

מבוא למחשבים

מצגת 2 – הוראות בסיסיות בשפת סף

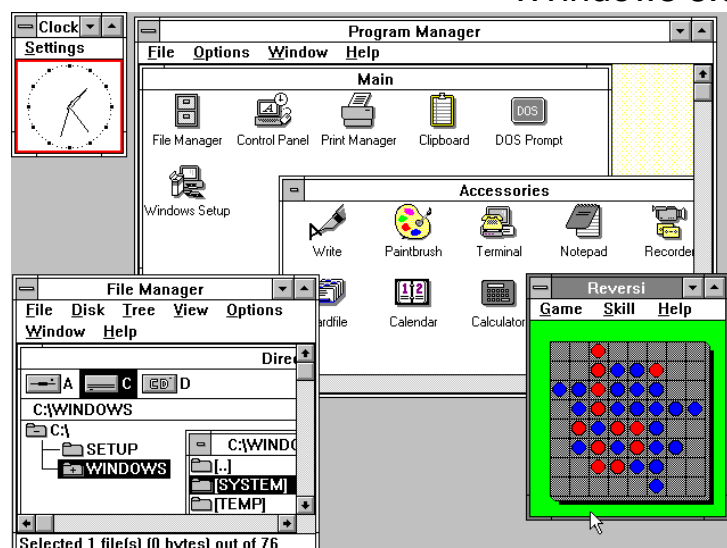
- מבוא
- אוגרי המעבד
- פקודת mov
- פקודות אריתמטיות

מבוא

- המעבד הוא לב המחשב והיחידה האחראית על עיבוד הנתונים והפקת המידע.
 - המעבד מורכב ממיליוני טרנזיסטורים שנמצאים על פיסת סיליקון זעירה.
 - הטרנזיסטורים הזעירים אוגרים בתוכם מתחים שונים המייצגים את השפה הבינארית שבה המחשב משתמש.
 - מתח מסוים בטרנזיסטור שקול לספרה הבינארית 1, ומתח אחר שקול לספרה הבינארית 0.
 - באמצעות קבוצות של אפסים ואחדים מצליח המעבד: ליצור ייצוגים של מספרים ולבצע עליהם פעולות מתמטיות או לוגיות.
- <https://www.youtube.com/watch?v=Fxv3JoS1uY8&t=1s>
- המעבד 8086 אותו ייצרה חברת אינטל בשנת 1978 היה מהצעדים הראשונים בפיתוח מעבדים למחשבים אישיים.



- Windows 3.0:



- חוק Moore (1965):
המעבדים מכפילים את מהירות העבודה שלהם כל 18 חודשים.

Table 1-2: Evolution of Intel's Microprocessors (from the 8086 to the Pentium Pro)

Product	8086	80286	80386	80486	Pentium	Pentium Pro
Year Introduced	1978	1982	1985	1989	1993	1995
Technology	NMOS	NMOS	CMOS	CMOS	BICMOS	BICMOS
Clock rate (MHz)	3–10	10–16	16–33	25–33	60, 66	150
Number of pins	40	68	132	168	273	387
Number of transistors	29,000	134,000	275,000	1.2 mill.	3.1 mill.	5.5 mill.
Physical memory	1M	16M	4G	4G	4G	64G
Virtual memory	None	1G	64T	64T	64T	64T
Internal data bus	16	16	32	32	32	32
External data bus	16	16	32	32	64	64
Address bus	20	24	32	32	32	36
Data types	8/16	8/16	8/16/32	8/16/32	8/16/32	8/16/32

סביבת עבודה

- MICROPROCESSOR EMULATOR - EMU8086 :
מדמה מעבדי 8086 ישנים, שהיו בשימוש במחשבי מקינטוש וחלונות משנות
ה-80 ותחילת שנות ה-90.

<https://www.malavida.com/en/soft/emu8086/download>

נכתב ע"י ניצן דולינסקי ©

אוגרי המעבד

- אוגרי המעבד Cpu registers הן יחידות זיכרון בתוך המעבד עצמו.
- כל פקודת מכונה מוגדרת ע"י פעולה ואופרנדים:

INSTRUCTION OP,OP

- אופרנד יכול להיות :
 1. אוגר
 2. קבוע (מספר הנשמר בתוך הקידוד של הפקודה עצמה).

אוגר AX

- ax מורכב מ-

AH	AL
----	----

- AL – 8 הביטים (בית) הנמוכים.
- AH – 8 הביטים (בית) הגבוהים.

אוגר BX

- bx מורכב מ-

BH	BL
----	----

- BL – 8 הביטים (בית) הנמוכים.
- BH – 8 הביטים (בית) הגבוהים.

פקודת mov

- mov - פעולת העתקה
- הפקודה ניתנת לביצוע באופן הבא: (מספר - immediate):

MOV register, register
MOV register, immediate

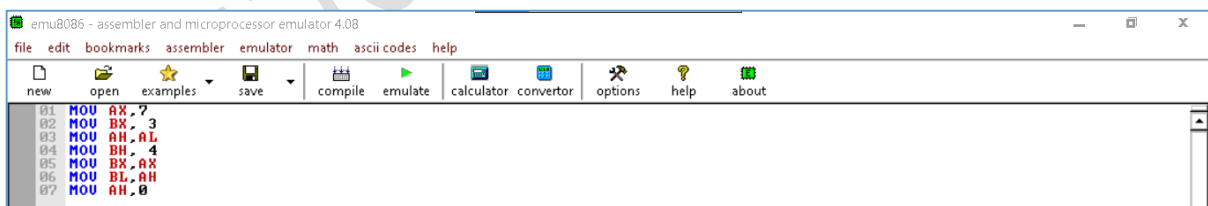
- פורמט פקודה: (src - מקור, dest - יעד)

MOV dest, src dest <= src

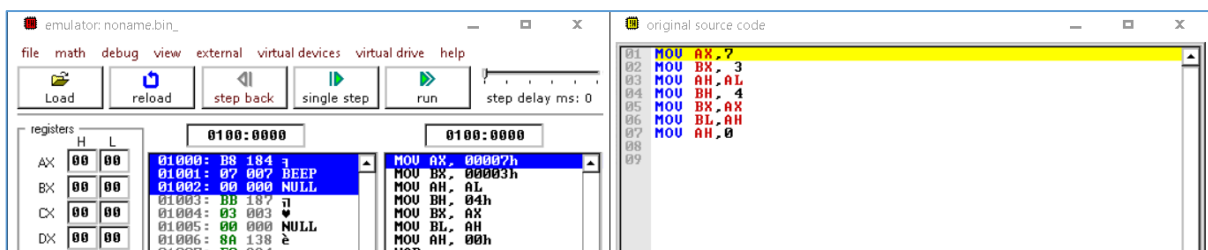
- דוגמה:

```
MOV AX,7
MOV BX, 3
MOV AH,AL
MOV BH, 4
MOV BX,AX
MOV BL,AH
MOV AH,0
```

- לחץ על emulate (בחלון עם אייקון ירוק):



- ארגן את 2 החלונות שנפתחו באופן הבא (אייקון אדום משמאל וצהוב מימין):



תרגיל 1

• כתוב קטע תוכנה:

1. העבר לאוגר AL את הערך 0
2. העבר ל-AH את הערך 1
3. העבר ל-BL את הערך 2
4. העבר ל-BH את הערך 3
5. העבר ל-BL את הערך 19
6. העבר לאוגר AL את הערך 12
7. העבר ל-AH את הערך 23
8. העבר ל-BL את הערך 44
9. העבר ל-BH את הערך 13
10. העבר ל-BX את הערך 40

נכתב ע"י ניצן דולינסקי ©

פקודות אריתמטיות

ADD, SUB פקודות

- פעולת חיבור
- הפקודה ניתנת לביצוע באופן הבא:

ADD/SUB register, register
ADD/SUB register, immediate

- פורמט פקודה:

ADD	dest, src	$\text{dest} \leq \text{dest} + \text{src}$
SUB	dest, src	$\text{dest} \leq \text{dest} - \text{src}$

- אם הפעולה מופעלת על אופרטור בן 8 ביט, התוצאה לא "גולשת" ל-AH.

- דוגמה:

```
MOV AX,2
ADD AX,3
MOV BX,5
ADD AX,BX
ADD BL,AL
ADD AL,4
ADD AH,5
ADD BH,7
ADD BL,8
SUB AX,6
SUB AL,4
SUB AH,3
MOV AL,254
ADD AI,1
ADD AI,1
ADD AI,1
```


תרגיל 2

- כתוב קטע תוכנה:
 1. העבר ל-AX את הערך 42
 2. הוסף ל-BX 24
 3. הוסף ל-AX 3
 4. הוסף ל-AX 15
 5. חסר מ-BL 1
 6. חסר מ-BX 1
 7. חסר BL מ-AL
 8. חסר מ-AX 4
 9. העבר לאוגר AX את הערך 13
 10. הוסף ל-AX את AX

פקודות inc, dec

- Inc - פעולת הוספת 1.
- Dec - פעולת חיסור 1.
- הפקודה ניתנת לביצוע באופן הבא:

INC/DEC register

- פורמט פקודה:

INC	dest	$dest \leq dest + 1$
DEC	dest	$dest \leq dest - 1$

- דוגמה:

MOV AX,12 INC AX INC AX DEC AX INC AL INC AH DEC AL DEC AL DEC AL MOV BX,17 DEC BL
--

תרגיל 3

- כתוב קטע תוכנה:
 1. העבר ל-BX את הערך 42
 2. הוסף ל-ax 1 (INC)
 3. חסר מ-ax 1 (DEC)
 4. חסר מ-ax 1
 5. חסר מ-al 32
 6. חסר מ-al 5
 7. העבר לאוגר AX את הערך 65
 8. חסר מ-al 17
 9. חסר מ-al 4

נכתב ע"י ניצן דולינסקי ©

פקודת MUL, DIV (8 ביט)

- הפקודות ניתנות לביצוע באופן הבא:

MUL/DIV register

- כפל 8 ביט, פורמט פקודה:

MUL	register	al <= al * register ah <= extend
-----	----------	-------------------------------------

- אם הפעולה מופעלת על אופרטור בן 8 ביט, התוצאה "גולשת" ל-AH.
- דוגמה:

MOV BX,2

MOV AX,2

MUL BL

MOV AX,5

MUL BL

MOV AX,127

MUL BL

MOV AX,128

MUL BL

MOV AX,129

MUL BL

• חילוק 8 ביט, פורמט פקודה:

DIV	register	$al \leq al / register$	(תוצאת חלוקה)
		$ah \leq ah \% register$	(תוצאת שארית)

- אם הפעולה מופעלת על אופרטור בן 8 ביטים (al), התוצאה "גולשת" ל-AH.
- דוגמה:

```
MOV BX,2
MOV AX,4
DIV BL

MOV AX,12
DIV BL

MOV AX,7
DIV BL

MOV BL,5
MOV AX,12
DIV BL
```

תרגיל 4

- כתוב קטע תוכנה (לאחר כל פקודה אפס את AH, BH):

1. העבר ל-AL 2
2. העבר ל-BL 3
3. כפול את AL ב-3
4. העבר ל-BL 4
5. כפול את AL ב-4
6. העבר ל-BL 5
7. כפול את AL ב-5
8. העבר לאוגר AX את הערך 40
9. העבר ל-BL 6
10. חלק את AL ב-6
11. העבר ל-BL 3
12. חלק את AL ב-3
13. העבר ל-BL 2
14. חלק את AL ב-2

- בכל שלב יש לציין בטבלה מהם ערכי הרגיסטרים

AH	AL	BH	BL	
				1
				2
				3
				...