שאלון X

חלק א: (33 נקודות)

בחלק זה 11 שאלות.

יש לבחור את התשובה הנכונה מבין האפשרויות.

משקל כל שאלה 3 נקודות.

יש לסמן X באופן ברור בטבלה שלפניכם

הערה: יש לסמן רק **אפשרות אחת** לכל שאלה!

לתשומת ליבכם: בדיקת התשובות מבוצעת רק בהתאם לכתוב בטבלה זו. הרישומים ליד השאלות עצמן לא נבדקים (ההתייחסות לרישומים אלה הינה כאל טיוטא)!

שאלה	א	ב	ג	ד	ה
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

שאלה 1

נתון קטע הקוד הבא, שהינו תרגום של פקודת פסיאודו XXX:

Address	Code	Basic	Source
0x00400000	0x000a0fc3	sra $1, $10, 0x0000001f	XXX $t1, $t2
0x00400004	0x002a4826	xor $9, $1, $10	
0x00400008	0x01214823	subu $9, $9, $1	

נתון כי ערכו של אוגר $t2 במקרה א' - הוא (5-), ונתון כי ערכו של אוגר $t2 במקרה ב' הוא (10). מה יהיה ערכו של אוגר $t1 בסוף קטע קוד זה, בכל אחד המקרים, ומה מבצעת פקודת הפסיאודו XXX?

א. מקרה א': $t1 = 5, מקרה ב': $t1 = 10, הפקודה XXX מחשבת את הערך המוחלט

ב. מקרה א': $t1 = 5, מקרה ב': $t1 = -10, הפקודה XXX מחשבת את הערך ההופכי

ג. מקרה א': $t1 = -5, מקרה ב': $t1 = -10, הפקודה XXX מחשבת את הייצוג השלילי של $t1

ד. מקרה א': $t1, = -5, מקרה ב': $t1 = 10, הפקודה XXX, שומרת ב $t2 את ערכו של $t1

ה. לא ניתן להבין מקטע הקוד מה הוא מבצע

שאלה 2:

נתון קטע הקוד הבא:

```
        addi    $s0, $zero, -1
        addi    $t0, $zero, -1
        xor     $v0, $v0, $v0
loopX:
        addi    $v0, $v0, 1
        srl     $s0, $s0, 1
        and     $t0, $t0, $s0
        bne     $t0, $zero, loopX
```

כמה פעמים תתבצע הלולאה ומה יהיה ערכו של אוגר $v0 בסיום קטע הקוד

א. הלולאה תתבצע פעם אחת, וערכו של אוגר $v0 בסיום קטע הקוד יהיה 1

ב. הלולאה תתבצע 32 פעמים, וערכו של אוגר $v0 בסיום קטע הקוד יהיה 1

ג. הלולאה תתבצע 32 פעמים, וערכו של אוגר $v0 בסיום קטע הקוד יהיה 32

ד. הלולאה תתבצע פעם אחת, וערכו של אוגר $v0 בסיום קטע הקוד יהיה 32

ה. זו לולאה אין סופית, כי אוגר $t0 תמיד יהיה שונה מ- 0

שאלה 3

נתונה פקודת המכונה הבאה והכתובת בה היא נמצאת:

Address	Code
0x00400060	0x116bfff3

איזו פקודה זו, ומה הכתובת של הפקודה הבאה שתתבצע?

א. אין מספיק נתונים לדעת איזו פקודה זו, והפקודה הבאה לביצוע תהיה בכתובת 0x00400064

ב. זו פקודה R, והפקודה הבאה לביצוע תהיה בכתובת 0x00400064

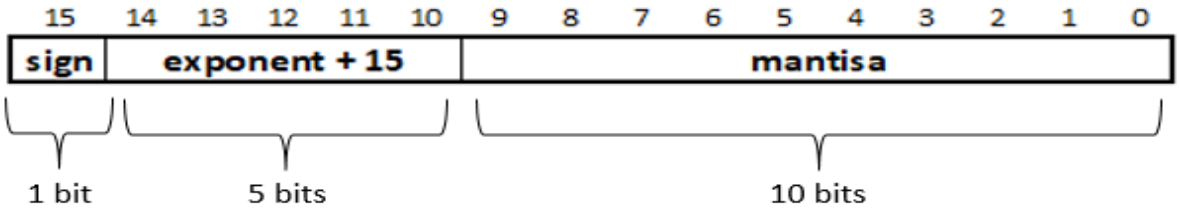
ג. זו פקודת Jump, והפקודה הבאה לביצוע תהיה בכתובת 0x0040002C

ד. זו פקודת BEQ, והפקודה הבאה לביצוע תהיה בכתובת 0x0040002C

ה. זו פקודת BEQ, והפקודה הבאה לביצוע תהיה בכתובת 0x00400030

יש להתייחס לנתון הבא בפתרון שאלות 4 ו-5:

במחשב העובד עם 16 סיביות/bits (במקום 32) נתון כי ייצוג מספרים ב- Floating Point הינו במבנה הבא:



שאלה 4

על בסיס הגדרת מבנה הייצוג ב Floating Point הנתון: כיצד נייצג, ב- 4 ספרות הקסה, את הערך (-35.75)?

א. 0x5078

ב. 0x9038

ג. 0x9478

ד. 0xD078

ה. 0xD238

שאלה 5

על בסיס הגדרת הייצוג ב Floating Point הנתון, מהו הערך העשרוני של **0x4EA0**?

א. 26.5

ב. 26.05

ג. 16.5

ד. 16.05

ה. 0.2605

שאלה 6:

ביצוע Push, לאוגר $x, למחסנית מבוצע באמצעות 2 הפקודות הבאות:

```
addi  $sp, $sp, -4
sw    $x, 0($sp)
```

הוחלט לפתח פקודת MIPS בסיסית לביצוע push – ובמקום לרשום 2 פקודות אלה, ניתן יהיה לרשום פקודה אחת, אשר תבצע בפקודה אחת את שביצעו 2 הפקודות הקודמות:

```
push  $x
```

נתונה תוכנית אשר, לפני פיתוח הפקודה החדשה, 15% מהפקודות הן פקודות עבור ביצוע ה push (כלומר, 85% "שאר התוכנית" ו- 15% זה זוג הפקודות לביצוע ה push).

נתונים נוספים על המעבד/תוכנית:

- CPI של פקודות אריתמטיות (כולל addi) הוא 2
- CPI של פקודות sw הוא 4
- CPI של פקודת ה **push החדשה** הוא 3

מהירות המעבד הינה 4Ggz

על בסיס נתונים אלה, מהו מדד ה- speedup בעקבות שינוי זה?

א. לא ניתן לחשב מאחר ולא נתונה כמות הפקודות (IC) ולא נתון CPI ממוצע של שאר הפקודות

ב. 1.081

ג. 1.111

ד. 1.0526

ה. 1.095

שאלה 7:

נתונים הערכים הבאים באוגרים $s1 , $s0:

$s0 = 0xFFFFFFFF $s1 = 0x00000001

נתונות 2 הפקודות הבאות:

```
slt    $t0 , $s0 , $s1
sltu   $t1 , $s0 , $s1
```

בהתאם ל- 2 פקודות אלה, מה יהיו הערכים באוגרים $t0 ו- $t1:

א. $t0 = 0, $t1 = 0

ב. $t0 = 1, $t1 = 0

ג. $t0 = 0, $t1 = 1

ד. $t0 = 1, $t1 = 1

ה. לא ניתן לדעת על בסיס הנתונים לשאלה

שאלה 8:

במהלך ביצוע פקודת LW, מתרחש אירוע Page Fault בעת הגישה לזיכרון הנתונים.

בהתבסס על תרשים 4.66, איזה מהמשפטים הבאים **<u>אינו נכון</u>**?

א. אוגר PC יקבל את הערך 0x80000180, הכתובת של ה exception handler

ב. התקלה מתרחשת בשלב ה- MEM, ולכן יבוצע flush לאוגר MEM/WB כדי למנוע מהפקודה שגרמה לתקלה להמשיך בצנרת.

ג. יבוצע flush לאוגרים: IF/ID, ID/EX ו- EX/MEM כדי למנוע מהפקודות שאחרי הפקודה שגרמה לתקלה להמשיך בצנרת

ד. דגל RegWrite יאופס, כדי למנוע כתיבה שגויה למקבץ האוגרים על ידי הפקודה שבשלב WB

ה. אוגר EPC יעודכן בכתובת הפקודה שגרמה לתקלה, ואוגר cause יעודכן בקוד התקלה

שאלה 9:

במהלך הקורס התייחסנו לשלוש דרכים לפתרון Control Hazard של פקודות Branch. בהתבסס על מה שלמדנו, איזה מהמשפטים הבאים **<u>אינו נכון</u>**?

א. הקדמת בדיקת ה branch לשלב ה- ID (תרשים 4.65) מאפשרת זיהוי מוקדם לצורך לדלג, צמצום הבעיה ל nop אחד, במחיר של data hazard כבר בשלב ID.

ב. במצב של Branch Delay Slot החומרה תמיד מבצעת את הפקודה שאחרי ה branch, כאשר באחריות התוכנה לדאוג לשים פקודה מתאימה. זו מוסכמה של תוכנה עם החומרה. באמצעות אופטימייזר, בכ- 50% מהמקרים ניתן למצוא פקודה שניתן למקם אותה אחרי פקודת ה Branch, וכך ולהימנע מ- nop.

ג. באמצעות שיטת חיזוי דינאמי בחומרה, נשלב יחידת BTB שבאופן שוטף תעבוד לפי תחזית דילוג לפקודת branch, ובמידה ותהיה החמצה, באמצעות יחידת Mispredict Detection Unit תזוהה ההחמצה ויהיה טיפול בניקוי הפקודות השגויות מהצנרת ותיקון התחזית.

ד. יחידת ה Mispredict Detection Unit מקבלת את נתוני "כיוון התחזית" לפיה עבדה יחידת ה BTB בשלב ה IF, מקבלת את נתוני ה Branch האמיתיים כפי שחושבו בשלב ה MEM, ובהתאם יכולה לזהות האם הייתה החמצת תחזית או שלא. במידה והייתה החמצה, יבוצע ניקוי של 3 פקודות מהצנרת, עדכון אוגר PC בפקודה הנכונה שצריכה הייתה להיכנס לצנרת, ועדכון של יחידת ה BTB בתחזית העדכנית.

ה. מצב ה Branch Delay Slot בשילוב הקדמת הבדיקה לשלב ה- ID הינו המנגנון היעיל והטוב ביותר בהיבט הביצועים, מאחר ורק ב- 50% מהמקרים נדרש לבצע תיקון עם nop יחיד. בזמן שבאמצעות חיזוי דינאמי, תמיד יש ניקוי של 3 פקודות באמצעות nop/bubble כי הזיהוי מבוצע רק בשלב ה MEM, ולכן חיזוי דינאמי פחות יעיל מ Branch Delay Slot.

יש להתייחס לנתונים הבאים בשאלות 10-11

נתון זיכרון מטמון עם המאפיינים הבאים:

- 4 way associative
- גודל שדה ה- tag הינו 12 סיביות
- יש 8192 שורות במטמון
- הכתובות במבנה של 32 סיביות

שאלה 10:

על סמך המידע הנתון לנו על המטמון, מה גודל הנתונים במטמון?

א. לא ניתן לחשב זאת על בסיס נתוני השאלה

ב. במטמון זה יש 2^{20} (1,048,576) מילים של נתונים

ג. במטמון זה יש 2^{18} (262,144) מילים של נתונים

ד. במטמון זה יש 2^{22} (4,194,304) מילים של נתונים

ה. במטמון זה יש 2^{21} (2,097,152) מילים של נתונים

שאלה 11:

במטמון הנתון לנו, ידוע כי הערך בשדה התג הוא: 0x101. בנוסף, נתון כי בעת ביצוע פקודת LB זוהה hit בשורה 4096 במטמון, בעת פניה לבית 2 במילה ה- 16 בבלוק.

על בסיס נתוני המטמון, ועל בסיס נתוני השאלה – מהי הכתובת בהקסה של ה Byte אליו ניגשת פקודת LB?

א. 0x10180042

ב. 0x10104096

ג. 0x10180050

ד. 0x80050101

ה. לא ניתן לדעת על בסיס נתוני השאלה

חלק ב – שאלות פתוחות

בחלק זה 2 שאלות. את התשובות לחלק זה יש לכתוב בשאלון הבחינה במקום המיועד

שאלה 12 (36 נקודות)

נתון קטע הקוד, שלהלן, באסמבלי של ה- MIPS:

Address	Code	Basic	Source
0xEFFF0004	0x012a4023	subu $8,$9,$10	1 subu $8 , $9 , $10
0xEFFF0008	0x8ce80008	lw $8,0x00000008($7)	2 lw $8 , 8($7)
0xEFFF000C	0xad480008	sw $8,0x00000008($10)	3 sw $8 , 8($10)
0xEFFF0010	0x00884820	add $9,$4,$8	4 add $9 , $4 , $8

נתון כי ערכי אוגרים 1-15, בתחילת ביצוע הקוד, הוא הערך 0xN00N000, כאשר N הוא מספר האוגר. כלומר:

- אוגר 1 שווה: 0x10010000
- אוגר 2 שווה: 0x20020000
- ...
- אוגר 10 שווה: 0xA00A0000
- ...
- אוגר 15 שווה: 0xF00F0000

בזיכרון, למעט זיכרון פקודות התוכנית (קטע הקוד – ושימו לב לכתובות קטע הקוד), כל **מילה (word)** בזיכרון – הערך הקיים בה הינו כתובת המילה בזיכרון (כלומר, כתובת הבית הראשון של כל מילה).

דוגמה 1: הערכים בזיכרון, בכתובת 0x10010000 ואילך יראו כך:

0x10010000				0x10010004				0x10010008				0x1001000C				0x10010010				.....
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14....

דוגמה 2: הערכים בזיכרון, בכתובת 0xF00F0000 ואילך, יראו כך:

0xF00F0000				0xF00F0004				0xF00F0008				0xF00F000C				0xF00F0010				.....
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14....

כמו כן, נתון שניתן לגשת לכל מרחב הזיכרון בפקודות SW/LW (כל עוד הכתובת מתחלקת ב- 4)

נתונים אלה תקפים לכל סעיפי השאלה

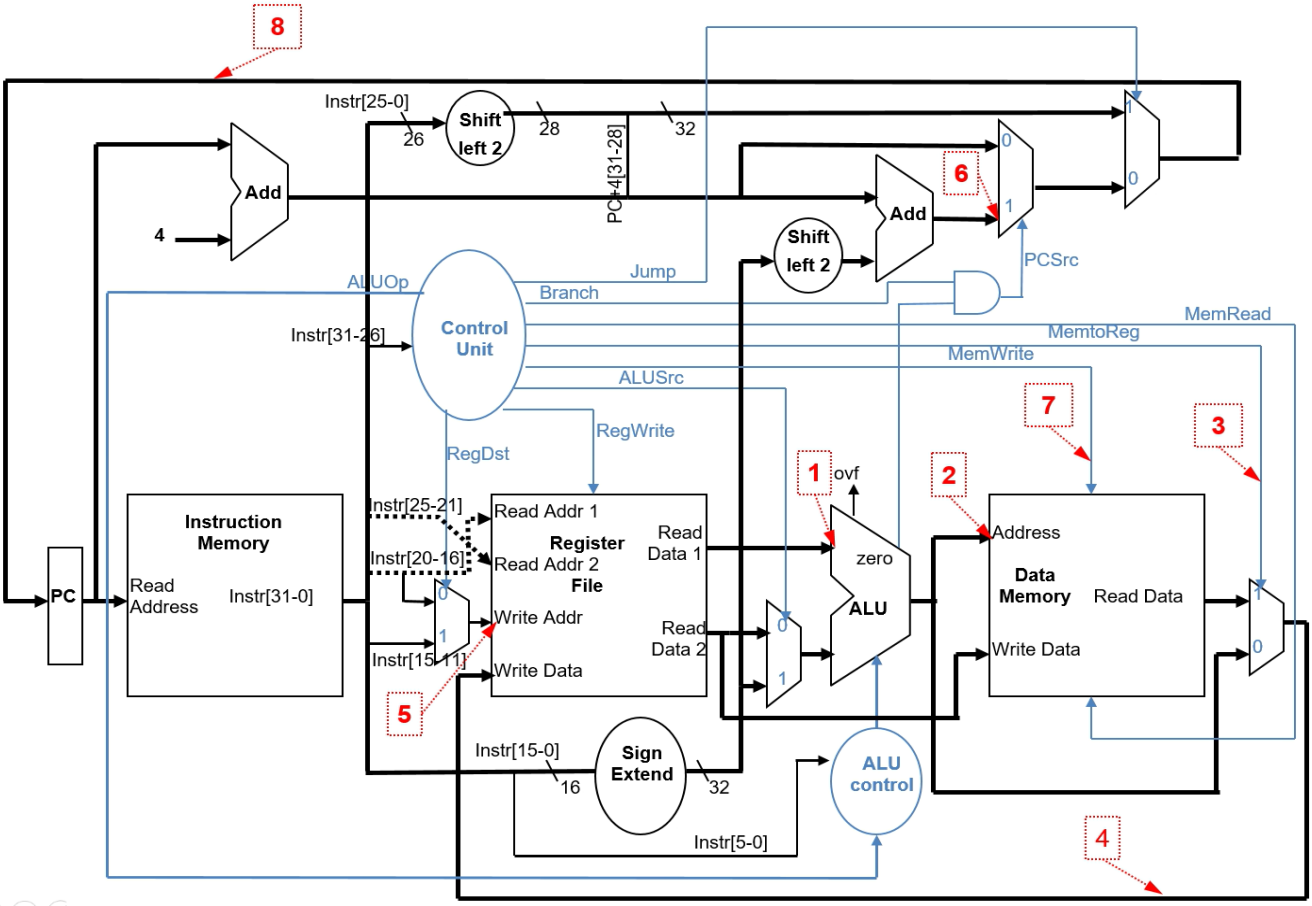
12.1. (16 נקודות, 2 נק' לכל סימון)

בתרשים שלהלן מעבד חד-מחזורי. בעת בניית המעבד, הטכנאי שחיווט את המעבד – טעה בחיווט וחיבר את סיביות 21-25 של הפקודה (Instr[21-25]) לכניסת Read Addr2 של מקבץ האוגרים, ואילו את סיביות 16-20 של הפקודה (Instr[16-20]) לכניסת Read Addr1 של מקבץ האוגרים.

שימו לב(!) החיווט של סיביות 16-20 אל ה- mux של RegDst **<u>בוצע נכון</u>** – וסיביות 16-20 של הפקודה נכנסות לכניסה 0 של ה mux.

כל הפקודות במעבד זה, ללא יוצא מהכלל, עובדות עם טעות זו של הטכנאי.

על התרשים סימונים של 8 נקודות. יש למלא את הטבלה, שבתחתית עמוד זה, עם הערכים העוברים על הקווים **<u>בבסיס הקסה</u>** בהנחה שהמידע נבדק לקראת סוף פעימת השעון בביצוע הפקודה השנייה (2) בקוד: **lw $8 , 8($7)**. במקרה של ערך לא ידוע, או שלא ניתן לדעת, יש לסמן X.



1	
2	
3	
4	

5	
6	
7	
8	

12.2. (16 נקודות, 2 נק' לכל סימון)

כעת, נריץ את קטע הקוד על מעבד MIPS **<u>העובד בטכנולוגית צנרת</u>** (על בסיס התרשים שלפניכם, המבוסס על תרשים 4.60). טעות החיווט שבוצעה במעבד החד מחזורי – מומשה גם בעת בניית מעבד הצנרת.

שימו לב: מעבר לטעות בחיווט למקבץ האוגרים (Register File) בוצע החיווט השגוי לנתוני מספרי אוגרים rs ו- rt (אוגרי המקור) הנרשמים באוגר הצנרת ID/EX. וודאו שאתם מזהים ומבינים את החיווט השגוי גם אל אוגר הצנרת ID/EX. החיווט IF/ID.Rt **לאוגר המטרה** לא נפגע ועובד תקין.

<u>השאלה:</u>

בפעימה זו, מחזור שעון 6 של רצף הפקודות הנתון, בשלב ה MEM (שלב 4) נמצאת בועה (כתוצאה מ load use). ניתן להניח שסיכוני נתונים בהרצת הקוד נפתרים על ידי יחידת: העברה קדימה (חיווט הכניסות למרבבים ForwardA, ForwardB כפי שמתואר בנספח לבחינה למקרים 1a, 1b, 2a, 2b), יחידת איתור סיכונים (HDU), וחציית מקבץ האוגרים.

על התרשים סימונים של 8 נקודות. יש למלא את הטבלה, שבתחילת העמוד הבא, עם הערכים העוברים על הקווים **<u>בבסיס הקסה</u>**. במקרה של ערך לא ידוע/לא ניתן לדעת - יש לסמן X.

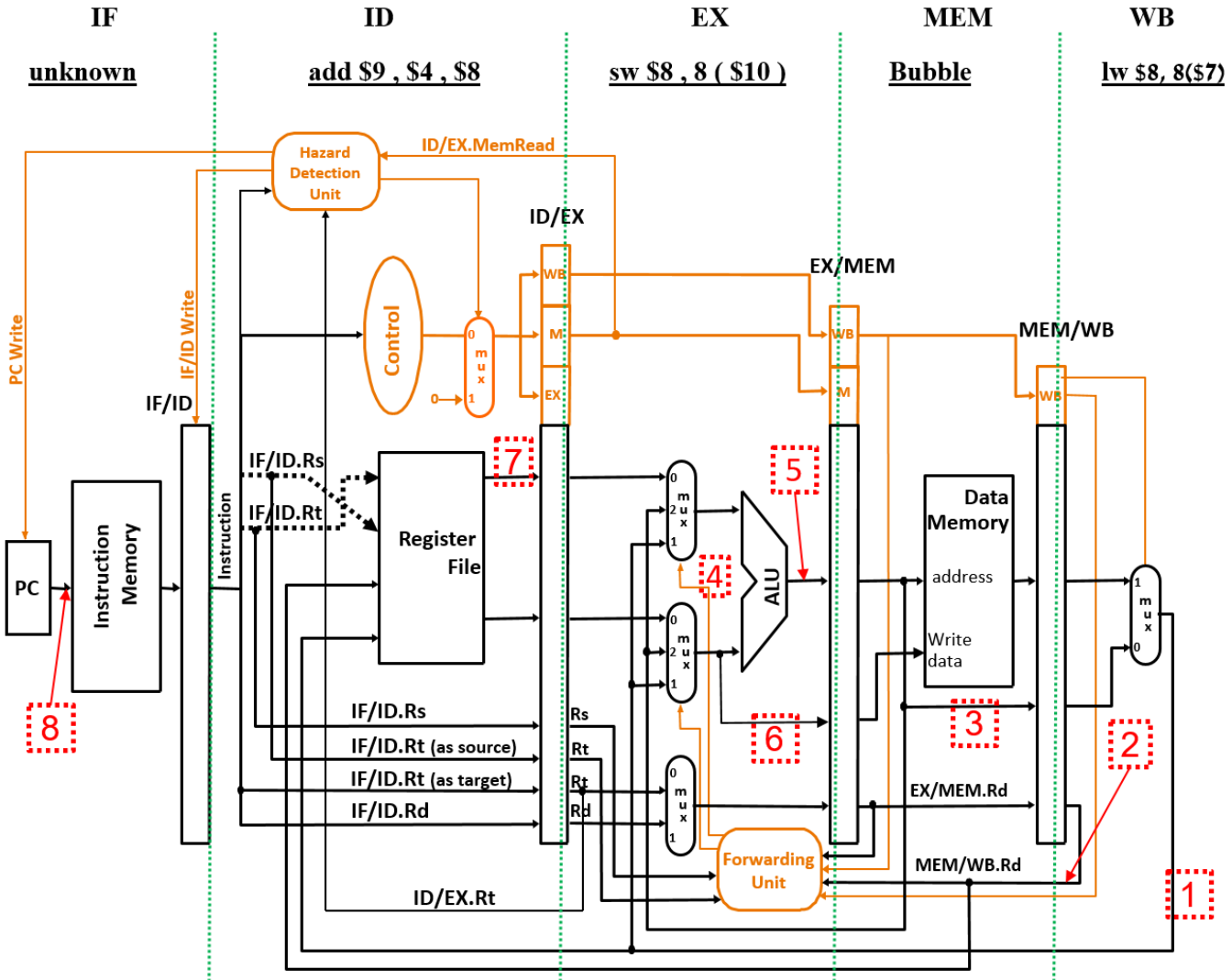


Figure 4.60

Clock Cycle 6

1	
2	
3	
4	

5	
6	
7	
8	

12.3. (4 נקודות)

לאור החיווט השגוי, איזה מהפקודות הבאות תעבודנה תקין, למרות החיווט השגוי. לאחר ציון הפקודות שתעבודנה תקין, יש להסביר (ב- 2-3 משפטים) מדוע פקודות אלה תעבודנה באופן תקין.

הפקודות: add , addi , sub , and , andi , or , ori , beq , slt , slti , lw , sw , j , jal

_____

_____

_____

_____

_____

_____

_____

_____

שאלה 13 (31 נקודות)

נתונה פרוצדורה בשם **whatDoIdo_X**. הפרוצדורה מקבלת 2 פרמטרים:

- $a0: כתובת תחילת מערך של מספרים (כל מספר בגודל מילה)
- $a1: N – מספר האיברים (ערכים) במערך (ה- length של המערך)

באזור הנתונים (.data) נתונה ההגדרה הבאה של המערך arr:

```
arr:  .word     0x503010A0, 0xB04090C0, 0x20806070
```

קוד הפרוצדורה:

```
whatDoIdo_X:
        addi    $t9 , $zero , 1
        sll     $t0 , $a1 , 2
        add     $t0 , $t0 , $a0
        addi    $t0 , $t0 , -4
mainLoop:
        beq     $a0 , $t0 , finish
        beq     $t9 , $zero , finish
        xor     $t9 , $t9 , $t9
        add     $t1 , $a0 , $zero
interLoop:
        bge     $t1 , $t0 , endInterLoop
        lw      $t2 , 0($t1)
        lw      $t3 , 4($t1)
        ble     $t2 , $t3, skip
        sw      $t3 , 0($t1)
        sw      $t2 , 4($t1)
        addi    $t9 , $zero , 1
skip:
        addi    $t1 , $t1 , 4
        j interLoop
endInterLoop:
        addi    $t0 , $t0 , -4
        j mainLoop
finish:
        lw      $v0 , 4($a0)
        jr $ra
```

ודאו שהבנתם את מימוש הפרוצדורה לפני שאתם עונים על השאלות הבאות

13.1. (15 נקודות)

מבוצעת קריאה לפרוצדורה whatDoIdo_X עם הפרמטרים הבאים:

- הפרמטר $a0 מכיל את כתובת מערך arr (כתובת תחילת המערך)
- הפרמטר $a1 מכיל את הערך 3

בסיום עבודת הפרוצדורה:

- איזה ערך יוחזר באוגר $v0?
- איך יראה המערך בזיכרון (**כ- 3 מילים/word** בהקסה!)

_____

_____

_____

_____

_____

כעת, בוצעו שינויים בפרוצדורה whatDoIdo. כל פקודה ששונתה מסומנת באופן מודגש (בשתי הגרסאות של הפרוצדורה)

קוד המקור (לנוחיות עבודה והשוואה)

```
whatDoIdo_X:
    addi   $t9 , $zero , 1
    sll    $t0 , $a1 , 2
    add    $t0 , $t0 , $a0
    addi   $t0 , $t0 , -4
mainLoop:
    beq    $a0 , $t0 , finish
    beq    $t9 , $zero , finish
    xor    $t9 , $t9 , $t9
    add    $t1 , $a0 , $zero
interLoop:
    bge    $t1 , $t0 , endInterLoop
    lw     $t2 , 0($t1)
    lw     $t3 , 4($t1)
    ble    $t2 , $t3, skip
    sw     $t3 , 0($t1)
    sw     $t2 , 4($t1)
    addi   $t9 , $zero , 1
skip:
    addi   $t1 , $t1 , 4
    j interLoop
endInterLoop:
    addi   $t0 , $t0 , -4
    j mainLoop
finish:
    lw     $v0 , 4($a0)
    jr $ra
```

הקוד עם השינויים – אותו יש לפענח

```
whatDoIdo_X:
  addi   $t9 , $zero , 1
  sll    $t0 , $a1 , 2   (הפקודה נמחקה)
  add    $t0 , $t0 , $a0
  addi   $t0 , $t0 , -1  (הפקודה שונתה)
mainLoop:
  beq    $a0 , $t0 , finish
  beq    $t9 , $zero , finish
  xor    $t9 , $t9 , $t9
  add    $t1 , $a0 , $zero
interLoop:
  bge    $t1 , $t0 , endInterLoop
  lbu    $t2 , 0($t1)   (הפקודה שונתה)
  lbu    $t3 , 4($t1)   (הפקודה שונתה)
  ble    $t2 , $t3, skip
  sb     $t3 , 0($t1)   (הפקודה שונתה)
  sb     $t2 , 4($t1)   (הפקודה שונתה)
  addi   $t9 , $zero , 1
skip:
  addi   $t1 , $t1 , 1   (הפקודה שונתה)
  j interLoop
endInterLoop:
  addi   $t0 , $t0 , -1  (הפקודה שונתה)
  j mainLoop
finish:
  lw     $v0 , 4($a0)
  jr $ra
```

13.2. (10 נקודות)

מבוצעת קריאה לפרוצדורה whatDoIdo_X, **לאחר השינוי**, עם הפרמטרים הבאים:

- הפרמטר $a0 מכיל את כתובת מערך arr
- הפרמטר $a1 מכיל את הערך **12**

בסיום עבודת הפרוצדורה:

- איזה ערך יוחזר באוגר $v0?
- איך יראה המערך בזיכרון (**כ- 3 מילים/word** בהקסה!)

שימו לב: נתוני המערך arr הינם הערכים המקוריים כפי שרשום בתחילת השאלה. אין קשר ואין תלות בין התשובה בסעיף זה לתשובה בסעיף הקודם.

_____

_____

_____

_____

_____

13.3. (6 נקודות)

מבוצעת קריאה לפרוצדורה whatDoIdo_X, **לאחר השינוי**, עם הפרמטרים הבאים:

- הפרמטר $a0 המכיל את כתובת מערך arr
- הפרמטר $a1 מכיל את הערך **6**

בסיום עבודת הפרוצדורה:

- איזה ערך יוחזר באוגר $v0?
- איך יראה המערך בזיכרון (**כ- 3 מילים/word** בהקסה!)

שימו לב: נתוני המערך arr הינם הערכים המקוריים כפי שרשום בתחילת השאלה. אין קשר ואין תלות בין התשובה בסעיף זה לתשובה בסעיפים הקודמים.

_____

_____

_____

_____

_____

טיוטא

פתרון X

חלק א: (33 נקודות)

בחלק זה 11 שאלות.

יש לבחור את התשובה הנכונה מבין האפשרויות.

משקל כל שאלה 3 נקודות.

יש לסמן X באופן ברור בטבלה שלפניכם

הערה: יש לסמן רק **אפשרות אחת** לכל שאלה!

לתשומת ליבכם: בדיקת התשובות מבוצעת רק בהתאם לכתוב בטבלה זו. הרישומים ליד השאלות עצמן לא נבדקים (ההתייחסות לרישומים אלה הינה כאל טיוטא)!

שאלה	א	ב	ג	ד	ה
1	X				
2			X		
3					X
4				X	
5	X				
6		X			
7		X			
8				X	
9					X
10		X			
11	X				

שאלה 1

נתון קטע הקוד הבא, שהינו תרגום של פקודת פסיאודו XXX:

Address	Code	Basic	Source
0x00400000	0x000a0fc3	sra $1, $10, 0x0000001f	XXX $t1, $t2
0x00400004	0x002a4826	xor $9, $1, $10	
0x00400008	0x01214823	subu $9, $9, $1	

נתון כי ערכו של אוגר $t2 במקרה א' - הוא (5-), ונתון כי ערכו של אוגר $t2 במקרה ב' הוא (10). מה יהיה ערכו של אוגר $t1 בסוף קטע קוד זה, בכל אחד המקרים, ומה מבצעת פקודת הפסיאודו XXX?

א. מקרה א': $t1 = 5, מקרה ב': $t1 = 10, הפקודה XXX מחשבת את הערך המוחלט

ב. מקרה א': $t1 = 5, מקרה ב': $t1 = -10, הפקודה XXX מחשבת את הערך ההופכי

ג. מקרה א': $t1 = -5, מקרה ב': $t1 = -10, הפקודה XXX מחשבת את הייצוג השלילי של $t1

ד. מקרה א': $t1, = -5, מקרה ב': $t1 = 10, הפקודה XXX, שומרת ב $t2 את ערכו של $t1

ה. לא ניתן להבין מקטע הקוד מה הוא מבצע

שאלה 2:

נתון קטע הקוד הבא:

```
        addi    $s0, $zero, -1
        addi    $t0, $zero, -1
        xor     $v0, $v0, $v0
loopX:
        addi    $v0, $v0, 1
        srl     $s0, $s0, 1
        and     $t0, $t0, $s0
        bne     $t0, $zero, loopX
```

כמה פעמים תתבצע הלולאה ומה יהיה ערכו של אוגר $v0 בסיום קטע הקוד

א. הלולאה תתבצע פעם אחת, וערכו של אוגר $v0 בסיום קטע הקוד יהיה 1

ב. הלולאה תתבצע 32 פעמים, וערכו של אוגר $v0 בסיום קטע הקוד יהיה 1

ג. הלולאה תתבצע 32 פעמים, וערכו של אוגר $v0 בסיום קטע הקוד יהיה 32

ד. הלולאה תתבצע פעם אחת, וערכו של אוגר $v0 בסיום קטע הקוד יהיה 32

ה. זו לולאה אין סופית, כי אוגר $t0 תמיד יהיה שונה מ- 0

מבנה ביטים:
00010001011010111111111111110011
לפי opcode זו פקודת BEQ לכן, לפי מבנה פקודת I:
000100 01011 01011 1111111111110011
ההשוואה בין בין אוגר 11 לאוגר 11 – כלומר תמיד מתקיים השוויון ותמיד תהיה קפיצה.
הקפיצה היא

שאלה 3

נתונה פקודת המכונה הבאה והכתובת בה היא נמצאת:

Address	Code
0x00400060	0x116bfff3

איזו פקודה זו, ומה הכתובת של הפקודה הבאה שתתבצע?

א. אין מספיק נתונים לדעת איזו פקודה זו, והפקודה הבאה לביצוע תהיה בכתובת 0x00400064

ב. זו פקודה R, והפקודה הבאה לביצוע תהיה בכתובת 0x00400064

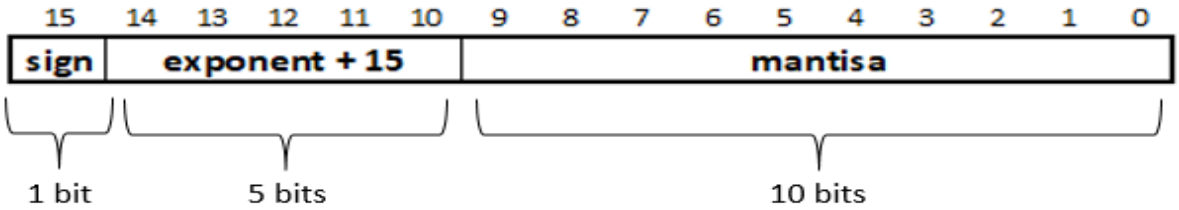
ג. זו פקודת Jump, והפקודה הבאה לביצוע תהיה בכתובת 0x0040002C

ד. זו פקודת BEQ, והפקודה הבאה לביצוע תהיה בכתובת 0x0040002C

ה. זו פקודת BEQ, והפקודה הבאה לביצוע תהיה בכתובת 0x00400030

יש להתייחס לנתון הבא בפתרון שאלות 4 ו-5:

במחשב העובד עם 16 סיביות/bits (במקום 32) נתון כי ייצוג מספרים ב- Floating Point הינו במבנה הבא:



שאלה 4

על בסיס הגדרת מבנה הייצוג ב Floating Point הנתון: כיצד נייצג, ב- 4 ספרות הקסה, את הערך (-35.75)?

$-35.75 = (-1) * (2^5 + 2^1 + 2^0 + 2^{-1} + 2^{-2}$
$= (-1) * [2^5 * (1 + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7})$
= 1 5 (1.)0001111
= 1 20 0001111
= 1 10100 0001111000

= 1101000001111000
= 1101 0000 0111 1000
= 0xD078

א. 0x5078

ב. 0x9038

ג. 0x9478

ד. 0xD078

ה. 0xD238

שאלה 5

על בסיס הגדרת הייצוג ב Floating Point הנתון, מהו הערך העשרוני של **0x4EA0**?

0x4EA0 = 0100 1110 1010 0000
→ 0 10011 1010100000
→ 0 10011 (1.) 1010100000
→ + 19 $1 + 2^{-1} + 2^{-3} + 2^{-5}$
→ + 4 $1 + 2^{-1} + 2^{-3} + 2^{-5}$
→ + 2^4 * $(1 + 2^{-1} + 2^{-3} + 2^{-5})$
→ $+ 2^4 + 2^3 + 2^1 + 2^{-1}$
→ 16 + 8 + 2 + 0.5 = 26. 5

א. 26.5

ב. 26.05

ג. 16.5

ד. 16.05

ה. 0.2605

שאלה 6:

ביצוע Push, לאוגר $x, למחסנית מבוצע באמצעות 2 הפקודות הבאות:

```
addi  $sp, $sp, -4
sw    $x, 0($sp)
```

הוחלט לפתח פקודת MIPS בסיסית לביצוע push – ובמקום לרשום 2 פקודות אלה, ניתן יהיה לרשום פקודה אחת, אשר תבצע בפקודה אחת את שביצעו 2 הפקודות הקודמות: **push $x**

נתונה תוכנית אשר, לפני פיתוח הפקודה החדשה, 15% מהפקודות הן פקודות עבור ביצוע ה push (כלומר, 85% "שאר התוכנית" ו- 15% זה זוג הפקודות לביצוע ה push).

נתונים נוספים על המעבד/תוכנית:

- CPI של פקודות אריתמטיות (כולל addi) הוא 2
- CPI של פקודות sw הוא 4
- CPI של פקודת ה **push החדשה** הוא 3

מהירות המעבד הינה 4Ggz

על בסיס נתונים אלה, מהו מדד ה- speedup בעקבות שינוי זה?

- על בסיס הנתונים ביצוע push באמצעות 2 פקודות, ה CPI המצטבר של 2 הפקודות ביחד הוא 6
- ה CPI של פקודת ה PUSH החדשה הוא 3
- מכאן שה speedup לביצוע PUSH הוא 2.
- נשתמש בחישוב speedup על בסיס "חוק אמדל" ונקבל:

Speedup = 1 / [(1-0.15) + (0.15 / 2)]
= 1 / (0.85 + 0.075)

א. לא ניתן לחשב מאחר ולא נתונה כמות הפקודות (IC) ולא נתון CPI ממוצע של שאר הפקודות

ב. 1.081

ג. 1.111

ד. 1.0526

ה. 1.095

שאלה 7:

נתונים הערכים הבאים באוגרים $s1 , $s0:

$s1 = 0x00000001 $s0 = 0xFFFFFFFF

נתונות 2 הפקודות הבאות:

```
slt    $t0 , $s0 , $s1
sltu   $t1 , $s0 , $s1
```

בהתאם ל- 2 פקודות אלה, מה יהיו הערכים באוגרים $t0 ו- $t1:

א. $t0 = 0, $t1 = 0

ב. $t0 = 1, $t1 = 0

ג. $t0 = 0, $t1 = 1

ד. $t0 = 1, $t1 = 1

ה. לא ניתן לדעת על בסיס הנתונים לשאלה

ערכו של $t0 יהיה 1: slt מתייחס לערכים כאל ערכים עם סימן, ולכן ההתייחסות לערכו של אוגר $s0 הינה כאל הערך (-1), ומאחר וזה קטן מ- 1 – תוצאת הפקודה הינה 1

ערכו של $t1 יהיה 0: sltu מתייחס לערכים כאל ערכים ללא סימן, ולכן ההתייחסות לערכו של אוגר $s0 הינה כאל הערך 4,294,967,295 (2^{32}-1), ומאחר וזה גדול מ- 1 – תוצאת הפקודה הינה 0

שאלה 8:

במהלך ביצוע פקודת LW, מתרחש אירוע Page Fault בעת הגישה לזיכרון הנתונים.

בהתבסס על תרשים 4.66, איזה מהמשפטים הבאים **אינו נכון**?

א. אוגר PC יקבל את הערך 0x80000180, הכתובת של ה exception handler

ב. התקלה מתרחשת בשלב ה- MEM, ולכן יבוצע flush לאוגר MEM/WB כדי למנוע מהפקודה שגרמה לתקלה להמשיך בצנרת.

ג. יבוצע flush לאוגרים: IF/ID, ID/EX ו- EX/MEM כדי למנוע מהפקודות שאחרי הפקודה שגרמה לתקלה להמשיך בצנרת

ד. דגל RegWrite יאופס, כדי למנוע כתיבה שגויה למקבץ האוגרים על ידי הפקודה שבשלב WB

ה. אוגר EPC יעודכן בכתובת הפקודה שגרמה לתקלה, ואוגר cause יעודכן בקוד התקלה

שאלה 9:

במהלך הקורס התייחסנו לשלוש דרכים לפתרון Control Hazard של פקודות Branch. בהתבסס על מה שלמדנו, איזה מהמשפטים הבאים **אינו נכון**?

א. הקדמת בדיקת ה branch לשלב ה- ID (תרשים 4.65) מאפשרת זיהוי מוקדם לצורך לדלג, צמצום הבעיה ל nop אחד, במחיר של data hazard כבר בשלב ID.

ב. במצב של Branch Delay Slot החומרה תמיד מבצעת את הפקודה שאחרי ה branch, כאשר באחריות התוכנה לדאוג לשים פקודה מתאימה. זו מוסכמה של תוכנה עם החומרה. באמצעות אופטימייזר, בכ- 50% מהמקרים ניתן למצוא פקודה שניתן למקם אותה אחרי פקודת ה Branch, וכך ולהימנע מ- nop.

ג. באמצעות שיטת חיזוי דינאמי בחומרה, נשלב יחידת BTB שבאופן שוטף תעבוד לפי תחזית דילוג לפקודת branch, ובמידה ותהיה החמצה, באמצעות יחידת Mispredict Detection Unit תזוהה ההחמצה ויהיה טיפול בניקוי הפקודות השגויות מהצנרת ותיקון התחזית.

ד. יחידת ה Mispredict Detection Unit מקבלת את נתוני "כיוון התחזית" לפיה עבדה יחידת ה BTB בשלב ה IF, מקבלת את נתוני ה Branch האמיתיים כפי שחושבו בשלב ה MEM, ובהתאם יכולה לזהות האם הייתה החמצת תחזית או שלא. במידה והייתה החמצה, יבוצע ניקוי של 3 פקודות מהצנרת, עדכון אוגר PC בפקודה הנכונה שצריכה הייתה להיכנס לצנרת, ועדכון של יחידת ה BTB בתחזית העדכנית.

ה. מצב ה Branch Delay Slot בשילוב הקדמת הבדיקה לשלב ה- ID הינו המנגנון היעיל והטוב ביותר בהיבט הביצועים, מאחר ורק ב- 50% מהמקרים נדרש לבצע תיקון עם nop יחיד. בזמן שבאמצעות חיזוי דינאמי, תמיד יש ניקוי של 3 פקודות באמצעות nop/bubble כי הזיהוי מבוצע רק בשלב ה MEM, ולכן חיזוי דינאמי פחות יעיל מ Branch Delay Slot.

יש להתייחס לנתונים הבאים בשאלות 10-11

נתון זיכרון מטמון עם המאפיינים הבאים:

- 4 way associative
- גודל שדה ה- tag הינו 12 סיביות
- יש 8192 שורות במטמון
- הכתובות במבנה של 32 סיביות

על בסיס מספר השורות – גודל שדה index הוא 13 (2^{13}=8192)
נתון כי גודל שדה ה tag הוא 12
כלומר, Block Offset + Byte Offset יש 7 סיביות, כלומר, יש 5 סיביות ב block offset – כלומר גודל בלוק הוא 32 מילים (128 בתים)

שאלה 10:

על סמך המידע הנתון לנו על המטמון, מה גודל הנתונים במטמון?

א. לא ניתן לחשב זאת על בסיס נתוני השאלה

ב. במטמון זה יש 2^{20} (1,048,576) מילים של נתונים

ג. במטמון זה יש 2^{18} (262,144) מילים של נתונים

ד. במטמון זה יש 2^{22} (4,194,304) מילים של נתונים

ה. במטמון זה יש 2^{21} (2,097,152) מילים של נתונים

יש 32 מילים בבלוק (2^5)
יש 2^{13} שורות
המטמון הוא 4 ways (2^2)
ומכאן, שגודל הנתונים במטמון, **במילים** הוא:
$2^2 * 2^{13} * 2^5 = 2^{20}$

שאלה 11:

במטמון הנתון לנו, ידוע כי הערך בשדה התג הוא: 0x101. בנוסף, נתון כי בעת ביצוע פקודת LB זוהה hit בשורה 4096 במטמון, בעת פניה לבית 2 במילה ה- 16 בבלוק.

על בסיס נתוני המטמון, ועל בסיס נתוני השאלה – מהי הכתובת בהקסה של ה Byte אליו ניגשת פקודת LB?

א. 0x10180042

ב. 0x10104096

ג. 0x10180050

ד. 0x80050101

ה. לא ניתן לדעת על בסיס נתוני השאלה

נתון Tag – המשמעות – אלה 12 הסיביות הגבוהות בכתובת
שורה 4096 במטמון: ב- 13 סיביות: 1000000000000
מילה 16 בבלוק: ב- 5 סיביות: 10000
בית 2 במילה: ב- 2 סיביות: 10
$00010000000110000000000001000010_{bin}$
$0001\ 0000\ 0001\ 1000\ 0000\ 0000\ 0100\ 0010_{bin}$
0x10180042

אפקה
המכללה האקדמית להנדסה בתל אביב

פתרון X

חלק ב – שאלות פתוחות

בחלק זה 2 שאלות. את התשובות לחלק זה יש לכתוב בשאלון הבחינה במקום המיועד

שאלה 12 (36 נקודות)

נתון קטע הקוד, שלהלן, באסמבלי של ה- MIPS:

Address	Code	Basic	Source
0xEFFF0004	0x012a4023	subu $8,$9,$10	1 subu $8 , $9 , $10
0xEFFF0008	0x8ce80008	lw $8,0x00000008($7)	2 lw $8 , 8($7)
0xEFFF000C	0xad480008	sw $8,0x00000008($10)	3 sw $8 , 8($10)
0xEFFF0010	0x00884820	add $9,$4,$8	4 add $9 , $4 , $8

נתון כי ערכי אוגרים 1-15, בתחילת ביצוע הקוד, הוא הערך 0xN00N000, כאשר N הוא מספר האוגר. כלומר:

- אוגר 1 שווה: 0x10010000
- אוגר 2 שווה: 0x20020000
- ...
- אוגר 10 שווה: 0xA00A0000
- ...
- אוגר 15 שווה: 0xF00F0000

בזיכרון, למעט זיכרון פקודות התוכנית (קטע הקוד – ושימו לב לכתובות קטע הקוד), כל **מילה (word)** בזיכרון – הערך הקיים בה הינו כתובת המילה בזיכרון (כלומר, כתובת הבית הראשון של כל מילה).

דוגמה 1: הערכים בזיכרון, בכתובת 0x10010000 ואילך יראו כך:

0x10010000				0x10010004				0x10010008				0x1001000C				0x10010010				.....
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14....

דוגמה 2: הערכים בזיכרון, בכתובת 0xF00F0000 ואילך, יראו כך:

0xF00F0000				0xF00F0004				0xF00F0008				0xF00F000C				0xF00F0010				.....
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14....

כמו כן, נתון שניתן לגשת לכל מרחב הזיכרון בפקודות SW/LW (כל עוד הכתובת מתחלקת ב- 4)

נתונים אלה תקפים לכל סעיפי השאלה

מעקב אחר ביצוע הקוד הנ"ל (בהתייחסות לחיווט השגוי כפי שמתואר בהמשך השאלה):

אחרי פקודה (1): אוגר 0x10010000 = $8 (בפועל מבוצע החיסור: $9-$10 = $8)

אחרי פקודה (2): אוגר 0x1001008 = $8 (בפועל, ערכו של אוגר $8 שהוא אוגר ה- rt, מגיע לנתוני read data1, ולכן חישוב הכתובת מבוצע לפי ערכו של אוגר $8, לאחר פקודה 1, בתוספת הערך המיידי – 8).

פקודה (3): כמו בפקודה (2), הכתובת אליה פונים מחושבת למעשה על בסיס ערכו של אוגר $8 בתוספת הערך המיידי 8 – כלומר, פונים לכתובת 0x10010010. הערך שייכתב בכתובת זו, הינו הערך שנמצא באוגר $10, שעקב החיווט השגוי הינו אוגר $10 שערכו הוא 0xA00A0000

פקודה (4): תבצע חיבור בין ערכי האוגרים $8 ו- $4, כך שהערך הצפוי להיכנס לאוגר $9 הינו תוצאת החיבור של אוגר $8 לאחר פקודה (2) בתוספת ערכו של אוגר 4 שהינו 0x40040000, כלומר 0x50050008

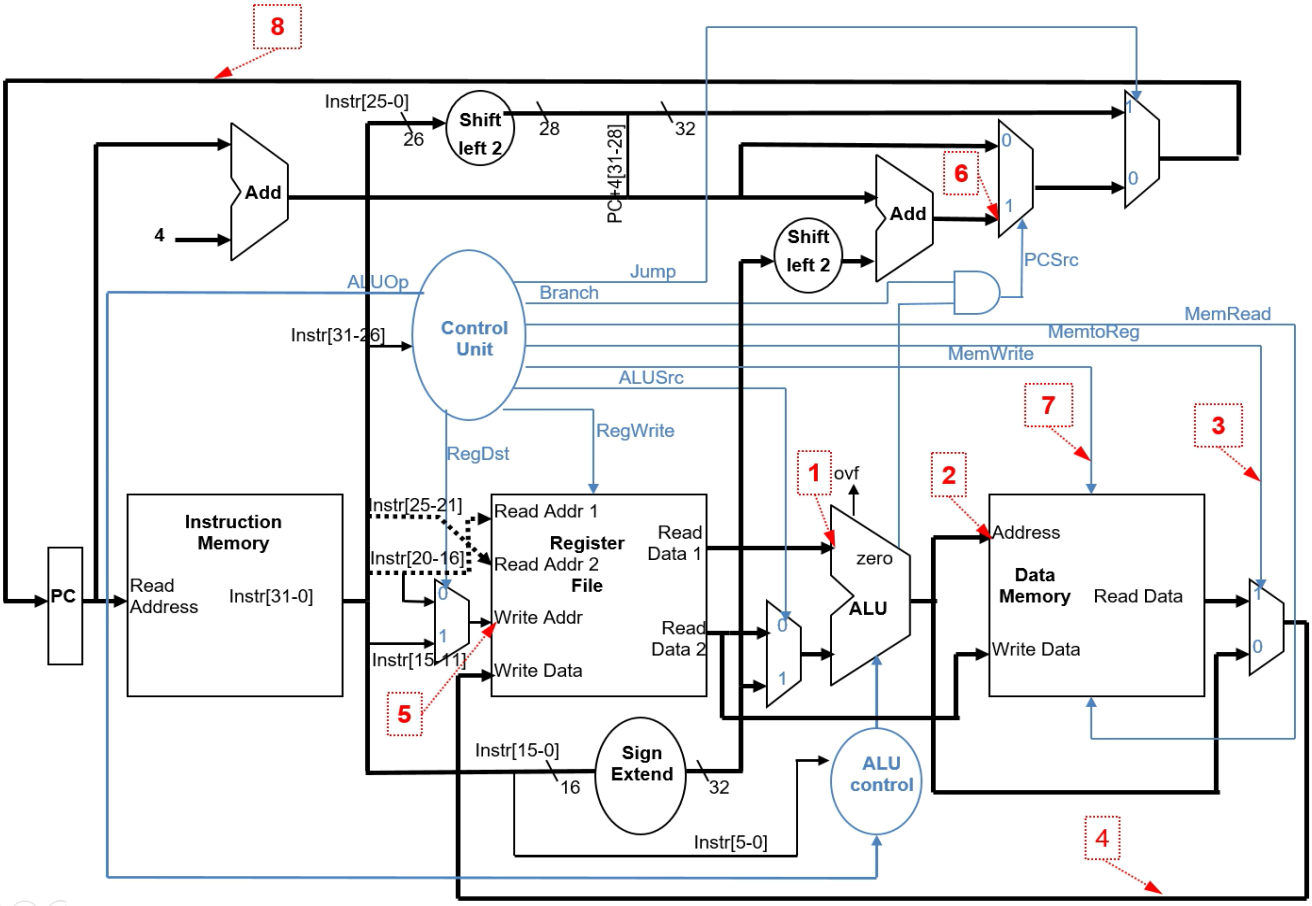
12.1. (16 נקודות, 2 נק' לכל סימון)

בתרשים שלהלן מעבד חד-מחזורי. בעת בניית המעבד, הטכנאי שחיווט את המעבד – טעה בחיווט וחיבר את סיביות 21-25 של הפקודה (Instr[21-25]) לכניסת Read Addr2 של מקבץ האוגרים, ואילו את סיביות 16-20 של הפקודה (Instr[16-20]) לכניסת Read Addr1 של מקבץ האוגרים.

שימו לב(!) החיווט של סיביות 16-20 אל ה- mux של RegDst **<u>בוצע נכון</u>** – וסיביות 16-20 של הפקודה נכנסות לכניסה 0 של ה mux.

כל הפקודות במעבד זה, ללא יוצא מהכלל, עובדות עם טעות זו של הטכנאי.

על התרשים סימונים של 8 נקודות. יש למלא את הטבלה, שבתחתית עמוד זה, עם הערכים העוברים על הקווים **<u>בבסיס הקסה</u>** בהנחה שהמידע נבדק לקראת סוף פעימת השעון בביצוע הפקודה השנייה (2) בקוד: **lw $8 , 8($7)**. במקרה של ערך לא ידוע, או שלא ניתן לדעת, יש לסמן X.



1	0x10010000 ערך אוגר 8 לאחר פקודה (1)
2	0x10010008 (חישוב כתובת ממנה מבוצע ה- LW)
3	memToReg = 1, להעברת ערך התוצאה מהזיכרון
4	0x10010008 – ערך המילה בכתובת (נקודה 2)

5	0x8 – מספר אוגר המטרה בפקודה
6	0xEFFF002C (חישוב כתובת דילוג של פקודת branch: 0xFFF0008+4+8*4)
7	memWrite = 0, זו פקודת LW ודגל זה הוא 0 בפקודה זו.
8	0xEFFF000C – כתובת הפקודה שאחרי פקודת ה LW.

12.2. (16 נקודות, 2 נק' לכל סימון)

כעת, נריץ את קטע הקוד על מעבד MIPS **<u>העובד בטכנולוגית צנרת</u>** (על בסיס התרשים שלפניכם, המבוסס על תרשים 4.60). טעות החיווט שבוצעה במעבד החד מחזורי – מומשה גם בעת בניית מעבד הצנרת.

שימו לב: מעבר לטעות בחיווט למקבץ האוגרים (Register File) בוצע החיווט השגוי לנתוני מספרי אוגרים rs ו- rt (אוגרי המקור) הנרשמים באוגר הצנרת ID/EX. וודאו שאתם מזהים ומבינים את החיווט השגוי גם אל אוגר הצנרת ID/EX. החיווט IF/ID.Rt **לאוגר המטרה** לא נפגע ועובד תקין.

<u>השאלה:</u>

בפעימה זו, מחזור שעון 6 של רצף הפקודות הנתון, בשלב ה MEM (שלב 4) נמצאת בועה (כתוצאה מ load use). ניתן להניח שסיכוני נתונים בהרצת הקוד נפתרים על ידי יחידת: העברה קדימה (חיווט הכניסות למרבבים ForwardA, ForwardB כפי שמתואר בנספח לבחינה למקרים 1a, 1b, 2a, 2b), יחידת איתור סיכונים (HDU), וחציית מקבץ האוגרים.

על התרשים סימונים של 8 נקודות. יש למלא את הטבלה, שבתחילת העמוד הבא, עם הערכים העוברים על הקווים **<u>בבסיס הקסה</u>**. במקרה של ערך לא ידוע/לא ניתן לדעת - יש לסמן X.

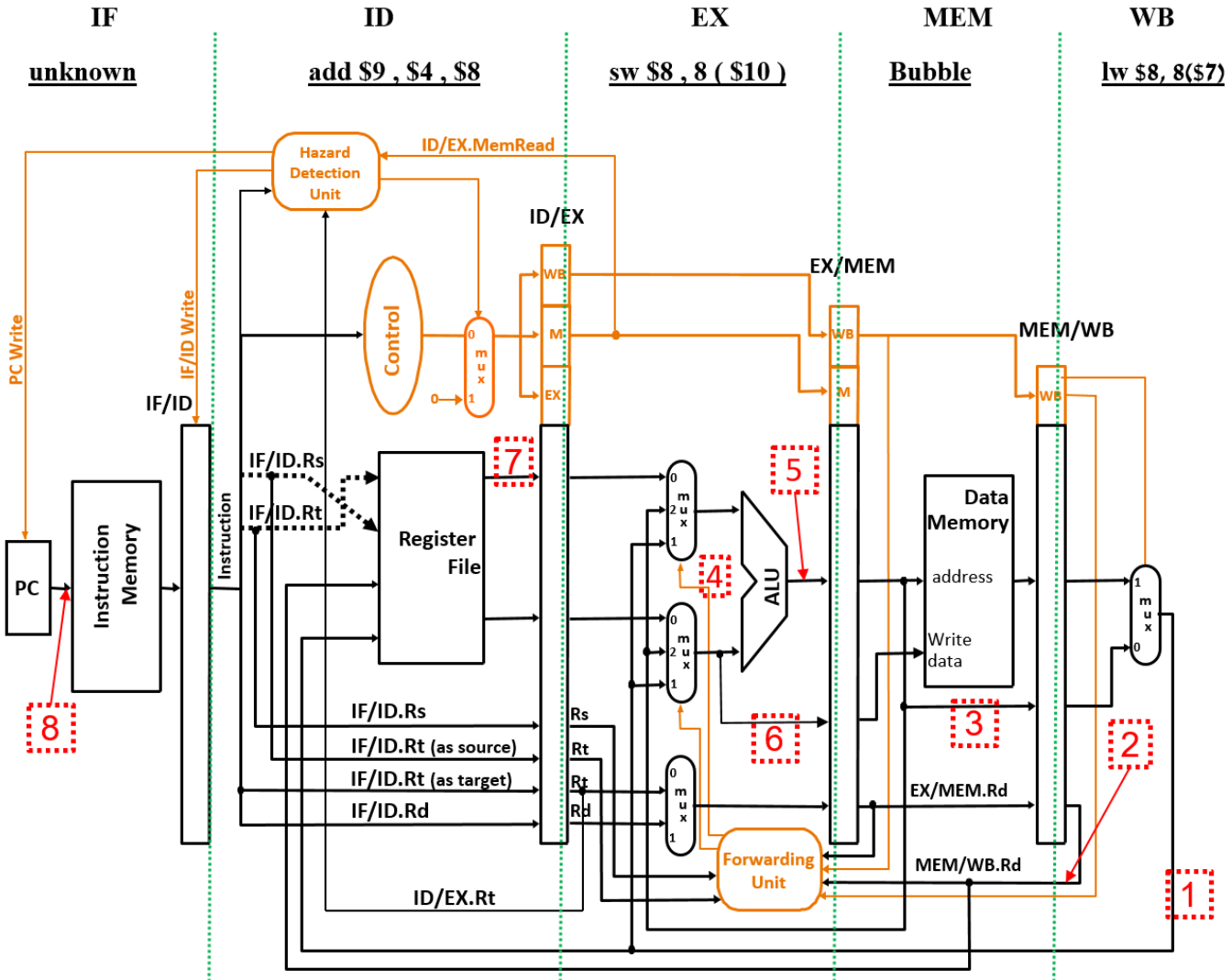


Figure 4.60

Clock Cycle 6

1	0x10010008 – תוצאת load לאוגר תוצאה 8
2	0x8 – מספר אוגר המטרה של פקודת lw
3	0x10010008 – זו אמנם פקודת bubble אך ביצוע חישוב הכתובת על בסיס ערך אוגר $8 כפי שחושב בפקודת subu הראשונה)
4	0x1 – מבוצע forward 2a לאוגר 8 שבפועל הוא אוגר rs, עקב בעיית החיווט
5	0x10010010 – הכתובת שפקודת sw תכתוב לתוכה ערך 0xA00A0000 – ערכו של אוגר $10,
6	שבפועל הוא אוגר rs, עקב בעיית החיווט, וערכו הוא זה שירשם בזיכרון בפקודת SW
7	0x10010008 – ערכו של אוגר $8, לאחר חציית מקבץ האוגרים (ולאחר ההחלפה עקב סוגיית החיווט)
8	0xEFFF0014 – כתובת הפקודה שלאחר פקודת ה ADD

12.3. (4 נקודות)

לאור החיווט השגוי, איזה מהפקודות הבאות תעבודנה תקין, למרות החיווט השגוי. לאחר ציון הפקודות שתעבודנה תקין, יש להסביר (ב- 2-3 משפטים) מדוע פקודות אלה תעבודנה באופן תקין.

הפקודות: add , addi , sub , and , andi , or , ori , beq , slt , slti , lw , sw , j , jal

הפקודות שתעבודנה תקין הן פקודות "האדישות" לסדר של אוגרים rs ו- rt:

- add , and , or , beq

או שלחלופין, לא מתבססות על תוכן אוגרים אלה:

- פקודות j , jal

פקודות ה R שצוינו – הן פקודות שאין משמעות לאיזה אוגר רשום כ rs ואיזה אוגר רשום כ rt, כי סדר האוגרים תחליפי – והתוצאה זהה.

פקודות ה jump לא מתבססות כלל על תכני האוגרים ולכן הן תעבודנה גם כן באופן תקין

שאלה 13 (31 נקודות)

נתונה פרוצדורה בשם **whatDoIdo_X**. הפרוצדורה מקבלת 2 פרמטרים:

- $a0: כתובת תחילת מערך של מספרים (כל מספר בגודל מילה)
- $a1: N – מספר האיברים (ערכים) במערך (ה- length של המערך)

באזור הנתונים (.data) נתונה ההגדרה הבאה של המערך arr:

```
arr:  .word     0x503010A0, 0xB04090C0, 0x20806070
```

קוד הפרוצדורה:

```
whatDoIdo_X:
        addi    $t9 , $zero , 1
        sll     $t0 , $a1 , 2
        add     $t0 , $t0 , $a0
        addi    $t0 , $t0 , -4
mainLoop:
        beq     $a0 , $t0 , finish
        beq     $t9 , $zero , finish
        xor     $t9 , $t9 , $t9
        add     $t1 , $a0 , $zero
interLoop:
        bge     $t1 , $t0 , endInterLoop
        lw      $t2 , 0($t1)
        lw      $t3 , 4($t1)
        ble     $t2 , $t3, skip
        sw      $t3 , 0($t1)
        sw      $t2 , 4($t1)
        addi    $t9 , $zero , 1
skip:
        addi    $t1 , $t1 , 4
        j interLoop
endInterLoop:
        addi    $t0 , $t0 , -4
        j mainLoop
finish:
        lw      $v0 , 4($a0)
        jr $ra
```

ודאו שהבנתם את מימוש הפרוצדורה לפני שאתם עונים על השאלות הבאות

13.1. (15 נקודות)

מבוצעת קריאה לפרוצדורה whatDoIdo_X עם הפרמטרים הבאים:

- הפרמטר $a0 מכיל את כתובת מערך arr (כתובת תחילת המערך)
- הפרמטר $a1 מכיל את הערך 3

בסיום עבודת הפרוצדורה:

- איזה ערך יוחזר באוגר $v0?
- איך יראה המערך בזיכרון (**כ- 3 מילים/word** בהקסה!)

$v0 = 0x20806070 (ערך המילה השנייה במערך, לאחר המיון)

Arr: 0xb04090c0, 0x20806070, 0x503010a0

שימו לב: הערך הראשון מתחיל עם הקסה b, כלומר זהו ערך שלילי, ולכן הוא הקטן ביותר ומופיע ראשון לאחר המיון

כעת, בוצעו שינויים בפרוצדורה whatDoIdo. כל פקודה ששונתה מסומנת באופן מודגש (בשתי הגרסאות של הפרוצדורה)

קוד המקור (לנוחיות עבודה והשוואה)

```
whatDoIdo_X:
    addi   $t9 , $zero , 1
    sll    $t0 , $a1 , 2
    add    $t0 , $t0 , $a0
    addi   $t0 , $t0 , -4
mainLoop:
    beq    $a0 , $t0 , finish
    beq    $t9 , $zero , finish
    xor    $t9 , $t9 , $t9
    add    $t1 , $a0 , $zero
interLoop:
    bge    $t1 , $t0 , endInterLoop
    lw     $t2 , 0($t1)
    lw     $t3 , 4($t1)
    ble    $t2 , $t3, skip
    sw     $t3 , 0($t1)
    sw     $t2 , 4($t1)
    addi   $t9 , $zero , 1
skip:
    addi   $t1 , $t1 , 4
    j interLoop
endInterLoop:
    addi   $t0 , $t0 , -4
    j mainLoop
finish:
    lw     $v0 , 4($a0)
    jr $ra
```

הקוד עם השינויים – אותו יש לפענח

```
whatDoIdo_X:
  addi   $t9 , $zero , 1
  ~~sll    $t0 , $a1 , 2~~   (הפקודה נמחקה)
  add    $t0 , $t0 , $a0
  addi   $t0 , $t0 , -1   (הפקודה שונתה)
mainLoop:
  beq    $a0 , $t0 , finish
  beq    $t9 , $zero , finish
  xor    $t9 , $t9 , $t9
  add    $t1 , $a0 , $zero
interLoop:
  bge    $t1 , $t0 , endInterLoop
  lbu    $t2 , 0($t1)   (הפקודה שונתה)
  lbu    $t3 , 4($t1)   (הפקודה שונתה)
  ble    $t2 , $t3, skip
  sb     $t3 , 0($t1)   (הפקודה שונתה)
  sb     $t2 , 4($t1)   (הפקודה שונתה)
  addi   $t9 , $zero , 1
skip:
  addi   $t1 , $t1 , 1   (הפקודה שונתה)
  j interLoop
endInterLoop:
  addi   $t0 , $t0 , -1   (הפקודה שונתה)
  j mainLoop
finish:
  lw     $v0 , 4($a0)
  jr $ra
```

13.2. (10 נקודות)

מבוצעת קריאה לפרוצדורה whatDoIdo_X, **לאחר השינוי**, עם הפרמטרים הבאים:

- הפרמטר $a0 מכיל את כתובת מערך arr
- הפרמטר $a1 מכיל את הערך **12**

בסיום עבודת הפרוצדורה:

- איזה ערך יוחזר באוגר $v0?
- איך יראה המערך בזיכרון (**כ- 3 מילים/word** בהקסה!)

שימו לב: נתוני המערך arr הינם הערכים המקוריים כפי שרשום בתחילת השאלה. אין קשר ואין תלות בין התשובה בסעיף זה לתשובה בסעיף הקודם.

$v0 = 0x80706050

Arr: 0x40302010, 0x80706050, 0xc0b0a090

שימו לב: המיון בוצע ברמת בית בודד, אך במבנה של word (זוכרים little endian order?) זה יראה כך. בנוסף, המיון ברמת הבתים הינו ללא סימן בפועל, מאחר ומבוצע load באמצעות פקודת lbu

13.3. (6 נקודות)

מבוצעת קריאה לפרוצדורה whatDoIdo_X, **לאחר השינוי**, עם הפרמטרים הבאים:

- הפרמטר $a0 המכיל את כתובת מערך arr
- הפרמטר $a1 מכיל את הערך **6**

בסיום עבודת הפרוצדורה:

- איזה ערך יוחזר באוגר $v0?
- איך יראה המערך בזיכרון (**כ- 3 מילים/word** בהקסה!)

שימו לב: נתוני המערך arr הינם הערכים המקוריים כפי שרשום בתחילת השאלה. אין קשר ואין תלות בין התשובה בסעיף זה לתשובה בסעיפים הקודמים.

$v0 = 0xb040c0a0

Arr: 0x90503010 , 0xb040c0a0 , 0x20806070

שימו לב: המיון בוצע ברמת בית בודד, אך במבנה של word (זוכרים little endian order?) זה יראה כך. בנוסף, המיון ברמת הבתים הינו ללא סימן בפועל, מאחר ומבוצעload באמצעות פקודת lbu. בפועל, ממין רק את 6 **הבתים** הראשונים במערך. ולכן, 2 הבתים הגבוהים בערך השני (0xb040 . בנוסף, ההשוואה בפועל הינה ללא סימן, מאחר ומבוצעת פקודת lbu