

הנחייה כללית לכל המבחן:

- יש לענות על **כל השאלות** בטופס השאלון בלבד. רק טופס השאלון ייסרק ויבדק!
- הקפידו לענות במקומות המסומנים למענה על השאלות.
- כל רישום שלא במקום המיועד לתשובות לא יבדק. רק התשובות במקום המיועד - קובעות!
- הקפידו על רישום התשובות באופן מלא ומסודר בהתאם להנחיות!

חלק א: (32 נקודות)

בחלק זה 8 שאלות.

יש לבחור את התשובה הנכונה מבין האפשרויות.

משקל כל שאלה 4 נקודות.

יש לסמן X באופן ברור בטבלה

הערה: יש לסמן רק אפשרות אחת לכל שאלה!

בדקו בקפדנות ובזהירות את הרישום בטבלה שבטופס התשובות, בכדי להימנע מטעויות העתקה ורישום.

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

נתונה תמונת סגמנט הקוד של MIP32, הלקוחה מה MARS איתו תרגלנו בקורס. ענו על שתי שאלות ראשונות (1-2) בהתבסס על סגמנט קוד זה.

Address	Code	Basic	Source
0x80400000	0x240b0006	addiu \$11,\$0,0x00000000	13 li \$t3, 6
0x80400004	0x3c010040	lui \$1,0x00008040	15 la \$a0, unknown
0x80400008	0x34240004	ori \$4,\$1,0x00000004	
0x8040000c	0x01645825	or \$11,\$11,\$4	16 or \$t3, \$t3, \$a0
0x80400010	0x20840008	addi \$4,\$4,0x00000008	17 addi \$a0, \$a0, 0x08
0x80400014	0x00800008	jr \$4	18 jr \$a0
0x80400018	0x2402000a	addiu \$2,\$0,0x0000000a	19 li \$v0, 10
0x8040001c	0x0000000c	syscall	20 syscall

שאלה 1

האם ניתן לדעת מה כתובת התווית unknown, אליה יש התייחסות בשורה 15 של ה-source? במידה וכן, מה כתובת התווית?

- ניתן לדעת, כתובת התווית היא 0x00000004
- ניתן לדעת, כתובת התווית היא 0x00008040
- ניתן לדעת, כתובת התווית היא 0x80400004
- ניתן לדעת, כתובת התווית היא 0x80400008
- ניתן לדעת, כתובת התווית היא 0x8040000c

שאלה 2

בעת הרצת הקוד, איזה מהמשפטים הבאים מסביר מה יקרה?

- יש בעיה בקוד - לא ניתן להבין מה קורה, במיוחד עקב הפקודה זן בשורה 18
- בפעם ראשונה שהקוד יגיע לפקודת זן (שורה 18) יבוצע דילוג לכתובת 0x80400004, ולמעשה תהיה לולאה אין סופית בין פקודת זן לבין פקודת ה-la (בשורה 15)
- בפעם ראשונה שהקוד יגיע לפקודת זן (שורה 18) יבוצע דילוג לכתובת 0x80400008, ובפעם השנייה שיגיע לפקודת זן יבצע דילוג לפקודה בכתובת 0x80400018 ויסיים את הקוד בפקודת syscall
- בפעם ראשונה שהקוד יגיע לפקודת זן (שורה 18) יבוצע דילוג לכתובת 0x8040000c, ובפעם השנייה שיגיע לפקודת זן יבצע דילוג לכתובת 0x80400014 ויכנס לולאה אין סופית
- בפעם ראשונה שהקוד יגיע לפקודת זן (שורה 18) יבוצע דילוג לכתובת 0x80400000, ובפעם השנייה שיגיע לפקודת זן יבצע דילוג לפקודה בכתובת 0x80400018 ויסיים את הקוד בפקודת syscall

שאלה 3

נתונים זמני הביצוע של חמשת השלבים בעבודת מעבד ה-MIPS:

IF – 550ps ID – 480ps EX – 520ps MEM – 550ps WB – 320ps

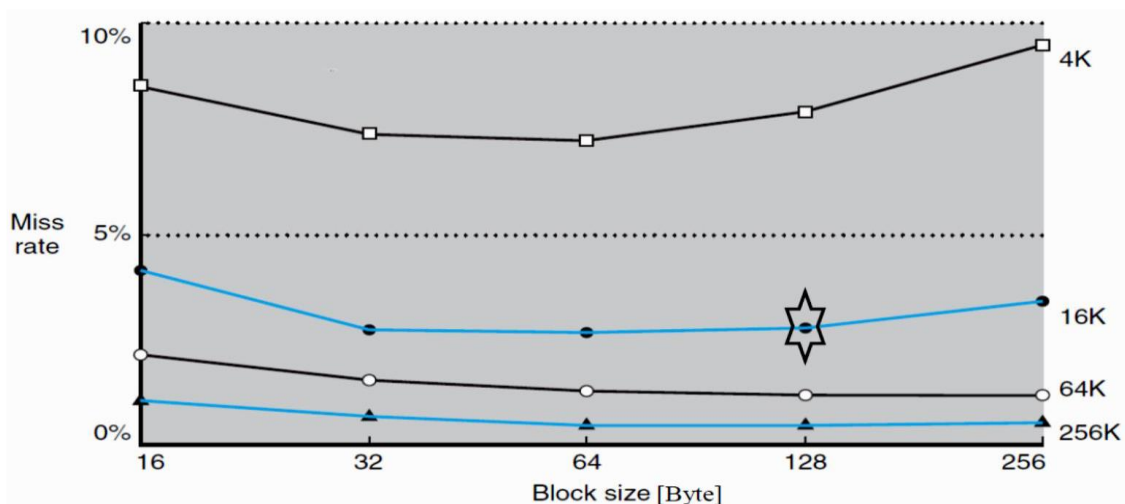
נתון שבמעבד צנרת יש תוספת ל CPI: עקב סיכוני נתונים יש תוספת של 0.8 cc/ins, ועקב סיכוני בקרה יש תוספת של 0.4 cc/ins. מעבר לזה אין השהיות נוספות.

מהו מדד ההאצה (speedup) בין מעבד העובד בטכנולוגית צנרת, לבין מעבד העובד באופן חד-מחזורי?

- א. 2
- ב. 0.5
- ג. 4.381818
- ד. 0.228216
- ה. 3.651515

שאלה 4

מצורף תרשים המתאר את שיעור ההחטאות (miss rate) בגישות למטמון כפונקציה של גודל הבלוק בביתים (bytes). כל קו מציין גודל מסוים של מטמון (גודל ה-data) הרשום מימין בהתאמה (4Kbytes, 16Kbytes, וכו').



בתרשים, מוקפת בכוכב נקודה המתארת זיכרון מטמון מסוים. על סמך הנתונים בתרשים, בהנחה שמדובר במטמון הממפה מרחב כתובות של **32 סיביות** בזיכרון הראשי במיפוי 2 ways set associative, מה גודל שדה התג?

- א. 6 סיביות
- ב. 7 סיביות
- ג. 18 סיביות
- ד. 19 סיביות

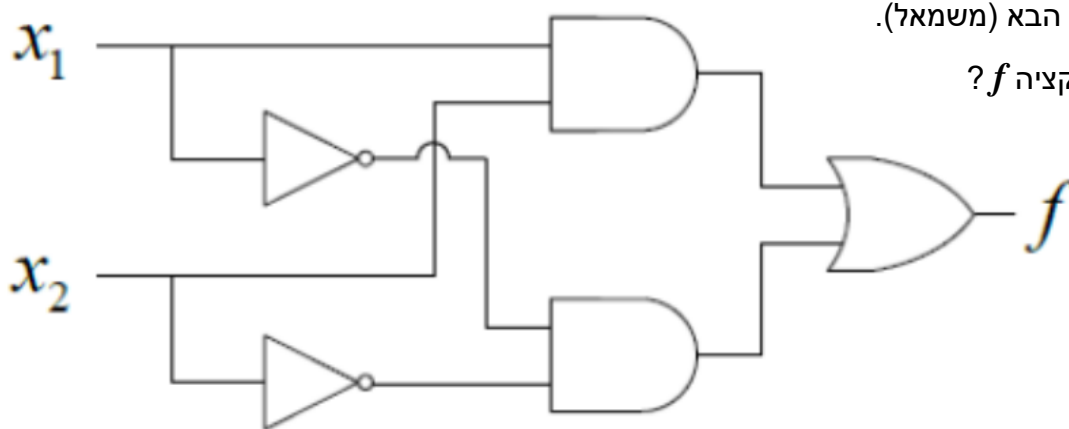
ה. לא ניתן לדעת, חסר הנתון של גודל שדה ה-index

שאלה 5

- בתרשים 4.60 בנספח לבחינה מופיע קו בשם **IF/ID Write** היוצא מיחידת ה HDU. מה תפקידו?
- לעכב את פקודת lw בזיהוי סיכון נתונים של $load\ use$
 - למנוע עדכון אוגר הצנרת IF/ID, ובכך הפקודה שבשלב ID תתבצע שוב במחזור שעון הבא
 - לבצע איפוס לאוגר הצנרת IF/ID, ובכך ליצור בועה בעקבות סיכון בקרה בצנרת
 - לבצע איפוס לאוגר הצנרת IF/ID, ובכך ליצור בועה בעקבות סיכון נתונים בצנרת
 - לא ניתן לדעת ללא פירוט של הפקודה עבודה יוצא ערך מיחידת ה HDU

שאלה 6

נתון המעגל הלוגי הבא (משמאל).
מה מחזירה הפונקציה f ?



- NOR
- XOR
- XNOR
- OR
- AND

שאלה 7

- על בסיס תרשים הצנרת הכולל את יחידות BTB ו MDU, מה תפקיד יחידת BTB?
- היחידה בודקת האם פקודת ה Branch עבדה בהתאם לחיזוי, ובמידה והייתה טעות דואגת לבצע flush ל- 3 פקודות לאחריה, ולהכניס ל- PC את כתובת הפקודה הנכונה
 - היחידה מזהה פקודות Branch. במידה זו פקודת branch: על בסיס טבלת Branch Table המכילה את נתון חיזוי של branch taken/not taken, מועברת הכתובת החזויה (ה Target PC או $4+PC$) ככתובת הפקודה הבאה אל ה PC. אחרת מחושב $4+PC$ כרגיל.
 - על בסיס טבלת Branch Table, המכילה את נתון חיזוי של branch taken/not taken מועבר מידע לאורך הצנרת, אל יחידת ה MDU כדי לתקן את החישוב של יחידה זו בשלב ה MEM, שיש טעות בחיזוי של ה MDU.
 - היחידה מזהה פקודות Branch, ובהתאם מעדכנת את יחידת ה MDU בתחזית שלה, וממתינה לשלב ה MEM בו נבדקת בפועל פקודת ה BEQ – ובהתאם מבצעת תיקונים בצנרת: הפיכת פקודת שגויות ל bubble ותיקון כתובת הפקודה הבאה ב PC.
 - כל התשובות (א' עד ד') נכונות

שאלה 8

בעת ביצוע פקודת SW מתרחש אירוע Page Fault בעת ניסיון הכתיבה ל Data Memory. איזה משפט מתאר באופן הנכון ביותר את מה שיתרחש במעבד. התבססו בתשובתכם על תרשים 4.66 בנספח החומרה:

א. זה אירוע exception לא מסונכרן, המתרחש בשלב MEM, וכתוצאה מכך יבוצע flush לפקודות SW ולפקודות שלפיה (משלב WB), ותיכנס כתובת 0x80000180, באמצעות מרבב, ל- PC כדי להגיע ל exception handler.

ב. זהו אירוע exception לא מסונכרן, וכתוצאה מכך המעבד יקרוס, והתוכנית תפסיק לעבוד.
ג. זה אירוע exception מסונכרן, המתרחש בשלב IF, וכתוצאה מכך יבוצע flush לפקודה באמצעות איפוס אוגר IF/ID, ותיכנס כתובת 0x80000180, באמצעות מרבב, ל- PC כדי להגיע ל exception handler.

ד. זה אירוע exception מסונכרן, המתרחש בשלב EXE, וכתוצאה מכך יבוצע flush לפקודות משלבים EXE, ID ו- IF, תיכנס כתובת 0x80000180, באמצעות מרבב, ל- PC כדי להגיע ל exception handler ואוגר EPC יקבל את הכתובת של פקודת SW כדי לחזור אליה בסיום הטיפול ב exception handler.

ה. זה אירוע exception מסונכרן, המתרחש בשלב MEM, וכתוצאה מכך יבוצע flush לפקודות SW ולפקודות שאחריה (משלבים EXE, ID ו- IF), ותיכנס כתובת 0x80000180, באמצעות מרבב, ל- PC כדי להגיע ל exception handler.

חלק ב' – שאלות פתוחות (68 נקודות)

בחלק זה 2 שאלות פתוחות.

יש לענות על השאלות בטופס המבחן, במקום המיועד לתשובות. שימו לב: הקפידו לענות על כל שאלה עם תשומת לב שאתם כותבים את תשובתכם במקום המיועד בהתאם לסעיף.

שאלה 9 – מעבד חד-מחזורי ומעבד צנרת (48 נקודות)

קטע הקוד הבא הינו קטע קוד בשימוש כל סעיפי השאלה. לפני שאתם מתעמקים בקטע הקוד הבא, קראו גם את שאר סעיפי השאלה, במיוחד את ההתייחסות לשינויי החומרה המתוארים. לאחר שתבינו גם את שינויי החומרה(!), עברו על הקוד והבינו מה קורה בקוד בהתייחסות לשינויי החומרה המתוארים.

נתון קטע הקוד שלהלן באסמבלי של ה- MIPS

Address	Code	Basic	Source
0x80400000	0x3c0a0002	lui \$10,0x00000002	1 lui \$10, 0x0002
0x80400004	0x34aa0100	ori \$10,\$5,0x00000100	2 ori \$10, \$5, 0x0100
0x80400008	0x00055380	sll \$10,\$5,0x0000000e	3 sll \$10, \$5, 14
0x8040000c	0x8cb0000c	lw \$16,0x0000000c(\$5)	4 lw \$16, 0xc(\$5)
0x80400010	0xacb00000	sw \$16,0x00000000(\$5)	5 sw \$16, 0x0(\$5)
0x80400014	0x22110004	addi \$17,\$16,0x00000004	6 addi \$17, \$16, 4

בתחילת ביצוע הקוד: נתון שערכו של אוגר \$5 שווה 0x10010000. ערך שאר האוגרים אינו ידוע.

נתון כי באזור (סגמנט) הנתונים (.data) – ערך כל מילה (word) הינו כתובת המילה בזיכרון. דוגמאות:

• ערך המילה הנמצאת בכתובת 0x10010050 – הוא 0x10010050

• ערך המילה הנמצאת בכתובת 0x10010280 – הוא 0x10010280

בסגמנט הקוד, ערך המילים הוא Code הפקודות עצמן.

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

9.1 מעבד חד מחזורי (24 נקודות)

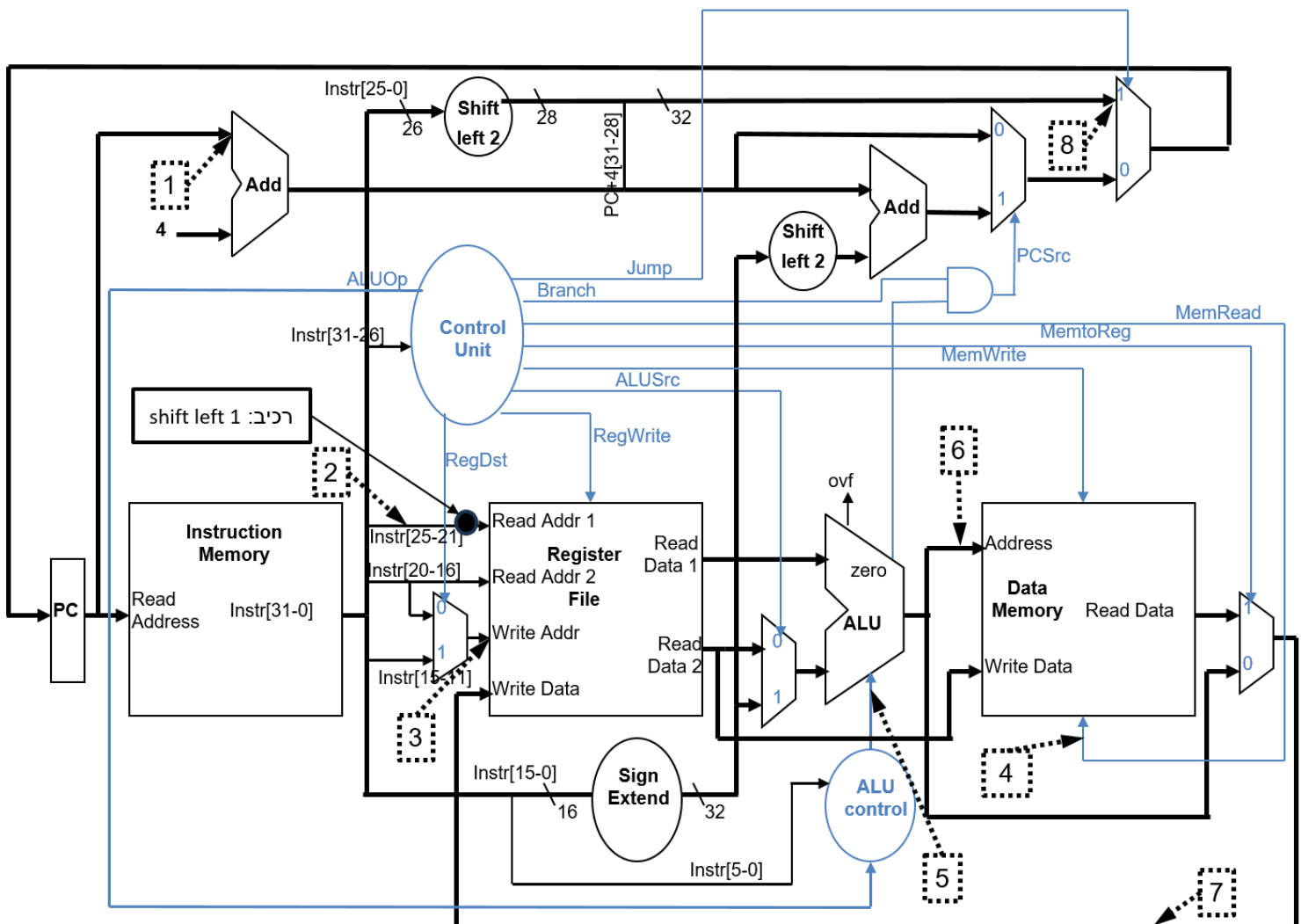
בתרשים שלהלן מעבד חד-מחזורי (המבוסס על תרשים 4.24). למעבד שבתרשים **נוסף רכיב של shift left 1**. הרכיב נמצא על הקו של Instr[21-25] שבין זיכרון הפקודות (Instruction Memory) לבין מקבץ האוגרים (Register File). הרכיב מבצע הזזה של סיבית אחת שמאלה ומכניס 0 מימין. מיקום הרכיב מסומן באמצעות עיגול שחור (●) (באופן בולט על גבי קו זה (בתוספת חץ המצביע על המקום)).

על התרשים מסומנות 8 נקודות (הממוספרות מ-1 עד 8). עליכם לסמן **בטבלה שבתחילת העמוד הבא** את הערכים העוברים על הקווים המסומנים **בבסיס הקסה** (3 נקודות לסימון), בהנחה שהמידע נבדק לקראת סוף פעימות השעון בביצוע הפקודה הרביעית בקוד:

lw \$16, 0xc(\$5)

ערך לא ידוע יש לסמן ב-X.

ניתן להיעזר בנתונים מטבלאות 4.12 ו-4.18, וב- "כרטיס הירוק" – בנספח החומרה למבחן.



← סעיף 9.1 (24 נקודות, 3 נקודות לכל סימון) – רשמו את הערכים בהקסה(!)

1	
2	
3	
4	

5	
6	
7	
8	

9.2 מעבד צנרת (18 נקודות, 3 נקודות לכל סימון)

בתרשים, שבעמוד הבא, מסומנות 6 נקודות (1-6). עליכם לסמן בטבלה שלהלן את הערכים העוברים על הקווים המסומנים בבסיס הקסה (3 נקודות לסימון), בהנחה שהמידע נבדק לקראת סוף פעימות השעון השישית (6):

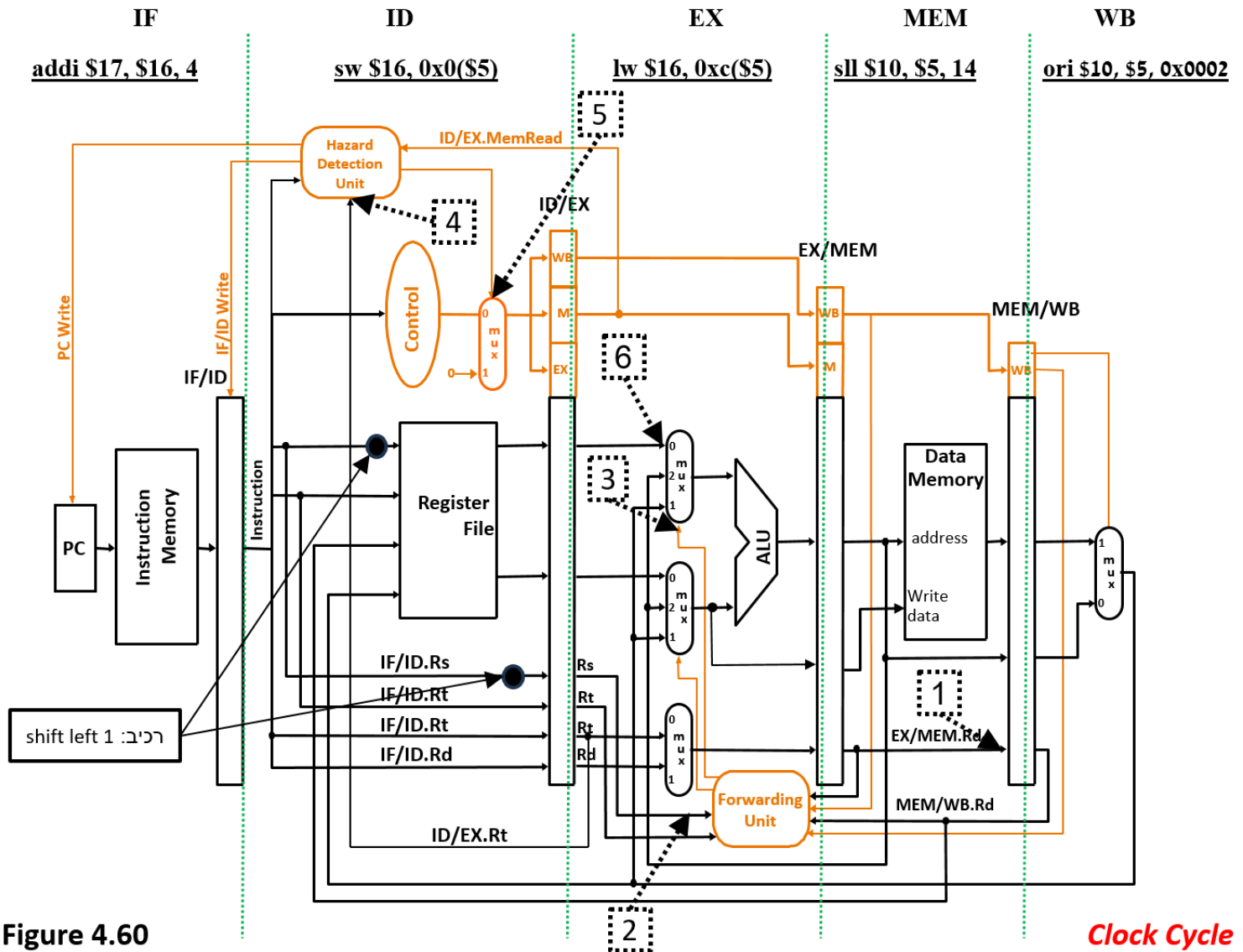
רשמו את הערכים בהקסה(!)

ערך לא ידוע יש לסמן ב-X.

1	
2	
3	

4	
5	
6	

התרשים שלפניכם, מבוסס על תרשים 4.60 בספר. בהתאמה לשאלה בסעיף קודם (9.1), לתרשים זה נוספו שני רכיבי shift left 1 על קווי rs בשלב ה-ID. מיקומי הרכיבים שהתווספו מסומנים באמצעות עיגול שחור (●) באופן בולט על גבי קווים אלה (בתוספת חץ המצביע על המקום).



בתרשים מתוארת פעימת השעון השישית בביצוע הקוד שבשאלה, על בסיס הנתונים המופיעים בתחילת השאלה. בפעימת שעון זו, פקודת ORI (הפקודה השנייה) נמצאת בשלב WB.

סיכוני הנתונים בהרצת הקוד נפתרים על ידי יחידת העברה קדימה (Forwarding Unit) ויחידת איתור הסיכונים (HDU), וחציית מקבץ האוגרים. שימו לב להשפעת הרכיב הנוסף על יחידות אלה.

9.3 מעבד צנרת (3 נקודות)

בהמשך לסעיף הקודם (9.2 מעבד הצנרת, במחזור שעון 6), רשמו בטופס התשובות את מיקום הפקודות שבקוד, לפי שלבי העבודה של מעבד הצנרת בתחילת מחזור שעון 7. כלומר, מחזור שעון אחד אחרי מחזור השעון המתואר בסעיף 9.2

יש לרשום, בטבלה כאן, לכל שלב בצנרת: **איזו פקודה** תהיה בכל שלב במחזור שעון 7. יש להוסיף הסבר קצר (2-3 משפטים) המסביר מדוע זה מיקום הפקודות בצנרת

שלב 1 - IF	שלב 2 - ID	שלב 3 - EXE	שלב 4 - MEM	שלב 5 - EX
				sl \$10, \$5, 14

9.4 מעבד צנרת (3 נקודות)

בהמשך לסעיפים 9.2 ו-9.3, עם התייחסות לרכיב ה shift left 1 כפי שמופיע בתרשים שבסעיף 9.2, רשמו מה יהיה ערכו של אוגר 17 עם סיום פקודת ה ADDI שבשורה 6 של הפקודות. יש לרשום את ערכו של אוגר 17 בסיום ביצוע הפקודה, בהקסה, בתוספת הסבר קצר.

שאלה 10 (20 נקודות)

נתונה פרוצדורה בשם whatDoldo. הפרוצדורה מקבלת 2 פרמטרים:

- \$a0: כתובת של מחרוזת שהוגדרה במקטע הנתונים (.data). כפי שמתואר מיד בהמשך השאלה.

- \$a1: קוד ascii של תו

במקטע הנתונים (.data) נתונה ההגדרה הבאה של המחרוזת **str**:

str: .asciiz "ALBCLLEBLLERL"

בתוכנית הראשית – אופן זימון הפרוצדורה:

```
la $a0, str
li $a1, 'L'
jal whatDoldo
```

קוד הפרוצדורה:

```
1  whatDoldo:
2  # $a0 – null terminated string address, $a1 – ascii code of a character
3      move $t9, $a0
4      move $t8, $a1
5      xor $v0, $v0, $v0
6  loopA:
7      lbu $t0, 0($t9)
8      beq $t0, $zero, endLoopA
9      bne $t0, $t8, cont1
10     addi $v0, $v0, 1
11     move $a0, $t9
12  loopB:
13     lb $t0, 1($a0)
14     sb $t0, 0($a0)
15     addi $a0, $a0, 1
16     bne $t0, $zero, loopB
17     j loopA
18  cont1:
19     add $t9, $t9, 1
20     j loopA
21  endLoopA:
22     jr $ra
```

בהמשך, ראה התייחסות בסעיף 10.4 (*) #

לפני שאתם עונים על השאלות, ודאו שהבנתם היטב את:

- אופן הגדרת המחרוזת במקטע הנתונים (.data)
- אופן זימון הפרוצדורה מהתוכנית הראשית
- מימוש הפרוצדורה

לאחר מכן, יש לענות על השאלות הבאות, **המתייחסות לקוד בפרוצדורה**, במקום המתאים בטופס התשובות בהתאם לסעיפי השאלה:

10.1 (5 נקודות)

עם סיום עבודת הפרוצדורה, מה יהיה הערך באוגר \$v0.
הוסיפו הסבר קצר: מה משמעות ערך זה?

10.2 (5 נקודות)

עם סיום עבודת הפרוצדורה, מה תהיה המחרוזת כפי שתופיע במקטע הנתונים (.data) בזיכרון עבור המחרוזת **str**.

הוסיפו הסבר קצר: מה ביצעה הפרוצדורה אשר השפיע, או שלא השפיע, על המחרוזת כפי שכתבתם.

10.3 (5 נקודות)

הסבירו בקצרה מה מבצעת הפרוצדורה.

שימו לב: יש להסביר מה עושה הפרוצדורה, ולא להסביר את הפקודות והלולאות ואיך הפרוצדורה עובדת. ההסבר שאתם נדרשים לתאר הינו הסבר של מה מטרת הפרוצדורה בהיבט הפונקציונאלי שלה. הסבר טכני של "כיצד עובדת" הפרוצדורה לא יתקבל!

10.4 (5 נקודות)

כעת, מבוצע שינוי בפרוצדורה. **שורה 17** בקוד (מודגשת, בתוספת הערה לשימוש בסעיף זה) – נמחקת מהקוד. עיינו בקוד שוב, בהנחה ששורה זו לא מתבצעת (כלומר, נמחקה), וענו על השאלות הבאות:

א. האם הפרוצדורה תעבוד באופן תקין? הסבירו ונמקו את התשובה שלכם (שימו לב, אין צורך להתייחס בסעיף זה לתוצאות – אלא להתמקד בהתייחסות לתקינות עבודת הפרוצדורה)

ב. בעקבות השינוי: עם סיום עבודת הפרוצדורה, מה תהיה המחרוזת כפי שתופיע במקטע הנתונים (.data). בזיכרון עבור המחרוזת **str**. מה יהיה הערך המוחזר באוגר **\$v0**?

ג. הסבירו מדוע השינוי בקוד הפרוצדורה השפיע באופן זה

הנחייה כללית לכל המבחן:

- יש לענות על **כל השאלות** בטופס השאלון בלבד. רק טופס השאלון ייסרק ויבדק!
- הקפידו לענות במקומות המסומנים למענה על השאלות.
- כל רישום שלא במקום המיועד לתשובות לא ייבדק. רק התשובות במקום המיועד - קובעות!
- הקפידו על רישום התשובות באופן מלא ומסודר בהתאם להנחיות!

חלק א: (32 נקודות)

בחלק זה 8 שאלות.

יש לבחור את התשובה הנכונה מבין האפשרויות.

משקל כל שאלה 4 נקודות.

יש לסמן X באופן ברור בטבלה

הערה: יש לסמן רק אפשרות אחת לכל שאלה!

בדקו בקפדנות ובזהירות את הרישום בטבלה שבטופס התשובות, בכדי להימנע מטעויות העתקה ורישום.

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

נתונה תמונת סגמנט הקוד של MIP32, הלקוחה מה MARS איתו תרגלנו בקורס. ענו על שתי שאלות ראשונות (1-2) בהתבסס על סגמנט קוד זה.

Address	Code	Basic	Source
0x80400000	0x240b0006	addiu \$11,\$0,0x00000000	13 li \$t3, 6
0x80400004	0x3c010040	lui \$1,0x00008040	15 la \$a0, unknown
0x80400008	0x34240004	ori \$4,\$1,0x00000004	
0x8040000c	0x01645825	or \$11,\$11,\$4	16 or \$t3, \$t3, \$a0
0x80400010	0x20840008	addi \$4,\$4,0x00000008	17 addi \$a0, \$a0, 0x08
0x80400014	0x00800008	jr \$4	18 jr \$a0
0x80400018	0x2402000a	addiu \$2,\$0,0x0000000a	19 li \$v0, 10
0x8040001c	0x0000000c	syscall	20 syscall

שאלה 1

האם ניתן לדעת מה כתובת התווית unknown, אליה יש התייחסות בשורה 15 של ה-source? במידה וכן, מה כתובת התווית?

א. ניתן לדעת, כתובת התווית היא 0x00000004

ב. ניתן לדעת, כתובת התווית היא 0x00008040

ג. **ניתן לדעת, כתובת התווית היא 0x80400004**

ד. ניתן לדעת, כתובת התווית היא 0x80400008

ה. ניתן לדעת, כתובת התווית היא 0x8040000c

שאלה 2

בעת הרצת הקוד, איזה מהמשפטים הבאים מסביר מה יקרה?

א. יש בעיה בקוד - לא ניתן להבין מה קורה, במיוחד עקב הפקודה זן בשורה 18

ב. בפעם ראשונה שהקוד יגיע לפקודת זן (שורה 18) יבוצע דילוג לכתובת 0x80400004, ולמעשה תהיה לולאה אין סופית בין פקודת זן לבין פקודת ה-la (בשורה 15)

ג. בפעם ראשונה שהקוד יגיע לפקודת זן (שורה 18) יבוצע דילוג לכתובת 0x80400008, ובפעם השנייה שיגיע לפקודת זן יבצע דילוג לפקודה בכתובת 0x80400018 ויסיים את הקוד בפקודת ה-syscall

ד. **בפעם ראשונה שהקוד יגיע לפקודת זן (שורה 18) יבוצע דילוג לכתובת 0x8040000c, ובפעם השנייה שיגיע לפקודת זן יבצע דילוג לכתובת 0x80400014 ויכנס לולאה אין סופית**

ה. בפעם ראשונה שהקוד יגיע לפקודת זן (שורה 18) יבוצע דילוג לכתובת 0x80400000, ובפעם השנייה שיגיע לפקודת זן יבצע דילוג לפקודה בכתובת 0x80400018 ויסיים את הקוד בפקודת ה-syscall

שאלה 3

נתונים זמני הביצוע של חמשת השלבים בעבודת מעבד ה-MIPS:

IF – 550ps ID – 480ps EX – 520ps MEM – 550ps WB – 320ps

נתון שבמעבד צנרת יש תוספת ל CPI: עקב סיכוני נתונים יש תוספת של 0.8 cc/ins, ועקב סיכוני בקרה יש תוספת של 0.4 cc/ins. מעבר לזה אין השהיות נוספות.

מהו מדד ההאצה (speedup) בין מעבד העובד בטכנולוגית צנרת, לבין מעבד העובד באופן חד-מחזורי?

א. 2

ב. 0.5

ג. 4.381818

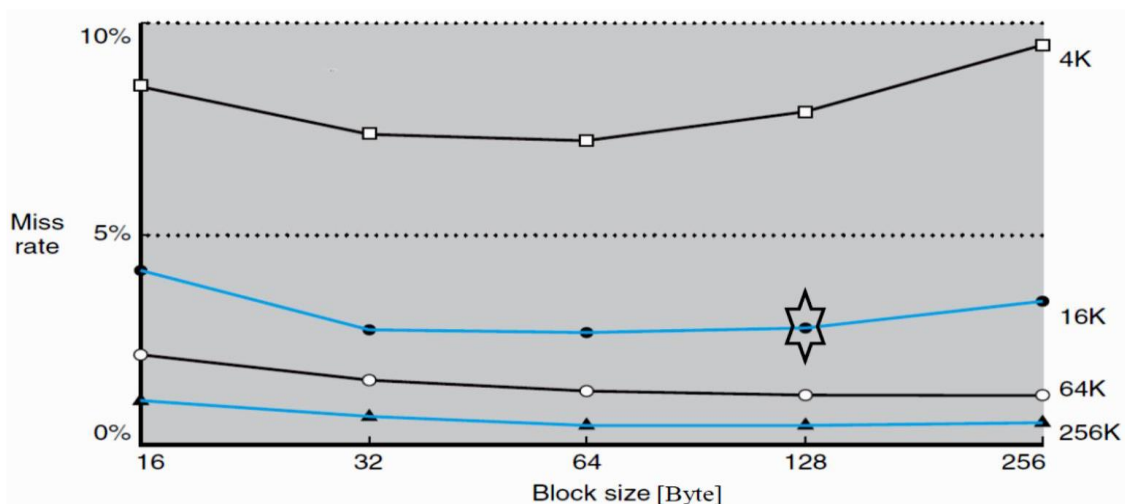
ד. 0.228216

ה. 3.651515

- מעבד חד מחזורי: זמן מחזור שעות (cc) $550+480+520+550+320 = 2420ps$
- מעבד צנרת: זמן מחזור שעות (cc) הינו לפי הארוך ביותר: 550ps
- נתון שעקב סיכוני הנתונים – יש תוספת למחזור השעות של 1.2 מחזורי שעות (כאשר לוקחים בחשבון את שני סוגי הסיכונים המפורטים), כלומר תוספת של $550 \cdot 1.2$ ל- 550ps, ולכן בסיכום – בצנרת משך ביצוע פקודה הוא: $550 + 1.2 \cdot 550 = 550 + 660 = 1210$
- כעת נחשב את ה speedup: $2420 / 1210 = 2$

שאלה 4

מצורף תרשים המתאר את שיעור ההחטאות (miss rate) בגישות למטמון כפונקציה של גודל הבלוק בביתים (bytes). כל קו מציין גודל מסוים של מטמון (גודל ה- data) הרשום מימין בהתאמה (4Kbytes, 16Kbytes, וכו').



בתרשים, מוקפת בכוכב נקודה המתארת זיכרון מטמון מסוים. על סמך הנתונים בתרשים, בהנחה שמדובר במטמון הממפה מרחב כתובות של **32 סיביות** בזיכרון הראשי במיפוי 2 ways set associative, מה גודל שדה התג?

א. 6 סיביות

ב. 7 סיביות

ג. 18 סיביות

ד. 19 סיביות

- גודל המטמון הנתון הוא 16K, כלומר $2^{14} \text{ bytes} = 2^4 \cdot 2^{10}$
- גודל בלוק הוא 128bytes, כלומר 2^7 bytes
- מכאן, שיש במטמון 2^7 בלוקים
- מאחר ונתון שיש 2 way set associative, המשמעות, הינה בכל way יש $2^6 = 2^7 / 2$ בלוקים (כלומר – שורות בכל way).
- כעת: 2 סיביות – byte offset, 5 סיביות block offset, 6 סיביות index, ומכאן – יש 19 סיביות ב-tag:
- $32 - 2 - 5 - 6 = 32 - 13 = 19$

ה. לא ניתן לדעת, חסר הנתון של גודל שדה ה- index

שאלה 5

בתרשים 4.60 בנספח לבחינה מופיע קו בשם **IF/ID Write** היוצא מיחידת ה HDU. מה תפקידו?

א. לעכב את פקודת lw בזיהוי סיכון נתונים של $load\ use$

ב. למנוע עדכון אוגר הצנרת IF/ID, ובכך הפקודה שבשלב ID תתבצע שוב במחזור שעון הבא

ג. לבצע איפוס לאוגר הצנרת IF/ID, ובכך ליצור בועה בעקבות סיכון בקרה בצנרת

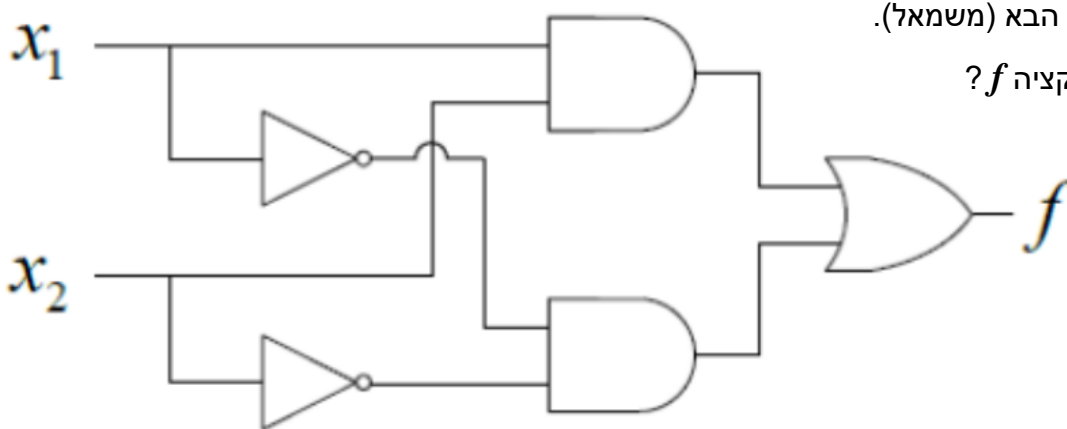
ד. לבצע איפוס לאוגר הצנרת IF/ID, ובכך ליצור בועה בעקבות סיכון נתונים בצנרת

ה. לא ניתן לדעת ללא פירוט של הפקודה עברה יוצא ערך מיחידת ה HDU

שאלה 6

נתון המעגל הלוגי הבא (משמאל).

מה מחזירה הפונקציה f ?



א. NOR

ב. XOR

ג. XNOR

ד. OR

ה. AND

שאלה 7

על בסיס תרשים הצנרת הכולל את יחידות BTB ו MDU, מה תפקיד יחידת BTB?

א. היחידה בודקת האם פקודת ה Branch עבדה בהתאם לחיזוי, ובמידה והייתה טעות דואגת

לבצע flush ל- 3 פקודות לאחריה, ולהכניס ל- PC את כתובת הפקודה הנכונה

ב. היחידה מזהה פקודות Branch. במידה זו פקודת branch: על בסיס טבלת Branch Table

המכילה את נתון חיזוי של branch taken/not taken, מועברת הכתובת החזויה (ה Target

PC או $4+PC$) ככתובת הפקודה הבאה אל ה PC. אחרת מחושב $4+PC$ כרגיל.

ג. על בסיס טבלת Branch Table, המכילה את נתון חיזוי של branch taken/not taken מועבר

מידע לאורך הצנרת, אל יחידת ה MDU כדי לתקן את החישוב של יחידה זו בשלב ה MEM,

שיש טעות בחיזוי של ה MDU.

ד. היחידה מזהה פקודות Branch, ובהתאם מעדכנת את יחידת ה MDU בתחזית שלה, וממתינה

לשלב ה MEM בו נבדקת בפועל פקודת ה BEQ – ובהתאם מבצעת תיקונים בצנרת: הפיכת

פקודת שגויות ל bubble ותיקון כתובת הפקודה הבאה ב PC.

ה. כל התשובות (א' עד ד') נכונות

שאלה 8

בעת ביצוע פקודת SW מתרחש אירוע Page Fault בעת ניסיון הכתיבה ל Data Memory. איזה משפט מתאר באופן הנכון ביותר את מה שיתרחש במעבד. התבססו בתשובתכם על תרשים 4.66 בנספח החומרה:

א. זה אירוע exception לא מסונכרן, המתרחש בשלב MEM, וכתוצאה מכך יבוצע flush לפקודות SW ולפקודות שלפיה (משלב WB), ותיכנס כתובת 0x80000180, באמצעות מרבב, ל- PC כדי להגיע ל exception handler.

ב. זהו אירוע exception לא מסונכרן, וכתוצאה מכך המעבד יקרוס, והתוכנית תפסיק לעבוד.
ג. זה אירוע exception מסונכרן, המתרחש בשלב IF, וכתוצאה מכך יבוצע flush לפקודה באמצעות איפוס אוגר IF/ID, ותיכנס כתובת 0x80000180, באמצעות מרבב, ל- PC כדי להגיע ל exception handler.

ד. זה אירוע exception מסונכרן, המתרחש בשלב EXE, וכתוצאה מכך יבוצע flush לפקודות משלבים EXE, ID ו- IF, תיכנס כתובת 0x80000180, באמצעות מרבב, ל- PC כדי להגיע ל exception handler ואוגר EPC יקבל את הכתובת של פקודת SW כדי לחזור אליה בסיום הטיפול ב exception handler.

ה. זה אירוע exception מסונכרן, המתרחש בשלב MEM, וכתוצאה מכך יבוצע flush לפקודות SW ולפקודות שאחריה (משלבים EXE, ID ו- IF), ותיכנס כתובת 0x80000180, באמצעות מרבב, ל- PC כדי להגיע ל exception handler.

חלק ב' – שאלות פתוחות (68 נקודות)

בחלק זה 2 שאלות פתוחות.

יש לענות על השאלות בטופס המבחן, במקום המיועד לתשובות. שימו לב: הקפידו לענות על כל שאלה עם תשומת לב שאתם כותבים את תשובתכם במקום המיועד בהתאם לסעיף.

שאלה 9 – מעבד חד-מחזורי ומעבד צנרת (48 נקודות)

קטע הקוד הבא הינו קטע קוד בשימוש כל סעיפי השאלה. לפני שאתם מתעמקים בקטע הקוד הבא, קראו גם את שאר סעיפי השאלה, במיוחד את ההתייחסות לשינויי החומרה המתוארים. לאחר שתבינו גם את שינויי החומרה (!), עברו על הקוד והבינו מה קורה בקוד בהתייחסות לשינויי החומרה המתוארים.

נתון קטע הקוד שלהלן באסמבלי של ה-MIPS

Address	Code	Basic	Source
0x80400000	0x3c0a0002	lui \$10,0x00000002	1 lui \$10, 0x0002
0x80400004	0x34aa0100	ori \$10,\$5,0x00000100	2 ori \$10, \$5, 0x0100
0x80400008	0x00055380	sll \$10,\$5,0x0000000e	3 sll \$10, \$5, 14
0x8040000c	0x8cb0000c	lw \$16,0x0000000c(\$5)	4 lw \$16, 0xc(\$5)
0x80400010	0xacb00000	sw \$16,0x00000000(\$5)	5 sw \$16, 0x0(\$5)
0x80400014	0x22110004	addi \$17,\$16,0x00000004	6 addi \$17, \$16, 4

בתחילת ביצוע הקוד: נתון שערכו של אוגר \$5 שווה 0x10010000. ערך שאר האוגרים אינו ידוע.

נתון כי באזור (סגמנט) הנתונים (.data) – ערך כל מילה (word) הינו כתובת המילה בזיכרון. דוגמאות:

- ערך המילה הנמצאת בכתובת 0x10010050 – הוא 0x10010050
- ערך המילה הנמצאת בכתובת 0x10010280 – הוא 0x10010280

בסגמנט הקוד, ערך המילים הוא Code הפקודות עצמן.

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

מעקב אחר הקוד הנ"ל (בהתייחסות מלאה לתוספת חומרה כפי שמתואר בסעיף 9.1 ו-9.2)

1. $10 = 0x00020000$
2. $10 = 0x00020100$ אוגר rs=5, עקב רכיב החומרה, בפועל הוא 10, ולכן החישוב על בסיס אוגר 10
3. $10 = 0x80400000$ אוגר rs=5, עקב רכיב החומרה, בפועל הוא 10, ולכן החישוב על בסיס אוגר 10
4. $16 = 0x8cb0000c$ אוגר rs=5, עקב רכיב החומרה, בפועל הוא 10. חישוב הכתובת אליה ניגש LW מבוסס בפועל על ערכו של אוגר 10 לאחר סעיף 3, בתוספת 0xc, כלומר כתובת 0x8040000c. כתובת זו בפועל הינה הכתובת של פקודת ה-LW עצמה, ולכן התוצאה הינה הערך הנמצא בכתובת זו. הערך בפועל זה ה-code של הפקודה כפי שמופיע בעמודת ה-code אין שינוי בערכי אוגרים. ערכו של אוגר 16 (למעשה code של פקודת LW) ייכתב בזיכרון בכתובת 0x80400000 (אוגר rs-5, עקב רכיב החומרה, ומכאן בפועל הוא 10, בתוספת 0x0 ולכן זו הכתובת) – שזו כתובת פקודה 1-LUI, והיא תוחלף בפועל בפקודת LW (עותק של פקודה 4)
6. $17 = 4$: אוגר rs=16, עקב רכיב החומרה, לאחר הזזה מקום אחד שמאלה הוא לכאורה שווה 32, אך בפועל – עקב העובדה שיש רק 5 סיביות, זה יהיה 0 (כדי לבטא 32 צריך 6 סיביות) – ואז אוגר 4 + 0 זה 4 ... 4

9.1 מעבד חד מחזורי (24 נקודות)

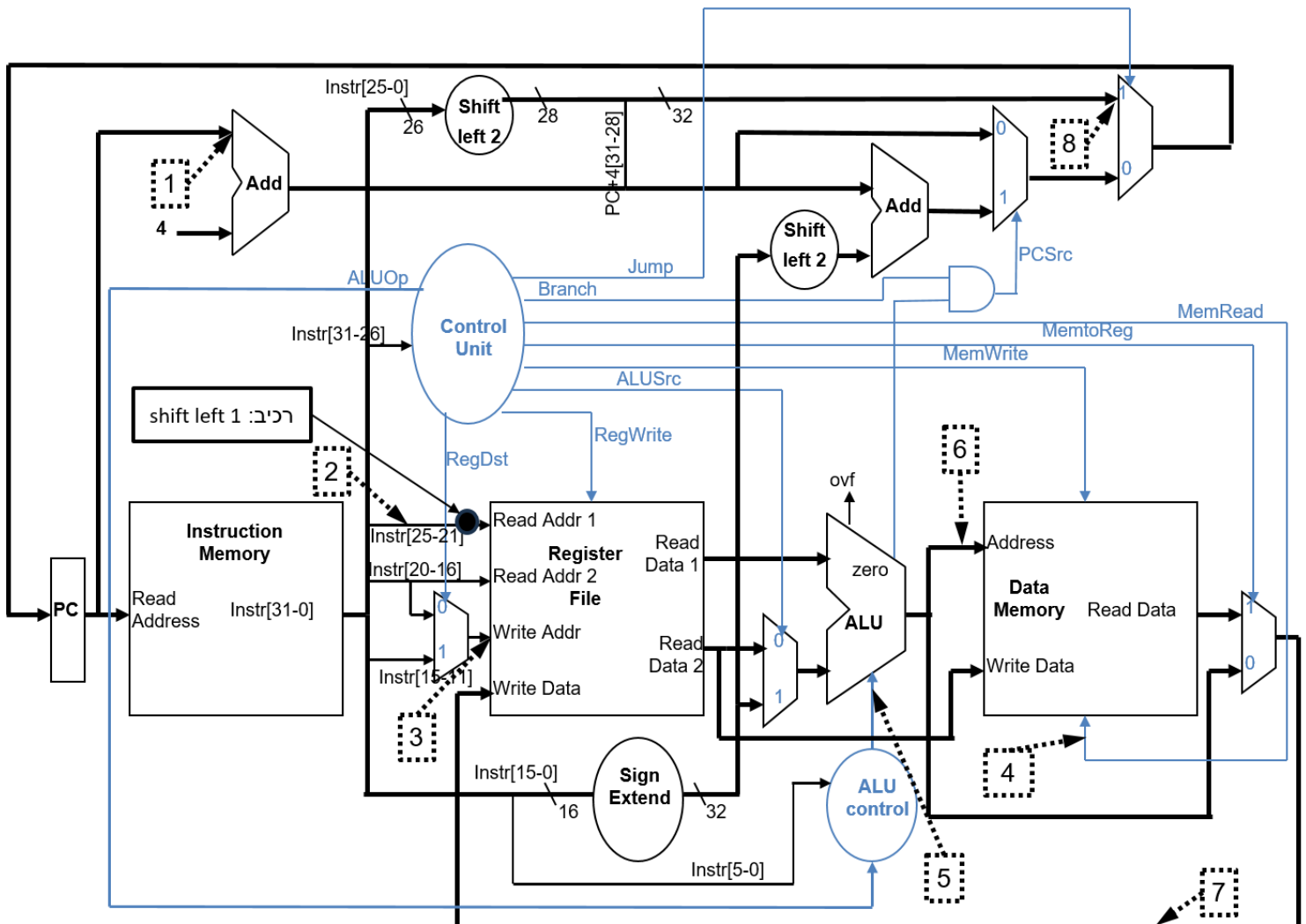
בתרשים שלהלן מעבד חד-מחזורי (המבוסס על תרשים 4.24). למעבד שבתרשים **נוסף רכיב של shift left 1**. הרכיב נמצא על הקו של Instr[21-25] שבין זיכרון הפקודות (Instruction Memory) לבין מקבץ האוגרים (Register File). הרכיב מבצע הזזה של סיבית אחת שמאלה ומכניס 0 מימין. מיקום הרכיב מסומן באמצעות עיגול שחור (●) (בתוספת חץ המצביע על המקום).

על התרשים מסומנות 8 נקודות (הממוספרות מ-1 עד 8). עליכם לסמן **בטבלה שבתחילת העמוד הבא** את הערכים העוברים על הקווים המסומנים **בבסיס הקסה** (3 נקודות לסימון), בהנחה שהמידע נבדק לקראת סוף פעימות השעון בביצוע הפקודה הרביעית בקוד:

lw \$16, 0xc(\$5)

ערך לא ידוע יש לסמן ב-X.

ניתן להיעזר בנתונים מטבלאות 4.12 ו-4.18, וב- "כרטיס הירוק" – בנספח החומרה למבחן.



← סעיף 9.1 (24 נקודות, 3 נקודות לכל סימון) – רשמו את הערכים בהקסה (!)

1	כתובת הפקודה: 0x804000c	5	ערך alu control לפי טבלה 4.12: 0x2 (0010 bin)
2	מספר אוגר rs לפני ה- shift: 0x5	6	לפי פקודות 1-3, אוגר 10 מכיל 0x80400000, אוגר rs הוא 5, ועקב shift בפועל 10 – ולכן בתוספת הקבוע: 0x8040000c
3	מספר אוגר rt – אוגר מטרה: 0xA	7	הכתובת מצביעה למעשה על פקודת ה LW עצמה, ולכן הערך שם הוא <u>קוד</u> פקודת LW והוא: 0x8cb0000c
4	דגל memRead – 0x1	8	בפירוק סיביות 0-25 של פקודת LW, הזזה שמלאה 2 מקומות, והדבקת 0x8 משמאל נקבל: 0x82c00030

9.2 מעבד צנרת (18 נקודות, 3 נקודות לכל סימון)

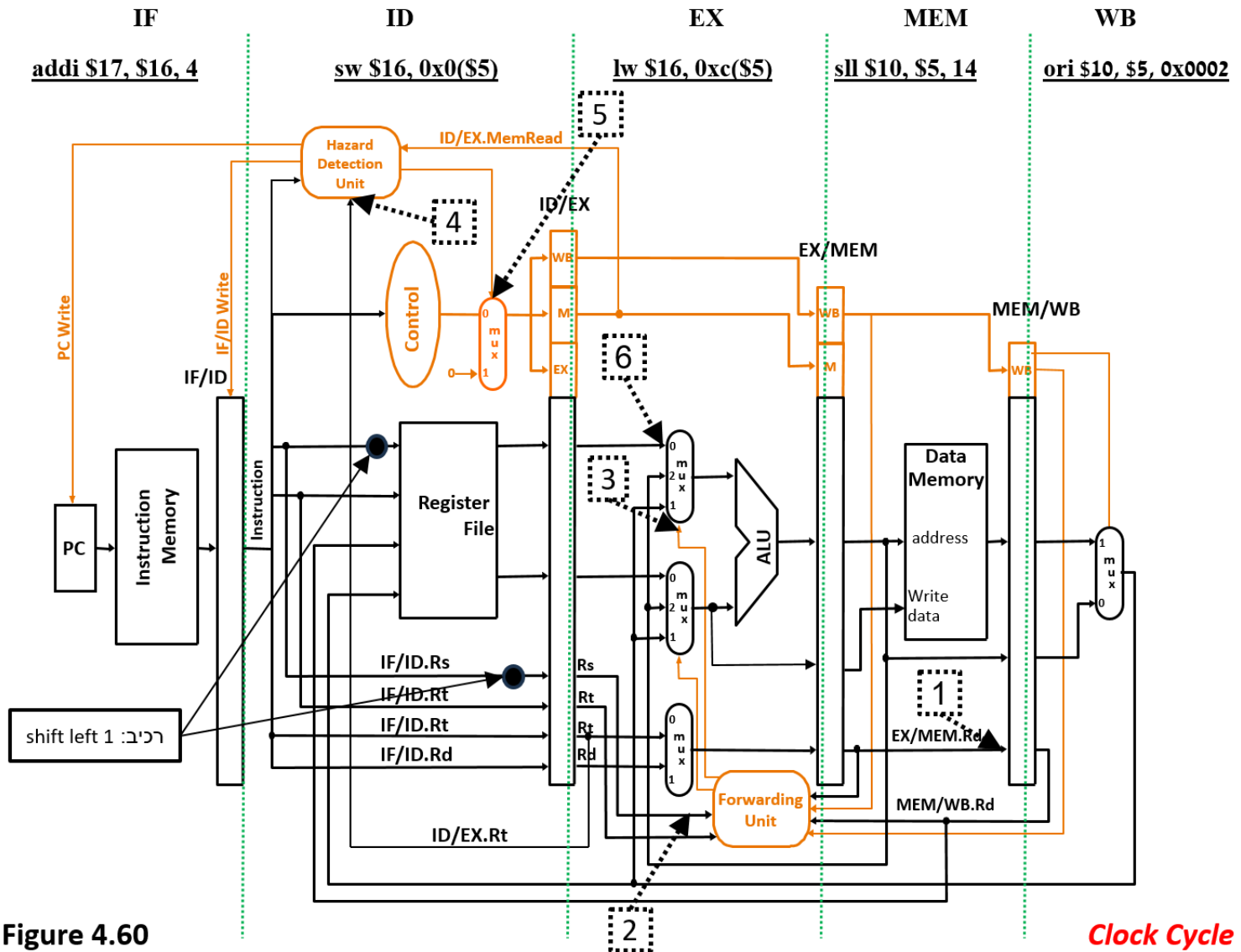
בתרשים, שבעמוד הבא, מסומנות 6 נקודות (1-6). עליכם לסמן בטבלה שלהלן את הערכים העוברים על הקווים המסומנים בבסיס הקסה (3 נקודות לסימון), בהנחה שהמידע נבדק לקראת סוף פעימות השעון השישית (6):

רשמו את הערכים בהקסה (!)

ערך לא ידוע יש לסמן ב- X.

1	0xA – מספר אוגר rt (מטרה) בפקודת SLL	4	0x10 (16) – מספר אוגר המטרה של פקודת LW (מספר אוגר rt) – ליחידת HDU
2	0xA – מספר אוגר rs של פקודת LW לאחר שבוצע shift למספר האוגר בשלב ה ID	5	0x1 – יחידת HDU תזהה שימוש באוגר 16 בפקודת LW כאוגר מטרה, ופקודת SW כאוגר מקור, ולכן נדרש bubble במחזור שעון 7
3	0x2 – יחידת forwarding unit מזהה את מקרה ה 1a לקידום הערך בין EX/MEM ולפי טבלת מקרי הקידום – הערך הוא 2	6	0x00020000 – זה ערך אוגר 10 לאחר ביצוע פקודת LUI (הראשונה בקוד). במחזור שעון 5 פקודת LUI הייתה בשלב WB, פקודת LW הייתה בשלב ID, ועקב חציית מקבץ האוגרים הערך של אוגר rs לפקודת LW הוא תוצאת LUI. במחזור שעון 6 זהו ערכו של הערך לפני ה-MUX ולפני שמבוצע קידום נתון משלב MEM

התרשים שלפניכם, מבוסס על תרשים 4.60 בספר. בהתאמה לשאלה בסעיף קודם (9.1), לתרשים זה נוספו שני רכיבי shift left 1 על קווי rs בשלב ה-ID. מיקומי הרכיבים שהתווספו מסומנים באמצעות עיגול שחור (●) באופן בולט על גבי קווים אלה (בתוספת חץ המצביע על המקום).



בתרשים מתוארת פעימת השעון השישית בביצוע הקוד שבשאלה, על בסיס הנתונים המופיעים בתחילת השאלה. בפעימת שעון זו, פקודת ORI (הפקודה השנייה) נמצאת בשלב WB.

סיכוני הנתונים בהרצת הקוד נפתרים על ידי יחידת העברה קדימה (Forwarding Unit) ויחידת איתור הסיכונים (HDU), וחציית מקבץ האוגרים. שימו לב להשפעת הרכיב הנוסף על יחידות אלה.

9.3 מעבד צנרת (3 נקודות)

בהמשך לסעיף הקודם (9.2 מעבד הצנרת, במחזור שעון 6), רשמו בטופס התשובות את מיקום הפקודות שבקוד, לפי שלבי העבודה של מעבד הצנרת בתחילת מחזור שעון 7. כלומר, מחזור שעון אחד אחרי מחזור השעון המתואר בסעיף 9.2

יש לרשום, בטבלה כאן, לכל שלב בצנרת: **איזו פקודה** תהיה בכל שלב במחזור שעון 7. יש להוסיף הסבר קצר (2-3 משפטים) המסביר מדוע זה מיקום הפקודות בצנרת

שלב 1 - IF	שלב 2 - ID	שלב 3 - EXE	שלב 4 - MEM	שלב 5 - EX
addi \$17, \$16, 4	sw \$16, 0x0(\$5)	bubble	lw \$16, 0xc(\$5)	sll \$10, \$5, 14

כפי שראינו בנקודות 4 ו-5 בסעיף קודם, יחידת HDU במחזור שעון 6 – מזהה שפקודת SW משתמשת באוגר 16 המתעדכן בפקודת LW, ולכן – בוצעה השהייה ונוצר bubble בין ה LW ל- SW

9.4 מעבד צנרת (3 נקודות)

בהמשך לסעיפים 9.2 ו-9.3, עם התייחסות לרכיב ה shift left 1 כפי שמופיע בתרשים שבסעיף 9.2, רשמו מה יהיה ערכו של אוגר 17 עם סיום פקודת ה ADDI שבשורה 6 של הפקודות. יש לרשום את ערכו של אוגר 17 בסיום ביצוע הפקודה, בהקסה, בתוספת הסבר קצר.

$4 = 17_{\$}$: אוגר rs=16, עקב רכיב החומרה, לאחר הזזה מקום אחד שמאלה הוא לכאורה שווה 32, אך בפועל – עקב העובדה שיש רק 5 סיביות, זה יהיה 0 (כדי לבטא 32 צריך 6 סיביות) – ואז אוגר $4 + 0$ זה 4 ... 4

שאלה 10 (20 נקודות)

נתונה פרוצדורה בשם whatDoldo. הפרוצדורה מקבלת 2 פרמטרים:

- \$a0: כתובת של מחרוזת שהוגדרה במקטע הנתונים (.data). כפי שמתואר מיד בהמשך השאלה.

- \$a1: קוד ascii של תו

במקטע הנתונים (.data) נתונה ההגדרה הבאה של המחרוזת **str**:

str: .asciiz "ALBCLLEBLLERL"

בתוכנית הראשית – אופן זימון הפרוצדורה:

```
la $a0, str
li $a1, 'L'
jal whatDoldo
```

קוד הפרוצדורה:

```
1  whatDoldo:
2  # $a0 – null terminated string address, $a1 – ascii code of a character
3      move $t9, $a0
4      move $t8, $a1
5      xor $v0, $v0, $v0
6  loopA:
7      lbu $t0, 0($t9)
8      beq $t0, $zero, endLoopA
9      bne $t0, $t8, cont1
10     addi $v0, $v0, 1
11     move $a0, $t9
12  loopB:
13     lb $t0, 1($a0)
14     sb $t0, 0($a0)
15     addi $a0, $a0, 1
16     bne $t0, $zero, loopB
17     j loopA
18  cont1:
19     add $t9, $t9, 1
20     j loopA
21  endLoopA:
22     jr $ra
```

בהמשך, ראה התייחסות בסעיף 10.4 (*) #

לפני שאתם עונים על השאלות, ודאו שהבנתם היטב את:

- אופן הגדרת המחרוזת במקטע הנתונים (.data)
- אופן זימון הפרוצדורה מהתוכנית הראשית
- מימוש הפרוצדורה

לאחר מכן, יש לענות על השאלות הבאות, **המתייחסות לקוד בפרוצדורה**, במקום המתאים בטופס התשובות בהתאם לסעיפי השאלה:

10.1 (5 נקודות)

עם סיום עבודת הפרוצדורה, מה יהיה הערך באוגר \$v0.
הוסיפו הסבר קצר: מה משמעות ערך זה?

ערכו של \$v0 יהיה 6. ערך זה מציין את כמות התווים שנמחקה מתוך המחרוזת

10.2 (5 נקודות)

עם סיום עבודת הפרוצדורה, מה תהיה המחרוזת כפי שתופיע במקטע הנתונים (.data) בזיכרון עבור המחרוזת **str**.

הוסיפו הסבר קצר: מה ביצעה הפרוצדורה אשר השפיע, או שלא השפיע, על המחרוזת כפי שכתבתם.

עם סיום הפרוצדורה, בזיכרון – במיקום של המחרוזת תהיה מחרוזת קצרה יותר – ללא האות שהוסרה, במקרה של דוגמה כאן, - ללא האות L (שהופיעה במחרוזת המקורית 6 פעמים). לפיכך, בזיכרון תהיה המחרוזת הבאה: **ABCEBER**

10.3 (5 נקודות)

הסבירו בקצרה מה מבצעת הפרוצדורה.

שימו לב: יש להסביר מה עושה הפרוצדורה, ולא להסביר את הפקודות והלולאות ואיך הפרוצדורה עובדת. ההסבר שאתם נדרשים לתאר הינו הסבר של מה מטרת הפרוצדורה בהיבט הפונקציונאלי שלה. הסבר טכני של "כיצד עובדת" הפרוצדורה לא יתקבל!

הפרוצדורה מקבלת מחרוזת מסוג null terminated (ascii): הפרוצדורה עוברת על כל התווים במחרוזת ומחפשת את התו שהתקבל כפרמטר ב \$a1. בכל פעם שמאתרת את התו, היא מסירה אותו מהמחרוזת, תוך כדי "צמצום" כל המחרוזת עד סופה (כולל סימן null terminated). במקביל, הפונקציה סופרת באוגר \$v0 את כמות התווים שהוסרה, ומחזירה כמות זאת לתוכנית המזמנת.

10.4 (5 נקודות)

כעת, מבוצע שינוי בפרוצדורה. **שורה 17** בקוד (מודגשת, בתוספת הערה לשימוש בסעיף זה) – נמחקת מהקוד. עיינו בקוד שוב, בהנחה ששורה זו לא מתבצעת (כלומר, נמחקה), וענו על השאלות הבאות:

א. האם הפרוצדורה תעבוד באופן תקין? הסבירו ונמקו את התשובה שלכם (שימו לב, אין צורך להתייחס בסעיף זה לתוצאות – אלא להתמקד בהתייחסות לתקינות עבודת הפרוצדורה)

כן, הפרוצדורה תעבוד תקין, כלומר תרוץ ותסיים תקין. מאידך, יהיה שוני בתוצאות הפרוצדורה

ב. בעקבות השינוי: עם סיום עבודת הפרוצדורה, מה תהיה המחרוזת כפי שתופיע במקטע הנתונים (.data). בזיכרון עבור המחרוזת **str**. מה יהיה הערך המוחזר באוגר \$v0?

המחרוזת בזיכרון תהיה כעת: ABCLEBLER

בכל פעם שיש רצף של 2 אותיות זהות, מאות שיש להסיר – רק אחת מהן מוסרת, ולא שתיהן כמו בקוד המקורי, ולכן, כעת הערך באוגר \$v0 יהיה 4

ג. הסבירו מדוע השינוי בקוד הפרוצדורה השפיע באופן זה

כתוצאה מהשינוי, כל פעם שיש רצף של 2 אותיות להסרה, השנייה מבניהן לא מוסרת. ההסבר לכך: הקוד מכווץ את המחרוזת למקום שנמחק. כאשר יש 2 אותיות ברצף, השנייה מבניהן מועתקת במקום הראשונה, וכך הלאה. כתוצאה מהשינוי, במקום לחזור ולבדוק שוב את אותה אות עוברים (דרך תווית cont) למיקום הבא, והמשמעות הלוגית הינה שבפועל מדלגים על האות השנייה ברצף של 2