

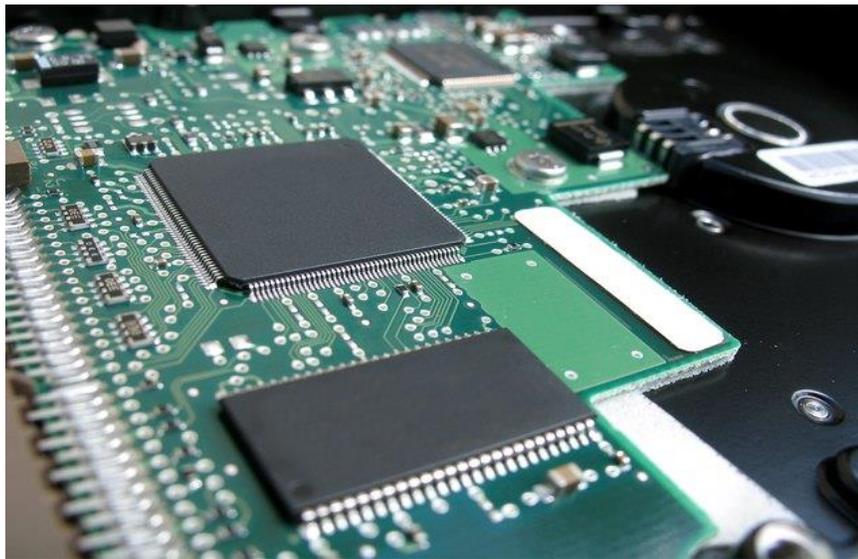


جامعة القادسية
كلية التربية



Lecture 17

Microprocessors



Prepared by:

Firas Abdulrahman Yosif

Stack Concept & applications

The stack segment (SS) is one of the important segments inside the main memory, as it is used to store return addresses when calling subroutines or interrupt services programs. Among the instructions that are used with the stack segments are the PUSH and the POP. Also, the SP register is used with the stack segment, where it refers to the top of the stack, which is represented by the address FFFF, because its size is 64KB. The contents of the SP register decrease by one when adding data to the stack every time the PUSH command is used, and the contents of the SP register increase by one each time when data is pulled. It is stacked via POP instruct.

يعتبر مقطع المكذسة stack segment (SS) من المقاطع المهمة داخل الذاكرة main memory حيث تستخدم في خزن عناوين الرجوع عند النداء على البرامج الفرعية (subroutines) او برامج خدمة المقاطعة (interrupt services) . ومن الابعازات التي تستخدم مع الـ stack segments هي ايعازات الـ PUSH والـ POP . يستخدم السجل SP مع مقطع المكذسة stack segment حيث يشير الى قمة المكذسة والتي تكون بالاسفل وتتمثل بالعنوان FFFF لان حجمها 64KB وتتناقص محتويات السجل SP بمقدار واحد كلما صعنا الى الاعلى عند اضافة بيانات الى المكذسة في كل مرة يستخدم اليعاز PUSH وتزداد محتويات السجل SP بمقدار واحد في كل مرة كلما نزلنا الى الاسفل عندما تسحب بيانات من المكذسة عن طريق اليعاز POP.

ومبدأ عمل الـ stack segment هو ان اخر قيمة تخزن داخل المكذسة (SS) هي اول قيمة تسحب وهذه الطريقة تسمى (First in First out) .

❖ Stack segment Instructions:

• **PUSH Instruction:** ايعاز الدفع

ايعاز الـ PUSH هو ايعاز يتعامل مع المكعدة حيث يدفع 2byte مرة واحدة الى داخل المكعدة stack .segment

The form: **PUSH Source**

Types of PUSH Instruction:

1. **PUSH reg. (16bit)**

Examples)

PUSH BX; ادفع بايتين من محتويات السجل الى داخل المكعدة

PUSH CX

PUSH SI

✓ ملاحظة: عند التعامل مع السجلات يجب ان تكون 2byte .

- PUSH AL FALSE (لايجوز استخدام سجل يتكون من بايت واحد)

Correct:

PUSH AX

✓ ملاحظة: لايجوز استخدام السجل IP مع ايعاز الـ PUSH وكذلك لايجوز استخدام البيانات بصورة مباشرة مع ايعاز الـ PUSH .

- PUSH 4030H False

Correction:

MOV CX, 4030H

PUSH CX

2. PUSH memory

عند التعامل مع الذاكرة ممكن استخدام العنونة المباشرة (direct addressing mod) او العنونة الغير مباشرة (indirect addressing mod).

Examples)

PUSH [SI]

PUSH [BP]

PUSH [2345H]

PUSH [4000H]

✓ ملاحظة: يكون التعامل مع موقعين من مواقع الذاكرة في كل عملية PUSH الى داخل المكعدة.

✓ ملاحظة: عند ادخال البيانات الى الـ stack segment باستخدام ايعاز الـ PUSH اولا يدخل الـ

high byte الى المكعدة ثم بعد ذلك يدخل الـ low byte .

✓ ملاحظة: يمكن استخدام ايعاز الـ PUSH مع مقاطع الذاكرة memory segmentation .

Examples)

PUSH CS

PUSH ES

• يمكن استخدام ايعاز الـ PUSH مع سجل الاعلام flag register :

Ex) PUSH F

المقصود بالايعاز اعلاه ادفع بمحتويات الـ Flag register والتي تتكون من 16 bit داخل المكعدة(SS).

Q1) What's the stack pointer address after execute these instructions?

MOV AX, 0AB3H

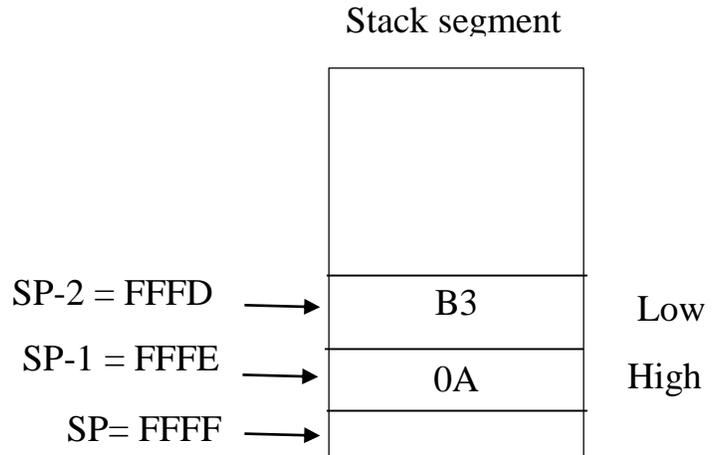
MOV SP, 0FFFFH

PUSH AX

HLT

Answer)

SP= 0FFFDH



• POP Instruction: ايعاز السحب

هذا اليعاز يسحب 2byte من المقطع stack segment ويتعامل مع 16 بت اي word .

ملاحظة: هذا اليعاز يسحب اولاً الـ low byte ثم بعد ذلك يسحب الـ high byte .

The form: **POP Source**

1. POP reg. (16 bit)

Examples)

- POP DI; سحب بايتين من داخل المكعدة ويضعها في السجل
- POP DX
- POP DS
- POP ES
- POP F; هذا اليعاز يمثل سحب بايتين من داخل المكعدة ويضعها في سجل الاعلام

2. POP memory:

- POP [SI]
- POP [BX]

في الابعاز اعلاه يتم سحب 2byte من داخل ال-stack ويضعها في موقع الذاكرة [BX] و [BX+1]
ملاحظة: لايجوز استخدام CS في ايعاز ال-POP .
ز هذه بعض الابعازات الخاطئة:

- POP CS; false
- POP 22H; false
- POP IP; false
- POP DH; false (لان تم استخدام سجل متكون من بايت واحد)

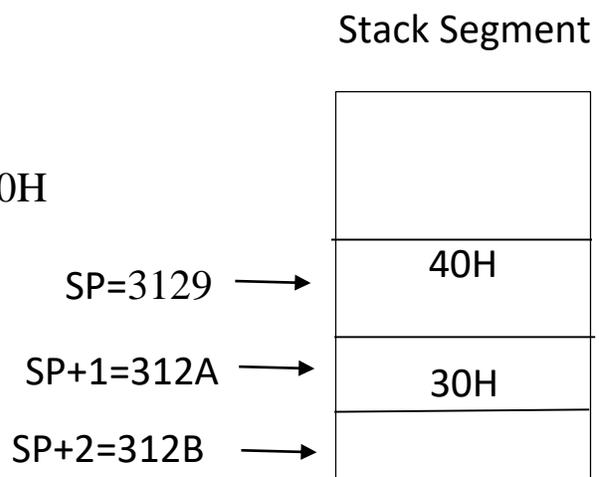
EX (1) What's address of SP after execute these instructions then find contain of BX?

MOV SP, 3129H

POP BX

HLT

Answer: SP=312B, BX=3040H



Ex(2) what's address of SP and contain of CX reg. after execute this program.

MOV SP, 0FFEEH

MOV AX, 1234H

PUSH AX

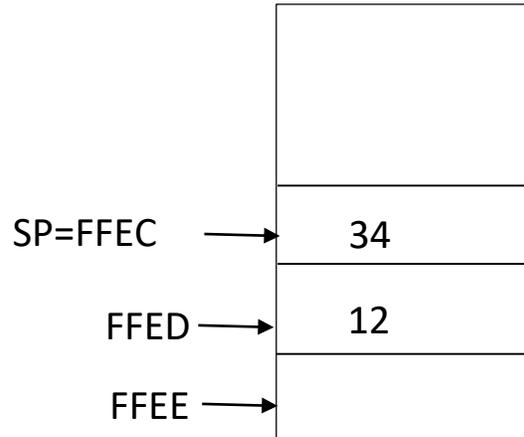
POP CX

HLT

Answer:

SP= 0FFEEH

CX= 1234H



• **XCHG Instruction:**

Format: XCHG dest., source

ملاحظة (1): لايجوز استخدام مقاطع الذاكرة في هذا اليعاز

ملاحظة (2): لايجوز استخدام البيانات بصورة مباشرة

Examples:

- XCHG DX, CX; يجب ان تكون احجام السجلات متساوية
- XCHG BL, CL
- XCHG [DI], BH

ملاحظة (3): ايعاز XCHG DX, CX يكافئ :

PUSH CX

PUSH DX

POP CX

POP DX

EX(3): Write a program to exchange contain between BX & CX , if BX= 02B1H, CX= 01FFH and SP= 0FFEEH. Using PUSH & POP instructions.

Answer:

MOV SP, 0FFEEH

MOV CX, 01FFH

MOV BX, 02B1H

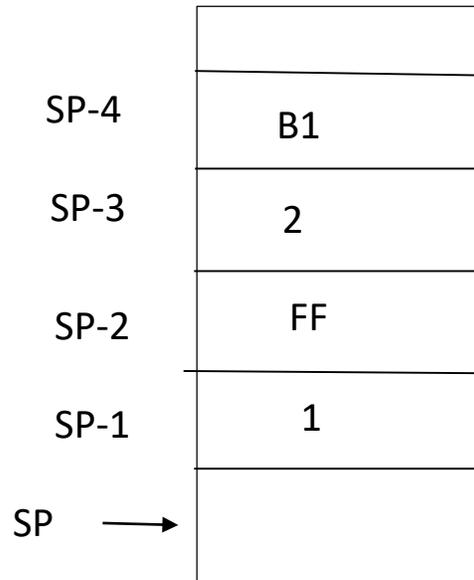
PUSH CX

PUSH BX

POP CX

POP BX

HLT



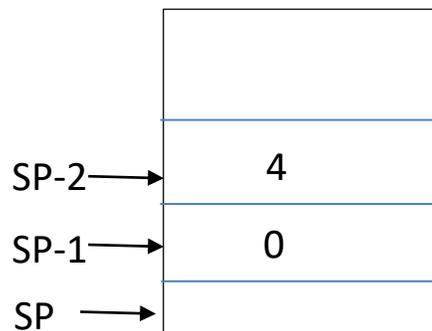
EX (4): Find contain of flag register after execution these instructions?

MOV CX, 04

PUSH CX

POPF

HLT



Solution)

0	0	0	0	NO	UP	DI	TF	PS	NZ	X	NA	X	PE	X	NC
---	---	---	---	----	----	----	----	----	----	---	----	---	----	---	----

EX (5): Find contain for this memory after execution this program.

MOV SP, 0FFEEH

MOV W.[BX], 3344H

PUSH [BX]

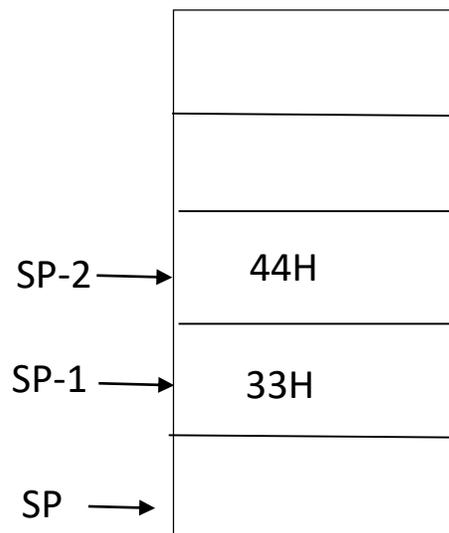
POP [BP]

HLT

Solution)

[BP]= 44H

[BP+1] =33H



Microprocessors

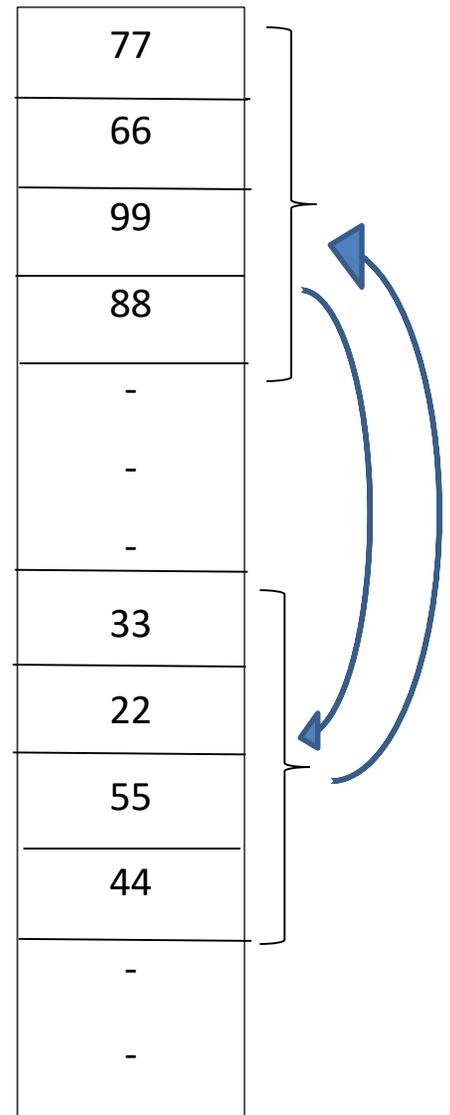
EX (6): Write a program to exchange the contents of memory locations start with (01f00H : 0001) and end with (01f00H : 0004) in DS, with contains memory location start with (01C00H : 2001H) in ES . Using PUSH&POP instructions.

Solution)

```
MOV AX, 01F00H
MOV DS, AX
MOV DI, 0001
MOV [DI], 6677H
MOV [DI+2], 8899H
MOV AX, 01C00H
MOV ES, AX
MOV SI, 2001H
ES: MOV [SI], 2233H
ES: MOV [SI+2], 4455H
MOV CL, 2
again: PUSH [DI]
ES: PUSH [SI]
POP [DI]
ES: POP [SI]
INC DI
INC DI
INC SI
```

DS: DI → DI=0001

ES: SI → SI= 2001H



Microprocessors

INC SI

DEC CL

JNZ again

HLT

The screenshot shows an emulator window titled "emulator: noname.bin_". The menu bar includes "file", "math", "debug", "view", "external", "virtual devices", "virtual drive", and "help". The toolbar contains "Load", "reload", "step back", "single step", "run", and a "step delay ms: 200" slider.

The registers window on the left shows the following values:

Register	H	L
AX	1C	00
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0038	
SS	0100	
SP	FFFE	
BP	0000	
SI	2009	
DI	0009	
DS	1F00	
ES	1C00	

The assembly code window shows the following instructions:

```
01031: 47 071 G PUSH w.[DI]
01032: 46 070 FT ES:
01033: 46 070 FT PUSH w.[SI]
01034: FE 254  POP w.[DI]
01035: C9 201  ES:
01036: 75 117  POP w.[SI]
01037: EE 238  INC DI
01038: F4 244  INC DI
01039: 90 144  INC SI
0103A: 90 144  INC SI
0103B: 90 144  DEC CL
0103C: 90 144  JNE 026h
0103D: 90 144  HLT
0103E: 90 144  NOP
0103F: 90 144  ...
```

The screenshot shows a "Random Access Memory" window with the address "1C00:2001" and an "update" button. The "table" radio button is selected. The memory dump shows:

1C00:2001	77	66	99	88	00	00	00	00	00-00	00	00	00	00	00
1C00:2011	00	00	00	00	00	00	00	00	00-00	00	00	00	00	00
1C00:2021	00	00	00	00	00	00	00	00	00-00	00	00	00	00	00

The screenshot shows another "Random Access Memory" window with the address "1F00:0001" and an "update" button. The "table" radio button is selected. The memory dump shows:

1F00:0001	33	22	55	44	00	00	00	00	00	00	00	00	00	00
1F00:0011	00	00	00	00	00	00	00	00	00	00	00	00	00	00
1F00:0021	00	00	00	00	00	00	00	00	00	00	00	00	00	00