**אסמבלר 24.10.18**

**מבנה המחשב**

a

MEMORY

CPu

I/O

Cpu to memory= address bus—חץ כחול

--Cpu to/back memory=data busחץ אדום

Cpu to memory=control bus—חץ ירוק

משמעות של גג היא שהפעולה מתבצעת במצב 0 (פעיל בנמוך '0' לוגי)

|  |  |  |
| --- | --- | --- |
| הפעולה שמבצע ה-CPU | WR | RD |
| מצב אסור | 0 | 0 |
| פעולת קריאה | 1 | 0 |
| פעולת כתיבה | 0 | 1 |
| מצב סרק(לא משתמש בפסים לצורך קריאה וכתיבה) | 1 | 1 |

כל מחשב מכיל שלוש יחידות בסיסיות :

1. CPU-יחידת עיבוד מרכזית, תפקידה לקרוא הוראות מהזיכרון ולבצען. ה-CPU היא היחידה המבצעת, המפקחת ומארגנת את כל הפעולות במערך המחשב. ה-CPU הוא החלק המרכזי החשוב ביותר והוא מהווה בעצם את המוח של כל המערכת.

* ביצוע-כל ההוראות שהמחשב מקבל לביצוע מבוצעות בתוך ה-CPU
* פיקוח-ה-CPU שולט על כל היחידות שמחוברות אליו וקובע את אופן פעולתן
* ארגון-במהירויות שבהן עובד המחשב יש צורך בתזמון מושלם של כל הפעולות המתקיימות בו, לצורך זה ה-CPU קובע את כל התזמונים הנדרשים

1. I/O -קלט פלט- זו היחידה שעוזרת לנו לתקשר עם "העולם החיצון", לדוגמא: מדפסת(פלט), מסך(פלט), מקלדת(קלט)
2. MEMORY-הזיכרון אוגר באופן זמני הוראות ונתונים הדרושים להפעלת המחשב. הזיכרון בנוי מהרבה תאים כאשר כל תא מזוהה על ידי מספר המוגדר ככתובת התא. פתיחת תא בזיכרון מתבצעת על ידי מתן כתובת לזיכרון. הגורם היחיד שיכול לתת את הכתובת לזיכרון הוא ה-CPU. עובר פרק זמן מסויים מהרגע שהמעבד מוסר את הכתובת ועד הרגע שבו התא נפתח (Memory Access Time-זמן גישה לזיכרון), זמן זה בזכרונות מהירים יכול להגיע עד עשרות ננו-שניות.

הוצאת מידע מתוך תא בזיכרון נקראת פעולת קריאה(READ),בפעולה זו מה שהיה קודם באותו התא נשאר בתא. הכנסת מידע לתוך תא בזיכרון נקראת כתיבה(WRITE),בפעולה זו מה שהיה קודם בתא נדרס\נמחק\נעלם.  
הקשר בין חלקי המחשב נעשה בעזרת שלוש קבוצות של מוליכים כאשר בכל מוליך עובר 0 או 1 לוגי.חלקי המחשב קשורים בינהם באמצעות מוליכים חשמליים שרמות המתח בינהם מתאימות לשני מצבים:0 לוגי אין מתח (0V) ו1 לוגי יש מתח (5V).  
אם רוצים להעביר ממקום למקום ביטים(סיביות\מידע), אנו למעשה מעבירים בקבוצות המוליכים אותות חשמליים,מוליכים אלו מחולקים לפי תפקידיהם לשלוש קבוצות כאשר כל קבוצה נקראת BUS (פס):

* Address bus-פס הכתובות-מסומן באותיות A0-An, בעזרת פס זה ה-CPU קובע לאיזו כתובת הוא פונה בזיכרון או ב-I/O.
* DATA BUS-פס הנתונים- מסומן באותיות D0-D7, דרכו ה-CPU מעביר נתון לזיכרון (פעולת כתיבה) או מקבל נתון מהזיכרון או הi/o (פעולת קריאה)
* CONTROL BUS-פס הבקרה- לא מסומן, בעזרת פס זה קובע ה-CPU האם הוא קורא נתונים או כותב נתונים לכתובת שאליה הוא פונה.

**משפחת ה-8086**

CPU

"המבנה של פון ניומן"-

חילוק של הCPU לשניים בכך נוצר

מצב של קיצור זמן העיבוד בחצי

שלבי ביצוע הפקודה:

כאמור ה-CPU קורא הוראות מהזיכרון ומבצע אותן. תהליך של כל ביצוע פקודה כולל שני שלבים:

1. OPCODE FETCH- הבאה הפקודה-
2. לאחר שהפקודה מגיעה מהזיכרון היא מפוענחת ומתבצעת , שלב זה נקרא "שלב ביצוע הפקודה".

המעבדים שנוצרו עד לפיתוחה של משפחת ה-8086 עבדו במבנה שנקרא "המבנה של פון ניומן",בצורה זו ה-CPU ניגש ומבין פקודה מהזיכרון, מבצע אותה ורק לאחר מכן ניגש ומביא פקודה נוספת. כלומר כל תהליך חדש של ביצוע פקודה מתחיל רק לאחר סיום פקודה קודמת.

**המבנה של משפחת ה-8086**

החידוש במשנה של משפחה זו הוא שהמעבד מורכב משתי יחידות עצמאיות:

1. BIU-Bus Interface Unit- יחידת הממשק לפס, תפקידה של יחידה זו הוא להביא הוראות ונתונים מהזיכרון ולשלוח נתונים לזיכרון.
2. ALU-Arithmetic logic unit-יחידה אריתמתית לוגית- תפקידה לבצע פעולות לוגיות כמו XOR NOT OR וכו' ופעולות חשבניות כמו חיבור וחיסור בינארי.

ההוראה הנוכחית שה-CPU מבצע היא זו שקובעת איזו פעולה צריכה יחידת הALU לבצע. פעולת ה-CPU בשיטה זו נקראת שיטת קו הצינור PipeLine. ההבדל הגדול בין המבנה של פון ניומן לבין המבנה של משפחת ה-8086 הוא בעובדה שיחידת הממשק לפס מביאה הוראות מהזיכרון בתהליך הנקרא Prefetch, כלומר מביאים פקודות מהזיכרון ומאחסנים אותן ביחידה הנקראת תור ההוראות. כאשר יחידת הביצוע מסיימת לבצע פקודה, הפקודה הבאה ממתינה לה כבר בתוך הPipeLine וכך נכסך הזמן שהיה דרוש להביא את הפקודה.

\*יש לציין שתור ההוראות מתנהג כמו זיכרון מסוג FIFO-first in first out הפקודה הראשונה שנכנסה תתבצע ראשונה.

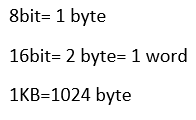
**שפת תכנות-אסמבלי(להוריד תוכנה emu8086)**

השפה הבסיסית שאותה מבין -CPU היא שפת מכונה (שפה של 0 ו 1 לוגיים) שפת סף (אסמבלי) היא שפה שכל פקודה בה מתורגמת לפקודה אחת בשפת מכונה. כאמור תוכנית מחשב היא אוסף של מספרים בינאריים הנמצאים בתאי הזיכרון בזה אחר זה, סדר המספרים של הפקודות קובע את משמעות התוכנית ואת פעולתו של ה-CPU.

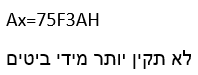
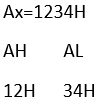
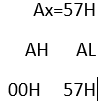
לכל מעבד יש כתובת התחלה שהיא כתובת התא שמכיל את ההוראה הראשונה בתוכנית ( ב8086 הכתובת FFFF0H) כדי לגרום ל-CPU לבצע תוכנית יש להפנות אותו לתא הראשון וזאת על ידי פעולת RESET ומשם הוא ממשיך בתוכנית ב"כוחות עצמו" על ידי ה-IP,

**האוגרים של ה-8086**

אוגר- זהו תא זיכרון בודד המשמש להגירה זמנית של נתונים (\*בשפת אסמבלי אני נכתוב תוכניות ונעזר באוגרים הקיימים ב-8086\*)

אוגרים כלליים ואוגרי נתונים(מכילים עד 16bit אם יש יותר זה שגוי):

|  |  |  |
| --- | --- | --- |
|  |  | שם האוגר |
| AL  8bit | AH  8bit | Ax |
| BL  8bit | BH  8bit | Bx |
| CL  8bit | CH  8bit | Cx |
| DL  8bit | DH  8bit | Dx |



אוגרי הצבעה ואינדקסים(אוגרים של 16BIT לא ניתנים לחלוקה):

* SI- מצביע על כתובת של נתון בזיכרון
* DI-מצביע על כתובת של נתון בזיכרון
* BP-מצביע עזר על המחסנית
* SP-מצביע המחסנית
* IP-מצביע על הכתובת של הפקודה הבאה לביצוע בתוכנית

אוגרי סגמנט(16BIT ללא חלוקה):

* DS-Data segment-אוגר סגמנט הנתונים
* ES-Extra segment-אוגר סגמנט נתונים נוסף
* SS-Stack segment- אוגר סגמנט המחסנית
* CS-Code segment-אוגר סגמנט הקוד\התוכנית

אוגר הדגלים:

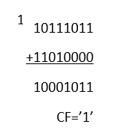
היחידה האריתמתית לוגית (ALU) ,תפקידה לבצע את הפעולות במערך המחשב, בנוסף יחידה זו מפיקה מידע על התוצאה שהתקבלה. מידע זה עובר לאוגר מיוחד שהסיביות שלו נקראות דגלים.

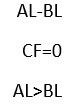
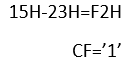
דגל זוהי סיבית שנמצאת בתוך אוגר הדגלים שגודלו 16 סיביות, ורק 9 מהן מנוצלות.

ישנם שני סוגי דגלים:

1. דגלי מצב שתפקידם להראות משהו על התוצאה שהתקבלה ביחידת ה-ALU.

* ZF-Zero flag- דגל זה מקבל 1 לוגי כאשר התוצאה היא 0. ודגל זה מקבל 0 לוגי כאשר התוצאה שונה מ-0. אחד השימושים לדגל זה הוא בדיקת שיוויון או אי שיוויון בין שני ערכים.
* CF-Carry flag-דגל הנשא- דגל זה מקבל 1 לוגי לאחר פעולת חיסור כאשר לווינו מדרגה דימיונית (כאשר מבצעים פעולת חיסור ממספר קטן פחות מספר גדול).  
  לאחר פעולת חיבור דגל זה מקבל 1 לוגי כאשר מתקבלת סיבית 9 או סיבית 17   
  \*אחד השימושים לדגל זה הוא בדיקת היחס קטן-גדול.

\*כאשר מחסרים שתי ספרות בשתי ספרות רושמים רק שתי ספרות מימין לשמאל של התוצאה



1. דגלי בקרה שתפקידם לעזור למתכנת לשלוט על אלמנטים מסויימים בפעולת המעבד