

## תרגיל בית מספר 1

תאריך הגשה: לא יאוחר מה 10.4 בחצות.

הגשה בבודדים בלבד.  
יש לכתוב פתרונות קצרים, מדויקים וברורים. פתרונות בלתי ברורים יגררו הורדת נקודות.

**רשימת פעולות מותרות על האוגרים והflip-flops מצורפת בסוף התרגיל.** לא ניתן להוסיף עוד פעולות אלא אם כן מצוין בשאלה עצמה.

### תרגיל 1:

(כל המספרים נתונים בהקסה).

תוכן ה PC במחשב (מנו) הינו 3AF.

תוכן ה AC במחשב הינו 7EC3.

תוכן תא 03AF בזיכרון הינו 951E.

תוכן תא 051E בזיכרון הינו 0AAC.

תוכן תא 0AAC בזיכרון הינו 8B9F.

א. מה שם הפקודה שתבצע?

ב. מהי הפעולה הבינארית שתבצע על ה AC?

ג. שרטטו את מבנה הזיכרון וסמנו את התאים הרלוונטיים (בדומה למה שעשינו בכיתה).

ד. צרו טבלה כמו שיצרנו בכיתה (מובאת בהמשך) ומלאו אותה עד לסיום הפקודה.

הטבלה (ניתן להוסיף עוד שורות אם יש צורך)

| SC   | PC | AR | DR | AC | IR | E | I |
|------|----|----|----|----|----|---|---|
| init |    |    |    |    |    |   |   |
| 0→1  |    |    |    |    |    |   |   |
| 1→2  |    |    |    |    |    |   |   |
|      |    |    |    |    |    |   |   |

### תרגיל 2:

נחליף את כל הפקודות שעוסקות בזיכרון (הפקודות המסומנות בשקפים בצהוב), עם הטבלה בהמשך (כולל כתובות ישירות ועקיפות)

נשתמש בסימונים הבאים:

לציון הכתובת האפקטיבית – נשתמש בסימון ef

נניח שה ALU יודע לבצע פקודת XOR ונסמנה בסימן  $\oplus$ .

ביצוע של חיסור יבוצע ע"י הפיכת המספר (משלים ל-1) והוספת 1

השלימו את כל הפקודות החל משלב  $T_4$

תוכן ה AC צריך להישאר כמות שהוא בסיום הפקודה, אלא אם כן הפקודה דורשת שינוי.

ניתן להשתמש ב TR במידת הצורך

| OP CODE | Symbol | Symbolic description                       | description             |
|---------|--------|--|-------------------------|
| 000     | XOR    | $AC \leftarrow AC \oplus M[ef]$            | XOR to AC               |
| 001     | MAD    | $M[ef] \leftarrow M[ef] + AC$              | Add AC to MEM           |
| 010     | SUB    | $AC \leftarrow AC - M[ef]$                 | Subtract MEM from AC    |
| 011     | SWP    | $AC \leftarrow M[ef], M[ef] \leftarrow AC$ | Swap MEM and AC         |
| 100     | SEQ    | If ( $AC = M[ef]$ ) $PC \leftarrow PC + 1$ | Skip if AC equal to MEM |
| 101     | BAP    | If ( $AC > 0$ ) $PC \leftarrow ef$         | Branch if AC positive   |

### תרגיל 3:

רוצים להוסיף פקודה בשם SBK שתבצע את :  $M[EA] \leftarrow M[EA] * 2^k$  כאשר k הוא מספר ה-1ים ב AC. קוד הפקודה SBK הוא 4. בסיום הפעולה AC ו- E מתאפסים.

לדוגמא (אם לא מצוין אחרת, הערכים בזיכרון נתונים בדצימלי, והערכים ב AC נתונים בבינארי) :

$M[EA] = 8$  ו-  $AC = 1010101000000000$  אזי בסיום הפקודה  $M[EA] = 8 * 2^4 = 128$

$M[EA] = 10$  ו-  $AC = 0000110000100000$  אזי בסיום הפקודה  $M[EA] = 10 * 2^3 = 80$

$M[EA] = 297$  ו-  $AC = 0$  אזי בסיום הפקודה  $M[EA] = 297 * 2^0 = 297$

$M[EA] = 1000(\text{hexa})$  ו-  $AC = 0000111100000000$  אזי בסיום הפקודה  $M[EA] = 1000(\text{hexa}) * 2^4 = 0$  מכיוון

שמתרחשת גלישה.

הניחו שאם ישנה גלישה היא לא באחריותכם.

הניחו שמרחיבים את sequencer ליותר מ- 4 סיביות.

$T_0$  :  $AR \leftarrow PC$

$T_1$  :  $IR \leftarrow M[AR], PC \leftarrow PC + 1$

$T_2$  :  $D_0, \dots, D_7 \leftarrow \text{Decode } IR(12-14), AR \leftarrow IR(0-11), I \leftarrow IR(15)$

$D_7 \dots IT_3$  :  $AR \leftarrow M[AR]$

$T_4$  :  $AC \leftarrow \text{SHR}(AC), E \leftarrow AC[0], AC[15] \leftarrow E$ , \_\_\_\_\_

$T_5$  : (if  $E = 1$  then  $DR \leftarrow DR + 1$ ),  $AC \leftarrow \text{SHR}(AC), E \leftarrow AC[0], AC[15] \leftarrow E$

$T_6$  : (if  $E = 1$  then  $DR \leftarrow DR + 1$ ),  $AC \leftarrow \text{SHR}(AC), E \leftarrow AC[0], AC[15] \leftarrow E$

...

$T_i$  : (if  $E = 1$  then  $DR \leftarrow DR + 1$ ),  $AC \leftarrow \text{SHR}(AC), E \leftarrow AC[0], AC[15] \leftarrow E$

$T_{i+1}$  : (if  $E = 1$  then  $DR \leftarrow DR + 1$ ),  $AC \leftarrow \text{SHR}(AC), E \leftarrow AC[0], AC[15] \leftarrow E$

$T_{i+2}$  :  $AC \leftarrow DR$

$T_{i+3}$  : \_\_\_\_\_

$T_{i+4}$  :  $AC \leftarrow AC + 1, DR \leftarrow$  \_\_\_\_\_

$T_{i+5}$  :  $AC \leftarrow DR, DR \leftarrow AC$

$T_{i+6}$  :  $E \leftarrow 0$ , (if  $DR = 0$  then \_\_\_\_\_)

\_\_\_\_\_  $T_{i+7}$  :  $AC \leftarrow SHL(AC), AC[0] \leftarrow E, E \leftarrow AC[15], DR \leftarrow DR + 1$   
 \_\_\_\_\_  $T_{i+8}$  :  $E \leftarrow 0$ , (if  $DR = 0$  then \_\_\_\_\_)  
 \_\_\_\_\_  $T_{i+9}$  :  $AC \leftarrow SHL(AC), AC[0] \leftarrow E, DR \leftarrow DR + 1$   
 ...  
 \_\_\_\_\_  $T_k$  :  $E \leftarrow 0$ , (if  $DR = 0$  then \_\_\_\_\_)

ענו על השאלות הבאות:

- איזו פקודה יש להוסיף בצעד  $T_4$  ?
- מה צריך להיות הערך של  $i$  בפקודה  $T_i$  ?
- מה צריך להיות כתוב בפקודה  $T_{i+3}$  ?
- השלימו את הפקודה  $T_{i+4}$
- השלימו את פקודה  $T_{i+6}$
- האם הפקודה  $T_{i+8}$  זהה לפקודה  $T_{i+6}$  ?
- האם הפקודה  $T_k$  זהה לפקודה  $T_{i+6}$  ?
- מה מספר הסיביות המינימלי של ה sequencer שנדרש בכדי לסיים את הפקודה בהצלחה?
- הערך של  $k$  בפקודה האחרונה יהיה גדול או קטן מ-60 ?

#### תרגיל 4:

הוחלט להחליף במחשב המחווט של מאנו את הפקודה ISZ בפקודה

MAV:  $M[EA] \leftarrow (M[EA-1] + M[EA+1]) / 2$

ממשו את הפקודה.

הניחו שה- SC הורחב ל-5 ביטים.

#### תרגיל 5

בשאלה זו אתם נדרשים לממש פעולת חיסור עם נשא (subtract with borrow) בין שני תאים בזיכרון. התא הראשון יהיה כרגיל מופנה מהפקודה והתא השני יבוא בכתובת העוקבת לפקודה. Operation code של הפקודה הינו 5. תוצאת החישוב תיכתב בכתובת המופנית על ידי הפקודה.

הגדרת חיסור עם נשא:

if carry = 0 then result  $\leftarrow$  A - B  
 else result  $\leftarrow$  A + B' // (carry = 1)

#### דוגמא:

| Address(hexa) | Value(binary)       |
|---------------|---------------------|
| 100           | 0101 0011 0000 0000 |
| 101           | 0000 0000 0000 0100 |
| ...           |                     |
| 300           | 0000 0000 0000 0110 |

הפקודה נמצאת בכתובת 100 (הקסדימלי)  
 מפנה לכתובת 300 (הקסדימלי) שם נמצא הערך A  
 בכתובת 101 (הקסדימלי) נמצא הערך B  
 ה carry נמצא כרגיל ב- E.  
 נניח שה carry = 0 לפני. התוצאה 2 תמצא בכתובת 300 (הקסדימלי) בסיום הפקודה.

```

T0 : AR <- PC
T1 : IR <- M[AR], PC <- PC+1
T2: AR <- IR, D7..D0 <- IR(14-12), I <- IR(15)
D7'I T3 : AR <- M[AR]
D5 T4 : DR <- M[AR]
D5 T5 : TR <- AC, AC <- DR, _____
D5 T6 : _____
D5 T7 : DR <- M[AR]
D5 T8 : DR <- AC, AC <- DR
D5 T9 : _____
_____
D5 T11 : AC <- AC + DR, _____
_____
D5 T13 : M[AR] <- AC, AC <- DR, SC <- 0

```

השלימו את השורות החסרות.

בהצלחה!!

### רשימת פעולות RTL שניתן לבצע על הרגיסטרים וה flip-flops:

- ניתן לשאול על הערך שנמצא ב I או ב E [ is E = 0/1 , is I = 0/1 ]
- ניתן לשאול על ה AC [ is AC = 0, is AC < 0, is AC ≥ 0 ]
- ניתן לשאול על ה DR [ is DR = 0 ]
- ניתן לבצע CLR, LOAD, INC על הרגיסטרים : AR, PC, DR, AC, TR
- ניתן לבצע AND, ADD בין ה AC ל DR
- ניתן לבצע משלים ל1 של ה AC [  $\overline{AC}$  ] וכן SHL או SHR של ה AC (כולל או לא כולל ה E)
- ניתן לבצע משלים ל1 של ה E [  $\overline{E}$  ]