

1. מה זה ארכיטקטורת מחשב?
- תכנון המבנה, התפקידו והאינטראקטיה בין רכיבי המחשב, כולל המעבד, הזיכרון, והתקני קלט/פלט.
2. ארכיטקטורת Von Neumann לעומת Harvard - יתרונות וחסרונות:
- Von Neumann: זיכרון משותף לתוכנה ולנתונים → גמישות גבוהה אך צואර בזיכרון כאשר לא זיכרון.
  - Harvard: זיכרון נפרד לתוכנה ולנתונים → ביצועים מהירים יותר עקב גישה מקילית, אך מורכב יותר לישום.
3. באיזה ארכיטקטורה בנוי 8086?
- Von Neumann
4. ייעוד של זיכרון ראשי (RAM):
- אחסון זמני של נתונים ותוכיות בזיכרון ריבוי כדי לאפשר גישה מהירה למעבד.
5. מה זה BIOS? על איזה סוג של זיכרון חוא נshrmo?
- מערכת בסיסית לאתחול המחשב ולבדיקות חומרה. נשמר בזיכרון ROM או Flash Memory.
6. תפקודים של פס כתובות, פס נתוני וקווי בקרה:
- פס נתוני: קובע כתובות זיכרון או בתקין קלט/פלט.
  - פס נתוני: מעביר נתונים בין המעבד לרכיבים אחרים.
  - קווי בקרה: שלוטים בפעולות המעבד וכוחם הערכת נתונים.
7. למה מתקווים הרגליים של 8086 בפרש ובכיב אלקטוני בכלל?
- הפינים שמאפשרים תקשורת בין הרכיב לרכיבים אחרים.
8. מה הרכיבים העיקריים של 8086?
- Prefetch
- יחידת בקרה, יחידת ALU, רגיסטרים, זיכרון טטומי, פסי נתונים ותוכיות, יחידת קדם-שליפה (Prefetch Unit).
9. מה זו Multiplexed בהקשר של 8086?
- שילוב של פס כתובות ונתונים על אותו פין להיסכון בכתובת החיבורם.
10. מה אפשר פס נתוני של ?Byte, Word, Double Word
- גישה ליחידה מידע בגודל 8 ביטים (Byte), 16 ביטים (Word), או 32 ביטים (Double Word).
11. מה אפשר פס כתובות של ?Byte, Word, Double Word
- קובע את כמות הזיכרון הנגיש; ב-8086 פס הכתובות הוא 20 ביטים ומאפשר גישה עד 1MB.
12. למה מחשב צריך CLOCK?
- לזמן פעולות המעבד, קביעת מהירות העיבוד וונכון הרכיבים.
13. מה תפקודים של אוגרים במעבד?
- אחסון זמני של נתונים, כתובות, תוכאות חישובים וניהול זרימת התוכנית.
14. מה זה שפת אסמביל? מה החבדל בין אסמביל ל-C או C-#?
- שפת תכנות ונווכה עם פקודות ישירות למעבד. אסמביל מתרגם קוד אסמביל לשפת מוכנה, בעוד ש-C-# הן שפות עליות המאפשרות תוכנות מותקן יותר.
15. אילו סוג רגיסטרים יש ב-8086? מה תפקודם?
- רגיסטרים כלליים (AX, BX, CX, DX): משמשים לאחסון נתונים וחישובים.
  - רגיסטרים מוחדים (FLAGS): מנחים את זרימת התוכנית ומאחסנים דגלי מצב.
  - רגיסטרים סגמנטיים (CS, DS, SS, ES): משמשים ללחוקת הזיכרון.
  - רגיסטרים של המחכנת (BP, SP): משמשים לניהול קראות פונקציות.
16. מה זו מהסנית בתוכנית? למה היא משמשת? אך היא עבדת?
- מכנה נתונים מסווג LIFO (האחרון שנכנס – הראשון שיצא), משמש לניהול קראות פונקציות, אחסון פרמטרים זמינים ושמירת כתובות זדרה, פועל עם PUSH (הכנסת) ו-POP (שליפה).

base 16 ↳ ↲

base 10 - ↳

$\frac{205}{41} \quad | \quad 5$

X Y Z Base 8  
3218

$$X \cdot 8 + Y = \square$$

$$\square \cdot 8 + Z = \text{Answer}$$

|     |   |
|-----|---|
| 205 | 5 |
| 41  | 0 |
| 8   | 1 |
| 1   | 3 |
| 0   | 1 |

→

5. (נק') בצעו המרה של מספר בינארי 11001101

$$\begin{array}{l}
 \text{Base 2} = 11001101 \\
 \text{Base 4} = \\
 \text{Base 8} = \\
 \text{Base 16} = \\
 \text{Base 10} = \text{205} \\
 \text{Base 3} = \\
 \text{Base 5} = \text{1310}
 \end{array}$$

|     |   |
|-----|---|
| 105 | 2 |
| 52  | 1 |
| 26  | 0 |
| 13  | 0 |
| 6   | 1 |
| 3   | 0 |
| 1   | 1 |
| 0   | 1 |

01101001

10010111

A ↴ up  
to 8bit!  
flip  
then +1

7. (נק') מצאו סכום ומכפלה של זוג מספרים חיוביים בינריים.

האיך  
205 \* 101 = ?

$$\begin{array}{r}
 10111 * 10011 = \\
 \hline
 \begin{array}{r}
 10111 \quad | \quad 1 \quad 1 \\
 101110 \quad | \quad 1 \quad \swarrow 10 \\
 -1011100 \quad | \quad 0 \quad \swarrow 100 \\
 -10111000 \quad | \quad 0 \quad \swarrow 1000 \\
 \hline
 101110000 \quad | \quad 1 \quad \swarrow 10000 \\
 \hline
 110110101
 \end{array}
 \end{array}$$

8. (נק') ביצעו חילוק של מספרים בינריים. התוצאה ומה השארית?

a.  $11001 / 101 = 101$        $11001 \% 101 = 0$

(≥)

11001 ÷ 101 / 101

$$\begin{array}{r}
 \overline{11001 \div 101} / 101 \\
 - \quad \underline{\underline{101}} \quad \downarrow \quad \downarrow \\
 \underline{101} \\
 - \quad \underline{101} \\
 \underline{0}
 \end{array}$$

השלמת חומרה:

Syntax for a variable declaration:

name DB value

name DW value

**DB** - stays for Define Byte.

**DW** - stays for Define Word.

name - can be any letter or digit combination, though it should start with a letter. It's possible to declare unnamed variables by not specifying the name (this variable will have an address but no name).

value - can be any numeric value in any supported numbering system (hexadecimal, binary, or decimal), or "?" symbol for variables that are not initialized.

נתון קידוד הבא:

```
1. msg: DB "ABBA abba" ; store string
2. OPR_BYTE: DB 16
3. RESULT1: DW 0 ; declare place to store result
4. RESULT2: DW 0
5. OPR1: DW 0x0102 ; declare first number
6. OPR2: DW 0x090A ; declare second number
7. start:
8. MOV AX, word OPR1 ; move first number to AX
9. MOV BX, word OPR2 ; move second number to BX
10. CLC ; clear the carry flag
11. ADD AX, BX ; add BX to AX
12. MOV DI, OFFSET RESULT1 ; move offset of result1 to DI
13. MOV word [DI], AX ; store result1
14. SUB AX, BX ; sub BX from AX
15. INC AX ; add 1 to AX
16. DEC BX ; dec 1 from BX
17. SUB BX, AX ; sub ax from bx
18. MOV DI, OFFSET RESULT2 ; move offset of result2 to DI
19. MOV word [DI], BX ; store result2
20. print reg ; print reg
```

$\begin{array}{r} 0102 \\ + 090A \\ \hline 0A0C \end{array}$

↓

AX



9. (נק') מה תהיה פרישת הזיכרון לאחר קימפול של התוכנית?

(טבלת ה-ASCII נמצאת בסוף סוף המבחן)

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| 41 | 42 | 42 | 41 | 20 | 61 | 62 | 62 | 61 | 70 | 0  | 0  | 0  | 0  | 02 | 01 |

|    |    |    |    |    |    |    |    |    |    |     |
|----|----|----|----|----|----|----|----|----|----|-----|
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | ... |
| UA | 09 |    |    |    |    |    |    |    |    |     |

~~0A0C~~

~~-090A~~

~~10102 AX~~

~~+  
1  
0103 AX~~

~~0909 BX~~

~~0909~~

~~-0103~~

~~0806~~

א' (נק') מה יהיה תוכן הרגיסטרים אחרי ביצוע של שורה 11 ?

AH = ~~0A~~ | AL = ~~0C~~

BH = ~~09~~ | BL = ~~0A~~

ב' (נק') מה יהיה תוכן הרגיסטרים אחרי ביצוע של שורה 17 ?

AH = ~~01~~ | AL = ~~03~~

BH = ~~08~~ | BL = ~~06~~

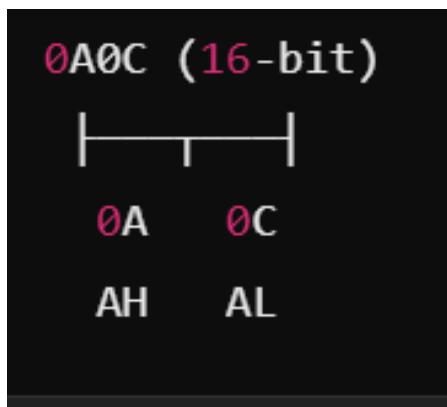
(נק') מה תהיה פרישת הזיכרון לאחר סיום של התוכנית?

|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| 41 | 42 | 42 | 41 | 20 | 61 | 62 | 62 | 61 | 70 | 0C | 0A | 06 | 08 | 02 | 01 |

---

|    |    |    |    |    |    |    |    |    |    |     |
|----|----|----|----|----|----|----|----|----|----|-----|
| 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | ... |
| 0A | 09 |    |    |    |    |    |    |    |    |     |

---

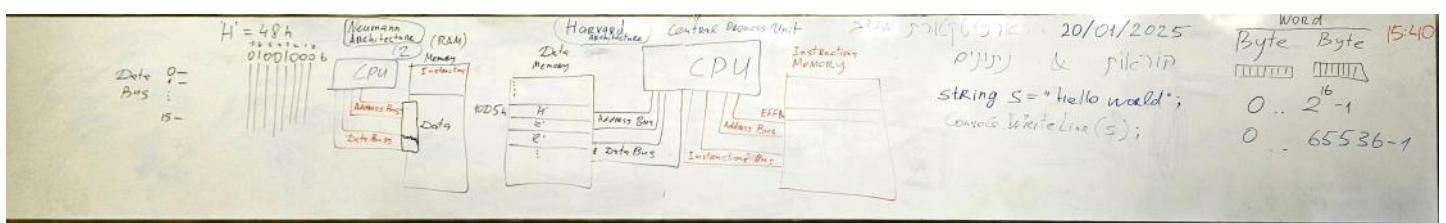


רְגִיסֶּטֶר 16 בָּיִט AL

רְגִיסֶּטֶר 8 בָּיִט AL

רְגִיסֶּטֶר 8 בָּיִט AH

רְגִיסֶּטֶר AH



11. (נק') כתבו תוכנית אסמבילר שתדפיס את השם שלכם יחד עם הגיל.  
שיםו לב לאורך המחרוזת שציריך להדפס.

לדוגמא:

My name is John Blues

I am 25 years old

```
1 name: DB "ido kahlon" ; store string
2 age: DB "age: 24" ;
3
4 start:
5 MOV AH, 0x13           ; move BIOS interrupt number in AH
6 MOV CX, 10             ; move length of string in cx
7 MOV BX, 0               ; mov 0 to bx, so we can move it to es
8 MOV ES, BX              ; move segment start of string to es, 0
9 MOV BP, OFFSET name    ; move start offset of string in bp
10 MOV DL, 0               ; start writing from col 0
11 int 0x10                ; BIOS interrupt
12
13 MOV CX, 7
14 MOV BP, OFFSET age
15 int 0x10
```