

اصول اساسی برنامه نویسی به زبان اسمبلی

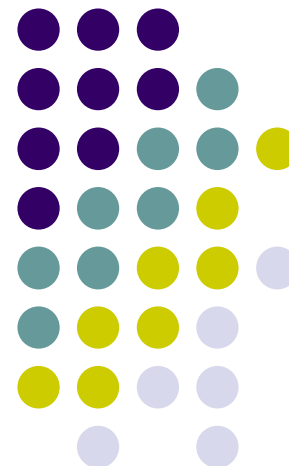
Richard C. Detmer :

: هاشمی اصل - مشحون

: زبانهای ماشین و برنامه سازی سیستم (رشته مهندسی کامپیوتر)

: 3 واحد

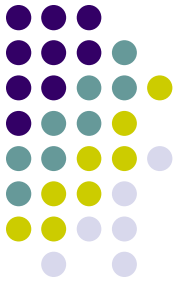
: مهندس داریوش نیک مهر و کاوه حقیقی



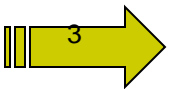
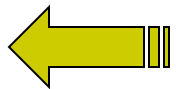


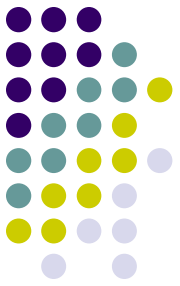
فهرست مطالب

- فصل اول : _____
- فصل دوم : _____
- فصل سوم : _____
- فصل چهارم : _____
- فصل پنجم : _____
- فصل ششم : _____
- فصل هفتم : _____
- فصل هشتم : _____
- فصل نهم : _____
- فصل دهم : _____ /
- فصل یازدهم : _____
- فصل دوازدهم : _____
- فصل سیزدهم : _____



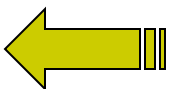
فصل اول

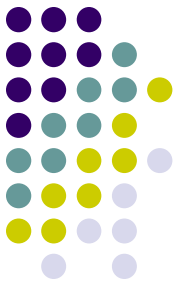




فهرست مطالب فصل اول

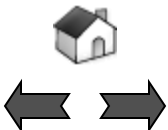
- نمایش داده ها در کامپیوتر
- اعداد دودویی و شانزده شانزدهی
- کدهای کارکتری
- نمایش مکمل 2 برای اعداد صحیح علامت دار

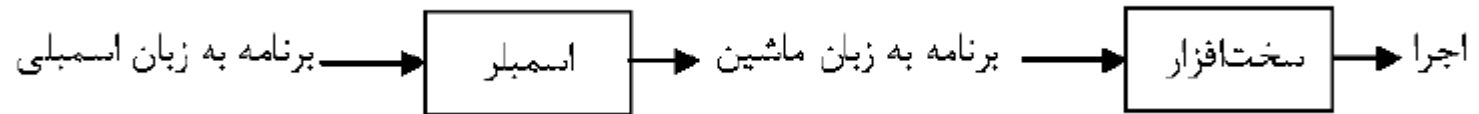
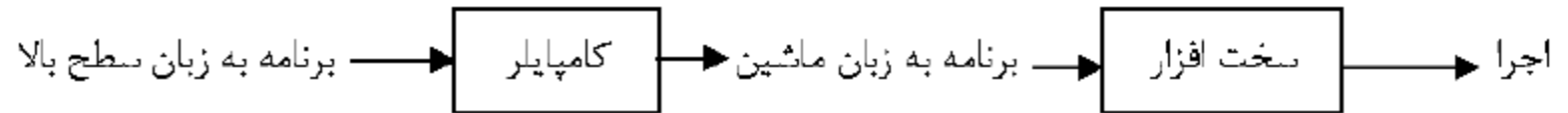
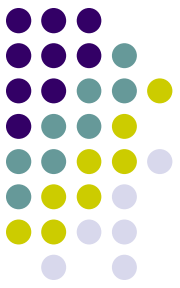


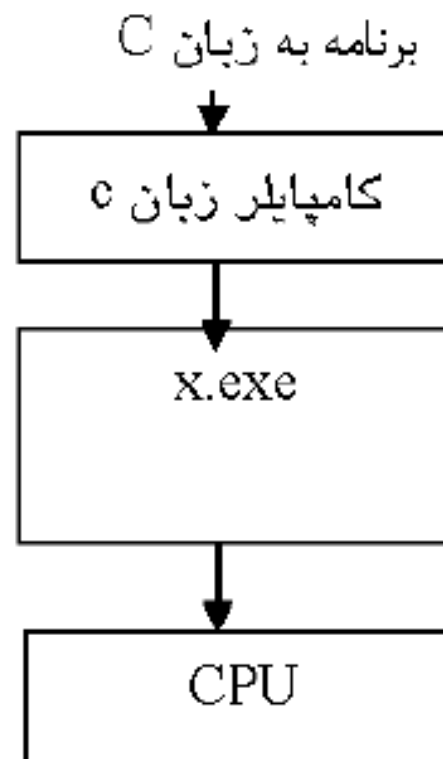
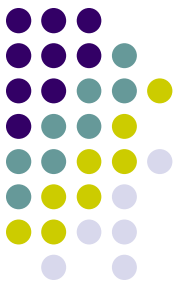


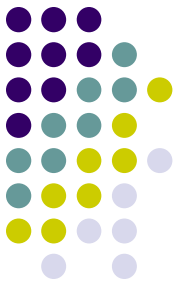
نمایش داده ها در کامپیوتر

در زبانهای سطح بالا نگران اینکه داده ها در کامپیوتر چگونه نمایش داده میشوند نیستیم ولی در زبان های اسمبلی بایستی بفکر چگونگی ذخیره داده باشیم و اغلب با کار تبدیل داده ها از یک نوع به نوع دیگر مواجه می باشیم.





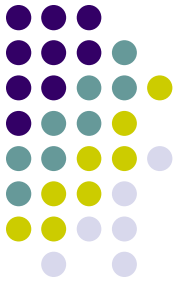




اعداد دودویی و شانزده شانزدهی

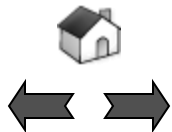
حافظه های کامپیوتر فقط می تواند ارقام 0 یا یک را در خود ذخیره نماید که به آنها بیت گفته میشود. در سیستم دودویی اعداد از بیت ها تشکیل شده اند.





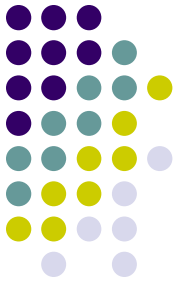
اعداد دودویی و شانزده شانزدهی

اعداد شانزدهی از ارقام ۰ تا ۱۵ تشکیل شده اند. برای راحتی، ارقام 10 تا ۱۵ را A تا F نشان داده می شود.





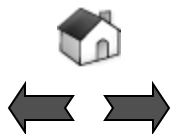
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

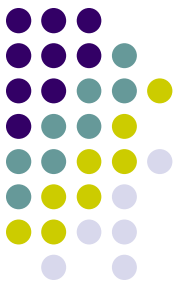


مثال :

1011

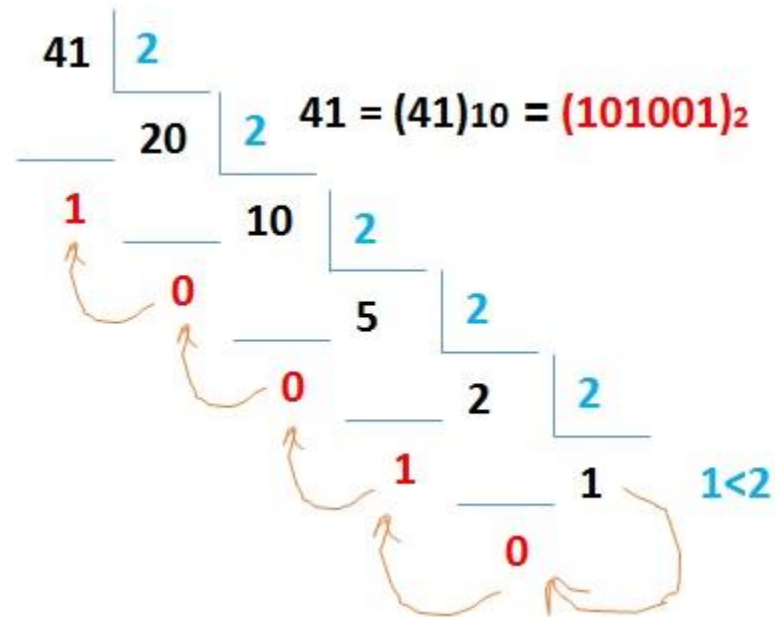
سیستم دودوئی شبیه سیستم دهدهی است با این تفاوت که ارقام از سمت راست به چپ به جای ارزش 1 ، 100 ، 1000 ، ارزش 1 ، 2 ، 4 ، 8 ، دارند. بنابراین 1011 در سیستم دودوئی معادل 11 می باشد.

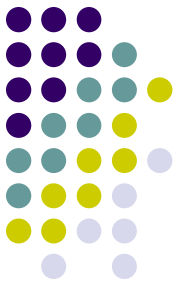




$(11100110)_2 = (?)_{10}$

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



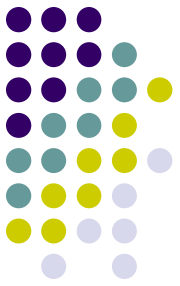


تبدیل اعداد شانزدهی به دودویی

هر رقم در سیستم شانزدهی بوسیله چهار رقم در سیستم دودویی قابل نمایش می باشد.

در اسلاید بعد مثالی آورده شده است.





مثال :

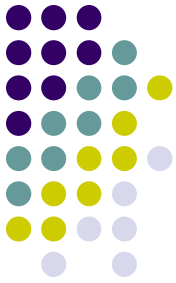
● 0100 معادل 4

● 1110 معادل E

برای تبدیل اعداد شانزدهی به دودوئی کافی است که به جای هر رقم، چهار بیت معادل آن قرار داد.

در اسلاید بعد مثالی آورده شده است.



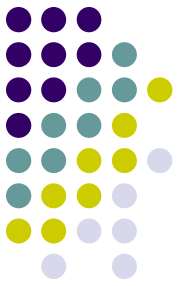


مثال :

برای تبدیل اعداد شانزدهی به دودویی کافی است که به جای هر رقم، چهار بیت معادل آن قرار داد.

2AD5 معادل 0010101011010101 در سیستم دودویی می باشد.



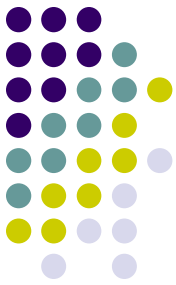


تبدیل اعداد دودوئی به شانزدهی

برای تبدیل اعداد دودوئی به شانزدهی، ارقام عدد داده شده را از سمت راست به ترتیب به صورت گروههای چهار بیتی درآورده آنگاه معادل هر گروه در سیستم شانزدهی را جایگزین می نماییم.

در اسلاید بعد مثالی آورده شده است.





مثال :

عدد **1010011101** در سیستم دودوئی در نظر بگیرید.

0010

2

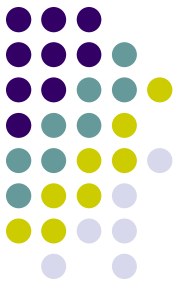
1001

9

1101

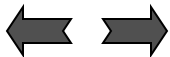
D

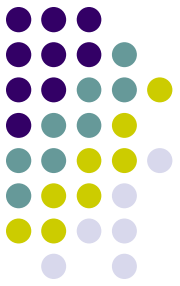




کدهای کرکتری

در کامپیوتر به حروف، ارقام، علامت ها کرکتر گفته می شود. به هر کرکتر یک کد هشت بیتی منحصر به فرد وابسته می شود که آنرا کد **ASCII** می نامند.





کرکترهای قابل چاپ

کرکتر	کد اسکی
0 تا 9	48 تا 57
A تا Z	65 تا 90
a تا z	97 تا 122



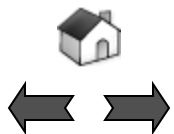


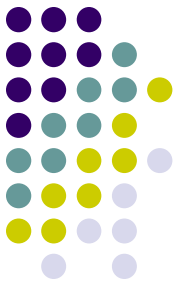
نکته :

. 126 32



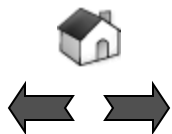
. 31 0

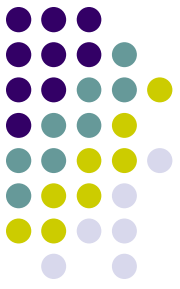




کرکترهای کنترلی

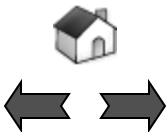
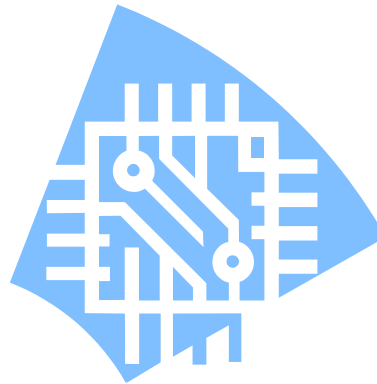
کرکتر	کد اسکی
ESC	27
CR	10
LF	13

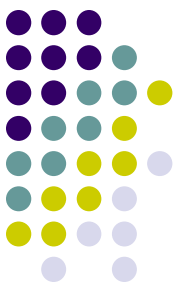




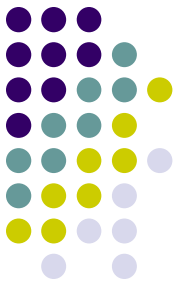
نمایش مکمل 2 برای اعداد صحیح علامتدار

اعداد منفی در کامپیوتر بصورت مکمل ۲ نمایش داده می شوند. وقتی یک عدد به شکل مکمل دو نشان داده می شود تعداد بیت های مورد استفاده باید از قبل مشخص گردد (32 , 16 , 8).





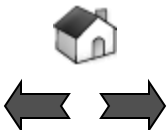
یک بایتی			
Max	Min	روش نمایش	نوع عدد
1111 1111 2^8-1 255	0000 0000 0	بدون علامت	بدون علامت
0111 1111 $+(2^7-1)$ +127	1111 1111 $-(2^7-1)$ -127	بیت علامت	علامتدار
0111 1111 $+(2^7-1)$ +127	1000 0000 $-(2^7-1)$ -127	مکمل ۱	
0111 1111 $+(2^7-1)$ +127	1000 0000 -2^7 -128	مکمل ۲	

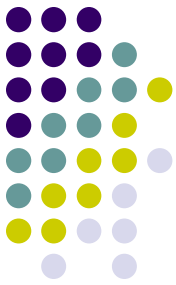


روش محاسبه مکمل 2 یک عدد :

- عدد را بصورت دودوئی درآورده.
- آنرا به تعداد بیت های مشخص شده تبدیل نموده .
- سپس صفر ها را 1 و 1 ها را به صفر تبدیل نموده .
- نتیجه را با یک جمع می نماییم.

در اسلاید بعد مثالی آورده شده است.





مثال :

-25

• 11001 معادل 25

• 00011001 هشت بیتی نموده

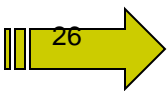
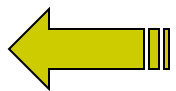
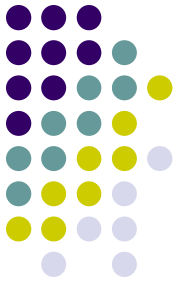
• 11100110 صفرها را به یک و یک ها را به صفر تبدیل نموده

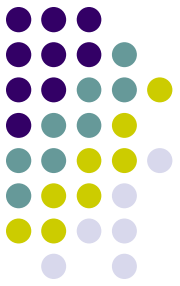
• 11100111 نتیجه را با یک جمع نموده

-25

11100111

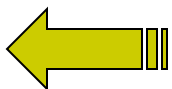


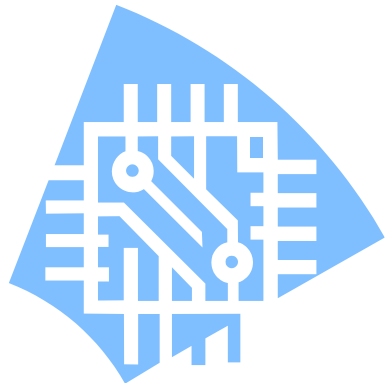




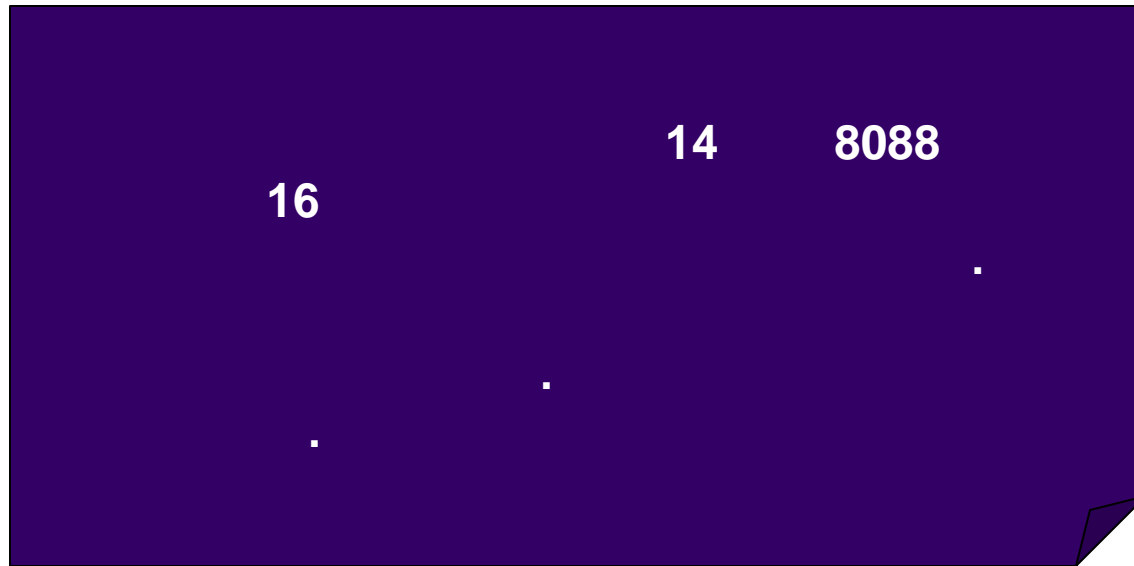
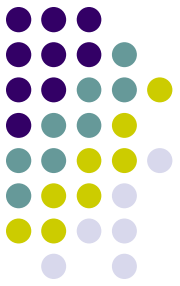
فهرست مطالب فصل دوم

- حافظه اصلی
- واحد پردازش مرکزی
- اسامی و اهداف ثبات ها





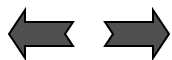
واحد پردازش مرکزی (CPU)



16

14

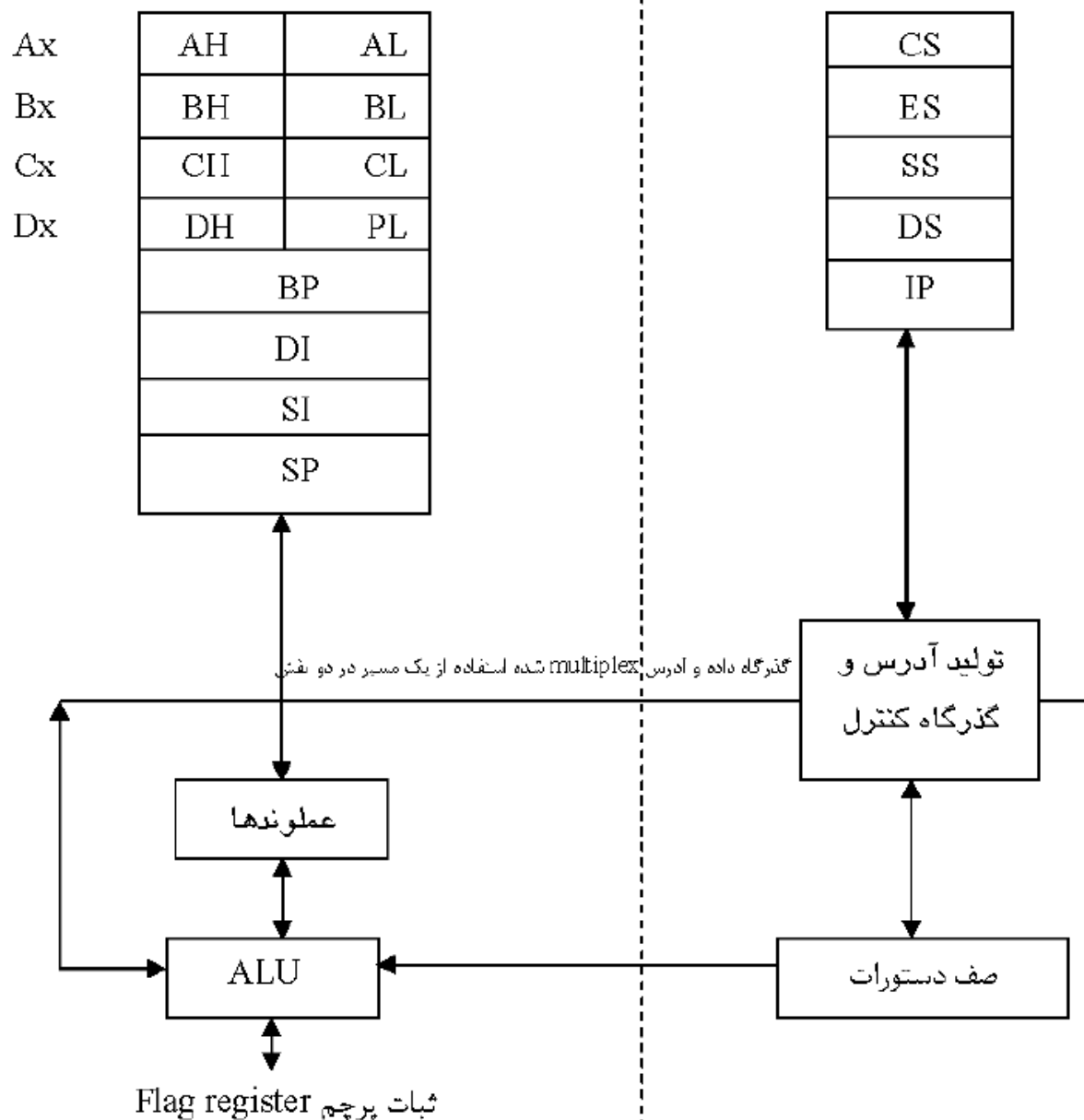
8088

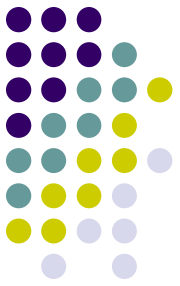




۱- واحد اتصال گذرگاه (BIU): decode و fetch

۲- واحد اجرا (EU)

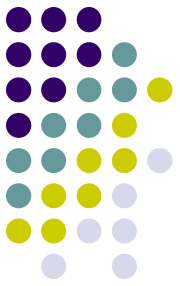




اسامی و اهداف ثبات ها

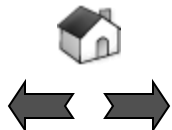
- **AX** اکومولاتور، کاربرد همگانی، بایت بالایی برابر **AH** و بایت پایینی برابر **AL**
- **BX** کاربرد همگانی، بایت بالایی برابر **BH** و بایت پایینی برابر **BL**
- **CX** کاربرد همگانی، بایت بالایی برابر **CH** و بایت پایینی برابر **CL**

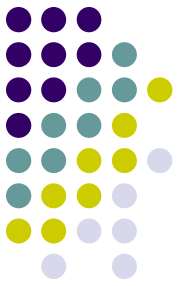




اسامی و اهداف ثبات ها

- **DX** کاربرد همگانی، بایت بالایی برابر **DH** و بایت پایینی برابر **DL**
- **CS** شماره سگمنت حافظه ای می باشد که دستورالعمل های اجرایی جاری در آنجا قرار دارد.
- **DS** سگمنت داده ها را می دهد.
- **ES** سگمنت فوق العاده را می دهد.

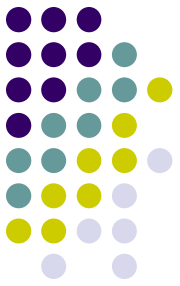




اسامی و اهداف ثبات ها

- **SS** سگمنت پشته را می دهد.
- **SP** اشاره گر پشته، افست بالای پشته در سگمنت پشته.
- **BP** اشاره گر مبنا، افست نقطه مراجعه (reference Point) در سگمنت پشته.

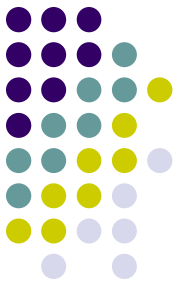




اسامی و اهداف ثبات ها

- **SI** اندیس منبع، افست رشته کاراکتری منبع در انتقال رشته های کاراکتری.
- **DI** اندیس مقصد: افست رشته کاراکتری مقصد.
- **IP** اشاره گر دستور العمل ها، افست دستور العمل بعدی در سگمنت کد برای دستیابی ثبات نشانه ها مجموعه ای از نشانه ها یا بیت های وضعیت.





ثبات نشانه

بعضی از ۱۶ بیت این ثبات برای نشان دادن نتیجه اجرای دستور عملها بوسیله دستور العمل های مختلف تغییر پیدا می کنند. هر کدام از این بیت ها را یک بیت وضعیت یا نشانه می گویند. اسامی برخی از این بیت ها عبارتند از :

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

.





OF (Overflow Flag) : اگر پس از انجام یک دستور بر روی اعداد علامت دار سر ریز رخ دهد این بیت برابر 1 می شود. (دو عدد یک بیتی با هم جمع شوند نتیجه در یک بایت جا نگیرد).

DF (Direction Flag) : اگر 1 باشد عملیات مربوط به رشته ها از راست به چپ و اگر صفر باشد از چپ به راست انجام می شود. (مثلاً برای کپی کردن).

IF (Interrupt Flag) : اگر 1 باشد « وقفه ها » فعال و اگر 0 باشد غیرفعال می شود.

TF (Trap Flag) : اگر 1 باشد برنامه دستور به دستور اجرا می شود. (برنامه کامل اجرا می شود).

SF (Sign Flag) : علامت نتیجه محاسبه شده در دستور قبل را نشان می دهد. صفر: مثبت یک: منفی

ZF (Zero Flag) : - صفر بودن یا نبودن نتیجه محاسبه شده در دستور قبل را نشان می دهد (هرگاه 1 بشود یعنی نتیجه صفر بوده).

AF (Auxiliary Flag) : اگر 1 باشد از بیت 3 به 4 رقم ثقلی داشته ایم .

PF (Parity Flag) : برای تست خطای انتقال داده ها به کار می رود اگر عدد حاصل تعداد زوج 1 داشته باشد PF نیز 1 می شود.

CF (Carry Flag) : اگر پس از انجام یک عمل بر روی اعداد بدون علامت سر زیر رخ دهد این بیت 1 می شود.