# מודל מופשט של פעולת המעבד

## המחשב

* המחשב הוא מערכת מורכבת המיועדת למשתמשים רבים, שלכל אחד מהם צרכים מגוונים. המשתמש ביישומי מחשב כמו משחקים, מעבד תמלילים, תוכנה לניהול משרד ועוד מתקשר עם המחשב בעזרת ממשק המשתמש הכולל בדרך כלל תפריטים.
* למחשב גם מנגנונים אחרים התומכים בתכניות המחשב ומאפשרים הרצתם, למשל מערכת ההפעלה.

המחשב בנוי מתכנה וחומה. החומרה זה כל החלקים הקשיחים שאנו יכולים לחוש. התכנה זה אוסף הוראות בשפת תכנות כל שהיא שנועדה לבצע משימה במחשב.

**התוכנה** היא זו המפעילה את החומרה ומאפשרת לכם לעשות דברים.

חשבו על החומרה כאילו היא מכשיר הטלוויזיה שלכם, ואילו התוכניות שבהן אתם צופים הן התוכנה.

**תוכנת המחשב** היא אוסף מאורגן של תכניות מחשב .באמצעות התוכנה המחשב מספק את שירותיו למשתמשים בו.

**תכנית מחשב** היא רצף של הוראות בשפת תכנות כלשהי שנועדו לבצע משימה מסוימת במחשב. מבחינה מהותית יותר, תכנית מחשב היא מימוש של אלגוריתם באמצעות שפת תכנות. הפקודות המפעילות בסופו של דבר, את המחשב נקראות שפת מכונה, פקודות אלה הן פקודות אלמנטריות ביותר, כשכל פקודה בנויה מרצף קצר של ביטים שמשמעותו מובנת על ידי המעבד. כתיבה ישירה של רצף הביטים שיבצע תכנית משמעותית היא פעולה מייגעת. כדי לקצר תהליך זה נוצרו שפות התכנות

**שפת התכנות** במהותה נועדה לגשר בין המחשב, ששפתו, הקרויה שפת מכונה, מורכבת מרצפי סיביות, לבין המתכנת שמעדיף לתקשר בצורה מילולית יותר. לכל שפת תכנות, בדומה לשפה טבעית, יש אוצר מילים משלה וכללי תחביר ייחודים, אך המרחק בין השתיים רב.

## https://upload.wikimedia.org/wikipedia/commons/thumb/6/6e/Von_Neumann_architecture_he.svg/497px-Von_Neumann_architecture_he.svg.pngארכיטקטורת פון ניומן

ארכיטקטורת פון נוימן או לעתים גם מכונת פון נוימן היא מודל שהציע בשנות הארבעים המתמטיקאי ג'ון פון נוימן למבנה המחשב.

במודל זה 2 חלקים עיקריים

זיכרון - המחשב משמש הן לאחסון התוכנית שהמחשב מבצע והן לאחסון הנתונים שתוכנית זו קוראת או כותבת.

בזיכרון מאוחסנים גם הפקודות לביצוע וגם הנתונים (אפשר לתת דוגמה של תכנת word)

מעבד - יחידה אריתמטית לוגית שתפקידה לבצע את פקודות התכנית.

גם בימינו מבוססים המחשבים על מודל זה, כאשר היחידה האריתמטית לוגית ויחידת הבקרה מאוחדים לכדי יחידת העיבוד המרכזית.

בהתאם לארכיטקטורה של פון ניומן מערכת מחשב כוללת ארבעה מרכיבים עיקריים

* המעבד (Processor) שתפקידו לבצע את פקודות התכנית.
* יחידת הזיכרון (Memory) בה מאוחסנים הנתונים וההוראות לביצוע.
* ערוצי תקשורת הנקראים "אפיקים" (buses) אשר מקשרים ומעבירים מידע בין הזיכרון למעבד.
* התקני קלט פלט.  
  התקני פלט הם התקנים בהם ניתן לראות את תוצאות ההרצה למשל מסך ומדפסת

התקני קלט הם התקנים איתם ניתן להטעין מידע למחשב למשל מקלדת, סורק ו - joy stick .

## זיכרון המחשב

במחשב יחידת זיכרון הבנויה כמערך.

בכל תא ניתן לאחסן מספר.

מספר זה יכול להיות נתון, הוראה או כתובת של תא בזיכרון.

המחשב פונה אל לתא מסוים כדי לבצע אחת משתי הפעולות:

קריאה של נתון או הוראה

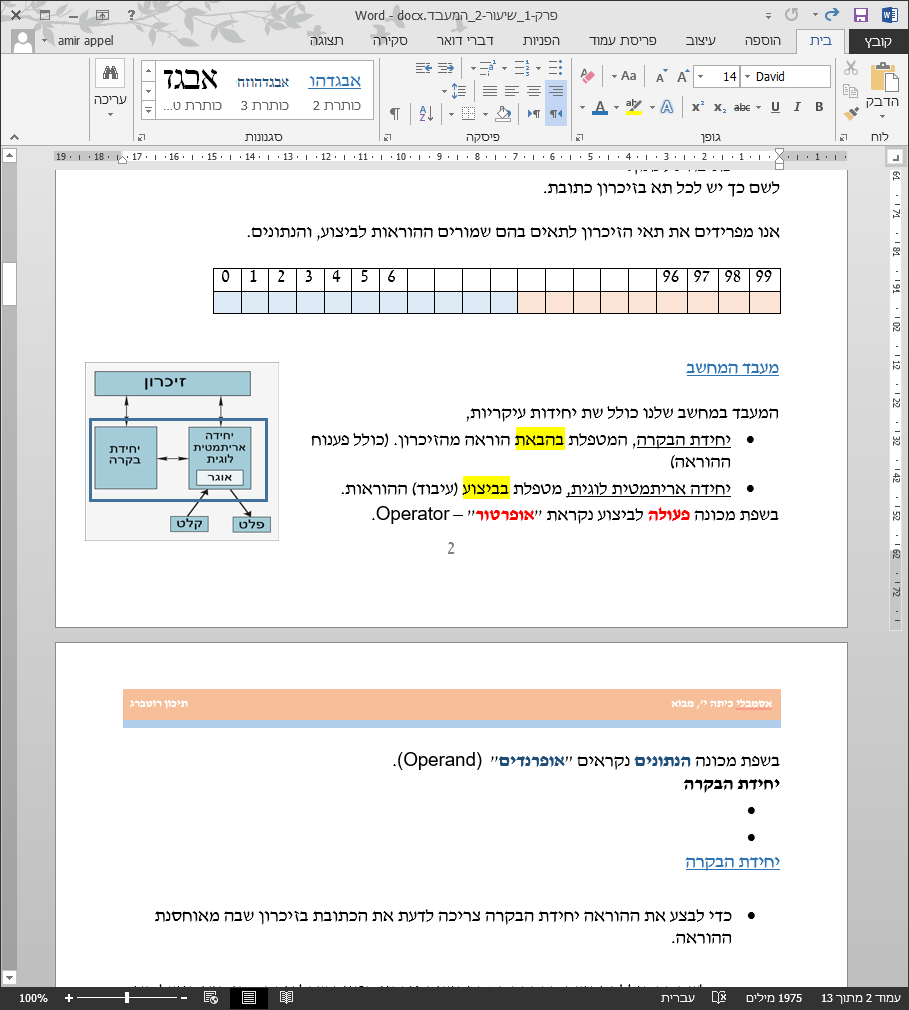
כתיבה של נתון.

לשם כך יש לכל תא בזיכרון כתובת.

אנו מפרידים את תאי הזיכרון לתאים בהם שמורים ההוראות לביצוע, והנתונים.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 99 | 98 | 97 | 96 |  |  |  |  |  |  |  |  |  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## מעבד המחשב



המעבד במחשב שלנו כולל שת יחידות עיקריות,

* יחידת הבקרה, המטפלת בהבאת הוראה מהזיכרון. (כולל פענוח ההוראה)
* יחידה אריתמטית לוגית, מטפלת בביצוע (עיבוד) ההוראות.

בשפת מכונה **פעולה** לביצוע נקראת "**אופרטור**" – Operator.

בשפת מכונה **הנתונים** נקראים "**אופרנדים**" (Operand).

## יחידת הבקרה

* כדי לבצע את ההוראה יחידת הבקרה צריכה לדעת את הכתובת בזיכרון שבה מאוחסנת ההוראה.

הרגיסטר הוא תא זיכרון הנמצא בתוך המעבד ובמקום לפנות לזיכרון המחשב כדי למשוך ממנו נתונים לביצוע חישובים שומרים את הנתונים בתוך תאי הזיכרון של המעבד וכך יותר קל למעבד להגיע למידע.

* לשם כך כוללת יחידת הבקרה כמה יחידות זיכרון קטנות שכל אחת מהן היא בגודל תא אחד היכול להכיל מספר בן 8 ספרות.
* כל יחידת זיכרון כזו נקראת **אוגר (register)**.

## יחידת הביצוע

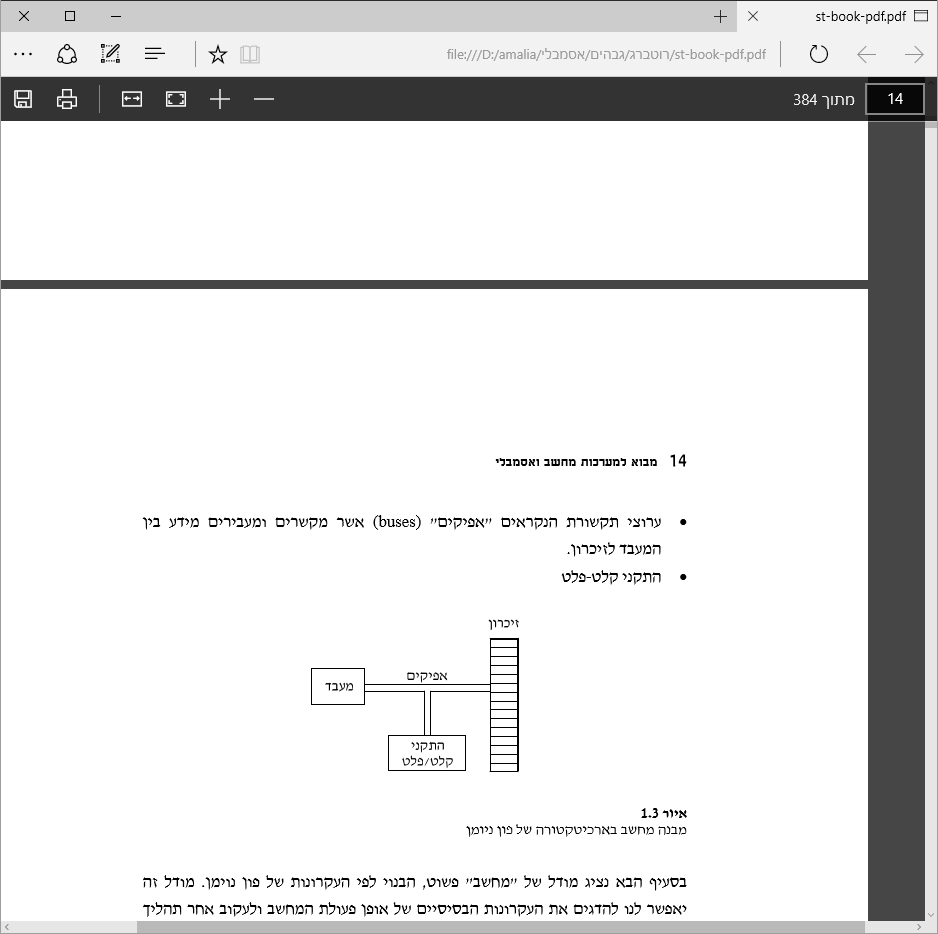
יחידה אריתמטית לוגית (ALU - Arithmetic Logic Unit), תפקידה לבצע את הפעולות המוגדרות בהוראה, כגון העתקת נתונים ממקום למקום, פעולות אריתמטיות ופעולות לוגיות.

מעבד:

יחידה אריתמטית לוגית (ALU - Arithmetic Logic Unit), תפקידה לבצע את הפעולות המוגדרות בהוראה, כגון העתקת נתונים ממקום למקום, פעולות אריתמטיות ופעולות לוגיות.

אוגרים. באוגרים אלה מאוחסנים הנתונים בהם משתמשת היחידה האריתמטית לוגית. *המעבד אינו מסוגל לעשות פעולות רבות בו זמנית. הוא מבצע פעולה אחת בזמן נתון. חוזקו הוא במהירות החישוב שלו.*

גם אנחנו בבואנו לבצע פעולת חיבור אנו מפרקים אותה בראשנן:

אם נפרק את הפעלה שאנחנו עושים בחיבור מספרים:

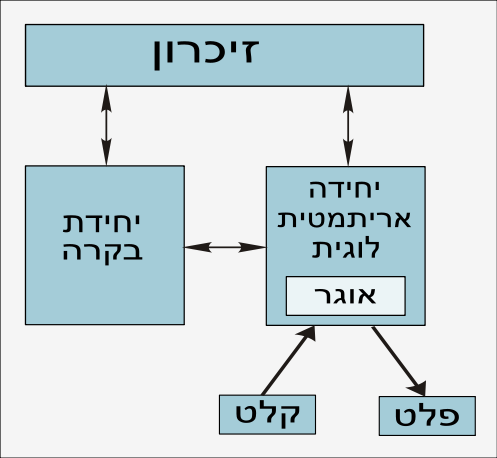
חושבים על מספר ראשון,   
לאחר מכן על מספר שני   
ואחר כך מחברים את שינם יחד.

כך המעבד:

מביא מספר ראשון(לרגיסטר אחד שבמעבד מהזיכרון)   
מביא מספר שני (לרגיסטר שני שבמעבד מהזיכרון) ומחבר את שניהים.

## הבנת פעולת המעבד

נגדיר משחב מופשט



* למחשב שלנו:
  + זיכרון
  + מעבד

המבצע את הפעולות הבאות:

* חיבור
* השמה (העתקה)

## זיכרון המחשב

למחשב שלנו 100 תאי זכרון.

* נאחסן את ההוראות בכתובות עליוניות של הזיכרון (למשל 49 - 0)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 99 | 98 | 97 | 96 |  |  |  |  |  |  |  |  |  | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

* נאחסן את הנתונים בכתובות הנמוכות של הזיכרון (למשל 99 - 50)

## יחידת הביצוע

* **יחידה אריתמטית לוגית** (ALU - Arithmetic Logic Unit), תפקידה לבצע את הפעולות המוגדרות בהוראה, כגון העתקת נתונים ממקום למקום, פעולות אריתמטיות ופעולות לוגיות.
* **שני אוגרים**, AX ו – BX. באוגרים אלה מאוחסנים הנתונים בהם משתמשת היחידה האריתמטית לוגית.

## כתיבת תכנית המחברת שני מספרים

* המחשב שלנו יכול לבצע שלושה סוגים של הוראות:
  + 1. **הוראת העברה - MOV**
    2. **פעולת חיבור ADD -**
    3. **והוראת סיום - QUIT**

## הוראת העברה

* **הוראת העברה mov** – מעתיקה תוכן של אופרנד המקור אל אופרנד היעד.   
  העתקה של נתון מתא אחד לתא אחר.  
  **אופרנד המקור** יכול להיות מספר קבוע, אוגר AX, אוגר BX או תא בזיכרון.  
  **אופרנד היעד** חייב להיות אחד משי האוגרים AX, BX או תא בזיכרון.
* mov AX, 5
* mov אופרנד יעד, אופרנד מקור
* פקודת mov, תעתיק את הערך 5 לתוך רגיסטר AX

## הוראת חיבור

**הוראת חיבור add** – כוללת שני אופרנדים

(שני נתונים מקור ויעד. את המקור מעתיקים ליעד)  
אופרנד מקור + אופרנד יעד ⇦ אופרנד יעד.

במחשב שלנו בהוראת החיבור 2 האופרנדים הם רגיסטרים

* לדוגמא אם ברגיסטר Ax מאוחסן הערך 3,   
  וברגיסטר Bxהערך 4:

Add Ax, Bx ⇦ Ax = 7

הוראת סיום – במחשב שלנו **Quit** מסיים את ריצת התוכנית.

## נסו בעצמכם

במעבד שלנו קיימם:

* + - זיכרון של 100 תאים
    - שלושה רגיסטרים ax, bx, ip

קיימות שלוש פעולות: mov, add, quit

הרצת התכנית בפייתון והתנסות של התלמידים.

להריץ מתוך הדרייב את Von\_Neumann.py

רגיסטרים: ax, bx, ip.

פקודות חוקיות: mov ax 5, mov [3] 7, mov[3] ax, mov bx ax, mov [5] bx

add ax 3

# תכנית לחיבור מספרים

* נגדיר שלושה שתנים בזיכרון בתאים [0] , [1] ו – [2].
* שימו את הנתון 23 במשתנה הראשון [0] ⇦ mov [0], 23  
  (תא זיכרון שכתובתו 0)
* שימו את הנתון 54 במשתנה [1] ⇦ mov [1], 54  
  (תא זיכרון שכתובתו 1)
* חשבו [0] + [1] ושימו את התוצאה בתא זיכרון – [2]
* שימו לב 🎔, בגלל מבנה המחשב לא ניתן לחבר שני ערכים השמורים בתי זיכרון שונים, לכן נעביר את הערכים לרגיסטרים, נחבר את הערכים ונעביר ת התוצאה לתא זיכרון חדש.

לתת לתלמידים לנסות לכתוב את הפעולות בעצמם ולבדוק כמה על הלוח.

דף תרגילים עמוד 1

## תכנית לחיבור מספרים

* נשתמש בהוראת **העברה** כדי להשים נתונים למשתנים.
* מאחר ולא קיימת הוראה שבה ניתן לחבר אופרנדים מתא בזיכרון, נעתיק את הנתונים מהמשתנים לאוגרים ax ו – bx.
* נחבר ונשמור את התוצאה באוגר AX.
* לסיום נעתיק את התוצאה מאוגר AX למשתנה חדש בזיכרון [5].
* בתא שכתובתו **0** נאחסן את **הערך 23**ובתא שכתובתו **1** נאחסן את **הערך 54**.

**התוכנית באסמבלי:**

|  |  |
| --- | --- |
| **נתונים**:    ----------------  **קוד:**   1. השם לתא זיכרון 0 את הערך 23 2. השם לתא זיכרון 1 את הערך 54 3. שים את תוכן תא [0] באוגר AX 4. שים את תוכן תא [1] באוגר BX 5. חבר את AX ו – BX ושים את התוצאה באוגר AX 6. שים את תוכן אוגר AX בתא זיכרון [2]. | DATASEG  ;---------------------  CODESEG  start:  mov [0], 23  mov [1], 54  mov ax [0]  mov bx, [1]  add ax, bx  mov [2], ax |

מעקב אחר ביצוע ההוראה.

|  |  |  |
| --- | --- | --- |
| רגיסטרים:  AX -  BX - | 1. השם לתא זיכרון [0] את הערך 23 | קוד אסמבלי  mov [0], 23 |
|  | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 23 |  |  |  |  |  |  |  |  |  |  | | זיכרון |
|  |  |  |
| רגיסטרים:  AX -  BX - | 1. השם לאת זיכרון [1] את הערך 54 | קוד אסמבלי  mov [1], 54 |
|  | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 23 | 54 |  |  |  |  |  |  |  |  |  | | זיכרון |
|  |  |  |
| רגיסטרים:  AX - 23  BX - | 1. שים את תוכן [0] באוגר AX | קוד אסמבלי  mov ax [0] |
|  | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 23 | 54 |  |  |  |  |  |  |  |  |  | | זיכרון |

|  |  |  |
| --- | --- | --- |
| רגיסטרים:  AX - 23  BX -54 | 1. שים את תוכן [1] באוגר BX | קוד אסמבלי  mov bx [1] |
|  | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 23 | 54 |  |  |  |  |  |  |  |  |  | | זיכרון |

|  |  |  |
| --- | --- | --- |
| רגיסטרים:  AX - 77  BX -54 | 1. חבר את AX ו – BX ושים את התוצאה באוגר AX | קוד אסמסלי  add ax, bx |
|  | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 23 | 54 |  |  |  |  |  |  |  |  |  | | זיכרון |

|  |  |  |
| --- | --- | --- |
| רגיסטרים:  AX - 77  BX -54 | 1. שים את תוכן אוגר AX בתא זיכרון [2]. | קוד באסמבלי  mov [2], ax |
|  | |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | 23 | 54 | 77 |  |  |  |  |  |  |  |  | | זיכרון |

## שאלות

כתבו בשפת מכונה או בשפה עברית המדמה שפת מכונה את התכניות הבאות בעזרת הוראות העתקה והוראות חיבור:

**שאלה 1:**

שימו את הערך 34 בכתובת 60 בזיכרון  
שימו 15 בכתובת 65 בזיכרון המחשב  
הגדילו את ערכו של [60] ב – 1

חברו את [60] ו – [65] ושימו את התוצאה ב – [65].

שימו לב 🎔, לא ניתן לחבר מספרים משני תאי זיכון חייבים להעביר לרגיסטרים את הערכים.

**שאלה 2:**

שימו 15 ברגיסטר Ax  
שימו 32 בכתובת 55, בזיכרון המחשב  
חשבו Ax + [55] + 20 ושימו את התוצאה ב – Bx.

שימו לב 🎔, ניתן לחבר רק שני מספרים בכל פעם.

## תשובות

**שאלה 1:**

שימו את הערך 34 בכתובת 60 בזיכרון  
שימו 15 בכתובת 65 בזיכרון המחשב  
הגדילו את ערכו של [60] ב – 1

חברו את [60] ו – [65] ושימו את התוצאה ב – [65].

שימו לב 🎔, לא ניתן לחבר מספרים משני תאי זיכון חייבים להעביר לרגיסטרים את הערכים.

|  |  |
| --- | --- |
| **נתונים**:  ----------------  **קוד:**   1. השם לאת זיכרון 60 את הערך 34 2. השם לתא זיכרון 65 את הערך 15 3. הוסף לתא זיכרון 60, את הערך 1 4. העבר את הערך של תא 60 לרגיסטר ax 5. הוסף לרגיסטר ax את הערך שת תא זיכרון 65 6. העבר את הערך שברגיסטר ax לתא זיכרון 65 | DATASEG  ;---------------------  CODESEG  start:  mov [60], 34  mov [65], 15  add [[60], 1  mov ax, [60]  add ax, [65]  mov [65], ax |

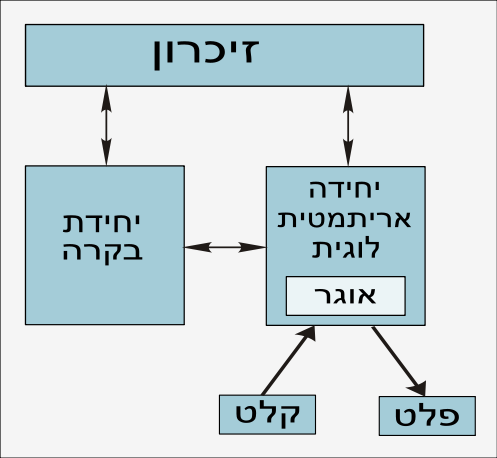
**שאלה 2:**

שימו 15 ברגיסטר Ax  
שימו 32 בכתובת 55, בזיכרון המחשב  
חשבו Ax + [55] + 20 ושימו את התוצאה ב – Bx.

שימו לב 🎔, ניתן לחבר רק שני מספרים בכל פעם.

|  |  |
| --- | --- |
| **נתונים**:    ----------------  **קוד:**   1. השימו לרגיסטר ax את הערך 15 2. השמו בתא זיכרון 55 את הערך 32 3. העבירו את הערך של תא [55] לרגיסטר ax 4. הוסיפו לרגיסטר ax את הערך של תא [55] 5. הוסיפו לרגיסטר ax את הערך 20 6. העבירו את הערך של ax לרגיסטר bx | DATASEG  ;---------------------  CODESEG  start:  mov ax, 15  mov [55], 32  add ax, [55]  add ax, 20  mov bx, ax |

# אסמבלר הבנת פעולת המעבד.



התגלגל לידנו מחשב הבנוי בארכיטקטורת פון ניומן

למחשב זיכרון בן 100 תאים:

0-49 תאים המכילים פקודות

50-99 תאים המכילים משתנים

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **99** | **98** | **97** | **96** |  |  |  |  |  |  |  | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

למחשב מעבד המכיל 3 אוגרים:

Ax, Bx – לאחסון מספרים לצרכי חישוב

אוגר IP – שומר את כתובת הפקודה הבאה לביצוע

המעבד מסוגל לבצע 2 פעולות:

1. העבר מהזיכרון לרגיסטר או מרגיסטר לזיכרון לדוגמא:
   1. mov AX, [0]
   2. mov [2], BX

כתובת בזיכרון תסומן ע"י סוגריים מרובעים

1. חיבור 2 רגיסטרים לתוך הראשון ביניהם. לדוגמא:
   1. Add AX, BX 🡨התוצאה ב AX
   2. Add BX, AX 🡨 התוצאה ב BX

שמרו לזיכרון המחשב את הנתונים הבאים:

הערך 15 לתא זיכרון בכתובת [0]

והערך 20 לתא זיכרון בכתובת [1]

כתבו קטע "קוד" המחבר את 15 ו 20 ושמרו את התוצאה לתא זיכרון בכתובת [2]

שימו לב 🎔, לא ניתן לחבר ערכים הנמצאים בתאי זיכרון, ניתן לחבר רגיסטר עם רגיסטר או רגיסטר עם זיכרון.

**פתרון:**

mov [0], 15

mov [1], 20

mov Ax, [0]

mov Bx, [1]

add, Ax, Bx

mov [2], Ax

**שרטטו את התוצאה של הרגיסטרים ושל זיכרון המחשב בסוף ההרצה.**

|  |  |
| --- | --- |
| **ערך** | **רגיסטר** |
| 15, 35 | AX |
| 20 | BX |

זיכרון המחשב

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **כתובת** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| ערך |  |  |  |  |  |  |  |  | 35 | 20 | 15 |

תרגילים נוספים:

mov [0], 34

mov [1], 15

add [1], 3

mov ax, [0]

add ax, [1]

mov [2], ax

**שאלה 1:**

כתבו את קטע הקוד הבא:

שימרו את המספר 34 לתא זיכרון [0]  
שימרו 15 את המספר לתא זיכרון [1]  
הגדילו את ערכו של תא זיכרון [1], ב – 3

חברו את המספרים השמורים בתאים [0] ו – [1],

ושמרו את התוצאה לתא זיכרון בכתובת [2].

**שרטטו את התוצאה של הרגיסטרים ושל זיכרון המחשב בסוף ההרצה.**

|  |  |
| --- | --- |
| **ערך** | **רגיסטר** |
| 34  52 | AX |
|  | BX |

זיכרון המחשב

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **כתובת** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| ערך |  |  |  |  |  |  |  |  | 52 | 15  18 | 34 |

***תא* זיכרון *אשר שמור בו ערך נקרה משתנה.***

**שאלה 2:**

mov [0], 10

mov [1], 12

add [0], 1

mov bx, [0]

add bx, [1]

add bx, 20

כתבו את קטע הקוד הבא:

שימו 10 במשתנה [0] (תא זיכרון פנוי ראשון ⇦ 0)  
שימו 12 במשתנה [1] ((תא זיכרון הבא הפנוי ⇦ 1)  
הגדילו את ערו של [0] ב – 1

חשבו [0]+[1]+20 ושימרו את התוצאה ל – [bx].

**שרטטו את התוצאה של הרגיסטרים ושל זיכרון המחשב בסוף ההרצה.**

|  |  |
| --- | --- |
| **ערך** | **רגיסטר** |
|  | AX |
| 43 | BX |

זיכרון המחשב

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **כתובת** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| ערך |  |  |  |  |  |  |  |  |  | 12 | 10  11 |

**בהצלחה**