...و این فرشتگان زمینی
کرونا را شکست می‌دهیم
با سپاس از حافظان سلامت کشور
[پزشکان،پرستاران وتمامی کادر درمانی]
aparat.com/goldiran

آموزشکده فنی دختران
سیرجان (کوثر)

زبان ماشین و اسمبلی

جلسه اول(تکمیلی)

فصل اول:

سیستم اعداد

مدرس:

موسویان

نمایش اعداد در مبناهای مختلف

مبنای 2 (Binary) ، مبنای 8 (Octal) ، مبنای 10 (Decimal) ، مبنای 16 (Hexadecimal)

اعداد: صحیح، اعشاری

اعداد صحیح: علامتدار(signed) ، بدون علامت (unsigned)

اعداد اعشاری: ممیزثابت (Pixel Point) ، ممیز شناور (Floating Point)

تبدیل اعداد از مبنای 10 به مبنای 2 ، 8 ،16 : تقسیمات متوالی به مبنای موردنظر

مبنا	رقمهای مورداستفاده	علامت مشخصه
2	0 , 1	b , B
8	0 - 7	o , O
10	0 - 9	d , D
16	0 - 9 , A - F	h , H

250 / 2 = 125 , 0
125 / 2 = 62 , 0
62 / 2 = 31 , 0
31 / 2 = 15 , 1
15 / 2 = 7 , 1
7 / 2 = 3 , 1
3 / 2 = 1 , 1
1 / 2 = 0, 1

$(250)_{10}=(11111000)_2$

250 / 8 = 31 , 2
31 / 8 = 3 , 7
3 / 8 = 0 , 3

$(250)_{10}=(372)_8$

1500 / 8 = 187 , 4
187 / 8 = 23 , 3
23 / 8 = 2 , 7
2 / 8 = 0 , 2

$(1500)_{10}=(2734)_8$

250 / 16 = 15 , 10
15 / 16 = 0 , 15

$(250)_{10}=(FA)_{16}$

1500 / 16 = 93 , 12
93 / 16 = 5 , 13
5 / 16 = 0 , 5

$(1500)_{10}=(5DC)_{16}$

تبدیل اعداد از مبنای ۲ به سایر مبناها

- **مبنای 2 به 10** : ضرب در توان های دو

مثال: $(11100110)_2 = (\ ?\ )_{10}$

$1*2^7 + 1*2^6 + 1*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0 = 128 + 64 + 32 + 0 + 0 + 4 + 2 + 0 = 230$

- **مبنای 2 به 8** : از سمت راست، ۳ رقم ۳ رقم جدا می کنیم و معادل دهدهی هر یک را می نویسیم.

مثال: $(11011001)_2 = (\ ?\ )_8$

011 011 001 = 331

- **مبنای 2 به 16** : از سمت راست، ۴ رقم ۴ رقم جدا می کنیم ومعادل دهدهی هریک را می نویسیم.

مثال: $(11011100)_2 = (\ ?\ )_{16}$

1101 1100 = DC

- **مبنای 8 و 16 به 10** : ضرب در توان های ۸ یا ۱۶

$(275)_8 = (\ ?\ )_{10}$ ⇒ $(5*8^0)+(7*8^1)+(2*8^2)=5+56+128=\mathbf{189}$

$(A2D5)_{16} = (\ ?\ )_{10}$ ⇒ $(5*16^0)+(13*16^1)+(2*16^2)+(10*16^3)=80+208+512+40960=\mathbf{41760}$

- **مبنای 8 به مبنای 2** : هر رقم را جداگانه به مبنای ۲ برده و بصورت ۳ رقم صفر و یک می نویسیم و در صورت لزوم صفر اضافه می کنیم تا سه رقمی شود.

$(275)_8 = (\ ?\ )_2 = (010\ \ 111\ \ 101)_2$

- **مبنای 16 به مبنای 2** : هر رقم را جداگانه به مبنای ۲ برده و بصورت ۴ رقم صفر و یک می نویسیم و در صورت لزوم صفر اضافه می کنیم تا چهار رقمی شود.

$(A2D5)_{16} = (\ ?\ )_2 = (1010\ \ 0010\ \ 1101\ \ 0101)_2$

اعداد صحیح

- **اعداد صحیح بدون علامت**

نحوه نمایش اعداد صحیح بدون علامت در داخل سیستم های کامپیوتری: این نوع اعداد به روش گفته شده بالا به مبنای ۲ برده شده و ذخیره می گردند.

- **اعداد صحیح علامت دار**

نحوه نمایش اعداد صحیح علامت دار در داخل سیستم های کامپیوتری:

۱. **روش بیت علامت (sign bit)** : سمت چپ ترین بیت را به علامت عدد اختصاص می دهیم، صفر: مثبت ؛ یک: منفی

۲. **روش مکمل ۱ (1's complement):**

اعداد مثبت : مشابه روش بیت علامت

اعداد منفی : حالت مثبت عدد را می نویسیم و تمامی بیت ها را مکمل می کنیم.

۳. **روش مکمل ۲ (2 's complement):**

اعداد مثبت : مشابه روش بیت علامت

اعداد منفی : حالت مثبت عدد را می نویسیم و از سمت راست تا رسیدن به اولین یک بدون تغییر و بقیه بیت ها را مکمل می کنیم.

اعداد اعشاری

در اغلب سیستم های سخت افزاری و نرم افزاری به دلیل مزایای زیر از روش مکمل ۲ برای اعداد علامتدار استفاده می شود.

۱. وجود نمایش منحصر بفرد برای عدد صفر

۲. امکان استفاده از عمل جمع بجای عمل تفریق

(این مزیت باعث می شود نیازی به سخت افزار جداگانه تفریق کننده نباشد و ازهمان جمع کننده استفاده شود که باعث صرفه جویی در هزینه ساخت cpu و همچنین کاهش حجم cpu می شود)

اعداد اعشاری

- **ممیز ثابت :** اعداد اعشاری که در آنها تعداد ارقام اعشار ثابت است.

نحوه تبدیل به مبنای ۲ :

روش اول: کل عدد را بدون در نظر گرفتن نقطه اعشار به عنوان یک عدد صحیح در نظر می گیریم و به مبنای ۲ تبدیل می کنیم. سپس تعداد ارقام اعشار را هم به صورت مبنای دو ذخیره می کنیم.

مثال: $(17/345)_{10} = (\ ?\ )_2$

$17345 = (0100\ 0011\ 1100\ 0001)_2$

تعداد ارقام اعشار $3 = (0000\ 0011)_2$

0	1	0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	1	1

اعداد اعشاری

روش دوم: سمت چپ نقطه اعشار به عنوان یک عدد صحیح و سمت راست آن هم به عنوان یک عدد صحیح ذخیره گردد.

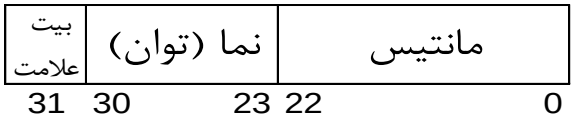
$17 = (0000\ 0000\ 0001\ 0001)_2$

$345 = (0000\ 0001\ 0101\ 1001)_2$

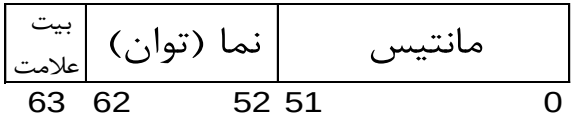
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	1	0	1	1	0	0	1

- **ممیز شناور:** اعداد اعشاری که در آنها ارقام اعشار ثابت نیستند و اعداد اعشاری خیلی بزرگ و خیلی کوچک (نزدیک به صفر) را هم شامل می شود.
  - ابتدا مانند اعداد ممیز ثابت به مبنای دو تبدیل می کنیم.
  - عدد بدست آمده را به صورت نماد علمی با مانتیس نرمال می نویسیم.$(0.1 < m < 1)$
  - سپس مانتیس و نما را به صورت جداگانه ذخیره می کنیم.
  - برای اعداد منفی از روش بیت علامت استفاده می کنیم و روشهای مکمل ۱ و ۲ استفاده نمی شود.

استاندارد ۳۲ بیتی اعداد ممیز شناور(float)



استاندارد ۶۴ بیتی اعداد ممیز شناور(double)



/

نمایش اعداد بدون علامت به صورت BCD

اگر یک عدد بدون علامت مبنای ۱۰ را مانند یک عدد مبنای ۱۶ در نظر بگیریم و به مبنای ۲ تبدیل کنیم، کد حاصل را BCD(Binary Coded Decimal) می گوییم. یعنی هر رقم عدد مبنای ۱۰ را به ۴ رقم باینری می نویسیم.

عدد بدون علامت مبنای ۱۰	binary	BCD
25	0001 1001	0010 0101
250	1111 1010	0010 0101 0000
1720	0110 1011 1000	0001 0111 0010 0000

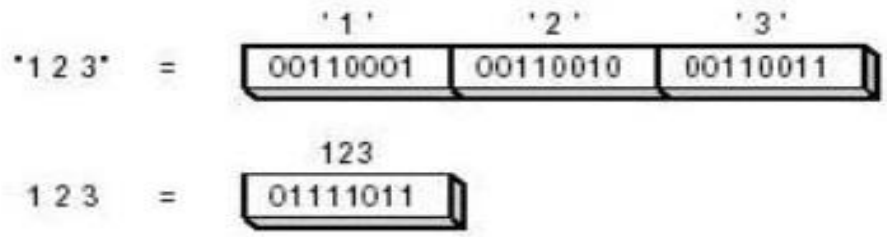
کد گذاری ASCII

کد ASCII:

کد گذاری ASCII (American Standard Code for Information Interchange) به حروف، ارقام، علائم و کاراکترهای مختلف یک عدد باینری ۷ بیتی نسبت می دهد و هشتمین بیت را ۰ درنظر می گیرد. به این صورت هر کاراکتر یک بایت را اشغال می کند.

روشن است که این روش برای نمایش اعداد مناسب نیست، چون در فرمت باینری یک بایت اعداد ۰ تا ۲۵۵ را نمایش می دهد، اما با کد ASCII یک بایت تنها برای نمایش یک رقم کافی است. به همین دلیل کلا این روش برای نمایش متن در حافظه استفاده می شود.

مثال: نمایش عدد ۱۲۳ با دو فرمت ASCIIو باینری



کد ASCII

نوع توسعه یافته این سیستم شامل 8 بیت برای هر کاراکتر است و 256 حالت مختلف را شامل می شود. کدهای 0 تا 127 برای کاراکترهای استاندارد، کدهای کنترلی و ارتباطی و مقادیر 128 تا 255 برای نمایش سمبل های گرافیکی و حروف یونانی هستند.

مثال: رشته "ABC123" به صورت 41h 42h 43h 30h 31h 32h نشان داده می شود.

یک کدگذاری کامل تر که جای ASCII را دارد می گیرد Unicode است. تفاوت کلیدی بین این دو نوع کدگذاری در این است که ASCII یک بایت را برای کدکردن یک کاراکتر استفاده می کند در حالیکه Unicode برای هر کاراکتر دو بایت را درنظر می گیرد. بنابراین کاراکترهای بیشتری را می تواند نمایش دهد که این برای نمایش کاراکترهای کلیه زبان های دنیا کاربردی است.

مثال: کدگذاری ASCII کد 41h یا 65 را به کاراکتر A می دهد. کدگذاری Unicode کد 0041h هگز را می دهد.

کد ASCII

نکته۱: تفاوت یک حرف بزرگ با یک حرف کوچک تنها در بیت شماره ۵ است؛ این بیت در حروف بزرگ 0 و در حروف کوچک 1 است. (" m"= 01101101 و "M"= 01001101)

نکته۲: ارقام 0 تا 9 کدهای 30h تا 39h را دارا می باشند.

نکته۳: کاراکترهای قابل چاپ بین 20h تا 7Eh است.

نکته۴: کاراکترهای 0 تا 1Fh و 7Fh کاراکترهای کنترلی نام دارند که قابل رویت نمی باشند.

نکته۵: کاراکتر ESC با کد 1Bh همراه با کاراکترهای دیگر اغلب برای یک عمل خاص به دستگاه های جانبی ارسال می شود.

نکته۶: کدهای 41 تا 5Ah کاراکترهای A تا Z و کدهای 61 تا 7Ah کاراکترهای a تا z هستند.

نکته۷: کاراکتر CRو LFبا کدهای 0Dh و 0Ah به ترتیب باعث حرکت مکان نما به شروع خط جاری و خط بعد می شود.

محاسبات در سیستم دودویی و هگزا دسیمال

عملیات جمع و تفریق در مبنای ۲ و ۱۶ از همان قوانین مبنای ۱۰ تبعیت می کند، با این تفاوت که به جای عدد۱۰ مبنای موردنظر (۲یا۱۶) را مد نظر می گیریم.

در مبنای دو داریم :

عملیات جمع: اگر مجموع ارقام یک ستون بزرگ تر یا مساوی ۲ باشد، ۲ واحد را از آن کم نموده و باقیمانده را می نویسیم در ضمن ۱ واحد به ارقام ستون سمت چپ اضافه می کنیم.

	مجموع SUM	رقم نقلی Carry
0+0	0	0
0+1	1	0
1+0	1	0
1+1	0	1
1+1+1	1	1

عملیات تفریق: اگر مقدار رقم اول کمتر از مقدار دوم باشد، از ستون سمت چپ آن ۱ واحد کم نموده و به ازای آن ۱۰ واحد به ستون سمت راست اضافه می کنیم و سپس عملیات تفریق را انجام می دهیم.

محاسبات در سیستم دودویی و هگزا دسیمال

مثال :

```
        1 1 1
  0 0 0 1 0 0 1 1   19+
  0 0 0 0 0 1 1 1    7
-----------------
  0 0 0 1 1 0 1 0
```

سخت افزار برای انجام عمل تفریق از همان مدار جمع کننده استفاده می کند یعنی با استفاده از رابطه زیر ابتدا عملیات تفریق را به جمع تبدیل می کند و سپس محاسبه را انجام می دهد.

A - B = A + (B عدد مکمل دو)

از طرفی داریم : مکمل دو = مکمل یک + ۱

برای مثال فرض کنید: عملیات **۷ – ۱۹**

```
1 1 1 1       1 1
  0 0 0 1 0 0 1 1
  1 1 1 1 1 0 0 1
-----------------
1 0 0 0 0 1 1 0 0
```

```
0000 0111    7
1111 1000    مکمل ۱ عدد ۷
+       1
-----------------
1111 1001    مکمل ۲ عدد ۷
```

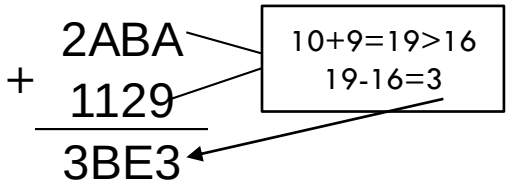
محاسبات در سیستم دودویی و هگزا دسیمال

مطابق نکات گفته شده در مبنای ۱۶ داریم:

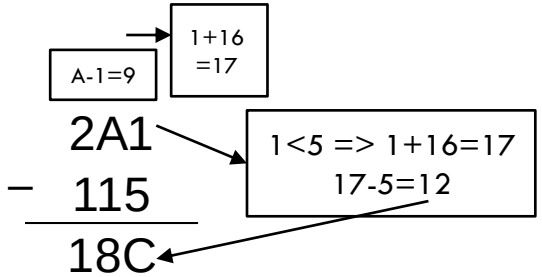
عملیات جمع: اگر مجموع ارقام یک ستون بزرگ تر یا مساوی ۱۶ باشد، ۱۶ واحد را از آن کم نموده و باقیمانده را می نویسیم در ضمن ۱ واحد به ارقام ستون سمت چپ اضافه می کنیم.

عملیات تفریق: اگر مقدار رقم اول کمتر از مقدار دوم باشد، از ستون سمت چپ آن ۱ واحد کم نموده و به ازای آن ۱۶ واحد به ستون سمت راست اضافه می کنیم و سپس عملیات تفریق را انجام می دهیم.

جمع دو عدد :



تفریق دو عدد:



پایان جلسه اول

موفق باشید