

# زبان ماشین و اسمبلی

زبان ماشین و اسمبلی

مدرس:

خانم مهندس مظاہری

ما انسان‌ها چگونه می‌توانیم با کامپیوتر ارتباط برقرار کنیم؟

**زبان برنامه نویسی**

زبانی است که برای کامپیوتر قابل فهم بوده و الگوریتمها با استفاده از آن به کامپیوتر داده می‌شوند.

به عنوان مثال دستورات زیر را در نظر بگیرید:

```
#include <iostream>
using namespace std;
int main() {
    int a , b;
    cin >> a >> b;
    cout << a+b;
    return 0;
```

**زبان ++C**

```
a = int(input())
b = int(input())
print(a+b)
```

**زبان Python**

هرچند نوع نگارش این دو زبان با هم متفاوت است، اما هر دوی آنها در اصل یک کار یکسان را انجام می دهند و از رایانه می خواهند دو عدد **a** و **b** را از ورودی دریافت کرده و حاصل جمع آنها را محاسبه کرده و نمایش دهد.

## زبان های برنامه نویسی:

اصولاً زبان های برنامه نویسی کامپیوتر به سه دسته کلی تقسیم می شوند:

- زبانهای سطح پایین
- زبانهای سطح میانه
- زبانهای سطح بالا

زبان ماشین و اسمبلی

## زبان های سطح پایین :

زبان های سطح پایین از لحاظ ساختار و ترجمه ، زبان هایی محسوب می شوند که با سخت افزار ارتباط نزدیکتری برقرار می کنند. (با استفاده از این زبان ها دسترسی به سخت افزار بیشتر و مدیریت حافظه بهتر و همچنین قدرت مانور برای کسانی که میخواهند با سخت افزار کار کنند بیشتر است)

## زبان ماشین (زبان صفر و یک)

- زبان ماشین، زبان ذاتی و انحصاری رایانه است که هنگام طراحی سخت افزار رایانه تعریف می شود.
- هر کامپیوتری زبان ماشین مخصوص به خود را دارد که وابسته به سخت افزار خود آن کامپیوتر است.
- زبان ماشین، شامل رشته ای از اعداد است که هر کدام از آنها نشان دهنده ی یک دستورالعمل یا مفهوم خاص برای کامپیوتر هستند.
- زبان ماشین سبب می شود که رایانه عملیات اصلی مربوط به خود را در هر بار راه اندازی، اجرا کند.

بعنوان مثال در زبان ماشین رشته ی عددی ۰۱۰۰۱۱۰۱ نشان دهنده عملیات جمع در کامپیوتر می باشد

## زبان های سطح میانی:

مشکلی که سر راه برنامه‌نویسان وجود داشت این بود که ارتباط با یک کامپیوتر به زبان باینری، برای انسان‌ها کاری دشوار و تقریباً غیر ممکن بود، به همین علت تصمیم گرفتند زبان‌هایی را برای ارتباط با رایانه‌ها طراحی کنند که تا حدودی به زبان انسان نزدیک‌تر بوده و برنامه‌نویسی با آن‌ها ساده‌تر باشد. برنامه‌نویسان به جای به کار بردن رشته‌ای از اعداد که رایانه بتواند به طور مستقیم آن‌ها را درک کند برای هر دستور یک معادل کلمه‌ای در نظر گرفتند و از آن به بعد به کمک آن کلمات برنامه‌نویسی می‌کردند، که زبان اسمبلی (Assembly) شکل گرفت. زبان اسمبلی را بعضی‌ها زبان سطح پایین و بعضی‌ها زبان سطح میانی معرفی می‌کنند.

به جای کد ۰۱۰۰۱۱۰۱ عبارت SUM که به معنای عملیات جمع تلقی می‌شد را به ماشین تحویل دادند

## اسمبلر:

در نهایت تمام این دستورات برای فهم کامپیوتر، باید به زبان باینری تبدیل می‌شد، که این فرایند (یعنی تبدیل دستورات اسمبلی به باینری) بر عهده‌ی اسمبلر بود.

**نکته:** تبدیل زبان اسمبلی خود احتیاج به پردازش جداگانه‌ای داشت، بنابراین از سرعت کمتری نسبت به زبان باینری برخوردار بود.

## زبان های سطح بالا:

زبان اسمبلی سبب افزایش سرعت برنامه نویسی شد اما هنوز هم برای انجام یک عمل ساده مستلزم دستورهای فراوانی بود و برنامه نویس باید به جزئیات توجه بیشتری نشان می داد و همچنین می بایستی اطلاعات کافی نسبت به پردازنده داشت. که این امر برای برنامه نویسان یک مشکل بود به همین علت برای افزایش سرعت برنامه نویسی، زبان های سطح بالا توسعه پیدا کردند که با استفاده از یک عبارت می توانند وظایف و اعمال وسیعتری را انجام دهند و امکانات و قابلیت های گسترده تری را در اختیار برنامه نویسان قرار می دهند و با کاربر ارتباط بهتری برقرار می کنند و به زبان انسان نزدیکترند.

## مترجم ها:

وظیفه مترجم تبدیل دستورات زبان سطح بالا به زبان قابل فهم برای کامپیوتر است.

دو نوع اصلی مترجم داریم که عبارتند از:

- **کامپایلر:** ابتدا کل برنامه زبان سطح بالا را بررسی کرده و در صورت نبود خطا، کل آن را به زبان ماشین تبدیل می کند و هم اکنون برنامه آماده اجرا است.
- **مفسر:** برنامه زبان سطح بالا را دستور به دستور به زبان ماشین تبدیل و همزمان آن را اجرا می کند.

## نکته:

هرچند زبان‌های سطح بالا سادگی و کارایی بیشتری را در اختیار برنامه‌نویس قرار می‌دهند و از محبوبیت بیشتری نزد برنامه‌نویسان برخوردارند. اما زبان‌های سطح پایین مانند باینری از سرعت بیشتری برخوردار هستند، چراکه فرایند تبدیل و کامپایل در آن‌ها وجود نداشته و پردازش کمتری توسط رایانه انجام می‌شود.

## کاربردها:

- امروزه در سیستم‌هایی که احتیاج به سرعت پردازش بالایی دارند، به‌عنوان مثال فضاپیماها، سیستم‌های بیمارستانی یا ربات‌های خاص از زبان‌های سطح پایین‌تر مانند اسمبلی و حتی باینری استفاده می‌شود.
- در برنامه‌نویسی‌های معمولی مانند اپلیکیشن‌های کامپیوتر، موبایل و وب از زبان‌های سطح بالا استفاده شده و وظیفه‌ی کامپایل یا ترجمه‌ی دستورات برعهده‌ی کامپیوتر گذاشته می‌شود.



## مثال :

- قطعه کد زیر می تواند اینگونه تعبیر شود که `basepay` و `overpay` را باهم جمع کن و حاصل آن را در `grosspay` ذخیره کن:

+1300042774

+1400593419

+1200274027

load basepay

add overpay

store grosspay

- کد اسمبلی مثال بالا بصورت زیر است:

$grosspay = basepay + overpay$

- برای مثال بالا داریم:

در هنگام کار با داده ها در حافظه با اصلاحاتی روبرو می شویم که در زیر به معرفی آنها می پردازیم:

**بیت:** حافظه از واحدهای کوچکی بنام بیت تشکیل شده است که هر بیت قابلیت نگاهداری یک ۰ یا ۱ را در خود دارد.

**بایت:** به هر ۸ بیت یک بایت گفته می شود که واحد اندازه گیری حافظه است.

کوچکترین واحد حافظه بیت است. واحد حافظه بزرگتر از بیت، بایت است که از بیت تشکیل شده است.

**واحدهای بزرگتری برای سنجش میزان حافظه وجود دارد مانند:**

**کلمه:** معمولا هر ۲ بایت را یک کلمه در نظر می گیرند.

**کیلوبایت (KB):** هر ۱۰۲۴ بایت یک کیلو بایت را تشکیل می دهد. در سیستم دودویی هر کیلو بایت معادل  $2^{10}$  بایت می باشد.

**واحدهای بزرگتر از کیلو بایت عبارتند از:**

1 MegaByte or 1MB = 1024 KiloByte(KB)

1 GigaByte or 1GB = 1024 MegaByte(MG)

1 TeraByte or 1TB = 1024 GigaByte(GB)

$$1\text{MB} = 1024 \text{ KB}$$

$$1\text{GB} = 1024 \text{ MB}$$

$$1\text{TB} = 1024 \text{ GB}$$

مثال: ۲ ترابایت چند کیلوبایت است؟

$$2 \text{ TB} = 2 \times 1024 \text{ GB} = 2 \times 1024 \times 1024 \text{ MB} = 2 \times 1024 \times 1024 \times 1024 \text{ KB}$$

مثال: ۱ گیگابایت چند کیلوبایت است؟

$$\text{GB} = 1024 \text{ MB} = 1024 \times 1024 \text{ KB}$$

## سیستم اعداد

- مبنا عبارت است از الفبای عدد نویسی.
- یک مبنا تعیین میکند که با چه رقم هایی می توان عدد ساخت.
- در سیستم عددی بر مبنای  $b$  می توان از  $b$  رقم که عضو مجموعه  $\{0,1,2,3,\dots,b-2,b-1\}$  می باشد، استفاده کرد و می توان اعداد را به هر مبنای دلخواهی نیز تبدیل نمود.

- در سیستم دهدهی (دسیمال) می توان از ۱۰ رقم که عضو مجموعه  $\{0,1,2,3,4,5,6,7,8,9\}$  می باشد، استفاده کرد.

- در سیستم دودویی (باینری) می توان از ۲ رقم که عضو مجموعه  $\{0,1\}$  می باشد، استفاده کرد.

- در سیستم عددی بر مبناء ۸ (اوکتال) می توان از ۸ رقم که عضو مجموعه  $\{0,1,2,3,4,5,6,7\}$  می باشد، استفاده کرد.

- در سیستم عددی بر مبناء ۱۶ (هگزا دسیمال) می توان از ۱۶ رقم که عضو مجموعه  $\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$

می باشد استفاده کرد و به منظور جلوگیری از ابهام، ارقام ۱۰ تا ۱۵ را به ترتیب با حروف A تا F نشان می دهند.

$$(1212)_{16} \longrightarrow (12c)_{16}$$

## مثال:

نمایش اعداد در مبنای ۷ را در نظر بگیرید. ارقامی که برای نمایش اعداد در این مبنا بکار می رود عضو مجموعه  $\{0,1,2,3,4,5,6\}$  می باشند. با در نظر گرفتن این مبنا نمایش های اعدادی که در این مبنا در ستون سمت چپ آمده اند درست و نمایش اعداد در ستون سمت راست نادرست است.

$(123456)_7$  ✓

$(7654321)_7$  ✗

$(654632)_7$  ✓

$(5432816)_7$  ✗

## تبدیل مبنایا (۴ روش کلی)

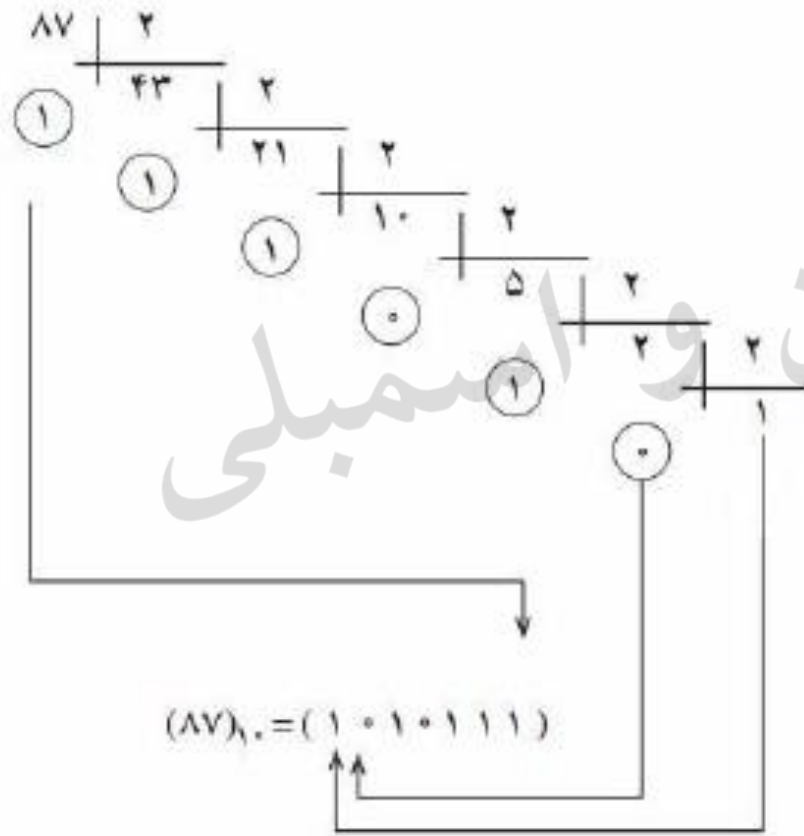
- جهت تبدیل اعداد صحیح از مبنای ۱۰ به هر مبنای دیگری می توان از روش تقسیم های متوالی استفاده کرد. (تقسیم مورد نظر بر همان عددی انجام می شود که قرار است تبدیل به آن مبنای صورت گیرد. نگهداری باقیمانده ها و آخرین خارج قسمت)

**مثال:** در زیر روند تبدیل مبنای  $(897)_{10}$  به مبنای 7 نشان داده شده است.

897	7		
896	128	7	
①	126	18	7
	②	14	2
			④

$$(897)_{10} = (2421)_7$$

**مثال:** در زیر روند تبدیل مبنای  $(87)_{10}$  به مبنای 2 نشان داده شده است.



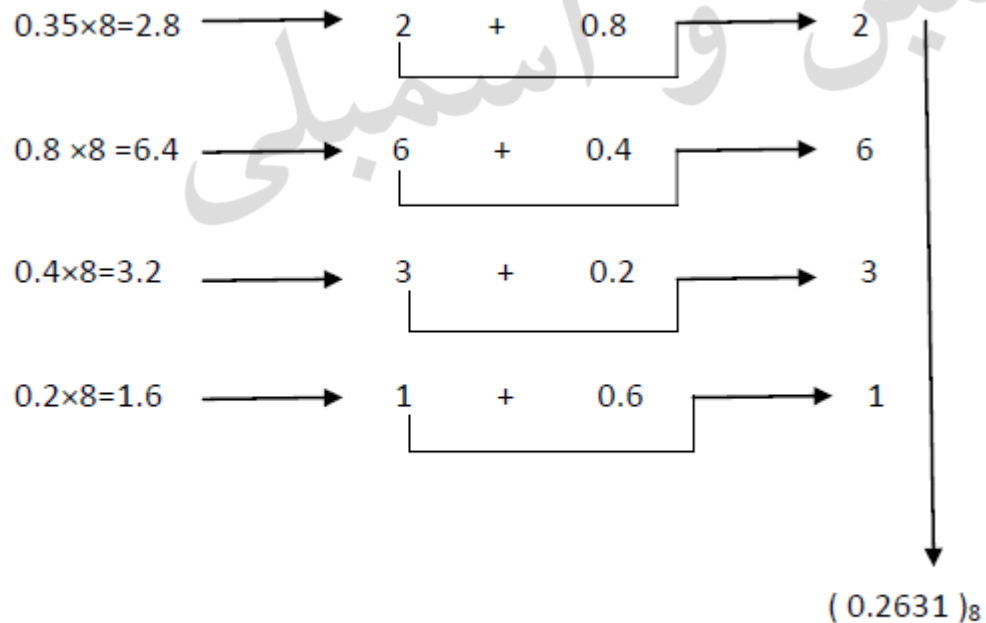
زبان ماشین و اسمبلی



- جهت تبدیل اعداد اعشاری از مبنای ۱۰ به هر مبنای دیگری می توان از روش ضرب های متوالی استفاده کرد. قسمت صحیح آن به روش گفته شده یعنی با تقسیمات متوالی به مبنای مورد نظر تبدیل می شود و سپس قسمت اعشاری آن، بطور مکرر در مبنای مورد نظر ضرب می شود.

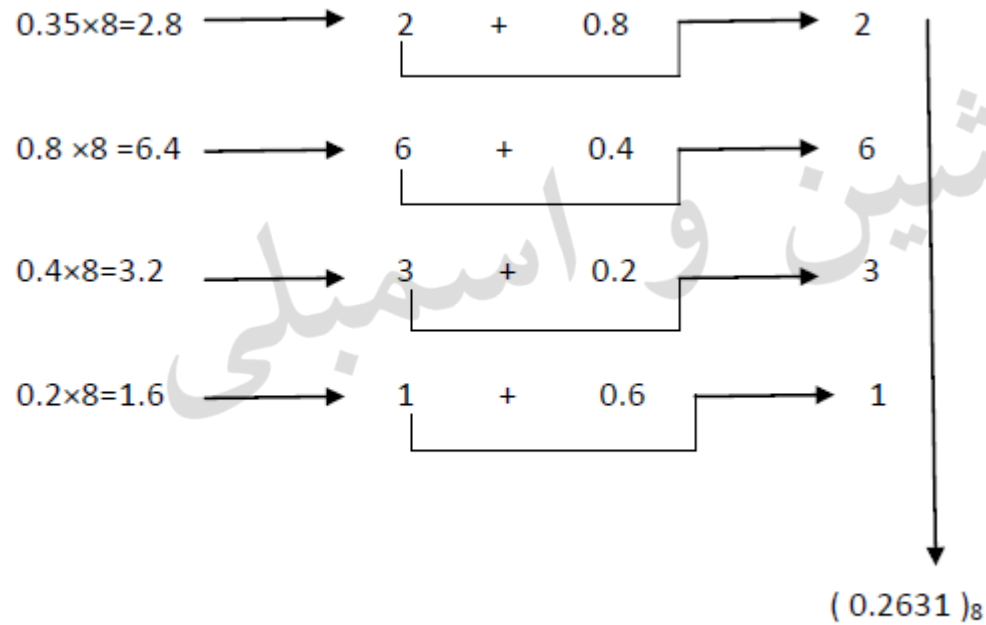
مثال:

$$(0.35)_{10} = (?)_8$$



**مثال:** در زیر روند تبدیل مبنای  $(28.35)_{10}$  به مبنای 8 نشان داده شده است

$$(28.35)_{10} = (?)_8$$



$$\begin{array}{r|l} 28 & 8 \\ \hline 24 & 3 \\ \hline & 4 \end{array}$$

$$(28)_{10} = (34)_8$$

$$(28.35)_{10} = (34.2631)_8$$

- جهت تبدیل اعداد صحیح و اعداد اعشاری از مبناهای دیگر به مبنا ۱۰ می توان از جایگاه یا وزن ارقام استفاده کرد.  
( اعداد در ارزش مکانی خود ضرب و حاصل ضرب ها با هم جمع می شود )

جایگاه یا وزن ارقام ← ۳ ۲ ۱ ۰ -۱ -۲ -۳

$$( a b c d / e f g )_N = (a \times N^3 + b \times N^2 + c \times N^1 + d \times N^0 + e \times N^{-1} + f \times N^{-2} + g \times N^{-3})_{10}$$

مثال:

جایگاه یا وزن ارقام ۳ ۲ ۱ ۰

- $(2421)_7 = 2 \times 7^3 + 4 \times 7^2 + 2 \times 7^1 + 1 \times 7^0 = 686 + 196 + 14 + 1 = (897)_{10}$

مثال:

جایگاه یا وزن ارقام ۳ ۲ ۱ ۰ -۱ -۲ -۳

- $(1010.101)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 8 + 0 + 2 + 0 + 0.5 + 0 + 0.125 = (10.625)_{10}$

- جهت تبدیل یک عدد از یک مبناء مشخص به مبناء مشخص دیگر، بهتر است که ابتدا عدد مورد نظر به مبناء ۱۰ تبدیل و سپس طبق نکات قبل عمل شود.

مثال:

- $(142)_5 = (?)_9$

$$(142)_5 \quad (?)_{10} \quad (?)_9$$

$$(142)_5 = 1 \times 5^2 + 4 \times 5^1 + 2 \times 5^0 = 25 + 20 + 2 = (47)_{10} = (52)_9$$

مثال:

- $(ABCF)_{16} = (?)_8$

$$(ABCF)_{16} = (?)_{10} = (?)_8$$

$$(ABCF)_{16} = A \times 16^3 + B \times 16^2 + C \times 16^1 + F \times 16^0 = 40960 + 2816 + 192 + 15 = (43983)_{10} = (125717)_8$$

↓
↓
↓
↓

10
11
12
15

تقسیمات متوالی

## مبناهای دو، هشت، شانزده

$$8=2^3$$

• هر رقم در مبناي هشت از ۳ رقم دودویی تشکیل شده است

مبناي ۸	۰	۱	۲	۳	۴	۵	۶	۷
مبناي ۲	۰۰۰	۰۰۱	۰۱۰	۰۱۱	۱۰۰	۱۰۱	۱۱۰	۱۱۱

**نکته:** برای تبدیل از مبنای ۸ به مبنای ۲، بازای هر رقم مبنای ۸، ۳ رقم در مبنای ۲ قرار داده می شود.  $8=2^3$

• **مثال:**

$$(367)_8 = (11110111)_2$$

**نکته:** تبدیل از مبنای ۲ به مبنای ۸ به سادگی انجام می گیرد، ابتدا ارقام را از راست به چپ بصورت دسته های ۳ تایی تقسیم کنید و سپس برای هر دسته معادل آن را در مبنای ۸ قرار دهید و برای دسته آخر اگر تعدادشان کمتر از ۳ بود به سمت چپ آن ۰ اضافه کنید تا تعدادشان به ۳ برسد.

• **مثال:**

$$(10011111)_2 = (?)_8$$

$$\begin{array}{ccc} 421 & 421 & 421 \\ \underline{010} & \underline{011} & \underline{111} \\ 2 & 3 & 7 \end{array} = (237)_8$$

**نمایش یک عدد در مبنای هشت نسبت به نمایش همان عدد در مبنای دو ارقام کمتری دارد**

$$16=2^4$$

• هر رقم در مبنای ۱۶ از ۴ رقم دودویی تشکیل شده است

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

**نکته:** برای تبدیل از مبنای ۱۶ به مبنای ۲ بازای هر رقم مبنای ۱۶، ۴ رقم در مبنای ۲ قرار می دهیم.

• **مثال:**

$$(3BF)_{16} = (111011111)_2$$

Diagram illustrating the conversion of hexadecimal (3BF)<sub>16</sub> to binary (111011111)<sub>2</sub>. The hexadecimal digits 3, B, and F are converted to their 4-bit binary equivalents: 3 → 0011, B → 1011, and F → 1111. The positions 8, 4, 2, 1 are indicated above the binary strings.

**نکته:** برای تبدیل اعداد از مبنای ۲ به ۱۶ از همان روش گفته شده برای مبنای ۸ استفاده می شود با این تفاوت که از سمت راست بصورت دسته های ۴ تایی جدا کنید و برای دسته آخر اگر تعدادشان کمتر از ۴ بود به سمت چپ آن ۰ اضافه تا تعدادشان به ۴ برسد.

• **مثال:**

$$(1010011101)_2 = (?)_{16}$$
$$\begin{array}{ccc} \underline{0010} & \underline{1001} & \underline{1101} = (29D)_{16} \\ 2 & 9 & D \end{array}$$

**یک عدد در مبنای شانزده نیز نسبت به نمایش همان عدد در مبنای دو ارقام کمتری دارد.**



کامپیوترها با سیستم دودویی کار می کنند پس چگونه می توان حروف و کلمات را در آنها ذخیره کرد؟ برای این کار به هر کاراکتر عددی را نسبت می دهیم که با عنوان «رمزگذاری کاراکترها» شناخته می شود.

## کد اسکی: ASCII

این روش برای نمایش حروف الفباء و کاراکترهای مختلف و همچنین نقل و انتقالات آنها در دستگاههای جانبی مختلف بکار رفته که از یک استاندارد خاص تبعیت شده است. این کد معمولاً از هفت بیت و گاهی از هشت بیت تشکیل شده است. در مجموع شامل ۱۲۸ کاراکتر به شرح زیر می باشد:

$$2^7=128 \quad 2^8=256$$

۲۶ کد شامل حروف کوچک انگلیسی

۲۶ کد شامل حروف بزرگ انگلیسی

۱۰ کد شامل اعداد 0 الی 9

۳۲ کد شامل کاراکترهای قابل چاپ مانند \$ @ % & \*

۳۴ کد شامل کاراکترهای غیر قابل چاپ (کنترلی) Delete Enter Ctrl

هر بار که یکی از کلیدهای صفحه کامپیوتر را فشار دهید یک کد اسکی برای پردازش به کامپیوتر فرستاده میشود

حرف	کد باینری	حرف	کد باینری
A	100 0001	0	011 0000
B	100 0010	1	011 0001
C	100 0011	2	011 0010
D	100 0100	3	011 0011
E	100 0101	4	011 0100
F	100 0110	5	011 0101
G	100 0111	6	011 0110
H	100 1000	7	011 0111
I	100 1001	8	011 1000
J	100 1010	9	011 1001
K	100 1011		
L	100 1100	space	010 0000
M	100 1101	.	010 1110
N	100 1110	(	010 1000
O	100 1111	+	010 1011
P	101 0000	\$	010 0100
Q	101 0001	#	010 1010
R	101 0010		010 1001
S	101 0011	-	010 1101
T	101 0100	/	010 1111
U	101 0101	:	010 1100
V	101 0110	=	011 1101
W	101 0111		
X	101 1000		
Y	101 1001		
Z	101 1010		

**نکته:** تفاوت حروف بزرگ با حروف کوچک فقط در بیت ششم است. این بیت در حروف بزرگ صفر و در حروف کوچک یک است.

"m" = 

0	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---



"M" = 

0	1	0	0	1	1	0	1
---	---	---	---	---	---	---	---

7 6 5 4 3 2 1 0  
**E**

0	1	0	0	0	1	0	1
---	---	---	---	---	---	---	---

7 6 5 4 3 2 1 0  
**e**

0	1	1	0	0	1	0	1
---	---	---	---	---	---	---	---

## جمع دودویی

$$\begin{array}{r} 0 \\ + \\ \hline 0 \\ 0 \end{array}$$

$$\begin{array}{r} 0 \\ + \\ \hline 1 \\ 1 \end{array}$$

$$\begin{array}{r} 1 \\ + \\ \hline 0 \\ 1 \end{array}$$

$$\begin{array}{r} 1 \\ + \\ \hline 1 \\ 10 \end{array}$$

در جمع آخر، ۱، رقم نقلی است که با بیت های بعدی جمع می شود.

مثال: جمع زیر را در مبنای ۲ انجام دهید.

$$\begin{array}{r} 13 \\ + \\ \hline 9 \\ \hline ? \end{array}$$

$$13 = (1101)_2$$

$$9 = (1001)_2$$

$$\begin{array}{r} 1101 \\ + \\ \hline 1001 \\ \hline 10110 \end{array}$$

$$(10110)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 16 + 0 + 4 + 2 + 0 = 22$$

امتحان نتیجه:

مثال: جمع زیر را در مبنای ۲ انجام دهید.

$$\begin{array}{r} 29 \\ + \\ \hline 17 \\ \hline ? \end{array}$$

$$29 = (11101)_2$$

$$17 = (10001)_2$$

$$\begin{array}{r} 11101 \\ + \\ \hline 10001 \\ \hline 101110 \end{array}$$

$$(101110)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 32 + 8 + 4 + 2 + 0 = 46$$

امتحان نتیجه:

نکته : جمع اعداد در مبناء ۱۶ بصورت زیر انجام می شود:

$$\begin{array}{r} 23D9 \\ + \\ \hline 94BE \\ \hline B897 \end{array}$$

$$9+E=23 \longrightarrow 23-16=7$$

$$1+D+B=25 \longrightarrow 25-16=9$$

$$1+3+4=8$$

$$2+9=B$$

زبان ماشین و اسمبلی



مثال: جمع زیر را در مبنای ۱۶ انجام دهید.

$$\begin{array}{r} 1 \quad 1 \\ 7 \ E \ C \ 6 \\ + \\ \hline 3 \ 4 \ 0 \ A \\ \hline B \ 2 \ D \ 0 \end{array}$$

$$6+A=6+10=16 \longrightarrow 16-16=0$$

$$C+0+1=12+0+1=13 \longrightarrow D$$

$$E+4=14+4=18 \longrightarrow 18-16=2$$

$$7+3+1=11 \longrightarrow B$$

## مکمل ها

الف) مکمل ۱: برای بدست آوردن مکمل ۱ بیت های آن عدد را معکوس می کنیم. (تبدیل ۱ ها به صفرها و تبدیل صفرها به ۱ ها)

ب) مکمل ۲: مکمل ۱ + ۱

## مقادیر منفی

اعداد و مقادیر منفی در کامپیوتر با استفاده از روش مکمل ۲ نمایش داده می شوند.

**الف)** مکمل ۱: برای بدست آوردن مکمل ۱ بیت های آن عدد را معکوس می کنیم. (تبدیل ۱ها به صفرها و تبدیل صفرها به ۱ها)

**ب)** مکمل ۲: مکمل ۱ + ۱

زبان ماشین و اسمبلی

**مثال:** مکمل ۲ عدد دودویی  $(10011101)_2$  را بدست آورید .

10011101  $\longrightarrow$  01100011

10011101  $\longrightarrow$  01100010 + 1 : مکمل ۱

01100010

+ 1

01100011

**نکته (راه سریع بدست آوردن مکمل ۲):** اولین صفرها و اولین یک کم ارزش ثابت و الباقی، تبدیل صفرها به یک ها و یک ها به صفرها

1111 → 0001  
1101100 → 0010100  
101101 → 010011

## برای نمایش یک مقدار منفی در کامپیوتر بایستی مراحل زیر طی نمود.

- ۱- ابتدا عدد را بدون علامت تصور نموده آن را به سیستم دودویی تبدیل نمایید.
- ۲- اگر تعداد ارقام کمتر از ۸ رقم باشد بایستی آنقدر رقم ۰ در سمت چپ اضافه کنیم تا ارقام ۸ رقم گردد و اگر تعداد ارقام بیشتر از ۸ رقم باشد بایستی آنقدر رقم ۰ در سمت چپ اضافه کنیم تا ارقام ۱۶ رقم گردد.
- ۳- سپس نتیجه حاصله را مکمل ۲ می گیریم .

**مثال:** عدد ۲۶ - را به سیستم دودویی تبدیل نمایید؟

$$26 \longrightarrow 11010$$

۱- ابتدا عدد ۲۶ را به سیستم دودویی تبدیل می نمایم.

۲- نتیجه بدست آمده را هشت رقمی می نمایم.

00011010

۳- سپس نتیجه حاصله را مکمل ۲ می گیریم.

11100101

0 ها را به 1 و 1 ها را به 0 تبدیل میکنیم.

نتیجه بدست آمده را با 1 جمع می نمایم.

11100101+

1

---

11100110

عدد 11100110 در سیستم دودویی نمایش 26- می باشد که یک بایت اشغال می نماید.

**مثال:** عدد ۳۵ - را به سیستم دودویی تبدیل نمایید؟

$$35 \longrightarrow 100011$$

۱- ابتدا عدد ۳۵ را به سیستم دودویی تبدیل می‌نماییم.

00100011

۲- نتیجه بدست آمده را هشت رقمی می‌نماییم.

۳- سپس نتیجه حاصله را مکمل ۲ می‌گیریم.

11011100

0 ها را به 1 و 1 ها را به 0 تبدیل می‌کنیم.

1101100 +

1

---

11011101

نتیجه بدست آمده را با 1 جمع می‌نماییم.

عدد 11011101 در سیستم دودویی معادل 35- می‌باشد که یک بایت اشغال می‌نماید.



**مثال:** عدد ۲۰ - در کامپیوتر به چه صورتی نمایش داده می شود؟

$$20 \rightarrow 10100$$

۱- ابتداء عدد را بدون در نظر گرفتن علامت به دودویی تبدیل می کنیم .

۲- نتیجه بدست آمده را هشت رقمی می نمایم. 00010100

۳- سپس نتیجه حاصله را مکمل ۲ می گیریم .

$$11101011$$

0 ها را به 1 و 1 ها را به 0 تبدیل میکنیم.

$$11101011+1=11101100$$

نتیجه بدست آمده را با 1 جمع می نمایم.

عدد 11101100 در سیستم دودویی معادل 20- می باشد که یک بایت اشغال می نماید.

**مثال:** عمل زیر را با استفاده از روش مکمل ۲ انجام دهید.

$$\begin{array}{r} 27 \\ - \\ 20 \\ \hline \end{array}$$

نکته: این عمل تفریق در حقیقت بمنزله جمع دو مقدار زیر می باشد.  $27+(-20)$  پس ابتدا مقادیر 20- و 27 را به سیستم دودویی تبدیل نموده.

با تقسیمات متوالی

$$27 \longrightarrow 11011$$

با استفاده از روش اسلایدهای قبل

$$-20 \longrightarrow 11101100$$

حال دو مقدار 20- و 27 را در سیستم دودویی با هم جمع می کنیم.

$$\begin{array}{r} 27 \\ - \\ 20 \\ \hline \end{array} \quad \longrightarrow \quad \begin{array}{r} 27 \\ + \\ -20 \\ \hline \end{array} \quad \longrightarrow \quad \begin{array}{r} 00011011 \\ + \\ 11101100 \\ \hline 100000111 \end{array} \longrightarrow \boxed{000000111}$$

با توجه به آنکه نتیجه جمع دو بایت بصورت یک بایت می باشد بیت 1 سمت چپ بایستی حذف گردد، نتیجه می شود 111 که برابر با 7 می باشد.

**مثال:** عمل زیر را با استفاده از روش مکمل ۲ انجام دهید.

$$\begin{array}{r} 534 \\ - \\ 281 \\ \hline \end{array}$$

نکته: این عمل تفریق در حقیقت بمنزله جمع دو مقدار زیر می باشد.  $534 + (-281)$  پس ابتدا مقادیر 281- و 534 را به سیستم دودویی تبدیل نموده.

با تقسیمات متوالی

$$534 \longrightarrow 1000010110$$

با استفاده از روش اسلایدهای قبل

$$-281 \longrightarrow 1111111011100111$$

حال دو مقدار 281- و 534 را در سیستم دودویی با هم جمع می کنیم.

$$\begin{array}{r} 534 \\ - \\ 281 \\ \hline \end{array} \quad \longrightarrow \quad \begin{array}{r} 534 \\ + \\ -281 \\ \hline \end{array} \quad \longrightarrow \quad \begin{array}{r} 000001000010110 \\ + \\ 1111111011100110 \\ \hline 1000000011111101 \end{array} \quad \longrightarrow \quad \boxed{000000011111101}$$

با توجه به آنکه نتیجه جمع دو کلمه بصورت یک کلمه می باشد بیت 1 سمت چپ بایستی حذف گردد، نتیجه می شود 11111101 که برابر با 253 می باشد.