

المعالجات الصغيرة و لغة التجميع

محاضرة 7

تعليمات المعالج 8086

د. فادي متوج

مثال 1

اكتب برنامج بلغة التجميع يقوم بما يلي : إذا كان
المسجل AX يحتوى على رقم **سالب** يتم وضع
الرقم **1-** في المسجل BX، إذا كان AX به **صفر**
يتم وضع الرقم **0** في المسجل BX أما إذا كان
المسجل AX به رقم **موجب** يتم وضع الرقم **1** في
المسجل BX

مثال 1



CMP AX, 0 ; افحص AX

JL NEGATIVE ; AX < 0

JE ZERO ; AX = 0

JG POSITIVE ; AX > 0

NEGATIVE :

MOV BX, -1 ; put -1 in BX

JMP END ; and exit

ZERO:

MOV BX, 0 ; put 0 in BX

JMP END ; and exit

POSITIVE :

MOV BX, 1 ; put 1 in BX

END:

اكتب برنامج بلغة التجميع يقوم بما يلي : إذا كان المسجل AX يحتوى على رقم **سالب** يتم وضع الرقم **1-** في المسجل BX، إذا كان AX به **صفر** يتم وضع الرقم **0** في المسجل BX أما إذا كان المسجل AX به رقم **موجب** يتم وضع الرقم **1** في المسجل BX



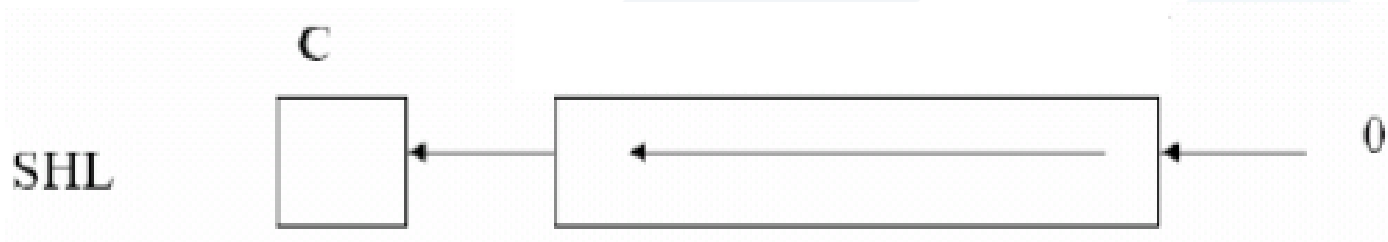
جامعة
المنارة
MANARA UNIVERSITY

مثال 2

إذا كان $DH=8AH$ و $CL = 3$ ما هي محتويات المسجلين CL و DH وكذلك علم الحمل بعد تنفيذ الأمر:

SHL DH, CL

الحل:





جامعة
المنارة
MANARA UNIVERSITY

مثال 2

إذا كان $DH=8AH$ و $CL = 3$ ما هي محتويات المسجلين CL و DH وكذلك علم الحمل بعد تنفيذ الأمر:

SHL DH,CL

الحل:

قبل تنفيذ الأمر كانت محتويات المسجل DH هي الرقم 10001010 بعد 3 ازاحات إلى اليسار تصبح محتوياته $50h = 01010000$ بينما يحتوى المسجل CL على قيمته السابقة (الرقم 3) ويحتوي علم الحمل على القيمة '0'.



جامعة
المنارة
MANARA UNIVERSITY

مثال 3

• ما هي محتويات المسجل DH و العلم CF بعد تنفيذ الجزء التالي من البرنامج:

```
MOV DH,8Ah
```

```
MOV CL,2
```

```
SHR DH,CL
```

• الحل:





جامعة
المنارة
MANARA UNIVERSITY

مثال 3

- ما هي محتويات المسجل DH و العلم CF بعد تنفيذ الجزء التالي من البرنامج:

```
MOV DH,8Ah
```

```
MOV CL,2
```

```
SHR DH,CL
```

- **الحل:** 22h، وتكون قيمة العلم CF هي '1'



جامعة
المنارة
MANARA UNIVERSITY

مثال 4

إذا كانت محتويات المسجل DH هي الرقم 8Ah وكانت محتويات علم الحمل هي الرقم $CF=1$ والمسجل CL يحتوى على الرقم 3 ما هي محتويات المسجل DH وعلم الحمل بعد تنفيذ الأمر:

RCR DH, CL

الحل





جامعة
المنارة
MANARA UNIVERSITY

مثال 4

إذا كانت محتويات المسجل DH هي الرقم 8Ah وكانت محتويات علم الحمل هي الرقم $CF=1$ والمسجل CL يحتوى على الرقم 3 ما هي محتويات المسجل DH وعلم الحمل بعد تنفيذ الأمر:

RCR DH, CL

CF	DH	
١	10001010	القيمة الابتدائية
٠	11000101	بعد الدورة الاولى نحو اليمين
١	01100010	بعد الدورة الثانية نحو اليمين
٠	10110001	بعد الدورة الثالثة نحو اليمين

الحل

أي محتويات المسجل DH هي الرقم B1h وعلم المحمول يساوي 0

- تسمح التعليمتان **IN** و **OUT** بتبادل المعطيات بين **AL** أو **AX** مع بوابات دخل/خرج تقع في فضاء عنوان بوابات المعالج 8086 .
- تقرأ التعليمية **IN** القيمة على مدخل بوابة وتضعها في المراكم (**AL** أو **AX**) ، في حين تضع أو تُخرج التعليمية **OUT** محتوى المراكم على مخرج بوابة. يمكن عنوان البوابات الواقعة في فضاء عنوان بوابات المعالج 8086 بطريقة مباشرة أو غير مباشرة.
- عنوان مباشرة: في هذه الحالة تحتوي تعليمية **IN** و **OUT** على رقم البوابة .
- إذا جرى تعيين **AL** كمعامل وجهة فإنه يجري تبادل بايت بين **AL** والبوابة.
- إذا جرى تعيين **AX** كمعامل وجهة فإنه يجري تبادل كلمة بين **AX** والبوابة .
- **مثال**: إذا علمت أن لوحة المفاتيح keyboard للحاسوب الشخصي موصولة على البوابة ذات الرقم 60h فيمكننا قراءة هذه البوابة كما يلي:

IN AL, 60h

- عنوان غير مباشرة: يجري وضع رقم البوابة في المسجل DX ومن ثم تعنون البوابة على نحو غير مباشر.
- **مثال:** إذا علمنا أن عنوان البوابة التفرعية LTP1 للحاسوب الشخصي هو 378h فيمكننا قراءة هذه البوابة كما يلي:

```
MOV DX, 378h      ;Pointer at LPT1: data port
IN AL, DX         ;Read data from printer port.
```

• **مثال:** اكتب تعليمات تقوم بقراءة بايت من البوابة التفرعية LTP1 للحاسوب الشخصي ثم تقوم بعكس البت الأولى و من ثم تعيد إخراج البايت على نفس البوابة .

MOV DX, 378h ;Parallel printer port address.

LoopForever: IN AL, DX ;Read character from input port.

XOR AL, 1 ;Invert the LSB bit.

OUT DX, AL ;Output data back to port.

JMP LoopForever ;Repeat forever.

تعليلة PUSH

- تستخدم تعليلة **PUSH** لتخزين معطيات في المكس على نحو مؤقت لحفظ محتوى المسجلات التي يُتوقع أن تتغير قيمها خلال تنفيذ إجرائية (برنامج فرعي).
- تستخدم التعليلة **PUSHF** لتخزين محتوى مسجل الأعلام في المكس.
- عند استخدام تعليلة **PUSH** لتخزين معامل من 16 بت في المكس يجري تنفيذ الخطوات

التالية:

1. يخزن البايت الأكثر أهمية للكلمة في الموقع الذي يعنونه SP-1
2. يخزن البايت الأقل أهمية في الموقع الذي يعنونه SP-2
3. ينقص SP بمقدار 2 $SP \leftarrow SP-2$

تعليلة PUSH

- يمكن أن يكون المعامل مسجل عام أو مسجل مقطع أو موقع ذاكرة.
- مثال:

PUSHF ; push flages
PUSH AX ; push accumulator
PUSH CX
PUSH [BX] ; push the location pointed by BX



جامعة
المنارة
MANARA UNIVERSITY

تعليمات المكس

تعليمة POP

- تنسخ التعليمة **POP** في معامل التعليمة آخر بايتين خزناتا في المكس (البايتين الموجودين في قمة المكس)، ومن ثم تزيد SP بـ 2 على النحو التالي:

1. يجري نسخ البايت الأقل أهمية من الموقع الذي يعنونه SP

2. يجري نسخ البايت الأكثر أهمية من الموقع الذي يؤشر إليه SP+1

3. يزداد SP بمقدار 2 : $SP \leftarrow SP+2$

- يمكن أن يكون المعامل في التعليمة POP مسجل عام أو مسجل مقطع، ما عدا CS أو موقع ذاكرة.
- نستخدم التعليمة **POPF** لنسخ محتوى المكس في مسجل الأعلام.
- مثال:

POP [BX] ; pop the location pointed by BX

POP CX

POP AX ; pop accumulator

POPF ; pop flages

- نخزن، عادة، في بداية كل إجرائية المسجلات التي يمكن أن يتغير محتواها خلال تنفيذ الإجرائية. فمثلا تخزن التعليمات التالية مسجل الأعلام و AX و BX في المكس:

PUSHF

PUSH AX

PUSH BX

وقبل العودة من الإجرائية نسحب المسجلات من المكس بالترتب العكسي، أي المخزن أخيراً يسحب أولاً:

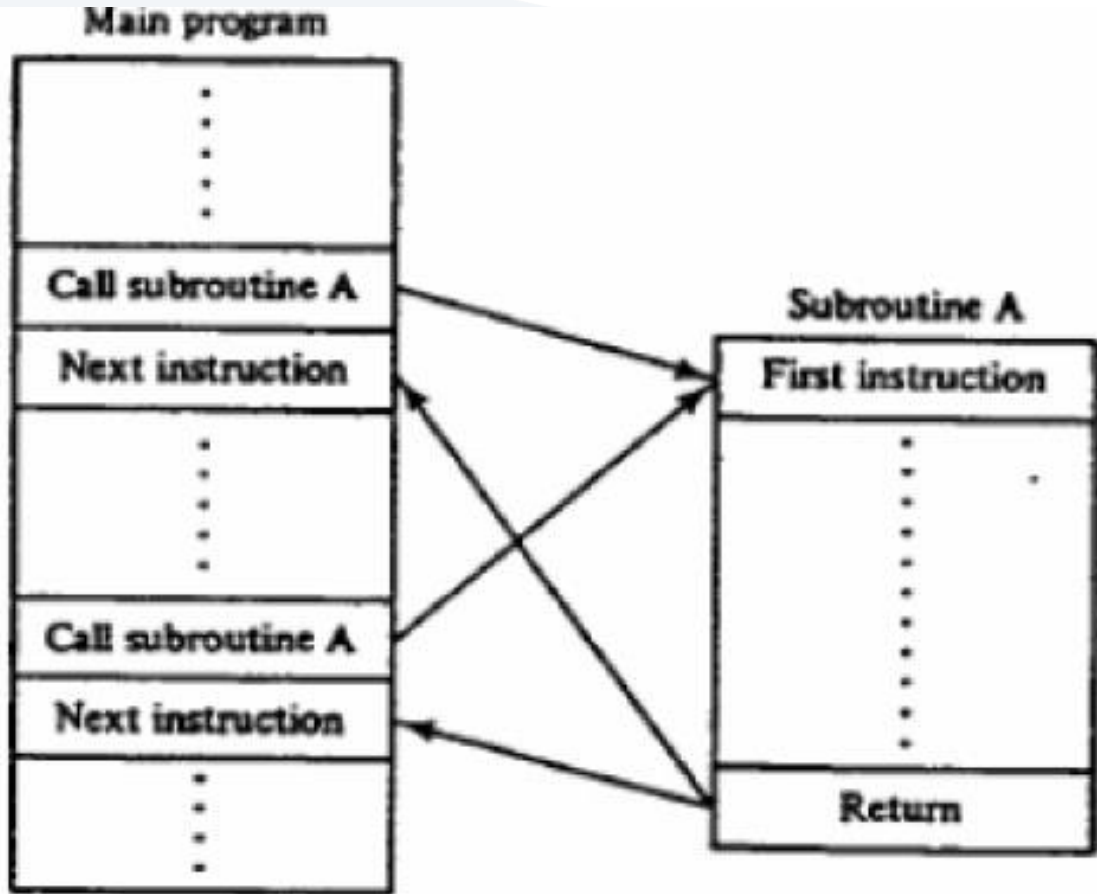
POP BX

POP AX

POPF



البرامج الفرعية



• إذا كان برنامجنا يُكرر تنفيذ مجموعة من التعليمات، مثل جمع قيم مصفوفة أو إيجاد القيمة العظمى بين مجموعة معطيات، فمن الأفضل وضع هذه التعليمات معاً ضمن وحدة تسمى **برنامجاً فرعياً** أو **إجرائية**

• يمكن طلب الإجرائية من أي نقطة من البرنامج الرئيسي باستخدام تعليمة **CALL** عندها يُخزن المعالج عنوان أول بايت من التعليمة التي تلي تعليمة CALL في المكس، ثم يَشحن IP بعنوان أول بايت من التعليمة الأولى في البرنامج الفرعي. يُنفذ المعالج تعليمات الإجرائية حتى يصل إلى تعليمة **RET** التي تُشير إلى نهاية الإجرائية. عندها يَشحن المعالج IP بالعنوان المخزن في المكس ويعود المعالج ليتابع تنفيذ التعليمات ابتداءً من التعليمة التي تلت تعليمة الاستدعاء CALL كما هو مبين في الشكل

- تعد وحدة الذاكرة جزءاً لا يتجزأ من أي نظام كمبيوتر والغرض الأساسي منها هو تخزين البرامج والبيانات.
- يمكن تقسيم نظام ذاكرة الكمبيوتر منطقياً إلى ثلاث مجموعات:
- ذاكرة المعالج Processor memory
- الذاكرة الأساسية أو الرئيسية Primary or main memory
- الذاكرة الثانوية Secondary memory

ذاكرة المعالج Processor memory

- تشير ذاكرة المعالج إلى المسجلات الموجودة داخل المعالج الصغير.
- تُستخدم هذه المسجلات للاحتفاظ بالبيانات والنتائج مؤقتًا عندما يكون الحسابات قيد الإجراء.
- نظرًا لأن مسجلات المعالج مصنوعة باستخدام نفس تقنية المعالج الصغير، فلا يوجد تفاوت في السرعة بين هذه المسجلات والمعالج.

الذاكرة الأساسية أو الرئيسية Primary or main memory

- تشير الذاكرة الأساسية أو الرئيسية إلى منطقة التخزين التي يمكن الوصول إليها مباشرة بواسطة المعالج الصغير.
- لذلك، يجب تخزين جميع البرامج والبيانات في الذاكرة الأساسية فقط قبل التنفيذ.
- في الذكريات الأساسية، يجب أن يكون وقت الوصول متوافقًا مع وقت القراءة / الكتابة للمعالج.
- لذلك، يتم استخدام ذواكر أنصاف النواقل فقط كذاكرة أساسية ويتم تصنيعها (أحدث الإصدارات) باستخدام تقنية CMOS
- تشمل الذاكرة الأساسية عادةً ROM و EPROM و DRAM و NVRAM

الذاكرة الثانوية Secondary memory

- تشير الذاكرة الثانوية إلى وسيلة التخزين التي تشتمل على أجهزة بطيئة مثل الأشرطة والأقراص المغناطيسية [القرص الصلب والقرص المرن والقرص المضغوط (CD)]
- تستخدم هذه الأجهزة لحفظ ملفات البيانات الكبيرة والبرامج الضخمة مثل أنظمة التشغيل (Operating Systems) والمترجمات (Compilers) وقواعد البيانات (Data Bases) والبرامج الدائمة وما إلى ذلك.
- يقوم نظام الحاسوب بنسخ البرامج والبيانات المطلوبة من الذاكرة الثانوية إلى الذاكرة الرئيسية ويعمل مباشرة مع الذاكرة الرئيسية فقط.