کنترل سخت‌افزاری

این سیستم‌ها شامل مداراتی هستند که با استفاده از رله‌ها و عناصر الکترونیکی مانند دیودها و ترانزیستورها ساخته می‌شوند. برنامه کنترل در این سیستم‌ها نتیجه روابط بین عناصر مدار الکتریکی است و به راحتی قابل تغییر نمی‌باشد. به عبارت دیگر تغییر در برنامه کنترل به معنی تغییر در سخت‌افزار سیستم است البته در برخی از این کنترل کننده‌ها که با استفاده از کلیدهای انتخاب کننده (Selectors)، یا شبکه دیودی (Diode Matrix) ساخته می‌شوند، برنامه کنترل را می‌توان تا اندازه‌ای تغییر داد. اگرچه اعمال این تغییرات، محدود و در برخی موارد بسیار مشکل است.

۱-۶-۲- سیستم‌های کنترل نرم‌افزاری

این کنترل کننده‌ها دارای حافظه‌ای هستند که برنامه کنترل در آن ذخیره می‌شود. مهمترین مزیت

این سیستم‌ها در آن است که نحوه کنترل را با تغییر برنامه و بدون نیاز به تغییر در سخت‌افزار سیستم می‌توان عوض کرد، زیرا نحوه کنترل سیستم توسط سخت‌افزار سیستم تعیین نمی‌شود بلکه برنامه‌ای که در حافظه ذخیره شده یعنی نرم‌افزار سیستم، نحوه کنترل را مشخص می‌کند لذا این سیستم‌ها بسیار قابل انعطاف بوده، کاربردهای فراوانی دارند. بسته به نوع حافظه این سیستم‌ها، شیوه تغییر در برنامه‌ها متفاوت است. اگر از حافظه RAM استفاده شود، بدون دخالت فیزیکی و تنها با اضافه یا کم نمودن چند سطر برنامه می‌توان برنامه جدید را به اجرا در آورد. در صورتی که از حافظه ROM استفاده شود به اجرا درآوردن برنامه جدید تنها با تعویض حافظه ROM امکان‌پذیر است. در شکل ۱-۶ انواع سیستم‌های کنترل نشان داده شده‌اند.



شکل ۱-۶: سیستم‌های کنترل سخت‌افزاری و نرم‌افزاری

حال که با مفاهیم منطقی و ساختار سیستم‌های کنترل آشنا شدیم در فصل آینده به بحث پیرامون PLC و سخت‌افزار و نرم‌افزار آن خواهیم پرداخت.