

**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 טיוטה)!**

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

## שאלה 1

נתון קטע הקוד הבא:

Address	Code	Basic	Source
0x00400010	0x2008ffff	addi \$8,\$0,0xffffffff	1 addi \$t0, \$zero, -1
0x00400014	0x3c01ffff	lui \$1,0xffffffff	2 xori \$t1, \$t2, 0xffffffff
0x00400018	0x3421ffff	ori \$1,\$1,0x0000ffff	
0x0040001c	0x01414826	xor \$9,\$10,\$1	
0x00400020	0x01284822	sub \$9,\$9,\$8	3 sub \$t1, \$t1, \$t0

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 מבצע קטע הקוד הנתון?

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, 3
addi $s1, $zero, -1
add $t0, $zero, $zero
xor $v0, $v0, $v0
loopY:
    addi $v0, $v0, 2
    or $t0, $t0, $s0
    sll $s0, $s0, 2
    bne $t0, $s1, loopY

```

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

א. זו לולאה אין סופית, כי אוגר \$t0 אף פעם לא יהיה שווה לאוגר \$s1

ב. הלולאה תתבצע פעם אחת, וערכו של אוגר \$v0 בסיום קטע הקוד יהיה 2

ג. הלולאה תתבצע 32 פעמים, וערכו של אוגר \$v0 בסיום קטע הקוד יהיה 16

ד. הלולאה תתבצע 16 פעמים, וערכו של אוגר \$v0 בסיום קטע הקוד יהיה 32

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

### שאלה 3

נתונה פקודת המכונה הבאה והכתובת בה היא נמצאת:

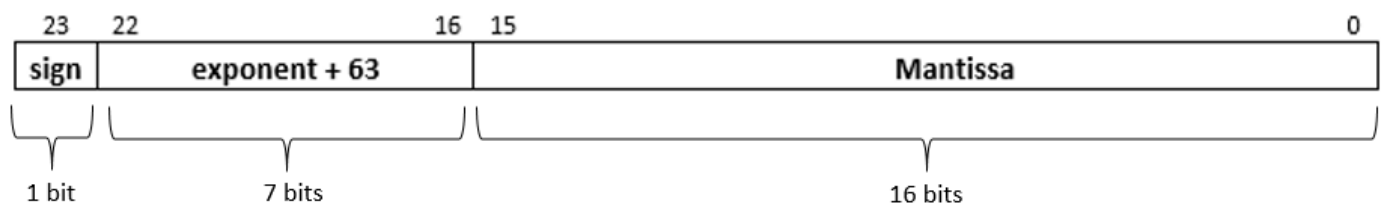
Address	Code
0x80400064	0x08100014

איזו פקודה זו, ומה הכתובת של הפקודה הבאה שתבצע?

- אין מספיק נתונים לדעת איזו פקודה זו, והפקודה הבאה לביצוע תהיה בכתובת 0x80400068
- זו פקודה R, והפקודה הבאה לביצוע תהיה בכתובת 0x80400068
- זו פקודת Jump, והפקודה הבאה לביצוע תהיה בכתובת 0x80400050
- זו פקודת Jump, והפקודה הבאה לביצוע תהיה בכתובת 0x00400050
- זו פקודת BEQ, והפקודה הבאה לביצוע תהיה בכתובת 0x00400050

### יש להתייחס לנתון הבא בפתרון שאלות 4 ו-5:

במחשב העובד עם 24 ביטים (במקום 32) נתון כי ייצוג מספרים ב- Floating Point הינו במבנה הבא:



### שאלה 4

על בסיס הגדרת מבנה הייצוג ב Floating Point הנתון: מהו הערך המוחלט הגדול ביותר שניתן לייצג, ומהו הערך המוחלט הקטן ביותר שניתן לייצג?

- א. מקסימום בערך מוחלט:  $\epsilon - 2^{128}$ , מינימום בערך מוחלט:  $2^{-126}$
- ב. מקסימום בערך מוחלט:  $\epsilon - 2^{127}$ , מינימום בערך מוחלט:  $2^{-127}$
- ג. מקסימום בערך מוחלט:  $\epsilon - 2^{64}$ , מינימום בערך מוחלט:  $2^{-62}$
- ד. מקסימום בערך מוחלט:  $\epsilon - 2^{63}$ , מינימום בערך מוחלט:  $2^{-63}$
- ה. מקסימום בערך מוחלט:  $\epsilon - 2^{32}$ , מינימום בערך מוחלט:  $2^{-30}$

### שאלה 5

על בסיס הגדרת הייצוג ב Floating Point הנתון, מהו הערך העשרוני של 0x422A00?

- א. 9.625
- ב. 9.3125
- ג. 8.625
- ד. 8.3125
- ה. 0.3125

### שאלה 6:

CPI	% מכלל התוכנית	סוג פקודה
3	25%	A
2	25%	B
1	40%	C
5	10%	D

בטבלה משמאל נתון תמהיל הפקודות ו- CPI של סוג פקודה בתוכנית. בניסיון לשפר ביצועים של המעבד, הצליחו להוריד את ה CPI של פקודות מסוג D, מ- 5 ל- 3. לשיפור זה יש מחיר: יש להגדיל ב 20% את כמות הפקודות מסוג B (למשל: נניח שיש 300 פקודות מסוג B, לאחר השינוי תהינה 360 פקודות מסוג B). כמות הפקודות, של שאר סוגי הפקודות, נותרה ללא שינוי

לאור שינוי זה, מהו ה speedup כתוצאה מהשינוי?

א. לא ניתן לחשב מאחר וחסרים נתונים (CCT, IC)

ב. 1.101

ג. 1.0488

ד. 1.0978

ה. 0.9149 (כלומר, ירידה בביצועים)

### שאלה 7

נתון כי ערכו של אוגר \$t0 הוא 0x80000000

נתונות 2 הפקודות הבאות:

addi \$s0, \$t0, -1

addiu \$s1, \$t0, -1

בהתייחסות לביצוע 2 פקודות אלה, סמנו את התשובה הנכונה:

א. בעת ביצוע פקודת addi – תהיה תעופה עקב overflow, הפקודה addiu תעבוד באופן תקין והתוצאה תהיה 0x7FFFFFFF

ב. שתי הפקודות תעבודנה באופן תקין, והתוצאה בשני המקרים תהיה 0x7FFFFFFF, כאשר בהתייחסות לתוצאה ב \$s0, זה הערך החיובי הגדול ביותר עקב carry (שלא גורם לתעופה)

ג. שתי הפקודות לא תעבודנה באופן תקין: במקרה של addi תהיה תעופה עקב overflow, ובמקרה של addiu תהיה תעופה עקב carry

ד. פקודת addi תבוצע באופן תקין – מתרחש carry אך זה לא גורם לתעופה. פקודות addiu תגרום לתעופה עקב אירוע overflow.

ה. לא ניתן לדעת על בסיס הנתונים לשאלה

### שאלה 8:

במהלך ביצוע פקודת SLL, עקב תקלת חומרה, מבנה הפקודה השתבש. כתוצאה מכך נוצר exception של "פקודה לא חוקית".

בהתבסס על תרשים 4.66, איזה מהמשפטים הבאים נכון?

א. התקלה תזוהה בשלב ה-IF. כתוצאה מכך: אוגר PC יעודכן בכתובת 0x80000180 לטיפול בתקלה, יבוצע flush לאוגר IF/ID בכדי למנוע מהפקודה הנוכחית להמשיך בצנרת, אוגר EPC יעודכן בכתובת הפקודה שגרמה לתקלה, ואוגר Cause יעודכן בקוד התקלה.

ב. התקלה תזוהה בשלב ה-ID. כתוצאה מכך: אוגר PC יעודכן בכתובת 0x80000180 לטיפול בתקלה, יבוצע flush לאוגרים EX/MEM ו-MEB/WB כדי למנוע מהפקודות שלפני זו שגרמה לתקלה להמשיך בצנרת, יבוצע flush לאוגר ID/EX בכדי למנוע מהפקודה הנוכחית להמשיך בצנרת, אוגר EPC יעודכן בכתובת הפקודה שגרמה לתקלה, ואוגר Cause יעודכן בקוד התקלה.

ג. התקלה תזוהה בשלב ה-IF. כתוצאה מכך יבוצע flush לאוגר IF/ID כדי למנוע מהפקודה שגרמה לתקלה להמשיך בצנרת, אוגר PC יעודכן בכתובת הפקודה הבאה (pc+4) מאחר והתקלה זוהתה בשלב ה-IF – אין צורך לטיפול של ה-exception handler, אוגר EPC ו-Cause יעודכנו בהתאמה בכדי לאפשר זיהוי של התקלה ומיקום המשך הקוד.

ד. התקלה תזוהה בשלב ה-ID. כאשר הפקודה תגיע לשלב ה-EX יעודכן אוגר EPC בכתובת החזרה, אוגר Cause יעודכן בקוד התקלה, יבוצע flush לאוגרי הצנרת, IF/ID, ID/EX, EX/MEM כדי למנוע מפקודה שגרמה לתקלה ואלה שאחריה להמשיך, אוגר PC יעודכן בכתובת 0x80000180 לטיפול בתקלה.

ה. התקלה תזוהה בשלב ה-ID. כתוצאה מכך: אוגר PC יעודכן בכתובת 0x80000180 לטיפול בתקלה, יבוצע flush לאוגר IF/ID כדי למנוע מהפקודה שלאחר זו שגרמה לתקלה להמשיך בצנרת, יבוצע flush לאוגר ID/EX כדי למנוע מהפקודה הנוכחית להמשיך בצנרת, אוגר EPC יעודכן בכתובת הפקודה שגרמה לתקלה, ואוגר Cause יעודכן בקוד התקלה.

### שאלה 9:

בנספח החומרה יש תרשים צנרת הכולל את מנגנון ה-BTB ויחידת ה-Mispredict Detection Unit. נתון כי פקודת BEQ, בעת היותה בשלב ה-IF, זוהתה עם predict direction שהינו **Not Taken** ובהתאמה ל- not taken נכנסו פקודות לצנרת לאחר פקודת ה-branch.

כאשר פקודת ה-BEQ הגיעה לשלב ה-MEM, דגל ה-Zero קיבל את הערך 1.

איזה מהמשפטים הבאים אינו נכון?

א. משמעות  $Zero = 1$  הינה שבפועל צריך היה להיות Branch Taken  
ב. דגל ה-Pred Direction של יחידת ה-BTB, שעל בסיס ערכו נלקחה ההחלטה בשלב ה-IF, יזוהה

**כשגוי** על ידי יחידת ה-Mispredict Detection Unit

ג. דגל ה-Pred Direction של יחידת ה-BTB, שעל בסיס ערכו נלקחה ההחלטה בשלב ה-IF, יזוהה **כנכון** על ידי יחידת ה-Mispredict Detection Unit

ד. יחידת ה-Mispredict Detection Unit תבצע flush לאוגרי הצנרת IF/ID, ID/EX, EX/MEM  
ה. יחידת ה-Mispredict Detection Unit תעדכן את אוגר PC בכתובת הדילוג של פקודת ה-BEQ (במקום pc+4 שהוזן בשלב ה-IF)

### יש להתייחס לנתונים הבאים בשאלות 10-11

נתון זיכרון מטמון עם המאפיינים הבאים:

- 8 way associative
- גודל בלוק הוא 2048 סיביות/bits
- במטמון יש  $131,072 (2^{17})$  סיביות Valid
- הכתובות במבנה של 32 סיביות

### שאלה 10:

נתונה הכתובת 0x1018008A.

על סמך המידע הנתון לנו על המטמון, סמן את התשובה הנכונה:

- Byte Offset = 0x2, Block Offset = 0x22, index = 0x1800, tag = 0x040
  - Byte Offset = 0x2, Block Offset = 0x22, index = 1800, tab = 0x08
  - Byte Offset = 0x8A, Block Offset = 0x00, index = 1800, tab = 0x040
  - Byte Offset = 0x0, Block Offset = 0x8A, index = 1800, tab = 0x08
- ה. לא ניתן לדעת על בסיס נתוני השאלה

### שאלה 11:

על סמך המידע הנתון לנו על המטמון, מהו הגודל הפיזי של כל המטמון (כולל valid + tag + data) ב- מילים (word)

- 1,048,576
- 8,302,592
- 8,392,704
- 8,433,664

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:

<u>Address</u>	<u>Code</u>	<u>Basic</u>	<u>Source</u>
0x00400080	0x20094004	addi \$9,\$0,0x00004004	1 addi \$t1, \$0, 0x4004
0x00400084	0x00094b80	sll \$9,\$9,0x0000000e	2 sll \$t1, \$t1, 14
0x00400088	0x20081000	addi \$8,\$0,0x00001000	3 addi \$t0, \$0, 0x1000
0x0040008c	0x00084280	sll \$8,\$8,0x0000000a	4 sll \$t0, \$t0, 10
0x00400090	0xad100088	sw \$16,0x00000088(\$8)	5 sw \$s0, 136(\$t0)
0x00400094	0x8d300010	lw \$16,0x00000010(\$9)	6 lw \$s0, 16(\$t1)
0x00400098	0x01104820	add \$9,\$8,\$16	7 add \$t1, \$t0, \$s0
0x0040009C	0x01084026	xor \$8,\$8,\$8	8 xor \$t0, \$t0, \$t0

**שימו לב:** הקוד כולל אתחולי אוגרים הנדרשים למימוש. ערך האוגרים בתחילת קוד זה – לא נתון!

צוות הטכנאים שבנה והרכיב את המעבד, קיבל לידיו את תכנון המעבד והדגלים. בנתוני טבלה 4.18 שקיבל צוות הטכנאים יש טעות הקלדה. טעות ההקלדה ממוקדת בהגדרות ה**בקרה הראשית** של פקודות LW ו-SW. צוות הטכנאים מימש את המעבד על בסיס הנתונים השגויים.

בטבלה המצורפת כאן, מודגשות טעויות ההגהה של הפקודות LW ו-SW. בדקו בתשומת לב את הטעויות (מודגש בריבוע המובלט) בהשוואה לנתוני טבלה 4.18 התקינה, כפי שמופיע לכם בנספח החומרה של המבחן. ודאו שאתם מזהים את הטעויות ומבינים את ההשלכות של טעויות אלה.

Instruction	Opcode	RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALUOp0	Jump
sw	43	0	1	1	1	1	0	0	0	0	0
lw	35	0	1	1	0	0	1	0	0	0	0

**כל סעיפי השאלה מבוססים על מעבד הכולל את המימוש על בסיס טבלה 4.18 השגויה כפי שרשום כאן.**

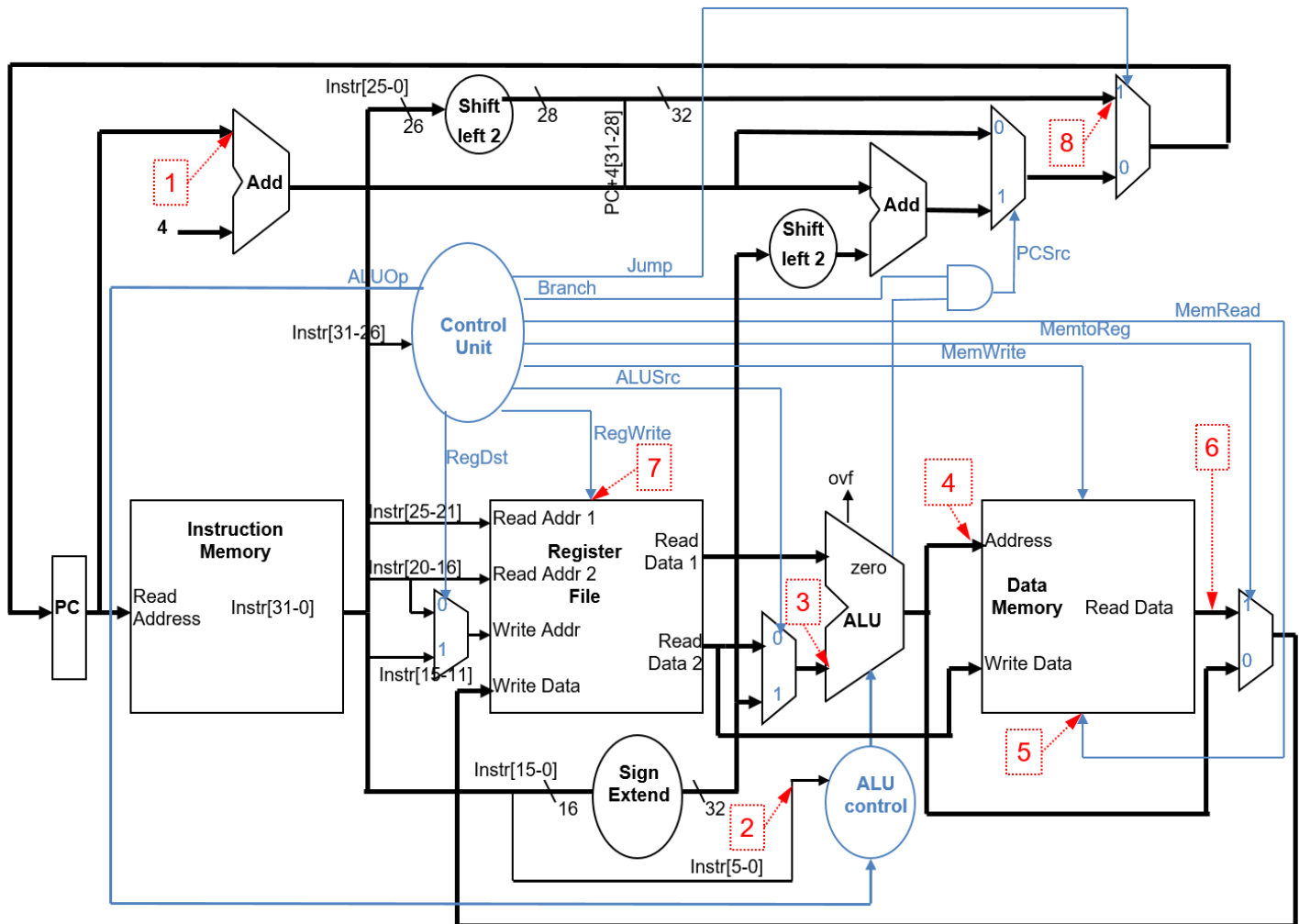
כמו כן, נתון שניתן לגשת לכל מרחב הזיכרון בפקודות SW/LW (כל עוד הכתובת מתחלקת ב-4)

**נתונים אלה תקפים לכל סעיפי השאלה**

## 12.1. (16 נקודות, 2 נקודות לכל סימון)

בתרשים שלהן מעבד חד מחזורי. הבקרה הראשית במעבד זה מומשה בהתאם לטבלה 4.18 השגויה, כפי שהוסבר בתחילת השאלה בעמוד הקודם.

על התרשים סימונים של 8 נקודות. יש למלא את הטבלה, שבתחתית עמוד זה, עם הערכים העוברים על הקווים **בבסיס הקסה**, בהנחה שהמידע נבדק לקראת סוף פעימת השעון בביצוע הפקודה החמישית (5) בקוד: **sw \$s0, 136(\$t0)**. במקרה של ערך לא ידוע, או שלא ניתן לדעת, יש לסמן X.



1	
2	
3	
4	

5	
6	
7	
8	



## 12.2. (20 נקודות)

כעת נריץ את קטע הקוד על מעבד MIPS **העובד בטכנולוגית צנרת** (על בסיס התרשים שלפניכם, המבוסס על תרשים 4.60). **שימו לב:** הבקרה הראשית, במעבד צנרת זה, מומשה בהתאם לטבלה 4.18 השגויה, כפי שהוסבר בתחילת השאלה. במידת הצורך, היעזרו בתרשים 4.51 לזיהוי מדויק של קווי הבקרה (דגלים).

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.

### השאלה:

בפעימה זו, מחזור שעון 9 של רצף הפקודות הנתון, בשלב ה-WB (שלב 5) נמצאת פקודת ה-SW (פקודה מס' 5 בקוד הנתון).

**חלק א' (4 נקודות): השלימו את רצף הפקודות במעבד – מהן הפקודות הנמצאות בשלבים IF, ID, EX, MEM, ובצנרת? יש לרשום את הפקודות, בהתאמה, במקום המיועד לכך בחלק העליון בתרשים.**

**חלק ב' (16 נקודות, 2 נקודות לכל סימון):** על התרשים סימונים של 8 נקודות. יש למלא את הטבלה, שבתחילת העמוד הבא, עם הערכים העוברים על הקווים **בבסיס הקסה**. במקרה של ערך לא ידוע/לא ניתן לדעת - יש לסמן X.

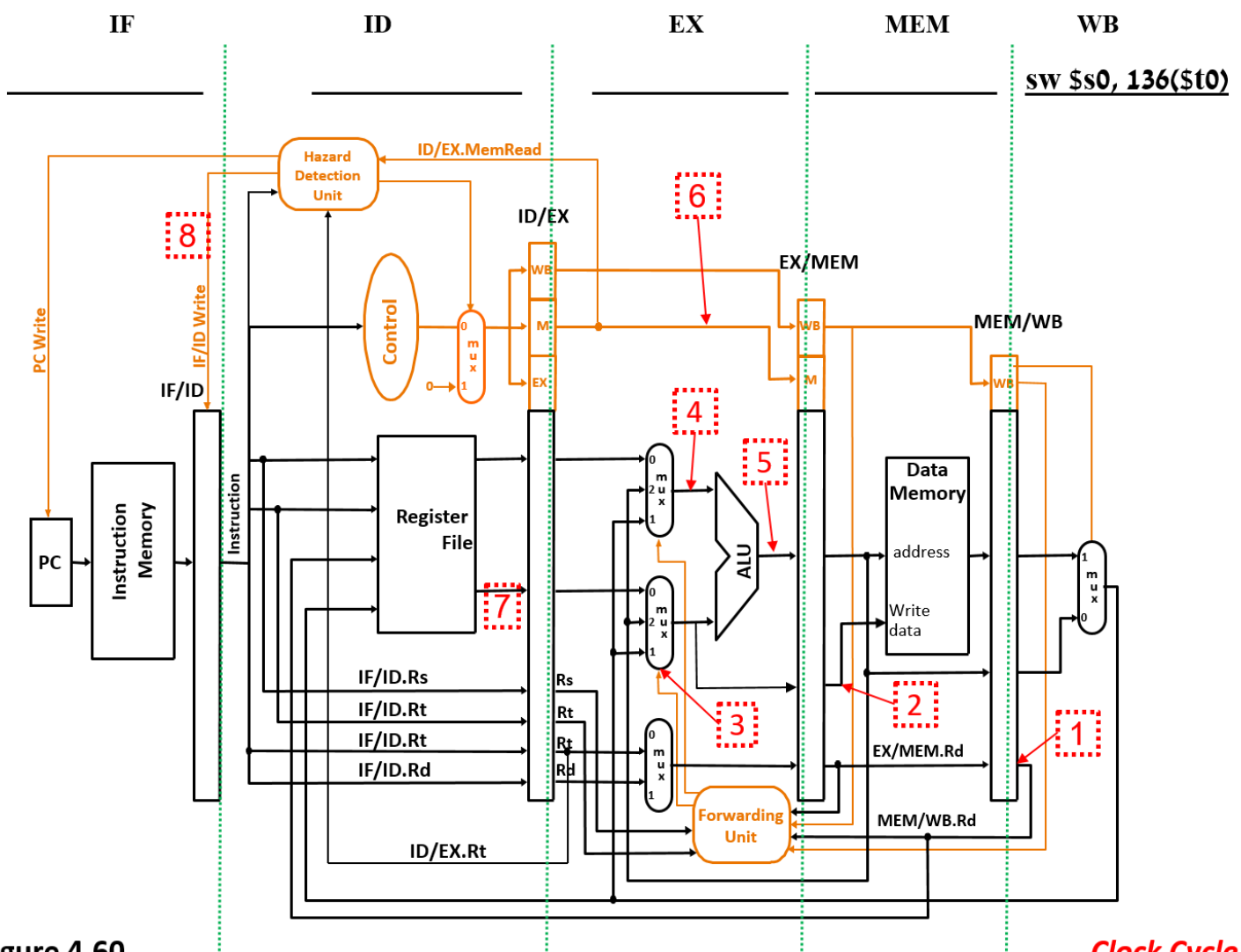


Figure 4.60

**Clock Cycle 9**



1	
2	
3	
4	

5	
6	
7	
8	

**שאלה 13 (31 נקודות)**

נתונה פרוצדורה בשם whatDoldo\_Y. הפרוצדורה מקבלת 4 פרמטרים:

- \$a0: כתובת תחילת מערך של מספרים (כל מספר בגודל מילה)
- \$a1: N – מספר האיברים (ערכים) במערך (ה-length של המערך)
- \$a2: M – מספר שלם חיובי – גדול מ-0, קטן מ-N
- \$a3: Ind – מכיל ערך 0 או 1

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

arr: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

קוד הפרוצדורה:

whatDoldo\_Y:

```
# $a0 - array address
# $a1 - array length
# $a2 - value ($a2 > 0 && $a2 < $a1)
# $a3 - Indication (0, 1)
xor $v0, $v0, $v0
addi $t9, $a1, 0
addi $a1, $a1, -1
div $a1, $a2
mflo $a1
addi $a1, $a1, 1
sll $t0, $a2, 2
beq $a3, $zero, cont
sll $t9, $t9, 2
addi $t9, $t9, -4
add $a0, $a0, $t9
neg $t0, $t0
```

cont:

```
lw $t1, 0($a0)
add $v0, $v0, $t1
add $a0, $a0, $t0
addi $a1, $a1, -1
bgtz $a1, cont
jr $ra
```

**ודאו שהבנתם את מימוש הפרוצדורה לפני שאתם עונים על השאלות הבאות**

13.1. (8 נקודות)

איזה ערך יוחזר באוגר  $v_0$  בסיום ריצת הפרוצדורה בהינתן הערכים הבאים בפרמטרים:

- $a_0$  – כתובת מערך arr
- $a_1 - 10$
- $a_2 - 2$
- $a_3 - 0$

---

---

---

13.2. (8 נקודות)

איזה ערך יוחזר באוגר  $v_0$  בסיום ריצת הפרוצדורה בהינתן הערכים הבאים בפרמטרים:

- $a_0$  – כתובת מערך arr
- $a_1 - 10$
- $a_2 - 4$
- $a_3 - 1$

---

---

---

### 13.3. (9 נקודות)

עקב טעות הקלדה, בהפעלת הפרוצדורה, עבור פרמטר \$a2\$ הוזן בטעות ערך שלילי – הערך **5**.  
הפרוצדורה מופעלת עם הערכים הבאים בפרמטרים:

- \$a0 – כתובת מערך arr
- \$a1 – 10
- \$a2 – (-5)
- \$a3 – 0

## הסבירו מה יקרה בעת ריצת הפרוצדורה:

- כמה פעמים תעבוד הלולאה שבפרוצדורה?
- מה להערכתכם יתרחש בעת ריצת הפרוצדורה?

[illegible]

### 13.4. (6 נקודות)

הסבירו ב- 2-3 משפטים מה עושה הפרוצדורה (בהנחה, כמובן, שכל הפרמטרים מועברים באופן תקין)

[illegible]

## This image shows a full page of blank, lined paper. It features approximately 20 evenly spaced horizontal black lines across its entire width, providing a guide for handwriting or typing. The paper itself is a clean, off-white color.

This image shows a full page of blank white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page, providing a guide for writing. There are no margins, text, or other markings on the paper.

[illegible]



**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 טיוטה)!**

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

באוגר 8 – יש את הערך ( -1 )  
ביצוע XOR של המספר ב \$t2, לתוך \$t1, למעשה שומר ב \$t1 את "היפוך הסיביות"  
של \$t1  
הפחתת \$t0 (שערכו -1) מתוצאת היפוך הסיביות (\$t1) – למעשה שוות ערך  
להוספת 1,  
כלומר, הקוד מבצע למעשה את שלבי "משלים 2"

## שאלה 1

נתון קטע הקוד הבא:

Address	Code	Basic	Source
0x00400010	0x2008ffff	addi \$8,\$0,0xffffffff	1 addi \$t0, \$zero, -1
0x00400014	0x3c01ffff	lui \$1,0xffffffff	2 xori \$t1, \$t2, 0xffffffff
0x00400018	0x3421ffff	ori \$1,\$1,0x0000ffff	
0x0040001c	0x01414826	xor \$9,\$10,\$1	
0x00400020	0x01284822	sub \$9,\$9,\$8	3 sub \$t1, \$t1, \$t0

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 מבצע קטע הקוד הנתון?  
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, 3
addi $s1, $zero, -1
add $t0, $zero, $zero
xor $v0, $v0, $v0
loopY:
addi $v0, $v0, 2
or $t0, $t0, $s0
sll $s0, $s0, 2
bne $t0, $s1, loopY
```

לפני הלולאה:  
\$s0 = 3, כלומר, סיביות 0,1 במילה עם ערך 1, כל שאר סיביות – 0  
\$s1 = -1, כלומר, כל הסיביות עם ערך 1  
\$v0 = 0, \$t0 = 0  
בלולאה, בכל סבב:  
\$v0 גדל ב- 2  
מבצע or של אוגר \$t0 עם \$s0, ובתוצאה ב \$t0, 2 סיביות מקבילות "נדלקות"  
מבוצעת הזזה שמאלה, 2 מקומות של סיביות באוגר \$s0, כלומר, בסבב הבא  
2 הסיביות ה- "דולקות" תהינה שני מקומות שמאלה  
הלולאה עובדת עד ש \$s1=\$t0, כלומר עד שכל הסיביות ב \$t0 "דולקות"  
מאחר וכל פעם 2 סיביות נדלקות – אחר 16 פעמים יסיים, ו- \$v0=32

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

א. זו לולאה אין סופית, כי אוגר \$t0 אף פעם לא יהיה שווה לאוגר \$s1  
ב. הלולאה תתבצע פעם אחת, וערכו של אוגר \$v0 בסיום קטע הקוד יהיה 2  
ג. הלולאה תתבצע 32 פעמים, וערכו של אוגר \$v0 בסיום קטע הקוד יהיה 16  
ד. הלולאה תתבצע 16 פעמים, וערכו של אוגר \$v0 בסיום קטע הקוד יהיה 32  
ה. הלולאה תתבצע 32 פעמים, וערכו של אוגר \$v0 בסיום קטע הקוד יהיה 32

00001000000100000000000000010100  
 00000100000000000000000000010100  
 0000010000000000000000000001010000  
 0000000001000000000000000001010000  
 0000 0000 0100 0000 0000 0000 0101 0000  
 0x00400050

## שאלה 3

נתונה פקודת המכונה הבאה והכתובת בה היא נמצאת:

Address	Code
0x80400064	0x08100014

איזו פקודה זו, ומה הכתובת של הפקודה הבאה שתבצע?

א. אין מספיק נתונים לדעת איזו פקודה זו, והפקודה הבאה לביצוע תהיה בכתובת 0x80400068

ב. זו פקודה R, והפקודה הבאה לביצוע תהיה בכתובת 0x80400068

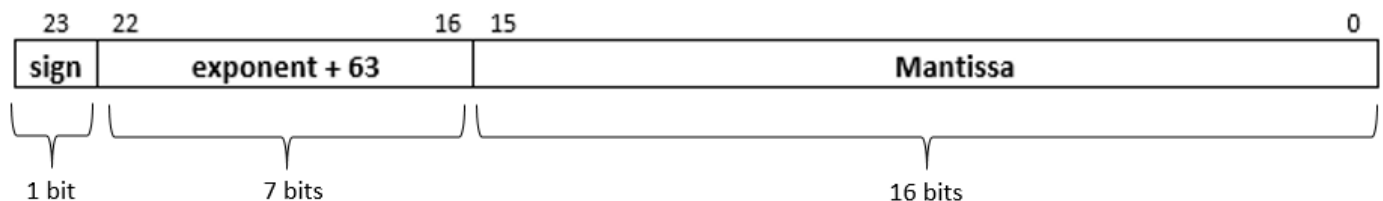
ג. זו פקודת Jump, והפקודה הבאה לביצוע תהיה בכתובת 0x80400050

ד. זו פקודת Jump, והפקודה הבאה לביצוע תהיה בכתובת 0x00400050

ה. זו פקודת BEQ, והפקודה הבאה לביצוע תהיה בכתובת 0x00400050

## יש להתייחס לנתון הבא בפתרון שאלות 4 ו-5:

במחשב העובד עם 24 ביטים (במקום 32) נתון כי ייצוג מספרים ב- Floating Point הינו במבנה הבא:



## שאלה 4

על בסיס הגדרת מבנה הייצוג ב Floating Point הנתון: מהו הערך המוחלט הגדול ביותר שניתן לייצג, ומהו הערך המוחלט הקטן ביותר שניתן לייצג?

א. מקסימום בערך מוחלט:  $\epsilon - 2^{128}$ , מינימום בערך מוחלט:  $2^{-126}$

ב. מקסימום בערך מוחלט:  $\epsilon - 2^{127}$ , מינימום בערך מוחלט:  $2^{-127}$

ג. מקסימום בערך מוחלט:  $\epsilon - 2^{64}$ , מינימום בערך מוחלט:  $2^{-62}$

ד. מקסימום בערך מוחלט:  $\epsilon - 2^{63}$ , מינימום בערך מוחלט:  $2^{-63}$

ה. מקסימום בערך מוחלט:  $\epsilon - 2^{32}$ , מינימום בערך מוחלט:  $2^{-30}$

## שאלה 5

על בסיס הגדרת הייצוג ב Floating Point הנתון, מהו הערך העשרוני של 0x422A00?

א. 9.625

ב. 9.3125

ג. 8.625

ד. 8.3125

ה. 0.3125

0x422A00 = 0100 0010 0010 1010 0000 0000  
 $\rightarrow 0 100010 0010101000000000$   
 $\rightarrow 0 100010 (1.0010101$   
 $\rightarrow + 66 \quad 1 + 2^{-3} + 2^{-5} + 2^{-7}$   
 $\rightarrow + 3 \quad 1 + 2^{-3} + 2^{-5} + 2^{-7}$   
 $\rightarrow + 2^3 * (1 + 2^{-3} + 2^{-5} + 2^{-7})$   
 $\rightarrow + 2^3 + 2^0 + 2^{-2} + 2^{-4}$   
 $\rightarrow 8 + 1 + 0.25 + 0.0625 = 9.3125$

## שאלה 6:

CPI	% מכלל התוכנית	סוג פקודה
3	25%	A
2	25%	B
1	40%	C
5	10%	D

בטבלה משמאל נתון תמהיל הפקודות ו- CPI של סוג פקודה בתוכנית. בניסיון לשפר ביצועים של המעבד, הצליחו להוריד את ה CPI של פקודות מסוג D, מ- 5 ל- 3. לשיפור זה יש מחיר: יש להגדיל ב 20% את כמות הפקודות מסוג B (למשל: נניח שיש 300 פקודות מסוג B, לאחר השינוי תהינה 360 פקודות מסוג B). כמות הפקודות, של שאר סוגי הפקודות, נותרה ללא שינוי

לאור שינוי זה, מהו ה speedup כתוצאה מהשינוי?

א. לא ניתן לחשב מאחר וחסרים נתונים (CCT, IC)

ב. 1.101

ג. 1.0488

ד. 1.0978

ה. 0.9149 (כלומר, ירידה בביצועים)

סוג פקודה	% מכלל התוכנית	CPI	תמהיל קודם - זו התמונה העדכנית - ס"ה - 105%	תמהיל ביחס ל- 100% "החדש" - ביחס - 105%	CPI ממוצע חדש
A	25%	3	25%	23.81%	0.7143
B	25%	2	30%	28.57%	0.5714
C	40%	1	40%	38.10%	0.3810
D	10%	3	10%	9.52%	0.2856

CPI ממוצע - מקורי:  $2.15 = (3 \cdot 0.25 + 2 \cdot 0.25 + 1 \cdot 0.4 + 5 \cdot 0.1)$

CPI ממוצע - חדש: 1.9523

ומכאן, שה speedup:  $2.15 / 1.9523 = 1.101$

## שאלה 7

נתון כי ערכו של אוגר \$t0 הוא 0x80000000  
נתונות 2 הפקודות הבאות:

addi \$s0, \$t0, -1  
addiu \$s1, \$t0, -1

בהתייחסות לביצוע 2 פקודות אלה, סמנו את התשובה הנכונה:

א. בעת ביצוע פקודת addi - תהיה תעופה עקב overflow, הפקודה addiu תעבוד באופן תקין והתוצאה תהיה 0x7FFFFFFF

ב. שתי הפקודות תעבודנה באופן תקין, והתוצאה בשני המקרים תהיה 0x7FFFFFFF, כאשר בהתייחסות לתוצאה ב \$s0, זה הערך החיובי הגדול ביותר עקב carry (שלא גורם לתעופה)

ג. שתי הפקודות לא תעבודנה באופן תקין: במקרה של addi תהיה תעופה עקב overflow, ובמקרה של addiu תהיה תעופה עקב carry

ד. פקודת addi תבוצע באופן תקין - מתרחש carry אך זה לא גורם לתעופה. פקודות addiu תגרום לתעופה עקב אירוע overflow.

ה. לא ניתן לדעת על בסיס הנתונים לשאלה

הערך 0x80000000:

- בהתייחסות כאל ערך ללא סימן, שווה -  $2^{31}$
- בהתייחסות כאל ערך עם סימן, שווה -  $2^{31}$

addi - יגרום לתעופה עקב overflow מאחר ויש חריגה מהערך השלילי הקטן ביותר האפשרי  
addiu - מאחר וההתייחסות לערך הינה כאל ערך ללא סימן, אין בעיה לבצע את החיסור והתוצאה תהיה  $2^{31}-1$ , וזה למעשה הערך הגדול ביותר שניתן לרשום במספרים עם סימן

### שאלה 8:

במהלך ביצוע פקודת SLL, עקב תקלת חומרה, מבנה הפקודה השתבש. כתוצאה מכך נוצר exception של "פקודה לא חוקית".

בהתבסס על תרשים 4.66, איזה מהמשפטים הבאים נכון?

א. התקלה תזוהה בשלב ה-IF. כתוצאה מכך: אוגר PC יעודכן בכתובת 0x80000180 לטיפול בתקלה, יבוצע flush לאוגר IF/ID בכדי למנוע מהפקודה הנוכחית להמשיך בצנרת, אוגר EPC יעודכן בכתובת הפקודה שגרמה לתקלה, ואוגר Cause יעודכן בקוד התקלה.

ב. התקלה תזוהה בשלב ה-ID. כתוצאה מכך: אוגר PC יעודכן בכתובת 0x80000180 לטיפול בתקלה, יבוצע flush לאוגרים EX/MEM ו-MEB/WB כדי למנוע מהפקודות שלפני זו שגרמה לתקלה להמשיך בצנרת, יבוצע flush לאוגר ID/EX בכדי למנוע מהפקודה הנוכחית להמשיך בצנרת, אוגר EPC יעודכן בכתובת הפקודה שגרמה לתקלה, ואוגר Cause יעודכן בקוד התקלה.

ג. התקלה תזוהה בשלב ה-IF. כתוצאה מכך יבוצע flush לאוגר IF/ID כדי למנוע מהפקודה שגרמה לתקלה להמשיך בצנרת, אוגר PC יעודכן בכתובת הפקודה הבאה (pc+4) מאחר והתקלה זוהתה בשלב ה-IF – אין צורך לטיפול של ה-exception handler, אוגר EPC ו-Cause יעודכנו בהתאמה בכדי לאפשר זיהוי של התקלה ומיקום המשך הקוד.

ד. התקלה תזוהה בשלב ה-ID. כאשר הפקודה תגיע לשלב ה-EX יעודכן אוגר EPC בכתובת החזרה, אוגר Cause יעודכן בקוד התקלה, יבוצע flush לאוגרי הצנרת, IF/ID, ID/EX, EX/MEM כדי למנוע מפקודה שגרמה לתקלה ואלה שאחריה להמשיך, אוגר PC יעודכן בכתובת 0x80000180 לטיפול בתקלה.

ה. התקלה תזוהה בשלב ה-ID. כתוצאה מכך: אוגר PC יעודכן בכתובת 0x80000180 לטיפול בתקלה, יבוצע flush לאוגר IF/ID כדי למנוע מהפקודה שלאחר זו שגרמה לתקלה להמשיך בצנרת, יבוצע flush לאוגר ID/EX כדי למנוע מהפקודה הנוכחית להמשיך בצנרת, אוגר EPC יעודכן בכתובת הפקודה שגרמה לתקלה, ואוגר Cause יעודכן בקוד התקלה.

### שאלה 9:

בנספח החומרה יש תרשים צנרת הכולל את מנגנון ה-BTB ויחידת ה-Mispredict Detection Unit. נתון כי פקודת BEQ, בעת היותה בשלב ה-IF, זוהתה עם predict direction שהינו **Not Taken** ובהתאמה ל- not taken נכנסו פקודות לצנרת לאחר פקודת ה-branch.

כאשר פקודת ה-BEQ הגיעה לשלב ה-MEM, דגל ה-Zero קיבל את הערך 1.

איזה מהמשפטים הבאים אינו נכון?

א. משמעות  $Zero = 1$  הינה שבפועל צריך היה להיות Branch Taken  
ב. דגל ה-Pred Direction של יחידת ה-BTB, שעל בסיס ערכו נלקחה ההחלטה בשלב ה-IF, יזוהה כשגוי על ידי יחידת ה-Mispredict Detection Unit

ג. דגל ה-Pred Direction של יחידת ה-BTB, שעל בסיס ערכו נלקחה ההחלטה בשלב ה-IF, יזוהה כנכון על ידי יחידת ה-Mispredict Detection Unit

ד. יחידת ה-Mispredict Detection Unit תבצע flush לאוגרי הצנרת IF/ID, ID/EX, EX/MEM  
ה. יחידת ה-Mispredict Detection Unit תעדכן את אוגר PC בכתובת הדילוג של פקודת ה-BEQ (במקום pc+4 שהוזן בשלב ה-IF)

גודל בלוק  $2^{11}$  סיביות  $\leftarrow 2^8$  בתים  $\leftarrow 2^6$  מילים, כלומר 64 מילים  
בבלוק, ומכאן שיש 6 סיביות ב block offset  
נתון  $2^{17}$  סיביות valid, וגם נתון 8 way associative. נחלק ב- 8  
ונמצא את כמות השורות ב way יחיד: ומכאן שיש  $2^{14}$  שורות  
במטמון (16,384)  
לאור זאת, בהתייחסות גם ל- 2 סיביות של byte offset, גודל ה-  
tag הוא  $32-14-6-2 = 10$

### יש להתייחס לנתונים הבאים בשאלות 10-11

נתון זיכרון מטמון עם המאפיינים הבאים:

- 8 way associative
- גודל בלוק הוא 2048 סיביות/bits
- במטמון יש  $131,072 (2^{17})$  סיביות Valid
- הכתובות במבנה של 32 סיביות

### שאלה 10:

נתונה הכתובת 0x1018008A.

0x1018008a  
00010000000110000000000010001010<sub>bin</sub>  
0001000000 01100000000000 100010 10 (בחלוקה לשדות)  
Tag = 0x040, index = 0x1800  
Block Offset = 0x22, Byte offset = 0x2

על סמך המידע הנתון לנו על המטמון, סמן את התשובה הנכונה:

א. **Byte Offset = 0x2, Block Offset = 0x22, index = 0x1800, tag = 0x040**

ב. Byte Offset = 0x2, Block Offset = 0x22, index = 1800, tab = 0x08

ג. Byte Offset = 0x8A, Block Offset = 0x00, index = 1800, tab = 0x040

ד. Byte Offset = 0x0, Block Offset = 0x8A, index = 1800, tab = 0x08

ה. לא ניתן לדעת על בסיס נתוני השאלה

### שאלה 11:

על סמך המידע הנתון לנו על המטמון, מהו הגודל הפיזי של כל המטמון (כולל valid + tag + data)  
ב- מילים (word)

א. 1,048,576

ב. 8,302,592

ג. 8,392,704

ד. **8,433,664**

ה. לא ניתן לדעת על בסיס נתוני השאלה

גודל בסיביות של valid+tag+data:  
 $1+10+2048 = 2059$   
בכל יחידת way יש  $2^{14}$  שורות  
יש 8 way  
הגודל הפיזי של כל המטמון בסיביות הוא:  
 $(2^3 * 2^{14} * 2059)_{\text{bits}}$   
נחלק ב- 32 כדי לקבל ביחידות word:  
 $(2^3 * 2^{14} * 2059)_{\text{bits}} / 2^5 =$   
 $= 2^{12} * 2059 = 8,433,664 \text{ words} \sim 2^{23}$

## 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
0x00400080	0x20094004	addi \$9,\$0,0x00004004	1 addi \$t1, \$0, 0x4004
0x00400084	0x00094b80	sll \$9,\$9,0x0000000e	2 sll \$t1, \$t1, 14
0x00400088	0x20081000	addi \$8,\$0,0x00001000	3 addi \$t0, \$0, 0x1000
0x0040008c	0x00084280	sll \$8,\$8,0x0000000a	4 sll \$t0, \$t0, 10
0x00400090	0xad100088	sw \$16,0x00000088(\$8)	5 sw \$s0, 136(\$t0)
0x00400094	0x8d300010	lw \$16,0x00000010(\$9)	6 lw \$s0, 16(\$t1)
0x00400098	0x01104820	add \$9,\$8,\$16	7 add \$t1, \$t0, \$s0
0x0040009C	0x01084026	xor \$8,\$8,\$8	8 xor \$t0, \$t0, \$t0

שימו לב: הקוד כולל אתחולי אוגרים הנדרשים למימוש. ערך האוגרים בתחילת קוד זה – לא נתון!

צוות הטכנאים שבנה והרכיב את המעבד, קיבל לידיו את תכנון המעבד והדגלים. בנתוני טבלה 4.18 שקיבל צוות הטכנאים יש טעות הקלדה. טעות ההקלדה ממוקדת בהגדרות **הבקרה הראשית** של פקודות LW ו-SW. צוות הטכנאים מימש את המעבד על בסיס הנתונים השגויים.

בטבלה המצורפת כאן, מודגשות טעויות ההגהה של הפקודות LW ו-SW. בדקו בתשומת לב את הטעויות (מודגש בריבוע המובלט) בהשוואה לנתוני טבלה 4.18 התקינה, כפי שמופיע לכם בנספח החומרה של המבחן. ודאו שאתם מזדהים את הטעויות ומבינים את ההשלכות של טעויות אלה.

Instruction	Opcode	RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALUOp0	Jump
sw	43	0	1	1	1	1	0	0	0	0	0
lw	35	0	1	1	0	0	1	0	0	0	0

**כל סעיפי השאלה מבוססים על מעבד הכולל את המימוש על בסיס טבלה 4.18 השגויה כפי שרשום כאן.**

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

### החזרה על טבלת ההגהה

מעקב אחר ביצוע הקוד הנ"ל (בהתייחסות לעעות ההגהה):

פקודה (1): אוגר \$9 = 0x00004004

פקודה (2): אוגר \$9 = 0x1001000 (בביטים: 00 0100 0000 0000 0100 00000000000000)

פקודה (3): אוגר \$8 = 0x00001000

פקודה (4): אוגר \$8 = 0x00400000 (בביטים: 00 0000 0001 0000 0000 0000 0000000000)

פקודה (5): עקב טעות ההגהה כפי שמתואר, פקודת SW מבצעת למעשה LW. הכתובת ממנה מבוצע ה load מחושבת לפי ערך אוגר \$8 + 0x88, ולכן הכתובת הינה 0x00400088. זו למעשה כתובת של פקודה (3) בקוד הנתון לנו. ומאחר, ובפועל, זו פקודת LW, אוגר \$8 = 0x20081000, שזהו ה code של הפקודה בזיכרון.

פקודה (6): עקב טעות ההגהה כפי שמתואר, פקודת LW מבצעת למעשה SW. הכתובת אליה מבוצע ה store מחושבת לפי ערך אוגר \$9 + 0x10, ולכן הכתובת הינה 0x10010010. שהערך שייכת לזיכרון בכתובת זו הינו למעשה ה code של פקודת ה addi (פקודה 3) שעודכן באוגר \$16 בפקודה הקודמת (5).

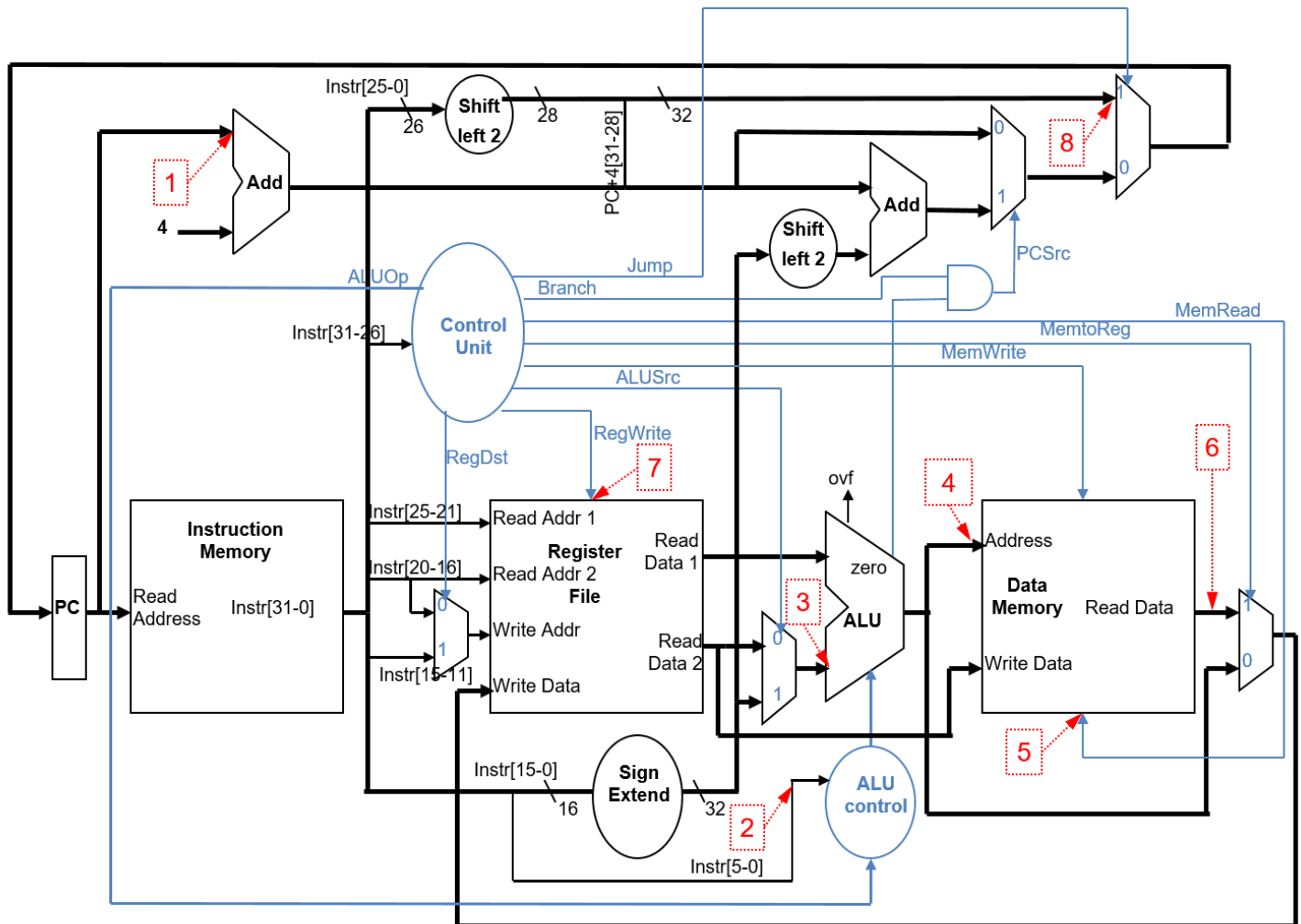
פקודה (7): מחושבת תוצאת החיבור של אוגרים \$8 ו-\$16 ולכן \$9 = 0x20081000+0x00400000=0x20481000

פקודה (8): מאפסת את אוגר \$8

## 12.1. (16 נקודות, 2 נקודות לכל סימון)

בתרשים שלהן מעבד חד מחזורי. הבקרה הראשית במעבד זה מומשה בהתאם לטבלה 4.18 השגויה, כפי שהוסבר בתחילת השאלה בעמוד הקודם.

על התרשים סימונים של 8 נקודות. יש למלא את הטבלה, שבתחתית עמוד זה, עם הערכים העוברים על הקווים **בבסיס הקסה**, בהנחה שהמידע נבדק לקראת סוף פעימת השעון בביצוע הפקודה החמישית (5) בקוד: **sw \$s0, 136(\$t0)**. במקרה של ערך לא ידוע, או שלא ניתן לדעת, יש לסמן X.



1	0x00400090 – כתובת פקודת ה SW
2	0x08 (6 הסיביות הנמוכות בפקודה. אין להם משמעות בפקודה, אבל זה מה שעובר בקו) ניתן לראות זאת בעמודת ה Basic)
3	0x00000088 (הערך המייד 136 בהקסה – ניתן לראות זאת בעמודת ה Basic)
4	0x00400088 – תוצאת החיבור ב ALU, וזו למעשה הכתובת ממנה מבוצע, בפועל, load.

5	0x1 – דגל memRead, עקב טעות ההגהה בתכנון הוא 1, למרות שזו פקודת SW
6	0x20081000 (הכתובת בנק' 4, זו כתובת פקודת ADDI בשורה 3 – וזה ה code שלה)
7	1 = regWrite, למרות שזו פקודת SW, עקב טעות ההגהה בתכנון, זה הדגל בפועל.
8	0x04400220 – זה חישוב של jump על בסיס קוד פקודת ה SW (ערך שלא עובר הלאה מאחר ודגל 0 = jump)



## 12.2. (20 נקודות)

כעת נריץ את קטע הקוד על מעבד MIPS **העובד בטכנולוגית צנרת** (על בסיס התרשים שלפניכם, המבוסס על תרשים 4.60). **שימו לב:** הבקרה הראשית, במעבד צנרת זה, מומשה בהתאם לטבלה 4.18 השגויה, כפי שהוסבר בתחילת השאלה. במידת הצורך, היעזרו בתרשים 4.51 לזיהוי מדויק של קווי הבקרה (דגלים).

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.

### השאלה:

בפעימה זו, מחזור שעון 9 של רצף הפקודות הנתון, בשלב ה-WB (שלב 5) נמצאת פקודת ה-SW (פקודה מס' 5 בקוד הנתון).

**חלק א' (4 נקודות): השלימו את רצף הפקודות במעבד – מהן הפקודות הנמצאות בשלבים IF, ID, EX, MEM בצנרת? יש לרשום את הפקודות, בהתאמה, במקום המיועד לכך בחלק העליון בתרשים.**

**חלק ב' (16 נקודות, 2 נקודות לכל סימון):** על התרשים סימונים של 8 נקודות. יש למלא את הטבלה, שבתחילת העמוד הבא, עם הערכים העוברים על הקווים **בבסיס הקסה**. במקרה של ערך לא ידוע/לא ניתן לדעת - יש לסמן X.

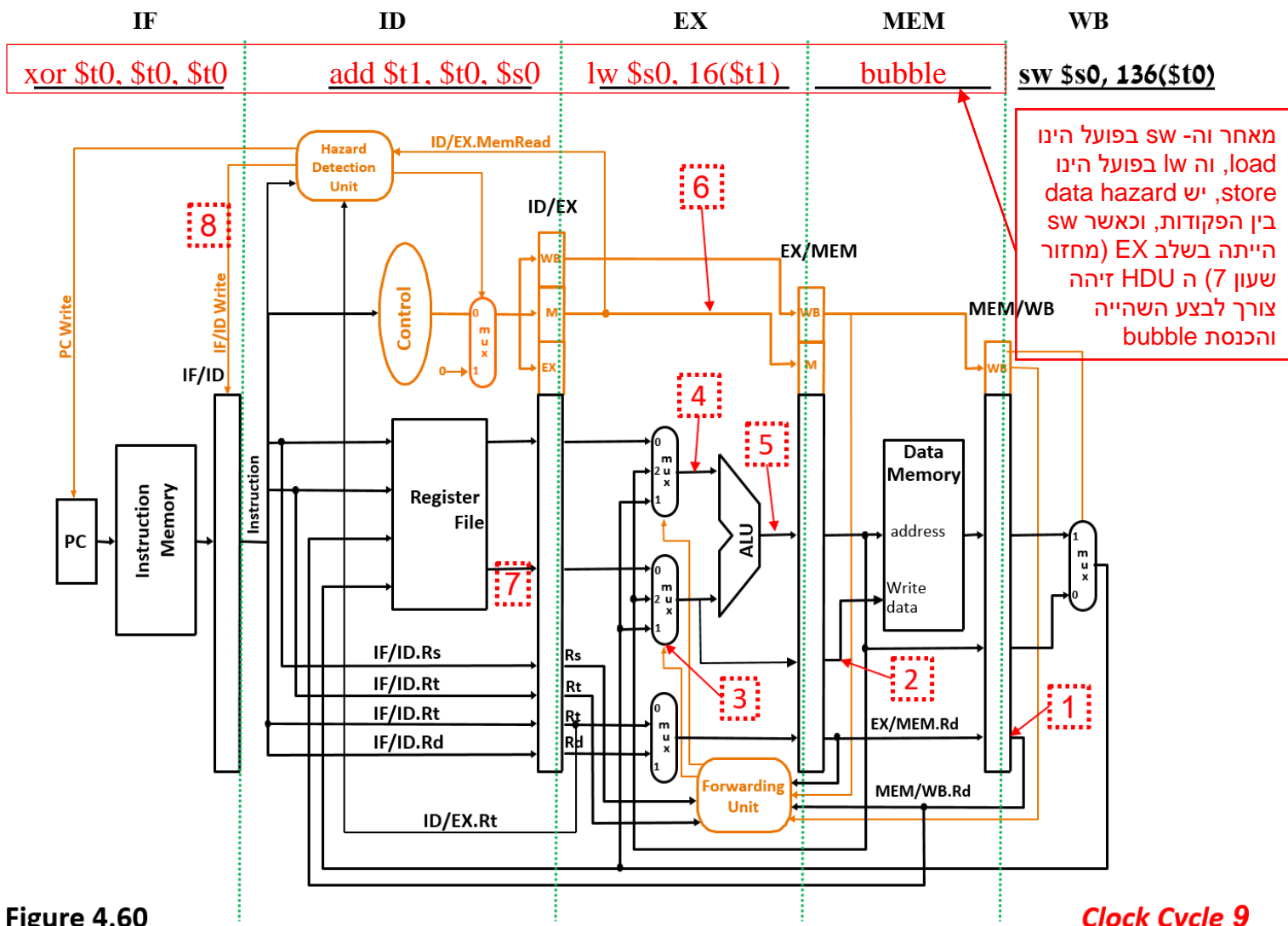


Figure 4.60

Clock Cycle 9

1	0x10 – מספר אוגר המטרה של פקודת sw, שבפועל זו פקודת LW, שהוא אוגר 16 <sub>10</sub>
2	X – זה תוכן אוגר \$s0 לפני מימוש פקודת ה sw (שפועל הינה load). בהגדרת השאלה – אין מידע על תכני האוגרים לפני קטע הקוד
3	0x1 – מבוצע forward 2b לאוגר 16 (\$s0), לקידום הערך משלב WB ל- EX
4	0x10010000 – ערך אוגר \$t1 אחרי ביצוע פקודה (4) בקוד הנתון

5	0x10010010 – הכתובת המחושבת שתהיה יעד הכתיבה בזיכרון (לפי פקודה 6)
6	001 – דגלי שלב mem של פקודת ה LW, שהם בפועל הדגלים של פקודת SW (עקב הטעות באפיון) – לפי טבלה 4.49
7	0x20081000 – זה הערך ש sw רושמת במקבץ האוגרים, ובחציית מקבץ האוגרים זה הערך של אוגר \$s0
8	0x1 – ביט IF/ID Write "רגיל" – כלומר, כתיבה לתוך אוגר IF/ID שהרי אין כאן השהיה

**שאלה 13 (31 נקודות)**

נתונה פרוצדורה בשם whatDoldo\_Y. הפרוצדורה מקבלת 4 פרמטרים:

- \$a0: כתובת תחילת מערך של מספרים (כל מספר בגודל מילה)
- \$a1: N – מספר האיברים (ערכים) במערך (ה-length של המערך)
- \$a2: M – מספר שלם חיובי – גדול מ-0, קטן מ-N
- \$a3: Ind – מכיל ערך 0 או 1

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

arr: .word 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

קוד הפרוצדורה:

whatDoldo\_Y:

```
# $a0 - array address
# $a1 - array length
# $a2 - value ($a2 > 0 && $a2 < $a1)
# $a3 - Indication (0, 1)
xor $v0, $v0, $v0
addi $t9, $a1, 0
addi $a1, $a1, -1
div $a1, $a2
mflo $a1
addi $a1, $a1, 1
sll $t0, $a2, 2
beq $a3, $zero, cont
sll $t9, $t9, 2
addi $t9, $t9, -4
add $a0, $a0, $t9
neg $t0, $t0
```

cont:

```
lw $t1, 0($a0)
add $v0, $v0, $t1
add $a0, $a0, $t0
addi $a1, $a1, -1
bgtz $a1, cont
jr $ra
```

הכנה לפני לולאה:

אוגר \$v0 – מאופס

אוגר \$t9 – ה-length של המערך (מעתיק מהפרמטר \$a1)

אוגר \$a1 – שווה ל:  $(length-1)/a2 + 1$

– תוצאת החילוק היא בשלמים, ולמעשה בודק כמה פעמים "נכנס" \$a2 בתוך \$a1, ובהמשך זה למעשה מונה לולאה

אוגר \$t0 – שווה  $a2*4$  (בהמשך, בלולאה, זה משמש כערך דילוג במערך בהתאם למספר שהוזן ב \$a2

נעשית בדיקה על אוגר \$a3 – אם הוא 0, מדלג ישירות ל cont. אם הוא 1, אז:

• אוגר \$t9 מוכפל ב-4, ומופחת 4 – כלומר, אינדקס מקום אחרון במערך

• אוגר \$a0, שהוא כתובת תחילת המערך – מתווסף לו \$t9, כלומר, הכתובת של המקום האחרון במערך

• אוגר \$t0 – מבוצע לו "הופכי", כלומר הופך לערך שלילי (מאחר ובמקור, \$a2 הוא ערך חיובי בלבד)

בנקודת פתיחה של הלולאה:

• אם  $a3=0$ , אז \$a0 מצביע על תחילת המערך, ו \$t0 הוא ערך חיובי השווה ל  $a2*4$

• אם  $a3=1$ , אז \$a0 מצביע על האיבר האחרון במערך, ו- \$t0 הוא ערך שלילי השווה ל:  $-(a2*4)$

הלולאה – בכל איטרציה:

• מבצעת load לערך נוכחי, לפי כתובת ערך נוכחי ב \$a0

• מוסיפה ערך נוכחי, כסכום מצטבר, לאוגר \$v0

• מקדמת את \$a0 למקום הבא – לפי \$t0 (בפועל – דילוג של \$a2 מקומות, אם  $a3=0$  – קדימה, אם  $a3=1$  אחורה)

• הלולאה עובדת כל עוד \$a1 לא מתאפס – כאשר \$a1 הוא מונה הלולאה, ובפועל מציין כמה פעמים אפשר לדלג ללא חריגה מהמערך

בסיום: \$v0 שווה לסכום האיברים בהתאם לדילוג שבוצע

**ודאו שהבנתם את מימוש הפרוצדורה לפני שאתם עונים על השאלות הבאות**

כלומר, מסכם את איברי המערך בדילוגים של \$a2 איברים, כאשר  $a3=0$  הדילוגים מתחילת המערך לסופו, וכאשר  $a3=1$  הדילוגים מסוף המערך לתחילתו

### 13.1. (8 נקודות)

איזה ערך יוחזר באוגר \$v0 בסיום ריצת הפרוצדורה בהינתן הערכים הבאים בפרמטרים:

- \$a0 – כתובת מערך arr
- 10 – \$a1
- 2 – \$a2
- 0 – \$a3

$$v0 = 25$$

התוצאה הינה סכום כל איבר שני, מתחילת המערך עד סופו (איבר 1, 3, 5, ...)

### 13.2. (8 נקודות)

איזה ערך יוחזר באוגר \$v0 בסיום ריצת הפרוצדורה בהינתן הערכים הבאים בפרמטרים:

- \$a0 – כתובת מערך arr
- 10 – \$a1
- 4 – \$a2
- 1 – \$a3

$$v0 = 18$$

התוצאה הינה סכום כל איבר רביעי, מסוף המערך עד תחילתו (איברים 10, 6, 2)

### 13.3. (9 נקודות)

עקב טעות הקלדה, בהפעלת הפרוצדורה, עבור פרמטר  $a_2$  הוזן בטעות ערך שלילי – הערך 5-  
הפרוצדורה מופעלת עם הערכים הבאים בפרמטרים:

- $a_0$  – כתובת מערך arr
- $a_1$  – 10
- $a_2$  – (-5)
- $a_3$  – 0

הסבירו מה יקרה בעת ריצת הפרוצדורה:

- כמה פעמים תעבוד הלולאה שבפרוצדורה?
- מה להערכתכם יתרחש בעת ריצת הפרוצדורה?

1. עקב הערך השלילי באוגר  $a_2$ , ערכו של אוגר  $a_1$  בחישוב מונה הלולאה – יהיה 0. מאחר ובדיקת מונה הלולאה בסוף הלולאה מבוצעת בהשוואה ל-0, לאחר הפחתת אחד – למעשה הפרוצדורה תיכנס לולאה אין סופית (כמעט...  $2^{32}$  פעמים).  
2. התוכנית צפויה לעוף עקב חריגה בפניה לכתובת בזיכרון. מאחר ובתוך הלולאה מעדכנים את כתובת הפניה למערך (פלוס/מינוס 4 בכל סבב). בשלב מסוים, תוך כדי ריצת הפרוצדורה, תהיה פניה לכתובת לא חוקית/מחוץ לטווח התקין של פניה לזיכרון.

### 13.4. (6 נקודות)

הסבירו ב-2-3 משפטים מה עושה הפרוצדורה (בהנחה, כמובן, שכל הפרמטרים מועברים באופן תקין)

הפרוצדורה מחשבת סכום איברי מערך נתון. בהתאם לכללים הבאים:

- כאשר  $a_3 = 0$ : החל מהאיבר הראשון עד לסוף המערך
- כאשר  $a_3 = 1$ : החל מהאיבר האחרון עד תחילת המערך
- ערכו של  $a_2$  קובע כמה איברים יש לדלג בעת חישוב הסכום (כאשר 2 – כל איבר שני, כאשר 3 – כל איבר שלישי, וכך הלאה)