

## سوالات تحلیلی :

1:

در پردازنده 8086، رجیستر ها معمولا به 4 بخش تقسیم بندی می شوند :

1\_ رجیستر های عمومی : (AX, BX, CX, DX) که هر کدام به دو بخش L, H دارد که این بخش ها 8 بیتی هستند (رجیستر های 16 بیتی که هر کدام به دو بخش هشت بیتی تقسیم شده اند)، از این رجیستر ها برای ذخیره داده های موقت استفاده می شود.

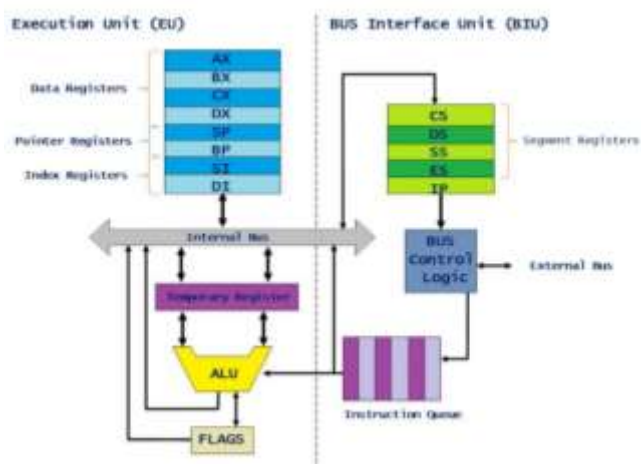
2\_ رجیستر های Segment: 4 تا از این نوع رجیستر ها در پردازنده داریم : CS (برای code segment)، DS (برای data segment)، SS (برای stack segment)، ES (برای extra segment)، هر کدام از این رجیستر ها 16 بیت هستند که کد و دستور ها در این سگمنت ها ذخیره می شوند.

3\_ رجیستر های index, pointer, pointers: هر همیشه آدرس یا مکان حافظه را ذخیره می کنند، در 8086 آنها معمولا آفستی را ذخیره می کنند که به کمک آن آدرس محاسبه می شود. (رجیستر IP: instruction pointer: معمولا آدرس دستور بعدی که قرار اسن اجرا شود را ذخیره می کند همچنین به عنوان یک آفست برای رجیستر CS نیز عمل می کند. رجیستر BP: base pointer: آدرس پایه حافظه را ذخیره می کند، همچنین به عنوان یک آفست برای SS عمل می کند. رجیستر SP: stack pointer: به مقدار بالایی stack اشاره می کند همچنین به عنوان آفست برای BP عمل می کند، رجیستر SI: source index: آفست آدرس Source را ذخیره میکند، رجیستر DI: destination index: آفست آدرس destination را ذخیره می کند.

4\_ رجیستر flag یا status: یک رجیستر 16 بیتی است که شامل 9 Flag است و مابقی بیت ها در آن بیکارند، این flag ها وضعیت پردازنده را بعد از هر عملیات حسابی یا منطقی نشان می دهد، اگر مقدار Flag یک باشد، flag، set می شود و اگر صفر باشد، reset می شود.

2:

ما به طور مستقیم نمی توانیم segment register ها را مقداردهی کنیم و برای انتقال باید از یک general purpose register استفاده کنیم (همانطور که در شکل میبینیم راه مستقیمی برای مقدار دهی segment register ها در طراحی میکرو پروسسور ها وجود ندارد، دلیل آن نیز می تواند کمتر مورد استفاده قرار گرفتن آن ها باشد و اینکه معماری آ» ساده تر باشد).



### 3:

انواع آدرسدهی در پردازنده 8086 :

1\_register mode :

هر دو عملوند رجیستر هستند (مثال: mov AX,BX )

2\_immediate mode :

عملوند منبع دیتای 8 یا 16 بیتی است ، عملوند مقصد نمی تواند داده ی immediate باشد(باید رجیستر باشد)(مثال :

( mov AX,1000h

3\_direct mode :

در این آدرسدهی ، آدرس effective مستقیماً در instruction به عنوان displacement داده می شود (مثال :

( mov AX,[0500]

4\_register indirect mode :

در این حالت ، آدرس effective در SI ، DI یا BX است(آدرس فیزیکی = آدرس Segment + آدرس effective )

(مثال : ( mov AX,[DI]

5\_based relative addressing mode :

از رجیسترهای پایه ( BX یا BP برای محاسبه effective address استفاده می شود)(بخش های پیش فرض برای محاسبه ی آدرس فیزیکی ds برای bx و ss برای Bp است.)

(مثال : ( mov AX,[BX]+20

6\_indexed relative addressing mode :

مانند حالت relative addressing mode کار می کند با این تفاوت که رجیسترهای DI و SI آدرس offset را نگهداری می کنند.

(مثال : ( mov AX,[SI] + 10

7\_based indexed addressing mode :

ترکیبی از حالت های 5 و 6 (مثال: ( mov AL,[BX][DI]+5

برنامه نویس می تواند control flag ها را مقدار دهی کند ، که رجیستر DF (direction flag) برای کنترل مسیر عملیات های رشته Set می شود ، ( IF (interrupt enable flag) که برای فعال یا غیر فعال کردن درخواست های وقفه قابل پوشش خارجی استفاده می شود ، TF (trap flag) : وقتی این Flag تنظیم می شود ، به برنامه اجازه می دهد که تک مرحله ای اجرا شود ، یعنی برای دیباگ شدن ، یک دستور را در یک زمان اجرا کند.