درس

برنامه نویسی کامپیوتر

Computer programming

1399

جزیرئیان

مبانی کامپیوتر، الگوریتم و فلوچارت

۱-۱ مبانی کامپیوتر

یکی از روش‌های ارتباطی استفاده از زبان قابل درک توسط طرفین است. یعنی چگونه بتوان نمادها و نشانه‌ها را به طرف مقابل شناساند. برخی از نشانه‌های ارتباطی عبارتند از:

ارقام: 0 تا 9

حروف الفبا: A تا Z و یا a تا z

نمادهای عملیاتی: +، −، *، / و ...

نشانه‌های ویژه: ?، ( )، & و ...

برای نمایش این علائم و نمادها می‌توان از یک سری کد یا رمز استفاده نمود. انسان در طول تاریخ بارها از رمزگذاری برای بیان مفاهیم خود استفاده کرده که منجر به ایجاد انواع زبان‌های ارتباطی مختلف نظیر اختراع خط، خط بریل، نُت موسیقی، کُدهای کامپیوتری و ... شده است. بعنوان مثال در حالت خیلی ساده می‌توان با خاموش یا روشن‌کردن یک لامپ به کسی فهماند که با روشن‌شدن آن ایستاده و با خاموش‌کردن لامپ بنشیند. زبان‌هایی که مراحل حل یک مساله یا فرآیند را برای کامپیوتر تشریح می‌کنند، زبان‌های برنامه‌نویسی نام دارند. زبان‌های برنامه‌نویسی نیز مانند زبان‌های محاوره‌ای و طبیعی دارای قوانین و مقررات خاصی هستند. معمولاً زبان‌های برنامه‌نویسی به دو دسته ذیل تقسیم می‌شوند:

۱) زبان‌های برنامه‌نویسی سطح پایین

۲) زبان‌های برنامه‌نویسی سطح بالا

زبان برنامه نویسی

مجموعه‌ای از دستورالعمل‌ها و ساختارها و قواعد که برای ارتباط بین برنامه‌های کاربردی و سیستم کامپیوتر از آن استفاده می‌شود.

تقسیم بندی زبان برنامه نویسی بر اساس درک و قابلیت خوانایی

۱) زبان سطح پایین

۲) زبان سطح میانی

۳) زبان سطح بالا

۱-۱-۱ زبان‌های برنامه‌نویسی سطح پایین (*Low level language*)

منظور زبان‌های برنامه‌نویسی است که به زبان کامپیوتر نزدیک‌تر است. نظیر زبان ماشین و زبان اسمبلی.

۱-۱-۱-۱ زبان ماشین (*Machine language*)

مجموعه فرامینی که مستقیماً توسط پردازنده کامپیوتر قابل رمزگشایی و اجرا باشد. زبان ماشین بصورت کدهای دودویی (*Binary*) است.

سوالی که معمولاً اینجا مطرح می‌شود این است که چگونه می‌توان دستورات اولیه قابل اجرا توسط کامپیوتر را به آن معرفی کرد؟ برای این کار از یک سری ریزپردازنده (*Micro processor*)، مدارات منطقی و گیت‌های دیجیتال بر مبنای پردازش‌های دودویی استفاده می‌گردد.

زبان ماشین در واقع زبان قابل فهم کامپیوتر بوده که از مجموعه اعداد 0 و 1 تشکیل شده است و دارای ترکیب‌بندی (*Format*) خاصی می‌باشد. در این زبان برنامه‌نویسی هیچ تفاوتی بین دستور و داده خام در کامپیوتر وجود ندارد. یعنی در واقع اعداد کُدهای اجرای دستورات و فرامین هستند و برنامه‌نویسی در آن با توجه به این‌که باید برنامه‌های تهیه‌شده در مبنای دو نوشته شوند، بسیار دشوار است. برای درک بهتر و آشنایی بیشتر با این زبان می‌توان به دفترچه پردازش‌گر مربوطه رجوع کرده و رمزهای موردنظر را گشود. از این زبان برنامه‌نویسی معمولاً برای تهیه سیستم‌عامل‌های مختلف (*Operating systems*) استفاده می‌گردد.

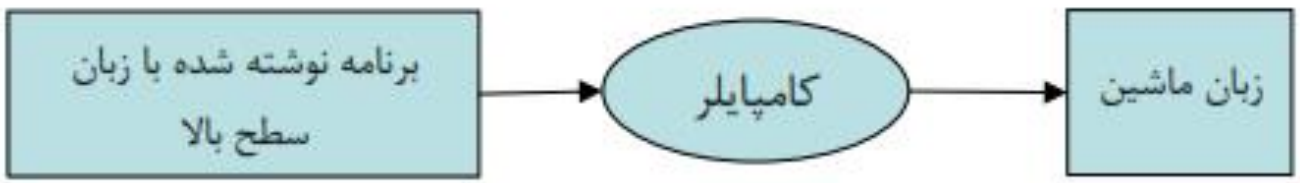
۱-۱-۲ زبان اسمبلی (***Assembly language***)

به مجموعه زبان‌های برنامه‌نویسی سطح پایینی اطلاق می‌گردد که در آن بجای کدهای زبان ماشین، با استفاده از یک سری کلمات و سمبل‌ها دستورالعمل‌های پردازنده کامپیوتر (*CPU*) نوشته می‌شود. بعنوان مثال در این زبان بجای فرمان `01011100 01011001` که بصورت کُدهای 0 و 1 است، دستور `ADD X` جایگزین می‌گردد. در زبان اسمبلی دستورات تا حد امکان کوتاه و نسبتاً با معنا هستند. مثلاً برای فرمان «محتوای X را در محل واحد محاسبه قرا بده.» عبارت `LOAD X` و بجای فرمان «حاصل را در Z ذخیره کن» عبارت `STORE Z` استفاده می‌شود.

معمولاً برای هر خانواده *CPU* یک زبان اسمبلی مخصوص به آن وجود دارد. از آنجا که در هر صورت زبان قابل فهم توسط کامپیوتر بصورت کدهای 0 و 1 (زبان ماشین) است، باید مترجمی زبان اسمبلی را به زبان ماشین تبدیل نماید که به آن اصطلاحاً *اسمبلر* گویند. زبان اسمبلی با زبان ماشین یک رابطه یک به یک دارد.

۱-۱-۲ زبان‌های برنامه‌نویسی سطح بالا (*High level language*)

منظور از زبان‌های برنامه‌نویسی سطح بالا، زبان‌هایی است که به زبان محاوره‌ای نزدیک‌تر باشد. امروزه با گسترش کامپیوترها احتیاج به زبان برنامه‌نویسی که استفاده از آن برای کاربر راحت‌تر باشد، احساس می‌گردد. مانند زبان‌های برنامه‌نویسی *C*، *Pascal*، *MATLAB* و ... که به زبان طبیعی نزدیک‌ترند. دستورات این زبان‌ها وابسته به یک پردازش‌گر خاص نیستند. برای ارتباط بین زبان‌های سطح بالا و زبان ماشین از *کامپایلر* (*Compiler*) استفاده می‌شود.



شکل ۱-۱ ارتباط بین زبان‌های سطح بالا ئ زبان ماشین

کامپایلر تمام برنامه را بررسی کرده و درصورتی که بدون اشکال باشد، آن را به زبان ماشین تبدیل می‌نماید. البته بعضی از زبان‌های برنامه‌نویسی بجای کامپایلر از مفسر (*Interpreter*) استفاده می‌کنند که برنامه را خط به خط تفسیر و اجرا می‌کند.

زبان سطح بالا

این زبان درکش برای انسان ساده بوده و عین زبان محاوره‌ای می‌باشد مانند زبان برنامه نویسی پاسکال، #c،++c و ...

اگر برنامه malab را بخواهیم جزء زبان‌های برنامه نویسی قرار دهیم. برنامه malab یک زبان برنامه نویسی سطح بالا است

به دلایل زیر برنامه نویسی matlab برای نقشه برداری مفید است

۱) یک زبان سطح بالا بوده و به راحتی می‌توان با آن کار کرد

۲) کار با ماتریس‌ها در آن بسیار ساده بوده

۳) وجود توابع مفید از قبل نوشته شده در آن

۴) امکانات بررسی مشکل در برنامه به صورت خط به خط و کاراکتر به کاراکتر

۵) محیط گرافیکی خوبی که دارد

۶) قابلیت انجام محاسبات سنگین با سرعت بالا و ...

۳-۱-۱ سیستم نمایش اعداد

اگر عدد اعشاری $a_n a_{n-1}...a_1 a_0 \bullet a_{-1} a_{-2}...a_{-m}$ در مبنای b باشد در این صورت هر یک از ارقام a_i با توجه به محل قرارگرفتن، دارای ارزش خاصی خواهند بود. بطوریکه داریم:

$$0 \le a_i \le b-1 \quad ; \quad i = -m,...,-1,0,1,...,n$$

بنابراین؛

$$(a_n a_{n-1}...a_1 a_0 \bullet a_{-1} a_{-2}...a_{-m})_b =$$

$$a_n \times b^n + a_{n-1} \times b^{n-1} + ...+ a_1 \times b^1 + a_0 \times b^0 + a_{-1} \times b^{-1} + ...+ a_{-m} \times b^{-m} = \sum_{i=-m}^{n} a_i \times b^i$$

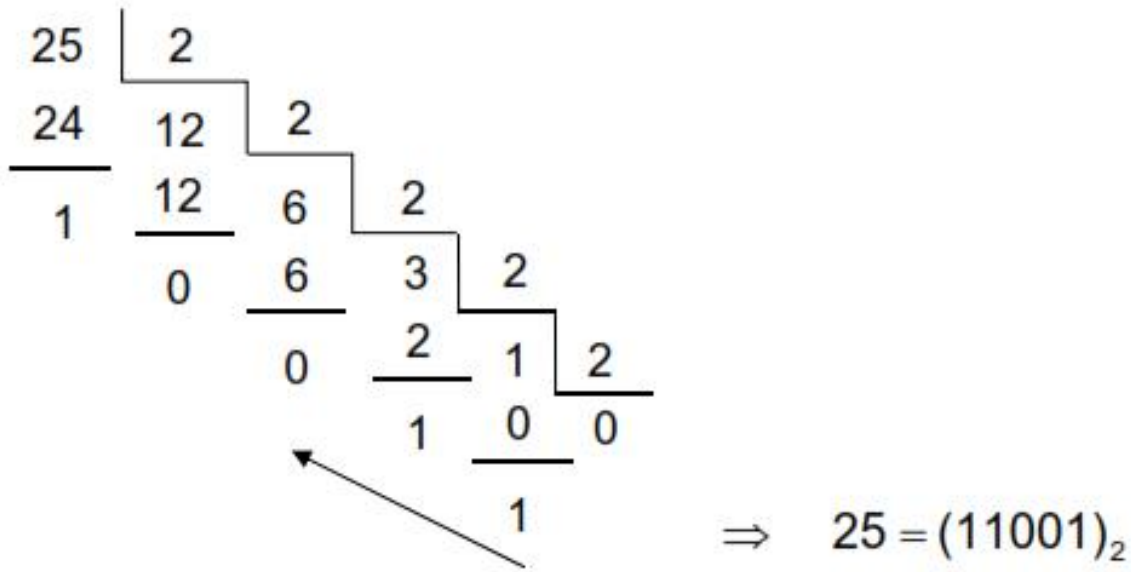
بعنوان مثال عدد ۵۳/۲۴ در مبنای ۱۰ بصورت ذیل است:

$$(53.24)_{10} = 5 \times 10^1 + 3 \times 10^0 + 2 \times 10^{-1} + 4 \times 10^{-2}$$

۱-۱-۳-۱ تبدیل اعداد صحیح به مبنای ۲

همانطور که گفته شد زبان قابل فهم کامپیوتر در هر صورتی بصورت کدهای 0 و 1، یعنی در مبنای دو می‌باشد. بنابراین در این قسمت نحوه نمایش اعداد در مبنای دو مورد بررسی قرار می‌گیرد. برای تبدیل یک عدد از مبنای ده به مبنای دو، این کار را با انجام تقسیم‌های متوالی عدد موردنظر بر عدد ۲ تا زمان صفرشدن عدد خارج قسمت ادامه می‌یابد. سپس

باقیمانده‌های حاصل از هر تقسیم نگهداری شده و از آخرین باقیمانده به اولین باقیمانده در کنار هم نوشته می‌شود. بعنوان مثال عدد ۲۵ در مبنای دو برابر است با:



۱-۱-۳-۲ تبدیل اعداد اعشاری به مبنای ۲

برای تبدیل این اعداد به مبنای دو، قسمت صحیح عدد مانند بند ۱-۱-۳-۱ با تقسیم‌های متوالی، به مبنای دو منتقل می‌شود. ولی برای تبدیل قسمت اعشاری، این قسمت با ضرب متوالی در عدد پایه مبنا، به مبنای دو انتقال می‌یابد. بدین صورت که بعد از هر مرتبه ضرب، قسمت صحیح عدد حاصل نگهداری شده و قسمت اعشاری عدد، دوباره در عدد ۲ ضرب می‌گردد. این کار تا زمانی ادامه می‌یابد که قسمت اعشاری عدد حاصل صفر شود. بعنوان مثال برای تبدیل عدد ۱۲/۲۵ به مبنای دو، عدد ۱۲ مانند قبل به مبنای دو منتقل می‌شود و داریم:

$$\Rightarrow \quad 12 = (1100)_2$$

ولی به منظور تبدیل قسمت اعشاری (۰/۲۵) به مبنای دو خواهیم داشت:

$$0.25 \times 2 = 1.0 \qquad \rightarrow 1$$
$$0.0 \times 2 = 0.0 \qquad \rightarrow 0$$

بنابراین عدد ۱۲/۲۵ در مبنای دو بصورت (1100.01) نمایش داده می‌شود. اما اگر با انجام ضرب‌های متوالی قسمت اعشاری عدد حاصل، صفر نشود این کار تا زمان پرشدن حافظه موردنظر ادامه می‌یابد.

۱-۱-۳-۳ انجام محاسبات در مبنای ۲

به منظور نمایش اطلاعات در کامپیوتر، آشنایی با نحوه انجام محاسبات در مبنای دو لازم است. بنابراین در این قسمت محاسبات ساده نظیر جمع، تفریق، ضرب و تقسیم توضیح داده می‌شود.

۱-۱-۳-۳-۱ عمل جمع و تفریق در مبنای ۲

عمل جمع و تفریق در مبنای دو تقریباً مشابه نحوه محاسبات مربوطه در مبنای ده است. با این تفاوت که در این مبنا، بجای سیستم دهدهی با سیستم دودویی سروکار داریم. یعنی در حالت جمع دو رقم ۱ باهم، نتیجه عدد صفر بوده و یک رقم ۱ به رقم سمت چپ اضافه می‌شود و یا همچنین در حالت تفریق رقم ۱ از صفر، در این صورت یک واحد مبنا (یعنی عدد ۲) از رقم سمت چپ به صفر اضافه شده و رقم ۱ از آن کسر می‌گردد. بنابراین در حالت کلی داریم:

$$0+0=0 \qquad 0-0=0$$
$$1+0=1 \qquad 1-0=1$$
$$0+1=1 \qquad 0-1=01$$
$$1+1=10 \qquad 1-1=0$$

بعنوان مثال جواب جمع و تفریق ذیل برابر است با:

$$\begin{array}{r} {\scriptstyle 1\ \ 1\ \ 1\ \ 1} \\ 1011011 \\ +\ 1001010 \\ \hline 10100101 \end{array} \qquad \begin{array}{r} {\scriptstyle 2} \\ {\scriptstyle 0\,0\,2} \\ 1011011 \\ -\ 1001110 \\ \hline 0001101 \end{array}$$

۱-۱-۳-۳-۲ عمل ضرب و تقسیم در مبنای ۲

عمل ضرب و تقسیم نیز در مبنای دو مشابه نحوه محاسبات مربوطه در مبنای ده می‌باشد. فقط باید در هنگام جمع ویا تفریق‌های لازم به موارد اشاره‌شده در بند ۱-۱-۳-۳-۱ توجه شود. بعنوان مثال جواب ضرب و تقسیم ذیل برابر است با:

$$\begin{array}{r} 1101 \\ \times\ 101 \\ \hline 1101 \\ +\ 0000 \\ 1101 \\ \hline 1000001 \end{array}$$

$$\begin{array}{r|l} 110'110.00 & 101 \\ \cline{2-2} -101 & \\ \cline{1-1} 0011 & \\ -000 & \\ \cline{1-1} 111 & \\ -101 & \\ \cline{1-1} 0100 & \\ -000 & \\ \cline{1-1} 1000 & \\ -101 & \\ \cline{1-1} 0110 & \\ -101 & \\ \cline{1-1} 001 & \end{array} \qquad 1010.11$$

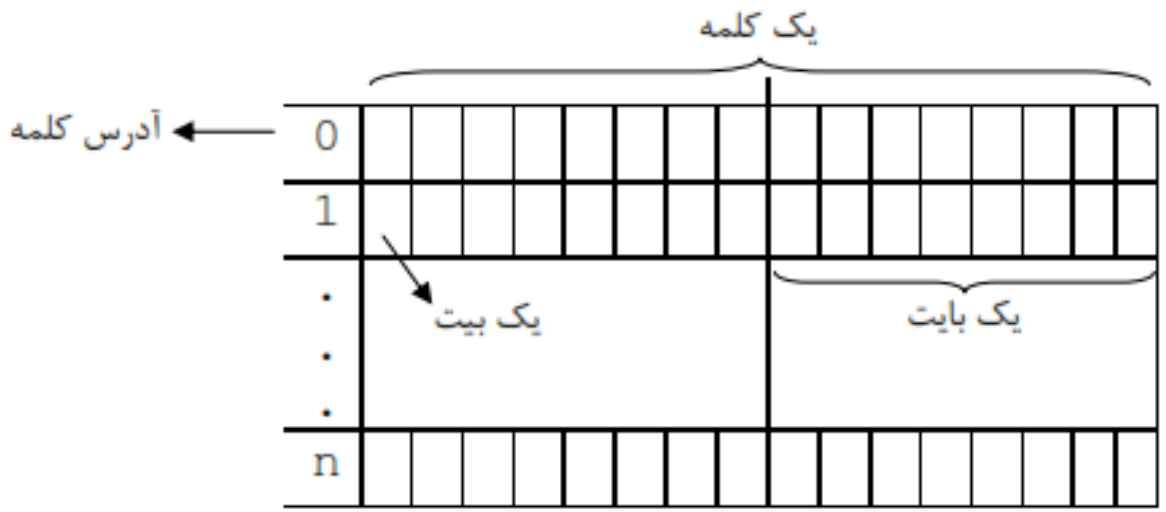
۴-۱-۱ واحد اندازه‌گیری حافظه

حافظه در واقع محل نگهداری داده‌ها و اطلاعات در کامپیوتر است و به دو بخش حافظه *اصلی* و *جانبی* تقسیم می‌شود. حافظه اصلی را می‌توان جدولی از خانه‌ها تصور کرد که به دو بخش حافظه فقط خواندنی (*ROM: Read Only Memory*) و حافظه با قابلیت دسترسی تصادفی (*RAM: Random Access Memory*) تفکیک می‌گردد. حافظه *ROM* مستلزم برنامه‌نویسی و ذخیره داده در زمان ساخت بوده و در واقع شامل برنامه تست اولیه سیستم، درایورهای ورودی و خروجی سیستم، برنامه فراخواننده سیستم‌عامل برای راه‌اندازی سیستم و الگوهای حروف برای حالت گرافیک و متن است. محتویات این حافظه توسط کاربر قابل تغییر نمی‌باشد. یک تراشه استاندارد *ROM* را نمی‌توان برنامه‌ریزی مجدد کرده و یا اطلاعات جدیدی را در آن نوشت. در صورتیکه داده‌ها درست نبوده و یا مستلزم تغییر و یا حتی ویرایش باشند، می‌بایست تراشه را دور انداخت و مجدداً از ابتدا عملیات برنامه‌ریزی یک تراشه جدید را انجام داد. ولی حافظه *RAM* حافظه با دستیابی تصادفی بوده و دسترسی به اطلاعات ذخیره شده در آن بسیار سریعتر از سخت دیسک‌های سنتی می‌باشد. همچنین در این حافظه فقط بخشی از اطلاعات برنامه‌های درحال‌اجرا قرار گرفته و تمام بخش‌های همه برنامه‌ها به حافظه *RAM* آورده نمی‌شود و با خاموش شدن کامپیوتر اطلاعات موجود در آن از بین می‌رود. بدلیل گران‌قیمت بودن تهیه حافظه اصلی با ظرفیت بالا و برخی محدودیت‌های موجود، معمولاً حافظه‌های دیگری نظیر دیسکت، *CD*، *DVD* و ... برای ذخیره داده‌ها و اطلاعات لازمند که به آنها حافظه‌های جانبی گویند.

یک دیسک سخت ۴۰ گیگابایتی، نمونه‌ای از یک ذخیره‌ساز داده کامپیوتری

یک SDRAMیک گیگابایتی، نمونه‌ای از یک ذخیره‌ساز داده کامپیوتری

کوچکترین جزء حافظه که می‌تواند 0 و یا 1 باشد را بیت (*Bit*) گویند. همچنین مجموعه‌ای از هشت بیت را یک بایت (*Byte*) نامند. هر بایت می‌تواند یک حرف (*Character*) را ذخیره نماید. مجموعه‌ای از چند بایت یک کلمه (*Word*) را تشکیل می‌دهند. هر کلمه حداقل شامل دو بایت بوده و دارای شماره‌ای است که به آن آدرس کلمه گویند. تعداد بیت‌های یک کلمه تشکیل طول کلمه را می‌دهند.



شکل۱-۲ بیت‌های یک کلمه از حافظه

مجموعه‌ای از ۱۰۲۴ بایت را یک کیلوبایت (*1KB*)، مجموعه‌ای از ۱۰۲۴ کیلوبایت را یک مگابایت (*1MB*) و به همین ترتیب مجموعه‌ای از ۱۰۲۴ مگابایت را یک گیگابایت (*1GB*) گویند. برای بالا بردن کارایی کامپیوترها معمولاً از تعداد بایت‌های بیشتری در یک کلمه استفاده می‌شود و براین اساس کامپیوتری که هر کلمه آن دو بایتی باشد به یک کامپیوتر ۱۶ بیتی معروف می‌باشد.

۱. **Bit (بیت)**: بیت کوچک ترین واحد حافظه است که فقط دو مقدار صفر (۰ یایک (۱) را می‌توان در آن ذخیره کرد.[۱]

۲. **Nibble (نیبل)**: معمولاً به مجموعهٔ ۴ بیت که کنار هم قرار گرفته باشند یک نیبل گفته می‌شود.

۳. **Byte (بایت)**: هر بایت معمولاً برابر ۸ بیت است، معمولاً حجم هر نویسهٔ غیر یونی‌کد (نویسه یعنی رقم‌ها، حروف یا علامت‌ها) برابر یک بایت است، به عبارتی هر نویسهٔ ساده یک بایت فضا اشغال می‌کند.

۴. **KB (کیلوبایت)**: هر کیلوبایت برابر ۱۰۰۰بایت است، به عبارتی هر کیلوبایت برابر $۱۰^{۳}$ بایت است.

۵. **KiB (کیبی بایت)**: هر کیبی‌بایت برابر ۱۰۲۴ بایت است، به عبارتی هر کیبی‌بایت برابر $۲^{۱۰}$ بایت است.

۶. **MB (مگابایت)**: هر مگابایت برابر ۱۰۰۰ کیلوبایت است، به عبارتی هر مگابایت برابر $۱۰^{۳}$ کیلوبایت است.

مقادیر بایت‌ها					
ده‌دهی		دودویی		نامشخص	
نام (نماد)	استاندارد بین‌المللی	نام (نماد)	مقدار	کاربرد عمومی (درکشورها)	مقدار
کیلوبایت (kB)	$۱۰^{۳}$	کیبی‌بایت (KiB)	$۲^{۱۰}$	کیلوبایت (KB)	$۲^{۱۰}$
مگابایت (MB)	$۱۰^{۶}$	مبی‌بایت (MiB)	$۲^{۲۰}$	مگابایت (MB)	$۲^{۲۰}$
گیگابایت (GB)	$۱۰^{۹}$	گیبی‌بایت (GiB)	$۲^{۳۰}$	گیگابایت (GB)	$۲^{۳۰}$
ترابایت (TB)	$۱۰^{۱۲}$	تبی‌بایت (TiB)	$۲^{۴۰}$	ترابایت (TB)	$۱۰^{۳}*۲^{۳۰}$
پتابایت (PB)	$۱۰^{۱۵}$	پبی‌بایت (PiB)	$۲^{۵۰}$	پتابایت (PB)	$۱۰^{۶}*۲^{۳۰}$
اگزابایت (EB)	$۱۰^{۱۸}$	اگزبی‌بایت (EiB)	$۲^{۶۰}$	اگزابایت (EB)	$۱۰^{۹}*۲^{۳۰}$
زتابایت (ZB)	$۱۰^{۲۱}$	زبی‌بایت (ZiB)	$۲^{۷۰}$	زتابایت (ZB)	$۱۰^{۱۲}*۲^{۳۰}$
یوتابایت (YB)	$۱۰^{۲۴}$	یوبی‌بایت (YiB)	$۲^{۸۰}$	یوتابایت (YB)	$۱۰^{۱۵}*۲^{۳۰}$
سوتابایت (SB)	$۱۰^{۲۷}$	سوبی‌بایت (SiB)	$۲^{۹۰}$	سوتابایت (SB)	$۱۰^{۱۸}*۲^{۳۰}$

۷. **MiB (مبی‌بایت)**: هر مبی‌بایت برابر ۱۰۲۴ کیبی‌بایت است، به عبارتی هر مبی‌بایت برابر $۲^{۲۰}$ کیبی‌بایت است.

۸. **GB (گیگابایت)**: هر گیگابایت برابر ۱۰۲۴ مگابایت است، به عبارتی هر گیگابایت برابر $۱۰^{۳}$ مگابایت است.

۹. **GiB (گیبی‌بایت)**: هر گیبی‌بایت برابر ۱۰۲۴ مبی‌بایت است، به عبارتی هر گیبی‌بایت برابر $۲^{۱۰}$ مبی‌بایت است.

۱۰. **TB (ترابایت)**: هر ترابایت برابر۱۰۲۴ گیگابایت است، به عبارتی هر ترابایت برابر$۱۰^{۳}$ گیگابایت است.

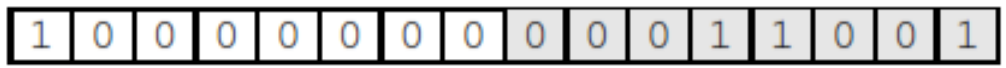
معمولاً کد مربوط به هر نشانه یا علامت با یک کلمه مشخص می‌شود. به منظور هماهنگی بین تمام کامپیوترها، کمیته استاندارد تبادل اطلاعات آمریکا کدهای استاندارد اسکی (*ASCII: American Interchange Information Code for Standard*) را در سال ۱۹۶۷ طراحی نمود. این کد در ابتدا از ۷ بیت تشکیل می‌شد. یعنی هر کاراکتر ۷ بیت را اشغال می‌کرد. با توسعه این سیستم بعدها کد اسکی ۸ بیتی به وجود آمد که با این کد، ۲۵۶ کاراکتر مختلف ($۲^۸$) قابل نمایش است. به عنوان مثال نمونه‌ای از سیستم کدگذاری اسکی بصورت جدول ۱-۱ می‌باشد.

جدول ۱-۱ نمونه‌ای از جدول کدهای اسکی

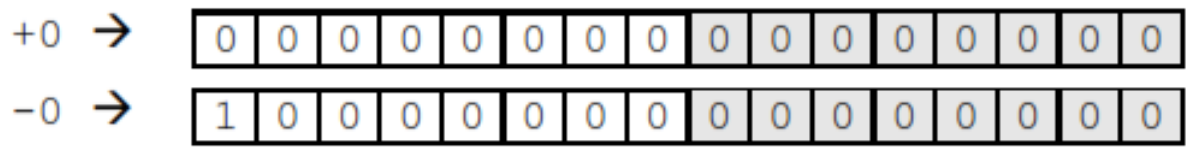
نشانه	کُد برای حافظه یک بایتی	مقدار
Null	00000000	0
+	00101011	43
0	00110000	48
5	00110101	53
9	00111001	57
A	01000001	65

۵-۱-۱ نحوه ذخیره‌سازی داده‌ها در حافظه

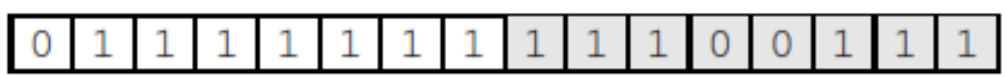
روش‌های مختلفی برای ذخیره‌سازی داده‌ها و اطلاعات در کامپیوتر وجود دارد. این اطلاعات می‌توانند بصورت داده‌های عددی یا رشته‌ای باشند. بعنوان نمونه برای ذخیره‌سازی اعداد صحیح در یک بایت یا کلمه می‌توان از روش *علامت و مقدار* استفاده نمود. بدین صورت که اولین بیت سمت چپ یک کلمه بعنوان بیت علامت و سایر بیت‌های کلمه برای نمایش قدرمطلق عدد موردنظر استفاده می‌شود. در این روش در بیت علامت، برای علامت مثبت رقم 0 و برای علامت منفی رقم 1 درنظر گرفته می‌شود. بنابراین بزرگترین و کوچکترین عدد قابل نمایش در یک کلمه n بیتی برابر $\pm(2^{n-1}-1)$ است. بعنوان مثال عدد ۲۵- به صورت ذیل در یک کلمه ۱۶ بیتی ذخیره می‌شود:



مشکل این روش در این است که برای نمایش صفر مثبت و منفی دو نمایش جداگانه وجود داشته و همچنین برای عمل تفریق باید مدار جداگانه‌ای طراحی گردد.



برای رفع این مشکل می‌توان از روش‌های دیگری نظیر متمم‌دو استفاده نمود. در این روش نمایش مثبت عدد را در مبنای دو درنظرگرفته و تمام صفرها را به یک و تمام یک‌ها را به صفر تبدیل می‌کنند (متمم‌یک). سپس یک واحد به آن اضافه شده و آخرین رقم سمت چپ نتیجه بدست آمده حذف می‌گردد. بعنوان مثال عدد ۲۵- بصورت زیر در یک کلمه ۱۶ بیتی ذخیره می‌شود:



حُسن این روش در این است که برای انجام عمل تفریق نیاز به مدار جداگانه‌ای نیست ولی عملیات تفریق را می‌توان به جمع تبدیل نمود. بعنوان مثالی دیگر عدد $(1001)_2 = 9$ را از عدد $(1101)_2 = 13$ کم می‌کنیم. دنباله عملیات بصورت ذیل خواهد بود:

$$\begin{array}{r} 1101 \\ -\ 1001 \\ \hline \end{array} \Rightarrow \begin{array}{r} 1101 \\ -\ 0110 \\ \hline \end{array} \Rightarrow \begin{array}{r} 1101 \\ +\ 0111 \\ \hline 10100 \end{array} \rightarrow (100)_2 = 4$$

حذف می‌شود

همچنین برای نمایش اعداد اعشاری می‌توان از روش نرمال‌سازی عدد استفاده نمود. بدین ترتیب که ابتدا عدد حاصل نرمال‌سازی شده و بصورت نما و مانتیس درمی‌آید:

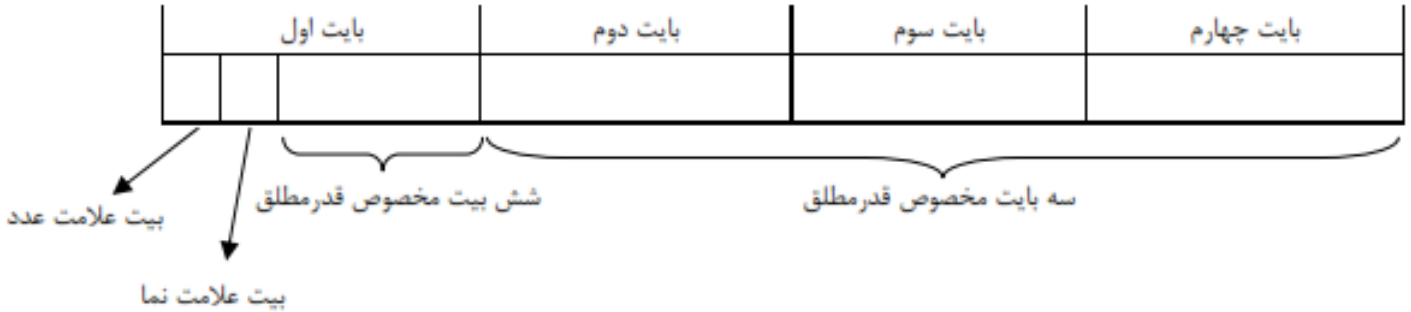
$a \times 10^b$

نما

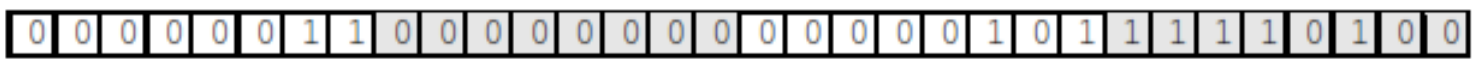
مانتیس

سپس برای ذخیره‌سازی عدد موردنظر در یک کلمه حافظه، در بایت اول دو بیت برای نمایش علامت عدد و نما و شش بیت دیگر برای ذخیره‌کردن قدرمطلق نما استفاده شده و در بایت‌های دیگر کلمه، قدرمطلق مانتیس ذخیره می‌گردد. بعنوان مثال برای ذخیره عدد ۱۵۲/۴ در یک کلمه ۳۲ بیتی ابتدا عدد موردنظر نرمالیزه می‌شود:

$$152.4 \rightarrow +0.1524 \times 10^{+3}$$



در این روش نیز در بیت علامت، برای علامت مثبت رقم 0 و برای علامت منفی رقم 1 درنظر گرفته می‌شود. همچنین در ذخیره‌سازی ارقام هر عدد در مبنای دو خانه‌های حافظه از سمت راست پُر شده و برای خانه‌های خالی رقم 0 جایگزین می‌گردد. بنابراین عدد ۱۵۲/۴ در کلمه ۳۲ بیتی بصورت زیر ذخیره می‌شود:



البته در کامپیوترها علاوه بر پایه دو از پایه هشت و پایه شانزده هم استفاده می‌شود، اگرچه عملیات درونی کامپیوتر در پایه دو انجام گرفته و داده‌های ورودی و اطلاعات خروجی نیز بصورت دهدهی هستند. استفاده از پایه‌های هشت و شانزده باعث آسانی برخی از عملیات در کامپیوترها می‌گردد که برای آشنایی با مباحث موردنظر می‌توان به کتاب‌های مبانی کامپیوتر در این زمینه رجوع کرد.

تمرین۱-۱:

۱) اعداد زیر را در مبنای دو محاسبه نمایید؟

32.61 125 14.7 210

۲) حاصل جمع و تفریق زیر را در مبنای دو بیابید؟

$(11101)_2+(101)_2$ $(10010)_2-(111)_2$

۳) حاصل ضرب و تقسیم زیر را محاسبه نمایید؟

$(1100101)_2/(111)_2$ $(1100)_2\times(10)_2$

۴) عدد ۱۳/۴- به چه صورت در یک کلمه ۱۶ بیتی ذخیره می‌شود؟