

מיקרו מעבדים ושפת אסמבלר תרגיל 2:

אחרית תרגיל- מאיה כתר

חלק יבש:

- (1) עבור כל אחת מהפקודות הבאות קבעו:
- א. האם הפקודה חוקית או לא - כלומר האם קיים opcode המתאר בדיוק את הפקודה הנ"ל.
- ב. אם הפקודה חוקית כתבו מהו ה-opcode המתאים לפקודה
אם הפקודה אינה חוקית – כתבו קוד אסמבלר חלופי המבצע פעולה שקולה

הפקודה	Opcode עבור פקודה חוקית	קוד חלופי- עבור פקודה לא חוקית
MOV AX, [BX+4]		
MOV AX, [BX-SI]		
MOV AX, [BX+2*400]		
PUSH [BX+SI]		
PUSH [BX+SI+DI]		
PUSH AL		
PUSH BX		

- (2) נכון/ לא נכון.
- אם נכון כתבו את המספר שישמר באוגר.
- א) האם הפקודות SHL DL, 4 ו- ROL DL, 4 שקולות?
- ב) האם הפקודות SHR CL, 2 ו- ROR CL, 2 שקולות?
- ג) האם הפקודות SHL DH, 4 ו- RCL DH, 4 שקולות?
- ד) האם הפקודות ROL AL, 3 ו- RCL AL, 3 שקולות?
- (3) א) באוגר BX יש מספר כלשהו. כתבו קטע אסמבלר קצר המחליף בין הבית התחתון לבית העליון של האוגר. בעזרת פקודות הזזה בלבד.
- לדוגמה: אם BX=5A9Ch לפני ביצוע הקוד, לאחר ביצוע קוד ההחלפה נקבל BX=9C5Ah.
- ב) באוגר AL יש מספר כלשהו. כתבו קטע אסמבלר קצר המחליף את ארבע הסיביות התחתונות של האוגר עם הארבע העליונות שלו.
- לדוגמה: אם AL=82h לפני ביצוע הקוד, לאחר ביצוע קוד ההחלפה נקבל AL=28h
- *אין צורך לכתוב קוד תוכנית מלא רק לכתוב את שורות הקוד המבצעות את ההחלפה,

חלק רטוב:

תרגיל 1:

כתבו תוכנית המדפיסה את תוכנו של אוגר AX למסך בפורמט הקסה-דצימלי.

הדגימו את התוכנית שלכם ע"י ביצוע הפעולות הבאות
איתחול AX לערך 0, והדפסתו.
איתחול AX ל -357 (מינוס שלוש מאות חמישים ושבע, דצימלי) והדפסתו.
איתחול AX ל 12B4h והדפסתו.

את התוכנית המלאה (עם 3 הדוגמאות) שמרו בקובץ בשם printAX.asm

תרגיל 2:

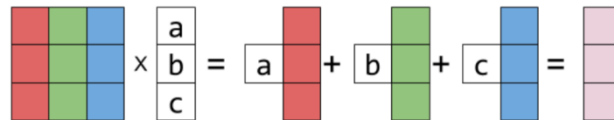
בשאלה זו נעסוק בכפל של מטריצה במטריצה.

(א) ראשית נכתוב קוד של מכפלה בין וקטור למטריצה:

תזכורת: ניתן לכתוב את המכפלה בין וקטור למטריצה כך:

$$A^{N \times M} * B^{M \times 1} = (AB)^{N \times 1}$$

$$(AB)_i = \sum_{j=1}^M a_{ij} b_j$$



בתרגיל נעבוד עם מטריצות בגודל $N \times N$ ווקטורים בגודל $N \times 1$.

על מנת לשמור את המטריצה בזיכרון נשטח אותה לצורה של מערך חד ממדי. כאשר כל N תאים רציפים מתייחסים לעמודה אחרת, כפי שניתן לראות באיור הבא:



עליכם לבנות תוכנית המקבלת מטריצה בגודל $N \times N$, וקטור בגודל N ומגדירה את הקבוע N שמייצג את גדלי המטריצה והווקטור. על התוכנית לבצע כפל בין המטריצה לווקטור ולשמור את התוצאה במערך בזיכרון.

המשתמש בתוכנית צריך להגדיר בתוך קוד התוכנית, בסגמנט הנתונים שלושה אלמנטים:

- (1) MAT - המטריצה $N \times N$ בייצוג החד ממדי שלה, מאותחלת בערכים שהמשתמש רוצה להכפיל
- (2) VEC - וקטור בגודל N , מאותחל בערכים שהמשתמש רוצה להכפיל.
- (3) RESULT - וקטור בגודל N . מערך לא מאותחל.

כמו כן בתוכנית מוגדר הקבוע " N " המוגדר לגודל של ממד המטריצה (N).

לדוגמה: עבור הכפלת המטריצה הבאה בגודל 3×3 בווקטור בגודל 3×1 נאתחל את סגמנט הנתונים באופן הבא:

N EQU 3

.data

MAT db 2,3,1,0A,8,1,0F,5,4
VEC db 7,0D,6
RESULT dw N dup(?)

$$\begin{pmatrix} 2 & A & F \\ 3 & 8 & 5 \\ 1 & 1 & 4 \end{pmatrix} \cdot \begin{pmatrix} 7 \\ D \\ 6 \end{pmatrix}$$

את תוצאת הכפל שמרו בתוך הווקטור RESULT. שימו לב כי תוצאת הכפל היא בספרות הקסה-דצימליות.

הערות:

- המספרים במטריצה הם *signed* כלומר המספר *F6h* שלילי. החיבור מודולו 16 ביט.
- התוכנית צריכה להתמודד עם כפל של וקטור ומטריצה עבור כל N בין 2 ל-32
- הגדירו את *MAT, VEC* כמערך של בתים (*DB*) ומערך *RESULT* כמערך של מילים (*DW*)
- שמרו בקובץ בשם *mulVec.asm*

(ב) כעת נכתוב קוד לכפל מטריצה במטריצה.

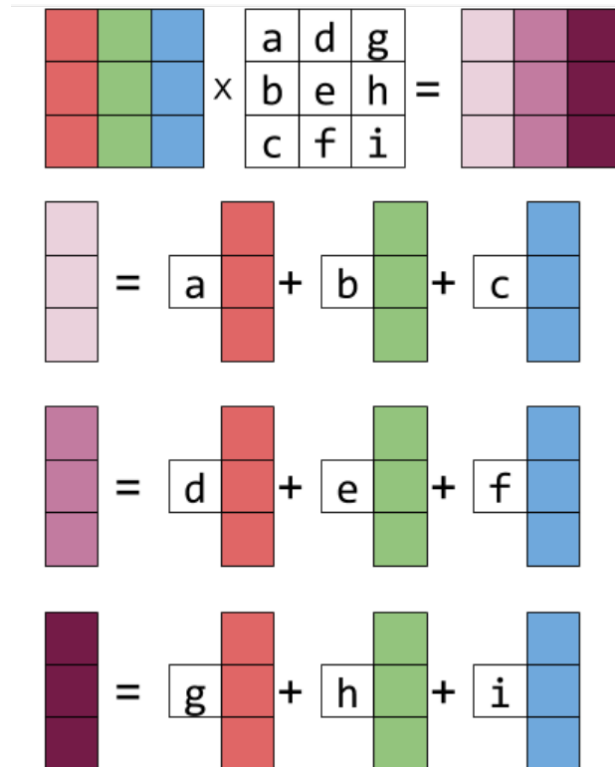
בתרגיל שלנו המטריצות יהיו ריבועיות מגודל $N \times N$

תזכורת: ניתן לכתוב את המכפלה בין מטריצה למטריצה כך:

$$A^{N \times N} * B^{N \times N} = (AB)^{N \times N}$$

$$Col_j(AB) = A * Col_j(B)$$

$$= \sum_{i=1}^N Col_i(A) b_{i,j}$$



המשתמש בתוכנית צריך להגדיר בתוך קוד התוכנית, בסגמנט הנתונים שלושה אלמנטים:

- (1) *MAT1* - מטריצה $N \times N$ בייצוג החד ממדי שלה, מאותחלת בערכים שהמשתמש רוצה להכפיל
- (2) *MAT2* - מטריצה $N \times N$ בייצוג החד ממדי שלה, מאותחלת בערכים שהמשתמש רוצה להכפיל
- (3) *RESULT* - מטריצה בגודל $N \times N$ בייצוג החד ממדי שלה כמערך לא מאותחל.

$$RESULT \text{ תשמור את תוצאת הכפל} \\ MAT1^{N \times N} * MAT2^{N \times N}$$

כמו כן בתוכנית מוגדר הקבוע "N" המוגדר לגודל של ממד המטריצה (N).

שימו לב כי תוצאת הכפל היא בספרות הקסה-דצימליות.

הערות ורמזים:

- המספרים במטריצה הם *signed* כלומר המספר *F6h* שלילי. החיבור מודולו 16 ביט.
- התוכנית צריכה להתמודד עם כפל של וקטור ומטריצה עבור כל N בין 2 ל-32
- כדאי להיעזר בקוד של סעיף א של כפל מטריצה בווקטור עמודה.
- הגדירו את *MAT1, MAT2* כמערך של בתים (DB) ומערך *RESULT* כמערך של מילים (DW).
- כל המטריצות מוגדרות בייצוג החד ממדי כפי שהוגדר בסעיף א.
- שמרו בקובץ בשם *mulMat.asm*

בשני הסעיפים אפשר להשתמש במשתנים נוספים כרצונכם

חובה להוסיף הערות והסברים לקוד.

שאלות עיוניות לתרגיל 2 :

- 1) הסבירו מדוע המערך *RESULT* הוא מערך של מילים (DW) ואילו *MAT1, MAT2* הם מערכים של בתים (DB)
 - 2) כיצד אתם בודקים את נכונות החישוב שלכם?
 - 3) כיצד ביצעתם את הכפל על מנת שגם תוצאת מכפלת המספרים תהיה *signed*?
- את התשובות לשאלות אלה יש להוסיף לדף התשובות של החלק היבש תחת הכותרת "שאלות עיוניות תרגיל רטוב"

הנחיות הגשה:

הגשה במודל עד תאריך 28.4.2022 . ההגשה בזוגות, אחד מכל זוג מגיש.

נא להוסיף בתחילת כל קובץ הערה עם שמות המגישים ותעודות זהות.

הגישו תיקיית *zip* בשם *Name1_ID1_Name2_ID2.zip* הכוללת את הקבצים: *mulMat.asm*, *mulVec.asm*, *printAX.asm*, *hw2.pdf*