دانشگاه بین المللی امام خمینی قزوین

دانشکدهٔ فنی و مهندسی

مبانی کامپیوتر و برنامه سازی

(دومین ترم کرونا!)

فصل دوم: سیستم اعداد و داده

نستوه طاهری جوان

nastooh@aut.ac.ir

فهرست مطالب

سیستم اعداد

✓ سیستم نمایش اعداد

1. با ارزش مکانی ارقام
   - مانند سیستم متداول دهدهی

352= 3*100 + 5*10 + 2*1

1. بدون ارزش مکانی ارقام
   - مانند سیستم اعداد رومی

I, II, III, IV, V, VI, VII, VIII, IX, X, XI, …

سیستم اعداد

- ✓ سیستم دهدهی (Decimal)
  - استفاده از ۱۰ نشانه برای اعداد

$$a_m a_{m-1} a_{m-2} \ldots a_1 a_0 =$$
$$a_m * 10^m + a_{m-1} * 10^{m-1} + a_{m-2} * 10^{m-2} + \ldots + a_1 * 10^1 + a_0 * 10^0$$

مثال: عدد 8425

$$(8 \times 10^3) + (4 \times 10^2) + (2 \times 10^1) + (5 \times 10^0) = 8000 + 400 + 2 + 5 = 8425$$

nastooh@aut.ac.ir

سیستم اعداد

- ✓ سیستم دودویی (Binary)
  - استفاده از تنها دو نشانه برای ارقام (صفر و یک)
  - ساده ترین سیستم اعداد

- ✓ استفاده از سیستم دهدهی در کامپیوتر، نیازمند مدارهای پیچیده تری است.
  - محاسبات در کامپیوتر در مبنای دو، صورت میگیرد، به دلیل سادگی و رسیدن به سرعت بالاتر...

مثال: آیا می توانید به صورت ذهنی محاسبه کنید که عدد 1011 در مبنای ۲، معادل چه عددی در مبنای ۱۰ است؟

سیستم اعداد

- سیستم دودویی (Binary)
  - بسط عدد:

$$a_m a_{m-1} \ldots a_2 a_1 a_0 =$$
$$(a_m \times 2^m) + (a_{m-1} \times 2^{m-1}) + \ldots + \ldots (a_2 \times 2^2) + (a_1 \times 2^1) + (a_0 \times 2^0)$$

  - مثال: بسط عدد $(101)_2$ مبنای ده: $(1\times 2^2) + (0\times 2^1) + (1\times 2^0)$
  - مثال: بسط عدد $(111011)_2$ در مبنای ده:

$$(1\times 2^5) + (1\times 2^4) + (1\times 2^3) + (0\times 2^2) + (1\times 2^1) + (1\times 2^0) =$$
$$32 + 16 + 8 + 0 + 2 + 1 = 59$$

سیستم اعداد

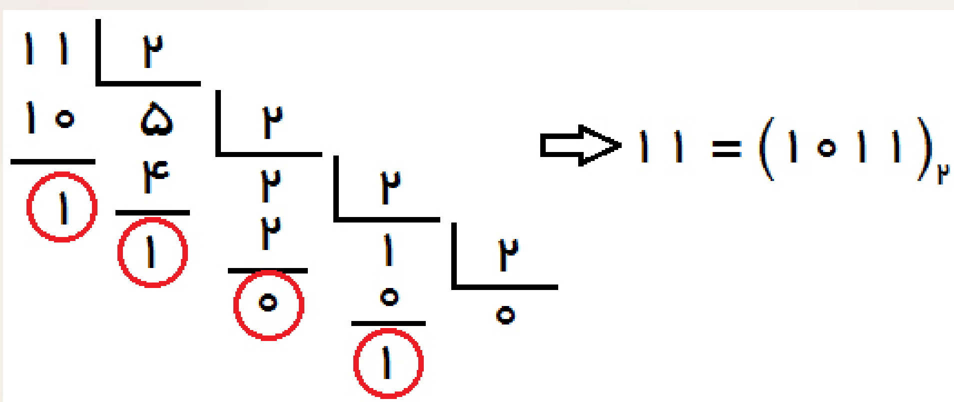
- سیستم دودویی (Binary)
  - جمع بندی:
    - کامپیوتر در مبنای دو کار میکند!
    - انسان با مبنای ده کار میکند.
      - محاسبات در مبنای دو و کار با صفر و یک های متوالی برای انسان طاقت فرساست.
    - این مبناها باید به یکدیگر تبدیل شوند!
    - وظیفهٔ تبدیل مبناها عموماً برعهدهٔ خود کامپیوتر است.
    - ورودی ها معمولا در مبنای ۱۰ از انسان گرفته می شوند و خروجی ها نیز در مبنای ۱۰ نمایش داده می شوند، اما کلیهٔ محاسبات در مبنای ۲ صورت میگیرند.
    - مبناهای ۸ و ۱۶ نیز در علوم کامپیوتر کاربردهای خاصی دارند.

nastooh@aut.ac.ir

سیستم اعداد

✓ تبدیل مبنا

- دهدهی به دودویی
  - استفاده از تقسیم های متوالی
  - تقسیم عدد به دو و محاسبهٔ باقیمانده و خارج قسمت
  - تقسیم خارج قسمت بر دو تا زمان صفر شدن خارج قسمت
  - نوشتن باقیمانده ها، <u>**از انتها به ابتدا**</u>، کنار هم
- مثال: تبدیل $(11)_{10}$ به مبنای دو:



سیستم اعداد

- تبدیل مبنا
  - دودویی به دهدهی
    - بسط عدد دودوی در مبنای ده

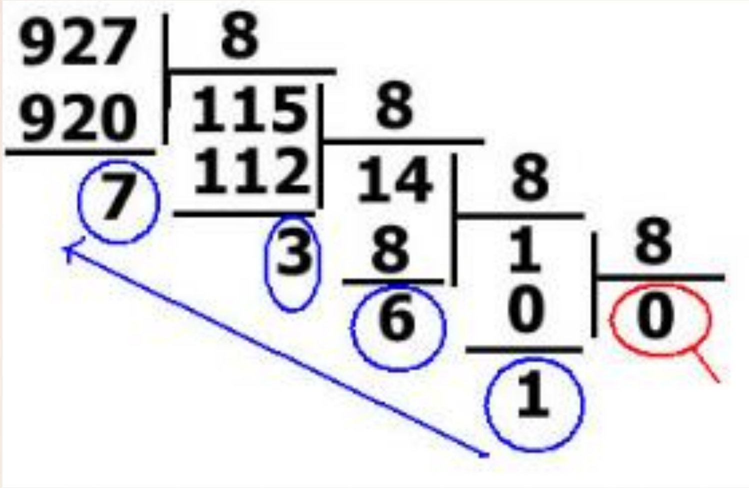
  - مثال: تبدیل $(11001)_2$ به مبنای ۱۰:

$$(11001)_2 = (1\times 2^4)+(1\times 2^3)+(0\times 2^2)+(0\times 2^1)+(1\times 2^0) = 16+8+0+0+1 = 25$$

nastooh@aut.ac.ir

سیستم اعداد

- ✓ تبدیل مبنا
  - ○ دهدهی به مبنای هشت (Octal)
    - ➢ یادآوری: اعداد مبنای هشت، فقط از هشت نماد 0 تا 7 تشکیل شده اند!
      - تقسیمات متوالی بر هشت
      - یادداشت باقیمانده ها از انتها
      - مثال: معادل مبنای هشت $(927)_{10}$ برابر $(1637)_8$



سیستم اعداد

- تبدیل مبنا
  - مبنای هشت به دهدهی
    - بسط عدد در مبنای ۱۰
    - مثال: تبدیل $(462)_8$ به مبنای ده:

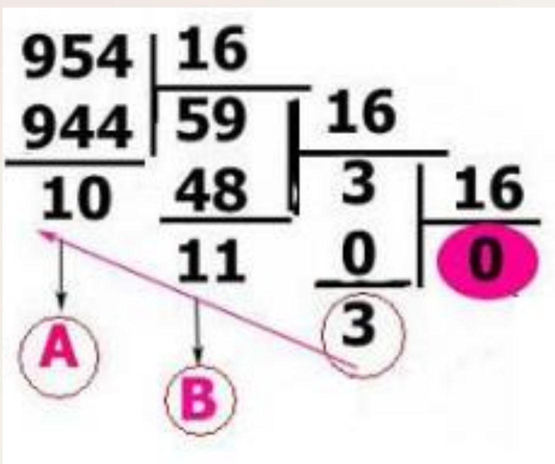
$$=4*8^2 + 6*8^1 + 2*8^0 = 4*64 + 6*8 + 2*1 = (306)_{10}$$

nastooh@aut.ac.ir

سیستم اعداد

✓ تبدیل مبنا

- دهدهی به مبنای شانزده (Hexadecimal)
  - یادآوری: اعداد مبنای شانزده، دارای نمادهای 0 تا 9 و A تا F هستند.
  - تقسیمات متوالی بر شانزده
  - یادداشت باقیمانده ها از انتها
  - برای باقیمانده های ۱۰ تا ۱۵، از نمادهای معادل آنها استفاده می کنیم.
  - مثال: معادل مبنای شانزده عدد $(954)_{10}$:



$(3BA)_{16}$

سیستم اعداد

✓ تبدیل مبنا

○ مبنای شانزده به دهدهی

- بسط عدد در مبنای ۱۰
- مثال: معادل مبنای ده عدد $(2B0F)_{16}$:

$$2*16^3 + 11*16^2 + 0*16^1 + 15*16^0$$

- یادآوری، نمادهای مبنای ۱۶:

10	16	10	16	10	16	10	16
0	0	4	4	8	8	12	C
1	1	5	5	9	9	13	D
2	2	6	6	10	A	14	E
3	3	7	7	11	B	15	F

nastooh@aut.ac.ir

سیستم اعداد

- تبدیل مبنا
  - مبنای دو به هشت
    - بسط عدد در مبنای ۸
      - محاسبات باید در مبنای هشت انجام شود. مثلا در جمع ها دیگر ده بر یک نداریم، بلکه هشت بر یک داریم و ...

راه ساده تر: هر سه رقم مبنای ۲، دقیقا معادل یک رقم مبنای هشت است. فقط کافیست از سمت راست سه رقم-سه رقم جدا کرده و معادل را جایگزین کنیم.

0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

مثال: معادل $(1011010)_2$

$$(001,011,010)_2 = (132)_8$$

nastooh@aut.ac.ir

سیستم اعداد

- تبدیل مبنا
  - مبنای هشت به دو
    - تقسیم های متوالی بر دو
      - محاسبات باید در مبنای هشت انجام شود. مثلا دیگر ده بر یک نداریم، بلکه هشت بر یک داریم و …

راه ساده تر: هر رقم مبنای هشت، دقیقا برابر ۳ رقم در مبنای دو است. فقط کافیست معادل را جایگزین کنیم.

مثال: معادل 8(2751)

$$\mathbf{(2751)_8 = (010,111,101,001)_2 = 10111101001}$$

سیستم اعداد

✓ تبدیل مبنا

○ مبنای دو به شانزده

- بسط عدد در مبنای ۱۶

➢ محاسبات باید در مبنای شانزده انجام شود. مثلا دیگر ده بر یک نداریم، بلکه شانزده بر یک داریم و ...

راه ساده تر: هر ۴ رقم مبنای شانزده، دقیقا برابر ۴ رقم در مبنای دو است. فقط کافیست از سمت راست معادل را جایگزین کنیم.

(جدول در اسلاید بعدی)

مثال: معادل $(111100101000100111)_2$

$$\mathbf{(0011,100,1010,0010,0111)_2=}$$

$$\mathbf{(3CA27)_{16}}$$

nastooh@aut.ac.ir

سیستم اعداد

✓ تبدیل مبنا

- مبنای شانزده به دو
  - تقسیم های متوالی بر دو
    - محاسبات باید در مبنای شانزده انجام شود. مثلا دیگر ده بر یک نداریم، بلکه شانزده بر یک داریم و …

راه ساده تر: هر رقم مبنای ۱۶، دقیقا معادل ۴ رقم مبنای دو است. فقط کافیست معادل را جایگزین کنیم.

(جدول در اسلاید بعدی)

مثال: معادل $(6B0F3)_{16}$

$$\mathbf{(6B0F3)_{16} = (0110,1011,0000,1111,0011)_2 = (1101011000011110011)_2}$$

سیستم اعداد

✓ تبدیل مبنا

○ جدول ۱۶ عدد ابتدایی در مبناهای چهارگانه

10	2	8	16
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

nastooh@aut.ac.ir

سیستم اعداد

- تبدیل مبنا
  - اعداد اعشاری
    - اعداد دهدهی اعشاری به دودویی
      - تبدیل قسمت صحیح و اعشاری به طور جداگانه
      - تبدیل قسمت صحیح همانند اسلایدهای قبل (تقسیم های متوالی بر دو)
      - تبدیل قسمت اعشاری، به کمک ضرب های متوالی در دو:
        - در هر مرحله قسمت اعشاری در دو ضرب می شود
        - رقم صحیحِ حاصل، یادداشت می شود (به ترتیب تولید)
        - دوباره قسمت اعشاری در دو ضرب می شود. تا زمانی که صفر شود
      - دو بخش صحیح و اعشاری با هم تلفیق می شود

سیستم اعداد

- تبدیل مبنا
  - اعداد اعشاری
    - اعداد دهدهی اعشاری به دودویی
      - مثال: تبدیل $(12.25)_{10}$ به مبنای دو
        - بخش صحیح: $(12)_{10}=(1100)_2$
        - بخش اعشاری:

$0.25*2=\underline{0}.5$

$0.5*2=\underline{1}.0$

- پس برای بخش اعشاری داریم: $(0.25)_{10}=(0.01)_2$
- جواب نهایی را اینگونه مینویسیم:

$(12.25)_{10}=1100.01)_2$

سیستم اعداد

✓ تبدیل مبنا

○ اعداد اعشاری

- اعداد دهدهی اعشاری به دودویی
  - مثال: تبدیل $(0.314)_{10}$ به مبنای دو
    - بخش اعشاری:

$$0.314*2=\underline{0}.628$$
$$0.628*2=\underline{1}.256$$
$$0.256*2=\underline{0}.512$$
$$0.512*2=\underline{1}.024$$

- تا کجا ادامه دهیم؟؟؟
- آیا حتما به صفر میرسیم؟
- هرچقدر دقت بیشتر می خواهیم، بیشتر ادامه می دهیم...
- پاسخ تا چهار رقم دقت: $(0.314)_{10}=(0.0101)_2$

nastooh@aut.ac.ir

سیستم اعداد

- تبدیل مبنا
  - اعداد اعشاری
    - اعداد دودویی اعشاری به دهدهی
      - به سادگی از همان بسط دادنِ همیشگی استفاده می کنیم.
      - حواسمان به توانهای عدد ۲ هست...
      - توضیح با یک مثال: تبدیل $(1110.01)_2$ به مبنای ده

$$(1110/01)_2 = (1\times 2^3) + (1\times 2^2) + (1\times 2^1) + (0\times 2^0) + (0\times 2^{-1}) + (1\times 2^{-2}) =$$

$$8+4+2+0+0+\frac{1}{4} = 14/25$$

nastooh@aut.ac.ir

سیستم اعداد

✓ تبدیل مبنا

○ اعداد اعشاری

- اعداد دودویی اعشاری به مبنای شانزده
  - آیا می توانید راهی برای این تبدیل بیابید؟
  - راهنمایی: از ممیز به سمت دو طرف پردازش کنید!!!

$(1001010.110101)_2 = (?)_{16}$

- اعداد مبنای شانزده اعشاری به مبنای دو
  - آیا می توانید راهی برای این تبدیل بیابید؟
  - راهنمایی: از ممیز به سمت دو طرف پردازش کنید!!!

$(A2.8C1)_{16} = (?)_2$

nastooh@aut.ac.ir

محاسبات دودویی

✓ جمع باینری

- همانند مبنای ده، فقط به جای ده بر یک، دو بر یک داریم...
  - یعنی داریم:
    - $0+0=0$
    - $0+1=1$
    - $1+0=1$
    - $1+1=10$ در این حالت، حاصل صفر است و یک (رقم نقلی) با رقم بعدی جمع میشود.
  - مثال:

$$\begin{array}{r} {\scriptstyle 1} \\ 110 \\ +\ 111 \\ \hline 1101 \end{array} \qquad \begin{array}{r} {\scriptstyle +1\,+1+1} \\ 101101 \\ +\ 011001 \\ \hline 1000110 \end{array} \qquad \begin{array}{rr} & {\scriptstyle 11111} \\ (85)_{10} & 01010101 \\ +(181)_{10} & +\ 10110101 \\ \hline (266)_{10} & 100001010 \end{array}$$

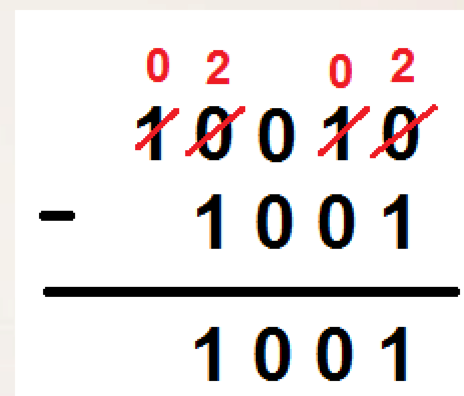
nastooh@aut.ac.ir

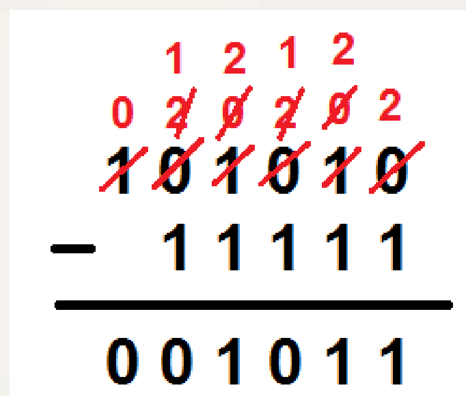
محاسبات دودویی

✓ تفریق باینری

○ همانند مبنای ده، فقط به جای ده بر یک، دو بر یک داریم...

- یعنی داریم:
  - 0-0=0
  - 1-0=1
  - 1-1=0
  - برای تفریق یک از صفر، از رقم کناری قرض میگیریم!
- مثال:





مبحث تفریق باینری ادامه دارد!

محاسبات دودویی

- ضرب باینری
  - بسیار شبیه به مبنای ده
    - یعنی داریم:
      - 0*0=0
      - 0*1=0
      - 1*0=0
      - 1*1=1
    - مثال:

```
      1 0 1
x     1 1 0
-----------
      0 0 0
+   1 0 1
+ 1 0 1
-----------
  1 1 1 1 0
```


nastooh@aut.ac.ir

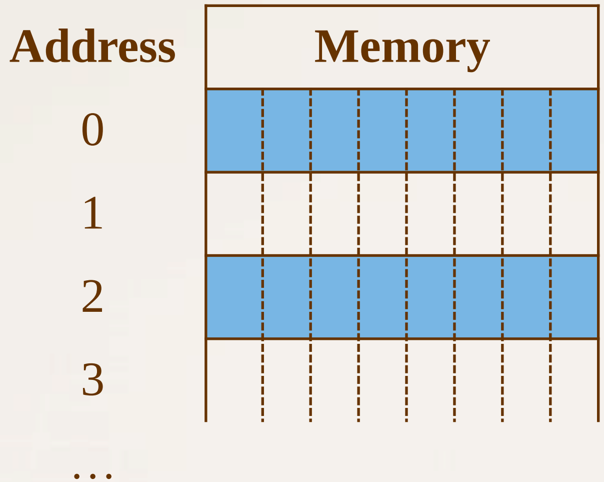
محاسبات دودویی

- تقسیم باینری
  - بسیار شبیه به مبنای ده

- تمرین: پس از کمی تفکر و تامل در مورد تقسیم باینری، ۵ مثال از تقسیم باینری حل کنید.

نگهداری داده ها در حافظه

- داده ها در حافظه در کلمات حافظه ذخیره می شوند.
  - میتوان گفت، کلمه واحد ذخیره سازی در حافظه است.
- طول کلمات حافظه از کامپیوتری به کامپیوتر دیگر متفاوت است.
- طول کلمات حافظه، معمولا توانی از ۲ است.
  - مثلا ۸ بیت یا یک بایت
  - ۱۶ بیت یا ۲ بایت
  - ۳۲ بیت یا ۴ بایت

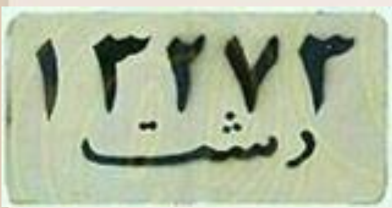


شمایی از یک حافظه با کلمات هشت بیتی

nastooh@aut.ac.ir

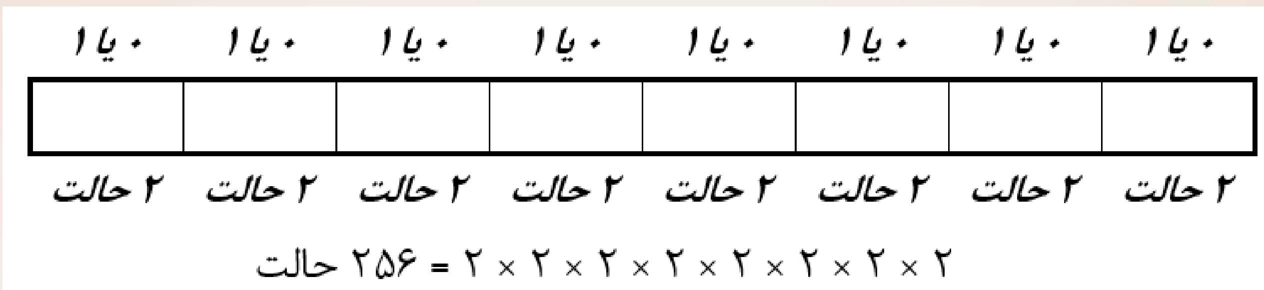
نگهداری داده ها در حافظه

- تمرین:
  - با توجه به ساختار پلاک خودروها در ایران، آیا می توانید **تعداد** خودروهای قابل پلاک شدن در هر شیوهٔ شماره گذاری را مشخص کنید؟



نگهداری داده ها در حافظه

- در یک بایت حافظه می توان ۲۵۶ حالت مختلف داده ذخیره کرد.



نگهداری داده ها در حافظه

- نگهداری اعداد صحیح مثبت در حافظه
  - به صورت مبنای ۲ از سمت راست در کلمه نوشته می شوند.
  - مابقی بیت ها با صفر پر می شود.

  - مثال: نگهداری عدد ۱۳ دهدهی در یک کلمهٔ هشت بیتی:

0	0	0	0	1	1	0	1

  - پس می توان نتیجه گرفت:
    - **محدودهٔ اعداد مثبت قابل ذخیره در یک کلمهٔ یک بیتی: ۰ تا ۱**
    - **محدودهٔ اعداد مثبت قابل ذخیره در یک کلمهٔ دو بیتی: ۰ تا ۳**
    - **محدودهٔ اعداد مثبت قابل ذخیره در یک کلمهٔ ۴ بیتی: ۰ تا ۱۵**
    - **محدودهٔ اعداد مثبت قابل ذخیره در یک کلمهٔ ۸ بیتی: ۰ تا ۲۵۵**
    - **محدودهٔ اعداد مثبت قابل ذخیره در یک کلمهٔ ۱۶ بیتی: ۰ تا $۲^{۱۶}-۱$**
    - **محدودهٔ اعداد مثبت قابل ذخیره در یک کلمهٔ ۳۲ بیتی: ۰ تا $۲^{۳۲}-۱$**

نگهداری داده ها در حافظه

- نگهداری اعداد صحیح منفی در حافظه
  - روش اول: علامت و مقدار
  - روش دوم: مکمل یک (Ones’ Complement)
  - روش سوم: مکمل دو (Two’s Complement)

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری اعداد صحیح منفی در حافظه
  - روش اول: علامت و مقدار
    - تقسیم یک کلمه به دو بخش
      1. بیت علامت (سمت چپ ترین بیت)
      2. مقدار عدد (مابقی بیت های سمت راست در کلمه)
    - نگهداری اعداد منفی همانند اعداد مثبت، فقط با بیت علامت یک
    - مثال: نمایش ۱۳+ و ۱۳- در روش علامت و مقدار در کلمات ۸ بیتی:

0	0	0	0	1	1	0	1

1	0	0	0	1	1	0	1

    - واضح است در کلماتی به طول M بیت، کوچکترین و بزرگترین عدد قابل نمایش $2^{(M-1)}-1$ و $-(2^{(M-1)}-1)$ هستند.

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری اعداد صحیح منفی در حافظه
  - روش دوم: مکمل یک
    - در این روش، اعداد منفی را از 11111...1 کم می کنند.
    - کماکان بیت سمت چپ، بیت علامت خواهد بود.
    - مثال: نمایش ۱۳+ و ۱۳- در روش مکمل یک در کلمات ۸ بیتی:

0	0	0	0	1	1	0	1

1	1	1	1	0	0	1	0

- **نکته مهم: انگار که تمام بیت های اعداد منفی، معکوس شده است!!**

- **با وجود روش علامت و مقدار چرا به مکمل یک نیاز داریم؟**

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری اعداد صحیح منفی در حافظه
  - روش دوم: مکمل یک
    - مزیت بزرگ روش مکمل یک: نیازی به مدار تفریق کننده نیست!!!
      - در واقع همه چیز با مدار جمع کننده انجام می شود.
    - روش تفریق: **عدد دوم** را به صورت مکمل یک نمایش می دهیم و سپس عملیات جمع را انجام می دهیم.
      - اگر بیت نقلی برای عملیات جمع پدید آمد، آن را دوباره با کل حاصل جمع می کنیم.

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری اعداد صحیح منفی در حافظه
  - روش دوم: مکمل یک
    - مثال از تفریق در مکمل دو:
    - در کلمات ۸ بیتی، از عدد ۱۱۰۱، کسر کنید عدد ۱۱۱۱ را.
      - نمایش عدد اول در هشت بیت: 00001101
      - نمایش عدد دوم در هشت بیت: 00001111
        - برای تفریق باید مکمل یکِ عدد دوم را نمایش دهیم، یعنی: 11110000
      - حال دو عدد را با هم جمع میکنیم، یعنی:

```
  00001101
+ 11110000
----------
  11111101
```

حاصل نمایش چه عددی است؟

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری اعداد صحیح منفی در حافظه
  - روش دوم: مکمل یک
    - مثال دوم از تفریق در مکمل دو:
    - در کلمات ۸ بیتی، از عدد ۱۱۰۰۱، کسر کنید عدد ۱۰۰۱ را.
      - نمایش عدد اول در هشت بیت: 00011001
      - نمایش عدد دوم در هشت بیت: 00001001
  - برای تفریق باید مکمل یکِ عدد دوم را نمایش دهیم، یعنی: 11110110
      - حال دو عدد را با هم جمع میکنیم، یعنی:

```
   00011001
+  11110110
-----------
  100001111
+         1
-----------
   00010000
```

حاصل نمایش چه عددی است؟

نگهداری داده ها در حافظه

- نگهداری اعداد صحیح منفی در حافظه
  - روش سوم: مکمل دو

➢ یک مشکل روش مکمل یک (و البته علامت و مقدار)

» **وجود دو نمایش جداگانه برای صفر**

» در روش مکمل یک، در کلمات هشت بیتی، این دو مقدار را مقایسه کنید

00000000

11111111

➢ برای نمایش اعداد در مکمل دو، ابتدا آنها را به مکمل یک تبدیل کرده، سپس یک واحد افزایش میدهیم!

➢ در این روش نیز، بیت سمت چپ، بیت علامت است.

➢ در این روش نیز به مدار تفریق کننده جداگانه نیاز **نداریم**.

➢ در این روش با M بیت، محدودهٔ اعداد 2^{M-1} و $-(2^{M-1}-1)$ را داریم.

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری اعداد اعشاری در حافظه
  - یادآوری: مفهوم ممیز شناور

➢ اعداد اعشاری در هر مبنایی می تواند به صورت ممیز شناور نمایش داده شود. مثال: نمایش عدد ۵۷۳:

$573 * 10^0$

$57.3 * 10^1$

$5.73 * 10^2$

$0.573 * 10^3$

$0.0573 * 10^4$

صورت کلی: هر عدد اعشار را می توان به صورت زیر نشان داد:

$$\pm f \times b^{\pm e}$$

که f مقدار عدد، b پایه و e توان نامیده می شود.

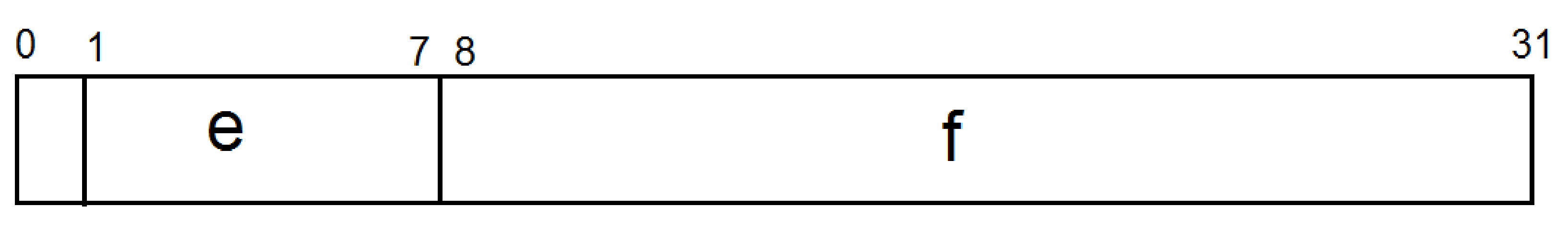
نگهداری داده ها در حافظه

- نگهداری اعداد اعشاری در حافظه
  - اگر f همیشه بین صفر و یک باشد، عدد را نرمال گویند.
  - برای مشخص کردن یک مقدار اعشاری باید مقادیر f و e و b تعیین گردند.
  - در کامپیوتر، چون همواره با یک مبنای خاص کار میکند، نیازی به ذخیرهٔ b نیست.

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری اعداد اعشاری در حافظه
  - یکی از استانداردهای ذخیره سازی اعداد اعشاری، در کلمات ۴ بایتی به صورت زیر است:



- بیت اول (شماره صفر): علامت بخش مقدار کسر
- هفت بیت بعد (از ۱ تا ۷) مقدار توان
  - توان میتواند مقداری بین ۶۴- تا ۶۳+ داشته باشد.
  - اما برای حذف علامت آن حین ذخیره سازی، با روش "افزودنی ۶۴" ذخیره می شود. (هنگام ذخیره ۶۴ واحد به آن افزوده می شود)
- ۲۴ بیت بعدی مقدار کسر
- مقدار پایه (b) را معمولا ۱۶ در نظر میگیرند.

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری اعداد اعشاری در حافظه
  - مثال: عدد $(121.84)_{10}$ را با استاندارد صفحه قبل ذخیره کنید.
    - با فرض مقدار پایه برابر ۱۶ حل میکنیم.
    - $(121)_{10}=(1111001)_2=(79)_{16}$
    - $(0.84)_{10}=(0.110101)_2=(0.D4)_{16}$
    - $(121.84)_{10}=(79.D4)_{16}=0.79D4 * 16^2$
    - پس داریم: $f=0.79D4$ و $e=2$
    - و برای ذخیره e به روش افزودنی ۶۴ داریم:
    - $e=2 + 64 = 66 = (1000010)_2$

0	1 … 7	8 … 31
0	1 0 0 0 0 1 0	0 1 1 1 1 0 0 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- مفهوم سرریزی (Overflow)
  - هنگام ذخیره اعداد در حافظه، اگر حجم عدد بیشتر از فضای تخصیص یافته باشد، بخشی از عدد از بین می رود، که به این اتفاق سرریزی گویند.
  - این اتفاق هم موقع ذخیره سازی اعداد صحیح رخ می دهد و هم هنگام ذخیرهٔ مقادیر اعشاری. به سناریوی رخ دادن هر دو خوب فکر کنید.
  - عموماً وظیفهٔ جلوگیری از سرریزی با برنامه نویسی است.
    - با انتخاب نوع داده و فضای ذخیره سازی مناسب از قبل...

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- نگهداری داده های کاراکتری در حافظه
  - امروزه، داده های عددی تنها داده هایی نیستند که کامپیوتر با آنها سر و کار دارد.
  - داده های کاراکتری یک رشته از نمادها هستند
  - مانند حروف الفبا، علائم ویژه مانند % و @ و حتی نشانه هایی مانند 0 تا 9
  - داده های کاراکتری نیز باید در مبنای دو ذخیره شوند.
  - برای این تبدیل استانداردهایی مانند زیر وجود دارد:
    - BCD
    - ASCII
    - EBCDIC
    - UniCode
  - هر کدام، طِی یک استاندارد، برای هر کاراکتر یک معادل در نظر میگیرند.

نگهداری داده ها در حافظه

- نگهداری داده های کاراکتری در حافظه
  - استاندارد ASCII
    - در ابتدا ۷ بیتی بود، سپس ۸ بیتی شدند.

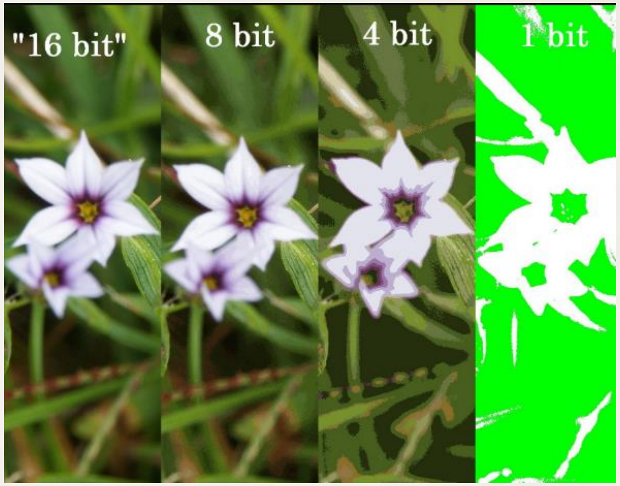
کاراکتر	مبنای 10
Space	32
@	64
*	42
#	35
[	93
{	123
(	40
-	61
…	

کاراکتر	مبنای 10
A	65
…	
Z	90
a	97
…	
z	122
0	48
…	
9	57

nastooh@aut.ac.ir

نگهداری داده ها در حافظه

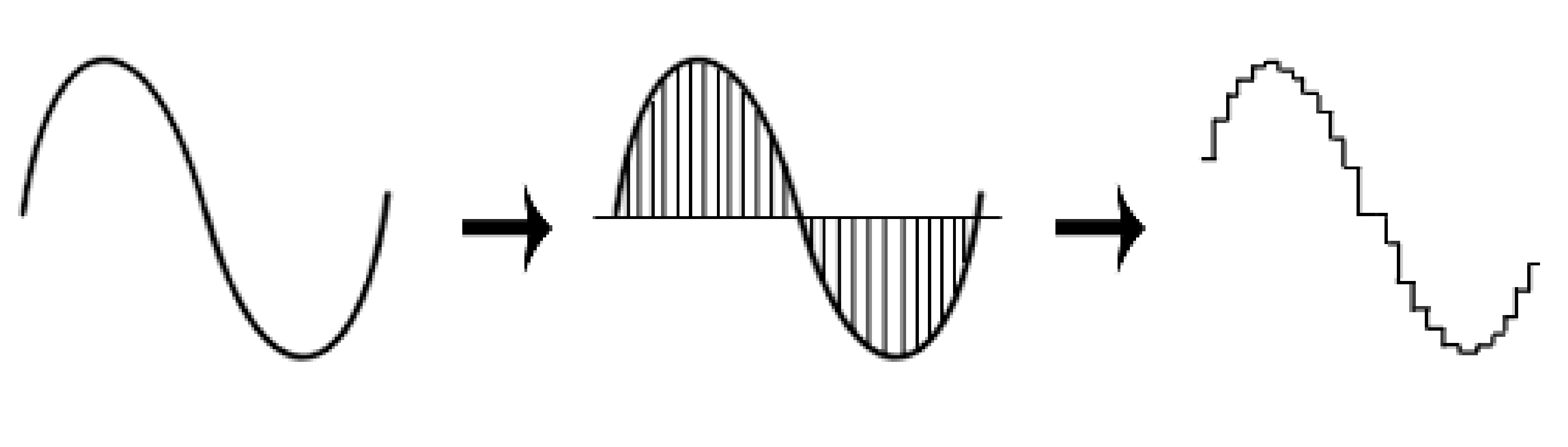
- نگهداری تصویر در حافظه
  - تصاویر دیجیتال در قالب تعدادی نقطه نورانی (پیکسل) ذخیره می شوند.
  - به ازای هر پیکسل باید اطلاعاتی ذخیره شوند.
    - اطلاعاتی مانند رنگ، نور و و و
  - کیفیت تصاویر دیجیتال، عموما به دو عامل بستگی دارد:
    - سایز تصویر (تعداد پیکسل ها)
    - تنوع رنگ قابل استفاده (عمق تصویر)



nastooh@aut.ac.ir

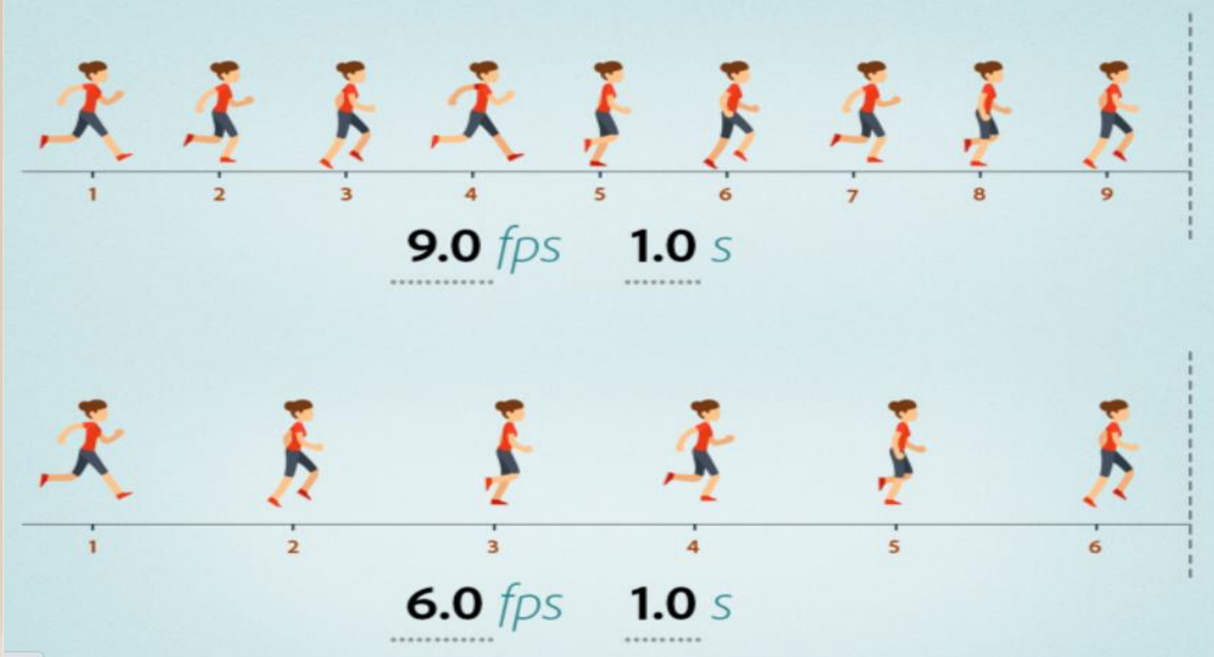
نگهداری داده ها در حافظه

- نگهداری صوت در حافظه
  - برای تبدیل صوت به اعداد باینری، باید از صوت نمونه برداری شود.
  - هر چه نرخ نمونه برداری بالاتر باشد، کیفیت صدای ذخیره شده افزایش می یابد، اما حجم آن نیز بالاتر می رود.
  - داده های مربوط به هر نمونه، باید به صورت عددی ذخیره شوند.



نگهداری داده ها در حافظه

- نگهداری ویدئو در حافظه
  - ویدئو در واقع نمایش دادن چند تصویر پشت سر هم و با سرعت مناسب است.
  - به دلیل خطای چشم انسان، **اثر** تصویر قبلی بر روی تصویر بعدی تاثیر می گذارد و انسان تصاویر را پیوسته می بیند!
  - این تصاویر ثابت باید به نحو مناسبی (با جزئیات کامل) ذخیره شده و به درستی نمایش داده شوند.



nastooh@aut.ac.ir

نگهداری داده ها در حافظه

- آیا امروزه نوع دادهٔ دیگری نیز در کامپیوتر ذخیره می شوند؟

در آینده چطور؟

(قدری به آن فکر کنید!)

nastooh@aut.ac.ir

پایان

nastooh@aut.ac.ir