معالجات مراجعة عامة

أنماط عمل الذاكرة DRAM

- يعتبر زمن الوصول الكبير وزمن الإنعاش في ذواكر DRAM المشكلة الرئيسية في تصميم الحواسيب عالية السرعة ولحل هذه المشكلة تم إضافة ذواكر SRAMذات حجم صغير مع ذواكر DRAMذات الحجم الكبير.
- يعرف زمن الوصول للذاكرة Address access time(TAA) الزمن الفاصل بين لحظة تطبيق العناوين على أرجل شريحة الذاكرة وزمن توضع المعطيات على أرجل شريحة الذاكرة ، بينما نعرف زمن دورة الذاكرة بأنه الزمن الفاصل بين وصوليين متتالين الى شريحة الذاكرة . إن زمن دورة الذاكرة =زمن الوصول للذاكرة في ذواكر SRAM بينما لا يساويها في ذواكر DRAM.

النمط القياسي لعمل الذاكرة DRAM

- زمن دورة الذاكرة أطول من زمن الوصول لأنه يتطلب تزويد عنوان الصف ومن ثم تزويد عنوان العمود لكل خلية في DRAM ، ويمثل زمن الوصول TAA زمن تزويد عنوان الصف TRAC وكما ذكرنا سابقا فان زمن الوصول لا يساوي زمن الدورة للذاكرة لوجود زمن إعادة الشحن TRP وبالتالي فان TRC=TRAC+TRP
- أي أن زمن دورة القراءة =زمن الوصول الى الصف +زمن إعادة الشحن

مثال

- لو كان زمن الوصول للذاكرة DRAMيساوي 100 nsفإن زمن القراءة حوالي 190 ns , حيث 90 ns زمن إعادة الشحن
- إن الزمن 100 ns يكفي الوصول إلى موقع واحد في الذاكرة أو الوصول إلى أكثر من موقع تحتاج إلى 190 nsلكل وصول بسبب زمن إعادة الشحن المطلوب داخلياً لذاكرة DRAMللوصول إلى الخلية التالية (الموقع التالي).
- إن الاختلاف بين زمن الوصول و زمن القراءة هو أحد الاختلافات الرئيسية بين DRAM , SRAM

- لحل هذه المشكلة نستخدم طريقة التداخل interlearing لملاءمة الذاكرة DRAMو نستخدم في هذه الطريقة مجموعتين من البنوك المتواضعة بشكل متجاور و يصل المعالج إلى كل مجموعة بشكل متناوب و في هذه الطريقة يقابل زمن إعادة الشحن في إحدى المجموعتين زمن الوصول إلى المجموعة التالية.
- أي خلال وصول المعالج إلى أحد البنوك فإن البنك الأخر يقوم بإعادة الشحن

- معالج 386 تردده 20 MHZ فإن دورة الذاكرة
- =2*(1/20) فاستخدام ذاكرة DRAM زمن وصولها 70 ns و زمن إعادة الشحن 65 ns أي زمن الدورة الكلي = 135 nsزمن طويل نسبياً بالنسبة للزمن الذي يعطيه المعالج 100 ns و تحل هذه المشكلة باستخدام طريقة التداخل الذاكري .

- قارن زمن المعالج الأصغري المطلوب لقراءة 150 مؤقتاً عشوائياً لبنك ما في كلتا الحالتين:
  - DRAM with TACC=100 ns , TRC=190 ns
  - SRAM with TACC=100 ns
- الحل : تحتاج الذاكرة DRAM إلى 190 ns إلى كل موقع
- الزمن الكلي 190*150=28500 nsلولوج المعالج إلى جميع المواقع 150
- في حالة الذاكرة SRAM يحتاج المعالج لولوج تلك المواقع إلى 100*150=15000 nsلأن tACC=tRC
.

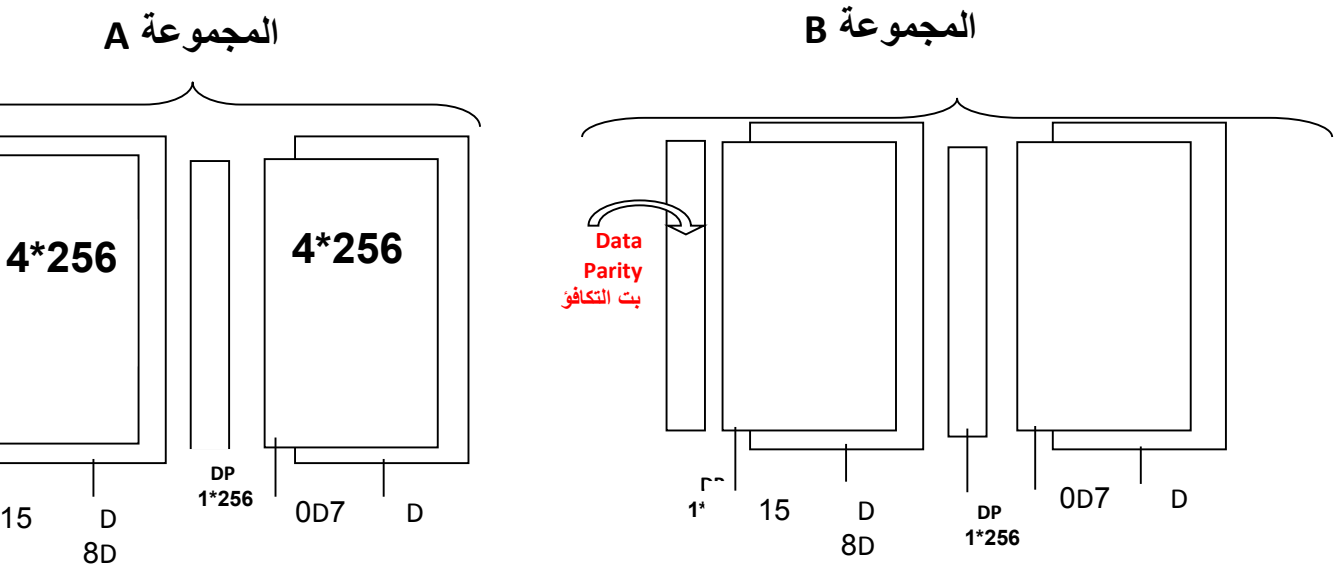
- **مثال1**: احسب زمن الوصول إلى 1024 بتاً عشوائياً لشريحة 1M*1 إذا كان TRC=165 ns , TRAC=85 ns
- **الحل:**
- من أجل النمط القياسي (العشوائي) لقراءة *1024 bit*
- *165*1024=168960 ns*
- **مثال2**:ما هو الزمن اللازم عند استخدام طريقة التداخل في المثال السابق
- الحل :*1024 * 85= 87040 ns*

مشكلة التداخل:

- المشكلة الرئيسية هي عند توسيع الذاكرة وفقاً لطريقة التداخل يجب إضافة مجموعتين من البنوك
- **مثال** *3*:بفرض أننا نستخدم ذاكرة DRAM (1M*1*)* طريقة التداخل بحيث أن كل
- مجموعة بحجم 4MB أوجد الشرائح و حجم الذاكرة الأصغري المضاف و عدد الشرائح .
- **الحل:** كل بنك يتعامل مع *8 bit* من المعطيات و بفرض *1M*9* لكل بنك فهناك 9 شرائح لكل بايت و هذا يعني أن عدد الشرائح *DRAM* الكلية هو *36* لكل مجموعة أو *72* شريحة *1M*1* الأول *8MB* من الذاكرة المتداخلة
-
- وإن إضافة أي ذاكرة بعد ذلك يجب أن يكون من مضاعفات 8 MB و بذلك نحتاج إلى 36 شريحة DRAM 1M*1 لزيادة ذاكرة النظام إلى 12M

- **مثال**:يمتلك الحاسوب 386 1M من الذاكرة DRAM باستخدام طريقة التصميم الداخلي بفرض أن الشرائح DRAM المستخدمة هو فقط 1*k 256 و 4*256k فما هو تنظيم الشريحة DRAM و ما هو عددها.
- الحل:باعتبار المعالج 386 يمتلك 16 bit معطيات فإنه يستخدم 512 KB لكل مجموعة A , B أو 4 بنوك من قياس 9*256K حيث يتألف كل مجموعة من بنكين من 9*256K . لذلك فإن عدد الشرائح الكلية هو 12 لأن كل بنك يستخدم 3 شرائح(شريحتان 4*256 و شريحة 1*256 لبت التكافؤ)

بت التكافؤ وظيفته فحص خطأ التكافؤ على لوحة النظام



- **المثال**:ما هو عدد الشرائح و حجم الذاكرة الأصغري المضاف للمثال السابق بفرض أن الشرائح DRAM المتاحة 256*1 و 256*4
- إن حجم الذاكرة المضاف الأصغري هو 1M باعتبار أنه لدينا بنكين لكل مجموعة من الذاكرة المتداخلة و هناك شريحتان 256*4 و شريحة 256*1 لبت التكافؤ و الذي يعني لدينا 3 شرائح لكل بنك و لهذا فإن أصغر ذاكرة مضافة تحتاج إلى 12 شريحة . 8 منها قياس 256*4 و 4 منها قياس 256*1 لبت التكافؤ و بالتالي فإن الناتج 1M

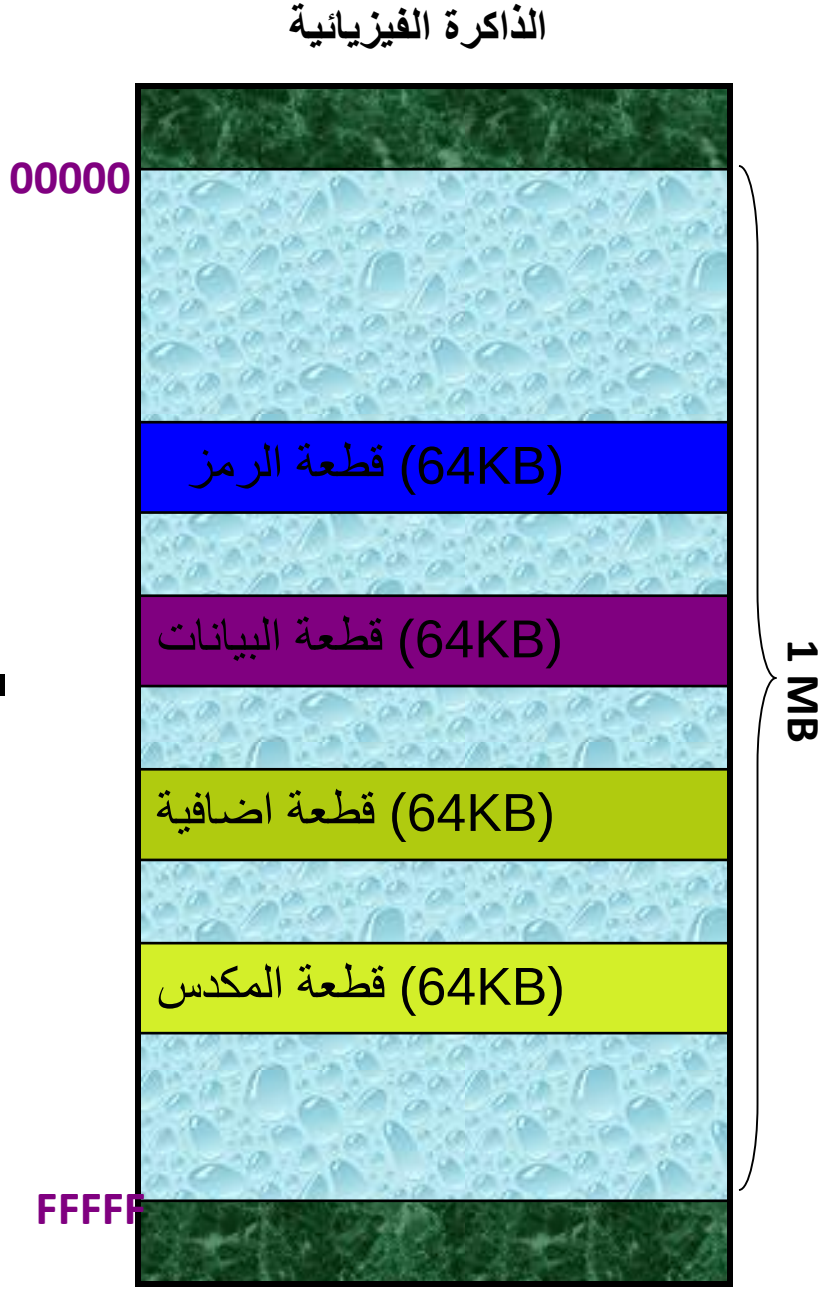
عمل الذاكرة DRAMفي نمط الصفحة

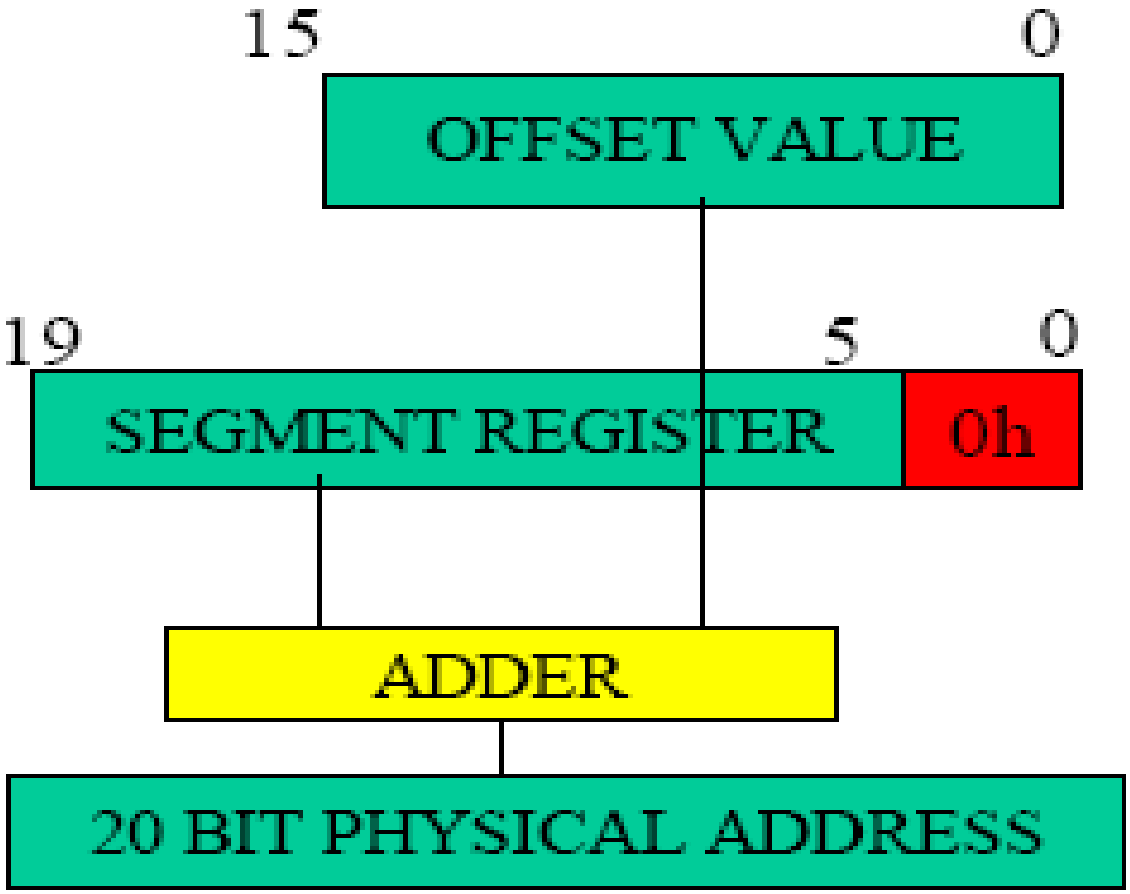
- نعني بنمط الصفحة في الذاكرة DRAM عدد خلايا الأعمدة في سطر ما ، وكما نعلم فان تنظيم خلايا التخزين في الذاكرة DRAM يكون وفق مصفوفة (N×N) Nعمود و N سطر ولقراءة خلية ما يطبق عنوان الصف أولا وتفعل اشارة RAS ومن ثم تطبق عنوان العمود وتفعل اشارة CAS وجاءت فكرة نمط الصفحة كون معظم حالات الوصول الى مواقع الذاكرة تتابعي ولا حاجة لتطبيق عنوان السطر والعمود لكل موقع كما هو الحال في النمط القياسي للذاكرة DRAM وبدلا من ذلك يطبق في نمط الصفحة عنوان الصف أولا ونمسك حالة RAS في عنوان الصف ومن ثم نطبق عنوان العمود ويتم تبديل اشارة CAS مرات عديدة وهذا يؤدي الى حفظها في عناوين الاعمدة حتى يتم الوصول الى العمود الأخيرمن الصفحة الحالية ومن ثم يطبق عنوان الصف التالي(الصفحة)وتتكرر العملية باستمرار وفي هذه الحالة يكون زمن الوصول الى الخلية الأولى هو زمن الوصول القياسي ويساوي زمن الوصول للصف والعمود TRAS اما زمن الوصول الى الخلية التالية فيكون أصغر بكثير وهكذا حتى الخلية الاخيرة من الصفحة أو الصفويشار الى هذا الزمن ب TCAC TIME
- OF COLUMM Access

الذاكرة المقطعة

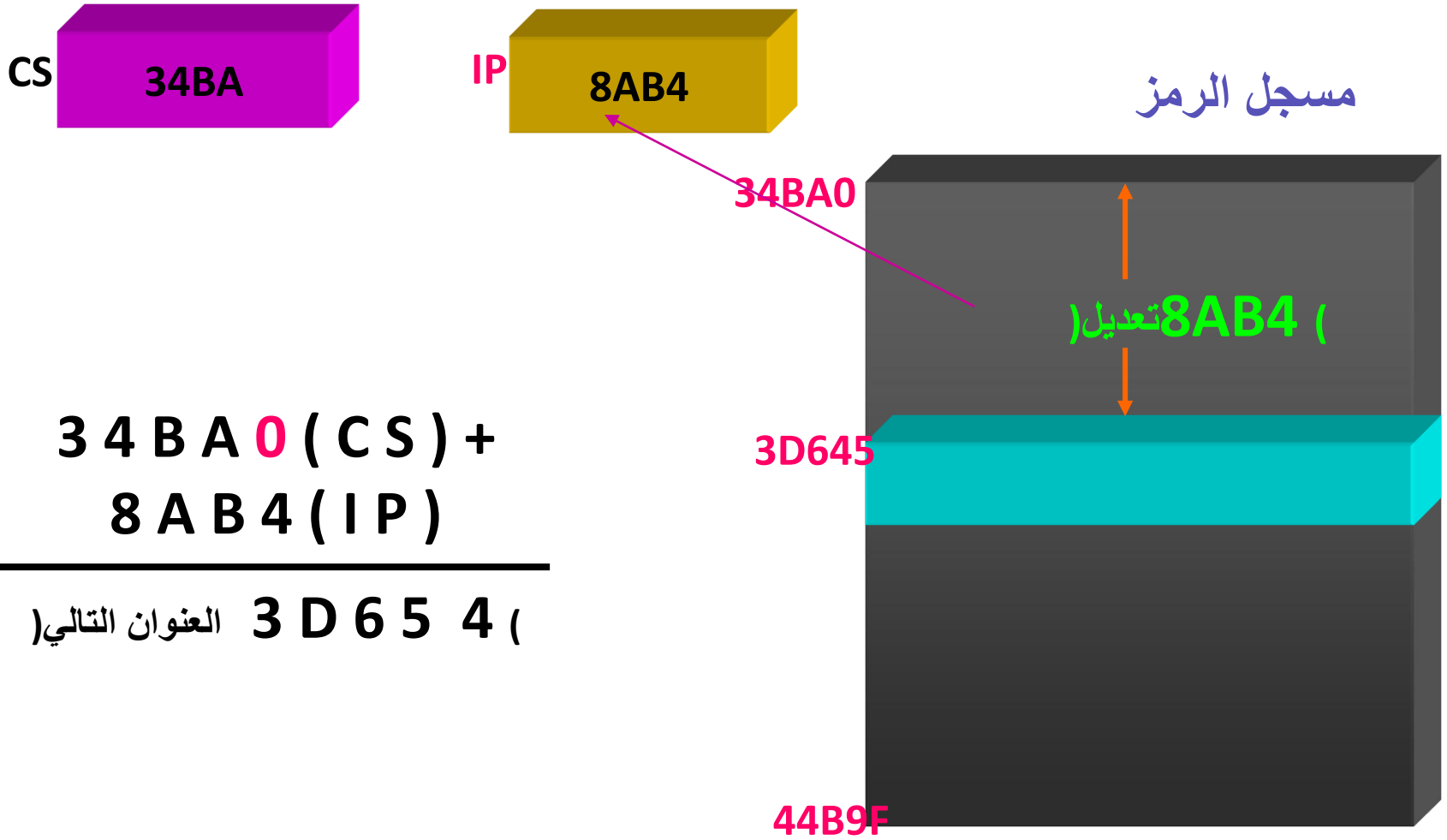
- الذاكرة في المعالج 8086 منظمة بشكل ذاكرة مقطعة
- المعالج 8086 قادر على عنونة 1 ميغا بايت من الذاكرة

- الذاكرة الفيزيائية المتوفرة يمكن أن تقسم إلى عدد من القطع المنطقية




15
0
OFFSET VALUE
19
5
0
SEGMENT REGISTER
0h
ADDER
20 BIT PHYSICAL ADDRESS

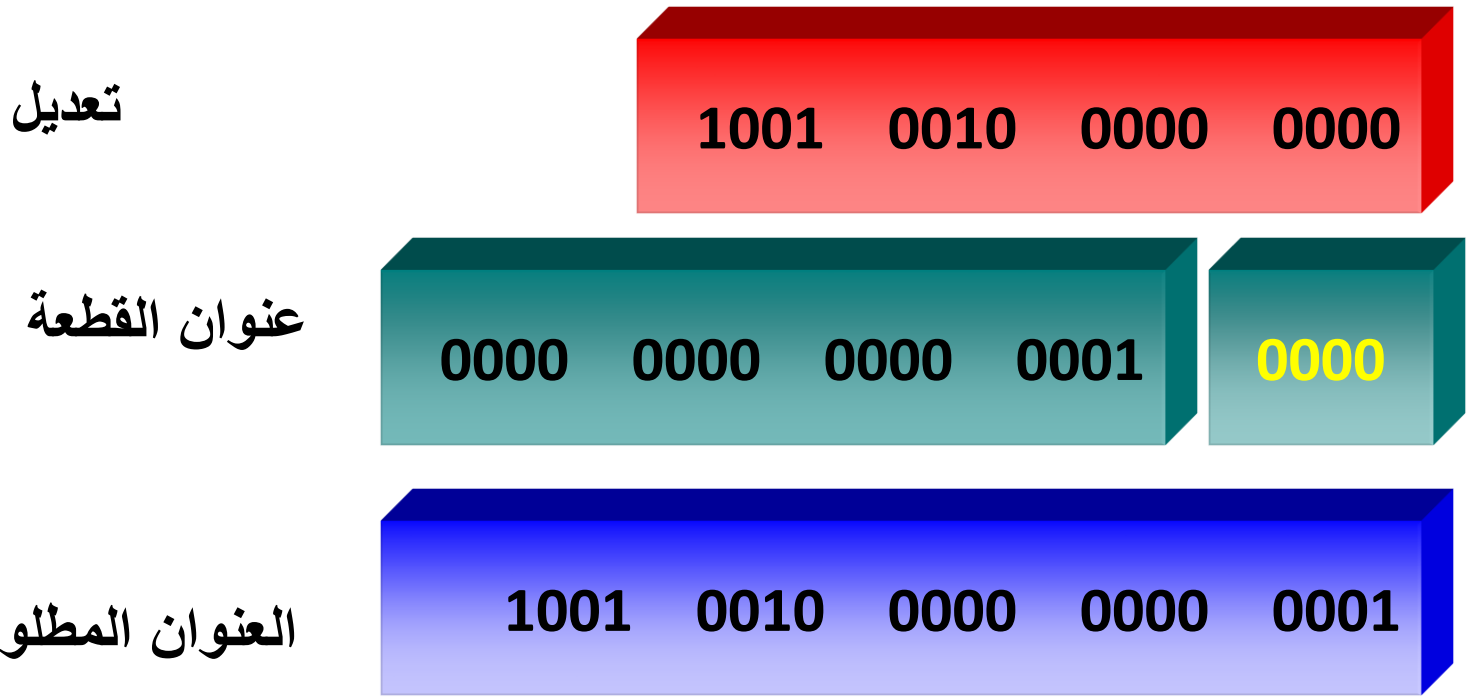
- المثال التالي يبين مخطط مسجل الرمز ومؤشر التعليمة لتشكيل العنوان



3 4 B A 0 (C S) +
8 A B 4 (I P)
---
3 D 6 5 4 (العنوان التالي)

• مثال عن حساب العنوان

- في حال بدأ مسجل المعطيات من الموقع 1000هـ ومرجع المعطيات يحتوي العنوان 29هـ , فأين تكون المعطيات الفعلية ؟



وحدة التنفيذ – مسجلات الأعلام

- 6 من الأعلام هي مؤشرات حالة تبين خصائص آخر تعليمة منطقية أو حسابية

على سبيل المثال , اذا كان المسجل AL=7Fh

ونفذت التعليمة :

فان التالي سوف يحدث :ADD AL , 1

AL = 80h

CF = 0;لايوجد أي منقول من البت السابع

PF = 0; له عدد فردي من الواحدات80h

AF = 1;يوجد منقول من البت الثالث إلى البت الرابع

ZF = 0;النتيجة لاتساوي الصفر

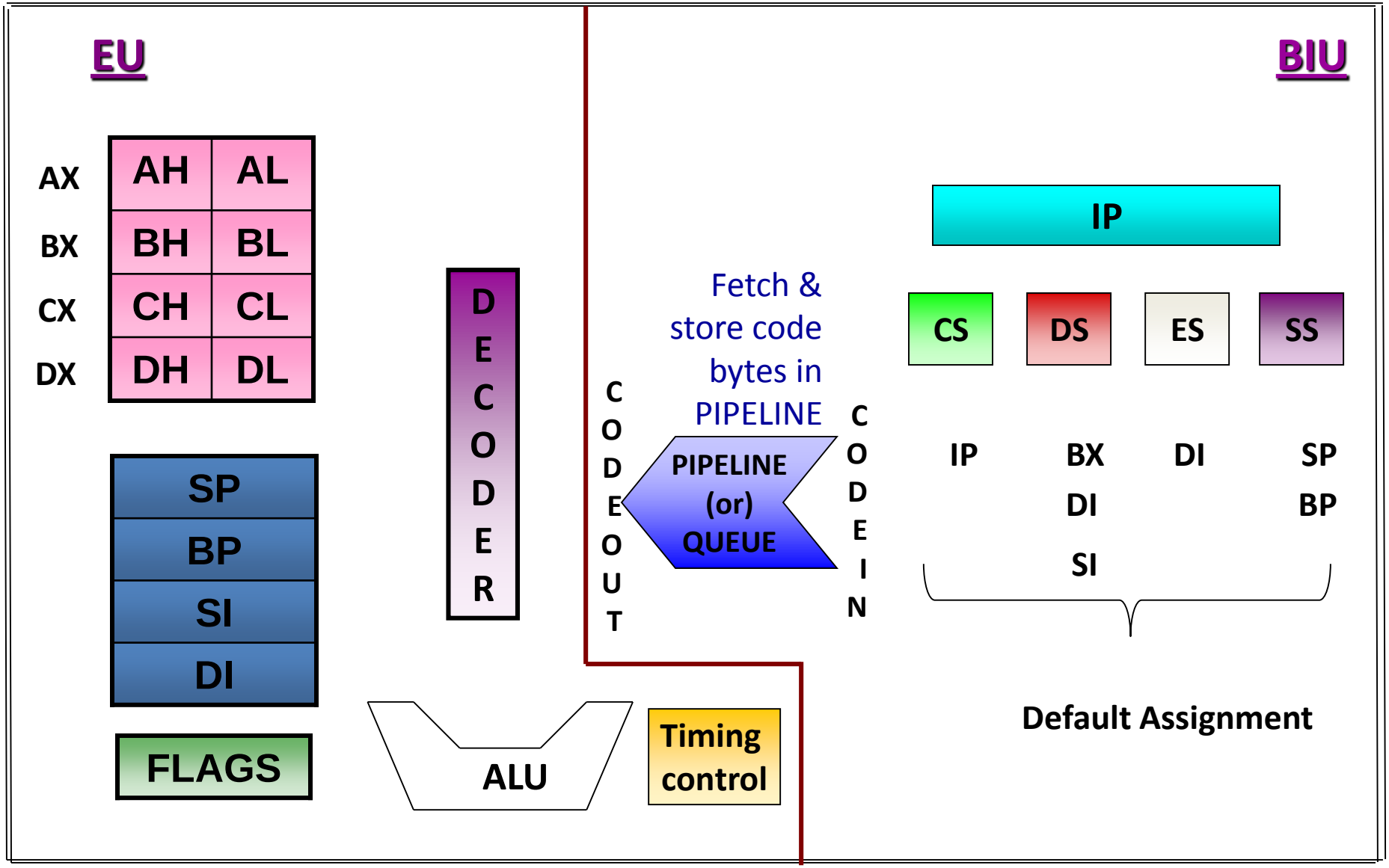
SF = 1;البت السابع يساوي الواحد

OF = 1;بت الإشارة قد تغير

مجموعة مسجلات العناوين والمقاطع :

- CS:IP
- SS:SP SS:BP
- DS:BX DS:SI
- DS:DI (من أجل عمليات أخرى للسلاسل)
- ES:DI (من أجل عمليات السلاسل)

ملخص عن المسجلات وخطوط المعطيات للمعالج الصغري :



- اذا تم تخزين اول 8 بتات في عنوان زوجي فان المعالج 8086 يستطيع قراءة الكلمة باكملها في العملية الواحدة
- على سبيل المثال إذا تم تخزين البيانات المؤلفة من 16 بت (H2607)في عنوان 00520
- MOV BX, [00520]
- 8086 يقرأ البايت الأول ويخزن البيانات في BL ثم يقرأ البايت الثاني ويخزن البيانات في BH.
- BL <-(00520)
- BH <- (00521)

- إذا تم تخزين البايت الأول من البيانات في عنوان فردي قانه يحتاج إلى عمليتين لقراءة البيانات المؤلفة من 16 بت.
- على سبيل المثال إذا تم تخزين البيانات المؤلفة من 16 بت(F520)في عنوان حتى H00521
- MOV BX, [00521]
- ***في العملية الأولى،*** 8086 يقرأ البيانات المؤلفة من 16 بت من الموقع
- 00520 ويخزن البيانات الموجودة في الموقع 00521 في سجل BL ويتجاهل البيانات في الموقع 00520.
- ***في العملية الثانية،*** 8086 يقرأ البيانات المؤلفة من 16 بت من الموقع 00522 موقع ويخزن البيانات الموجودة في الموقع 00522 في السجل BHويتجاهل البيانات الموجودة في الموقع00523

Example 3: if **CS** = 002AH, and **IP** = 0023H, write the **logical address** that they represent, then map it to **Physical address.**

Solution:

Logical address = **CS : IP**

002A : 0023

Physical address = (**CS** X 10H) + **IP** = 002A0 +0023 = 002C3

Example 4: if **CS** = 002BH, and **IP** = 0013H, write the **logical address** that they represent, then map it to **Physical address.**

Solution:

Logical address = **CS : IP**

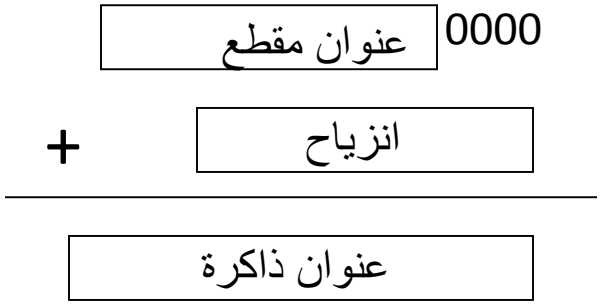
002B : 0013

Physical address = (**CS** X 10H) + **IP** = 002B0 +0013 = 002C3

Physical addresses are identical

حساب عنوان الذاكرة

- يجب أن تخزن عناوين المقاطع في مسجلات المقاطع .
- يتم توليد الانزياح من مسجلات المؤشرات و مسجل مؤشر التعليمة IP و القيم الفورية.



- أمثلة:

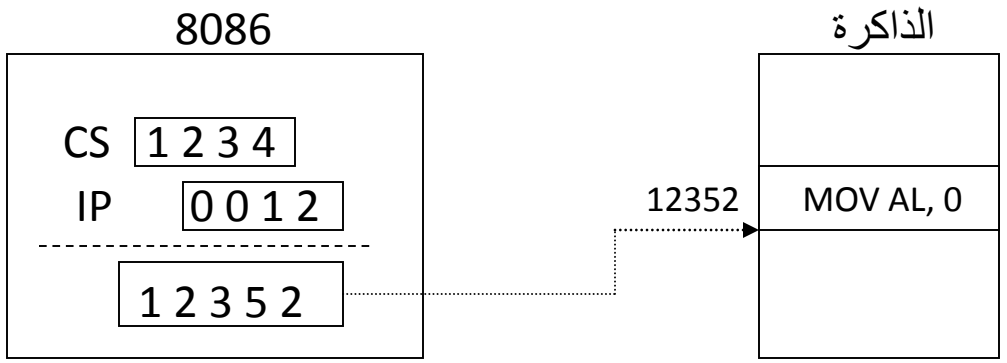
SS	5	0	0	0	0
SP +		F	F	E	0
عنوان المكدس	5	F	F	E	0

CS	3	4	8	A	0
IP +		4	2	1	4
عنوان التعليمة	3	8	A	B	4

DS	1	2	3	4	0
DI +		0	0	2	2
عنوان المعطيات	1	2	3	6	2

جلب التعليمات

❑من أين يتم جلب التعليمة التالية ؟



❑تحديث محتوى IP :

— بعد جلب أي تعليمة , يتم تحديث المسجل IP على النحو التالي :

IP = IP + طول التعليمة التي تم جلبها

— مثال :
ان طول التعليمة **MOV AL, 0** هو 2 بايت , لذا بعد جلب هذه التعليمة يتم تحديث IP لتصيح قيمته :
IP = 0014

الوصول الى ذاكرة المعطيات

- هناك عدة طرق لتوليد عنوان الذاكرة عند الوصول الى معطيات الذاكرة , تسمى هذه الطرق بأنماط العنونة .

- أمثلة :

— *عنونة مباشرة* **MOV AL, [0300H]:**

(DS=1234H بافتراض)

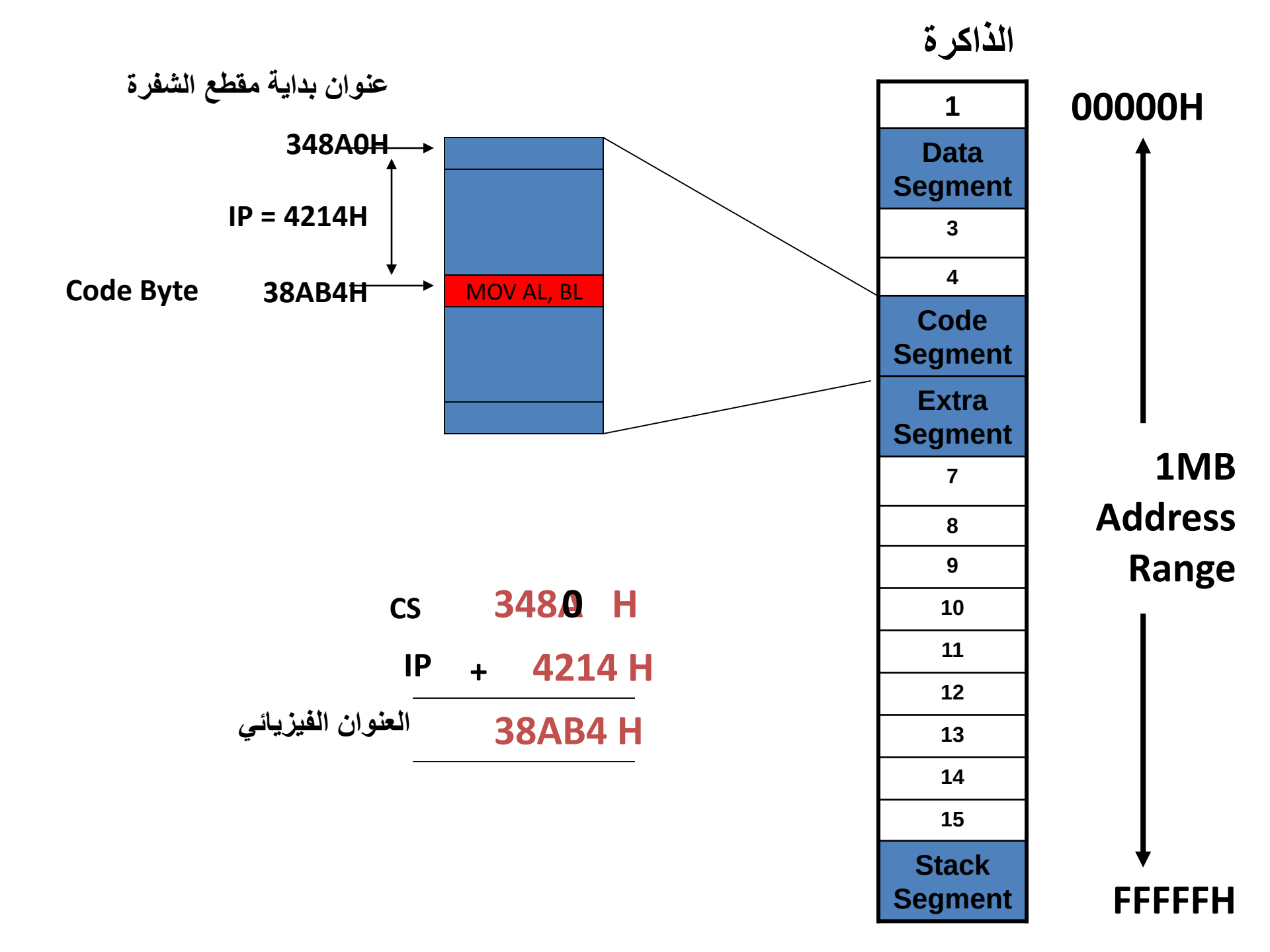
DS	1	2	3	4	0
		0	3	0	0
عنوان الذاكرة	1	2	6	4	0

— *عنونة غير مباشرة بالسجل* **MOV AL, [SI]:**

(DS=1234H بافتراض)

(SI=0310H بافتراض)

DS	1	2	3	4	0
		0	3	1	0
عنوان الذاكرة	1	2	6	5	0


الذاكرة
عنوان بداية مقطع الشفرة
348A0H
IP = 4214H
Code Byte
38AB4H
MOV AL, BL
1
Data Segment
3
4
Code Segment
Extra Segment
7
8
9
10
11
12
13
14
15
Stack Segment
00000H
1MB Address Range
FFFFFH
CS 348A0 H
IP + 4214 H
العنوان الفيزيائي 38AB4 H

تعليمات المعالج الصغري 8086
باستخدام لغة الاسمبلي

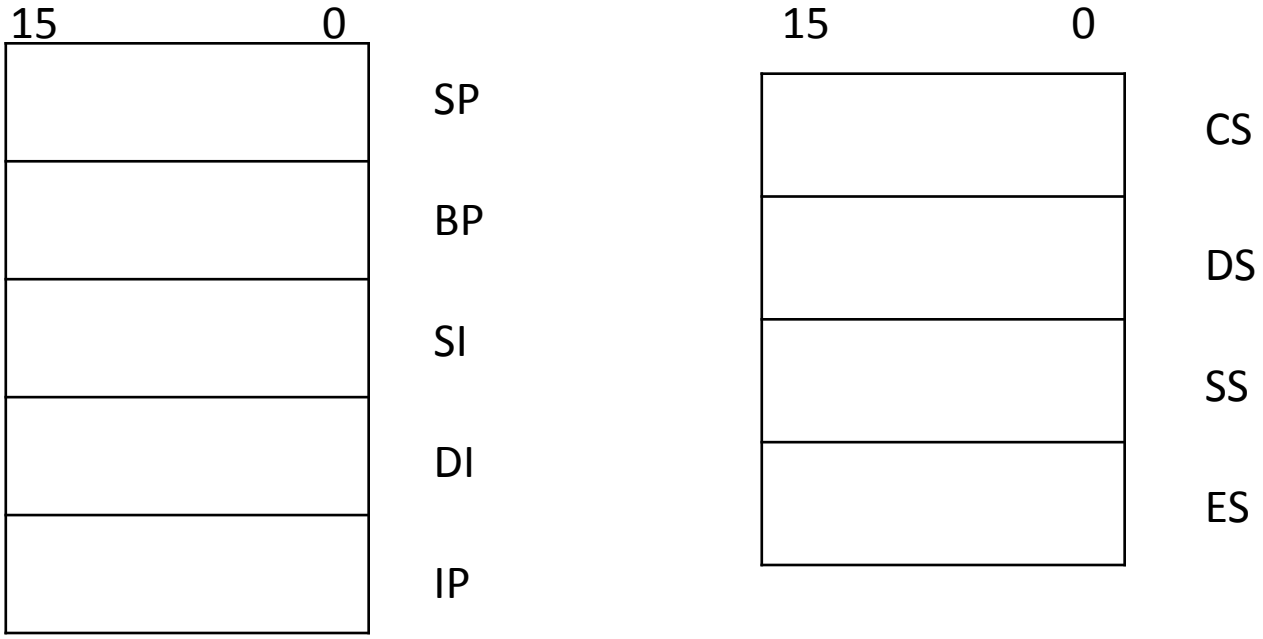
ميزات المعالج 8086

- يتعامل هذا المعالج مع مسجلات من طول 16 bit
- له 20 خط عنونة ويتعامل مع ذاكرة بحجم 1 MB
- له سجلات أغراض عامة عددها 14 بطول 16bit
- عرض مجال خطوط العنونة فيه 20 bit وعرض مجال خطوط المعطيات 16 bit

مسجلات الأغراض العامة

AH	AL	AX=AH+AL
BH	BL	BX=BH+BL
CH	CL	CX=CH+CL
DH	DL	DX=DH+DL

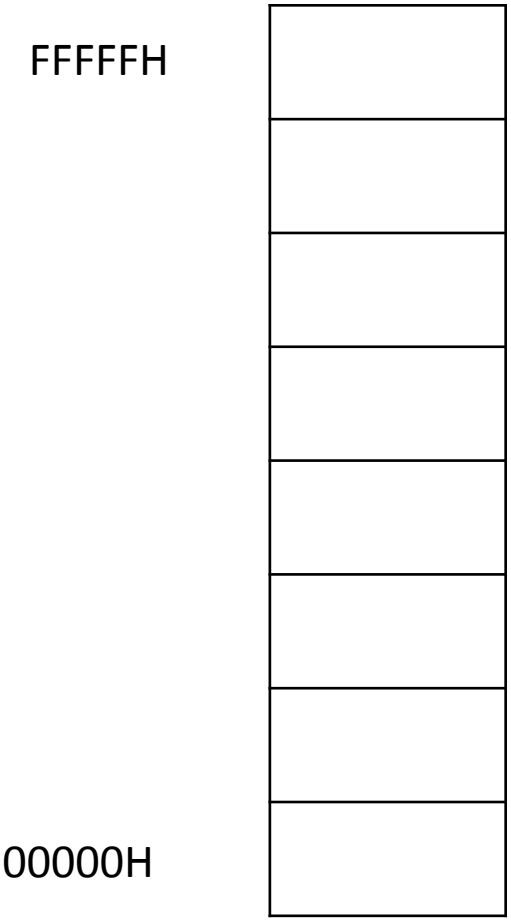
مسجلات الأغراض الخاصة



FLAG REGISTERS

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

PROGRAMM MEMORY



- المعالج 8086 يتعامل مع أنواع الكلمات :
- Byte (8bit)
- Word (16bit)
- Short integers (16bit)
- Integers (16bit)
- Double word (32bit)
- Long integers (32bit)
- Strings

- أنواع العنونة بالتعليمات:
- عنونة فورية
- عنونة باستخدام السجلات
- عنونة باستخدام الذاكرة
- عنونة باستخدام المنفذ I\O
- عنونة علائقية
- عنونة مضمنة

- 1- العنونة الفورية : يمكننا استخدام 8 أو 16 بت كجزأ من التعليمة

مثل : MVI AL,34H MVI SI,1212H

- 2- العنونة باستخدام السجلات :

مثل : MOV AL,CL MOV DS,SI

- 3- العنونة باستخدام الذاكرة : ولها عدة أنواع
  - أ- مباشرة : MOV AL,DS MOV DS,AL

MOV AX,DS

- ب- غير مباشرة : تكون العناوين المتأثرة بأحد زوج السجلين BX or BP او محتوى احد السجلين SI or DI

مثل :

MOV BX,[CX] ;DS=10000H CX=2000H

Physical address 12000H=30 12001H=40

BX=4030 BH=40H BL=30H

- ج- جمع المحتوى :

مثال : MOV CX,[BX+DI] ;DS=10000H

BX=2000H DI=0030H

Physical address 12030H=20 12031=10

CX=1020H CH=10 CL=20

- د- عنونة ذو علاقة بالمسجلات :

مثال:

MOV CX,[BX+0003H]

DS=10000H BX=1000H

base+dis=1003H displacement=3

11003H=30H 11004=40H

CX=4030H CH=40 CL=30

- ه- زيادة للمحتوى :

مثال:

MOV AL,[BX+SI+10H]

- و- السلاسل :

التعليمة التي تستخدم السلاسل تفترض تلقائياً أن SI عند اول بايت لتعليمة المصدر
و DI عند اول بايت لتعليمة الوجهة

مثال: MOVS BYTE

Initially : [DF]=0 [DS]=30000H
[SI]=0600H

[ES]=50000H [DI]=0400H

30600H=38H 50400H=45H

After execution :

50400h=38H [SI]=0601H [DI]=0401H

- 4- العنونة بالمنفذ : I\O

توجد لدينا نوعين مستخدمين , مباشر (IN AL,02H) و غير مباشر (IN AL,DXH)

- 5- العنونة العلائقية :

JNC 08H ;IF Carry=0 , PC is loaded with current pc contents + 8bit signed value

- 6- العنونة المضمنة :

CLC : clear carry flag

توجيهات لغة التجميع
بالمعالج 8086

لغة الاسمبلي عبارة عن برنامج يحول من لغة التجميع للغة الماكينة.

- ASSUME : تستخدم لتدل المجمع على أسماء المقاطع المنطقية
- DB(Define Double Word) : تعرف كلمة مضاعفة أي

تحجز من الذاكرة 4 بايت

- DQ (Define Quad Word) : تعرف أربع كلمات تحجز من الذاكرة 8 بايت
- END : تدل المجمع على نهاية البرنامج.
- ENDP : تدل المجمع على نهاية الإجرائية.
- ENDM : تدل المجمع نهاية الماكرو MACRO.
- ENDS : تدل المجمع على نهاية المقطع.
- EQU(equate) : تستخدم لإعطاء اسم لقيمة معينة.
- EVEN : تدل المجمع على زيادة محتوى موقع العداد لموقع العنوان التالي من الذاكرة.
- GLOBAL : تستخدم لوضع موجه خارجي.

- GROUP: تدل المجمع على المجموعة المنطقية المسماة بعد التوجيه.
- LABEL: تعطي أسماء لمواقع في الذاكرة.
- MACRO: تعطي تسلسل للتعليمات.
- NAME: الاسم المخصص يمكن أن يعطى لأي برنامج يستخدم هذه التوجيهية .
- OFFSET: عملية تحدد تغير المتحول .
- ORG: تدفع المجمع على وضع موقع العداد بعنوان ذاكرة مخصص بعد التوجيهية .
- PTR: تشير إلى البايت أو الكلمة أو الكلمة المضاعفة للمتحول
- PUBLIC

- SIZE: عملية تحدد عدد البايتات المخصصة لعنصر المعطيات.
- TYPE: عملية يحدد فيها قيمة المتحول .
- SEGMENT: عملية تستخدم قبل المتحول أو الإجرائية .
- RECORD: تحدد نمط البتات الموجودة ضمن بايت أو كلمة.

تعليمات المعالج

8086

- **MOV** : تستخدم هذه التعليمة لنقل بايت أو كلمة معطيات من متحول المصدر لمتحول الهدف

MOV DL,[BX]

- **PUSH** : push source. لحفظ البارمترات في المكدس.
- **POP** : pop destination.لاسترجاع البارمترت من المكدس
- **XCHG** : تستخدم هذه التعليمة لاستبدال متحول المصدر بمتحول الهدف وبالعكس.
- **XLAT** : تقوم هذه التعليمة بجمع محتوى المسجل AL مع محتوى المسجل BX وتعتبر الناتج إزاحة بالنسبة لمقطع المعطيات

MOV AL,[BX+AL]

- IN : تقوم هذه التعليمة بنسخ القيمة من المدخل للمراكم.
- OUT : تظهر الكلمة أو البايت إلى المخرج .
- LEA,LES,LDS : تستعمل هذه التعليمات من أجل عملية نقل المعطيات لتحميل مسجل مقطع أو مسجل أغراض عامة بعنوان بشكل مباشر من الذاكرة . التعليمة LEA وظيفتها تحميل مسجل بعنوان فعال

  أما LDS فهي لتحميل مسجل ما ومسجل مقطع المعطيات DS

  وتعليمة LES وظيفتها تحميل مسجل ما ومسجل مقطع المعطيات الإضافي ES .
- ADD : جمع المصدر مع الوجهة .

  مثال ADD DX,BX

- ADC: جمع مع أخذ المحمول بعين الاعتبار.
- INC: عملية زيادة للوجهة .
- AAA: تستخدم هذه التعليمة لتصحيح ناتج جمع عددين بشيفرة لغة الأسكي.
- DAA: تستخدم هذه التعليمة لتصحيح ناتج جمع عددين بشيفرة العشري المرمز ثنائياً.
- SUB: تستخدم هذه التعليمة لطرح عددين
- SBB: تستخدم هذه التعليمة لطرح عددين مع أخذ بت الإستعارة بعين الاعتبار
- CMP: تقوم هذه التعليمة بمقارنة بايت أو كلمة وتؤثر على الأعلام .

- MUL: تستخدم هذه التعليمة لضرب بايت أو كلمة بدون إشارة.
- IMUL: تقوم هذه التعليمة بضرب مع أخذ الإشارة بعين الاعتبار.
- IDIV: تقوم هذه التعليمة بإجراء عملية القسمة مع إشارة لبايت أو كلمة
- NOT: تعكس كل بت في المعامل.
- AND: تقوم هذه التعليمة بعملية AND المنطقية بين المتحولين.
- OR: تقوم هذه التعليمة بعملية OR المنطقية بين المتحولين.
- XOR: تقوم هذه التعليمة بعملية XOR المنطقية بين المتحولين.

- SHL: تقوم هذه التعليمة بإزاحة بتات المتحول لليسار حيث يتم إدخال صفر إلى LSB

مثال : BX=11100101 11010011 CF=0

SHL BX,1

CF=1 BX=11001011 10100110

- SHR: تقوم هذه التعليمة بإزاحة بتات المتحول إلى اليمين يتم إدخال صفر إلى MSB

مثال : SI=10010011 10101101 CF=0

SHR SI,1

SI=01001001 11010110 CF=1

- ROL: تقوم هذه التعليمة بدوران بتات المتحول نحو اليسار حيث ال MSB تأخذ مكان ال LSB.

مثال : BX=10111011 01110101 CF=0

ROL BX,1

CF=1 BX=01110110 11101011

- ROR: تقوم هذه التعليمة بدوران بتات المتحول نحو اليمين حيث ال LSB تأخذ مكان ال MSB

مثال: BX=00111011 01110101 CF=0

ROR BX,1

CF=1 BX=10011101 10111010