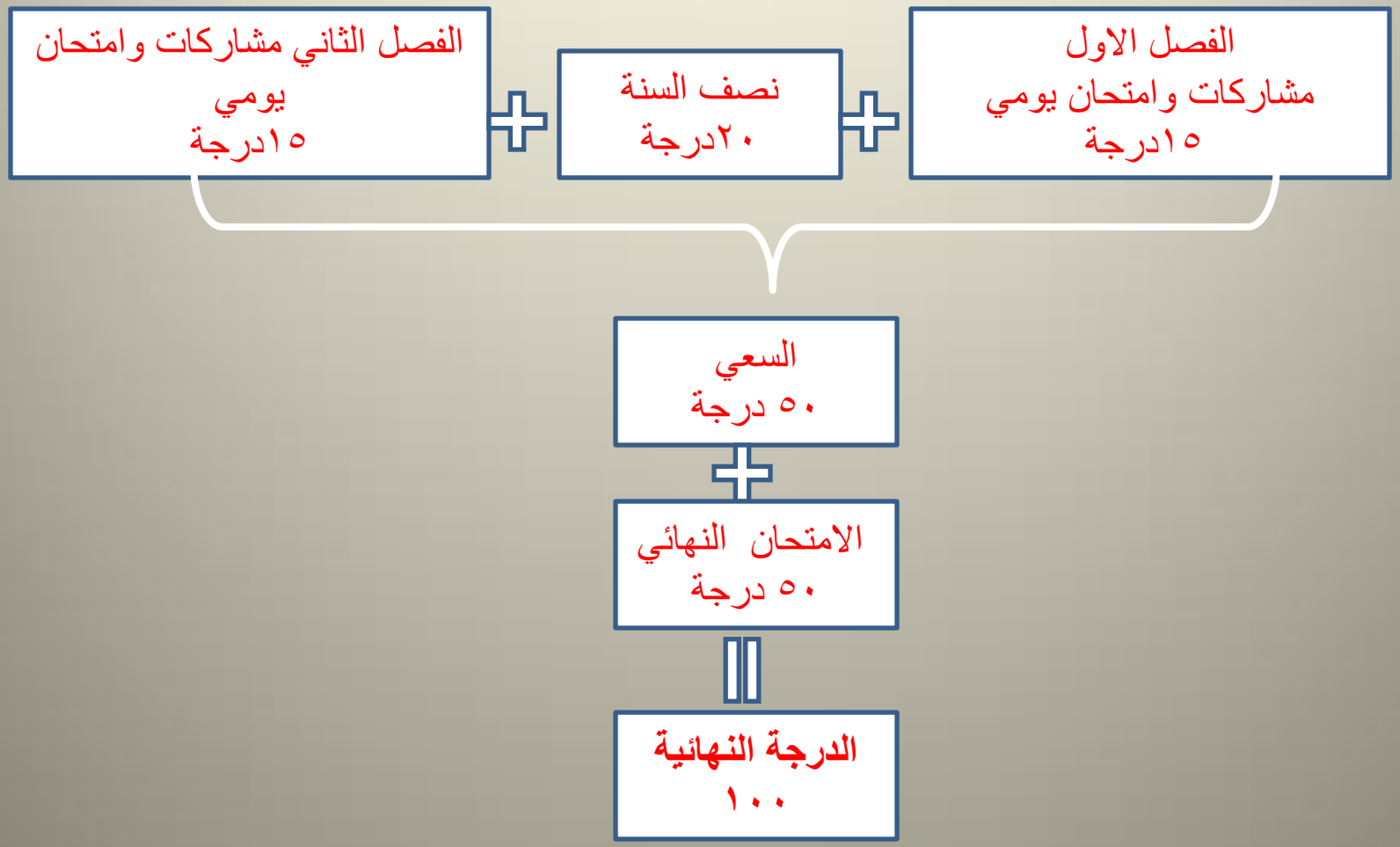
المعالجات المايكروية



Week	Topics Covered
1	Cpu architecture &- three-bus system architecture
2	Memory & register
3-4-5	fetch and execute & physical address
6-7	Addressing mode
8	Instruction set (form ,orthogonally , number of addressing) - Decoding
9-10	Data transfer instruction
11-12	Arithmetic instruction
13-14-15	Logic instruction
16-17	- String instruction
18-19	Transfer of control instruction
20	Coding the program
21-22-23	Stack (concepts and applications)
24	Brief introduction to machine code
25-26-27	Interrupt and interrupts service routines
28-29-30	i/o port

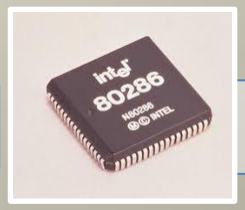
Microprocesser
المعالج الدقيق

- تعريف المعالج الدقيق:احد المكونات الالكترونية الرقمية القابلة للبرمجة؛ أي أنه شريحة ذات أطراف عديدة تستقبل الأوامر وتقوم بتنفيذها تباعا حسب برنامج مخزن مسبقا في شريحة ذاكرة خارجية.

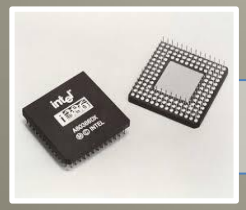
وبذلك يستخدم المعالج الدقيق بالعديد من الوظائف، مثل التحكم في عملية صناعية أو متغير طبقاً للمدخلات من الحساسات (المستشعرات) الإلكترونية أو إعدادات المستخدم

٨٠٨٠ سنة ١٩٧٣

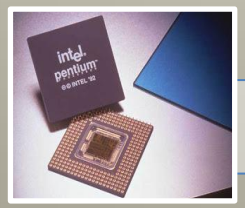
معالج ٨٠٨٥ اطلق ١٩٧٧



(mp 80286)
اطلق هذا المعالج فى عام ١٩٨٣



المعالج
٨٠٣٨٦
اطلق
١٩٨٨

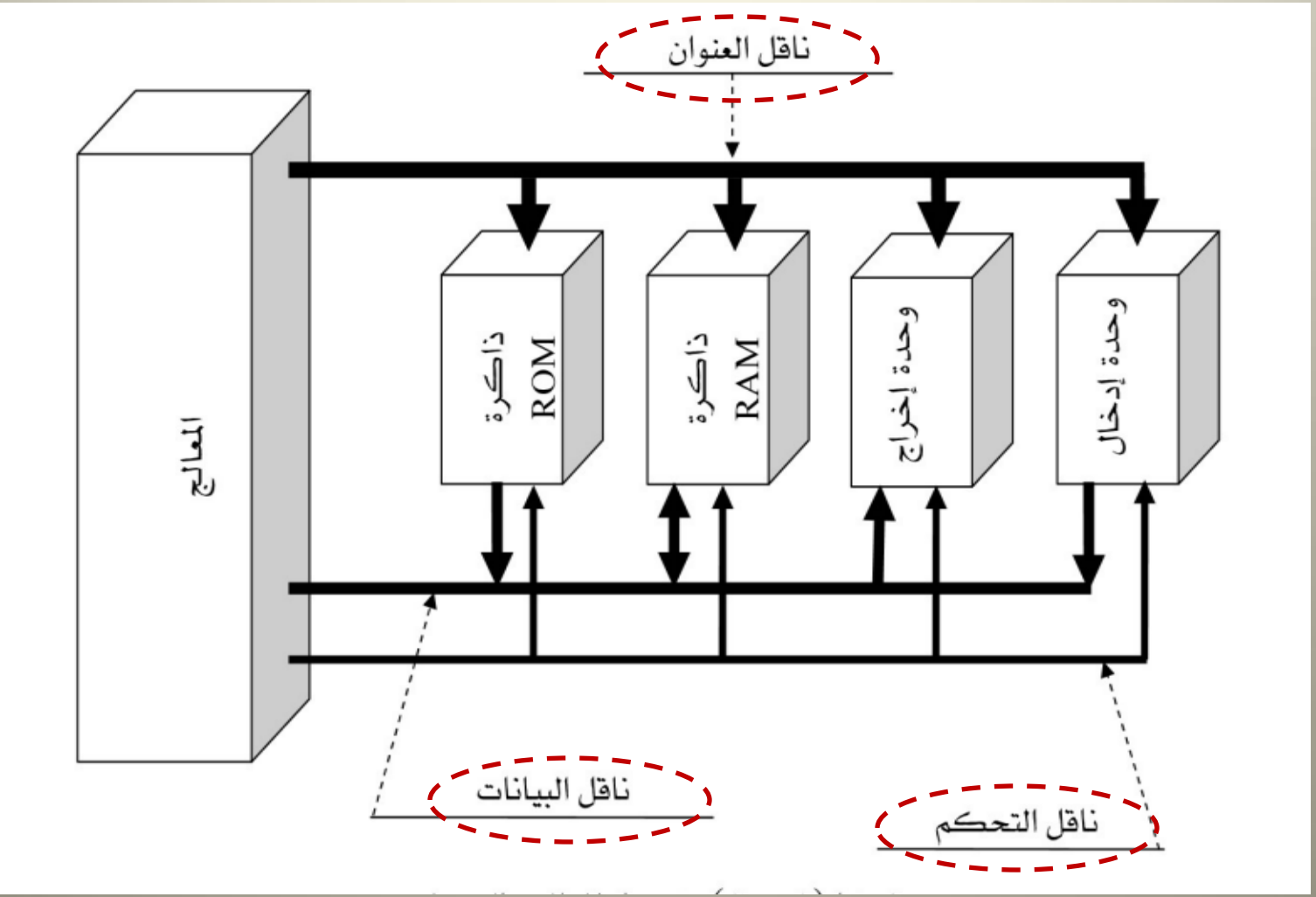


64Bit Micro Processor
١٩٩٥pintum

يوفر أربع نوى تنفيذ مستقلة في حزمة معالج واحدة اطلق ٢٠٠٩

cori9المعالج يضم ١٨ نواة

علاقة المعالج مع اطراف الحاسوب



الناقل(bus)

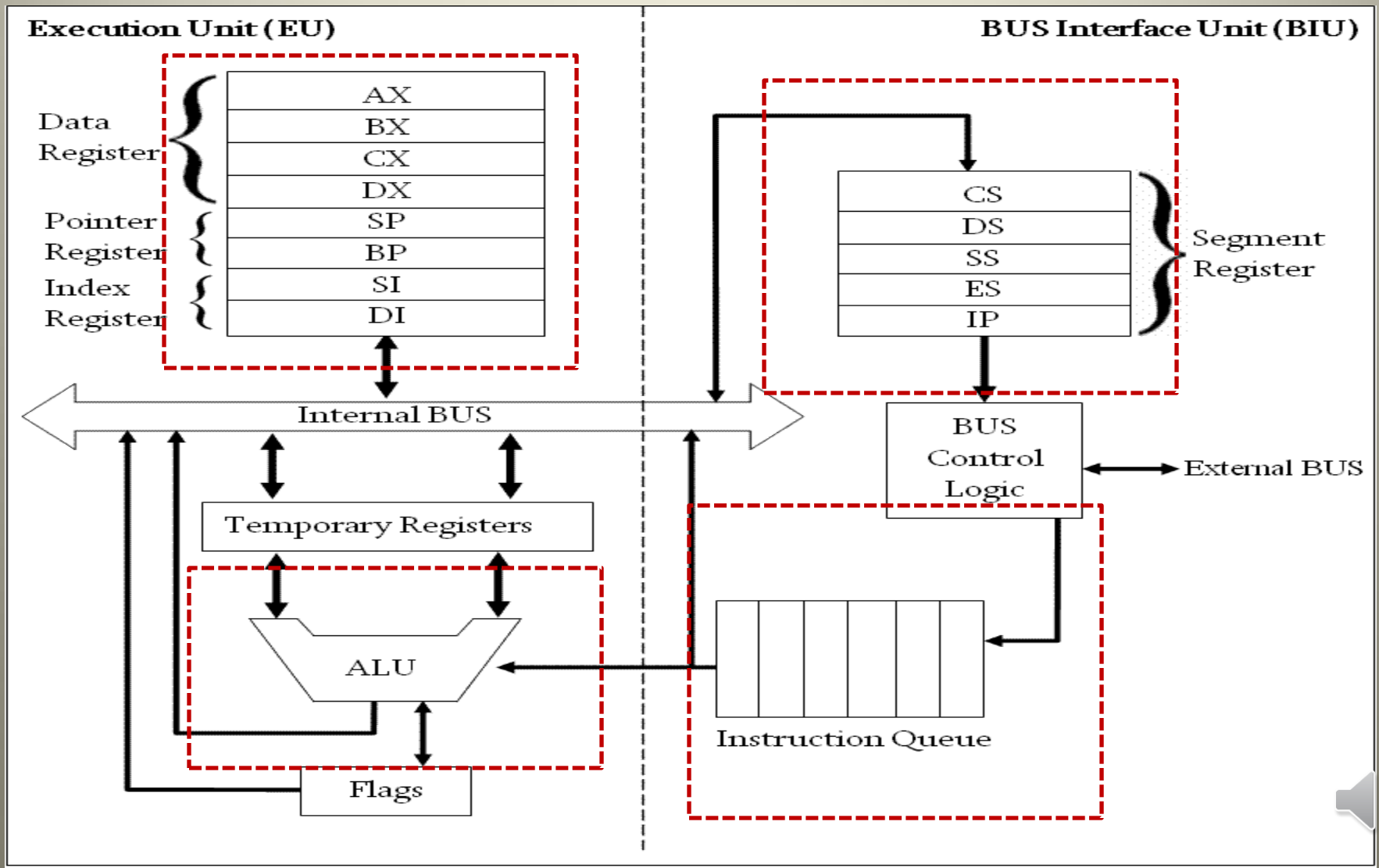
الناقل(bus) :
مجموعة من الاسلاك الكهربائية 5v/0v

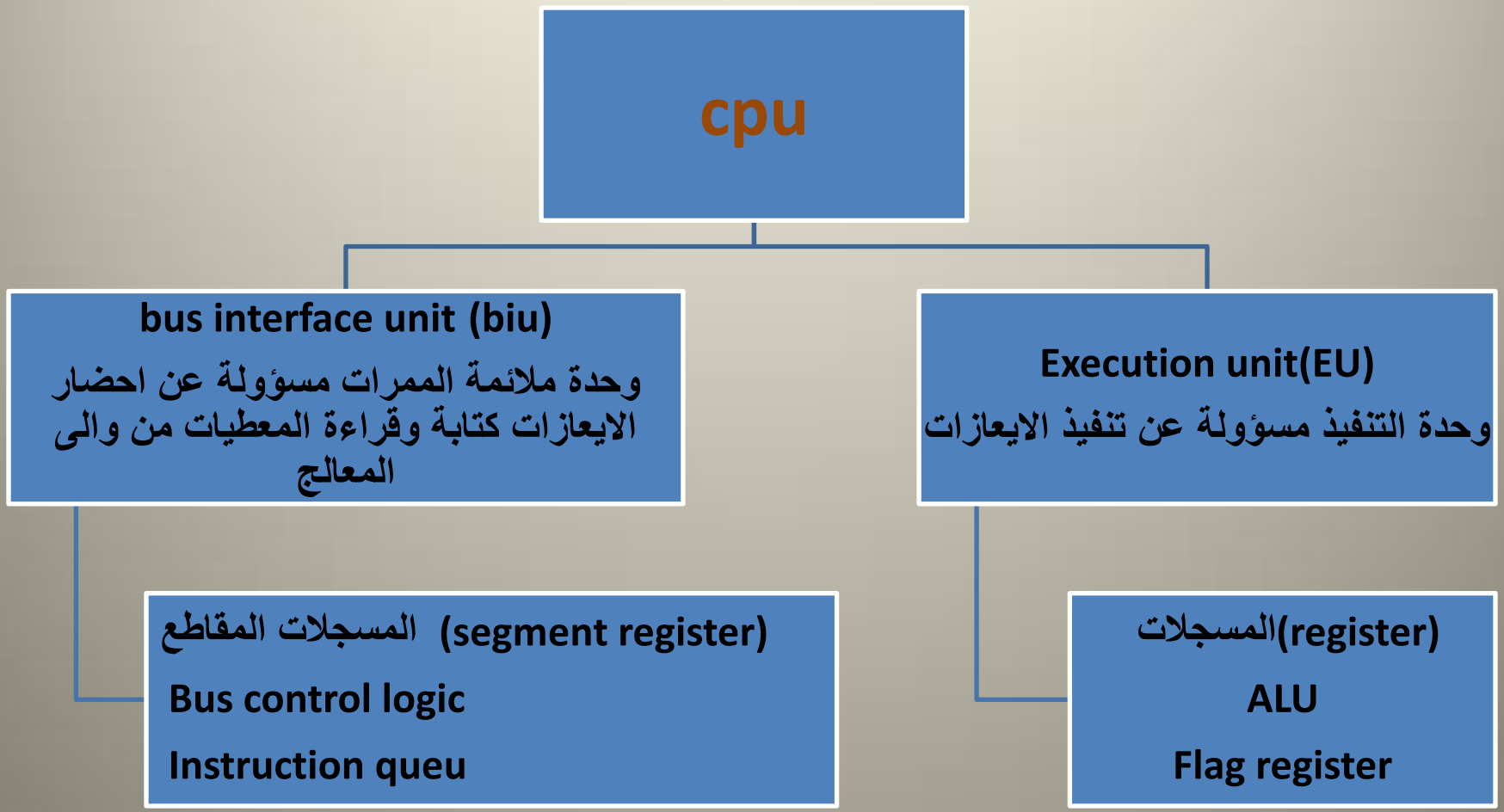
Data bus(16 bit)
مسؤولة عن نقل البيانات من والى المعالج وتكون باتجاهين

Address bus(16 bit)
مسؤولة عن نقل العناون من المعالج الى الاطراف الاخرى تكون باتجاه واحد

Control bus (1 bit)
خطوط السيطرة مسؤولة عن تنسيق عمل الناقلين السابقين تكون باتجاه واحد

البنية المعالج الدقيق




cpu
bus interface unit (biu)
وحدة ملائمة الممرات مسؤولة عن احضار الايعازات كتابة وقراءة المعطيات من والى المعالج
Execution unit(EU)
وحدة التنفيذ مسؤولة عن تنفيذ الايعازات
المسجلات المقاطع (segment register)
Bus control logic
Instruction queu
المسجلات(register)
ALU
Flag register

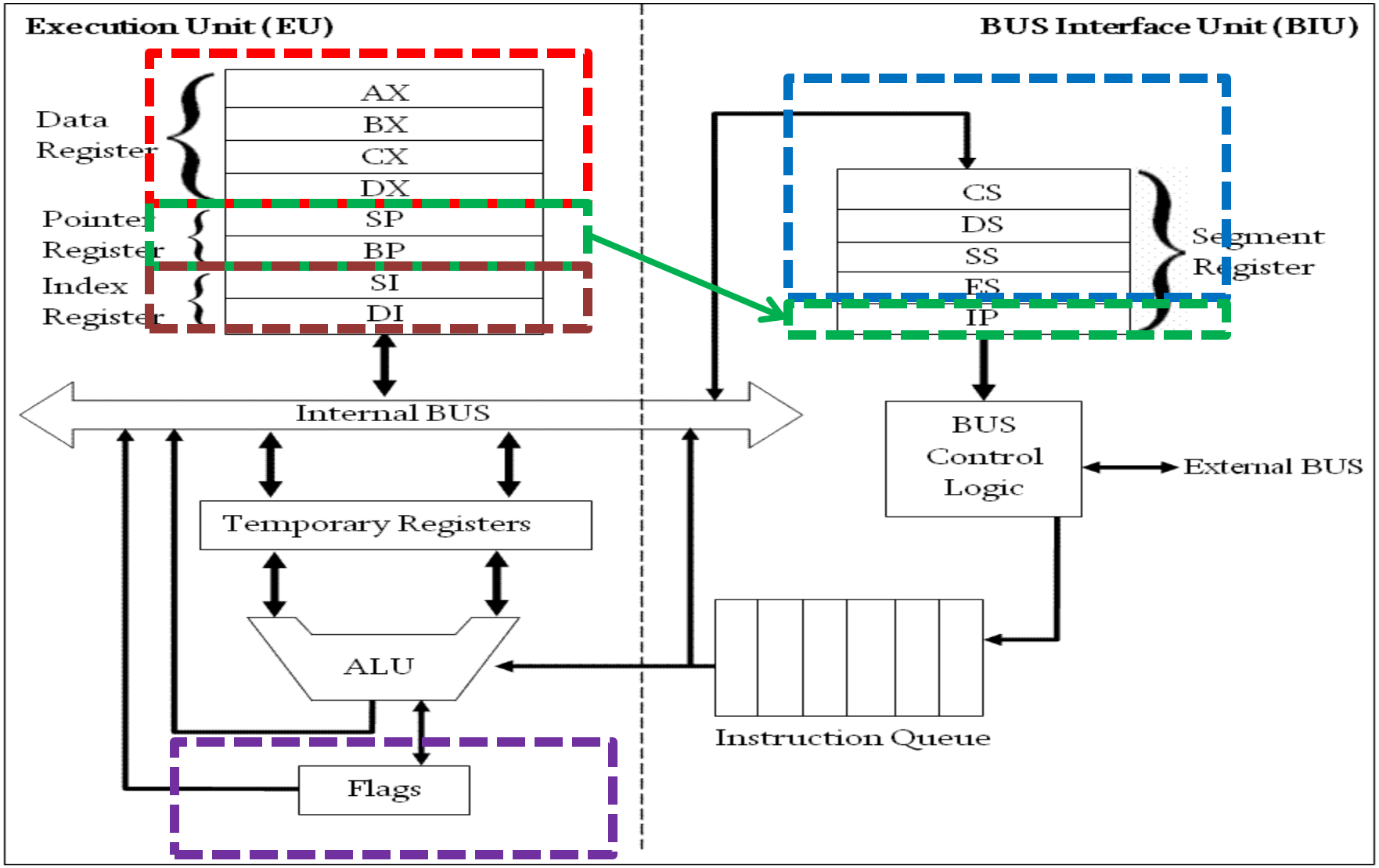
واجب بيتي

ملاحظة:حاول الاجابة عن هذه الاسئلة وتثبيتها على دفتر

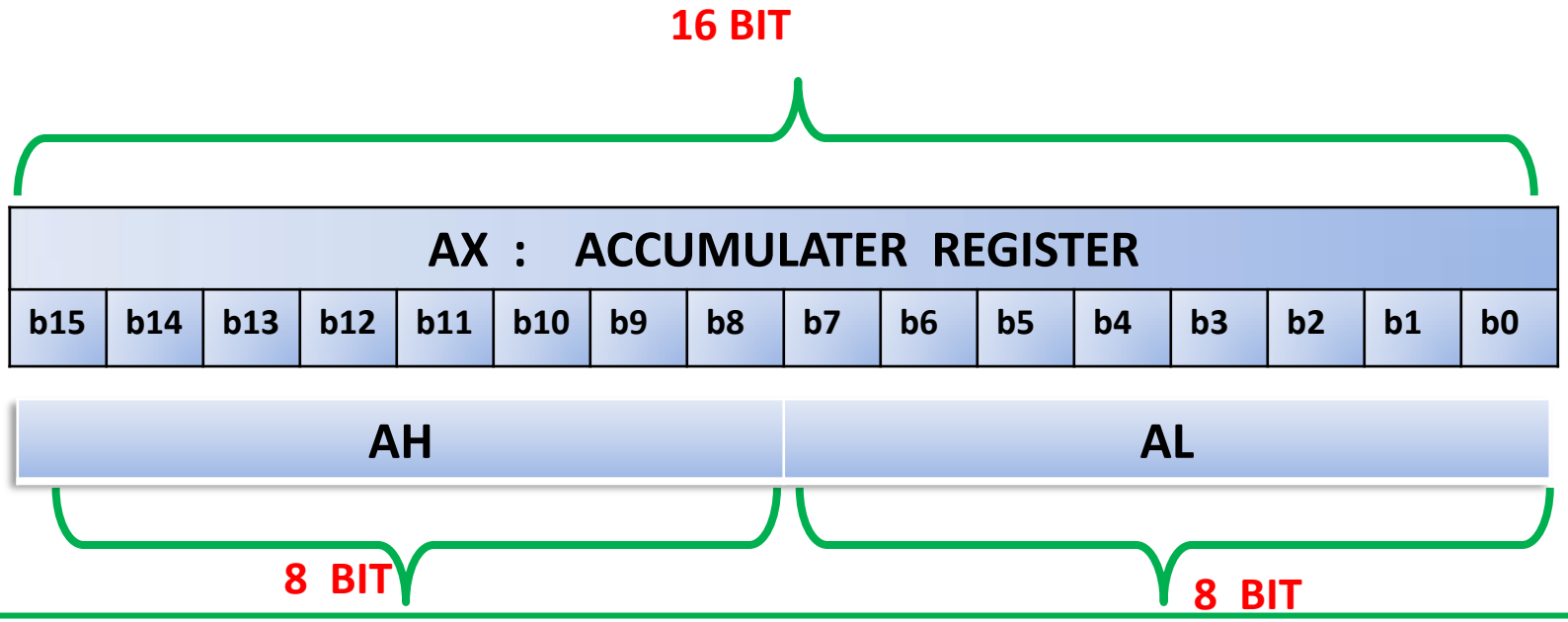
- عرف المعالج الدقيق cpu؟
- عرف الناقل bus وماهي انواعه؟
- ماهي اجزاء وحدة التنفيذ EU ؟

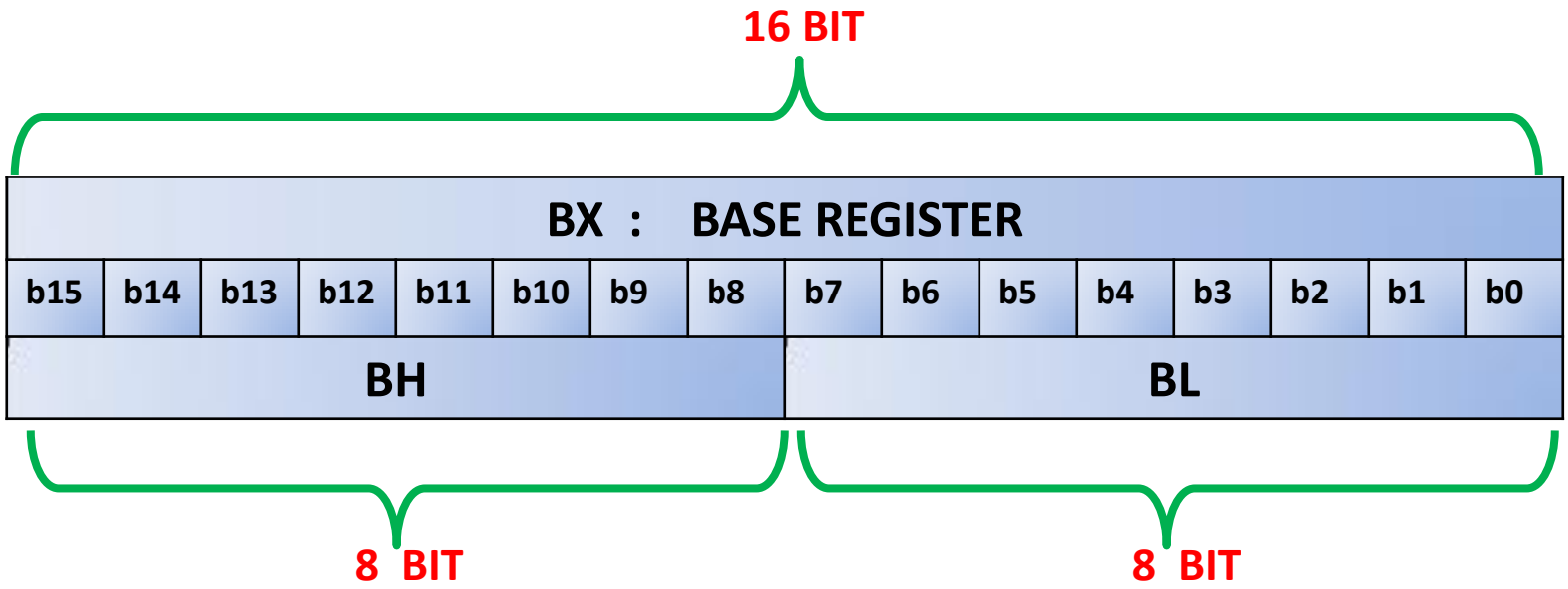
Regesiter
المسجلات

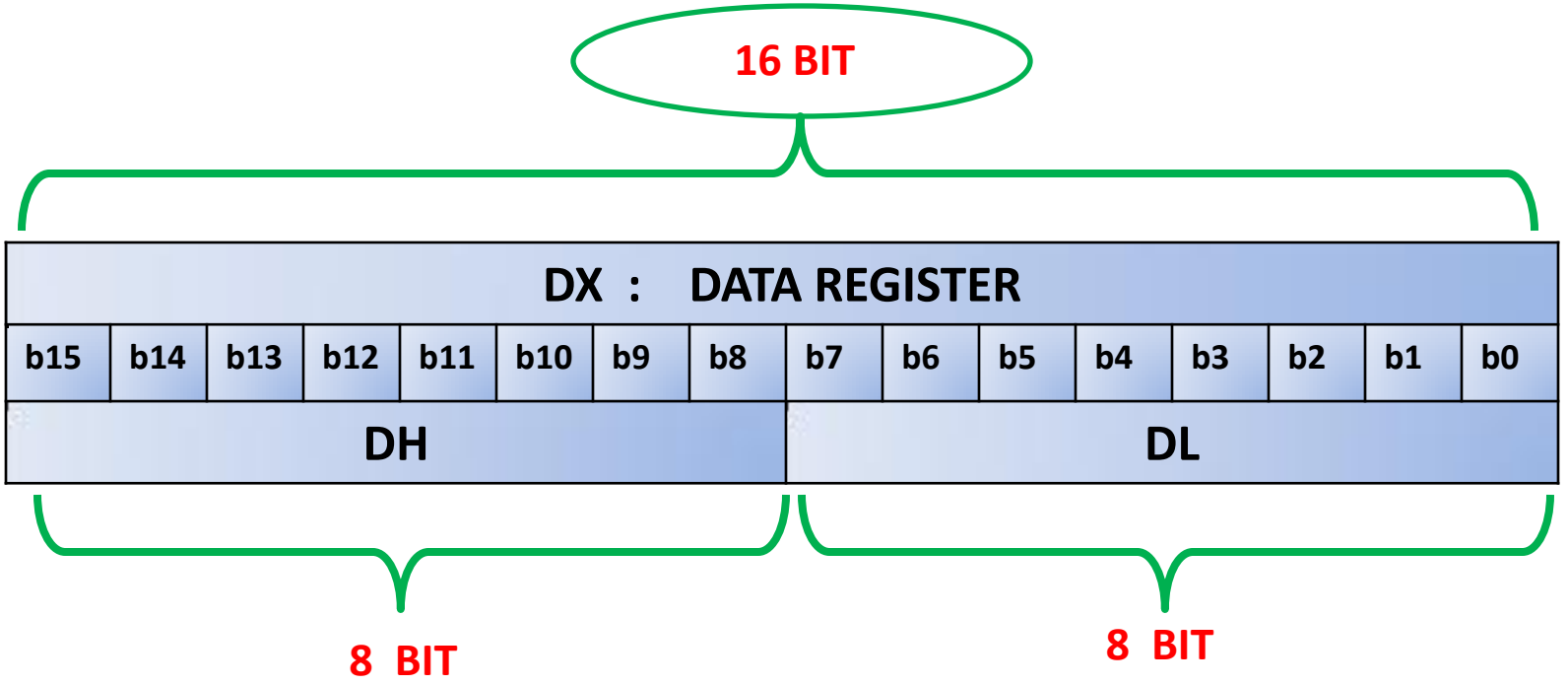
- تستخدم المسجلات في المعالج لتخزين المعلومات بشكل مؤقت هذه المعلومات يمكن ان تكون بيانات اما بطول **واحد byte او 2 byte .**
- لذا يمكن ان نتعامل مع مسجلات بطول **واحد byte او 2byte .**

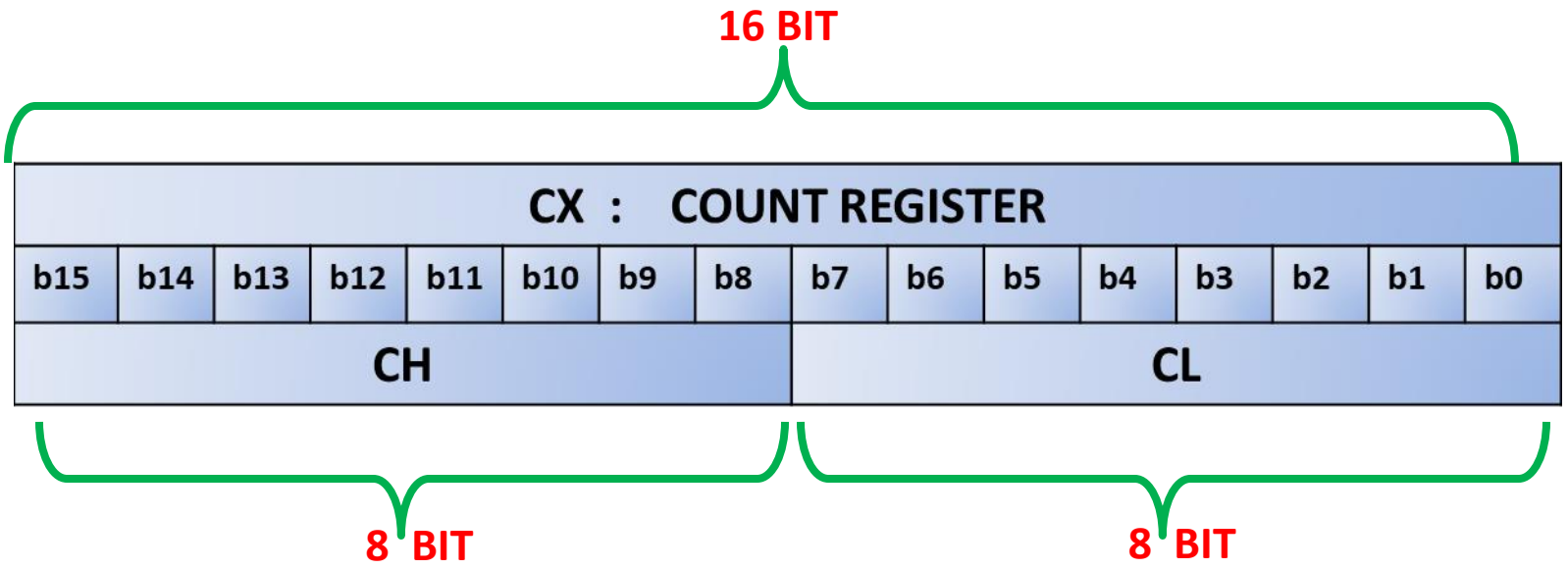

Execution Unit (EU)
BUS Interface Unit (BIU)
Data Register
AX
BX
CX
DX
Pointer Register
SP
BP
Index Register
SI
DI
CS
DS
SS
ES
IP
Segment Register
Internal BUS
BUS Control Logic
External BUS
Temporary Registers
ALU
Instruction Queue
Flags

Data register






16 BIT
DX : DATA REGISTER
b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
DH
DL
8 BIT
8 BIT


16 BIT
CX : COUNT REGISTER
b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
CH
CL
8 BIT
8 BIT

• Pointer register (SP, BP, IP)

16 BIT

SP: STACK POINTER															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

SP: هذا المسجل يشير الى مقطع الذاكرة STACK SEGMENT (SS)

16 BIT

BP: BASE POINTER															
b15	b14	b13	b12	B11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

BP: هذا المسجل يشير الى قاعدة مقطع الذاكرة STACK SEGMENT (SS)

16 BIT

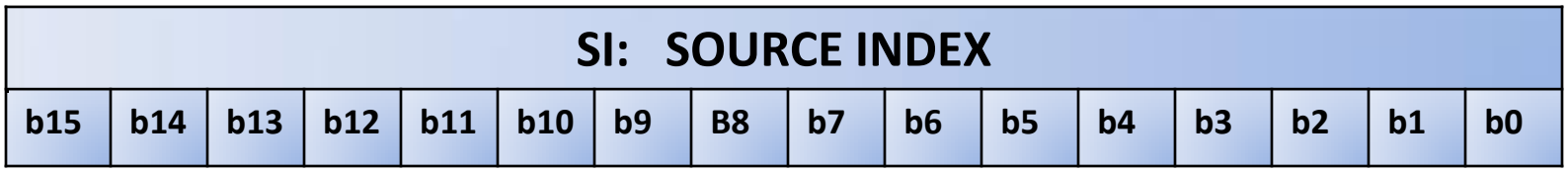
IP: INSTRUCTION POINTER															
b15	b14	b13	b12	B11	b10	b9	b8	B7	b6	b5	b4	b3	b2	b1	b0

IP:هذا المسجل يشير الى مقطع الذاكرة CODE SEGMENT (CS)

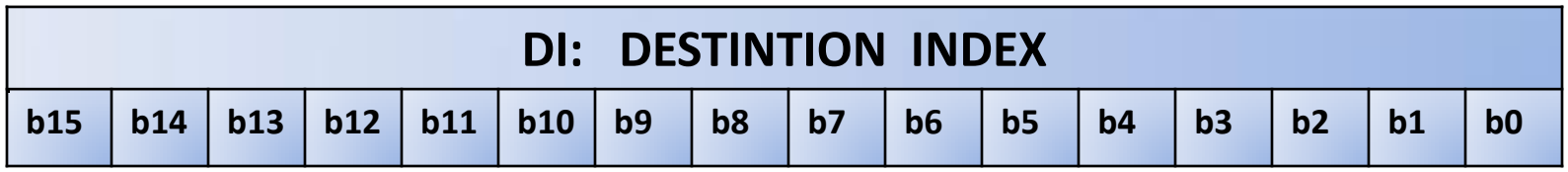
INDEX REGESITER (SI , DI)

هذه المسجلات تشير الى مقطع الذاكرة (DS) DATA SEGMENT

16 BIT



16 BIT



SEGMENT REGISTER

هذه المسجلات تحمل عناوين تشير الى بداية مقطع الذاكرة

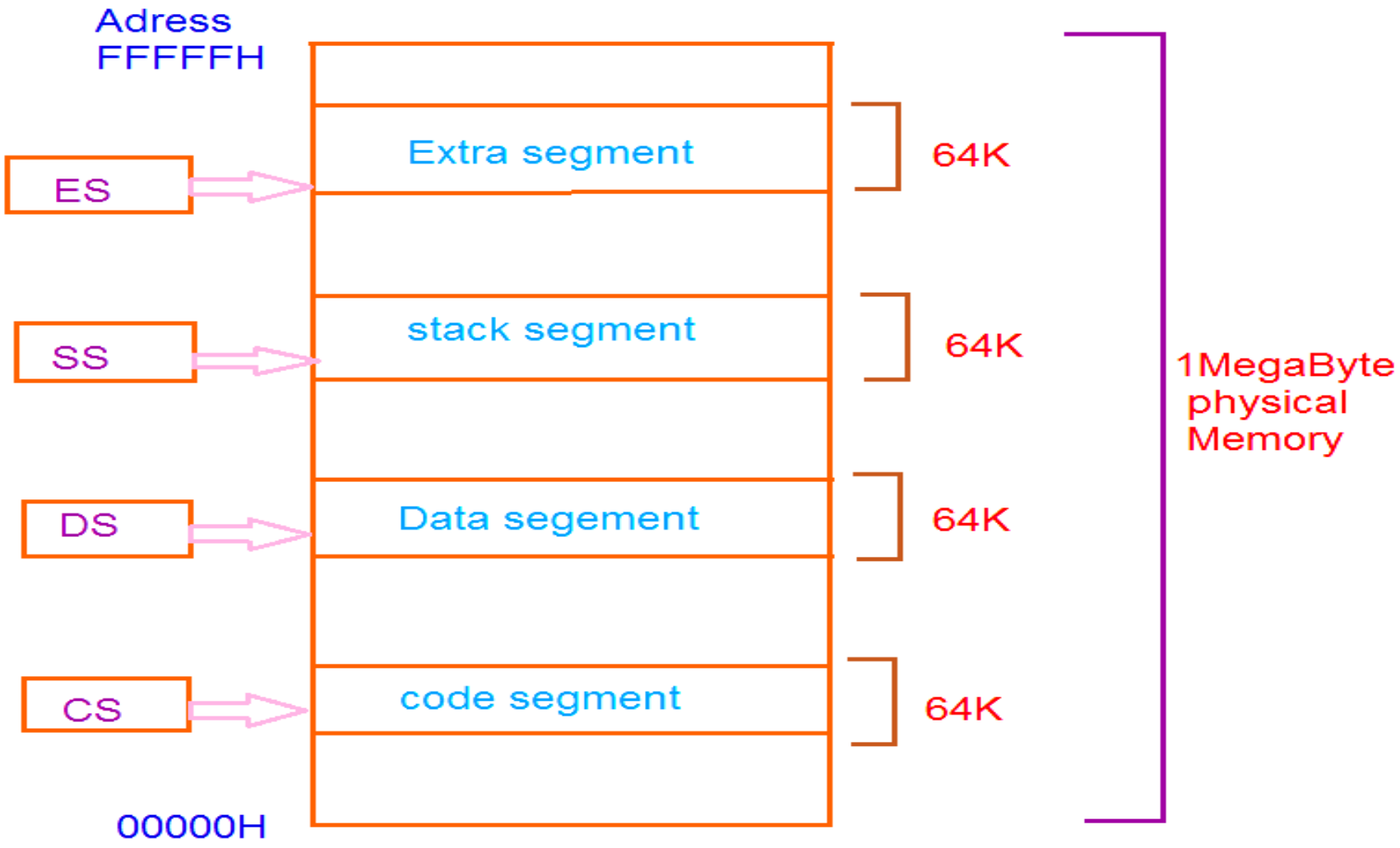
CS: CODE SEGMENT															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

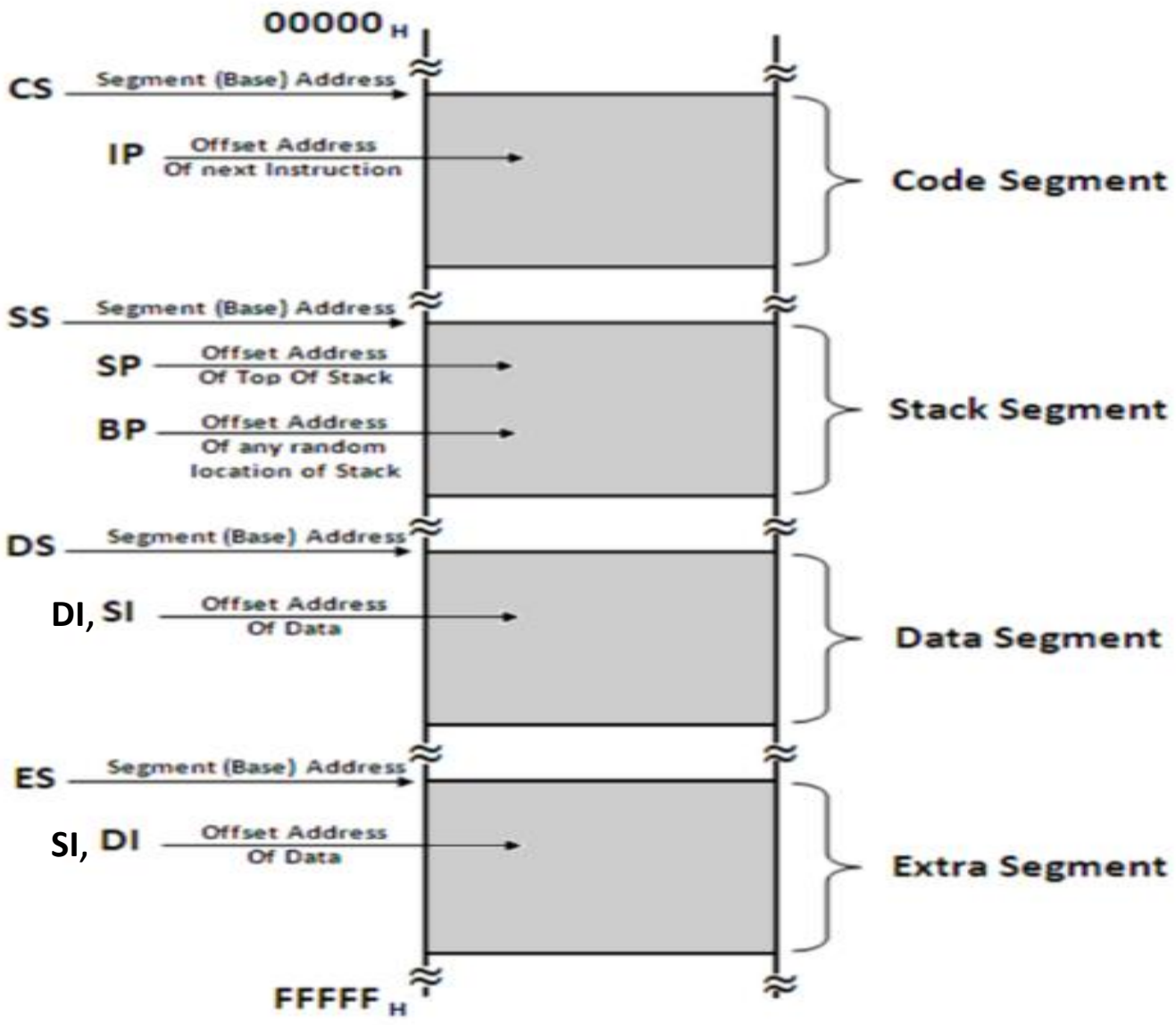
DS: DATA SEGMENT															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

SS: STACK SEGMENT															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

ES: EXSTRA SEGMENT															
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0

MEMORY: الذاكرة هي مكان لخزن البيانات وتقاس بوحدة BYTE




00000 H
CS
Segment (Base) Address
IP
Offset Address
Of next Instruction
Code Segment
SS
Segment (Base) Address
SP
Offset Address
Of Top Of Stack
BP
Offset Address
Of any random
location of Stack
Stack Segment
DS
Segment (Base) Address
DI, SI
Offset Address
Of Data
Data Segment
ES
Segment (Base) Address
SI, DI
Offset Address
Of Data
Extra Segment
FFFFF H

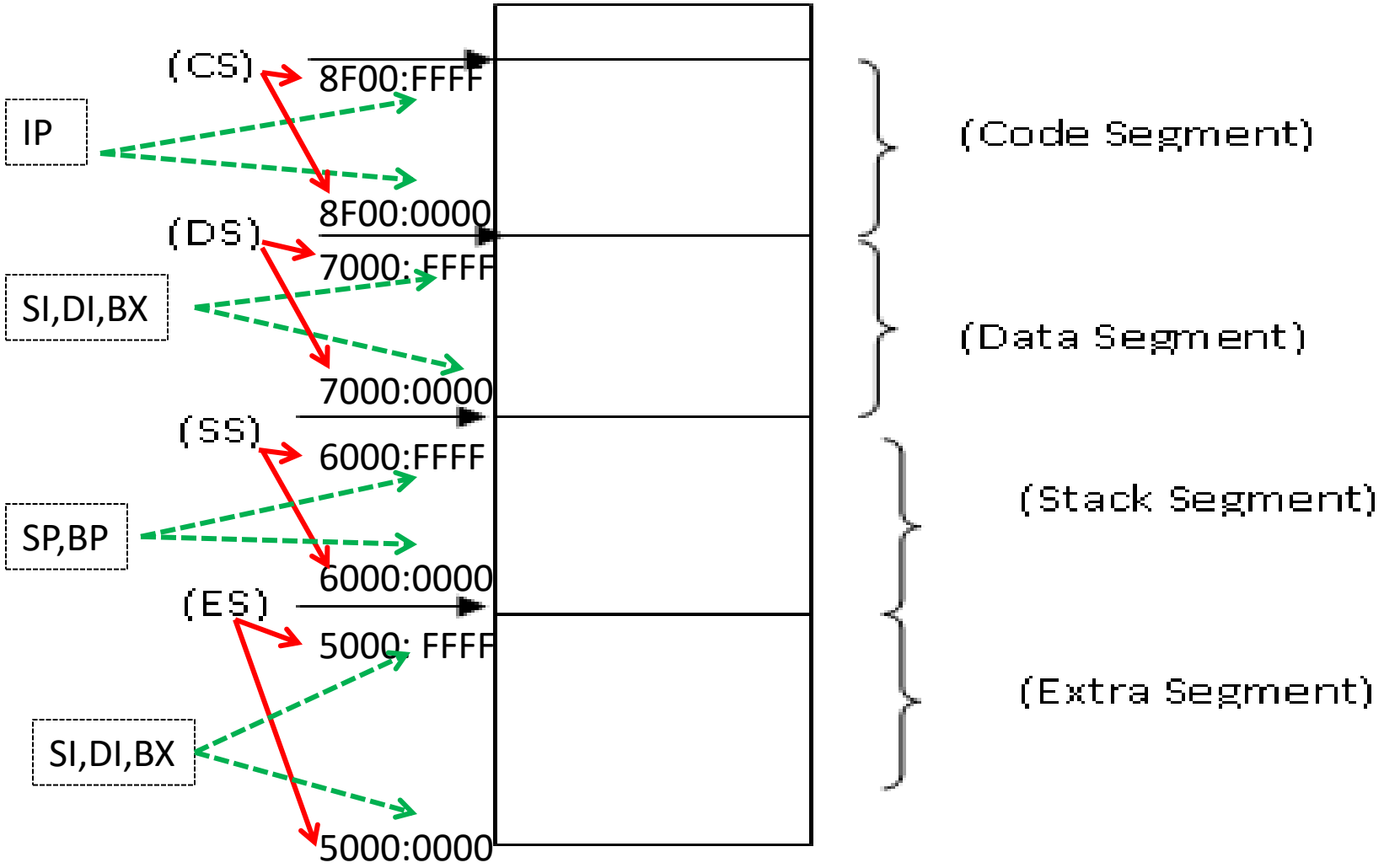


Figure: Memory Segmentation
(Code, Data, Stack and Extra Segments)

واجب بيتي

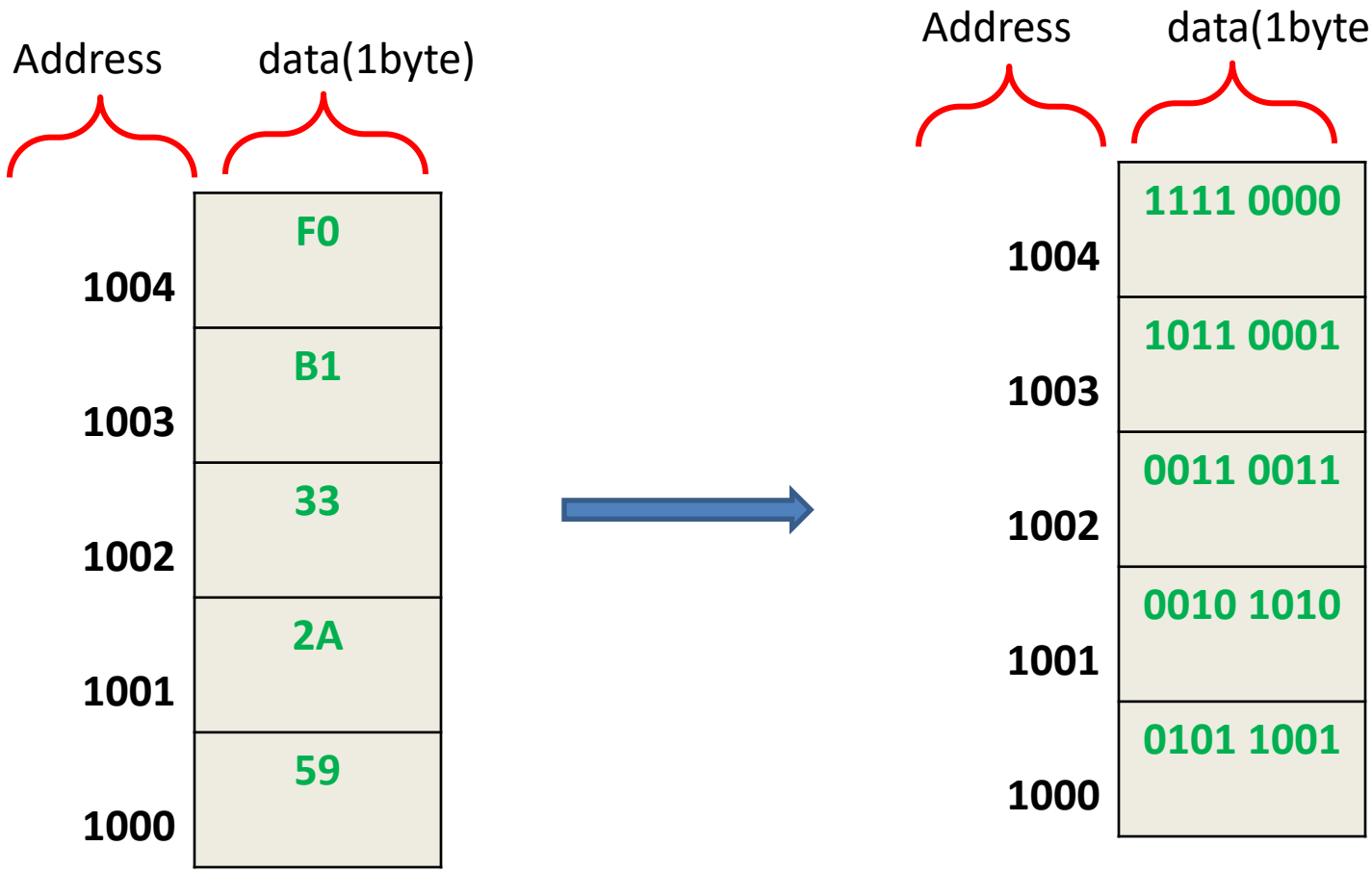
- اكتب الاعداد من واحد الى 20بنظام hex
- اجمع العددين ١٦(٣٦) مع ١٦ (٩٤) بنظام hex
- اجمع العددين ١٦(1A) مع ١٦ (B2) بنظام hex

memory

binary

59 → 0101 1001

2A → 0010 1010



كيفية التعامل مع الذاكرة

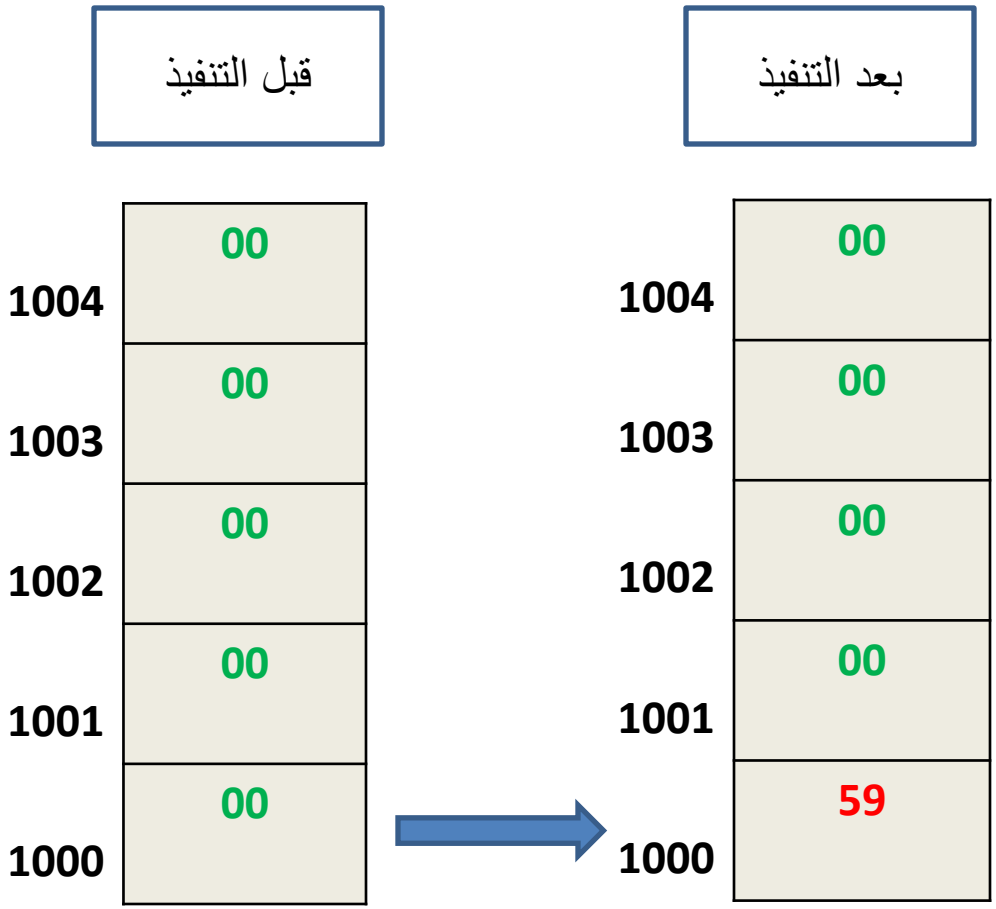
مثال١ :

اكتب برنامج لخزن القيمة ٥٩في العنوان ١٠٠٠

MOV AL , 59H

MOV [1000] , AL

ملاحظة: عند التعامل مع العنوان يجب وضع العنوان بين اقواس

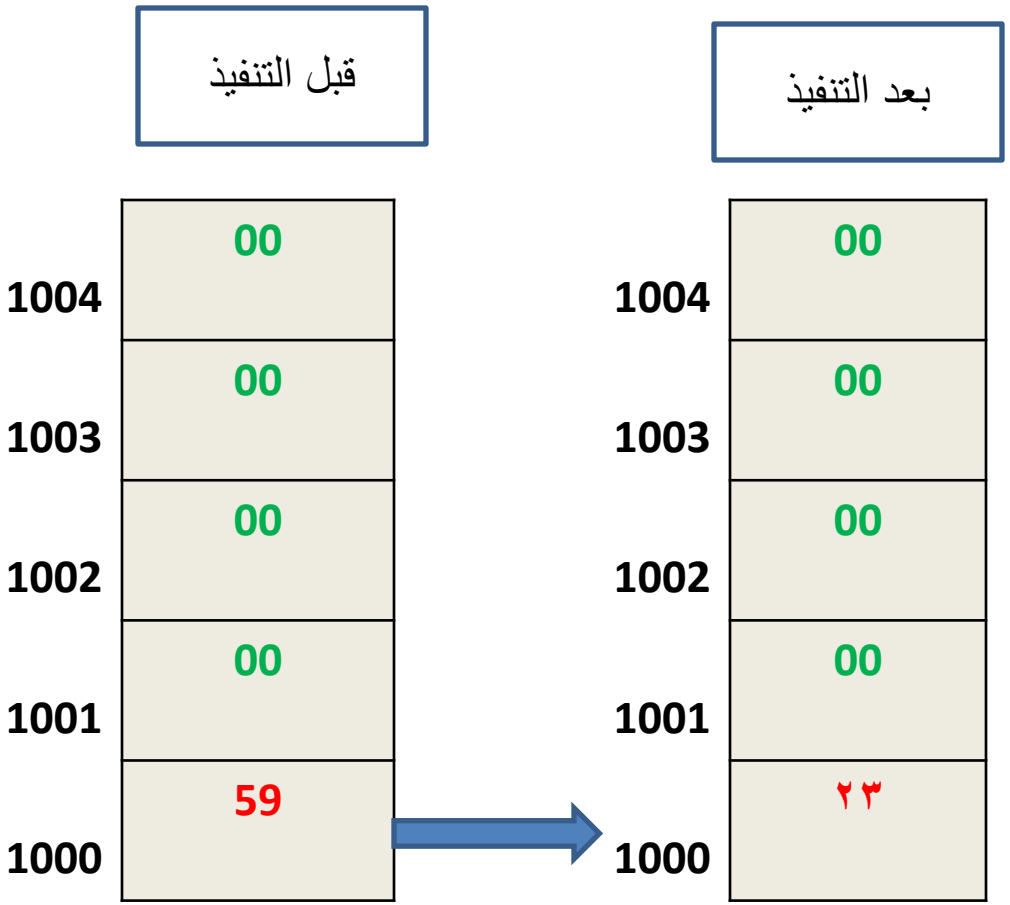


مثال٢:

اكتب برنامج لخزن القيمة ٢٣ في العنوان ١٠٠٠

MOV AH , ٢٣H

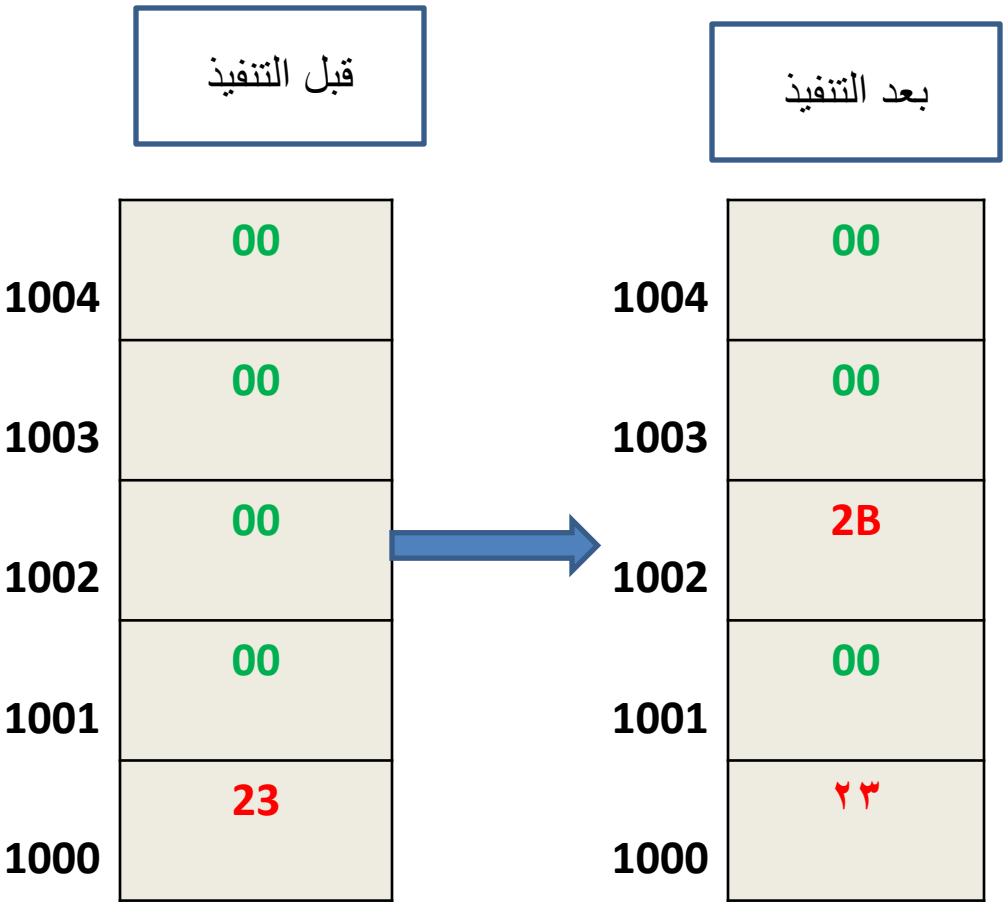
MOV [1000] , AH



مثال٣:
اخزن القيمة 2B في محتويات العنوان ١٠٠٢

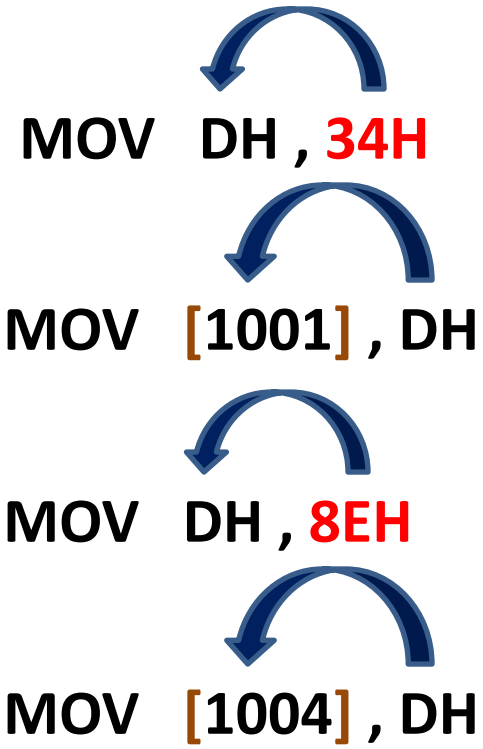
MOV DH , 2BH

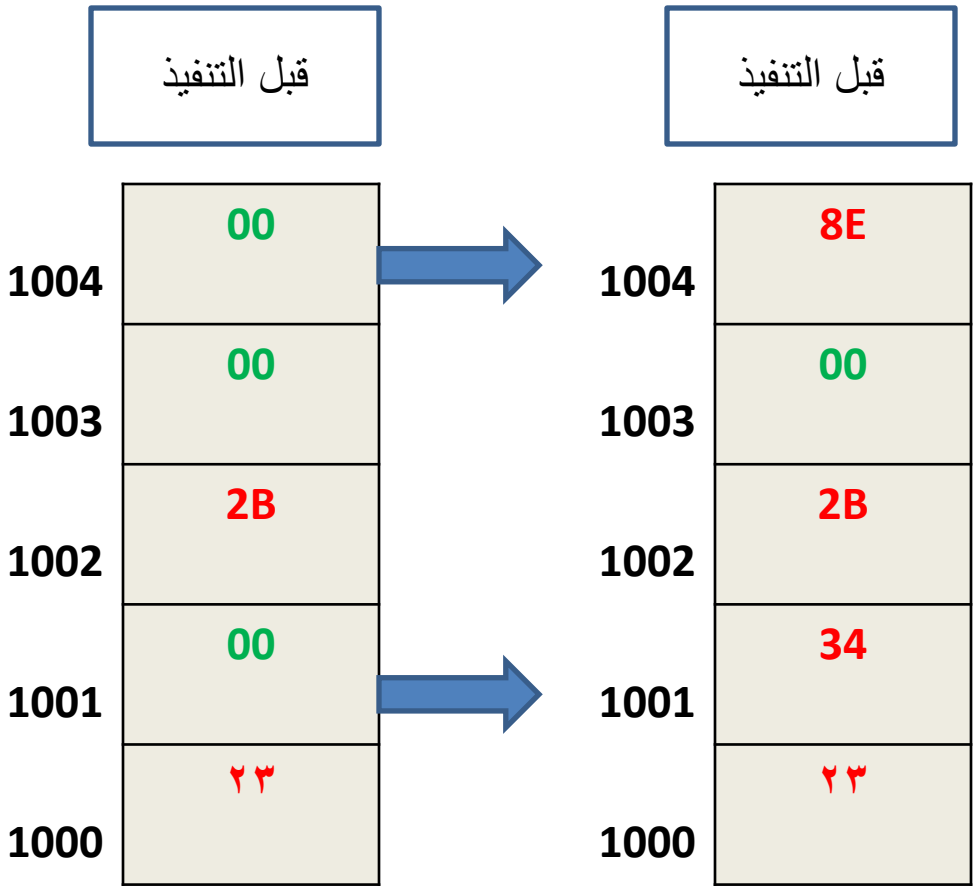
MOV [1002] , DH



مثال ٤:

اخزن القيمتين ٣٤و8E في محتويات الذاكرة عند الموقعين ١٠٠١و ١٠٠٤

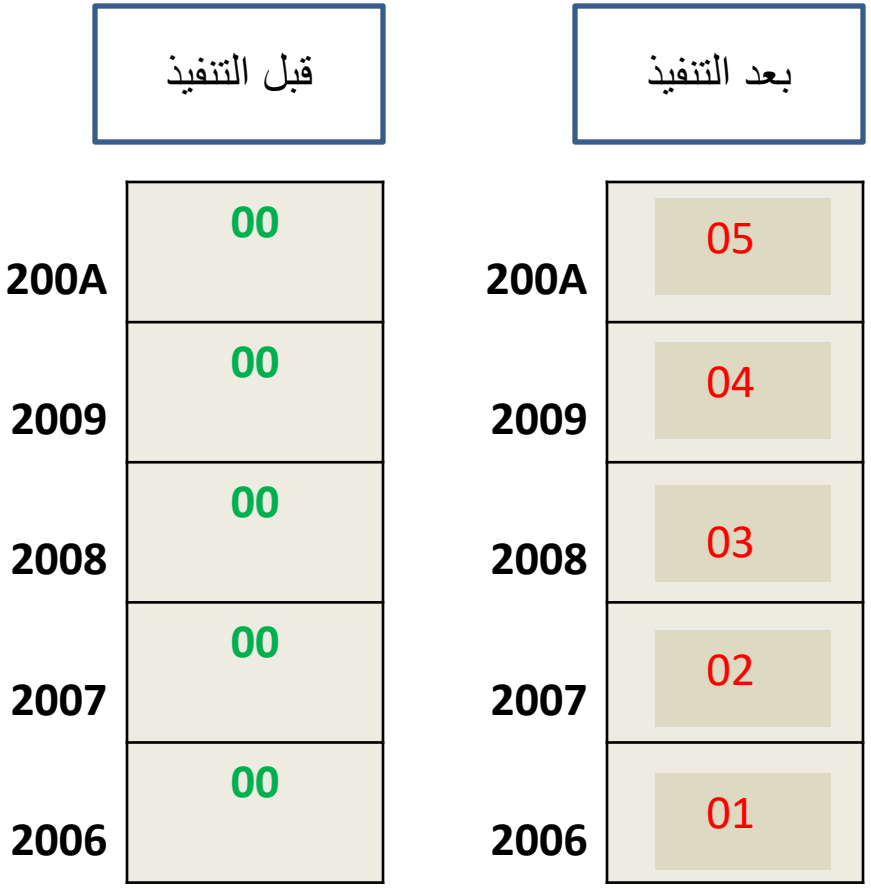




مثال 5:

اكتب برنامج لخزن القيم (1,2,3,4,5) في محتويات الذاكرة (2006,2007,2008,2009,200A)

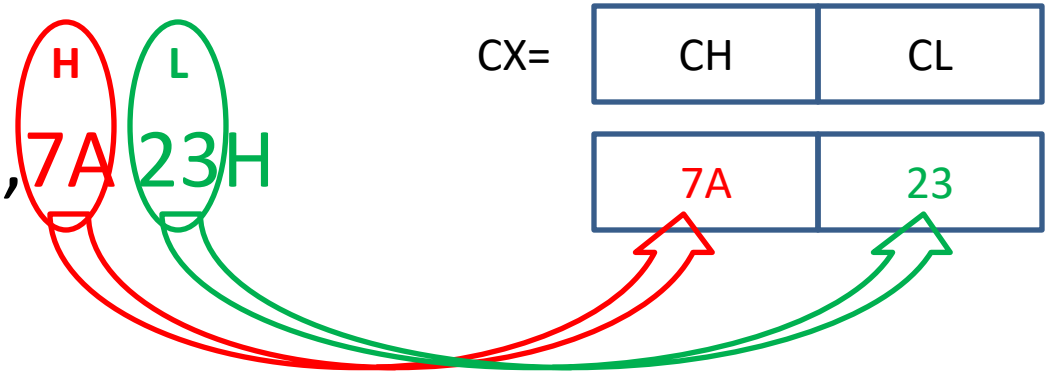
```
MOV   DH , ٠١H
MOV   [2006] , DH
MOV   DL , ٠٢H
MOV   [2007] , DL
MOV   AL , ٠٣H
MOV   [2008] , AL
MOV   AH , ٠٤H
MOV   [2009] , AH
MOV   BH , ٠٥H
MOV   [200A] , BH
```



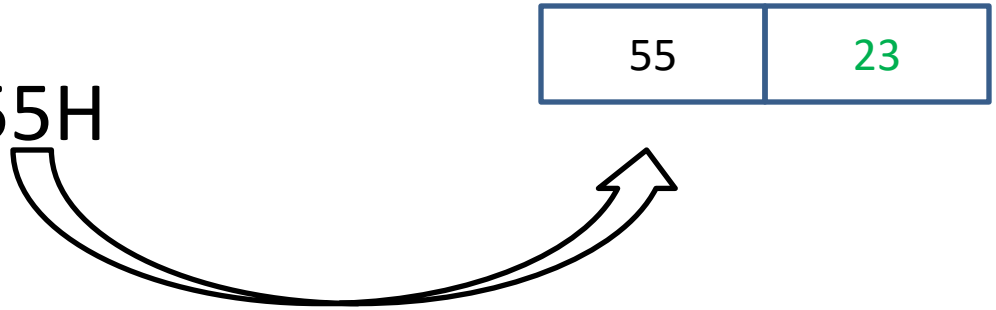
اختبار

- ماهي قيمة المسجل CXبعد تنفيذ الايعازات التالية:
- MOV CX ,7A23H
- MOV CH , 55H
- هل الايعاز التالي صح ام خطا ولماذا؟
- MOV 2000,AL
- ماهو عمل الايعاز التالي؟
- MOV AH , 71H
- MOV [1200],AH

- MOV CX ,7A 23H



- MOV CH , 55H



CX=5523

- MOV AH , 71H
- MOV [1200],AH

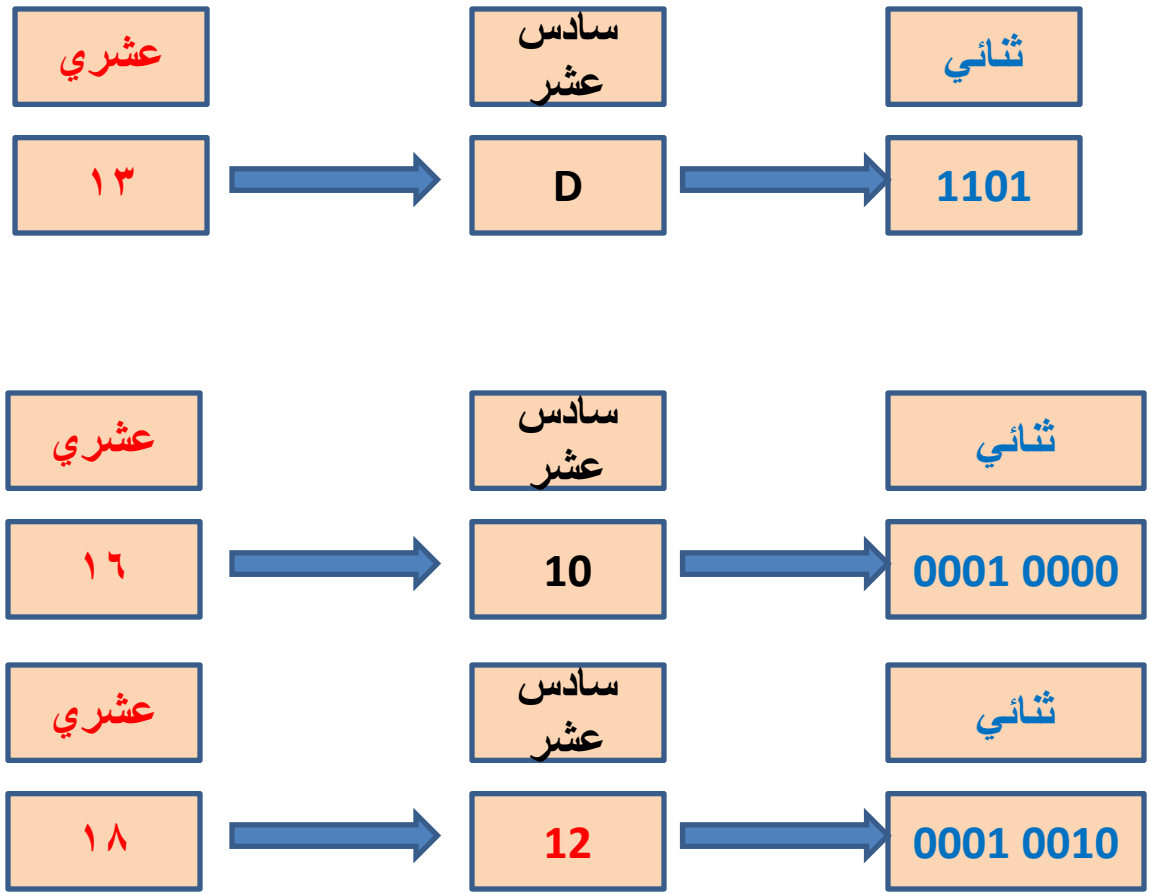
AH= 71

	قبل التنفيذ	بعد التنفيذ
1204	00	00
1203	00	00
1202	00	00
1201	00	00
1200	00	71

الاعداد بالنظام السادس عشر

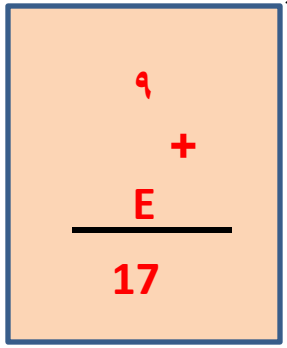
العشري	ثنائي	السادس عشر
١٣	1101	D
١٤	1110	E
١٥	1111	F
١٦	0001 0000	10
١٧	0001 0001	11
١٨	0001 0010	12
١٩	0001 0011	13

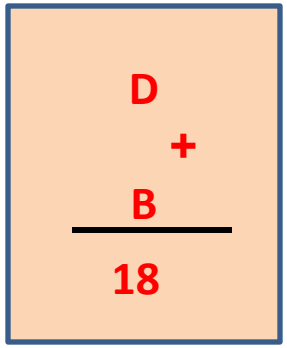
اسئلة


عشري
سادس عشر
ثنائي
١٣
D
1101
عشري
سادس عشر
ثنائي
١٦
10
0001 0000
عشري
سادس عشر
ثنائي
١٨
12
0001 0010

اسئلة

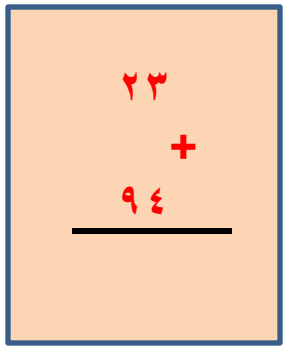
- اجمع العددين مكتوبين بالنظام السادس عشر

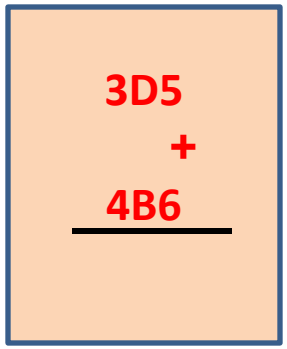




واجب بيتي

- اجمع العددين مكتوبين بالنظام السادس عشر





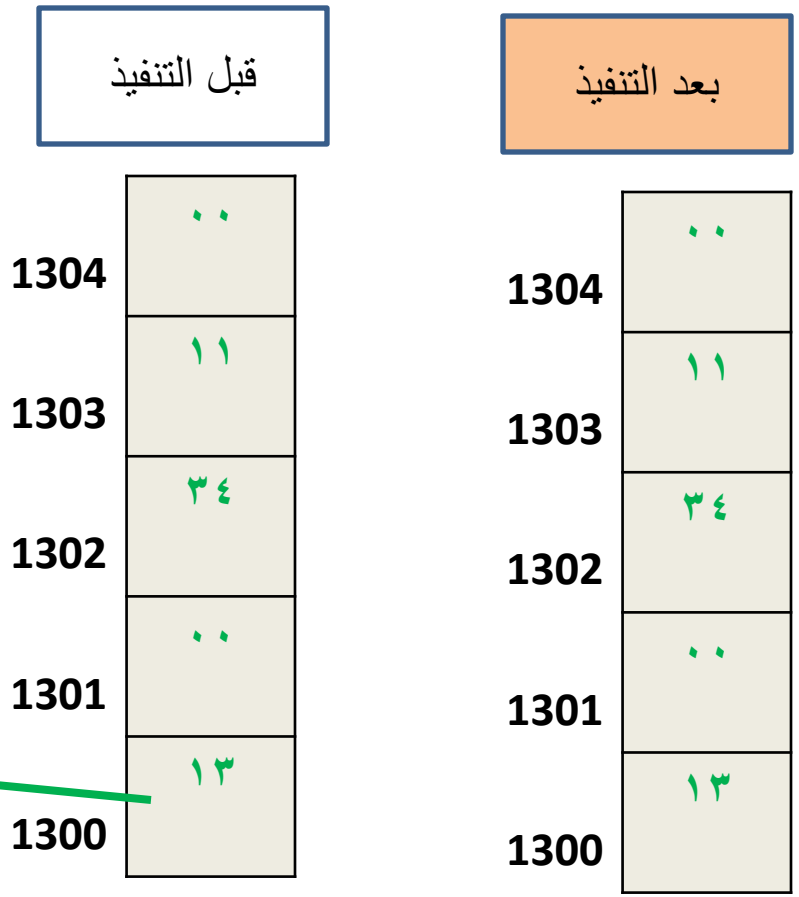
القراءة من الذاكرة

مثال:اقرأ بايت من الذاكرة عند العنوان ١٣٠٠ الى المسجل dh

قيمة المسجل DH قبل التنفيذ
DH=00

MOV DH , [1300 H]

قيمة المسجل DH بعد التنفيذ
DH=13

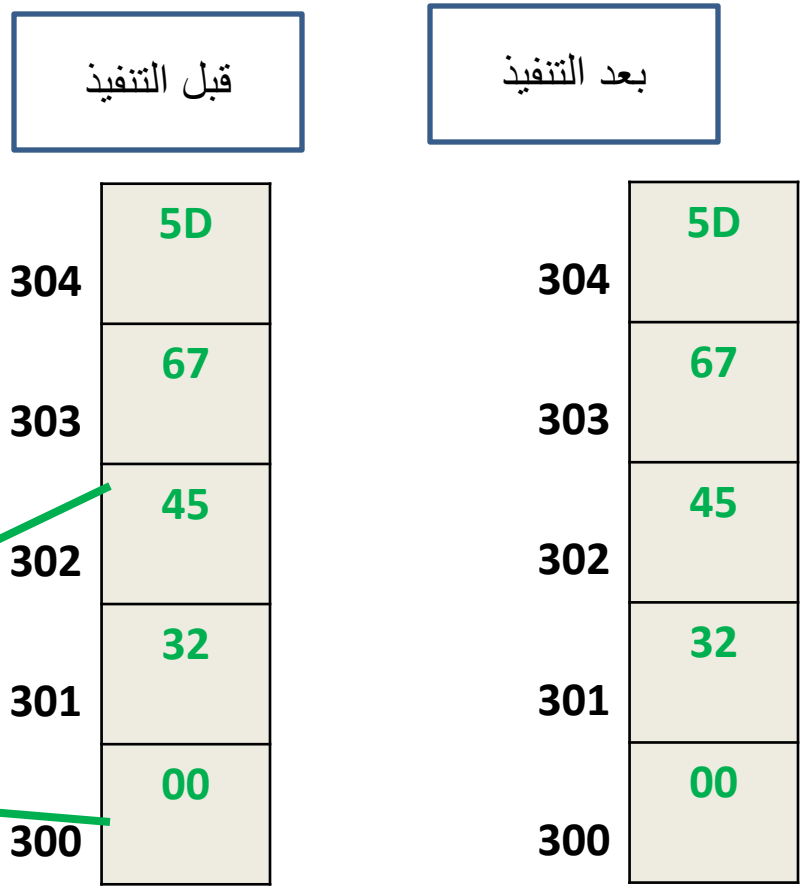


مثال:انقل محتويات الذاكرة عند العنوان ٣٠٠,302 الى المسجل AH,BL على التوالي

قيمة المسجلين AH,BLقبل التنفيذ
AH=00,BL=00

```
MOV   AH , [300 H]
MOV   BL , [302 H]
```

قيمة المسجلين AH,BLبعد التنفيذ
AH=00
BL=45



انماط العنونة

١. المسجلية
٢. الفورية
٣. المباشرة
٤. المسجلية غير مباشرة
٥. النسبية القاعدية
٦. النسبية الدليلية
٧. النسبية الدليلية القاعدية

المسجلية

- تستخدم المسجلات لتخزين البيانات ولايستخدم الذاكرة في هذا النمط
- مثال:

- MOV BX, DX

- MOV ES, AX

- MOV BH, DL

الفورية

يقوم بنقل قيمة ثابتة الى المسجلات

MOV AX, 672C H

MOV BH, 41 H

MOV BX , 2219 H

ماعدا مسجلات المقاطع يجب نقل القمة الفورية الى مسجل عام ومن ثم الى مسجل المقطع

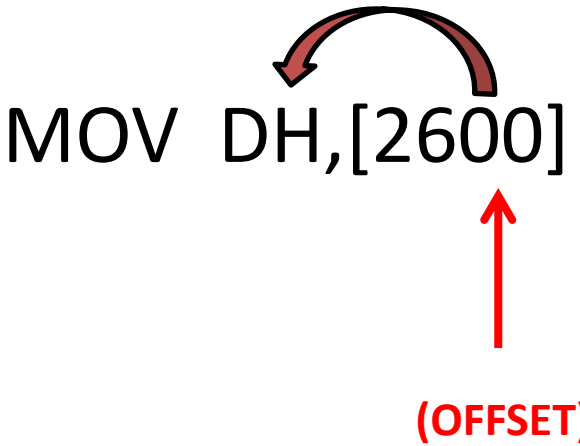
مثال

انقل القيمة ٦٥٤٧ الى المسجل DS

```
MOV  AX , 6547H
MOV  DS,  AX
```

نمط العنونة االمباشرة

- في هذا النمط توجد البيانات داخل الذاكرة ويذكر عنوان الذاكرة بصورة مباشرة داخل الايعاز مثال:



نمط العنونة المسجلية الغير مباشرة

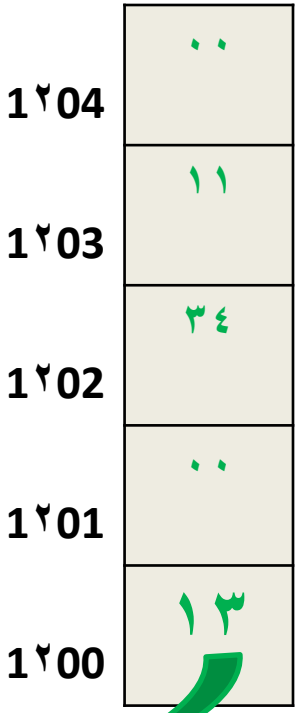
يخزن عنوان الذاكرة في احدى المسجلات SI,DI,BX

مثال:ماهي قيمة المسجل AH بعد تنفيذ هذا البرنامج

```
MOV   BX ,  1200H
MOV    AH ,  [BX]
```

يشابه عمل
الايعاز

```
MOV   AH ,  [1200H]
```

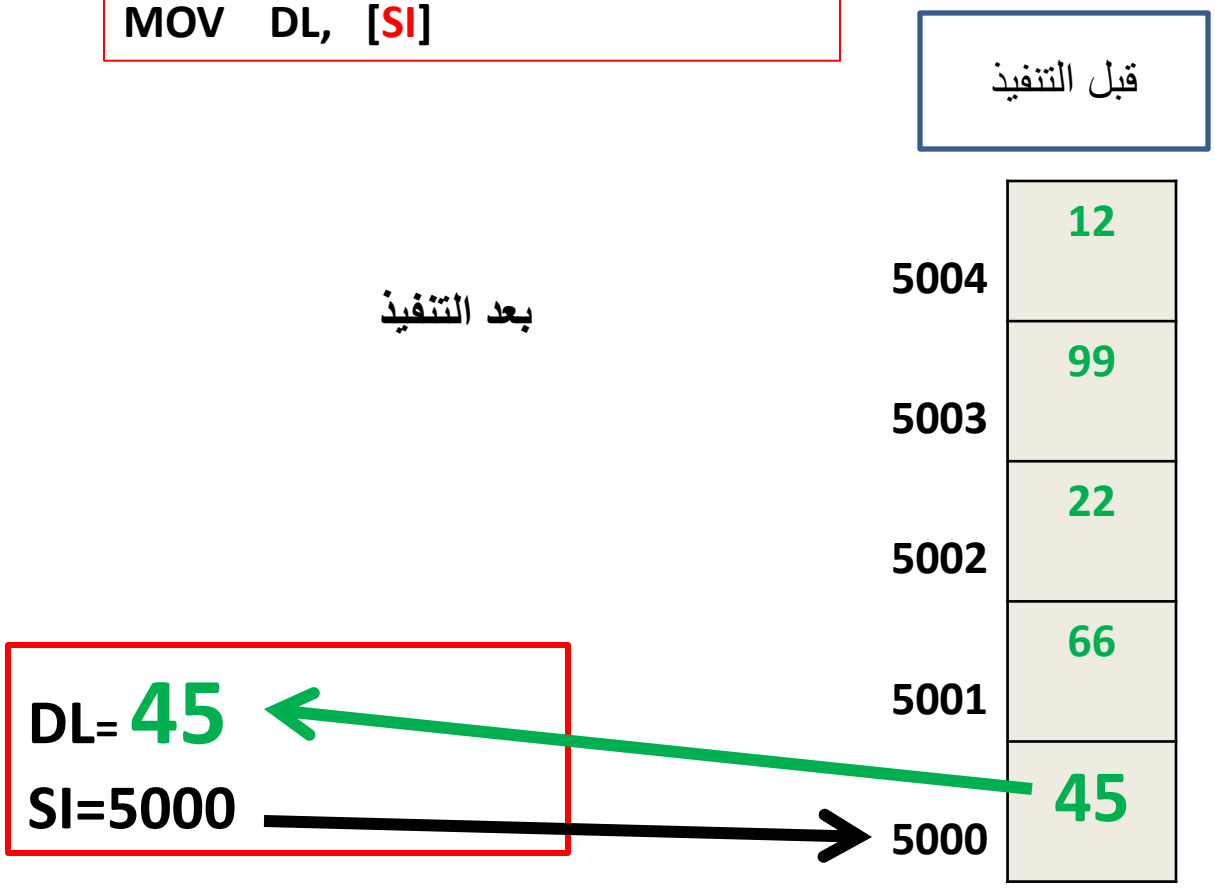


قيمة AH بعد التنفيذ هي

AH= 13

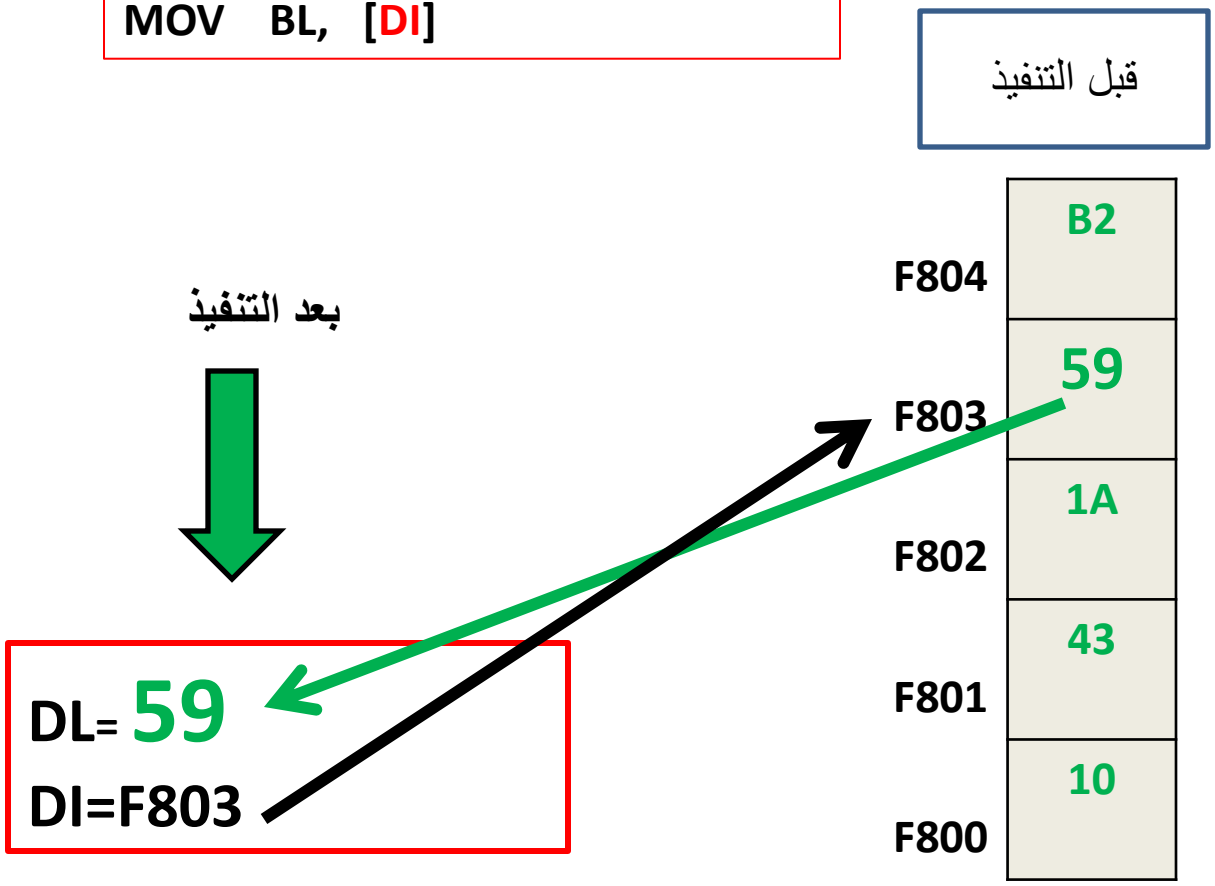
مثال:ماهو عمل هذا البرنامج

```
MOV  SI,  5000H
MOV  DL,  [SI]
```



مثال:ماهو عمل هذا البرنامج

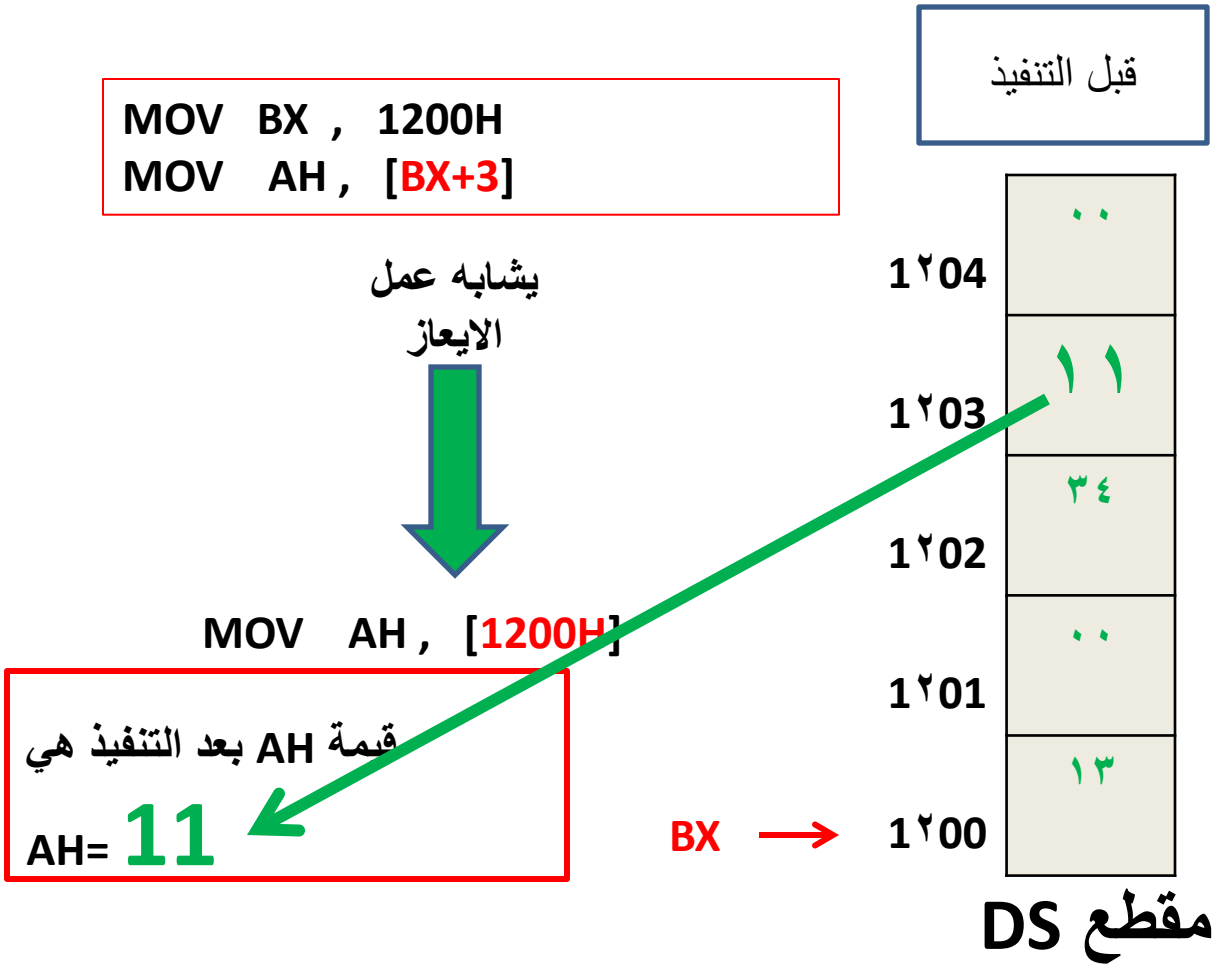
```
MOV   DI,   F803H
MOV   BL,   [DI]
```



نمط العنونة النسبية القاعدية

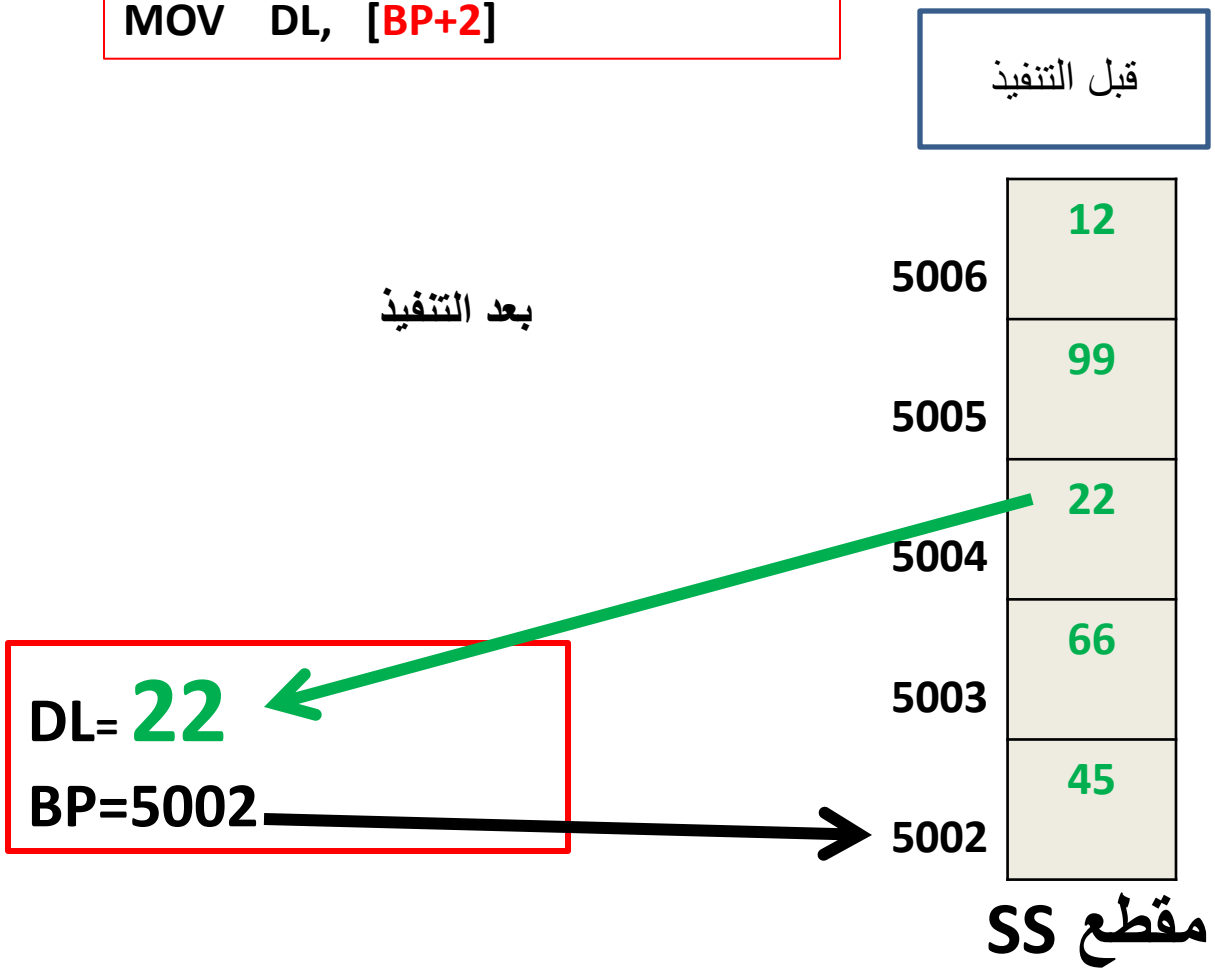
يخزن عنوان الذاكرة في احدى المسجلات BP,BX مع قيمة ازاحة

مثال:ماهو عمل هذال البرنامج



مثال:ماهو عمل هذا البرنامج

```
MOV   BP,   5002H
MOV    DL,   [BP+2]
```



نمط العنونة النسبية الدليلية

يخزن عنوان الذاكرة في احدى المسجلات SI,DI مع قيمة ازاحة

مثال:ماهو عمل هذال البرنامج

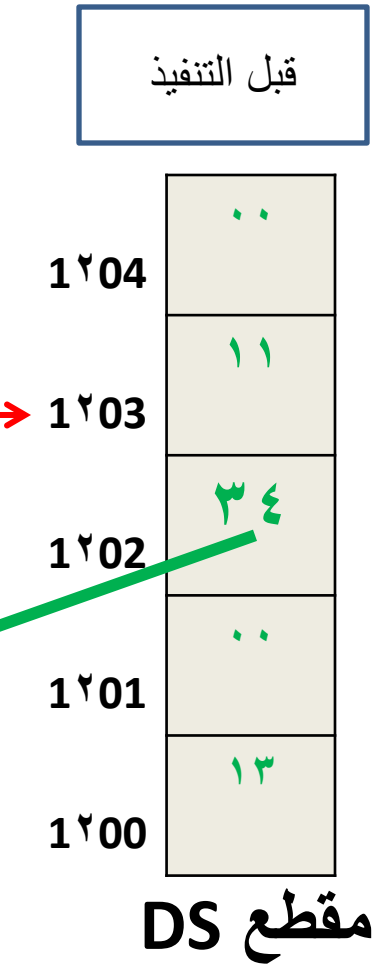
```
MOV  SI,  1203H
MOV  AH ,  [SI-1]
```

يشابه عمل
الايعاز

```
MOV  AH ,  [1202H]
```

قيمة AH بعد التنفيذ هي

AH= 34



نمط العنونة النسبيةالدليلية القاعدية

نحصل على هذا النمط منخلال دمج نمط العنونة النسبية الدليلية مع العنونة النسبيةالقاعدية

مثال:ماهو عمل هذال البرنامج

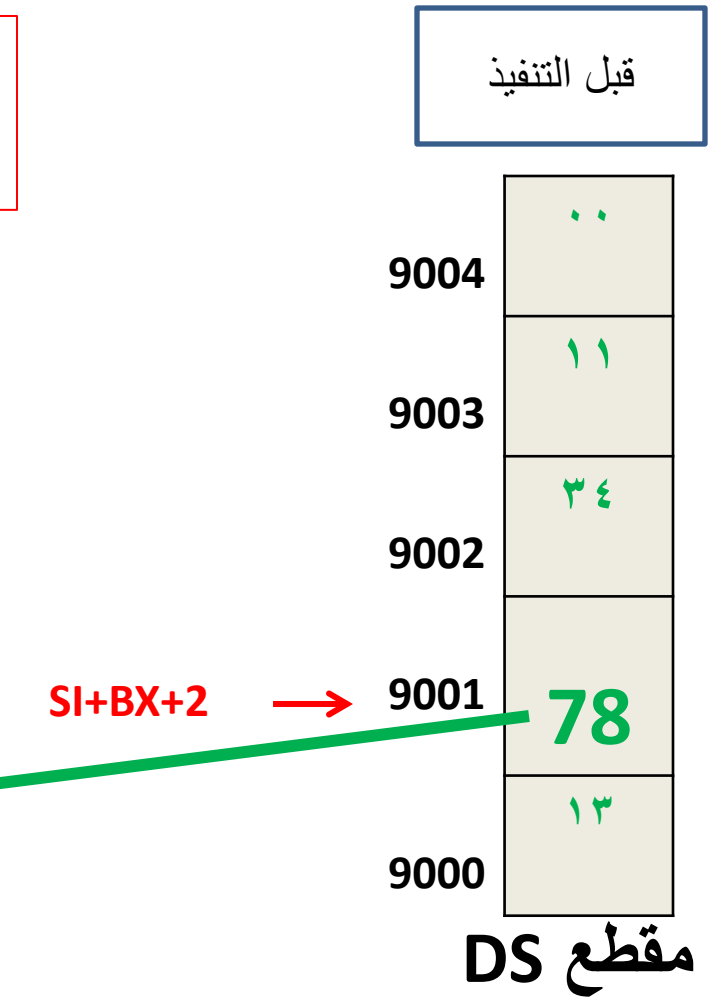
```
MOV  SI,  5001
MOV  BX,  4000
MOV   AH ,  [BX+SI+2]
```

4000 + 5001 + 2 =9001

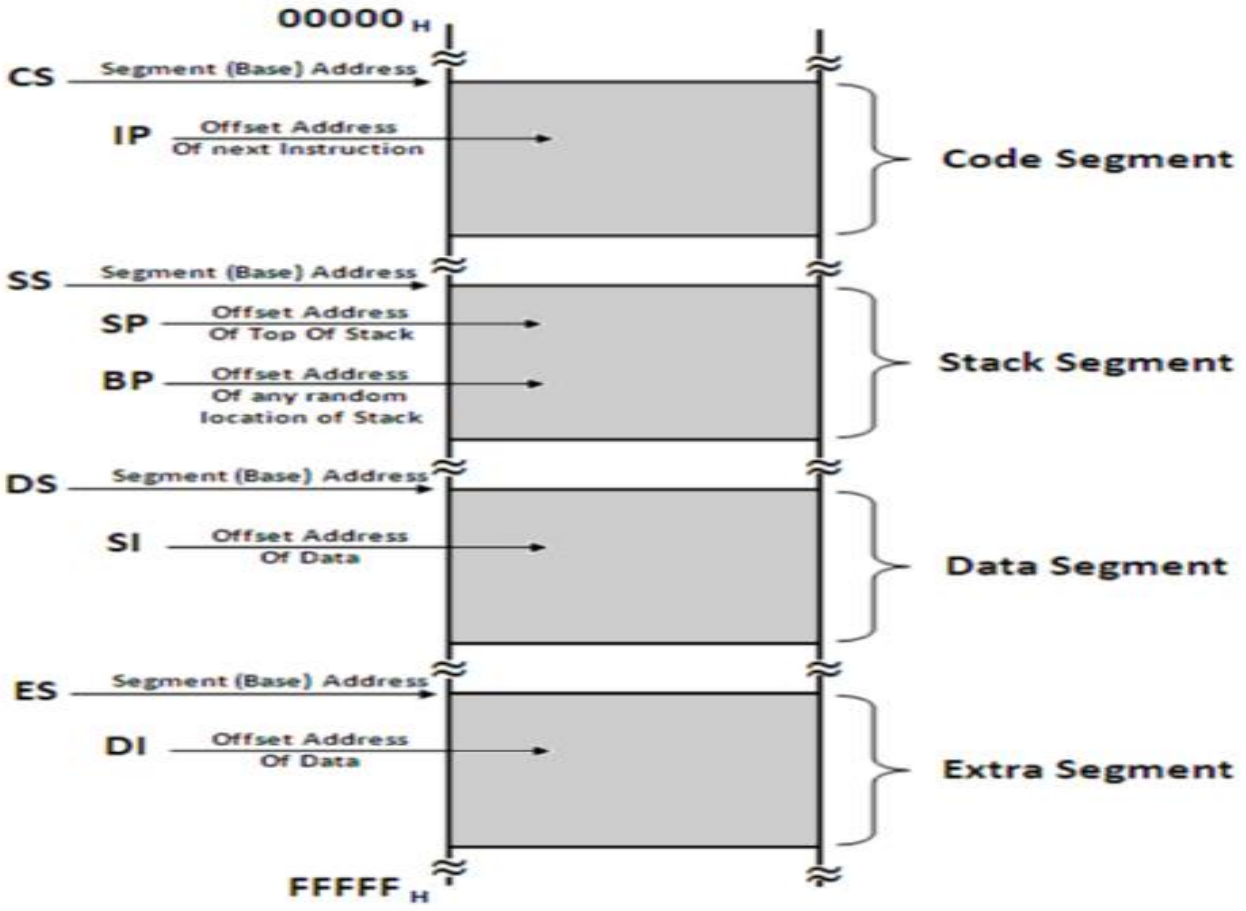
MOV AH , [9001]

قيمة AH بعد التنفيذ هي

AH= 78



مقدمة


00000 H
CS
Segment (Base) Address
IP
Offset Address
Of next Instruction
Code Segment
SS
Segment (Base) Address
SP
Offset Address
Of Top Of Stack
BP
Offset Address
Of any random
location of Stack
Stack Segment
DS
Segment (Base) Address
SI
Offset Address
Of Data
Data Segment
ES
Segment (Base) Address
DI
Offset Address
Of Data
Extra Segment
FFFFF H

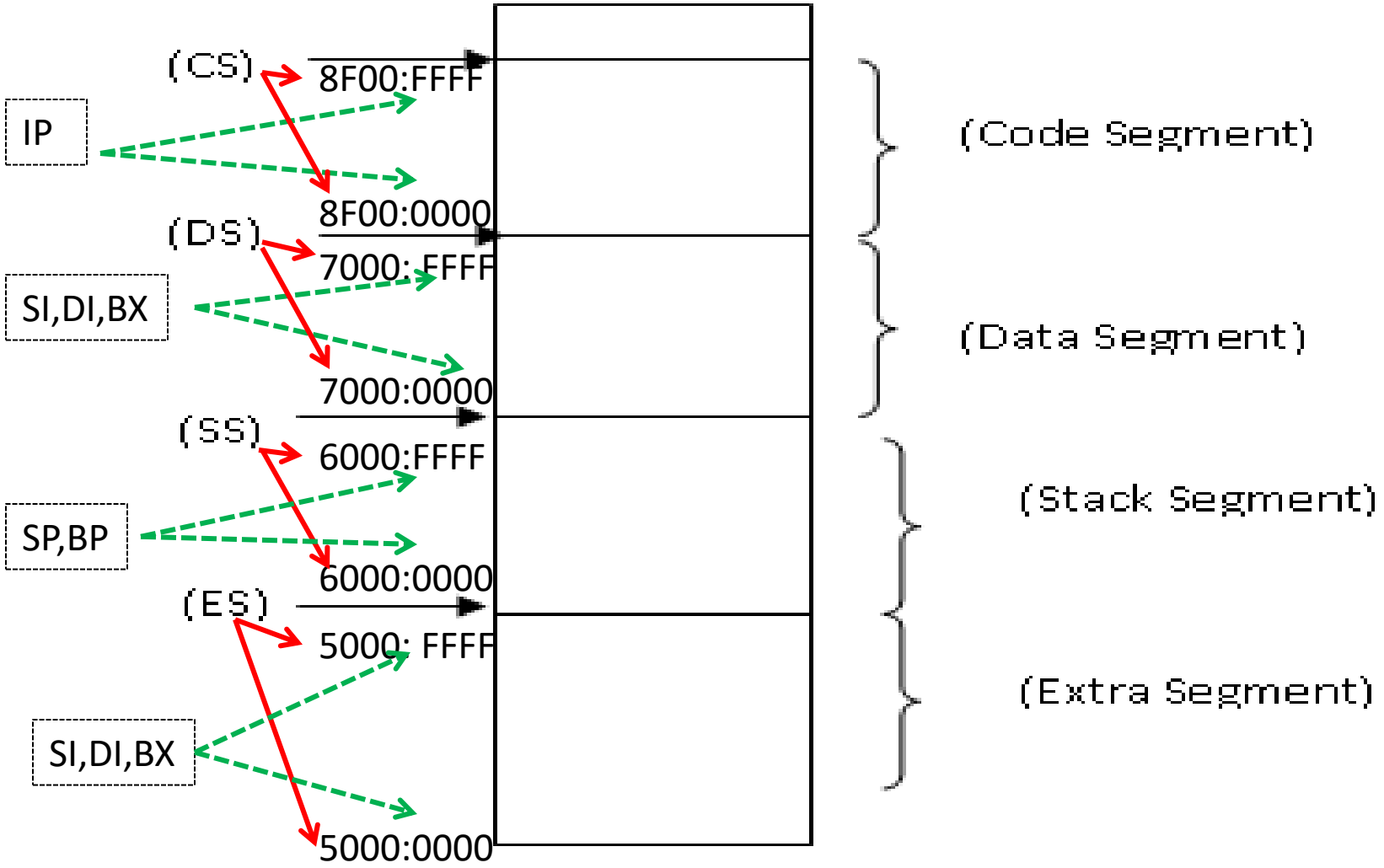


Figure: Memory Segmentation
(Code, Data, Stack and Extra Segments)

اختبار

- اذكرمسجلات مقاطع الذاكرة ومسجلات الازاحة التي تشير اليها

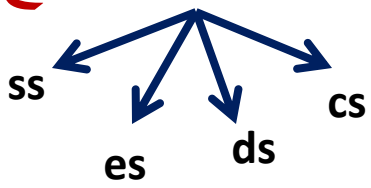
الحل (ملاحظة حفظ هذا الجدول)

مسجلات الازاحة (offset)	مسجلات المقاطع(segment)
ip	cs
Si,di,bx	ds
Si,di,bx	es
Bp,sp	ss

Physical address
العنوان الفيزيائي

- يحسب العنوان الفيزيائي من دمج عنوان الازاحة مع عنوان المقطع من خلال المعادلة التالية:

العنوان الفيزيائي = عنوان المقطع *١٠ + عنوان الازحة



ملاحظة عنوان الازاحة دايما يذكر بين اقواس

مثال: ماهو عنوان الفيزيائي الذي سيتم الوصول اليه من خلال تنفيذ الايعازات التالية ثم حدد محتوياته

اذا علمت ان DS=5000, ES=320, SS=5621

- Mov BL, 77H
- Mov [1200H],BL

- الحل:
- كتابة القانون : العنوان الفيزيائي = عنوان المقطع *١٠ + عنوان الازحة
- **عنوان المقطع: دايما مع ايعاز MOV يتم التعامل مع مسجل DS**
- **عنوان الازحة : هو كل رقم مذكور بين اقواس**
- **العنوان الفيزيائي =DS*10+1200**
- **العنوان الفيزيائي =5000*10+1200**
- **العنوان الفيزيائي =50000+1200**
- **العنوان الفيزيائي= ٥١٢٠٠**

عنوان فيزيائي		عنوان الازحة
٥١٢٠٢		١٢٠٢
٥١٢٠١		١٢٠١
٥١٢٠٠	٧٧	١٢٠٠

مثال: ماهو عنوان الفيزيائي الذي سيتم الوصول اليه من خلال تنفيذ الايعازات التالية اذا علمت ان DS=1256, ES=3440, SS=1021

- MOV SI , 2054H
- MOV DH , [SI]

- الحل:
- كتابة القانون : **العنوان الفيزيائي = عنوان المقطع *١٠ + عنوان الازحة**
- **عنوان المقطع: دايما مع ايعاز MOV يتم التعامل مع مسجل DS**
- **عنوان الازحة : هو كل رقم مذكور بين اقواس**
- **العنوان الفيزيائي =DS*10+2054**
- **العنوان الفيزيائي =1256*10+2054**
- **العنوان الفيزيائي =12560+2054**
- **العنوان الفيزيائي =145B4**

```
 1 2 5 6 0
         +
   2 0 5 4
-----------
 1 4 5 B 4
```

مثال: ماهو عنوان الفيزيائي الذي سيتم الوصول اليه من خلال تنفيذ الايعازات التالية اذا علمت ان DS=377, ES=3440, SS=1021

- MOV DI , 1A35H
- MOV AH , [DI]

- الحل
- **العنوان الفيزيائي = عنوان المقطع *١٠ + عنوان الازحة**
- **العنوان الفيزيائي =DS*10+ 1A35**
- **العنوان الفيزيائي =377*10+ 1A35**
- **العنوان الفيزيائي =3770 + 1A35**
- **العنوان الفيزيائي =51A5**

```
  3 7 7 0
      +
  1 A 3 5
-----------
  5 1 A 5
```

مثال: ماهو عنوان الفيزيائي الذي سيتم الوصول اليه من خلال تنفيذ الايعازات التالية اذا علمت ان DS=377, ES=3440, SS=1021

- MOV DI , 1A35H
- MOV AH , [DI+2]

- الحل
- **العنوان الفيزيائي = عنوان المقطع * ١٠ + عنوان الازحة**
- **العنوان الفيزيائي =(1A35+2) +DS*10**
- **العنوان الفيزيائي =(1A35+2) +377*10**
- **العنوان الفيزيائي =3770 + 1A37**
- **العنوان الفيزيائي =51A7**

```
  3 7 7 0
      +
  1 A 3 7
-----------
  5 1 A 7
```

اختبار

- **ماهو عنوان الفيزيائي الذي سيتم الوصول اليه من خلال تنفيذ الايعازات التالية اذا علمت ان DS=377, ES=3040, SS=B021**
- MOV DI , 1A35H
- MOV BX ,1000H
- MOV AH , [BX+DI+2]

- **ماهو عنوان الفيزيائي الذي سيتم الوصول اليه من خلال تنفيذ الايعازات التالية اذا علمت ان DS=4667, ES=3040, SS=B021**
- **وما هي محتويات <u>الذاكرة بعد التنفيذ</u> ؟**
- **MOV BX , 5084H**
- **MOV [BX-1] , BL**

التعامل مع الذاكرة
&
flag register

- MOV AX , 3498 H
- MOV [1200 H] , AX

AX =	AH	AL
	34	98

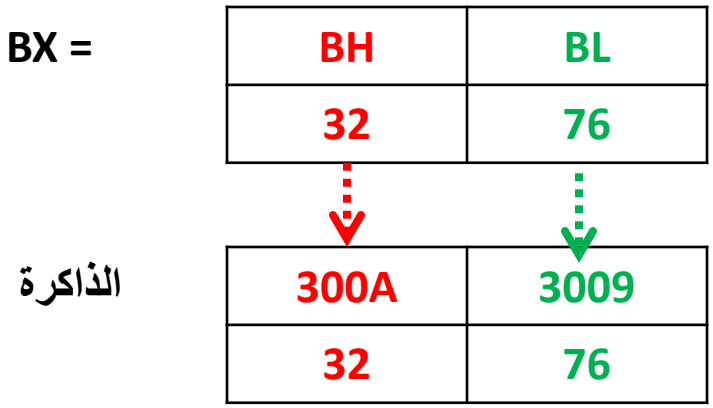
الذاكرة	١٢٠١	١٢٠٠
	٣٤	٩٨

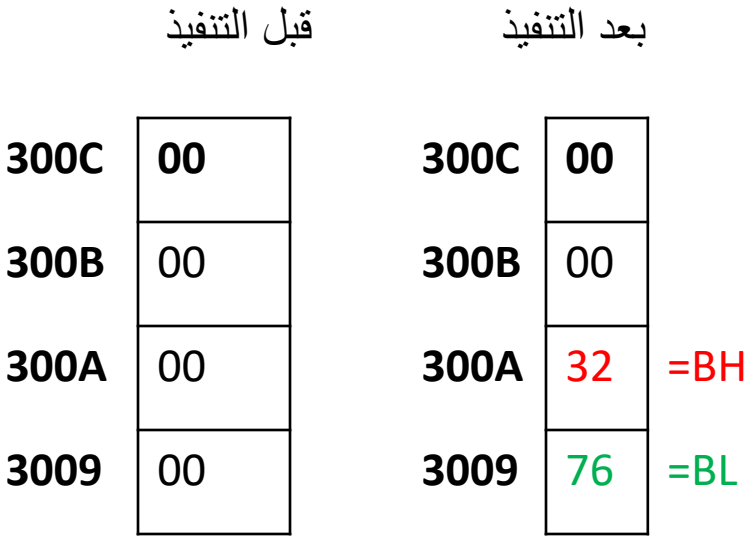
	قبل التنفيذ	بعد التنفيذ	
1203	00	00	
1202	00	00	
1201	00	٣٤	=AH
1200	00	٩٨	=AL

ملاحظة : القيمة ذات المرتبة العليا تذهب الى العنوان الكبير والقيمة ذات المرتبة الصغرى تذهب الى العنوان الصغير

مثال:ارسم محتويات الذاكرة بعد تنفيذ البرنامج التالي

- MOV BX,3276 H
- MOV [3009 H], BX



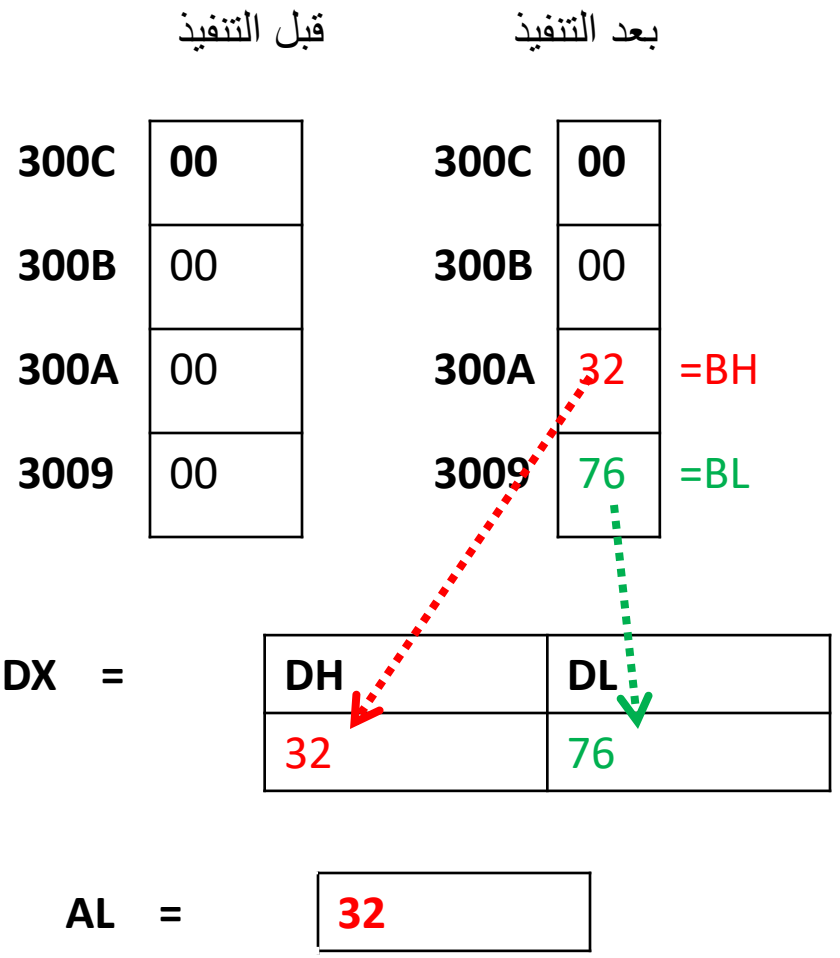


مثال:ارسم محتويات الذاكرة بعد تنفيذ البرنامج التالي وما هي قيمة AL ,DX

- MOV BX,3276 H
- MOV [3009 H] ,BX
- MOV DX,[3009H]
- MOV AL, [300AH]

BX =	BH	BL
	32	76

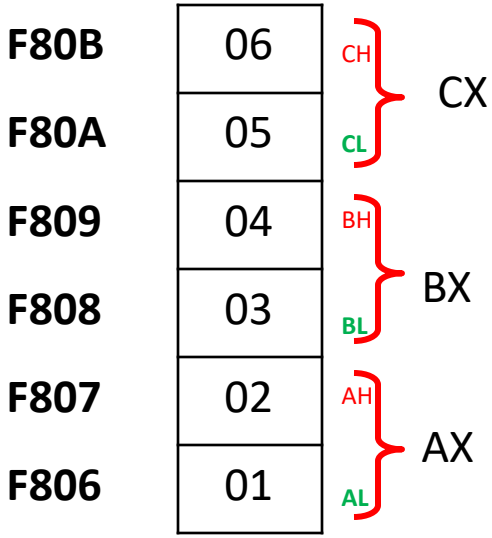
الذاكرة	300A	3009
	32	76



مثال: اكتب برنامج بلغة التجميع لخزن القيم 1,2,3,4,5,6 في الذاكرة بدأ من العنوان F806

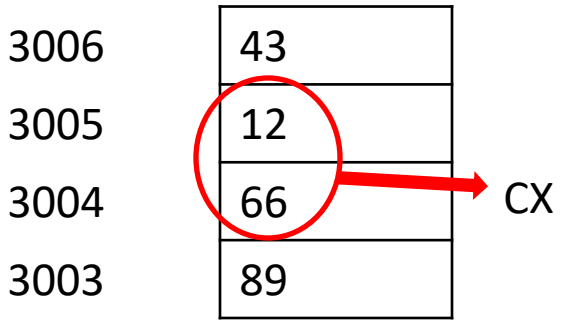
```
MOV   AX , 0102H
MOV   [F806 H],ax

MOV  BX , 0304H
MOV   [F808 H],bx

MOV  CX , 0506H
MOV   [F809 H],cx
```

بعد التنفيذ



اختبار

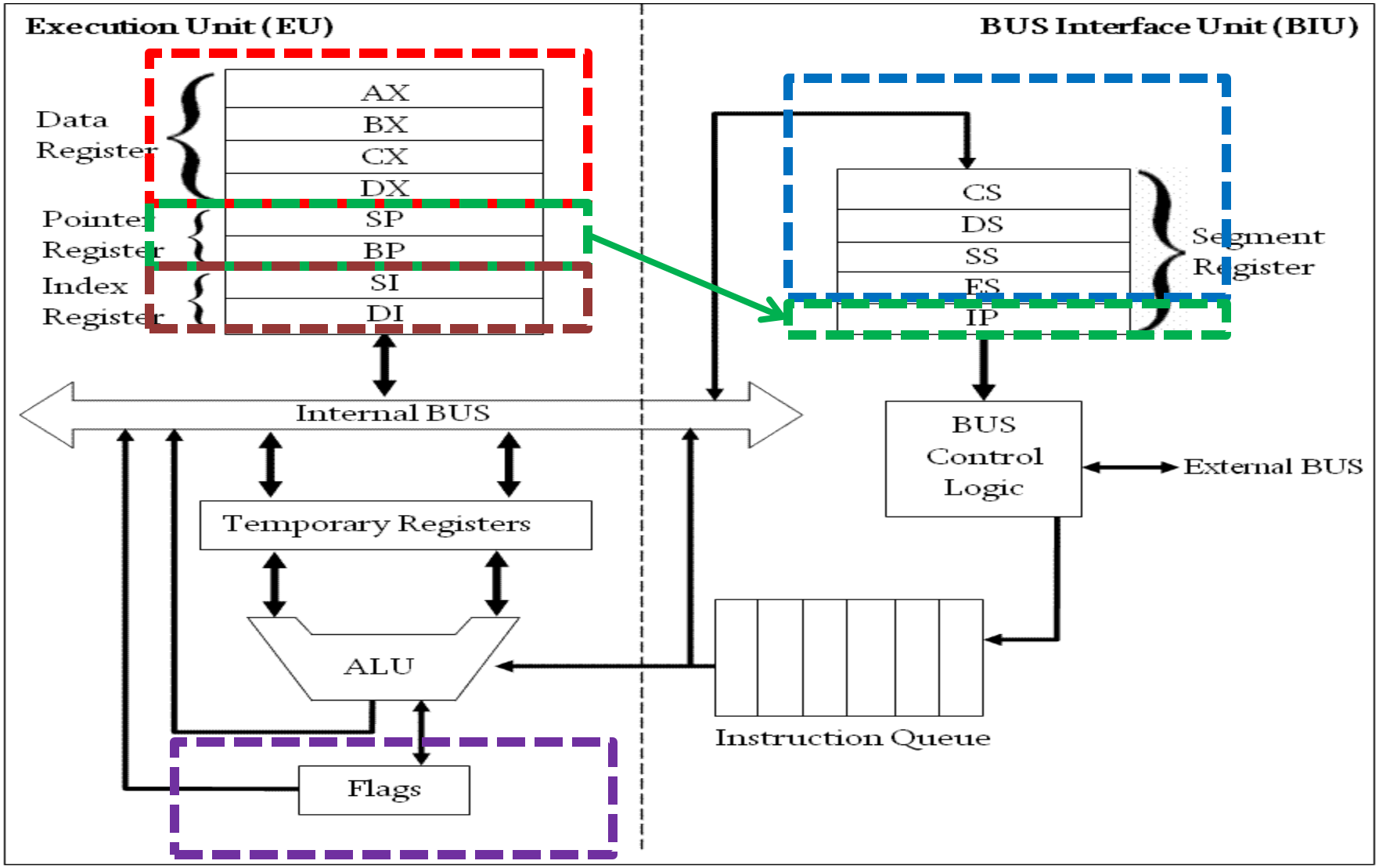
- س١) لديك الذاكرة التالية انقل القيمة ٦٦و١٢ الموجودة امامك في الذاكرة الى المسجل CX(اي ان قيمة CX=1266 بعد تنفيذ برنامجك)



- س٢) ماهي محتويات الذاكرة وماهي قيمة AX بعد تنفيذ البرنامج التالي

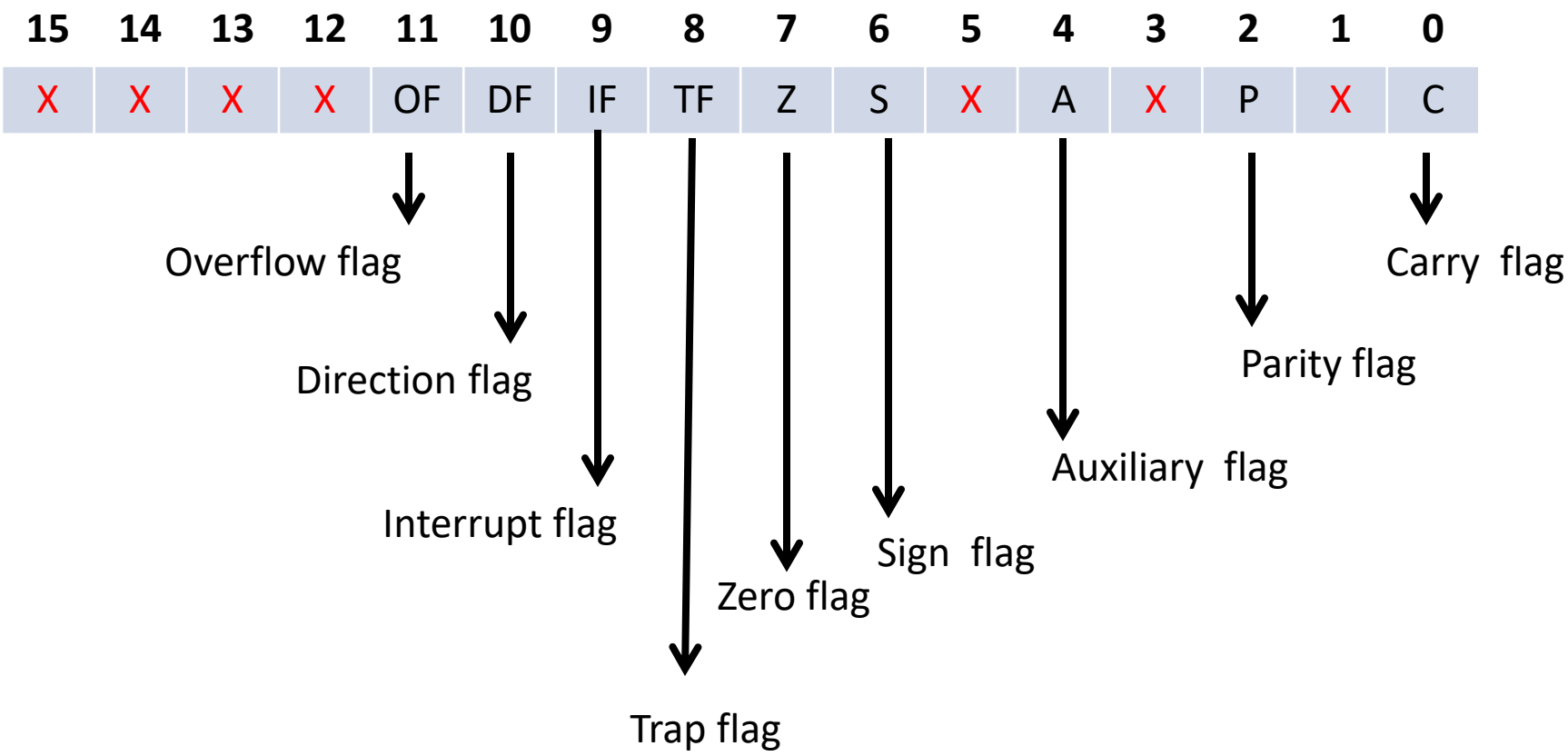
- MOV BX, 654H
- MOV AX ,1100 H
- MOV [2388 H] ,BX
- MOV AH , [2388 H]

Flag register



flag register ماهو

هو مسجل حجمه ١٦ بت(2 byte) يحتوي على اعلام تتاثر وتؤثر بالعمليات الرياضية والمنطقية



علامة x هو بت غير مستخدم

- Carry flag: هذا العلم هو بت يشير الى ١ اذا كان نتيجة العملية يوجد تحميل من البت ٧ /١٥ ويشير الى ٠ اذا لايوجد تحميل
- parity flag : هذا العلم هو بت يشير الى ١ اذا كان نتيجة العملية (الرياضية او المنطقية) تحتوي على عدد زوج من «١» ويشير الى ٠ اذا اذا كان نتيجة العملية (الرياضية او المنطقية) تحتوي على عدد فردي من «١»

- Auxiliary flag: هذا العلم هو بت يشير الى ١ اذا كان نتيجة العملية يوجد تحميل من البت (٤ /٧) ويشير الى ٠ اذا لايوجد تحميل

- Zero flag: هذا العلم هو بت يشير الى ١ اذا كان نتيجة العملية صفرية ويشير الى ٠ اذا كان نتيجة العملية غير صفرية
- sign flag: هذا العلم هو بت يشير الى ١ اذا كان نتيجة العملية سالبة ويشير الى ٠ اذا كان نتيجة العملية غير موجبة
- trap flag : هذا العلم هو بت يشير الى ١ اذا كان التنفيذ خطوة واحدة فقط
- Interrup flag: هذا العلم هو بت يشير الى ١ اذا كان هناك قطع

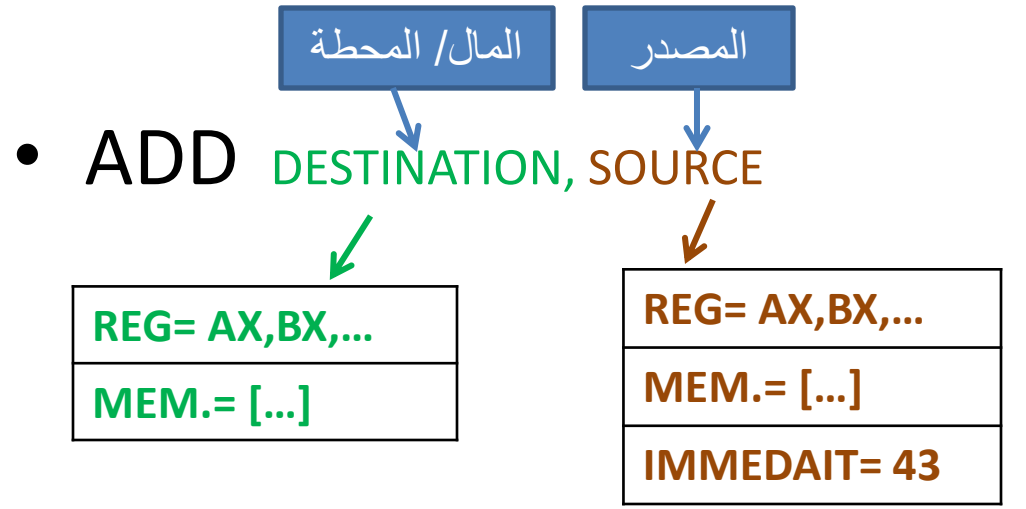
- direction flag : هذا العلم هو بت يشير الى ١/٠ اشارة الى اتجاه نقل البيانات الى الاعلى او الى الاسفل

- overflow flag : هذا العلم هو بت يشير الى ١ اشارة الى النتيجة كبيرة جدا

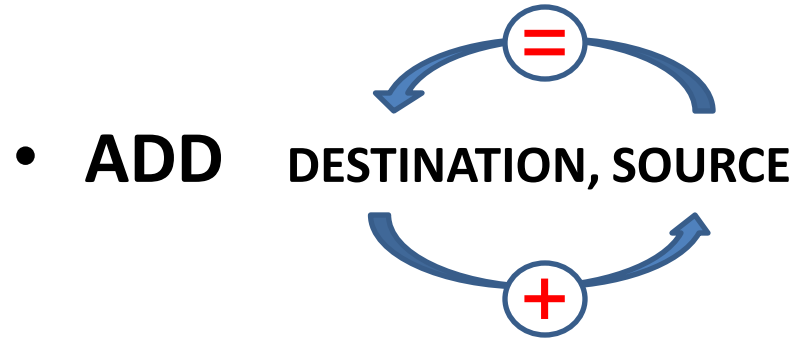
ايعاز الجمع

ADD

- الصيغة للايعاز



- عمل الايعاز



DESTINATION =DESTINATION +SOURCE

مثال:ماهي قيمة AL بعد تنفيذ الايعازات التالية؟

- MOV AL, 72H
- ADD AL , 19 H

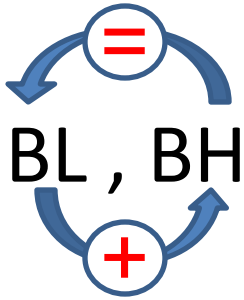
- الحل:
- من الايعاز الاول
- الايعاز الثاني

AL = 72

AL = AL + 19
AL= 72 + 19
AL = 8B

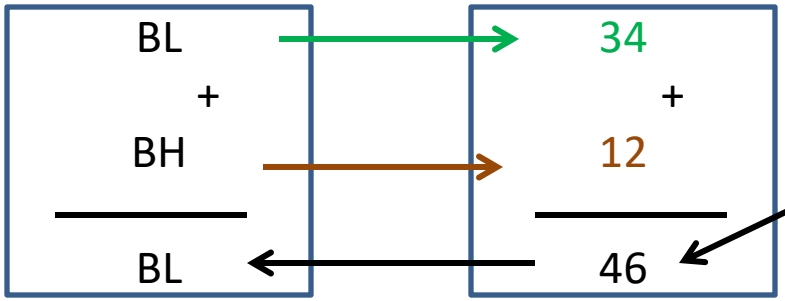
مثال:ماهي قيمة BXبعد تنفيذ الايعازات التالية؟

- MOV BX , 1234H



- ADD BL , BH

BX =	BH	BL
	12	34



BX =	BH	BL
	12	46

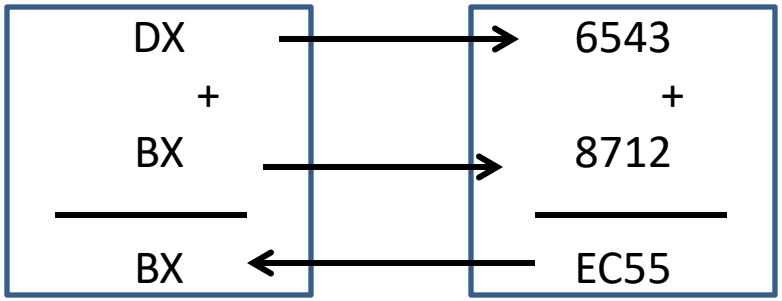
BX =1264

مثال:ماهي قيمة BXبعد تنفيذ الايعازات التالية؟

MOV DX , 6543 H

MOV BX , 8712 H

ADD BX ,DX

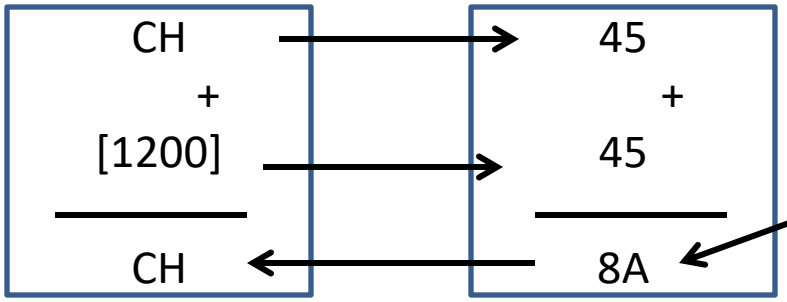


مثال:ماهي قيمة CXبعد تنفيذ الايعازات التالية؟

- MOV CX ,4532 H
- MOV [1200 H] , CH
- ADD CH , [1200 H]

1202	**00**
1201	00
1200	45

CX=	CH	CL
	45	32

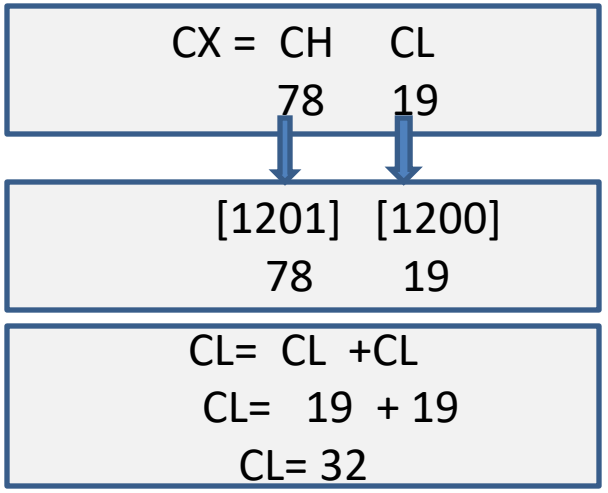


CX=	CH	CL
	8A	32

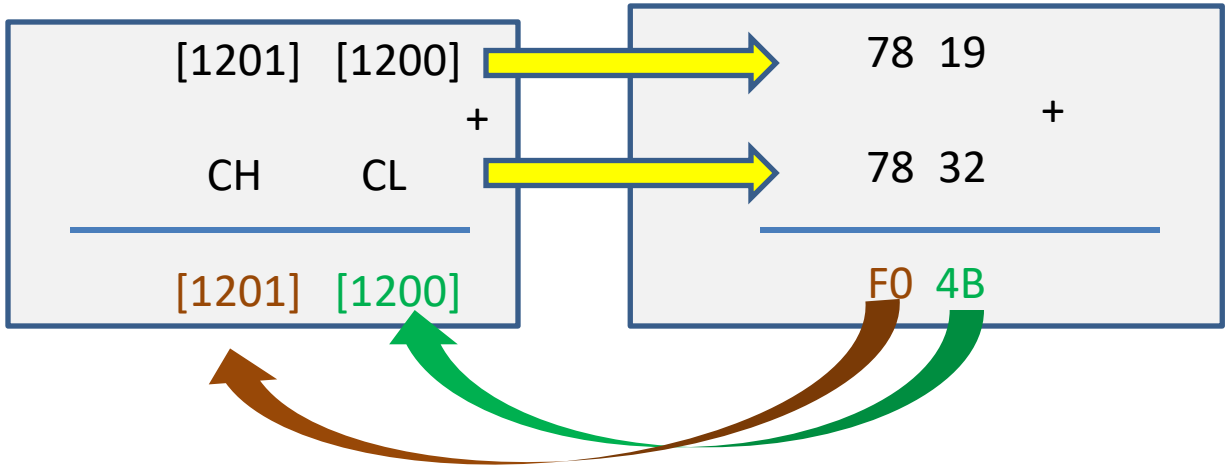
CX=8A32

مثال:ماهي قيمة CXوماهي محتويات الذاكرة بعد تنفيذ البرنامج التالي؟

- Mov cx ,7819 h
- Mov [1200 h] , cx
- ADD CL,CL
- ADD [1200 H] ,CX



1202	**00**
1201	F0
1200	4B

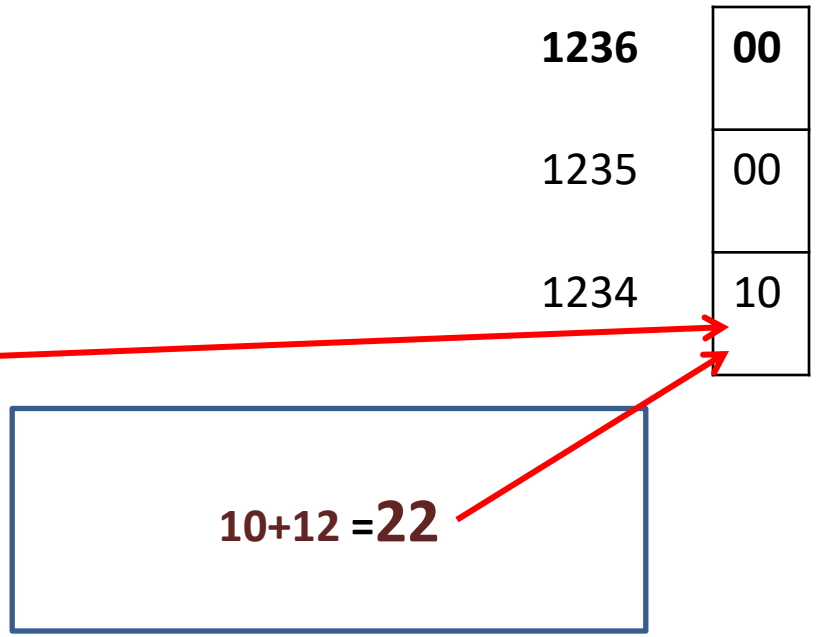


حساب العنوان الفيزيائي مع الايعاز ADD

مثال:احسب العنوان الفيزيائي الذي سيصل اليه البرنامج التالي اذا علمت ان DS=8000H, وماهي محتويات الذاكرة بعد تنفيذ البرنامج

- MOV BX,1234H
- ADD [BX],BH

- P.A.=DS*10+ الازاحة
- P.A.=80000 + 1234
- P.A.= 81234



الايعاز SUB

- SUB: هو طرح قيمتين من بعضهما وتكون الصيغة العامة للايعاز هي
- SUB DEST., SOURCE

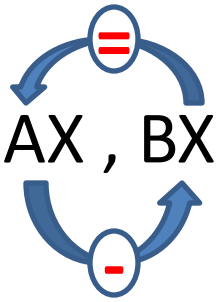
REG.	AX,BX.. SI,DI.. AL,BH..
MEM.	[..]

REG.	AX,BX.. SI,DI.. AL,BH..
MEM.	[..]
IMM.	12,123 4…..

- مثال
- SUB AX, BX
- SUB [3009H],AX
- SUB [4400H],DH
- SUB CX,3456H
- SUB SI,[3009H]

عمل الايعاز SUB

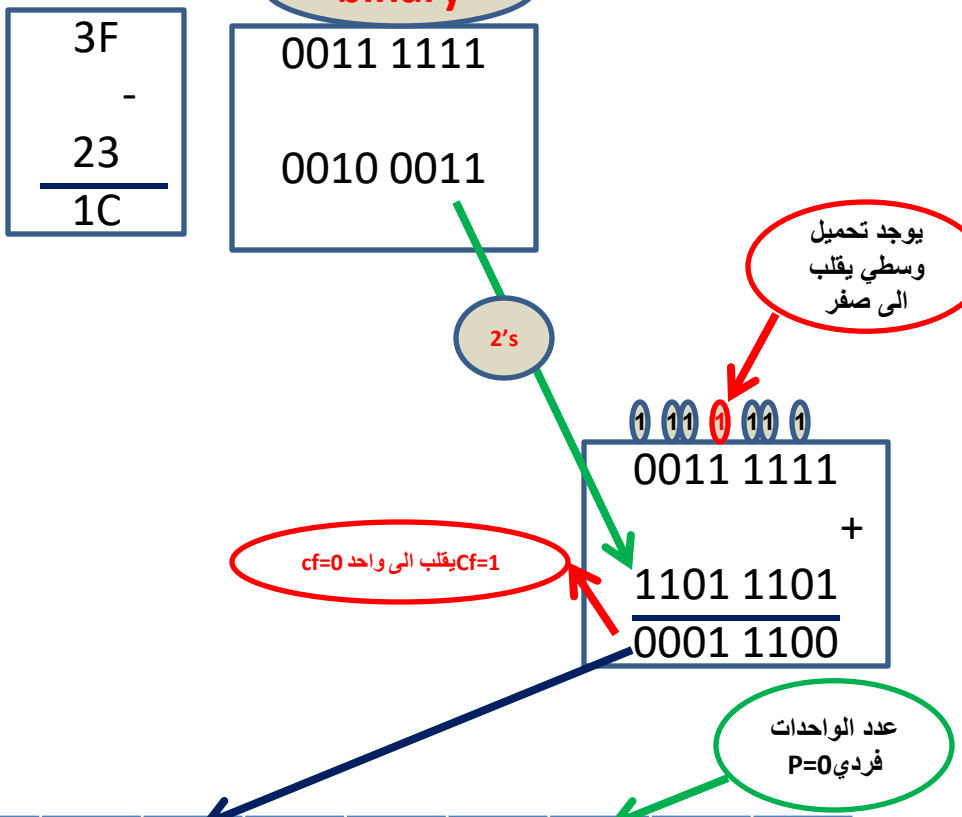
- SUB AX , BX



- AX=AX-BX

- مثال
  - MOV AL,3F
  - MOV BH,23

SUB AL,BH



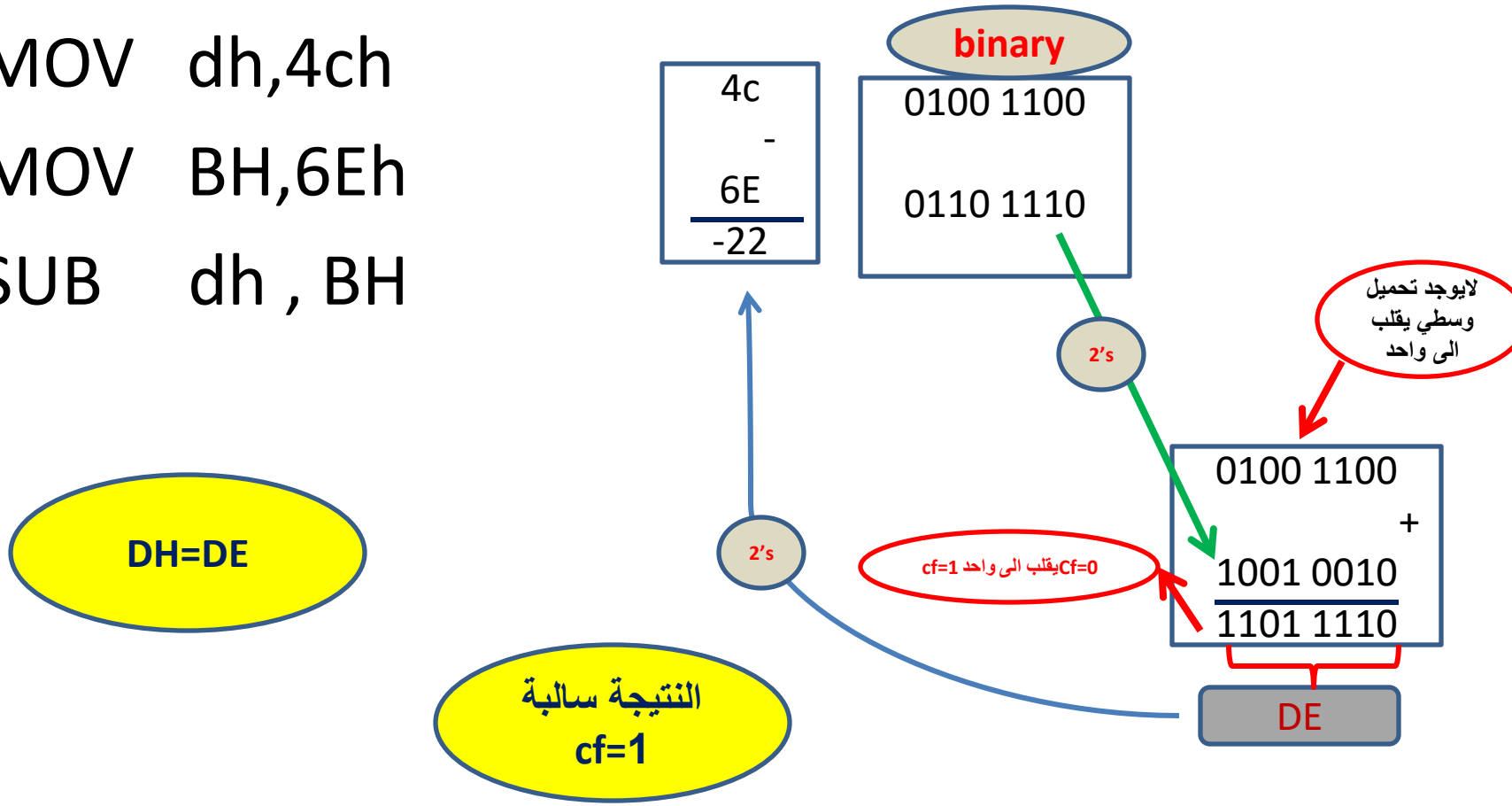
- خطوات الطرح

1. خذ المتمم الثاني (2' complement) للمطروح(بعد الفارزة)
2. اجمعه مع المطروح منه(قبل الفارزة.)
3. اقلب علم المحمل(CF)والمحمل الوسطي(AF)

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	0	0	0	0	0

مثال:ماهو تاثير البرنامج التالي على مسجل الاعلام وماهي قيمة المسجل dh

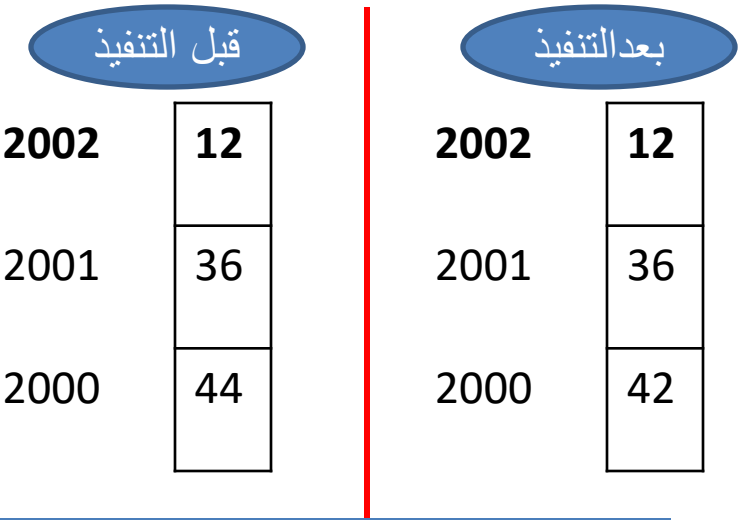
- MOV dh,4ch
- MOV BH,6Eh
- SUB dh , BH



X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	1	0	1	0	1	0	1

اكتب برنامج لانقاص محتويات الذاكرةعند العنوان 2000 بمقدار (2)

- MOV DH,02H
- SUB [2000H],DH



- ماهو العنوان الفيزيائي الذي سيصل اليه البرنامج اعلاه اذا علمت ان DS=A003

- P.A.=DS*10+ الازاحة
- P.A.= A0030+ 2000
- P.A.= A2030

اختبار

- س1)اكتب برنامج لحل المعادلة التالية باستخدام لغة التجميع علما ان الارقام مكتوبة بالنظام العشري

  (اخزن النتيجة في المسجل BX)

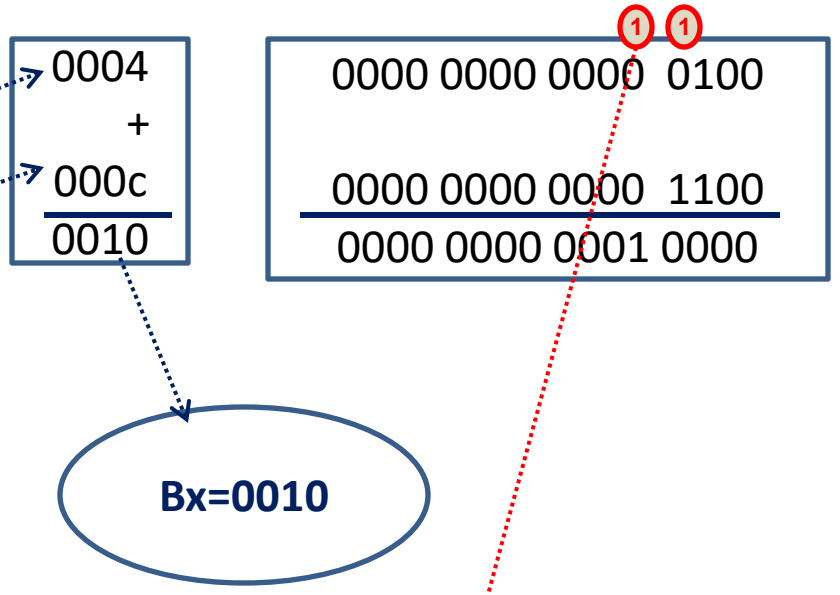
- (4+12)-20
- س2)ارسم مسجل الاعلام (FLAG REGISTER)بعد تنفيذ البرنامج اعلاه مع تفسير وتوضيح لكل علم

حل الاختبار

س1)اكتب برنامج لحل المعادلة التالية باستخدام لغة التجميع علما ان الارقام مكتوبة بالنظام العشري
(اخزن النتيجة في المسجل BX)

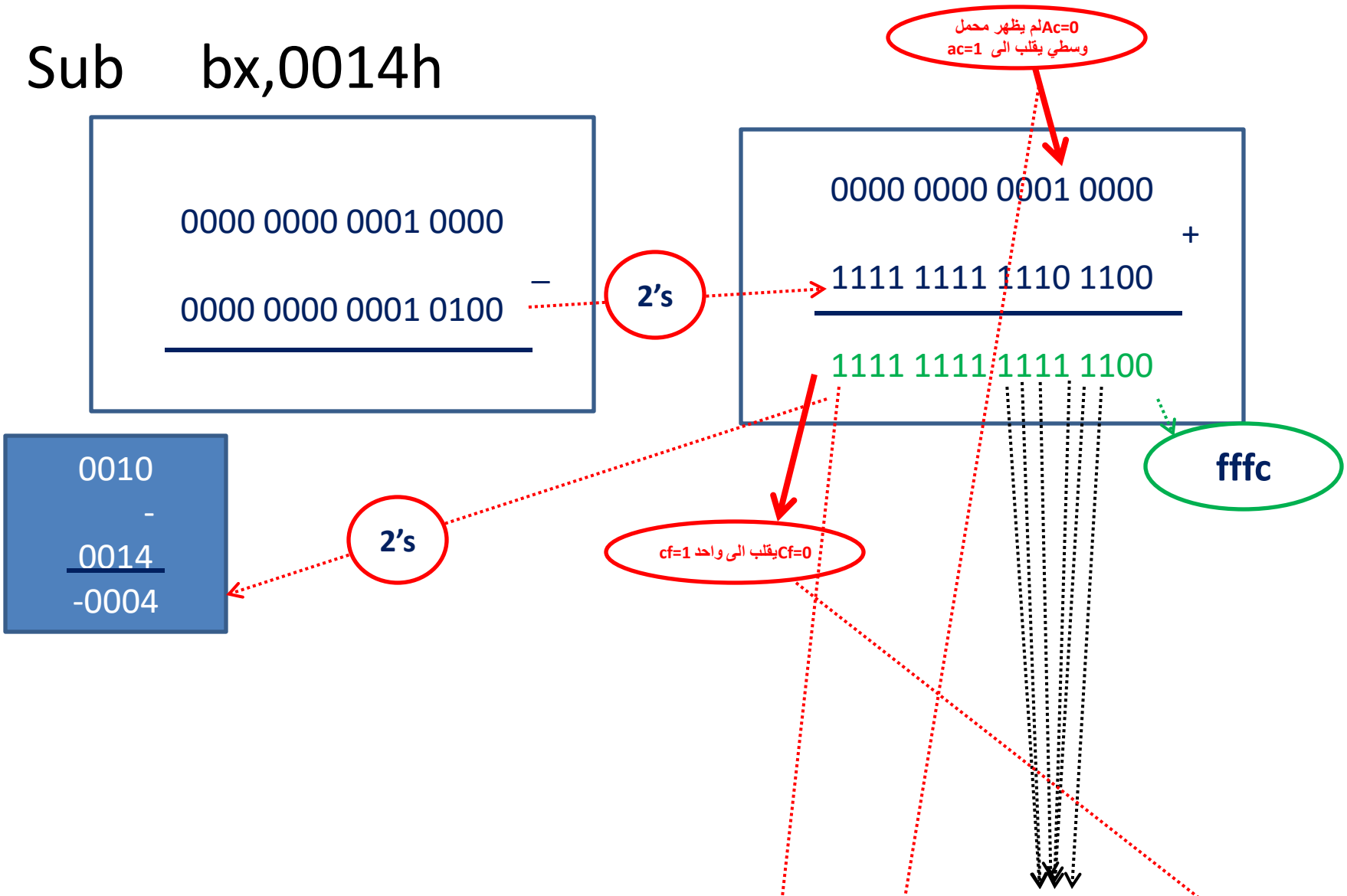
(4+12)-20

- Mov bx, 04h
- Mov ax, 0ch
- Add bx, ax
- Sub bx,0014h



X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	1	0	0	0	0

- Sub bx,0014h



X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	1	0	1	0	1	0	1

ايعاز الضرب MUL

- صيغة الايعاز

- MUL OPER2.

REG.	AL,BH..
MEM.	[..]

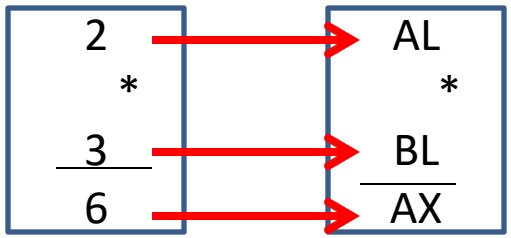
- عمل الايعاز:وضع الرقم الاول(oper1) في المسجل AL والرقم الثاني في (.OPER2)اما مسجل او ذاكرة والنتيجة تخزن في المسجل AX

- AX = AL * OPER2.

مثال اكتب برنامج لضرب العددين 2*3

- نضع احد العددين في المسجل AL
- نضع العدد الثاني في اي مسجل او ذاكرة(مثلا نختار المسجل BL)

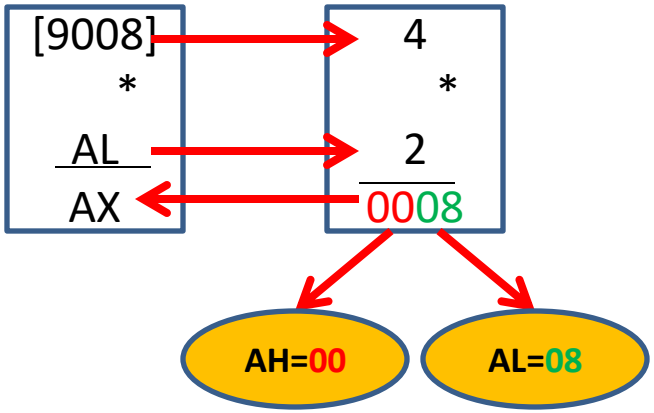
- MOV AL, 2H
- MOV BL,3H
- MUL BL

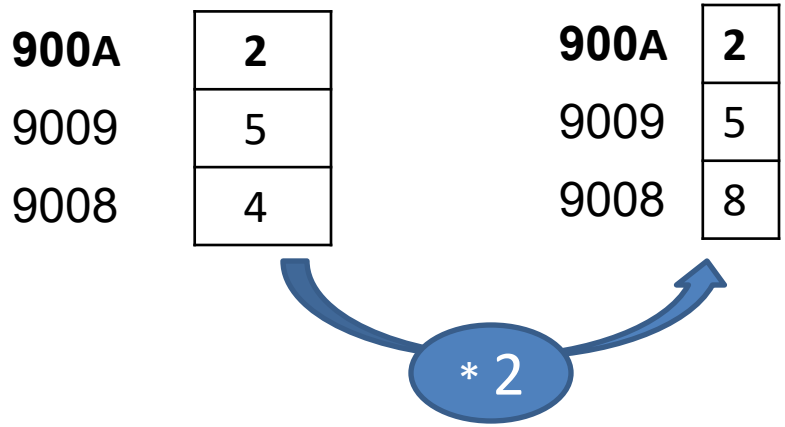


- النتيجة : AX=0006

اكتب برنامج لمضاعفة محتويات الذاكرة عند العنوان[9008]
(ملاحظة محتويات الذاكرة قيمها عشوائية)؟

```
MOV   AL ,02H

MUL  [9008H]

MOV  [9008H] , AL
```

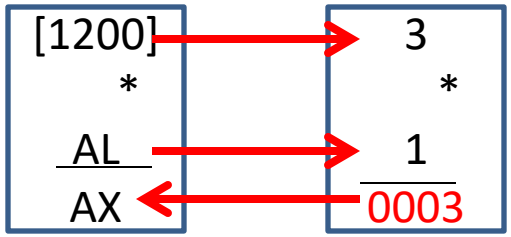




ماهو العنوان الفيزيائي الذي سيصل اليه الايعاز الاخير وماهي قيمة المسجل AXعلما DH=3

- MOV AX,8901H
- MOV DS, AX → DS=8901
- MOV [1200H] , DH
- MUL [1200]

- P.A.=DS*10+ الازاحة
- P.A.=8901*10+ 1200
- P.A.=89010+1200
- P.A.=8A210



اختبار

- اكتب برنامج لضرب محتويات العنوانين [6008]مع[7006] مع بعضهما وضع النتيجة في المسجل BX(ملاحظة محتويات الذاكرة قيمها عشوائية)
- ماهو العنوان الفيزيائي الذي سيصل اليه البرنامج اعلاه اذا علمت ان قيمة المسجل DS=8000

- البرنامج ادناه يقوم بضرب ثلاثة ارقام التالية مع بعضها 2*4*3 ويضع النتيجة في المسجل bx

المطلوب:- يوجد ايعاز ناقص اكتب هذا الايعاز ليكتمل البرنامج

```
MOV   AL, 2H
MOV  BL, 4 H
MUL  BL
MOV   DH, 3H
MUL  DH
```

المكدس(stack)

- المكدس : هو مقطع في ذاكرة RAM ويستخدم من قبل المعالجة المركزية للتخزين المؤقت للمعلومات, يشير الى هذا المقطع مسجلين مؤشر المقطع للمكدس SS ومؤشر ازاحة للمكدس SP ويجب تحميل هذين المسجلين بقيم قبل تخزين اواسترجاع معلومات من المكدس ويتم ذلك عن طريق عمليتين هما
- PUSH/POP

PUSH

- PUSH: هي عملية تخزين قيم داخل المكدس وتتم بخطوات ثابتة:

1. يتم انقاص محتويات المسجل SP بمقدار واحد
2. حشر الجزء العلي من معامل الايعاز PUSHفي المكدس

[SP-1]=[high]

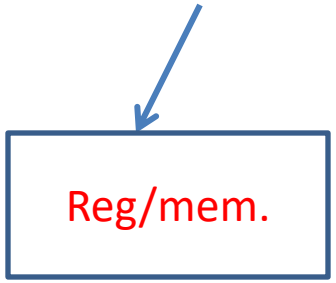
3.يتم انقاص محتويات المسجل SP بمقدار اثنان

4. حشر الجزء الواطىء من معامل الايعاز PUSHفي المكدس

[SP-2]=[low]

- صيغة الايعاز push

- Push operand

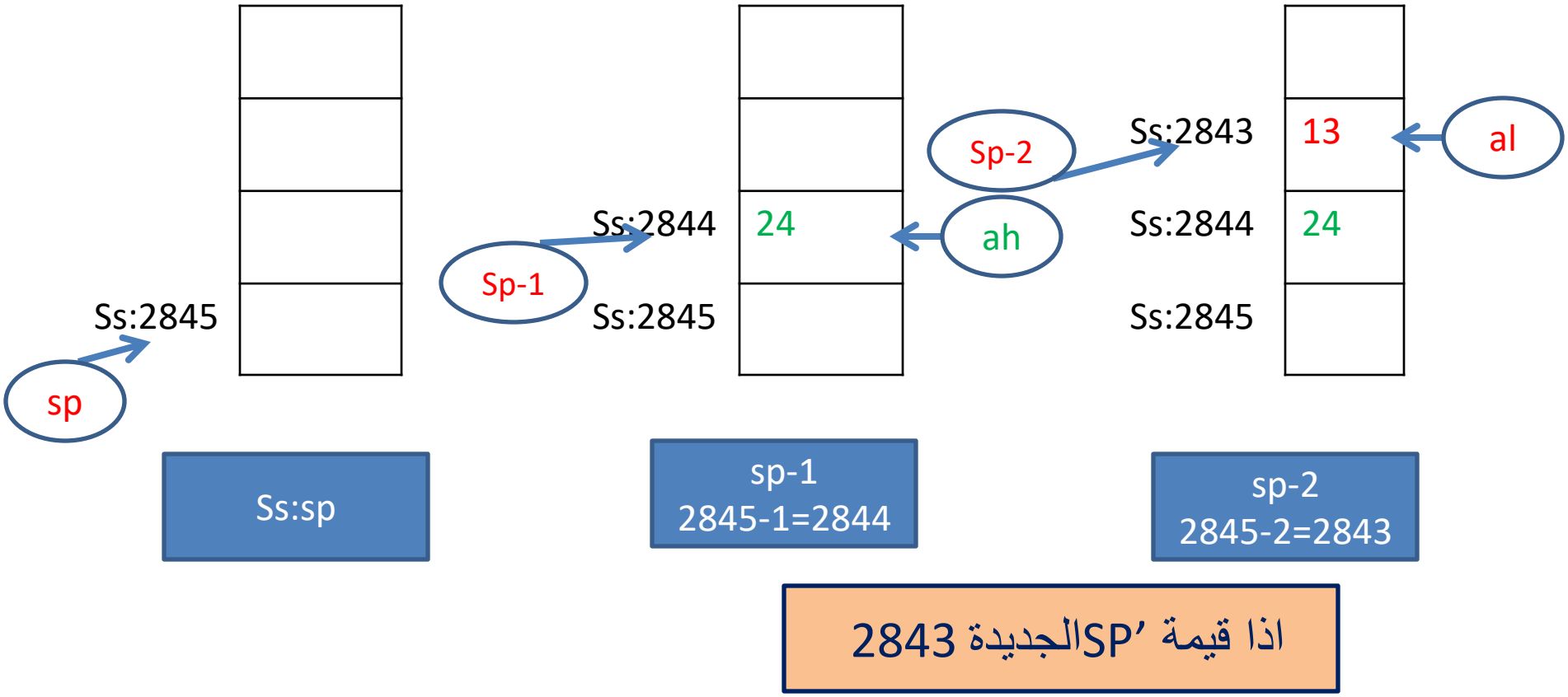


- مثال

- Push ax
- Push bx
- Push [1234]

مثال: لنفرض ان sp=2845h,ax=2413
بين محتويات المكدس وقيمة sp بعد تنفيذ التعليمات التالية

- Push ax

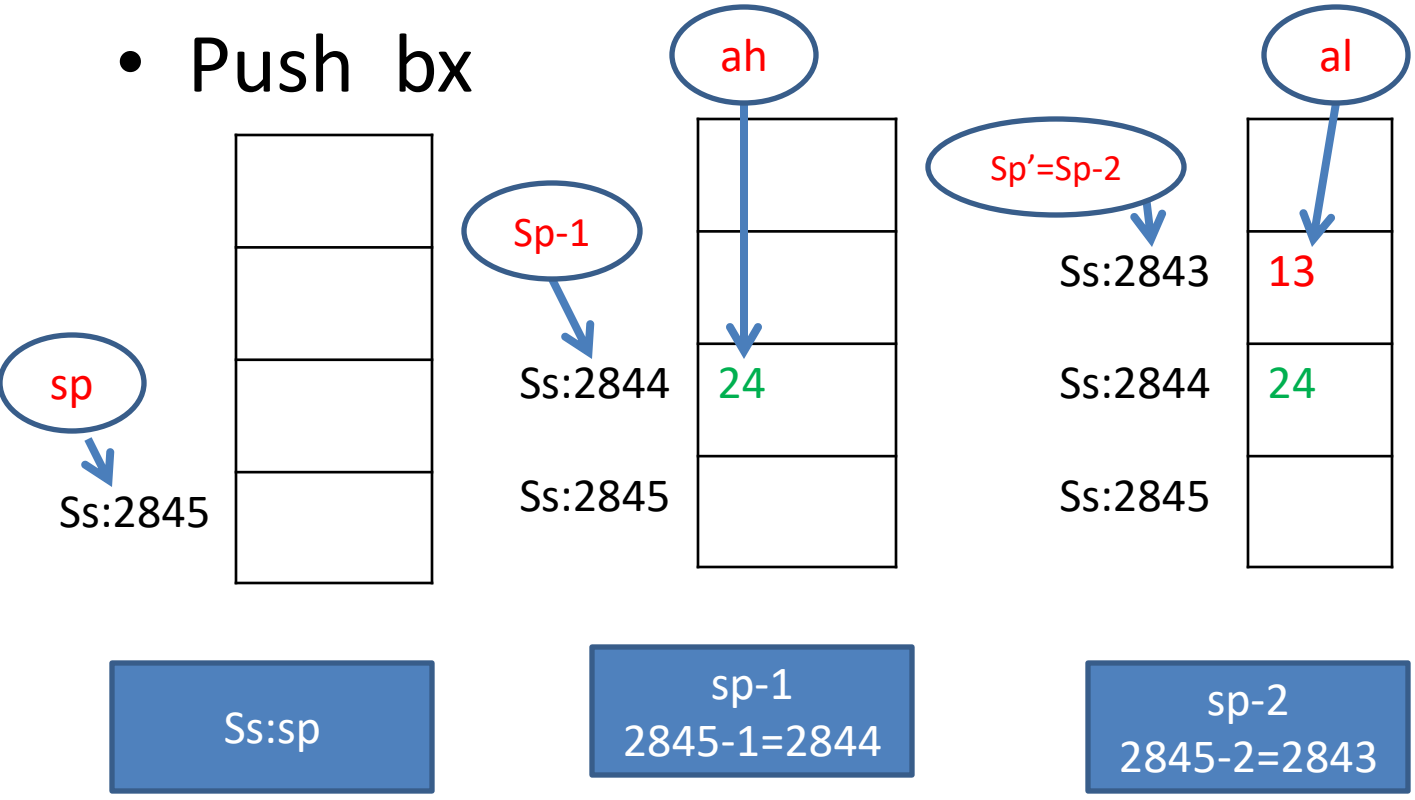


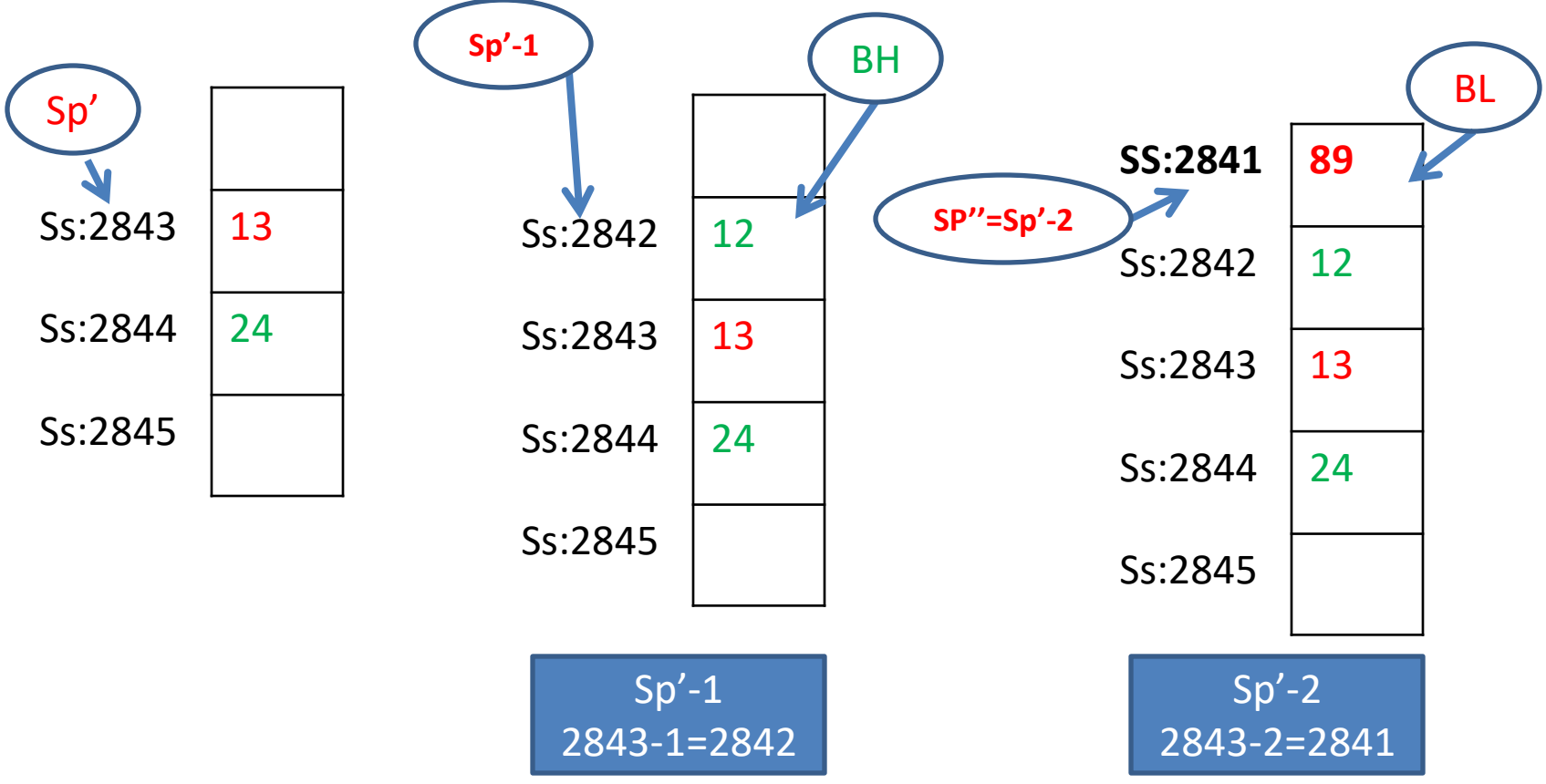
مثال: لنفرض ان

bx=1289,sp=2845h,ax=2413

بين محتويات المكدس بعد تنفيذ التعليمات التالية

- Push ax
- Push bx





اذا قيمة ’’SPالجديدة 2841

خزن محتويات الذاكرة داخل الstack

- Push [mem.]
  - مثال: اذا علمت ان محتويات الذاكرة عند ds عند العنوان [9004]=8B,[9005]=33 وان قيمة sp=2007 ماهي محتويات المكدس بعد تنفيذ البرنامج التالي
- Push [9004]
- Push [9004]

PUSH

- PUSH: هي عملية تخزين قيم داخل المكدس وتتم بخطوات ثابتة:

1. يتم انقاص محتويات المسجل SP بمقدار واحد
2. حشر الجزء العالي من معامل الايعاز PUSHفي المكدس

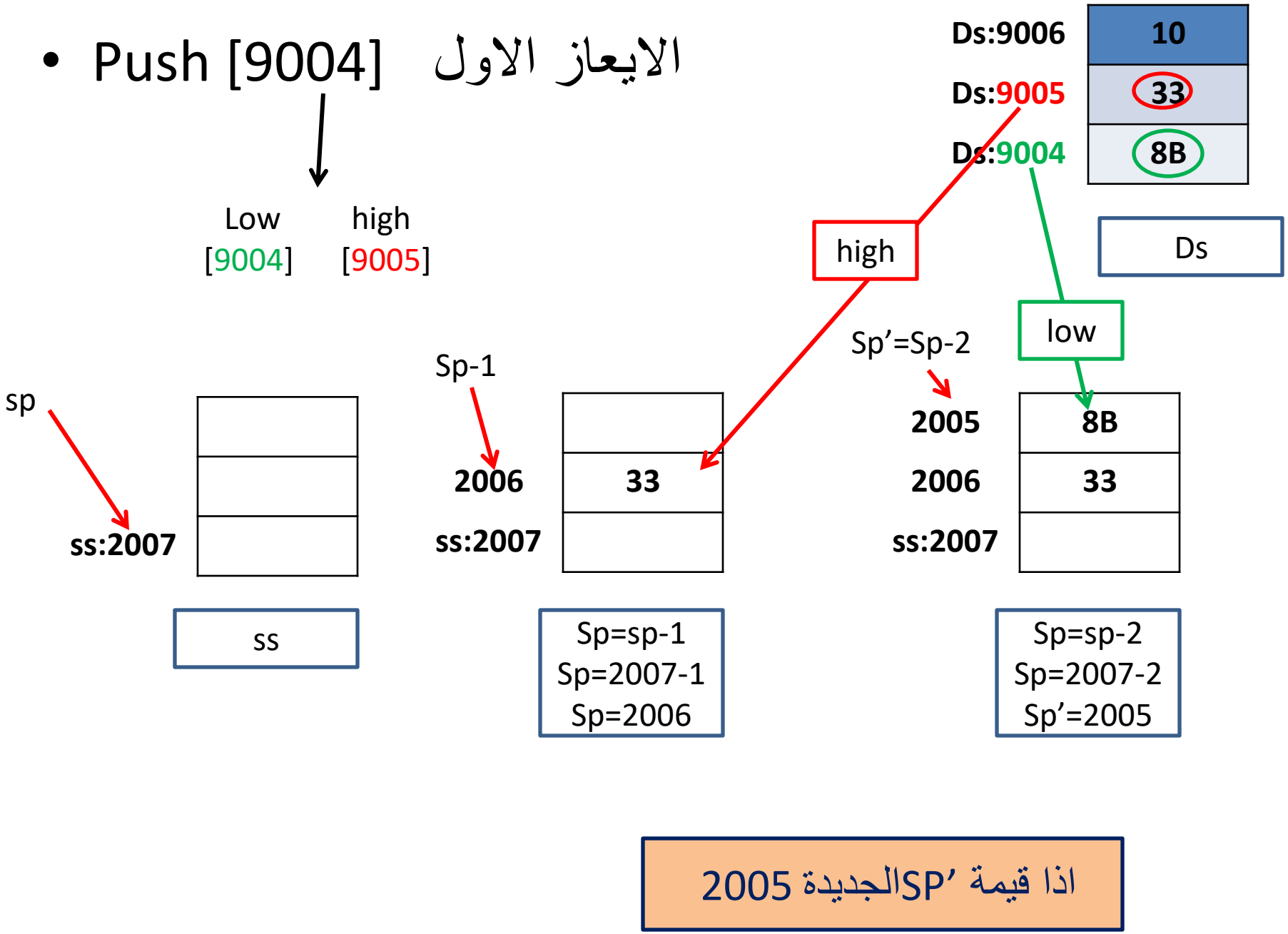
[SP-1]=[high]

3.يتم انقاص محتويات المسجل SP بمقدار اثنان

4. حشر الجزء الواطىء من معامل الايعاز PUSHفي المكدس

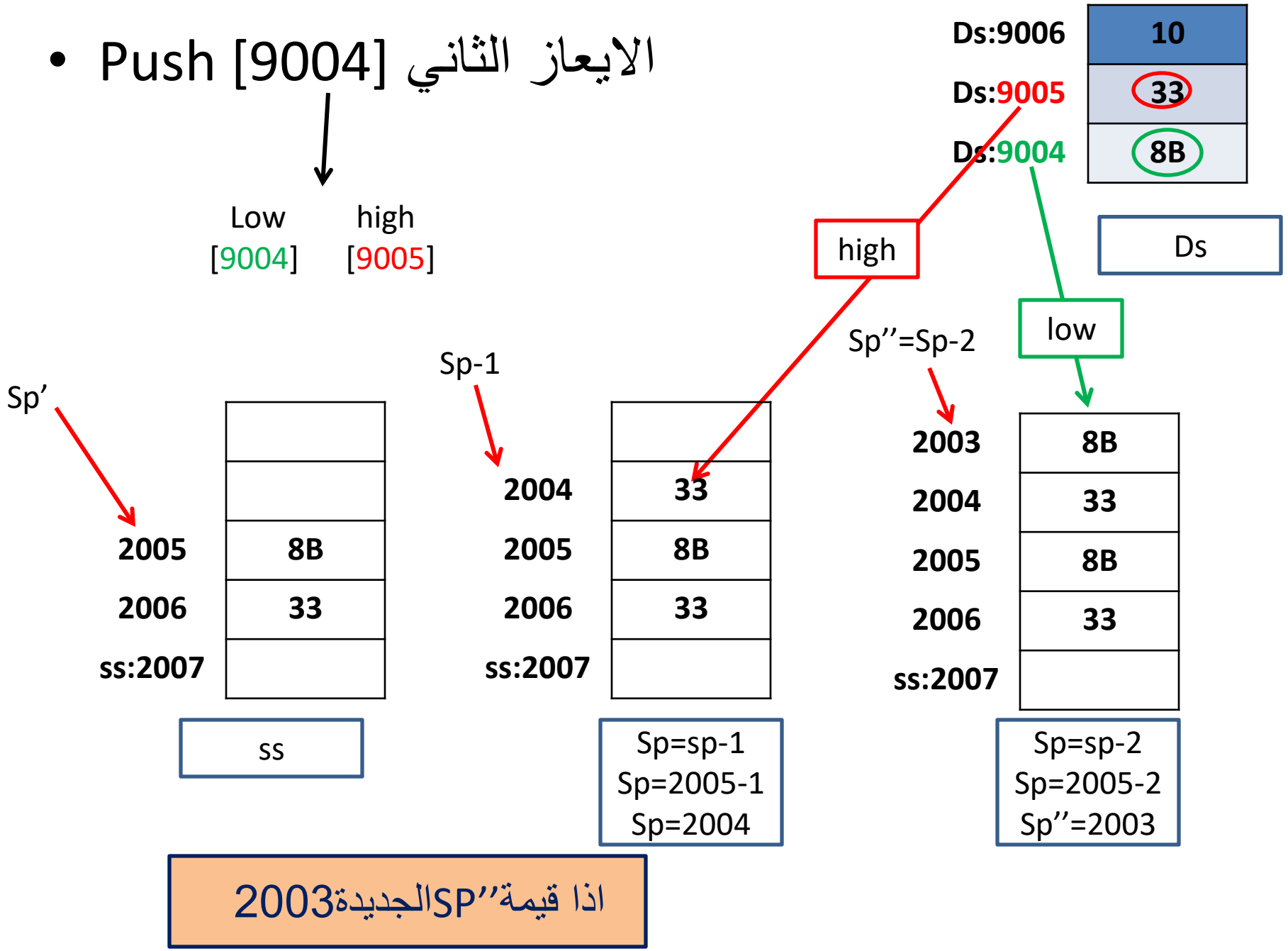
[SP-2]=[low]

- Push [9004] الايعاز الاول



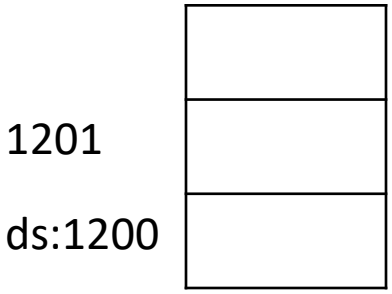
اذا قيمة SP'الجديدة 2005

- Push [9004] الايعاز الثاني

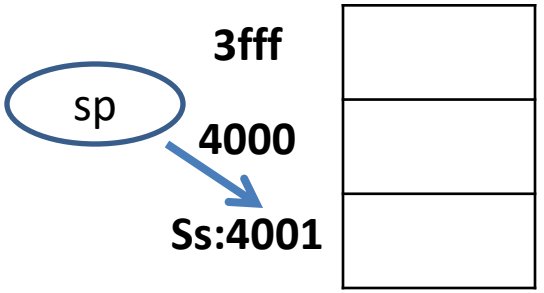


احسب العنوان الفيزيائي الذي سيصل اليه هذا الايعاز اذا علمت ان ds=9000,ss=8000,sp=4001

- Push [1200]
- P.A.=ds*10+الازاحة
- P.A.=90000+1200
- P.A.=91200 الاول
- P.A.=91201 الثاني



- P.A.=ss*10+الازاحة
- P.A.=80000+ sp
- P.A.=80000+4000
- P.A.=84000 الاول
- P.A.=83fff الثاني



الاختبارات

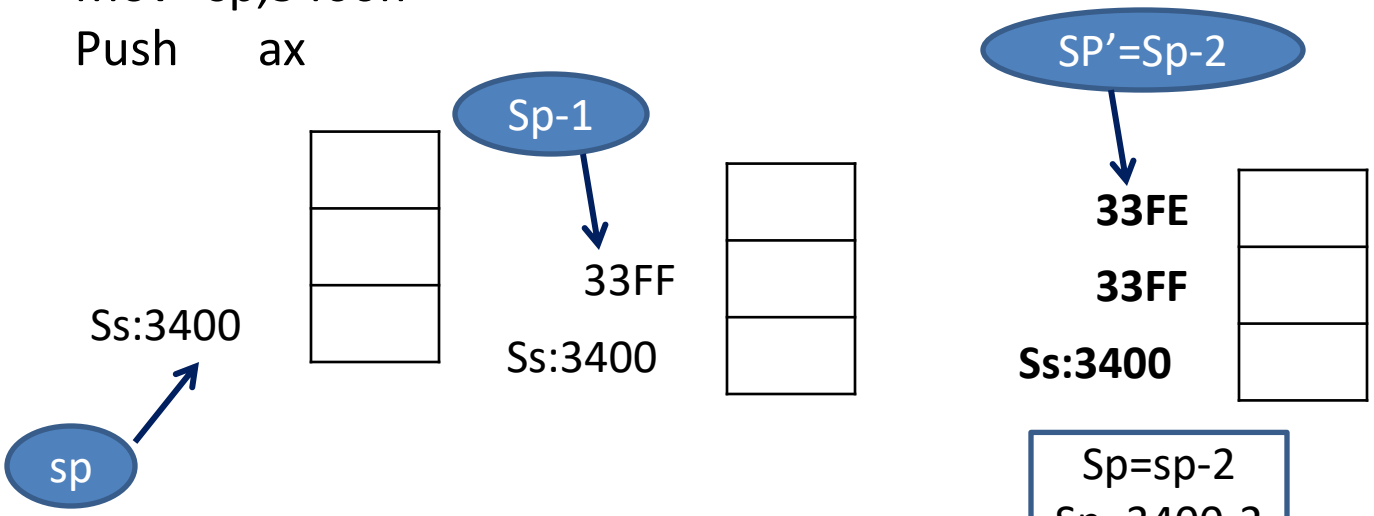
- ماهو العنوان الفيزيائي الذي سيصل اليه الايعاز التالي اذا علمت ان ds=3009,ss=9770
- Mov sp,3400h
- Push ax
- ماهي محتويات المكدس اذا علمت ان قيمة ax=6711h

الحل

ماهو العنوان الفيزيائي الذي سيصل اليه الايعاز التالي اذا علمت ان ds=3009,ss=9770

Mov sp,3400h

Push ax

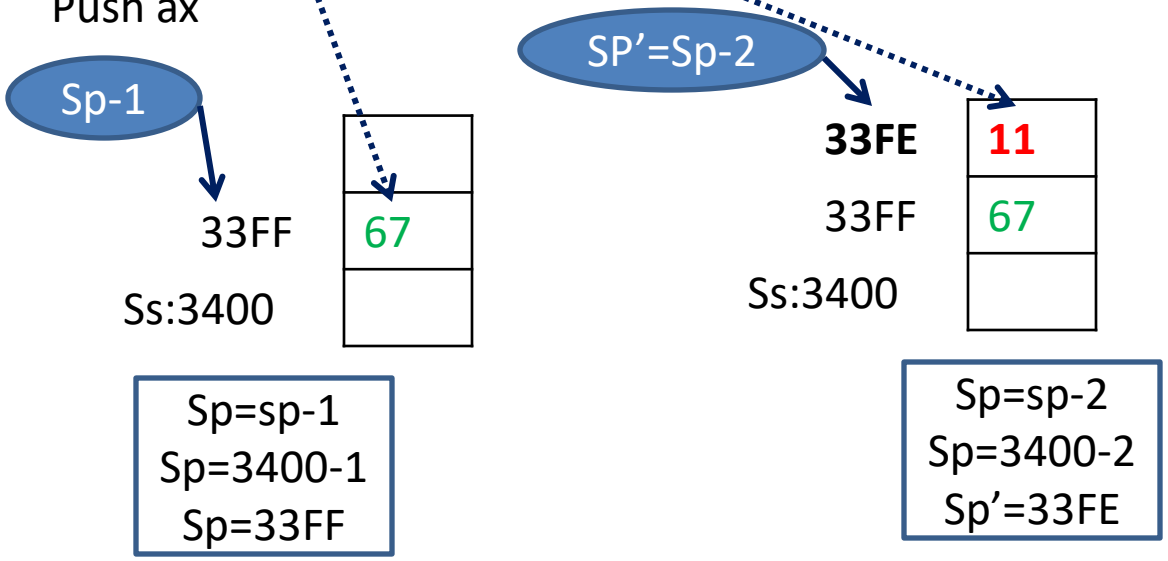


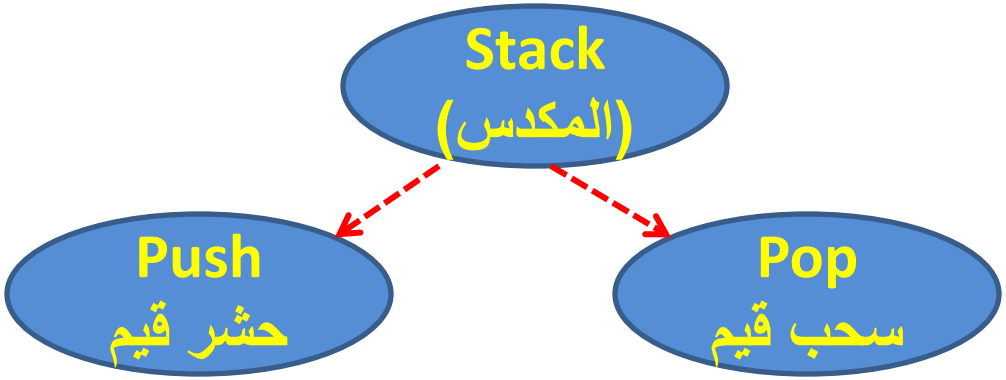
Sp=sp-2
Sp=3400-2
Sp'=33FE

P.A.=ss*10+الازاحة
P.A.=9770*10+SP
P.A.=97700+33FF
P.A.=9AAFF العنوان الاول
P.A.=9AAFE العنوان االثاني

ماهي محتويات المكدس اذا علمت ان قيمة ax=6711h

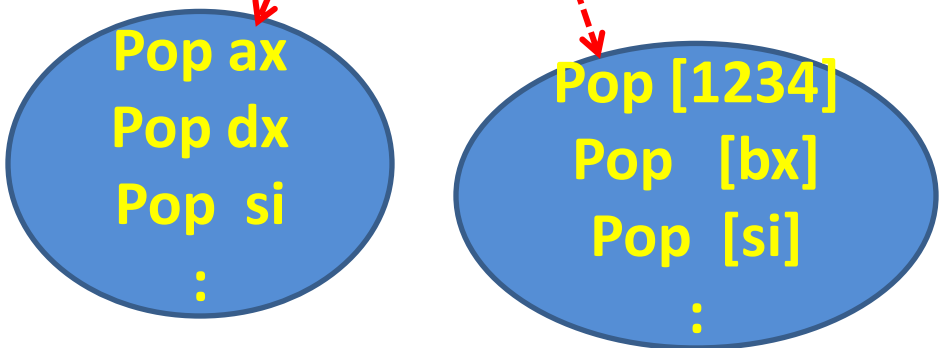
- Mov sp,3400h
- Push ax





- Pop:سحب قيم من داخل المكدس(stack) ووضعها في معامل الايعازpop
- صيغة الايعاز

- Pop reg./mem.

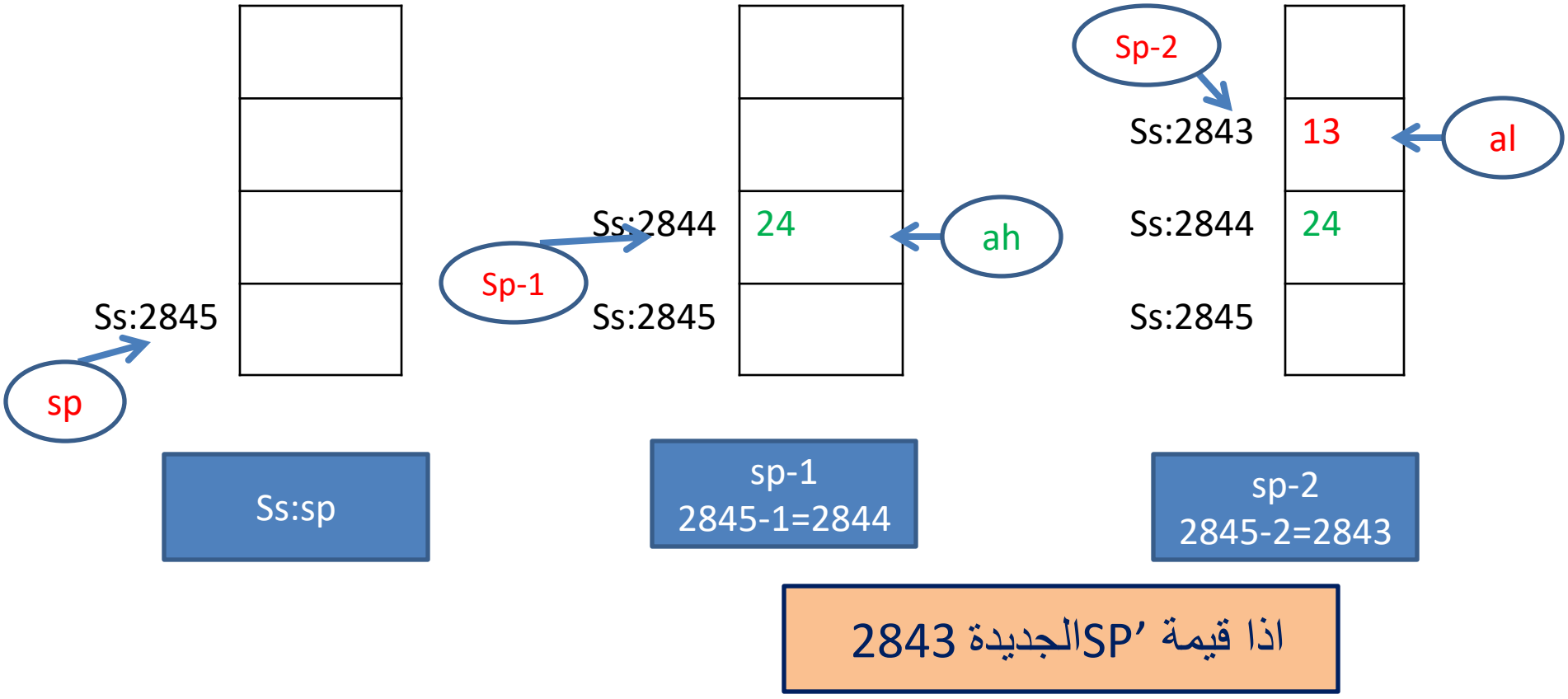


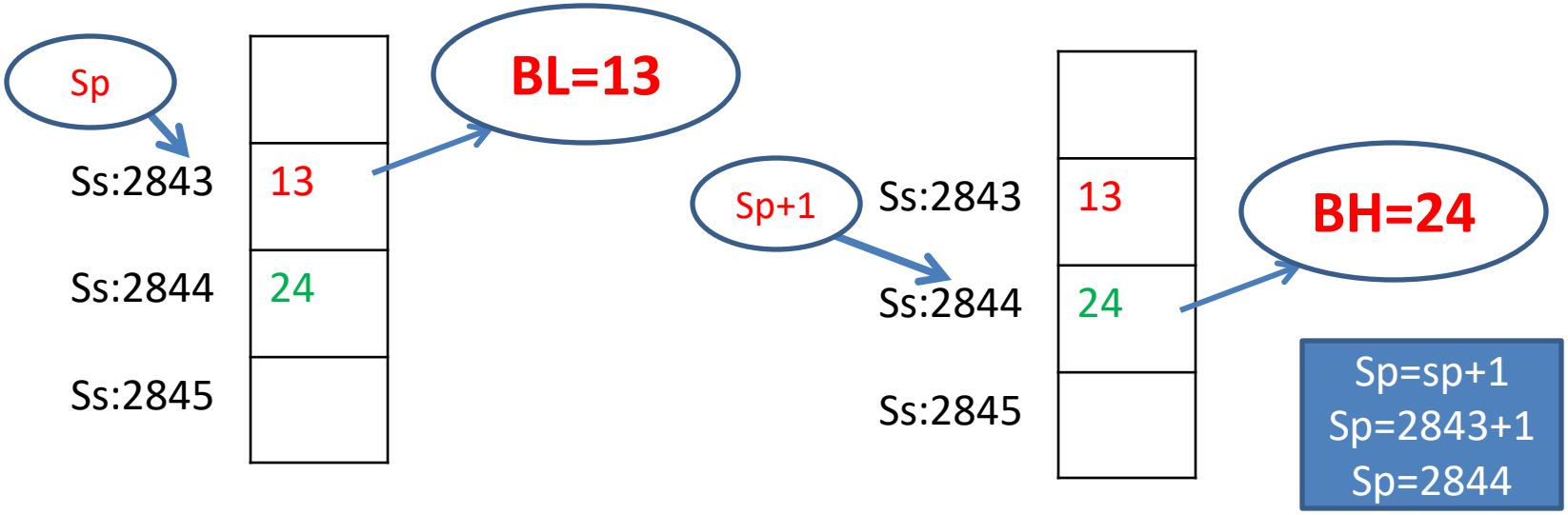
خطوات pop

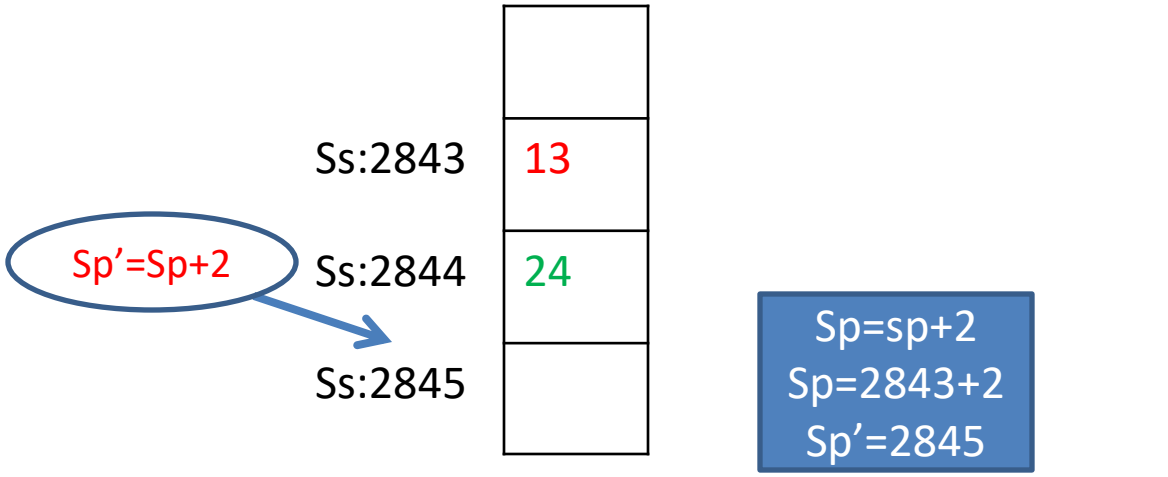
- 1- سحب محتويات المكدس (التي يشير اليها sp) الى الجزء الواطىء من المعامل
- Low=[sp]
- 2- زيادة ال spبمقدار واحد
- sp+1
- 3- سحب محتويات المكدس (التي يشير اليها sp) الى الجزء العالي من المعامل
- 4- زيادة ال spبمقدار اثنان
- Sp=sp+2

مثال: لنفرض ان sp=2845h,ax=2413
بين محتويات المكدس وقيمة sp,bx بعد تنفيذ التعليمات التالية

- Push ax
- Pop bx

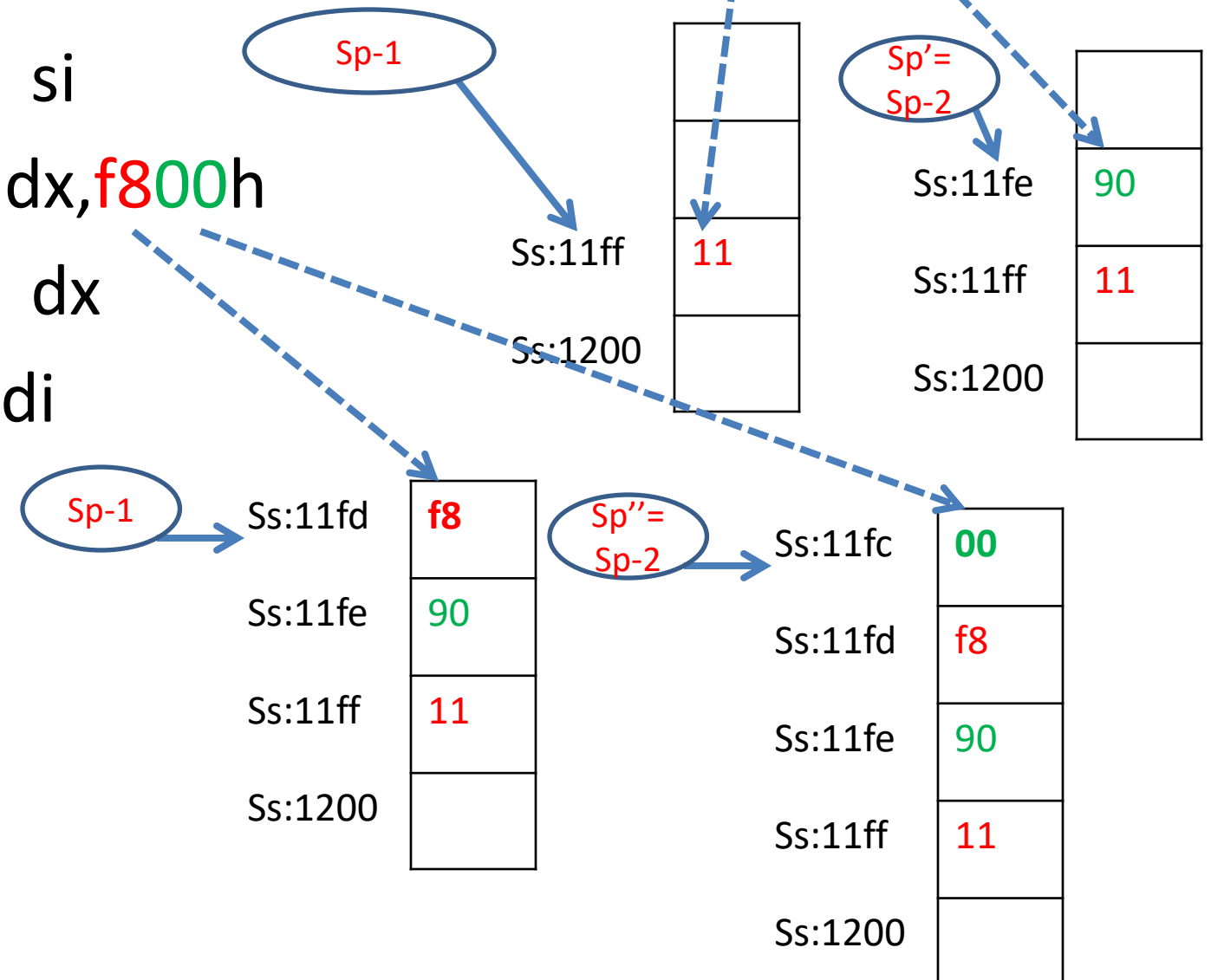


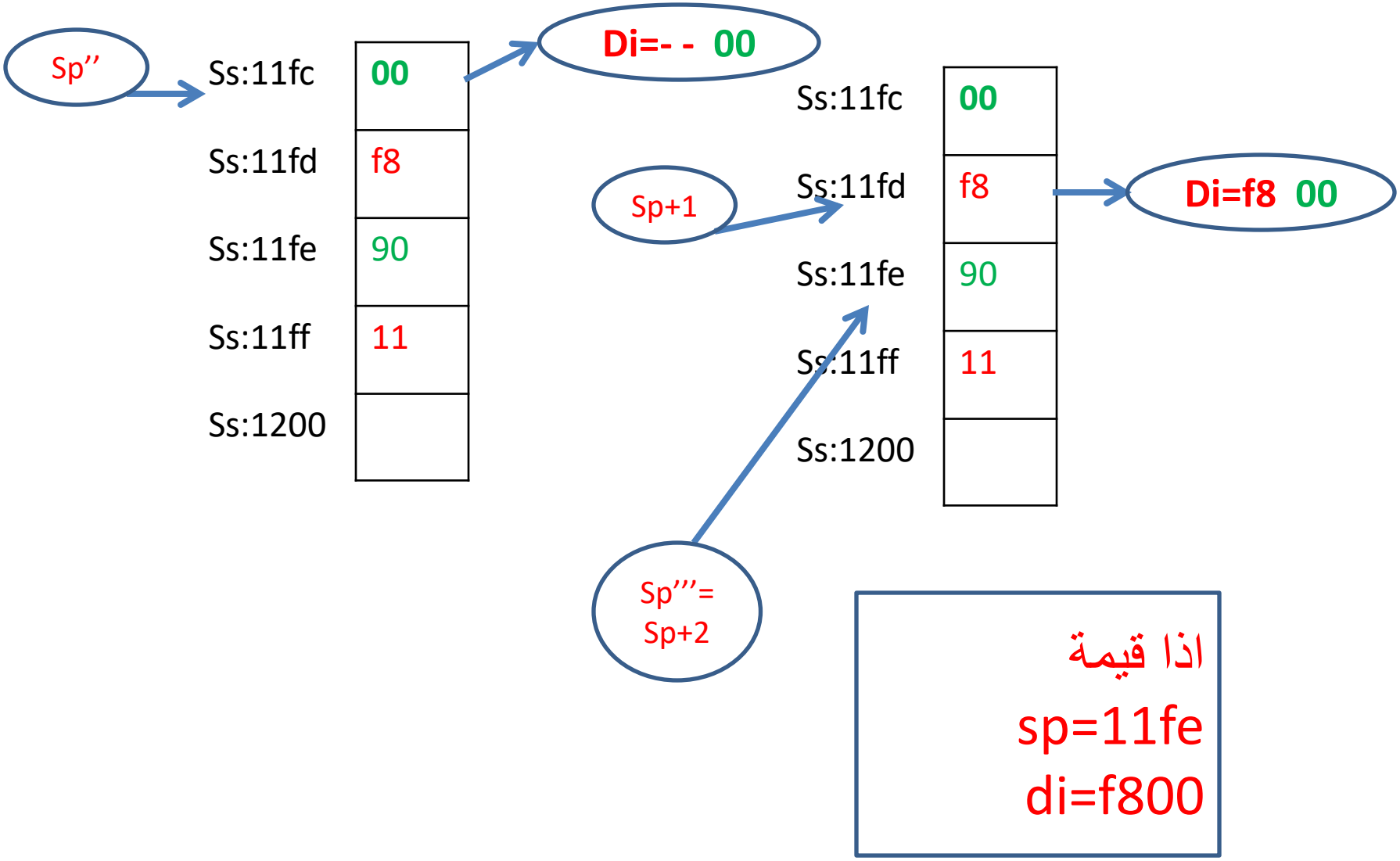

Sp
Ss:2843
13
Ss:2844
24
Ss:2845
BL=13
Sp+1
Ss:2843
13
Ss:2844
24
Ss:2845
BH=24
Sp=sp+1
Sp=2843+1
Sp=2844


Ss:2843
13
Sp'=Sp+2
Ss:2844
24
Ss:2845
Sp=sp+2
Sp=2843+2
Sp'=2845
اذا قيمة
sp=2845
Bx=2413

مثال: لنفرض ان sp=1200h,si=1190 بين محتويات المكدس وقيمة sp,di بعد تنفيذ التعليمات التالية

- Push si
- Mov dx,f800h
- Push dx
- Pop di




Sp''
Ss:11fc
00
Ss:11fd
f8
Ss:11fe
90
Ss:11ff
11
Ss:1200
Di=- - 00
Sp+1
Ss:11fc
00
Ss:11fd
f8
Ss:11fe
90
Ss:11ff
11
Ss:1200
Di=f8 00
Sp'''=
Sp+2
اذا قيمة
sp=11fe
di=f800

اختبار

- 1) اكتب برنامج لاستبدال قيمة bxبدل dx باستخدام ايعازات المكدس

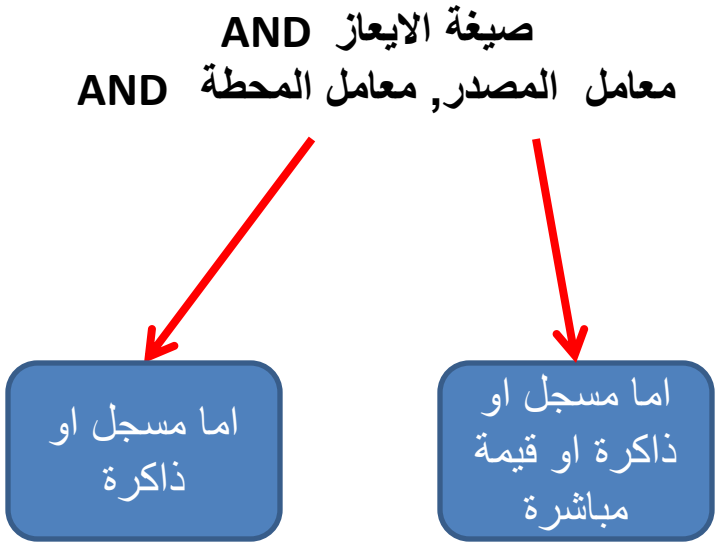
- 2)اكتب برنامج لجمع محتويات المكدس عند العنوان
- [9002]+[9003] → ah

الايعازات المنطقية

AND,OR,XOR,SHIFT & COMPARE

ايعاز AND

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

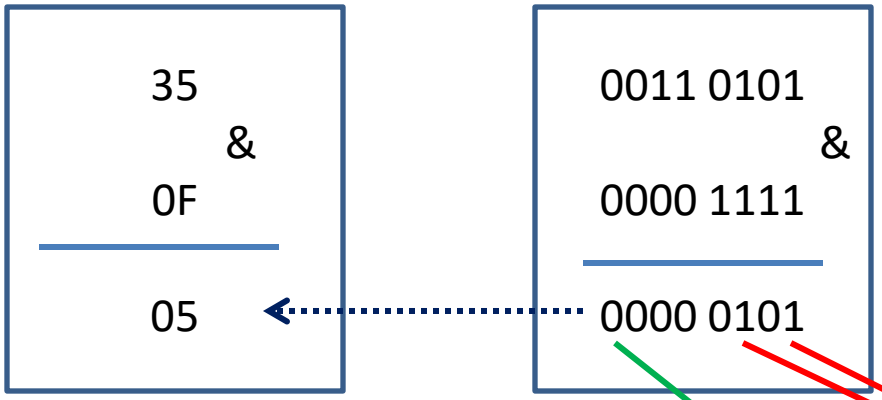


مثال

بين نتيجة التعليمتين التاليتين وماهو تاثيرها على مسجل الاعلام

- MOV BL,35h
- AND BL, 0Fh

دائما يقوم الايعاز AND بتصفير العلم CF وبقية الاعلام تتاثر حسب النتيجة



عدد الواحدات زوجي PF=1

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	0	0	1	0	0

فوائد الايعاز AND

- جعل بت معينة تساوي صفر
- مثال : اجعل البت الاول من المسجلdh يساوي صفر افرض قيمة dh=39h

39h=0011 1001

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1	1	1	0	0	1

نقوم بايجاد رقم جميع بتاته واحد ماعدا البت الاول نجعله صفر

الحل

AND DH , FEh

39
&
FE

38

0011 1001
&
1111 1110

0011 1000

مثال : اجعل البت الرابع من المسجلALيساوي صفر افرض قيمة AL=58hثم بين قيمة المسجل AL ومسجل الاعلام بعد تنفيذ البرنامج

- الحل

58h=0101 1000

b7	b6	b5	b4	b3	b2	b1	b0
0	1	0	1	1	0	0	0

نقوم بايجاد رقم جميع بتاته واحد ماعدا البت الرابع نجعله صفر

F7=1111 0111

AND AL, F7h

58 & f7 = 50

0101 1000 & 1111 0111 = 0101 0000

AL=50

عدد الواحدات PF=1زوجي

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	0	0	1	0	0

مثال : اجعل البت الثامن من الذاكرة عند العنوان 4500يساوي صفر افرض قيمة [4500]=C1ثم بين محتويات الذاكرة بعد تنفيذ البرنامج ومسجل الاعلام

- الحل

4502	**09**
4501	ff
4500	c1

b7	b6	b5	b4	b3	b2	b1	b0
1	1	0	0	0	0	0	1

نقوم بايجاد رقم جميع بتاته واحد ماعدا البت الثامن نجعله صفر

7f=0111 1111

AND [4500h], 7Fh

C1 & 7F = 41

1100 0001 & 0111 1111 = 0100 0001

[4500] =41

بعد التنفيذ

4502	**09**
4501	ff
4500	41

اختبار

- مثال :اكتب برنامج لجعل البت الثاني والخامس من الذاكرة عند العنوان 4000يساوي صفر افرض قيمة
73=[4000]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج
- مثال : اكتب برنامج لجعل جميع البتات من الذاكرة عند العنوان 4000يساوي صفر افرض قيمة 73=[4000]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج
- And [4000h],00

مثال : اجعل البت الثاني والخامس من الذاكرة عند العنوان 4000يساوي صفر افرض قيمة 73=[4000]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج

4002	**09**
4001	32
4000	73

b7	b6	b5	b4	b3	b2	b1	b0
0	1	1	1	0	0	1	1

- الحل

نقوم بايجاد رقم جميع بتاته واحد ماعدا البت الثاني والخامس

E D= 1110 1101
AND [4000h], ED

73
&
ED
61

0111 0011
&
1110 1101
0110 0001

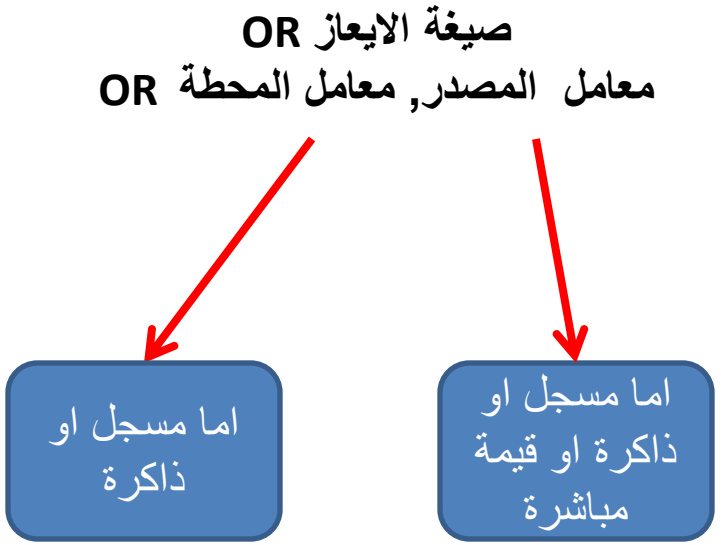
4002	**09**
4001	32
4000	61

الايعازات المنطقية

AND,OR,XOR,SHIFT & COMPARE

ايعاز OR

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

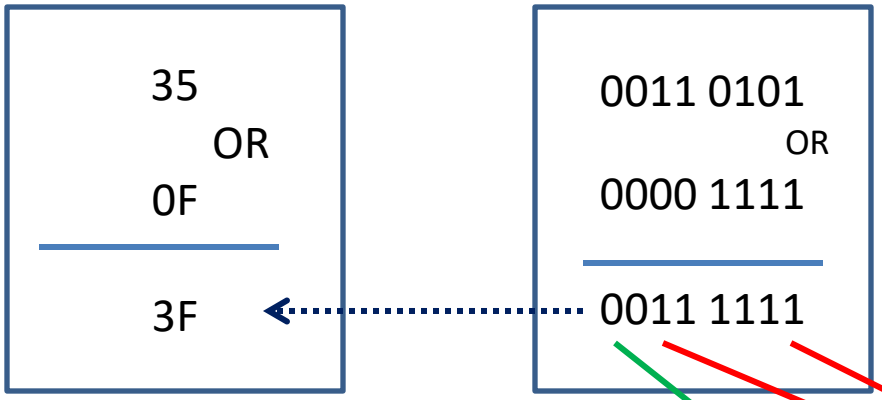


مثال

بين نتيجة التعليمتين التاليتين وماهو تاثيرها على مسجل الاعلام

- MOV BL,35h
- OR BL, 0Fh

دائما يقوم الايعاز OR بتصفير العلم CF وبقية الاعلام تتاثر حسب النتيجة

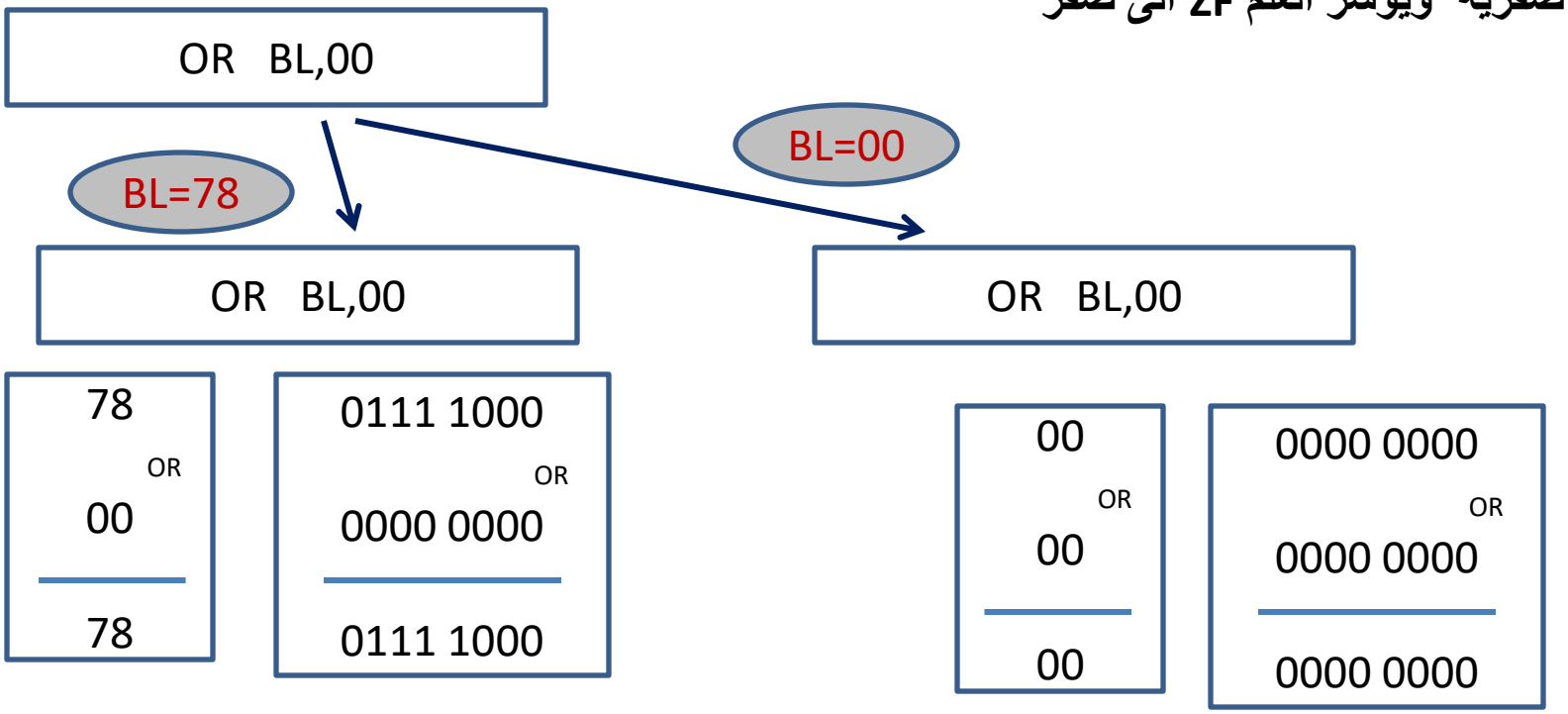


عدد الواحدات زوجي PF=1

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	0	0	1	0	0

فوائد الايعاز OR

- **اولا: اختبار المعامل صفر ام لا**
- **مثال :التعليمة التالية تبين هل ان المسجل BL صفر ام لا فاذا كان المسجل BLصفر ستظهر النتيجة صفرية ويؤشر العلم ZF الى صفر**



Z	S	X	A	X	P	X	C
0	0	0	0	0	1	0	0

Z	S	X	A	X	P	X	C
1	0	0	0	0	1	0	0

فوائد الايعاز OR

- ثانيا: جعل بت معينة تساوي واحد
- مثال : اجعل البت الاول من المسجلdh يساوي واحد افرض قيمة dh=38h

38h=0011 1000

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1	1	1	0	0	0

DH=39

نقوم بايجاد رقم جميع بتاته
اصفار ماعدا البت الاول نجعله
واحد الحل

OR DH , 01h

38
OR
01
39

0011 1000
OR
0000 0001
0011 1001

مثال : اجعل البت الثامن من الذاكرة عند العنوان 1500يساوي واحد افرض قيمة 41=[1500]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج

- الحل

1502	**09**
1501	ff
1500	41

b7	b6	b5	b4	b3	b2	b1	b0
0	1	0	0	0	0	0	1

نقوم بايجاد رقم جميع بتاته اصفار ماعدا البت الثامن نجعله واحد

80=1000 0000

OR [1500h], 80h

0100 0001
or
1000 0000
1100 0001

41
or
80
C1

[1500] =C1

بعد التنفيذ

1502	**09**
1501	ff
1500	C1

اختبار

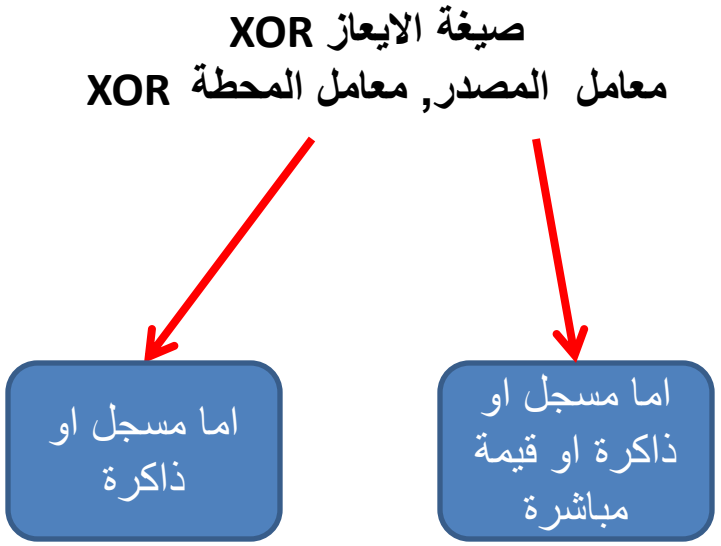
- مثال :اكتب برنامج لجعل البت الثاني والخامس من الذاكرة عند العنوان 4000يساوي واحد مع عدم تغييربقية البتات افرض قيمة 56=[4000]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج
- مثال : اكتب برنامج لجعل جميع البتات من الذاكرة عند العنوان 4000يساوي واحد افرض قيمة 90=[4000]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج

الايعازات المنطقية

AND,OR,XOR,SHIFT & COMPARE

ايعاز XOR

X	Y	X xor Y
0	0	0
0	1	1
1	0	1
1	1	0



مثال

تتبع البرنامج التالي وماهو تاثيرها على مسجل الاعلام

- MOV BL,54h
- XOR BL, 78h

دائما يقوم الايعاز XORبتصفير العلم CF وبقية الاعلام تتاثر حسب النتيجة

54
XOR
78
2C

0101 0100
XOR
0111 1000
0010 1100

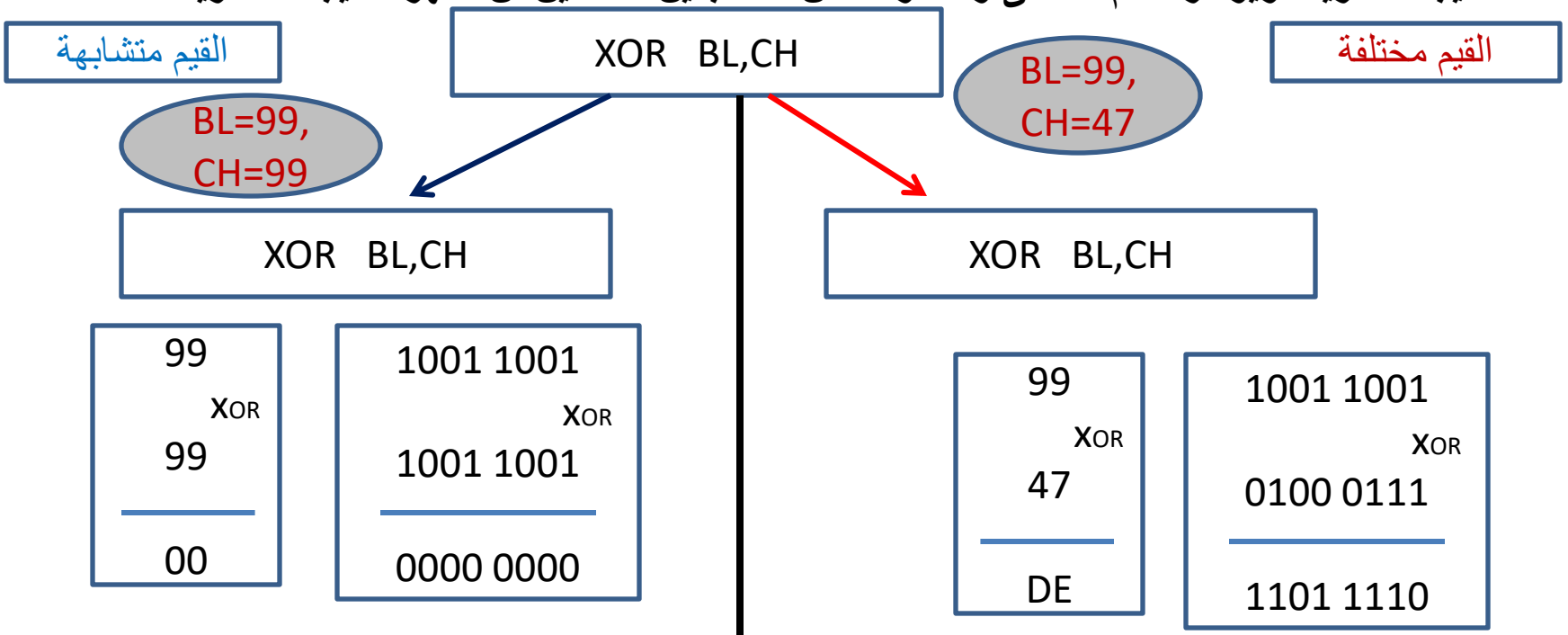
BL=2C

عدد الواحدات فرديPF=0

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	0	0	0	0	0

فوائد الايعاز XOR

- **اولا: اختبار هل ان المعاملين يحتويان نفس القيمة**
- **مثال :التعليمة التالية تبين هل ان المسجل BL يساوي المسجل CH فاذا كان المسجلين متشابهين ستظهر النتيجة صفرية ويؤشر العلم ZF الى واحد واذا كان المسجلين مختلفين لن تظهر النتيجة صفرية**



Z	S	X	A	X	P	X	C
1	0	0	0	0	1	0	0

Z	S	X	A	X	P	X	C
0	0	0	0	0	1	0	0

فوائد الايعاز XOR

- ثانيا: قلب قيمة بت معينة
- مثال : اقلب البت الاول من المسجلdh وافرض قيمة dh=38h

38h=0011 1000

1

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1	1	1	0	0	0

DH=39

نقوم بايجاد رقم جميع بتاته
اصفار ماعدا البت الاول نجعله
واحد الحل 01=0000 0001
XOR DH , 01h

38
XOR
01
39

0011 1000
XOR
0000 0001
0011 1001

مثال : اقلب البت الثالث والسادس من الذاكرة عند العنوان 1200افرض قيمة 61=[1200]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج

1202	**09**
1201	ff
1200	61

b7	b6	b5	b4	b3	b2	b1	b0
0	1	1	0	0	0	0	1

- الحل

نقوم بايجاد رقم جميع بتاته اصفار ماعدا البت الثالث والسادس نجعله واحد

24=0010 0100

XOR [1200h], 24h

61
Xor
24
65

0110 0001
Xor
0010 0100
0100 0101

[1200] =65

بعد التنفيذ

1202	**09**
1201	ff
1200	65

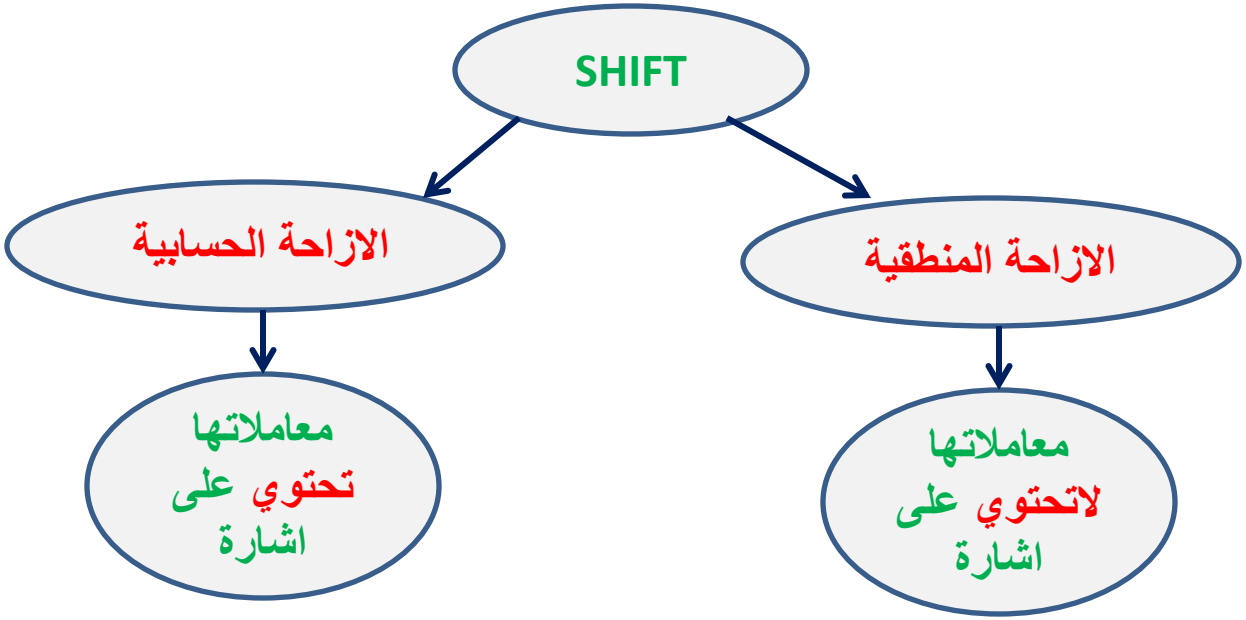
اختبار

- س1:اكتب برنامج لايجاد المتمم الاول لمحتويات الذاكرة عند العنوان 8000 افرض قيمة 56=[8000]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج
- س2: اكتب برنامج لايجاد المتمم الثاني من الذاكرة عند العنوان 8000 افرض قيمة 56=[8000]ثم بين محتويات الذاكرة بعد تنفيذ البرنامج

الايعازات المنطقية

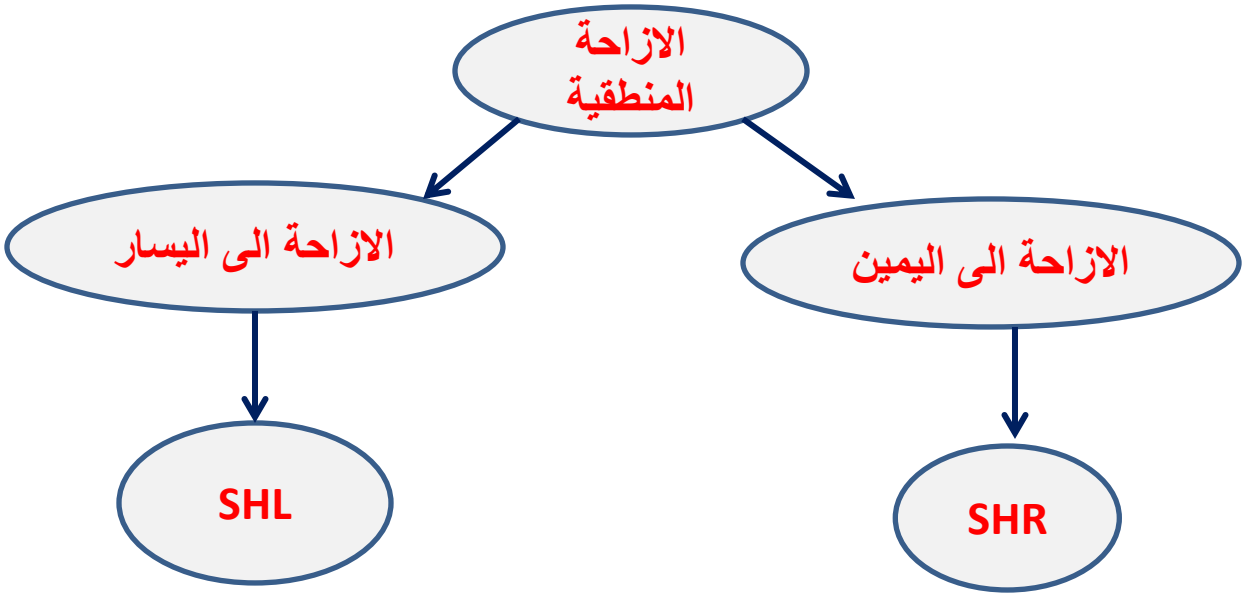
AND,OR,XOR,SHIFT & COMPARE

ايعاز الازاحة SHIFT



الازاحة المنطقية

- نستطيع باستخدام ايعازات الازاحة ازاحة محتويات مسجل او موقع ذاكري الى اليمين او الى اليسار, ويحدد عدد مرات التزحيف بشكل مباشر اذا كان عدد التزحيف بت واحدة بينما يحدد من خلال مسجل **CL** اذا كان اكثر من بت واحدة,

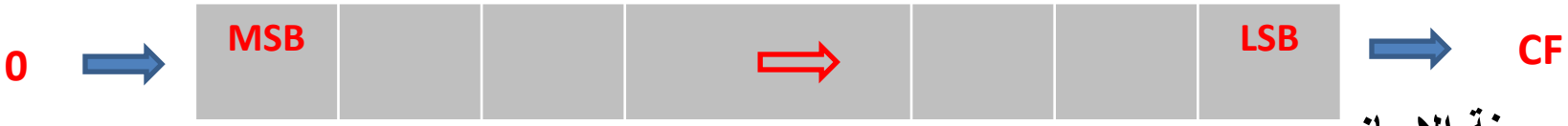


SHR

- تقوم هذه التعليمة بازاحة بتات المعامل الى اليمين بت تلو الاخرى ومن اجل ذلك تنقل البت الاقل اهمية

least significant bit (LSB)الى علم CF وتحل قيمة الصفر محل البت الاعلى اهمية

- most significant bit (MSB)

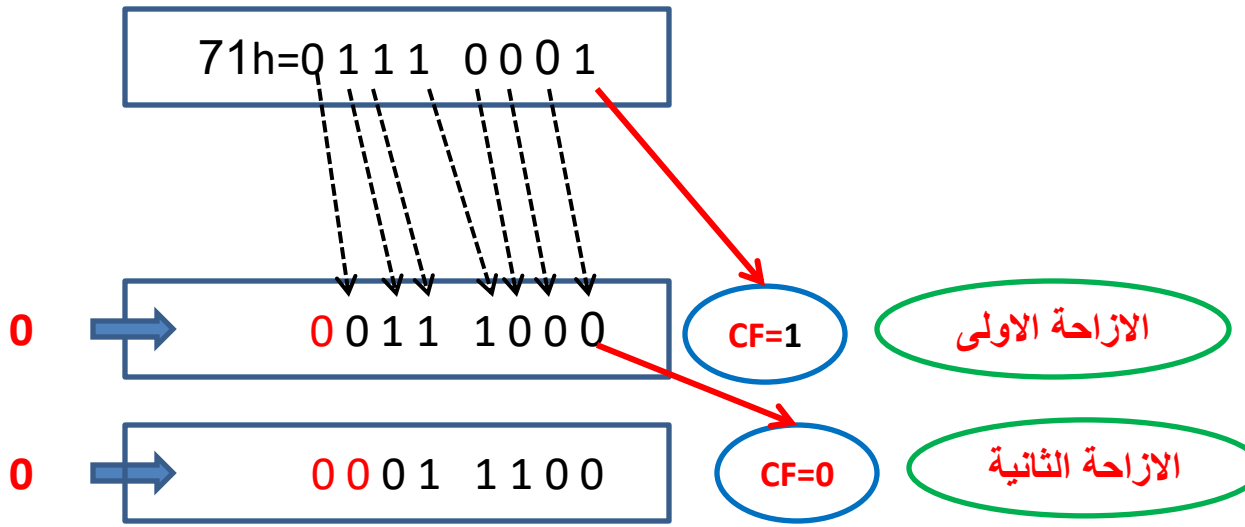


- صيغة الايعاز

- SHR معامل المراد ازاحة بتاته , عدد مرات التزحيف

مثال
بين نتيجة تنفيذ التعليمة SHR

- MOV AL,71H
- MOV CL,02H
- SHR AL,CL

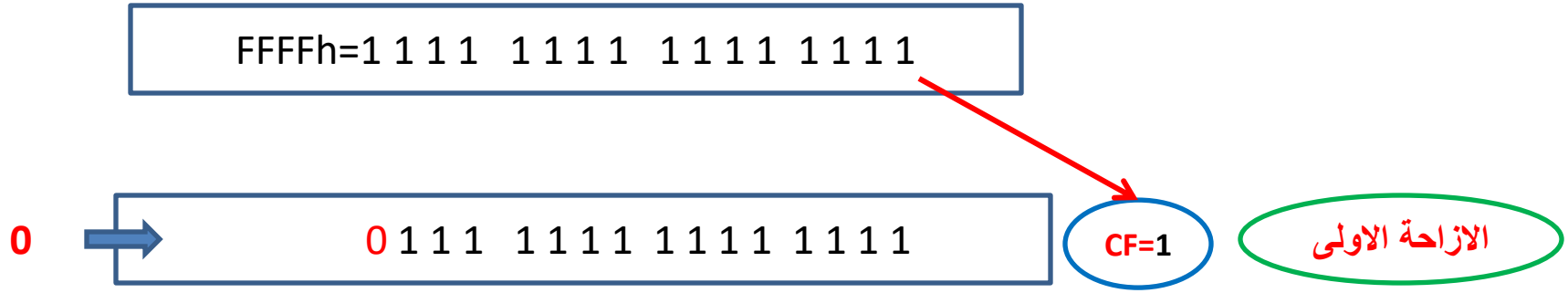


AL=1c
CL=02
CF=0

مثال

زحف المسجل DX=FFFFبمقدار واحد الى جهة اليمين

- SHR DX,01

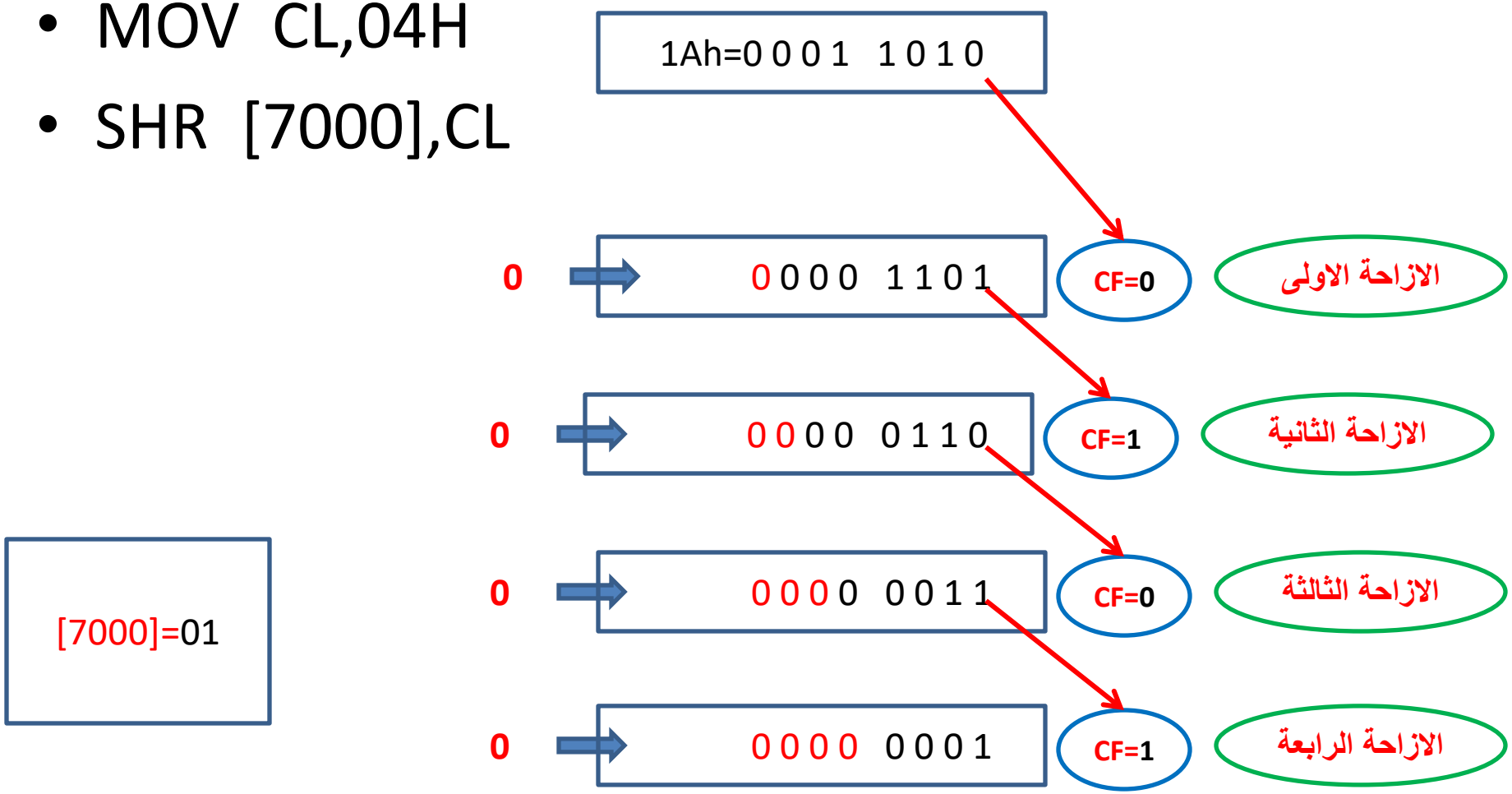


DX=7FFF

مثال

اكتب برنامج لتزحيف محتويات الذاكرة [7000]=1A بمقدار اربع مرات

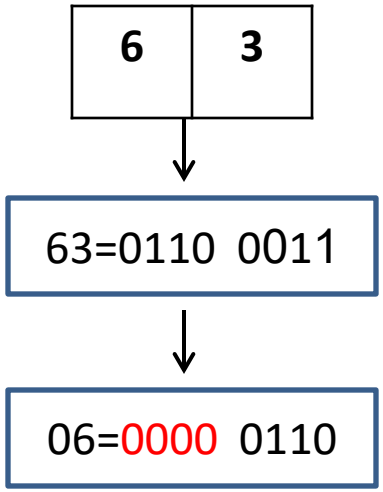
- MOV CL,04H
- SHR [7000],CL



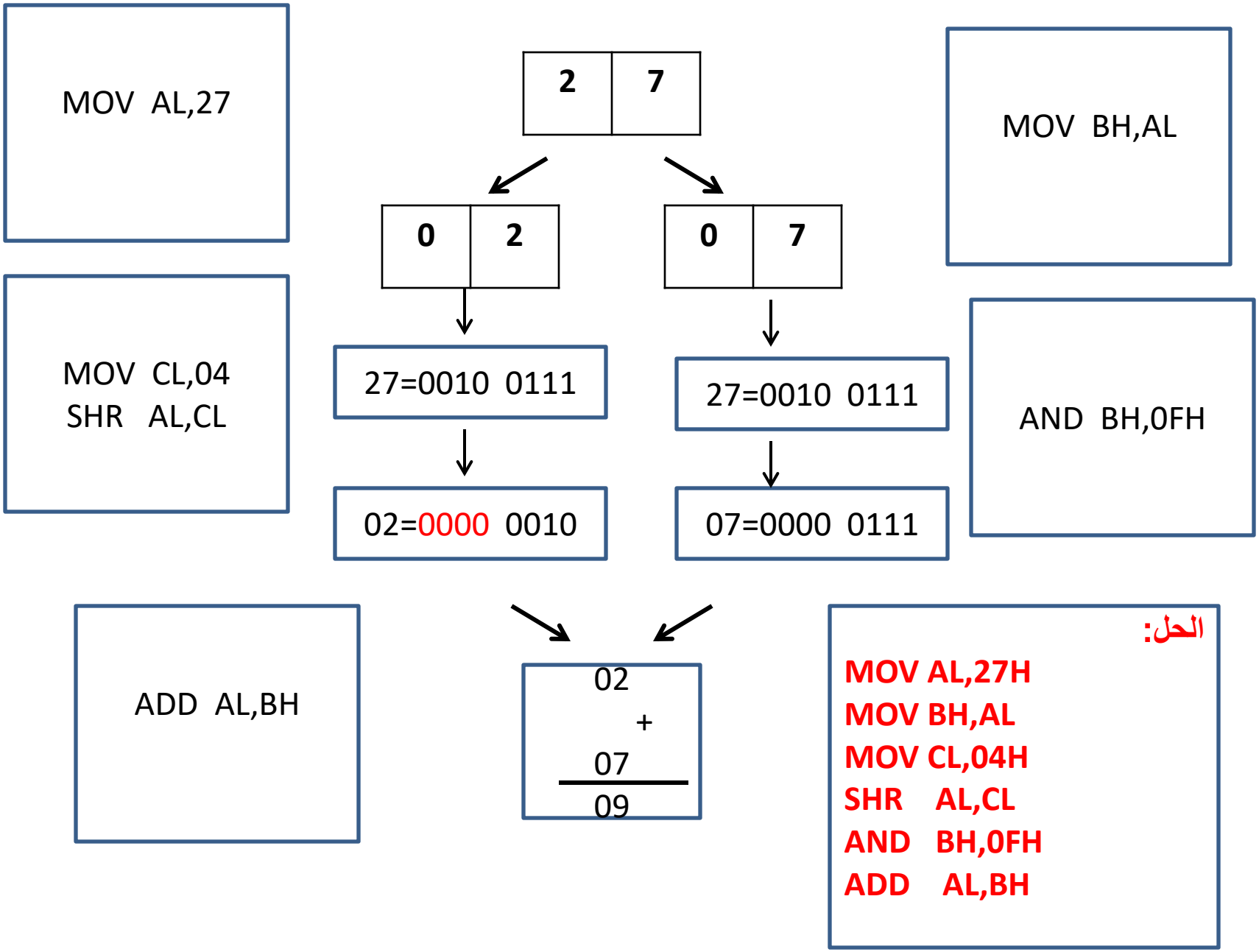
الاختبارات

- س1) اكتب برنامج لتزحيف المرتبة العليا الى المرتبة الدنيا من الرقم 63 (63--->06) باستخدام ايعازات التزحيف
- س2) اكتب برنامج لجمع مراتب العدد مع بعضها مثلا (27---> 09) او(14----> 05) والخ....

س1) اكتب برنامج لتزحيف المرتبة العليا الى المرتبة الدنيا من الرقم 63
(63---> 06) باستخدام ايعازات التزحيف




```
MOV  DH, 63
MOV  CL,04
SHR   DH,CL
```


MOV AL,27
2 7
MOV BH,AL
0 2
0 7
MOV CL,04
SHR AL,CL
27=0010 0111
27=0010 0111
AND BH,0FH
02=0000 0010
07=0000 0111
ADD AL,BH
02
+
07
09
الحل:
MOV AL,27H
MOV BH,AL
MOV CL,04H
SHR AL,CL
AND BH,0FH
ADD AL,BH

الايعازات المنطقية

AND,OR,XOR,SHIFT & COMPARE

SHL
الازاحة الى اليسار

- تقوم هذه التعليمة بازاحة بتات المعامل الى اليساربت تلو الاخرى ومن اجل ذلك تنقل البت الاعلى اهمية

most significant bit (MSB)الى علم CF وتحل قيمة الصفر محل البت الاقل اهمية

- least significant bit (LSB)

- صيغة الايعاز

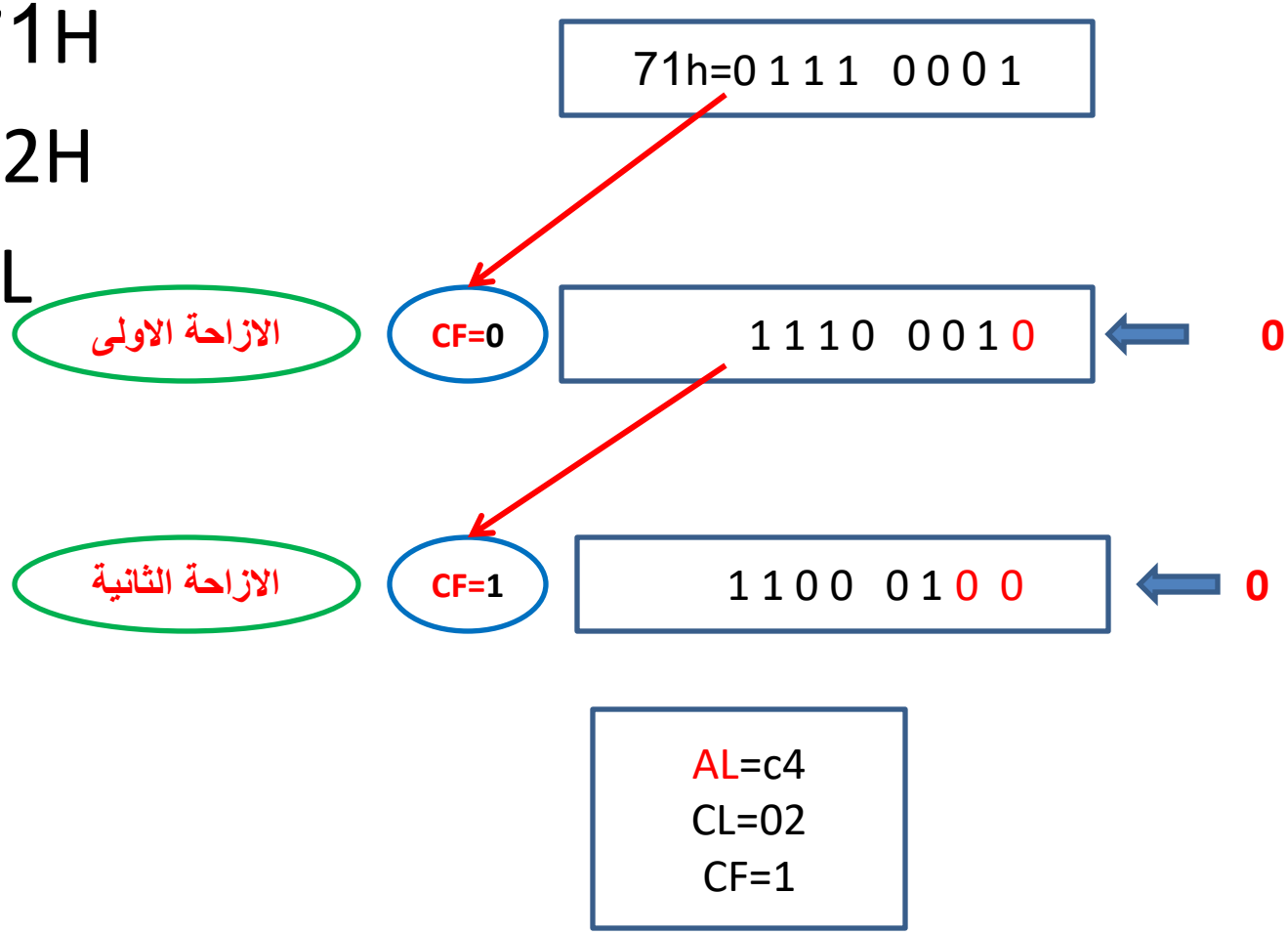
- SHL معامل المراد ازاحة بتاته , عدد مرات التزحيف

مسجل او ذاكرة

عدد ثابت او مسجل CL

مثال
بين نتيجة تنفيذ التعليمة SHL

- MOV AL,71H
- MOV CL,02H
- SHL AL,CL

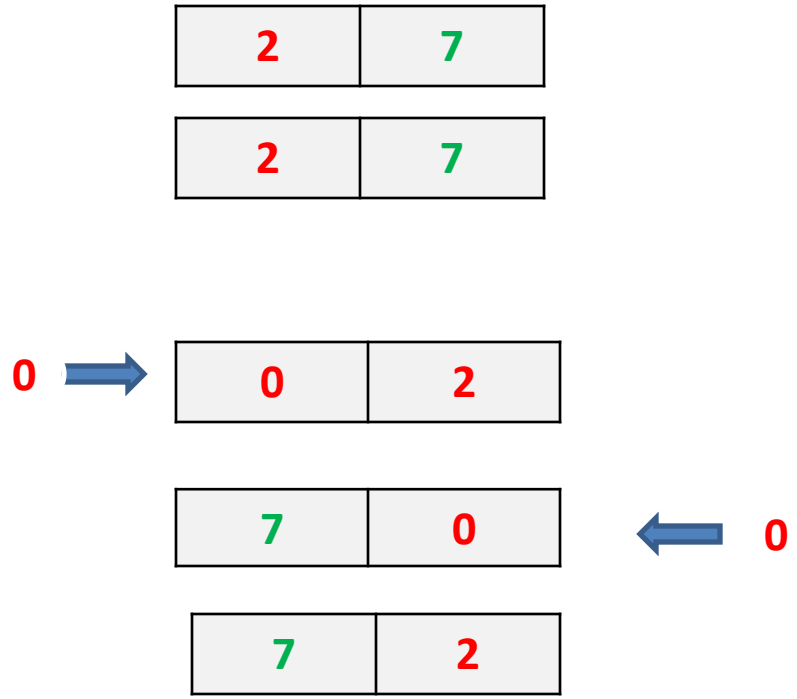


اختبار
ايعاز shl

- س1) اكتب برنامج لقلب مراتب القيم مثلا 27 الى 72 او اي رقم اخر عشوائي باستخدام ايعازات التزحيف

الحل

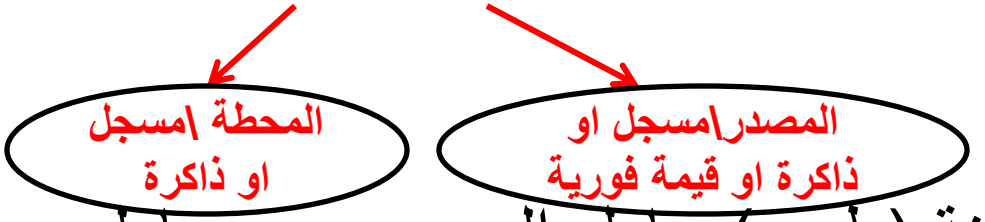
- Mov AH,27H
- MOV BL,AH
- MOV CL,04
- SHR AH,CL
- SHL BL,CL
- ADD AH,BL



الايعاز المقارنة cmp

- صيغة الايعاز

- Cmp dest.,source



- يقوم هذا الايعاز بمقارنة (طرح)معامل المصدر مع معامل المحطة ثم يقوم بتغيير الاعلام وفقا للنتيجة مع عدم التاثير النتيجة على المعاملات اي تبقى بدون تغيير
- على الرغم من تاثير النتيجة على الاعلامof,zf,pf,sf,af,cf
- الا انه يستفاد من العلمين zf,cf فقط

تاثير ايعاز المقارنة cmp على الاعلام

معاملات المقارنة	CF	ZF
المحطة > المصدر	0	0
المحطة = المصدر	0	1
المحطة < المصدر	1	0

مثال

```
Mov  al,05
Cmp  al,06
```

AL= 05ستبقى بدون تغيير

Al

-

06

05

-

06

-1

0000 0101

+

1111 1010

1111 1111

Cf=0
يقلب الى
cf=1

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	1	0	1	0	1	0	1

تاثير ايعاز المقارنة cmp على الاعلام

معاملات المقارنة	CF	ZF
المحطة > المصدر	0	0
المحطة = المصدر	0	1
المحطة < المصدر	1	0

مثال

```
Mov  al,06
Cmp  al,05
```

AL= 06ستبقى بدون تغيير

```
Al      -
05
```

```
06      -
05
1
```

Cf=1
يقلب الى
cf=0

```
0000 0110      +
1111 1011
0000 0001
```

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	0	0	0	0	0

تاثير ايعاز المقارنة cmp على الاعلام

معاملات المقارنة	CF	ZF
المحطة > المصدر	0	0
المحطة = المصدر	0	1
المحطة < المصدر	1	0

مثال

```
Mov  al,06
Cmp  al,06
```

AL= 06ستبقى بدون تغيير

```
Al
-
06
```

```
06
-
06
0
```

```
0000 0110
+
1111 1010
0000 0000
```

Cf=1
يقلب الى
cf=0

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								1	0	0	0	0	1	0	0

ايعاز المقارنة cmp

- صيغة الايعاز

- Cmp dest.,source

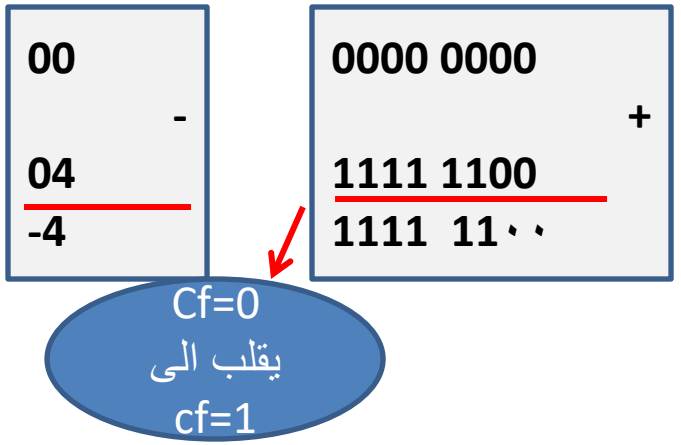
ZF	CF	معاملات المقارنة
0	0	المحطة > المصدر
1	0	المحطة = المصدر
0	1	المحطة < المصدر

اكتب برنامج لمقارنة العدد ٠٤ مع محتويات الذاكرة عند العنوان ٦٠٠٠ وبين تاثير ذلك على مسجل الاعلام

6002	**04**
6001	01
6000	00

الحل

```
Mov  al,04
Cmp  [6000],al
```



X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	1	0	1	0	1	0	1

ماهي قيم مسجل الاعلام بعد تنفيذ البرنامج التالي

```
Mov  ax,0487h
Cmp  [f800h],ax
```

f802	88
F801	01
F800	65

01 65

-

04 87

FCDE

0000 0001 0110 0101

+

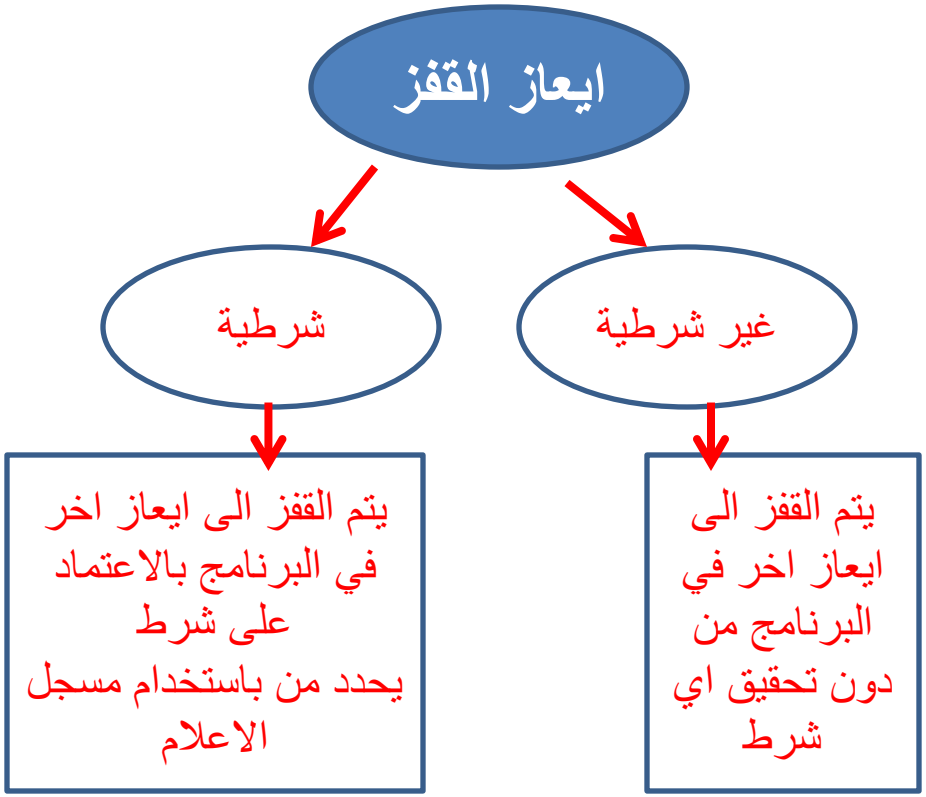
1111 1011 0111 1001

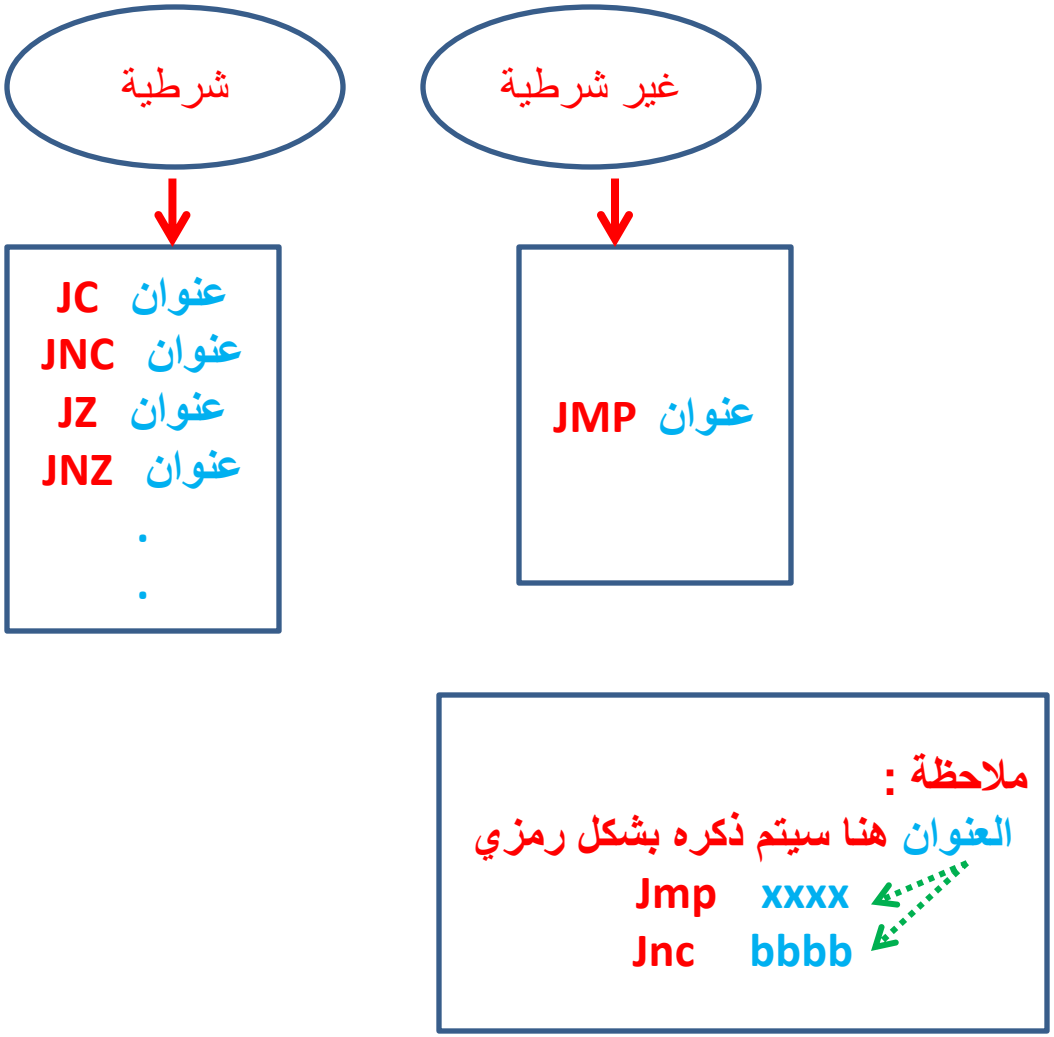
1111 1100 1101 1110

Cf=0
يقلب الى
cf=1

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	1	0	1	0	1	0	1

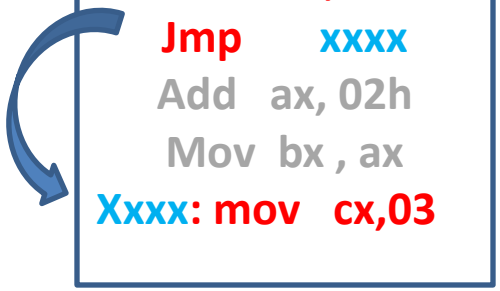
ايعازات القفز




غير شرطية
شرطية
JMP عنوان
JC عنوان
JNC عنوان
JZ عنوان
JNZ عنوان
.
.
ملاحظة :
العنوان هنا سيتم ذكره بشكل رمزي
Jmp xxxx
Jnc bbbb

مثال(القفز الغير مشروط)

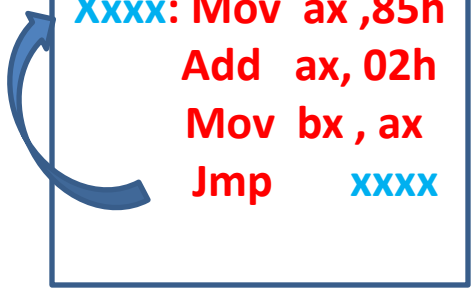
تتبع البرنامجين التاليين ثم وضح قيم ax,bx



الحل:

Ax=85h

Bx=00



الحل:

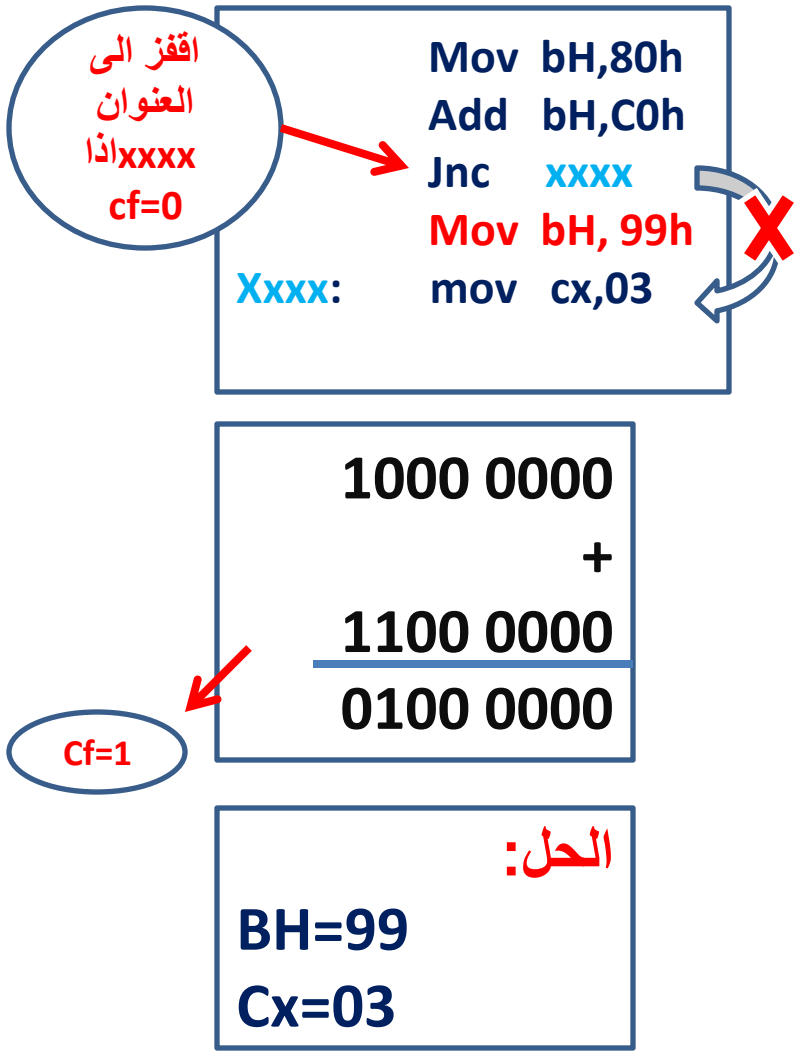
Ax=87h

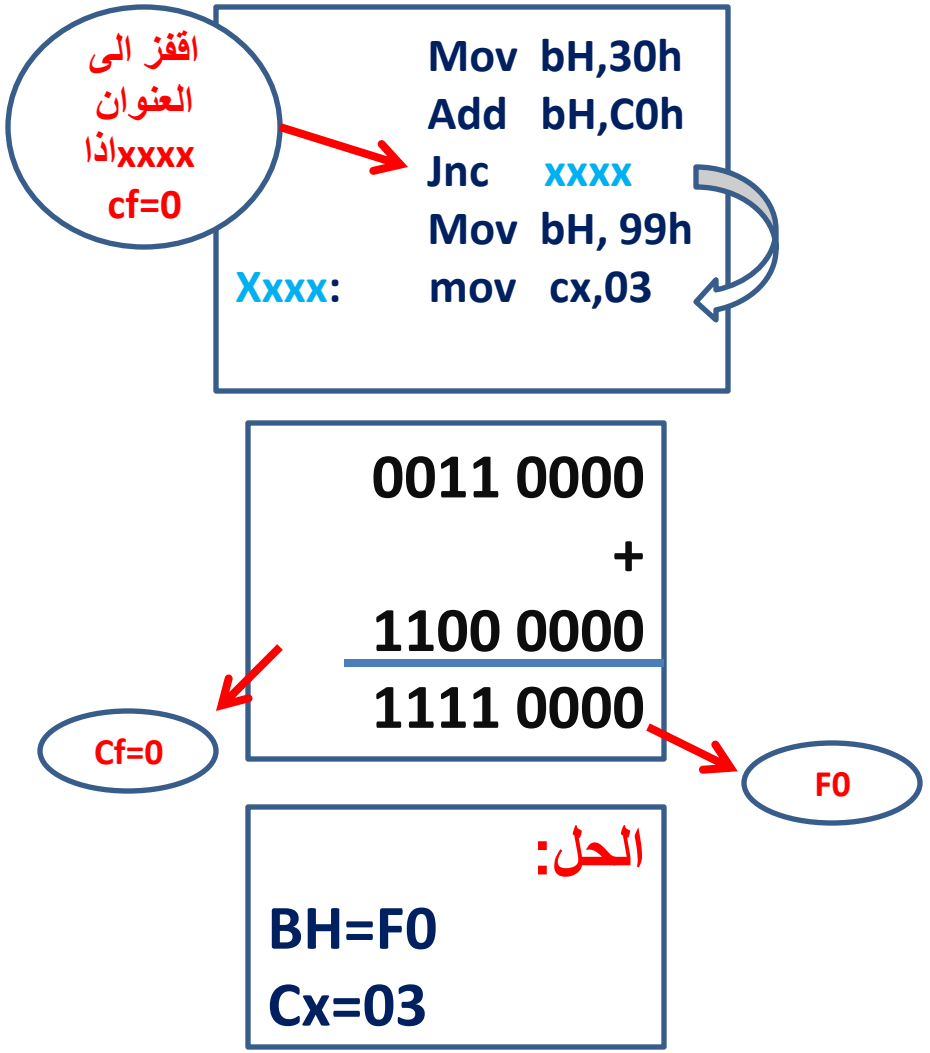
Bx=87h

ملاحظة البرنامج هنا لن يتوقف سيستمر في حالة دوران

مثال(القفز المشروطJNC)

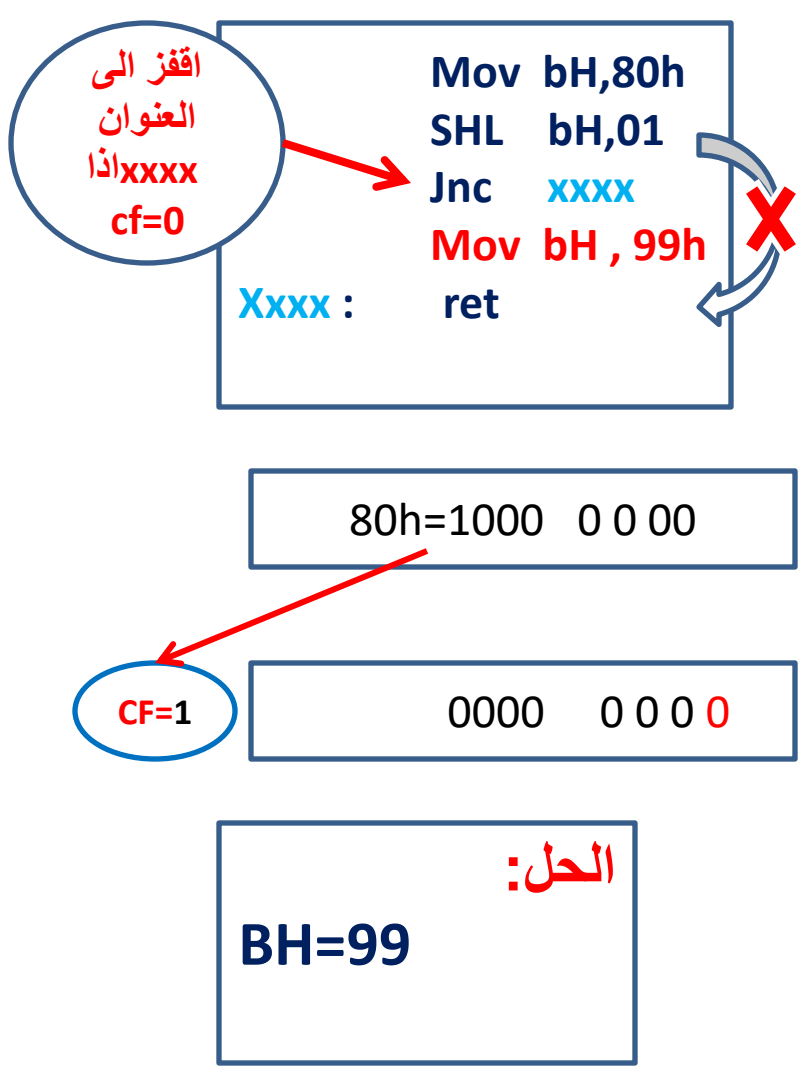
تتبع البرنامجين التاليين ثم وضح قيمcx,bH

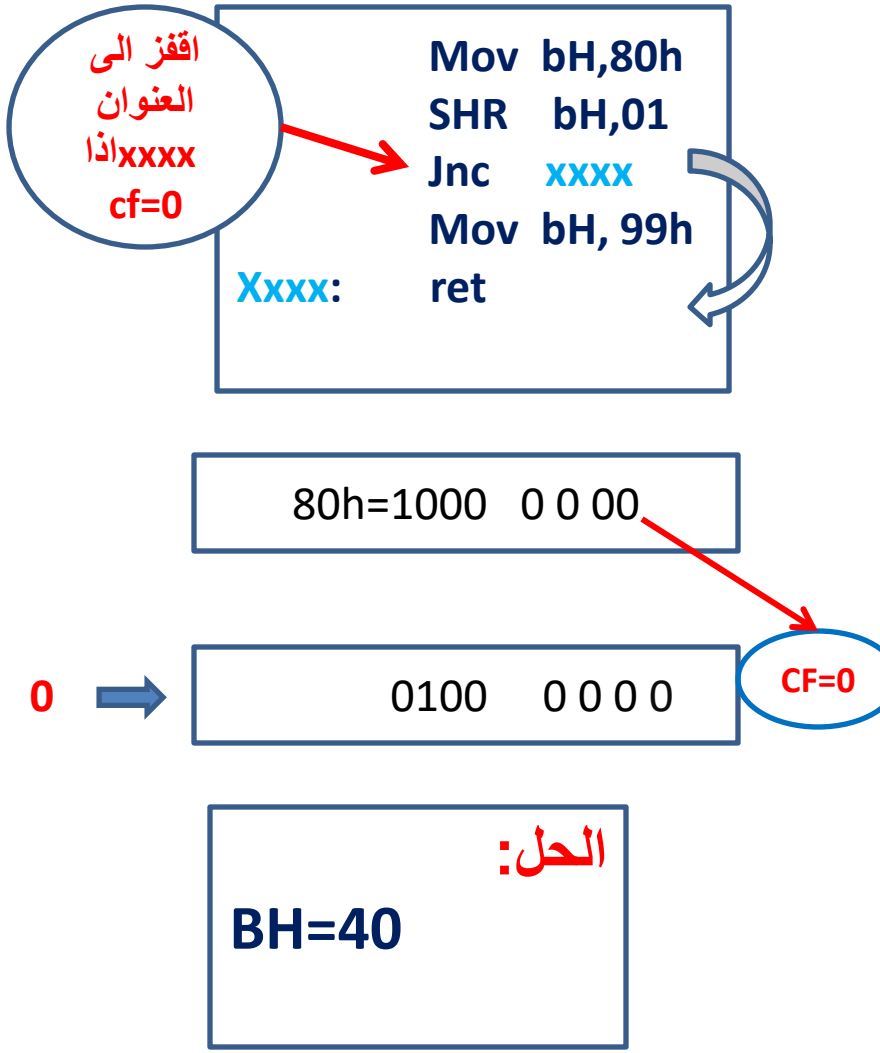




مثال

تتبع البرنامجين التاليين ثم وضح قيم bx

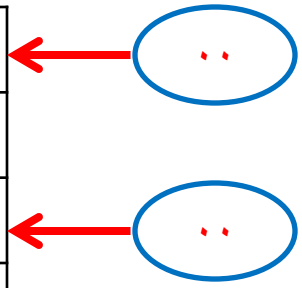




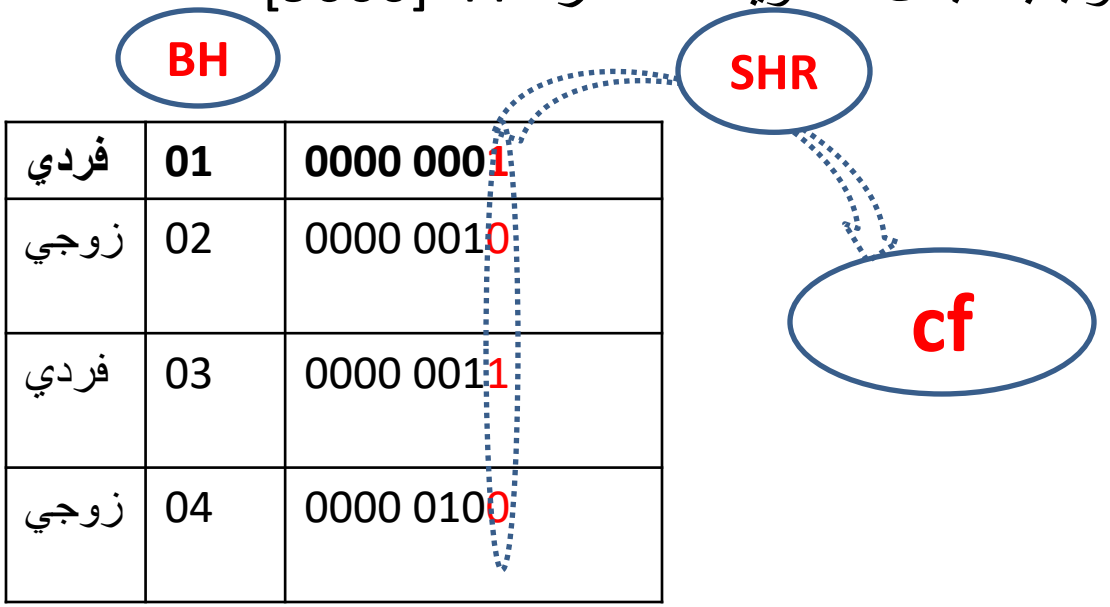
اختبار

- اكتب برنامج لتصفير محتويات الذاكرة [9000] اذا كان العدد في المسجل BH فردي واذا كان زوجي اجعل محتويات الذاكرة [9000]=FF
- اكتب برنامج لتصفير محتويات الذاكرةابتداءا من العنوان [9000] الى العنوان [9003]اذا كان الذاكرة تحتوي على العدد ٥

٩٠٠٣	٥
٩٠٠٢	01
٩٠٠١	5
٩٠٠٠	٩٩



اكتب برنامج لتصفير محتويات الذاكرة [9000] اذا كان العدد في المسجل BH سالب
واذا كان موجب اجعل محتويات الذاكرة [9000]=FF



فردي	**01**	**0000 0001**
زوجي	02	0000 0010
فردي	03	0000 0011
زوجي	04	0000 0100

```
MOV  BH,04H
MOV  AL,00
MOV  AH,0FFH
SHL  BH,01
JNC   PPP
      MOV  [9000H],AL
      JMP   EEE
PPP: MOV   [9000H],AH
EEE:  RET
```

هل العدد سالب
ام موجب

```
        MOV  AH,00
        MOV AL,FF
        SHR    bH,01
        Jnc     xxxx
        Mov  [9000],AL
        JMP  RRR
Xxxx :      MOV [9000H],AH
RRR:         RET
```

القفز المشروط& ASCII

- (JNZ,JZ)
- JNZ :ينفذ القفز في هذا الايعاز اذا كان ZF=0
- JZ : ينفذ القفز في هذا الايعاز اذا كان ZF=1
- ملاحظة:

INC: هذا الايعاز يقوم بزيادة بمقدار واحد يشبه الايعاز ADD

INC BH ➔ ADD BH,1

DEC: هذا الايعاز يقوم بنقصان بمقدار واحد يشبه الايعاز SUB

DEC BH ➔ SUB BH,1

اكتب برنامج لملى الذاكرة بالقيم 1,2,3,4,5 ابتدا من العنوان 5000 الى نهاية العنوان 5004

```
        MOV  DI,5000H
        MOV   CL,05H
        MOV AL,1
BBB:    MOV  [DI],ALH
        INC    AL
        INC    DI
        DEC    CL
        JNZ   BBB
```

AL	DI	CL	[DI]
1	5000	05	1
2	5001	4	2
3	5002	3	3
4	5003	2	4
5	5004	1	5
6	5005	0	

5004	5
5003	4
5002	3
5001	2
5000	1

الحل:

اولا: يجب الاشارة الى بداية الذاكرة عن طريق مسجل

ثانيا: استخدام عداد لعدد مواقع الذاكرة

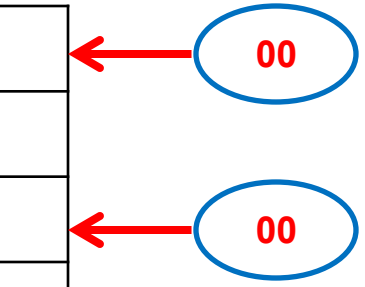
ثالثا : تحميل مسجل ALبالقيمة واحد

اكتب برنامج لتصفير محتويات الذاكرةابتداءا من العنوان [9000] الى العنوان [9003]اذا كان الذاكرة تحتوي على العدد 5

```
        MOV  SI,9000H
        MOV   CL,04H
BBB:    CMP   [SI],5H
        JNZ    XXX
        MOV  [SI],00H
XXX:    INC    SI
        DEC    CL
        JNZ   BBB
```

SI	CL	[SI]
9000	04	99
9001	**3**	**5**
9002	**2**	**1**
9003	**1**	**5**
9004	**0**	

9003	~~5~~
9002	01
9001	~~5~~
9000	99



الحل:

اولا: يجب الاشارة الى بداية الذاكرة عن طريق مسجل

ثانيا: استخدام عداد لعدد مواقع الذاكرة

ثالثا : المقارنة مع الرقم 5

رابعا تصفير الموقع المطابق للعدد 5

ASCII
American Standard Code for Information Interchange

- هي مجموعة رموز ونظام ترميز مبني على الابجدية اللاتينية بالشكل الذي تستخدم به في الانكليزية الحديثة. من أكثر الاستخدامات شيوعا للنصوص المكتوبة بالآسكي، استخدامها في أنظمة الحاسوب
- فعند الضغط على مفتاح 0 في لوحة االمفاتيح يتم ارسال الشفرة 30H والواحد 31Hوهكذا

حدول الرموز

حرف لاتيني كبيرA	A	1000001	41
حرف لاتيني كبيرB	B	1000010	42
حرف لاتيني كبيرC	C	1000011	43
حرف لاتيني كبيرD	D	1000100	44
حرف لاتيني كبيرE	E	1000101	45
حرف لاتيني كبيرF	F	1000110	46
حرف لاتيني كبيرG	G	1000111	47
حرف لاتيني كبيرH	H	1001000	48
حرف لاتيني كبيرI	I	1001001	49
حرف لاتيني كبيرJ	J	1001010	4A
حرف لاتيني كبيرK	K	1001011	4B
حرف لاتيني كبيرL	L	1001100	4C
حرف لاتيني كبيرM	M	1001101	4D
حرف لاتيني كبيرN	N	1001110	4E

حرف لاتيني صغير a	a	1100001	61
حرف لاتيني صغير b	b	1100010	62
حرف لاتيني صغير c	c	1100011	63
حرف لاتيني صغير d	d	1100100	64
حرف لاتيني صغير e	e	1100101	65
حرف لاتيني صغير f	f	1100110	66
حرف لاتيني صغير g	g	1100111	67
حرف لاتيني صغير h	h	1101000	68

رقم صفر	0	0110000	30
رقم واحد	1	0110001	31
رقم إثنان	2	0110010	32
رقم ثلاثة	3	0110011	33
رقم أربعة	4	0110100	34
رقم خمسة	5	0110101	35
رقم ستة	6	0110110	36
رقم سبعة	7	0110111	37
رقم ثمانية	8	0111000	38
رقم تسعة	9	0111001	39

اكتب برنامج لملى الذاكرة بالقيم ABCDE ابتدا من العنوان 3000 الى نهاية العنوان 3004

```
        MOV  DI,3000H
        MOV   CL,05H
        MOV AL,41H
BBB:    MOV  [DI],ALH
        INC     AL
        INC     DI
        DEC     CL
        JNZ   BBB
```

AL	DI	CL	[DI]
41=A	3000	05	A
42=B	3001	4	B
43=C	3002	3	C
44=D	3003	2	D
45=E	3004	1	E
46=F	3005	0	

3004	E
3003	D
3002	C
3001	B
3000	A

الحل:

اولا: يجب الاشارة الى بداية الذاكرة عن طريق مسجل

ثانيا: استخدام عداد لعدد مواقع الذاكرة

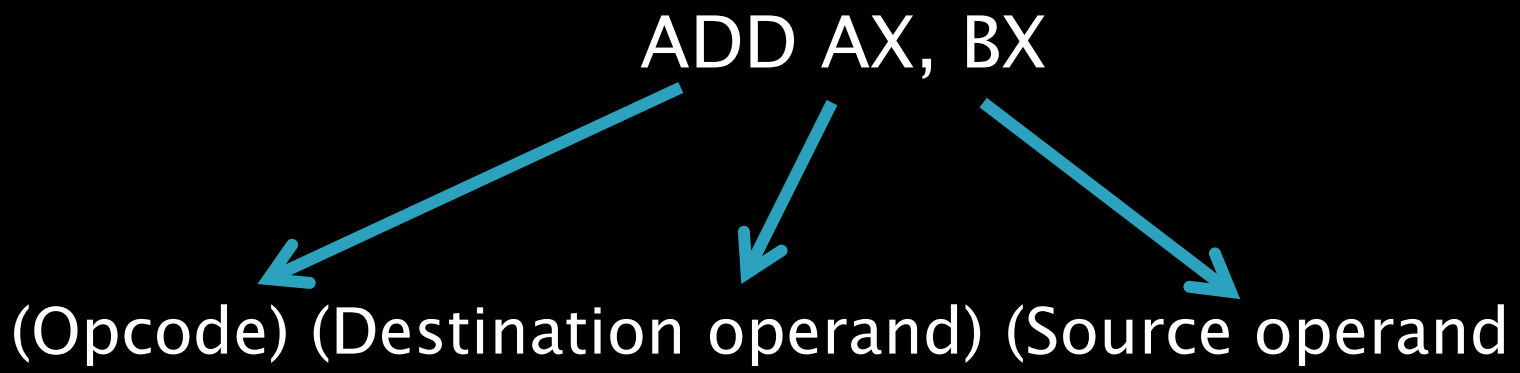
ثالثا : تحميل مسجل ALبالقيمة 41

اختبار

- اكتب برنامج لاستبدال الحرف Z اينما وجد بالحرف M في ذاكرة تبدأ بالعنوان 3200 يوجد فيها الكلمة بحجم 6 BYTE

3205	**T**
3204	Z
3203	P
3202	Z
3201	T
3200	C

Instruction Format 8086



general instruction form for the 8086

- An instruction can be coded with 1 to 6 bytes
- – 3 R/M Displacement or data (optional) up

Opcode – 6	D – 1	W –1	1 st byte
MOD – 2	Reg – 3	R/M – 3	3 byte
Displacement or data (optional) up to 4 bytes			

Converting Assembly Language Instructions to Machine Code

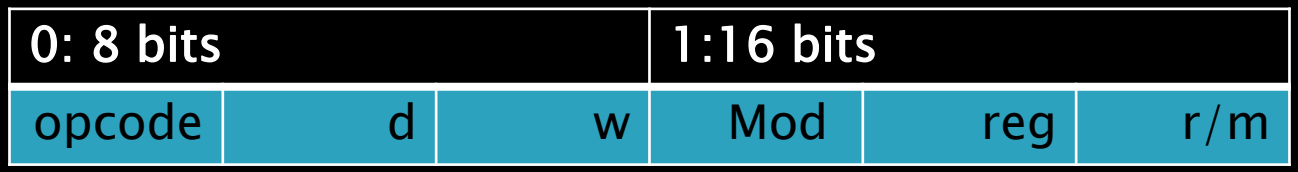
- Byte 1 contains three kinds of information
- Opcode field (6 bits) specifies the operation (add, subtract, move)
- Register Direction Bit (D bit) Tells the register operand in REG field in byte 2 is source or destination operand

1: destination 0: source

MOV Ah , [3000H] MOV [7000H] , CX

D=1 D=0

- Data Size Bit (W bit) Specifies whether the operation will be performed on
- 8-bit(W=0) or 16-bit(W=1) data
- Add al,90 pop bx



Byte 2 has three fields

- Mode field (MOD)
- Register field (REG) used to identify the register for the first operand
- Register/memory field (R/M field)

reg	W=0	W=1
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

- 2-bit MOD field and 3-bit R/M field together speci
- the second operand

MODE FILDE INCODING

CODE	EXPLENATION
00	Memory mod no displasment follows
01	Memory mod 8-bit displasment follows
10	Memory mod 16-bit displasment follows
11	Register mod(no displasment)

excepting when r/m=110 then 16-bit displasment follow

Regsiter/memory (r/m) filedncoding

MOD = 11			EFFECTIVE ADDRESS CALCULATION			
R/M	W = 0	W = 1	R/M	MOD = 00	MOD = 01	MOD = 10
000	AL	AX	000	(BX) + (SI)	(BX) + (SI) + D8	(BX) + (SI) + D16
001	CL	CX	001	(BX) + (DI)	(BX) + (DI) + D8	(BX) + (DI) + D16
010	DL	DX	010	(BP) + (SI)	(BP) + (SI) + D8	(BP) + (SI) + D16
011	BL	BX	011	(BP) + (DI)	(BP) + (DI) + D8	(BP) + (DI) + D16
100	AH	SP	100	(SI)	(SI) + D8	(SI) + D16
101	CH	BP	101	(DI)	(DI) + D8	(DI) + D16
110	DH	SI	110	DIRECT ADDRESS	(BP) + D8	(BP) + D16
111	BH	DI	111	(BX)	(BX) + D8	(BX) + D16

EXAMPLE

- MOV AL,02
- Opcode for MOV = 1011

opcode	W	REG

- W bit = 0 (8-bits)
- Reg=000
-

opcode	W	REG
1011	0	000

B0 02

اوجد CODE الايعازات التالية

- MOV DX,6622H

1011	W	REG

Ascii

في أوائل الأربعينيات من القرن الماضي ، كانت أجهزة الحاسب الآلي تؤدي عمليات الحساب من جمع و طرح إلخ و ليس بإستطاعتها التعامل مع الحروف و الرموز .و كان من الضروري جدا في ذلك الوقت إيجاد طريقة معينة لجعل جهاز الحاسب يستطيع قراءة الحروف و طباعتها ، لأنه في الواقع لايفهم إلا لغة الأرقام .

و من هنا أتت الفكرة بأن يتم إسناد عدد من الأرقام إلى الحروف الأبجدية و الرموز لكي يستطيع الجهاز قراءتها و التعامل معها .

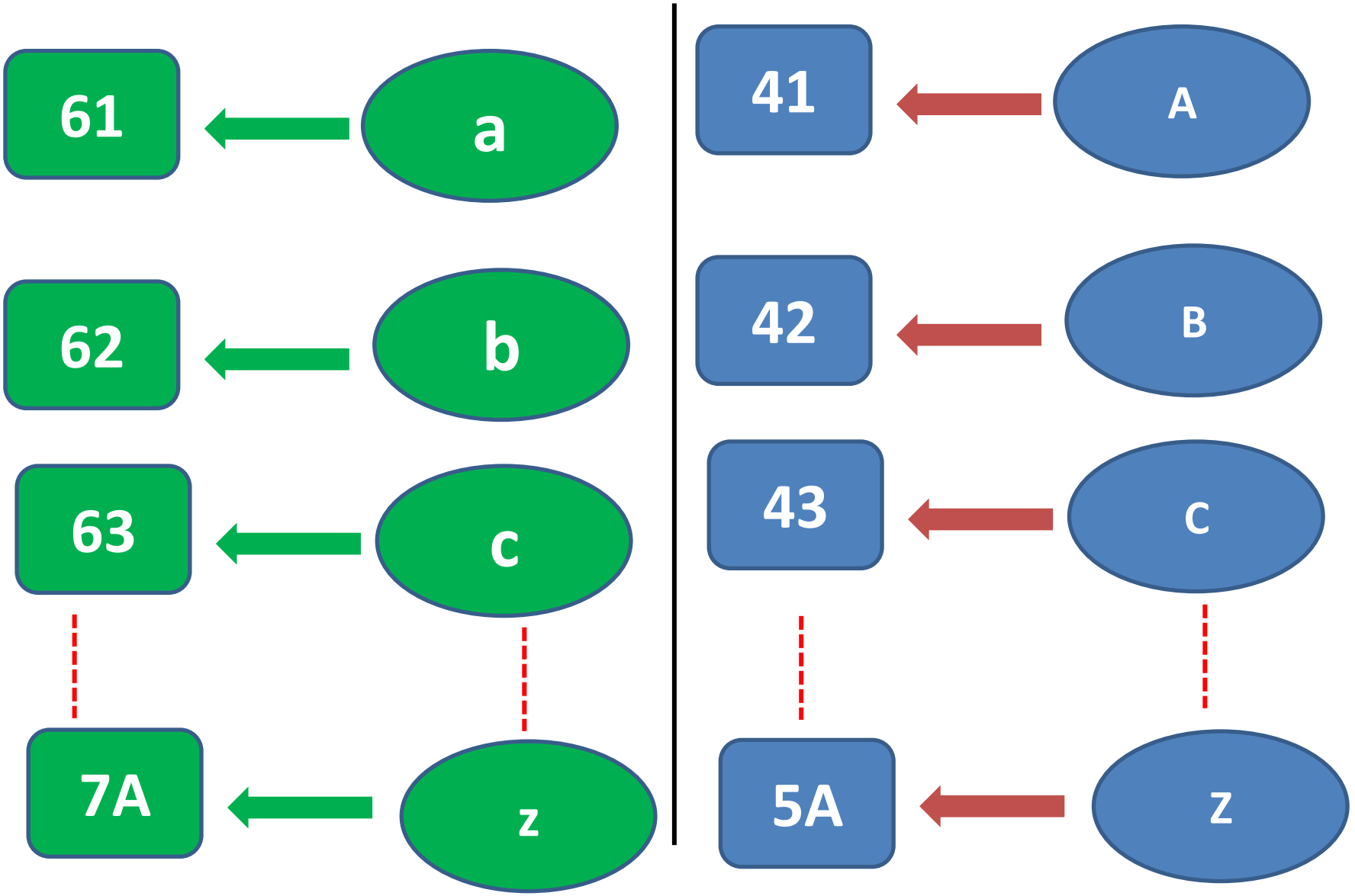
تعريف ascii

- لأسكي كود American Standard Code for Information Interchangeو إختصاره ASCIIهو نظام ترميز أنشئ سنة ١٩٦٣ بالإعتماد على النظام اللذي كان يستخدم في الآلة الكاتبة TELEPRINTو هو نظام يستخدم ٧ بت فقط لتخزين المعلومات. هذا النظام يهدف إلى تمكين المستخدم من إستعمال الحروف و النصوص في جهاز الحاسب لكتابة نص معين .و بما أن الكمبيوتر لا يمكنه قراءة إلا الأرقام فقط ، فلقد قام العاملون على هذا النظام بإسناد قيمة رقمية لكل حرف و رمز و بذلك نجحوا في تمثيلها على أجهزة الحواسيب .

امثلة عن ASCII

- مثال على ذلك :
- حرف : Aالرقم المقابل له ب الأسكي كود يساويHEX 41 .
  حرف : aالرقم المقابل له ب الأسكي كود يساوي HEX 61
- و في هذا المثال يتم تقديم الحروف و الرموز في الأسكي كود و قيمتها بالنسبة لأنظمة العد المختلفة .

Dec : نظام العد العشري .
Hex : نظام العد السداسي عشرة .
Oct : نظام العد الثماني .
Bin : نظام العد الثنائي .
Char : نظام الأسكي كود


61
a
62
b
63
c
7A
z
41
A
42
B
43
C
5A
Z

Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char	Dec	Hex	Oct	Bin	Char
0	0x00	000	0000000	NUL	32	0x20	040	0100000	space	64	0x40	100	1000000	@	96	0x60	140	1100000	`
1	0x01	001	0000001	SOH	33	0x21	041	0100001	!	65	0x41	101	1000001	A	97	0x61	141	1100001	a
2	0x02	002	0000010	STX	34	0x22	042	0100010	"	66	0x42	102	1000010	B	98	0x62	142	1100010	b
3	0x03	003	0000011	ETX	35	0x23	043	0100011	#	67	0x43	103	1000011	C	99	0x63	143	1100011	c
4	0x04	004	0000100	EOT	36	0x24	044	0100100	$	68	0x44	104	1000100	D	100	0x64	144	1100100	d
5	0x05	005	0000101	ENQ	37	0x25	045	0100101	%	69	0x45	105	1000101	E	101	0x65	145	1100101	e
6	0x06	006	0000110	ACK	38	0x26	046	0100110	&	70	0x46	106	1000110	F	102	0x66	146	1100110	f
7	0x07	007	0000111	BEL	39	0x27	047	0100111	'	71	0x47	107	1000111	G	103	0x67	147	1100111	g
8	0x08	010	0001000	BS	40	0x28	050	0101000	(	72	0x48	110	1001000	H	104	0x68	150	1101000	h
9	0x09	011	0001001	TAB	41	0x29	051	0101001	)	73	0x49	111	1001001	I	105	0x69	151	1101001	i
10	0x0A	012	0001010	LF	42	0x2A	052	0101010	*	74	0x4A	112	1001010	J	106	0x6A	152	1101010	j
11	0x0B	013	0001011	VT	43	0x2B	053	0101011	+	75	0x4B	113	1001011	K	107	0x6B	153	1101011	k
12	0x0C	014	0001100	FF	44	0x2C	054	0101100	,	76	0x4C	114	1001100	L	108	0x6C	154	1101100	l
13	0x0D	015	0001101	CR	45	0x2D	055	0101101	-	77	0x4D	115	1001101	M	109	0x6D	155	1101101	m
14	0x0E	016	0001110	SO	46	0x2E	056	0101110	.	78	0x4E	116	1001110	N	110	0x6E	156	1101110	n
15	0x0F	017	0001111	SI	47	0x2F	057	0101111	/	79	0x4F	117	1001111	O	111	0x6F	157	1101111	o
16	0x10	020	0010000	DLE	48	0x30	060	0110000	0	80	0x50	120	1010000	P	112	0x70	160	1110000	p
17	0x11	021	0010001	DC1	49	0x31	061	0110001	1	81	0x51	121	1010001	Q	113	0x71	161	1110001	q
18	0x12	022	0010010	DC2	50	0x32	062	0110010	2	82	0x52	122	1010010	R	114	0x72	162	1110010	r
19	0x13	023	0010011	DC3	51	0x33	063	0110011	3	83	0x53	123	1010011	S	115	0x73	163	1110011	s
20	0x14	024	0010100	DC4	52	0x34	064	0110100	4	84	0x54	124	1010100	T	116	0x74	164	1110100	t
21	0x15	025	0010101	NAK	53	0x35	065	0110101	5	85	0x55	125	1010101	U	117	0x75	165	1110101	u
22	0x16	026	0010110	SYN	54	0x36	066	0110110	6	86	0x56	126	1010110	V	118	0x76	166	1110110	v
23	0x17	027	0010111	ETB	55	0x37	067	0110111	7	87	0x57	127	1010111	W	119	0x77	167	1110111	w
24	0x18	030	0011000	CAN	56	0x38	070	0111000	8	88	0x58	130	1011000	X	120	0x78	170	1111000	x
25	0x19	031	0011001	EM	57	0x39	071	0111001	9	89	0x59	131	1011001	Y	121	0x79	171	1111001	y
26	0x1A	032	0011010	SUB	58	0x3A	072	0111010	:	90	0x5A	132	1011010	Z	122	0x7A	172	1111010	z
27	0x1B	033	0011011	ESC	59	0x3B	073	0111011	;	91	0x5B	133	1011011	[	123	0x7B	173	1111011	{
28	0x1C	034	0011100	FS	60	0x3C	074	0111100	<	92	0x5C	134	1011100	\	124	0x7C	174	1111100	|
29	0x1D	035	0011101	GS	61	0x3D	075	0111101	=	93	0x5D	135	1011101	]	125	0x7D	175	1111101	}
30	0x1E	036	0011110	RS	62	0x3E	076	0111110	>	94	0x5E	136	1011110	^	126	0x7E	176	1111110	~
31	0x1F	037	0011111	US	63	0x3F	077	0111111	?	95	0x5F	137	1011111	_	127	0x7F	177	1111111	DEL

المقاطعات في المعالج الدقيق
Interrupts in 8086 microprocesso

- ما هي المقاطعة في المعالج الدقيق 8086 Interrupt؟
- ما هي المهام التي يقوم بها المعالج الدقيق 8086عندما يواجه مقاطعة؟
- أنواع المقاطعات في المعالج الدقيق 8086

ما هي المقاطعة في المعالج الدقيق 8086 Interrupt؟

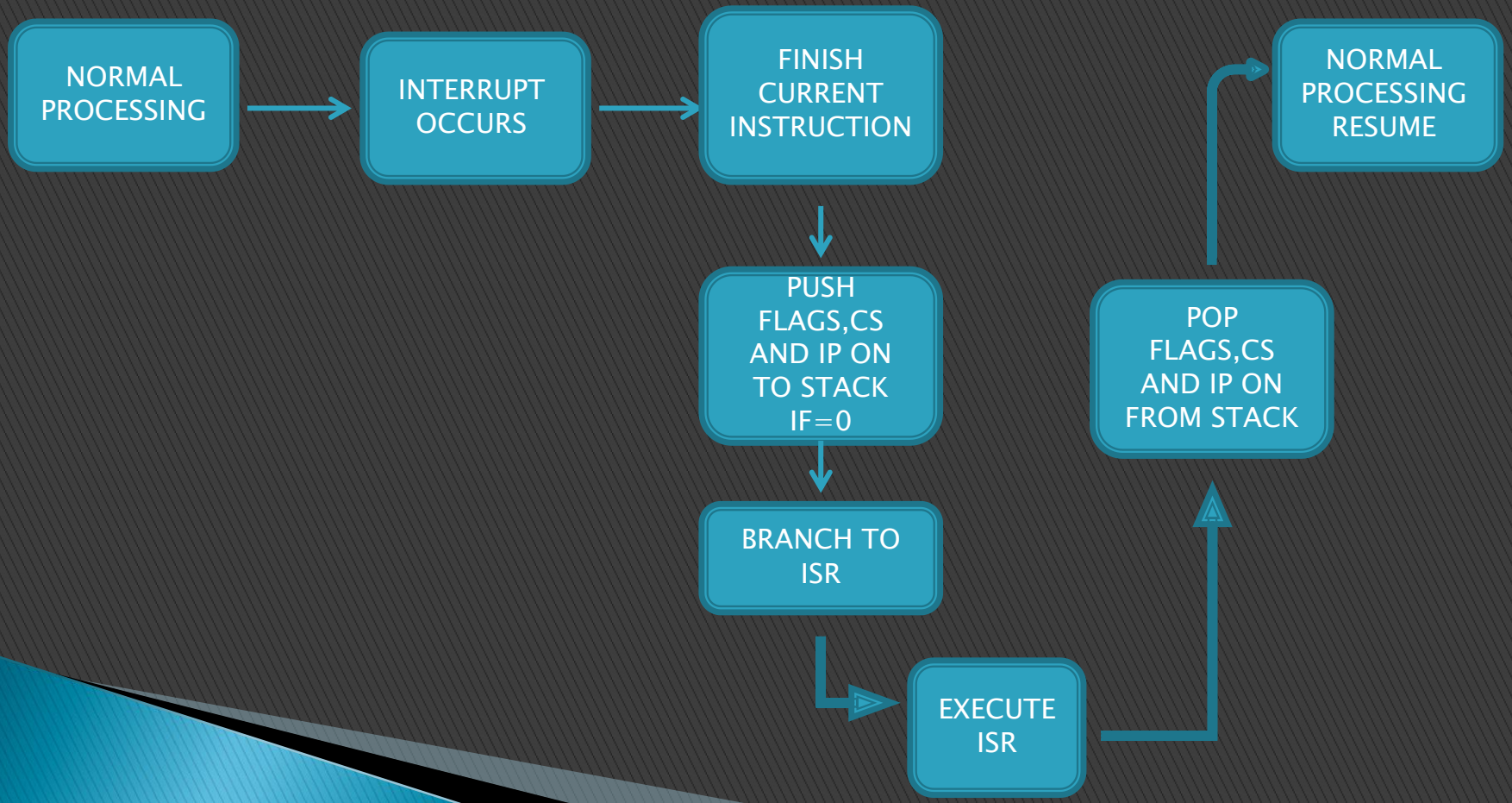
المقاطعة هي طريقة إنشاء توقف مؤقت أثناء تنفيذ البرنامج وتسمح للأجهزة الطرفية بالوصول إلى المعالج الدقيق، يستجيب المعالج الدقيق لهذا المقاطعة باستخدام (ISR Interrupt Service Routine) –، وهو برنامج قصير لإرشاد المعالج .

اذا المقاطعة هي حالة توقف المعالج الدقيق مؤقتًا عن العمل في مهمة مختلفة ثمّ العودة إلى مهمته السابقة، المقاطعة هي حدث أو إشارة تتطلب انتباه وحدة المعالجة المركزية ((CPU، يسمح هذا التوقف للأجهزة الطرفية بالوصول إلى المعالج الدقيق، عند حدوث مقاطعة، يكمل المعالج تنفيذ التعليمات الحالية ويبدأ في تنفيذ روتين خدمة المقاطعة أو معالجة المقاطعة (Interrupt Handler)

(ISR)هو برنامج يخبر المعالج بما يجب القيام به عند حدوث المقاطعة، بعد تنفيذ (ISR)، تعود السيطرة إلى الروتين الرئيسي حيث تمت مقاطعتها. حول كيفية التعامل مع المقاطعة.

- المشكلة التي يواجهها المعالج اثناء تفاعله مع الاجهزة المرتبطة هو عدم معرفة الطرف هل جاهز للعمل ام لا.
- ولحل هذه المشكلة يتم فحص اشارة القطع عند نهاية كل ايعاز اذا كانت الاشارة فعالة سيتم الانتقال لتنفيذ برنامج خدمة القطع ISR

X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	1	0	1	0	1	0	1



ما هي المهام التي يقوم بها المعالج الدقيق 8086عندما يواجه مقاطعة؟

- عندما يواجه المعالج الدقيق مقاطعة:

يتم دفع قيمة سجل العلم (flag register) في المكدس (stack)، هذا يعني أنّ قيمة (((SP (Stack Pointer، يتم تقليلها أولاً بمقدار (٢) ثمّ يتم دفع قيمة سجل العلم إلى عنوان ذاكرة مقطع المكدس.

- يتم دفع قيمة عنوان ذاكرة البداية لـ (Code Segment) (CS) في المكدس.
- يتم دفع قيمة (Instruction Pointer) (IP)مؤشر التعليمات للايعاز التالي إلى المكدس

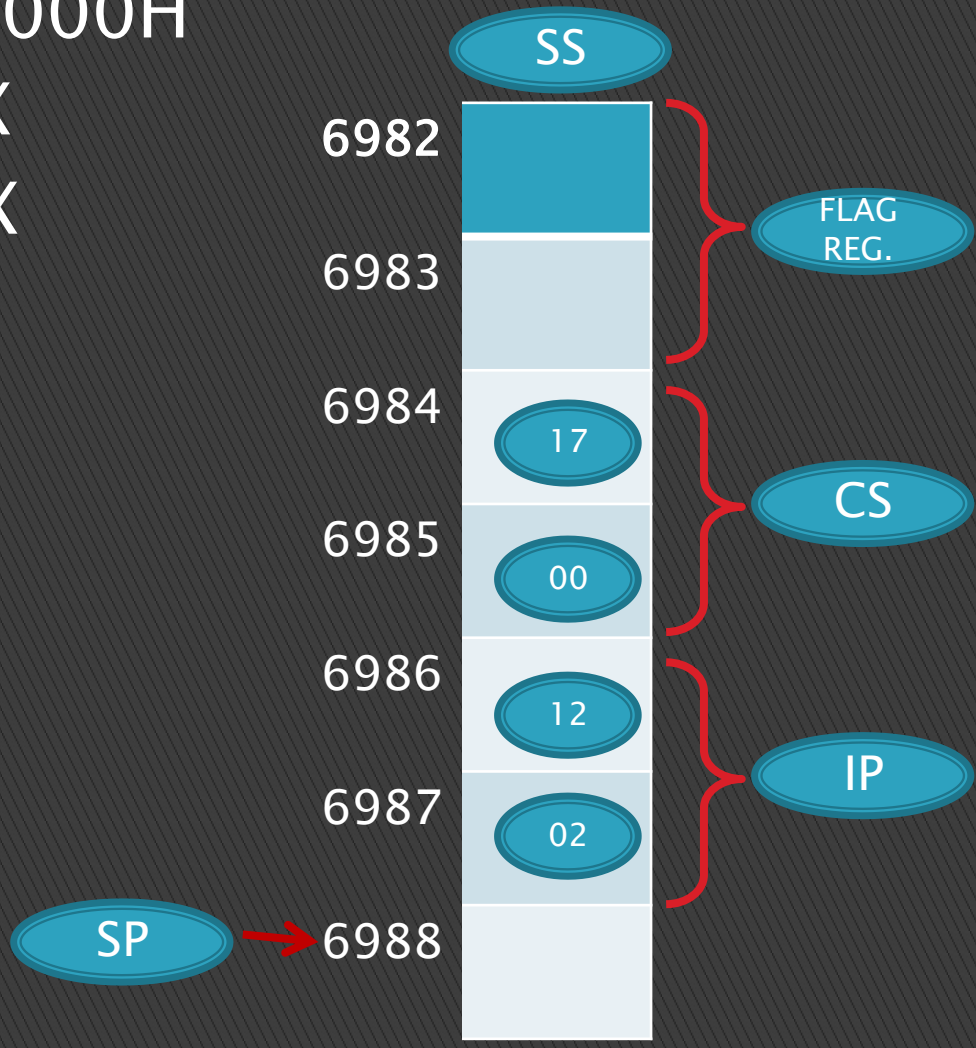
- يتم تحميلIP, (CS)من جدول القطوعات الموجود من العنوان
- 0000-03FF وهو كيلو بايت واحد يحوي على 256 نوع من القطوعات
- جعل قيمة العلم TF,IF تساوي صفر

اذا حدث قطع خارجي عند تنفيذ الايعاز الاول من البرنامج التالي كيف سيعالجها CPU اذا علمت SP=6988

1700:1200 MOV AX,9000H
1700:1202 ADD CX,BX
1700:1204 SUB DX,AX

CS

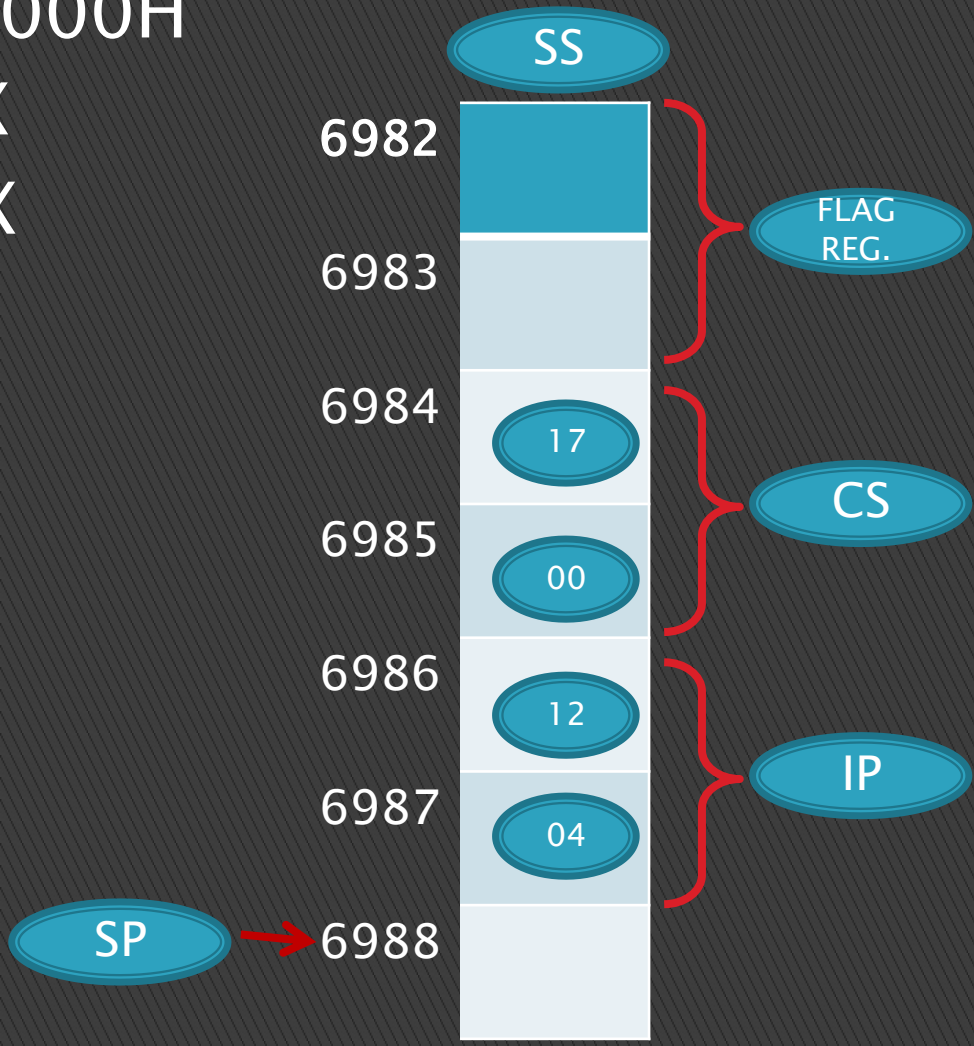
IP

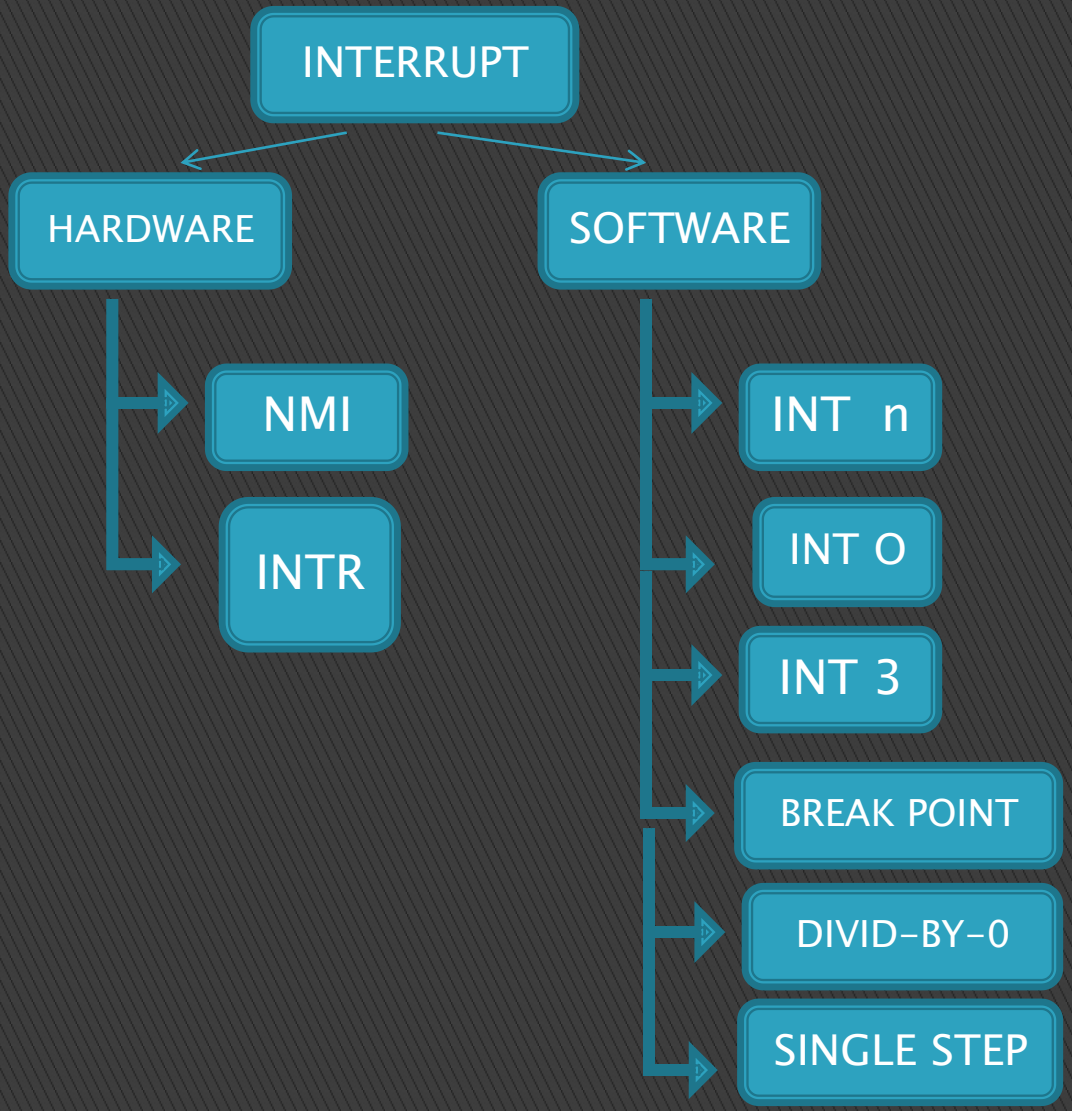


اذا حدث قطع خارجي عند تنفيذ الايعاز الثاني من البرنامج التالي كيف سيعالجها CPU اذا علمت SP=6988

1700:1200 MOV AX,9000H
1700:1202 ADD CX,BX
1700:1204 SUB DX,AX

CS IP




INTERRUPT
HARDWARE
SOFTWARE
NMI
INTR
INT n
INT O
INT 3
BREAK POINT
DIVID-BY-0
SINGLE STEP

اولوية القطوعات

HARDWARE

SOFTWARE

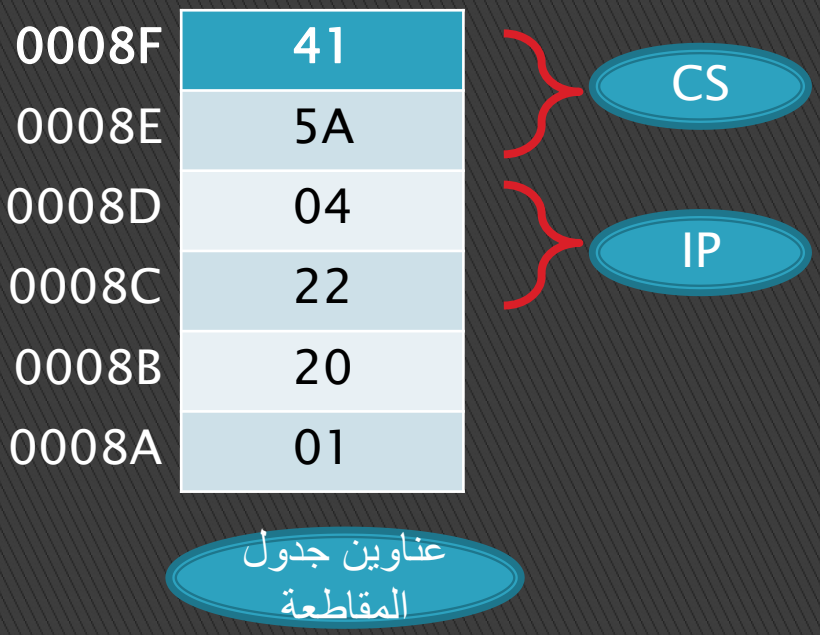
NAME	PRIORTY	ADDRES VECTOR TABLE
NMI	2	00008-0000B
INTR	3	n*4
INT n	1	n*4
INT 3(BREAK POINT)	1	0000C-0000F
INT O	1	00010-00013
DIVID-BY-0	1	00000-00003
SINGLE STEP	4	00004-00007

- عند وجود ايعاز حالي ينفذ لايستجاب للقطع الا عند الانتهاء من تنفيذ الايعاز
- القطوعات الداخلية لها اولوية الاعلى (ماعدا single step) في حالة حدوثها في ان واحد
- اذا حدث قطع INTR والايعاز الحالي يسبب القسمة على صفر فان القسمة على صفر تنفذ اولا
- اذا حدث INTR,NMI في ان واحد فان NMI ينفذ اولا
- عند حدوث القطوعات تباعا الخارجي له اولوية على الداخلي
- اذا كان المعالج ينفذ قطع داخلي وحدث قطع خارجي فان القطع الداخلي سوف يتوقف وينفذ القطع الخارجي
- NMI:non maskbel interrupt لايمكن حجب هذا القطع يتم الاستجابة له في حالة البدء ISR

مثال :ماهي قيمة CS,IP التي سيتم تحميلها عند تنفيذ القطع الموجود في البرنامج التالي علما ان عناوين جدول المقاطعة كما موضح اعلاه

INT 23H

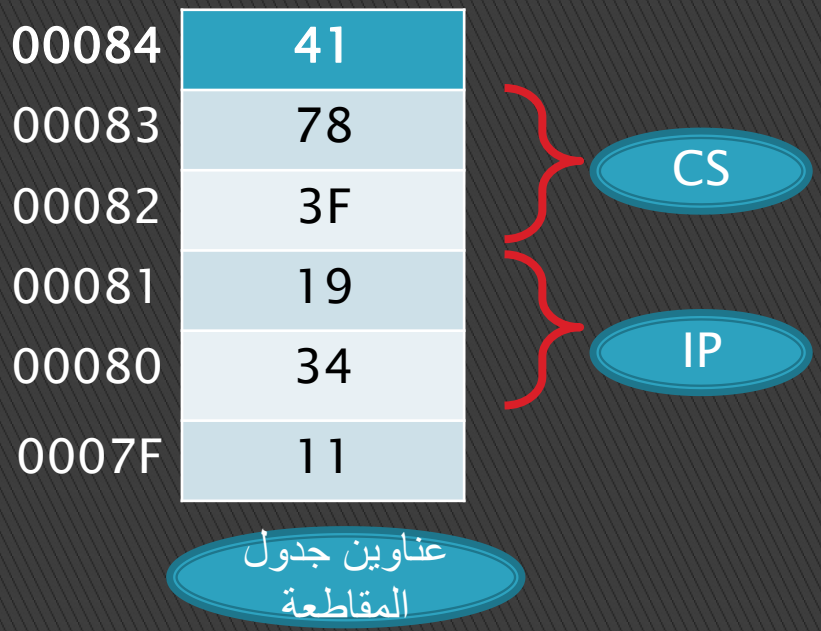
١. نضرب الرقم 4*23
٢. 0000 0000 1000 1100=0000 0000 0010 0011*4
٣. 008C
٤. IP=0422
٥. CS=415A



مثال :ماهي قيمة CS,IP التي سيتم تحميلها عند تنفيذ القطع الموجود في البرنامج التالي علما ان عناوين جدول المقاطعة كما موضح اعلاه

INT 20H

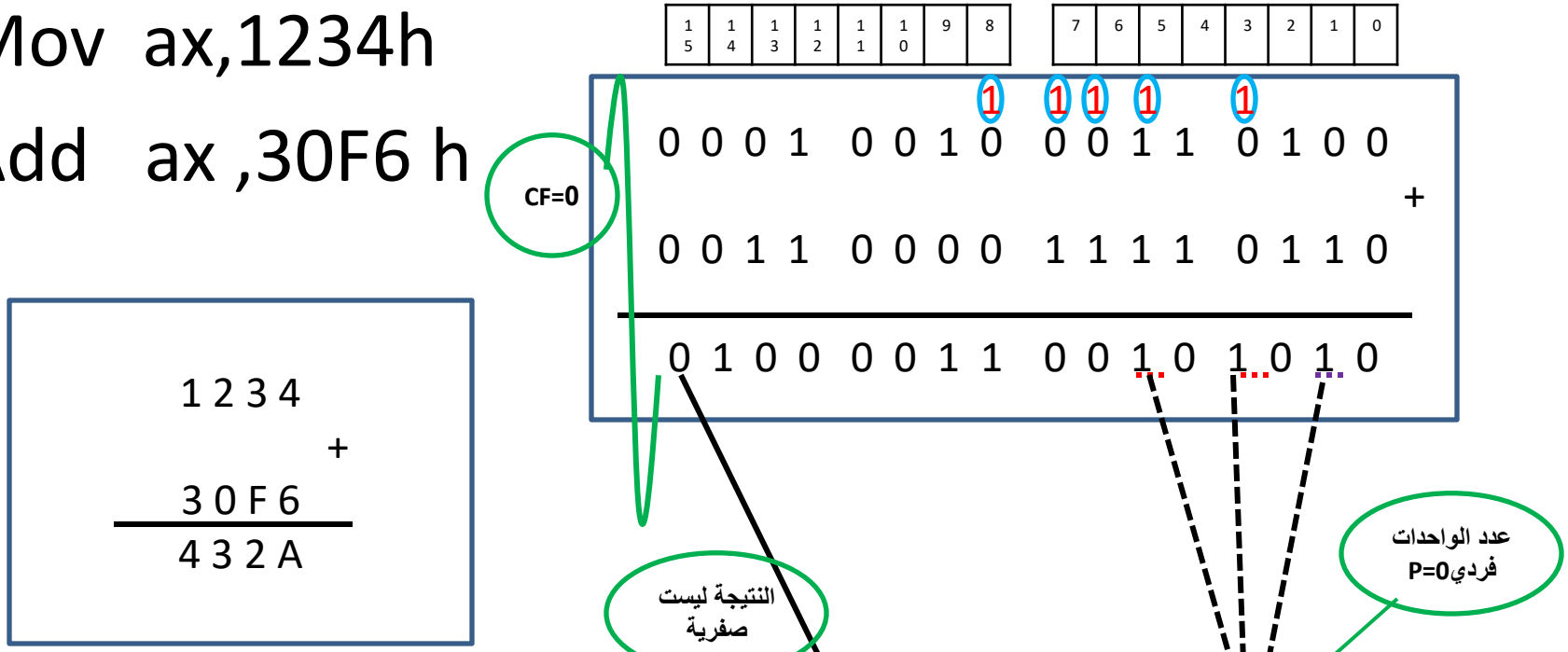
١. نضرب الرقم 4*20
٢. 0000 0000 1000 0000=0000 0000 0010 0000*4
٣. 0080
٤. IP=1934
٥. CS=783F



تاثير ايعاز الجمع (add)
على مسجل الاعلام(flag register)

- مثال ماهي الاعلام التي ستتاثر بالايعاز التالي

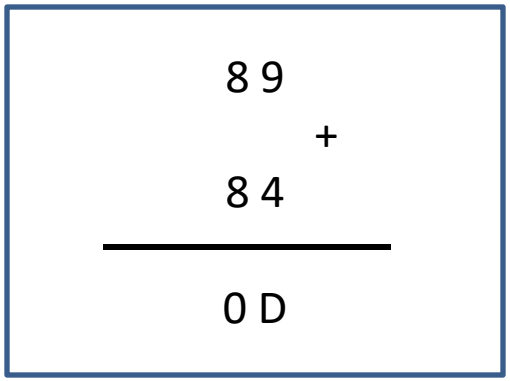
- Mov ax,1234h
- Add ax ,30F6 h

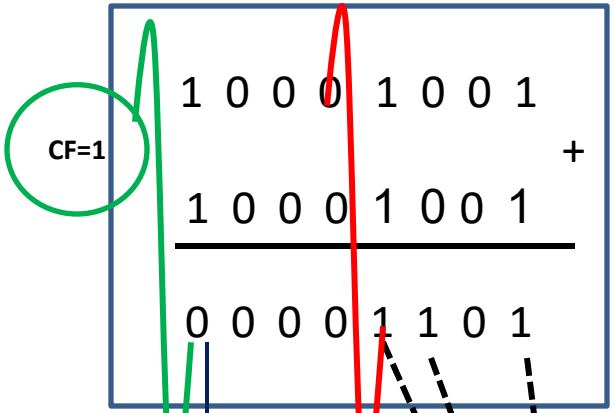


15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	0	0	0	0	0

تاثير ايعاز الجمع (add)
على مسجل الاعلام(flag register)

- MOV DH,89 H
- Add DH, 8cH

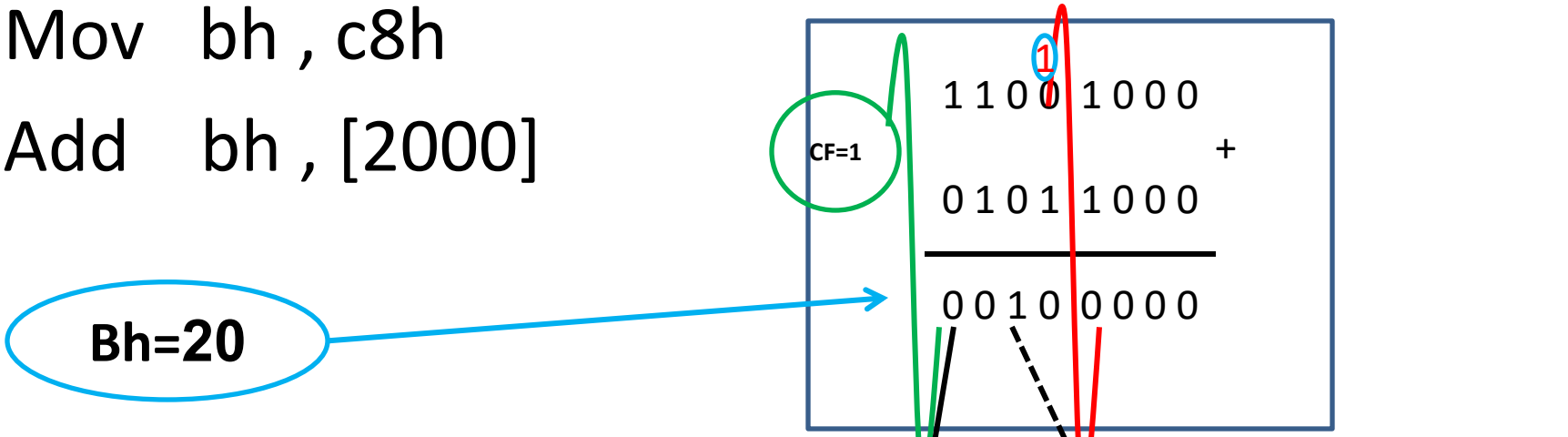




15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	0	0	0	0	1

مثال:اذا علمت ان محتوى الذاكرة [2000]=58
ماهو تاثير البرنامج التالي على مسجل الاعلام

- Mov bh , c8h
- Add bh , [2000]



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	OF	DF	IF	TF	Z	S	X	A	X	P	X	C
								0	0	0	1	0	0	0	1

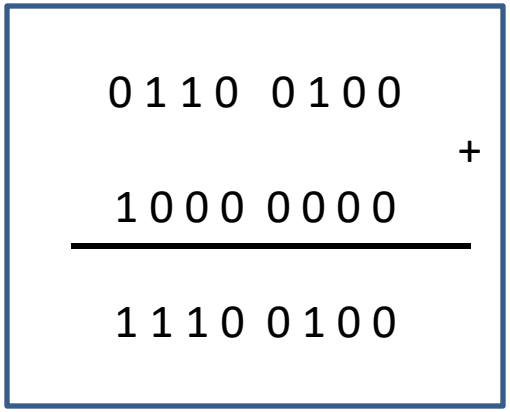
اختبار

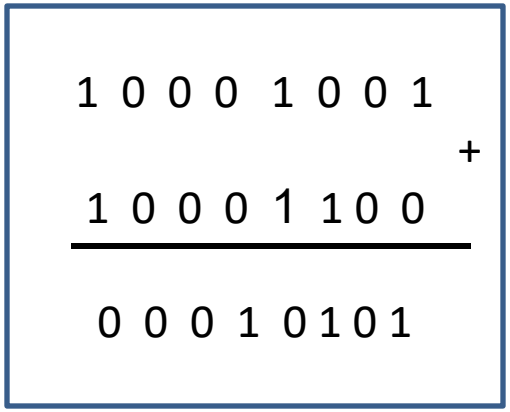
- ماهو تاثير البرنامج التالي على مسجل الاعلام وماهي قيمة المسجل siوماهي محتويات الذاكرة بعد تنفيذ البرنامج التالي
- اذا علمت ان محتوى الذاكرة [4900]=E1,[4901]=71

- Mov BL , 39 H
- ADD BL, BL
- ADD [4900] ,BL
- ADD BL,[4901]
- MOV SI,[4900]

اكتب برنامج لجعل قيمة sf,pf=1 مع بقاء جميع الاعلام اصفاربعد التنفيذ باستخدام ايعازين فقط

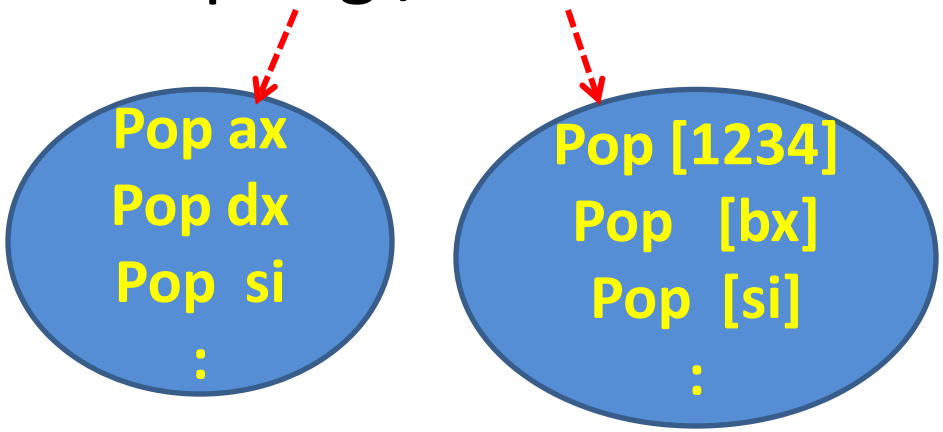
- Mov bh,64h
- Add bh,80h




1 0 0 0 1 0 0 1
+
1 0 0 0 1 1 0 0
0 0 0 1 0 1 0 1

- Pop:سحب قيم من داخل المكدس(stack) ووضعها في معامل الايعازpop
- صيغة الايعاز

- Pop reg./mem.

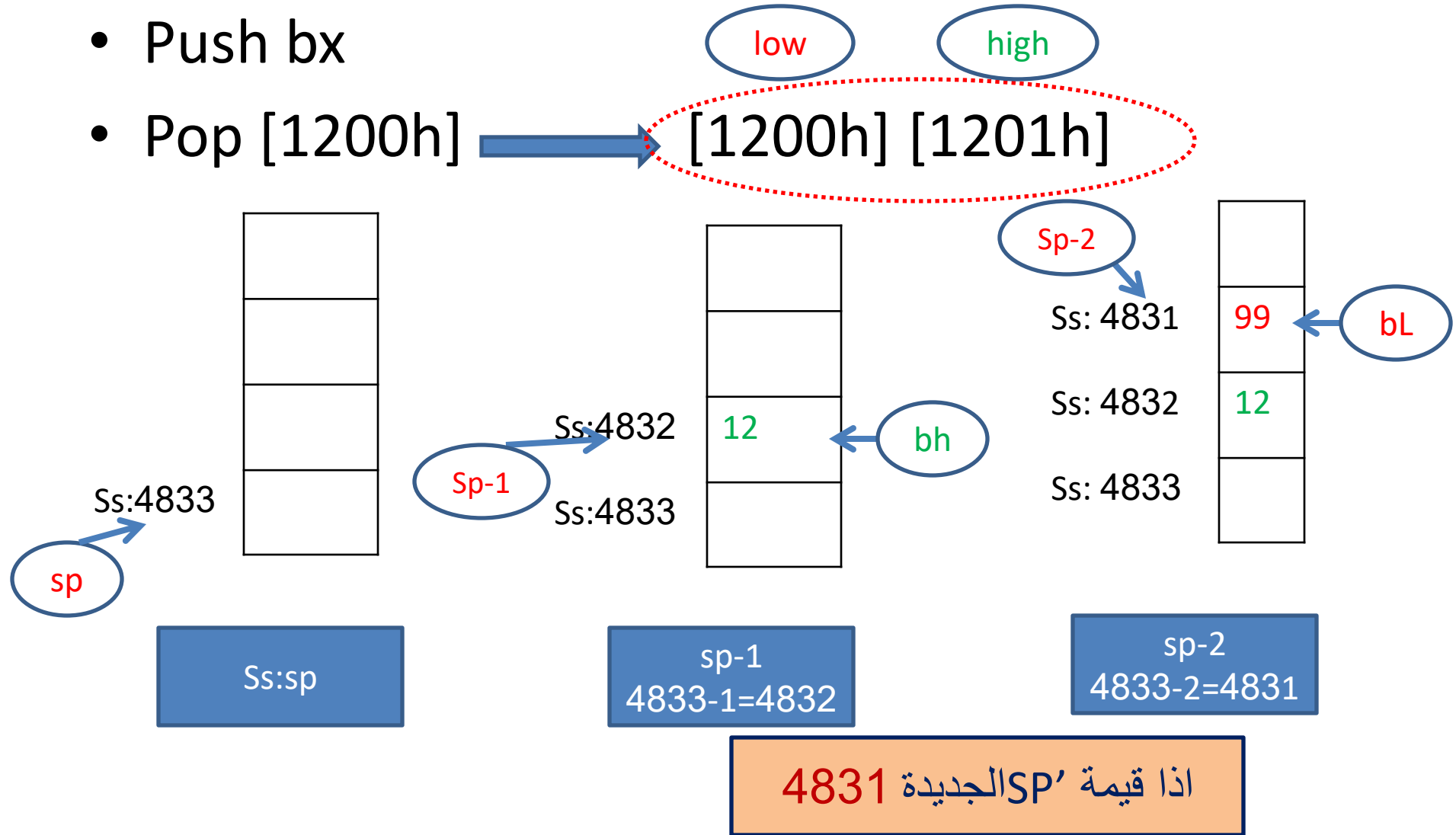


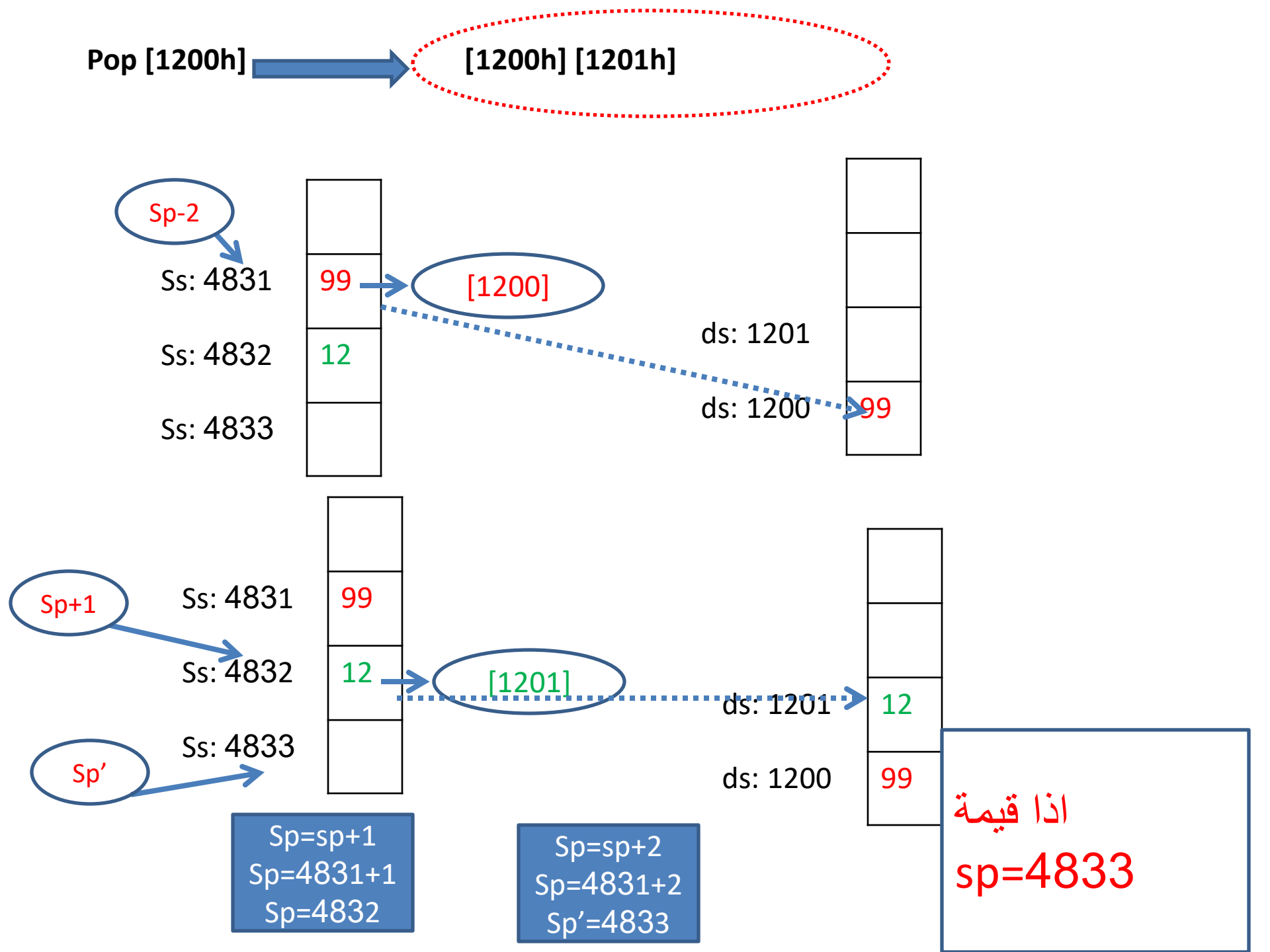
خطوات pop

- 1- سحب محتويات المكدس (التي يشير اليها sp) الى الجزء الواطىء من المعامل
- Low=[sp]
- 2- زيادة ال spبمقدار واحد
- sp+1
- 3- سحب محتويات المكدس (التي يشير اليها sp) الى الجزء العالي من المعامل
- 4- زيادة ال spبمقدار اثنان
- Sp=sp+2

مثال: لنفرض ان sp=4833h,bx=1299
بين محتويات المكدس ومحتويات sp,[1200] بعد تنفيذ التعليمات التالية

- Push bx
- Pop [1200h]




Pop [1200h]
[1200h] [1201h]
Sp-2
Ss: 4831
Ss: 4832
Ss: 4833
99
12
[1200]
ds: 1201
ds: 1200
99
Sp+1
Sp'
Ss: 4831
Ss: 4832
Ss: 4833
99
12
[1201]
ds: 1201
ds: 1200
12
99
Sp=sp+1
Sp=4831+1
Sp=4832
Sp=sp+2
Sp=4831+2
Sp'=4833
اذا قيمة
sp=4833

اختبار

استخرج من محتويات الذاكرة في مقطع مكدس ssعند العنوان 5006,5007الى مقطع الذاكرة عند العنوان 3000و3001