برنامه نویسی به زبان ماشین و اسمبلی

برنامه نویسی سیستمی کامپیوترهای شخصی (۸۰۸۶)

(ویژه دانشجویان دوره کاردانی کامپیوتر دانشگاه فنی و حرفه‌ای)

تهیه و تألیف: آزاد نوری

سرشناسه	:	نوری، آزاد، ۱۳۶۲ -
عنوان و نام پدیدآور	:	برنامه‌نویسی به زبان ماشین و اسمبلی : برنامه نویسی سیستمی کامپیوترهای شخصی (۸۰۸۶) (ویژه دانشجویان دوره کاردانی کامپیوتر دانشگاه فنی و حرفه‌ای)/ تهیه و تالیف آزاد نوری.
مشخصات نشر	:	تهران : پندار پارس، ۱۳۹۴.
مشخصات ظاهری	:	۲۰۸ ص.: مصور، جدول.
شابک	:	978-600-6529-95-0
وضعیت فهرست نویسی	:	فیپا
یادداشت	:	کتابنامه.
عنوان دیگر	:	برنامه نویسی سیستمی کامپیوترهای شخصی (۸۰۸۶) (ویژه دانشجویان دوره کاردانی کامپیوتر دانشگاه فنی و حرفه‌ای).
موضوع	:	اسمبلی (زبان برنامه نویسی کامپیوتر)
موضوع	:	زبان‌های برنامه‌نویسی کامپیوتر
رده بندی کنگره	:	QA۷۶/۷۳ /الف۱۳۹۴۵ ۹۳ن
رده بندی دیویی	:	۱۳۶/۰۰۵
شماره کتابشناسی ملی	:	۳۹۳۵۸۵۳

انتشارات پندارپارس

دفتر فروش: انقلاب، ابتدای کارگرجنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶

info@pendarepars.com **www.pendarepars.com**

تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸

نام کتاب	:	**برنامه نویسی به زبان ماشین و اسمبلی**
ناشر	:	**انتشارات پندار پارس**
گرداوری و تالیف	:	**آزاد نوری**
چاپ نخست	:	**مهر ماه ۹۴**
شمارگان	:	**۱۰۰۰ نسخه**
طرح جلد	:	**سارا یعسوبی**
چاپ، صحافی	:	**روز**
قیمت	:	**۱۵۰۰۰ تومان**

شابک : ۰-۹۵-۶۵۲۹-۶۰۰-۹۷۸

فهرست مطالب

پیشگفتار

زبان برنامه نویسی اسمبلی، زبان برنامه نویسی سخت افزار است و بیشترین نزدیکی را به زبان ماشین دارد. با توجه به مشکل بودن کار با زبان ماشین (صفر و یک)، برنامه نویسان از این زبان برای نوشتن برنامه‌های خود در کاربردهای مختلف سخت افزاری و صنعتی استفاده می‌کنند.

زبان ماشین و اسمبلی یکی از دروس مقطع کاردانی و کارشناسی دانشجویان رشته کامپیوتر است که با توجه به تفاوت‌های خاص آن با زبان‌های برنامه نویسی سطح بالا نیاز به دیدگاه و بینش متفاوت برای یادگیری آن وجود دارد. بر عکس زبان‌های سطح بالا نظیر ویژوال بیسک، سی شارپ و غیره که در آن‌ها نیاز به دانش قبلی در مورد ساختار سخت افزاری سیستم و اجزاء داخلی وجود ندارد، در زبان اسمبلی برنامه نویس باید ابتدا یک دید کلی از معماری سیستم و اجزاء داخلی پردازشگر پیدا کند و سپس با استفاده از امکانات داخلی سیستم و ویژگی‌های خاص هر واحد سخت افزاری اقدام به نوشتن برنامه خود کند.

کتاب‌های مرجع زیادی برای این درس وجود دارد که همگی آن‌ها به دلیل مرجع کامل بودن و حجیم بودن برای دانشجویان دوره کاردانی سنگین بوده و مناسب نیستند و علاوه بر این، بسیاری از این کتاب‌ها قدیمی بوده و مدت‌هاست که بروز رسانی نشده‌اند (بیش از ۱۰ الی ۱۵ سال است که مطالب آن‌ها بروز رسانی نشده است) لذا پس از تجربه چندین ساله تدریس این درس در دانشگاه‌های مختلف تصمیم بر این گرفته شد تا کتاب حاضر تهیه و در اختیار دانشجویان دوره کاردانی کامپیوتر دانشگاه فنی و حرفه‌ای قرار گیرد.کتاب به‌گونه‌ای تنظیم شده است که شامل تمامی سرفصل‌های مصوب وزارت علوم در دوره کاردانی بوده و ضمن پرهیز از ذکر مطالب اضافی و حجم بالا، تلاش شده است با زبانی ساده و روان مفاهیم تشریح شده و از مثال‌های مناسب و کافی استفاده شود. ضمنا تلاش شده است که با توجه به تغییرات سخت افزاری و خصوصا تغییرات در نسل‌های جدید پردازنده‌ها از منابع جدیدتر و بروزتر استفاده شود.

این کتاب برای دانشجویان کاردانی و حتی کارشناسی سایر دانشگاه‌ها نیز قابل استفاده است.

از آنجا که زبان اسمبلی وابسته به ویژگی‌های سخت افزار است و از امکانات داخلی پردازشگر، حافظه و سایر قسمت‌ها استفاده می‌کند، لذا برنامه نویس بدون آشنایی با معماری کامپیوتر و اجزاء داخلی آن قادر به برنامه نویسی نخواهد بود. در بخش اول این کتاب مفاهیم

مقدماتی شامل سیستم اعداد و معماری کامپیوتر مورد بررسی قرار گرفته است و سپس در بخش دوم به آموزش مفاهیم برنامه نویسی زبان اسمبلی پرداخته شده است.

در پایان لازم می دانم از زحمات آقای مهندس بهزاد نوری که در مراحل آماده سازی کتاب، یاری رسان بنده بودند تشکر نمایم.

فصل اول

سیستم اعداد

مقدمه

با توجه به اینکه زبان ماشین، زبان صفر و یک است و به عبارت دیگر سخت افزار از مبنای دودویی (Binary) استفاده می‌کند و از طرفی در محاسبات روزمره انسان‌ها، مبنای ده‌دهی (Decimal) مورد استفاده قرار می‌گیرد، لذا برنامه نویس باید با سیستم اعداد و نحوه تبدیل مبناهای مختلف به یکدیگر آشنا باشد.

علاوه بر مبناهای فوق در برنامه نویسی سخت افزار، برای راحت‌تر شدن نمایش اطلاعات و استفاده از تعداد رقم‌های کمتر در نمایش اطلاعات از مبنای شانزده (HexaDecimal) و بعضا مبنای هشت (Octal) استفاده می‌شود. لذا برنامه نویس باید با نحوه نمایش اعداد در مبناهای مختلف آشنا بوده و بتواند با اعداد در مبناهای مختلف کار کند. در این فصل نحوه نمایش اعداد در مبناهای مختلف، محاسبات در هر مبنا و همچنین نحوه تبدیل اعداد از یک مبنا به مبنای دیگر توضیح داده شده است.

سیستم اعداد

نکته ۱: در مبنای r از ارقام 0 تا r-1 استفاده می‌شود.

لذا در مبنای ۲ از ارقام ۰ و۱ استفاده می‌شود و در مبنای ۱۰ از اعداد ۰ تا ۹ استفاده می‌گردد. همچنین در مبنای ۸ از ارقام ۰ تا ۷ و به همین ترتیب در مبنای ۱٦ از ارقام ۰ تا ۹ و کاراکترهایA تاZ (به‌جای ۱۰ تا ۱۵) استفاده می‌شود.

سیستم دودویی(باینری)

همانطور که قبلا گفته شد در این مبنا از ارقام ۰ و ۱ برای نمایش اطلاعات استفاده می‌شود. ارقام از راست به چپ شماره گذاری می‌شوند و هر کدام دارای ارزش مکانی خاصی هستند که برابر ($(\text{شماره بیت})^2$) است.

شماره بیت ها	0	1	2	3	4	5	6	7
عدد در مبنای دودویی	*	*	*	*	*	*	*	*
ارزش عددی	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7
ارزش عددی دهدهی	1	2	4	8	16	32	64	128

شکل ۱-۱ سیستم دودویی (باینری)

علامت * در شکل فوق می‌تواند ۰ یا ۱ باشد. هر بیتی که ۱ باشد معادل ارزش عددی دهدهی خود را ایجاد می‌کند. نهایتا مجموع ارزش‌های ایجاد شده معادل عدد دهدهی نمایش داده شده است.

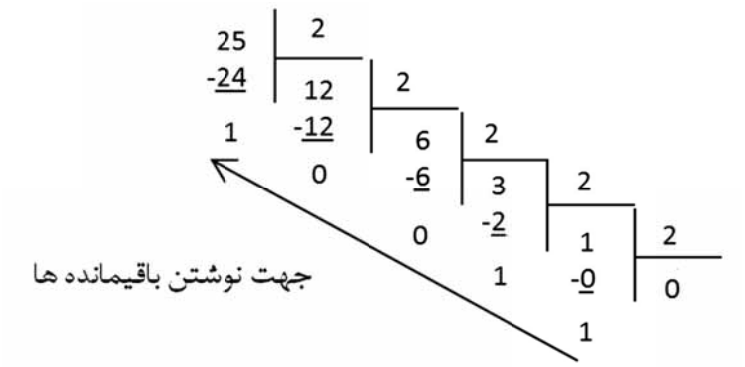
تبدیل مبنای ۱۰ به ۲

برای این کار به نکته زیر دقت کنید.

نکته ۲ : برای تبدیل مبنای ۱۰ به هر مبنای دیگر (۲،۸،۱۶) کافی است عدد دهدهی را متوالیا بر مبنای مورد نظر تقسیم کرد و باقیمانده‌ها را از آخر به اول نوشت.

مطابق نکته فوق برای تبدیل عدد دهدهی به دودویی کافی است آن را متوالیا بر ۲ تقسیم کنیم تا زمانی که خارج قسمت صفر شود، آنگاه باقیمانده‌ها را از آخر به اول بنویسیم.

مثال :عدد ۲۵ معادل کدام عدد دودویی است؟



در نتیجه می‌توان نوشت:

$25_{10} = 11001 = 0001\ 1001_2$

اگر عدد دهدهی دارای قسمت اعشار نیز باشد، کافی است قسمت بعد از ممیز را متوالیا در عدد ۲ ضرب کنیم تا جائیکه دیگر رقمی پشت ممیز نماند. سپس اعداد صحیح به دست آمده را از بالا به پایین می‌نویسیم.

به عنوان نمونه فرض کنید می‌خواهیم عدد ۲۵/۱۲۵ را به مبنای ۲ ببریم:

قسمت صحیح عدد که ۲۵ است را مانند روش مثال قبل به طور جداگانه محاسبه می‌کنیم. (تقسیم متوالی بر ۲) $25 = 11001$

اما برای قسمت اعشار (بعد از ممیز) که ۰/۱۲۵ است داریم:

از بالا به پایین فقط قسمت‌های صحیح را می‌نویسیم

$25.125_{10} = 11001.001_2$

0.125 * 2 = 0.250
0.250 * 2 = 0.500
0.500 * 2 = 1.000

مثال : عدد 25.875 را به معادل باینری آن تبدیل کنید.

قسمت صحیح قبلا محاسبه شده است و فقط قسمت اعشاری را محاسبه می‌کنیم.

0.875 * 2 = 1.750
0.750 * 2 = 1.500
0.500 * 2 = 1.000

0.875=111 => $25.875_{10} = 11001.111_2$

نکته ۳: برای تبدیل مبناهای ۲،۸ و ۱۶ به مبنای ۱۰، ارزش مکانی ایجاد شده هر رقم را محاسبه و نتایج را با هم جمع می‌کنیم.

ارزش مکانی هر رقم در هر مبنایی با رابطه زیر محاسبه می‌شود : رقم * $^{\text{شماره رقم}}$ (مبنا)

تبدیل مبنای ۲ به ۱۰

مطابق نکته ۳، برای این کار ابتدا بیت‌ها را شماره گذاری می‌کنیم. سپس به ازای **بیت‌هایی** که مقدار **۱** دارند، ارزش مکانی (وزنی) هر بیت ($^{\text{شماره بیت}}$۲)، را محاسبه کرده و نتایج را باهم جمع می‌کنیم.

مثال: عدد ۱۰۱۱۰۱/۱۰۱ را به معادل دهدهی تبدیل کنید.

فقط بیت‌هایی را که مقدار ۱ دارند در نظر می‌گیریم

5 4 3 2 1 0 -1 -2 -3
101101.101

$1*2^5+1*2^3+1*2^2+1*2^0+1*2^{-1}+1*2^{-3} = 32+8+4+1+0.5+0.125 = 45.625$

➔ $101101.101_2 = 45.625_{10}$

(مبناهای ۸ و ۱۶ نیز به همین شکل و مطابق نکته ۳ به مبنای ۱۰ تبدیل می‌شوند.در ادامه همین فصل خواهیم دید)

حتی می‌توان خود اعداد دهدهی را نیز به همین روش نمایش داد به عنوان مثال:

123.45 = 1*102+2*101+3*100+4*10-1+5*10-2
=100+20+3 +0.4+0.05

سیستم اعداد هگزادسیمال (شانزده)

در این مبنا برای نمایش اعداد از ارقام ۰ تا ۹ و کاراکترهای A تا F استفاده می‌شود. در جدول۱-۱ می‌توانید ارقام هگزا دسیمال و معادل ده‌دهی و باینری آن‌ها را ببینید. هر رقم هگزا دسیمال معادل ٤ بیت باینری است.

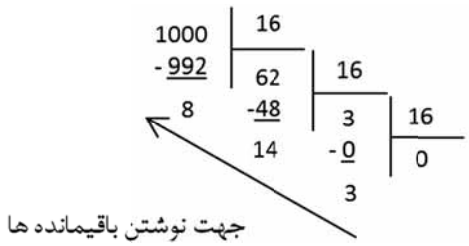
جدول ۱-۱ اعداد هگزادسیمال

ارقام هگزا دسیمال	معادل باینری (۲)	معادل دسیمال (۱۰)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

تبدیل مبنای ۱۰ به ١٦

مطابق نکته ۲ در همین فصل کافی است عدد ده‌دهی را متوالیا بر ١٦ تقسیم کنیم تا جائیکه به صفر برسیم آنگاه باقیمانده‌ها را از آخر به اول بنویسیم. فقط باید دقت کرد که به جای اعداد ۱۰ تا ۱۵ از کاراکتر معادل مطابق جدول ۱-۱ استفاده شود.

مثلا فرض کنید می‌خواهیم عدد ۱۰۰۰ را به مبنای ١٦ ببریم.



لذا داریم : $1000_{10}=3E8_{16}$

تبدیل مبنای ١٦ به ١٠

مطابق نکته شماره ٣ باید ارقام را شماره گذاری و پس از محاسبه ارزش مکانی ارقام، آن‌ها را با هم جمع کرد.

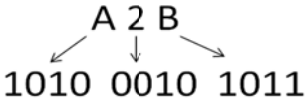
مثال :عدد 3E8 را به مبنای ١٠ ببرید.

$$3E8 = 3*16^2+E*16^1+8*16^0 = 3*256+14*16+8*1 = 1000$$

تبدیل مبنای ١٦ به ٢

هر رقم هگزادسیمال معادل ٤ رقم(بیت) باینری است لذا کافی است هر رقم هگز را در ٤ بیت نمایش داده و به ترتیب پشت سرهم نوشت.

مثال : عدد A2B را به مبنای دو (باینری) تبدیل کنید.

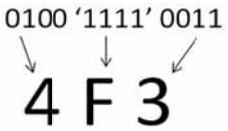


➔ A2B=1010 0010 1011

تبدیل مبنای ٢ به ١٦

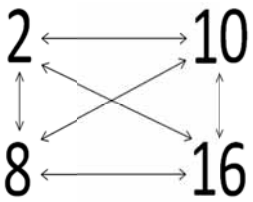
برای این کار کافی است از راست به چپ ٤ رقم ٤ رقم بیت‌ها را جدا نموده و به ازای هر ٤ رقم (بیت) معادل هگز آن را نوشت.

مثال: عدد ۰۱۰۰۱۱۱۱۰۰۱۱ به مبنای ١٦ ببرید.



نکته ٤: با تغییر مبنا فقط نحوه نمایش هر عدد تغییر می‌یابد اما مقدار ارزش آن تغییر نخواهد کرد

چنانچه در شکل روبرو می‌بینید برای تبدیل مبناها به یکدیگر می‌توان مستقیم یا غیر مستقیم اقدام کرد مثلا تبدیل ١٠ به ١٦ هم مستقیما انجام پذیر است و هم از طریق تبدیل١٠ به ٢ و سپس ٢ به ١٦ یا تبدیل ١٠ به ٨ و سپس ٨ به ١٦.



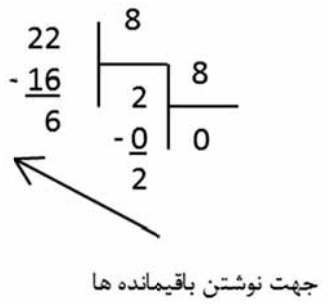
مبنای ٨ (اوکتال)

در این مبنا از اعداد ٠ تا ٧ برای نمایش اعداد استفاده می‌شود.هر رقم اوکتال معادل ٣ بیت (رقم) باینری است.

تبدیل مبنای ۱۰ به ۸

مطابق نکته ۲ کافی است عدد دهدهی را متوالیا بر ۸ تقسیم نموده و باقیماندهها را از آخر به اول نوشت.

مثال:عدد ۲۲ را به مبنای ۸ تبدیل کنید.



درنتیجه داریم: $22_{10}=26_8$

تبدیل مبنای ۸ به ۱۰

مطابق نکته شماره ۳ باید ارقام را شماره گذاری و پس از محاسبه ارزش مکانی ارقام، آنها را با هم جمع کرد.

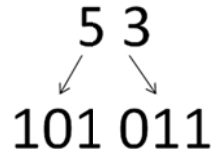
مثال:عدد ۲۶ در مبنای ۸ را به مبنای ۱۰ ببرید.

$2*8^1+6*8^0=16+6=22$

تبدیل مبنای ۸ به ۲

هر رقم اوکتال (هشت) معادل ۳ بیت باینری است. لذا کافی است به ازای هر رقم معادل باینری آن را در ۳ بیت نوشت.

مثال: عدد ۵۳ اوکتال را به باینری تبدیل کنید.

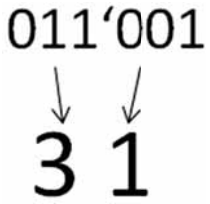


$\mathbf{53_o=101\ 011_B}$

تبدیل مبنای ۲ به ۸

برای اینکار کافی است عدد مورد نظر را از راست به چپ ۳ رقم ۳ رقم جدا نموده و به ازای هر ۳ رقم، معادل اوکتال آن را بنویسید.

مثال: عدد ۱۱۰۰۱ را به معادل مبنای ۸ آن تبدیل کنید.



پس می‌توان نوشت:

$011001_2=31_8$ یا به عبارت دیگر $011001_B=31_O$

محاسبات در مبنای دودویی و هگزا دسیمال

عملیات جمع و تفریق در مبنای ۲ و ۱٦ از همان قوانین مبنای ۱۰ تبعیت می‌کند. با این تفاوت که به جای عدد ۱۰ مبنای مورد نظر (۲ یا ۱٦) را مد نظر قرار می‌دهیم. مثلا در عملیات مبنای ۱۰ (ده‌دهی) داریم:

عملیات جمع: اگر مجموع ارقام یک ستون بزرگ‌تر یا مساوی ۱۰ باشد، ۱۰ واحد را از آن کم نموده و باقیمانده را می‌نویسیم در ضمن ۱ واحد به ارقام ستون سمت چپ اضافه می‌کنیم.

عملیات تفریق: اگر مقدار رقم اول (مفروق منه) کمتر از مقدار رقم دوم (مفروق) باشد، از ستون سمت چپ آن ۱ واحد کم نموده و به ازای آن ۱۰ واحد به ستون سمت راست اضافه می‌کنیم و سپس عملیات تفریق را انجام می‌دهیم.

اکنون همین قوانین برای سایر مبناها نیز صادق است فقط کافی است به جای عدد ۱۰، عدد مبنای مورد نظر (۲ یا ۱٦) را قرار دهیم.

پس در مبنای ۲ داریم:

عملیات جمع: اگر مجموع ارقام یک ستون بزرگ‌تر یا مساوی ۲ باشد، ۲ واحد را از آن کم نموده و باقیمانده را می‌نویسیم در ضمن ۱ واحد به ارقام ستون سمت چپ اضافه می‌کنیم.

عملیات تفریق: اگر مقدار رقم اول (مفروق منه) کمتر از مقدار رقم دوم (مفروق) باشد، از ستون سمت چپ آن ۱ واحد کم نموده و به ازای آن ۱۰ واحد به ستون سمت راست اضافه می‌کنیم و سپس عملیات تفریق را انجام می‌دهیم.

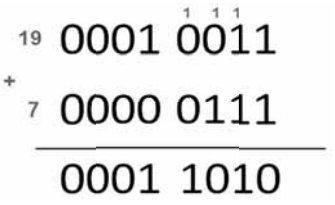
همچنین می‌توان از جدول ۱-۲ برای عملیات جمع استفاده کرد.

جدول ۱-۲ عملیات جمع در مبنای باینری

	SUM(مجموع)	CARRY(رقم نقلی)
0+0	0	0
0+1	1	0
1+0	1	0
1+1	0	1
1+1+1	1	1

مثال: عملیات جمع زیر را در سیستم باینری انجام دهید.

19+7

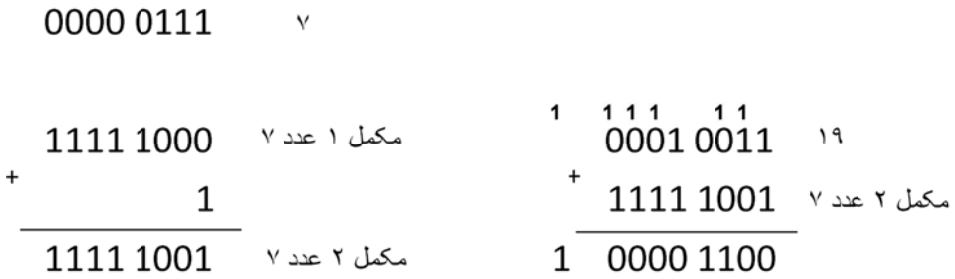


سخت افزار برای انجام تفریق دودویی از همان مدار جمع کننده استفاده می‌کند یعنی با استفاده از رابطه زیر ابتدا عملیات تفریق را به جمع تبدیل کرده و سپس محاسبه را انجام می‌دهد.

A-B=A+(B مکمل دو عدد)

از طرفی داریم: مکمل دو = مکمل یک + ۱ و همچنین مکمل یک از تبدیل ۰ به ۱ و ۱ به ۰ به دست می‌آید.

به عنوان مثال فرض کنید می‌خواهیم عملیات ۷-۱۹ را در سیستم باینری انجام دهیم.



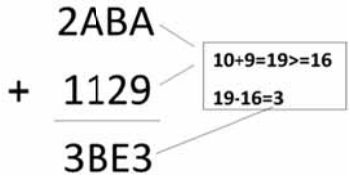
مطابق نکات گفته شده در ابتدای این بخش در مبنای ۱۶ داریم:

عملیات جمع: اگر مجموع ارقام یک ستون بزرگ‌تر یا مساوی ۱۶ باشد، ۱۶ واحد را از آن کم نموده و باقیمانده را می‌نویسیم در ضمن ۱ واحد به ارقام ستون سمت چپ اضافه می‌کنیم.

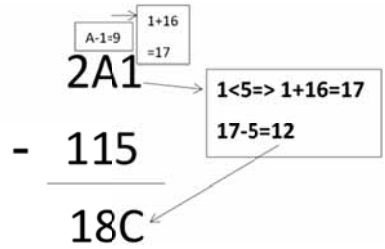
عملیات تفریق: اگر مقدار رقم اول (مفروق منه) کمتر از مقدار رقم دوم (مفروق) باشد، از ستون سمت چپ آن ۱ واحد کم نموده و به ازای آن ۱۶ واحد به ستون سمت راست اضافه می‌کنیم و سپس عملیات تفریق را انجام می‌دهیم.

در مثال‌های زیر جمع و تفریق در مبنای هگزا دسیمال را می‌بینید.

جمع دو عدد

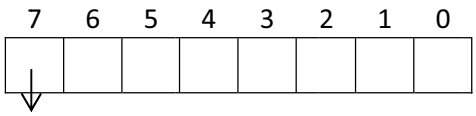


تفریق دو عدد



نحوه ذخیره سازی اعداد در حافظه

اعداد در حافظه به دو شکل علامت دار و بدون علامت قابل ذخیره سازی است. در حالت علامت‌دار آخرین بیت به عنوان بیت علامت[۱] در نظر گرفته می‌شود که اگر ۰ باشد به معنای عدد مثبت و اگر ۱ باشد به معنای عدد منفی است. در شکل زیر ساختار یک بایت حافظه را در حالت علامت دار می‌بینید.



بیت علامت

در حالت علامت دار می‌توان در یک بایت از عدد ۱۲۸- تا ۱۲۷+ را ذخیره کرد اما اگر اعداد بدون علامت ذخیره شوند می‌توان در یک بایت حافظه، از ۰ تا ۲۵۵ را ذخیره نمود.

---

[۱] Sign Bit

نکته۵ : با n بیت می‌توان 2^n عدد یا حالت منحصر بفرد (غیر تکراری) را نمایش داد. این اعداد از 0 تا 2^n-1 است. به عبارت دیگر بزرگ‌ترین عددی که در n بیت قابل نمایش است، 2^n-1 است. (در حالت بدون علامت)

به عنوان مثال با ۲ بیت می‌توان ٤ عدد مختلف (۰ تا $۲^۲$-۱) را نمایش داد: ۰۰،۰۱،۱۰،۱۱

با توجه به نکته فوق، در حالت بدون علامت، بزرگ‌ترین عددی که می‌توان در یک بایت (هشت بیت) نمایش داد معادل $۲^۸$-۱=۲۵۵ است و این عدد زمانی حاصل می‌شود که تمام بیت‌ها ۱ باشند (۱۱۱۱ ۱۱۱۱) و کوچک‌ترین عدد قابل نمایش ۰ است که مربوط به زمانی است که همه بیت‌ها ۰ باشند (۰۰۰۰ ۰۰۰۰). به عبارت دیگر از 00 تا FF هگز قابل نمایش است.

نکته ٦: اعداد منفی به شکل مکمل ۲ در حافظه ذخیره می‌شوند.

به عنوان مثال عدد ۲۰- اینگونه در حافظه ذخیره می‌شود.

عدد ۲۰ : ۰۰۰۱ ۰۱۰۰ اکنون از این عدد مکمل دو گرفته می‌شود.

مکمل دو عدد ۲۰ : $\underline{1}$۱۱۱۰ ۱۱۰۰ نشان دهنده منفی بودن عدد

مثال: عدد علامت دار ۱۰۱۰ ۱۱۱۱ که در حافظه ذخیره شده است معادل کدام عدد ده‌دهی است؟

روش اول: با توجه به اینکه بیت علامت ۱ است پس می‌توان نتیجه گرفت که عدد منفی است و به شکل مکمل ۲ در حافظه ذخیره شده است.اکنون کافی است مجددا از آن مکمل ۲ بگیریم چون داریم: **مکمل دو (مکمل دوعدد x) برابر است با خود عدد x.**

پس از گرفتن مکمل دو به دست می‌آید: ۰۱۱۰ ۰۰۰۰ و پس از تبدیل به مبنای ۱۰ عدد ٦- به دست می‌آید.

روش دوم:

با توجه به بیت علامت و منفی بودن عدد، به جای گرفتن مکمل دو کافی است ابتدا معادل ده‌دهی عدد فوق (۱۰۱۰ ۱۱۱۱) را به دست آوریم و سپس آن را منهای $۲^۸$ یعنی ۲٥٦ کنیم.

1111 1010 = 128+64+32+16+8+2=250

250-256=-6

همانگونه که می‌بینید عدد ٦- به دست آمد که معادل نتیجه روش قبلی است.

مثال: عدد علامت دار ۰۱۰۱ ۰۰۰۰ معادل کدام عدد دهدهی است؟

با توجه به بیت علامت این عدد مثبت است، پس مستقیما به دهدهی تبدیل می‌شود که عدد ۵+ به دست می‌آید.

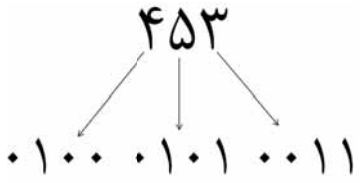
در جدول ۱-۳ محدوده اعداد قابل نمایش در یک و دو بایت در مبنای دهدهی و هگزادسیمال قابل مشاهده است.

جدول ۱-۳ محدوده اعداد قابل نمایش در حافظه

	دو بایت		یک بایت	
حالت ذخیره سازی	علامت دار	بدون علامت	علامت دار	بدون علامت
محدوده اعداد قابل نمایش	۳۲۷۶۸- تا ۳۲۷۶۷+	۰ تا ۶۵۵۳۵	۱۲۸- تا ۱۲۷+	۰ تا ۲۵۵
محدوده اعداد قابل نمایش(هگز)	FFFF تا 0000		FF تا 00	

اعداد دهدهی BCD[1]

برای سهولت انتقال اطلاعات از ورودی به حافظه و از حافظه به خروجی و همچنین سهولت در نمایش اعداد دهدهی، از روش BCD استفاده می‌شود. در این روش هر رقم در ٤ بیت نمایش داده می‌شود. به عنوان مثال برای تبدیل عدد BCD ٤٥٣ کافی است ارقام را از هم جدا و هر رقم را در ٤ بیت باینری نمایش دهیم.



برای تبدیل کد باینری به دهدهی BCD کافی است از راست به چپ ٤ بیت ٤ بیت جدا نموده و معادل آن را بنویسیم. عدد BCD به دو صورت BCD فشرده و غیر فشرده به کار می‌رود. در حالت BCD فشرده در هر بایت می‌توان دو رقم BCD را ذخیره کرد در حالیکه در BCD غیر فشرده در هر بایت یک رقم ذخیره می‌شود (در نیمه سمت راست) و در نیمه سمت چپ صفر قرارمی‌گیرد.

[1] **B**inary **C**oded **D**ecimal

فصل دوم

معماری کامپیوتر