

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
0xa0400000	0x2010a040	addi \$16,\$0,0x0000a040	13 addi \$s0, \$zero, 0xa040
0xa0400004	0x00108400	sll \$16,\$16,0x00000010	14 sll \$s0, \$s0, 16
0xa0400008	0x22100008	addi \$16,\$16,0x00000008	15 addi \$s0, \$s0, 8
0xa040000c	0x02000008	jr \$16	16 jr \$s0
0xa0400010	0x22100004	addi \$16,\$16,0x00000004	17 addi \$s0, \$s0, 0x4
0xa0400014	0x02000008	jr \$16	18 jr \$s0
0xa0400018	0x2402000a	addiu \$2,\$0,0x0000000a	19 li \$v0, 10 # Exit
0xa040001c	0x0000000c	syscall	20 syscall

שאלה 1

כאשר הקוד יבצע בפעם הראשונה את הפקודה JR **בשורה 16**: באיזו כתובת נמצאת הפקודה הבאה שתבוצע (כלומר, הפקודה שתבוצע אחרי פקודת JR)?

- לא ניתן לדעת, לא נתונה תווית בקוד ולא ברור לאן יבוצע הדילוג
- ניתן לדעת, הכתובת אליה יבוצע הדילוג היא 0xa0400008 (שורה 15 בקוד)
- ניתן לדעת, הכתובת אליה יבוצע הדילוג היא 0xa0400000 (שורה 13 בקוד)
- ניתן לדעת, הכתובת אליה יבוצע הדילוג היא 0xa0400018 (שורה 19 בקוד)
- ניתן לדעת, הכתובת אליה יבוצע הדילוג היא 0xa0400010 (שורה 17 בקוד)

שאלה 2

בעת הרצת הקוד, איזה מהמשפטים הבאים מסביר מה יקרה?

- יש בעיה בקוד - לא ניתן להבין מה קורה, במיוחד עקב פקודות JR בשורות 16 ו- 18, שלא ברור מה יעשו.
- בפעם ראשונה שהקוד יגיע לפקודת JR, בשורה 16, יבוצע דילוג לכתובת 0xa0400008 (שורה 15), ולמעשה תהיה לולאה אין סופית בין פקודת JR בשורה 16 לבין הפקודה בשורה 15.
- בפעם ראשונה שהקוד יגיע לפקודת JR, בשורה 16, יבוצע דילוג לכתובת 0xa0400008 (שורה 15), ובפעם השנייה שיגיע לפקודת JR (בשורה 16) יבצע דילוג לכתובת 0xa0400010 (שורה 17), וכאשר יגיע לפקודת JR בשורה 18 יכנס ללולאה אין סופית כאשר ידלג שוב ושוב לכתובת 0xa0400014 (שורה 18)
- בפעם ראשונה שהקוד יגיע לפקודת JR, בשורה 16, יבוצע דילוג לכתובת 0xa0400008 (שורה 15), ובפעם השנייה שיגיע לפקודת JR (בשורה 16) יבצע דילוג לפקודה בכתובת 0xa0400010 (שורה 17), וכאשר יגיע לפקודת JR בשורה 18 ידלג לכתובת 0xa0400018 (שורה 19) ויגיע לסיום הקוד בשורה 20.
- בפעם ראשונה שהקוד יגיע לפקודת JR, בשורה 16, יבוצע דילוג לכתובת 0xa0400010 (שורה 17), וכאשר יגיע לפקודת JR בשורה 18 ידלג לכתובת 0xa0400018 (שורה 19) ויגיע לסיום הקוד בשורה 20.

שאלה 3

תוכנית רצה, על מעבד שתדירותו 3GHz, במשך 12 שניות. לצורך שיפור ביצועים, הוחלט להחליף את כל פקודות הכפל ב- 3 על ידי 2 פקודות חיבור, כלומר:

$\text{mul } x, x, 3 \rightarrow \begin{array}{l} \text{add } \$at, x, x \\ \text{add } x, \$at, x \end{array}$

נתון כי ה- CPI של פקודת כפל הוא 6, ושל פקודת חיבור הוא 1. לאחר קומפילציה מחדש התוכנית רצה ב- 10.5 שניות. כמה פקודות כפל הוחלפו?

א. לא ניתן לחשב מאחר ולא ידוע לנו מהו ה IC הכולל של התוכנית

ב. $1.125 * 10^8$

ג. $1.125 * 10^9$

ד. $0.888 * 10^8$

ה. $0.888 * 10^9$

שאלה 4

נתון זיכרון מטמון שהיקף/גודל הנתונים (data) הינו 64 Kbytes. המטמון ממפה מרחב כתובות של 32 סיביות בזיכרון הראשי

נתון כי המיפוי הינו 4 ways set associative

נתון כי במטמון יש $1024 (2^{10})$ סיביות Valid

על בסיס נתונים אלה – מה גודלו של שדה ה- Tag?

א. 14 סיביות

ב. 16 סיביות

ג. 18 סיביות

ד. 20 סיביות

ה. לא ניתן לחשב מאחר ואין נתון של גודל הבלוק במטמון

שאלה 5

הסיבה לכך ששיטת המשלים לשתיים לייצוג מספרים עם סימן היא השימושית ביותר במחשבים היא:

א. בשיטה זו אין הבדל בין הלוגיקה של חיבור מספרים חיוביים לזו של חיבור שליליים

ב. התנאי לגלישה מהתחום (overflow) זהה לזו שבשיטת ייצוג מספרים ללא סימן ולכן אותו חיווי גלישה (overflow) מגדיר גלישה בשתי שיטות הייצוג ללא סימן ומשלים לשתיים

ג. זוהי הדרך היחידה לייצג מספרים שליליים וחיוביים

ד. בשיטה זו ההבדל היחיד בין שני מספרים בעלי אותו ערך מוחלט מתבטא בתוכן סיבית יחידה

ה. אין הסבר מיוחד לכך, פשוט החליטו כך

שאלה 6

נתון המעגל הלוגי הבא (משמאל):

מה מחזירה הפונקציה F ?

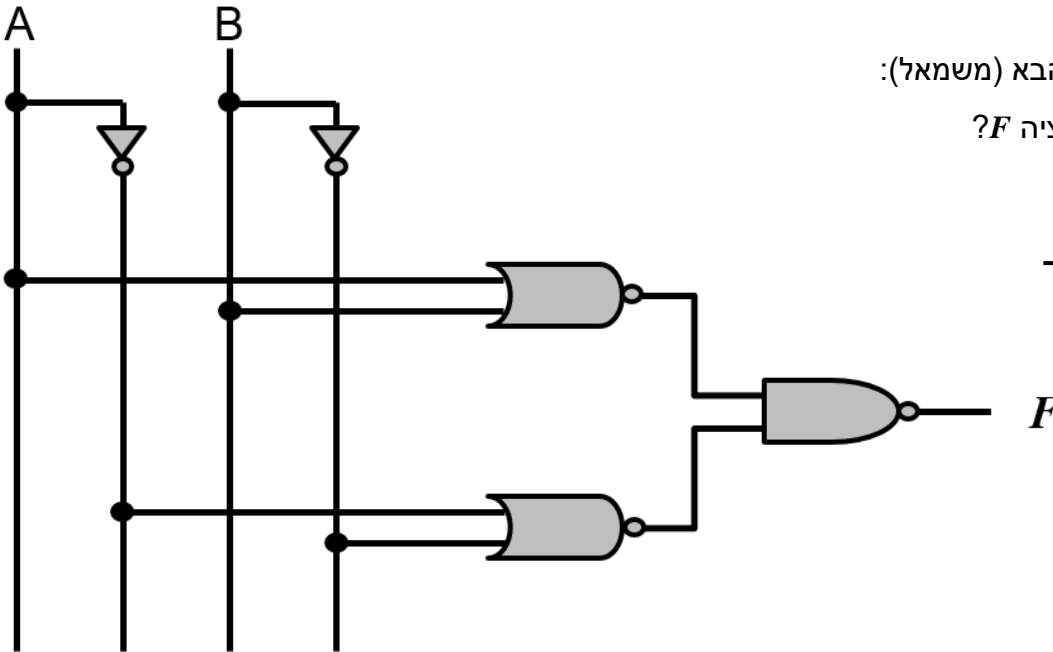
א. $A \text{ XOR } B$

ב. $\overline{A \text{ XOR } B}$

ג. $A \text{ OR } B$

ד. 0

ה. 1



שאלה 7

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

א. יחידה זו מקבלת את נתון דגל PC Src שמחושב בפועל בשלב MEM של ה-Branch, ובאמצעות

Predicted Direction של יחידת ה-BTB (שעבר לאורך הצנרת) מזוהה האם החיזוי (שבוצע

בשלב ה-IF של פקודת ה-Branch) תקין או שגוי. במידה וזוהה כשגוי מבוצע טיפול בניקוי

הפקודות שיש לנקות מהצנרת, עדכון PC בכתובת הנכונה, ותיקון התחזית ב-BTB

ב. יחידה זו מקבלת את נתון דגל PC Src שחושב בפועל בשלב MEM, ובאמצעות Predicted

Direction של יחידת ה-BTB (שעבר לאורך הצנרת) מזוהה האם החישוב תקין או שגוי. במידה

וזוהה כשגוי מבוצע תיקון של דגל PC Src שחושב, על בסיס התחזית. כמו כן, מבוצע טיפול

בניקוי הפקודות שיש לנקות מהצנרת.

ג. יחידה זו אחראית לביצוע flush של פקודות מהצנרת. על בסיס נתוני החיזוי המועברים מה-BTB

דרך קווי הצנרת – אם החיזוי הוא branch taken, מבוצע ניקוי של הפקודות השגויות מתוך

הצנרת.

ד. יחידה זו מקבלת את מספר אוגר המטרה של הפקודה ואת דגל Mem Read. בנוסף, מקבלת

את מספרי האוגרים שבשימוש בפקודה שלאחר פקודה נוכחית. היה ויש שימוש באוגר המטרה

בפקודה שלאחר הפקודה הנוכחית, מזוהה צורך ליצור bubble – ובאמצעות מגנון של דגלי ה

flush היוצאים מיחידה זו – נוצרים ה bubble במקומות הנדרשים.

ה. כל התשובות (א' עד ד') נכונות

שאלה 8

ב-MIPS העובד בטכנולוגיית צנרת: קו הבקרה IF Flush (ניתן לראות בתרשימים 4.65 ו-4.66 בנספח החומרה) וקו הבקרה IF/ID.Write (ניתן לראות בתרשימים 4.60 בנספח החומרה) – פועלים שניהם על אוגר הצנרת IF/ID. מה ההבדל בין קווים אלה?

א. אין הבדל בין ביניהם. זה פשוט כינוי שונה לאותה בקרה בתרשימים שונים בצנרת, המבצעים

פעולה של מחיקת אוגר הצנרת IF/ID

ב. קו IF.Flush נוצר על ידי ה-Control ומטרתו למחוק כל פקודה לא חוקית שנכנסה לאוגר הצנרת

IF/ID, ואילו קו IF/ID Write נוצר על ידי HDU כאשר מזוהה שימוש באוגר מטרה של LW

בפקודה שלאחריו

ג. קו IF Flush נוצר על ידי ה-Control כאשר מזוהה שוויון בבדיקת BEQ בשלב ID, ומיועד לנקות

את אוגר הצנרת מהפקודה שלאחר BEQ באמצעות מחיקת הפקודה והפיכת הקידוד שלה ל-

0, ואילו קו IF/ID Write נוצר על ידי HDU כאשר מזוהה שימוש באוגר מטרה של LW בפקודה

שלאחריו והופך את הפקודה שב IF/ID לבועה (Bubble)

ד. קו IF Flush נוצר על ידי ה-Control כאשר מזוהה שוויון בבדיקת BEQ בשלב ID, ומיועד לנקות

את אוגר הצנרת מהפקודה שלאחר BEQ באמצעות מחיקת הפקודה והפיכת הקידוד שלה ל-

0, ואילו קו IF/ID Write נוצר על ידי HDU כאשר מזוהה שימוש באוגר מטרה של LW בפקודה

שלאחריו ומיועד לעכב את הפקודה שב IF/ID מחזור שרון, באמצעות מניעת כתיבה על IF/ID.

ה. קו IF/ID Write נוצר על ידי ה-Control כאשר מזוהה שוויון בבדיקת BEQ בשלב ID, ומיועד

לנקות את אוגר הצנרת מהפקודה שלאחר BEQ באמצעות מחיקת הפקודה והפיכת הקידוד

שלה ל-0, ואילו קו IF Flush נוצר על ידי HDU כאשר מזוהה שימוש באוגר מטרה של LW

בפקודה שלאחריו ומיועד לעכב את הפקודה שב IF/ID מחזור שרון, באמצעות מניעת כתיבה

על IF/ID.

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
0x00400400	0x20104004	addi \$16,\$0,0x00004004	13 addi \$16, \$zero, 0x4004
0x00400404	0x8e08000c	lw \$8,0x0000000c(\$16)	14 lw \$8, 0xc(\$16)
0x00400408	0x00108380	sll \$16,\$16,0x0000000e	15 sll \$16, \$16, 14
0x0040040c	0xae080000	sw \$8,0x00000000(\$16)	16 sw \$8, 0(\$16)
0x00400410	0x1604ffff	bne \$16,\$4,0xffffffff	17 bne \$16, \$4, cont
0x00400414	0x01454822	sub \$9,\$10,\$5	18 sub \$9, \$10, \$5

בתחילת ביצוע הקוד: נתון שערכו של כל אוגר שווה למספרו הוא. כלומר, אוגר \$1 שווה 1, אוגר \$2 שווה 2,, אוגר \$10 שווה 10 ... וכך הלאה עד אוגר \$31.

נתון כי באזור (סגמנט) הנתונים (.data), אשר מתחיל בכתובת 0x1001000 מוגדרים נתונים באופן הבא:

Values: .word 0x11111111, 0x22222222, 0x33333333, 0x44444444, 0x55555555

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, עם תוספת טיפול ב-bne). במעבד שבתרשים בוצע שינוי לפני המחבר של PC. במקום הערך 4, **נכנס הערך 8** (מסומן ב-☆). ערכו של PC מקודם בהתאמה לשינוי זה.

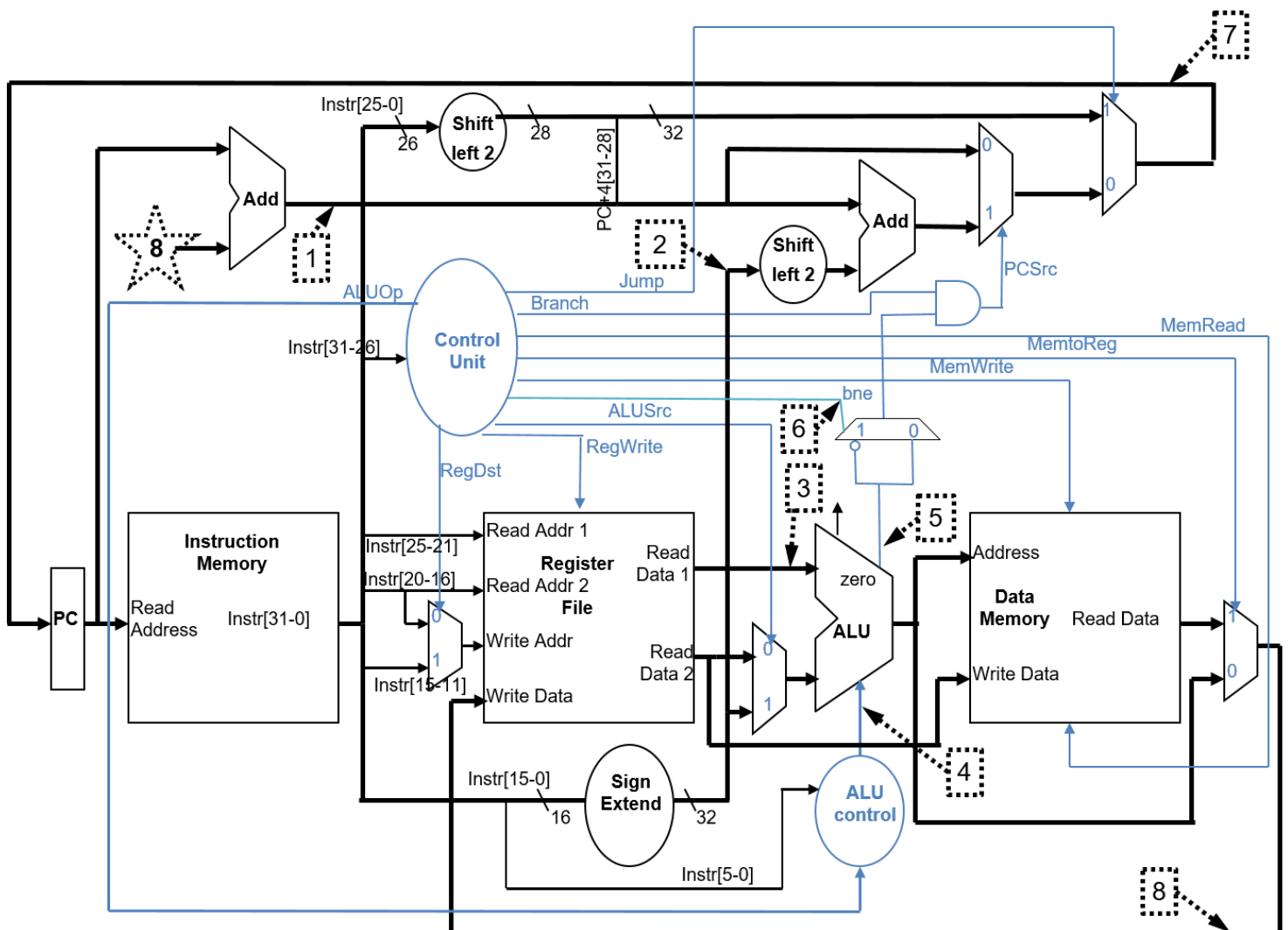
הניחו שהפקודה שמתבצעת בפעימת השעון הראשונה הינה פקודת ADDI שבשורה 13.

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 BNE (שורה 17 בקוד שבעמוד קודם).

bne \$16, \$4, cont

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 (3 נקודות)

בעמוד הבא נתון תרשים 4.66 הכולל טיפול בפקודות branch בשלב ה ID. הניחו שטיפול זה כולל גם את פקודות BNE ולא רק את BEQ. כמו כן, התרשים **כולל את שינוי החומרה** המתואר במעבד החד-מחזורי בסעיף 9.1 (גם כאן, מודגש השינוי בעזרת "כוכב").

רשמו בטופס התשובות את מיקום הפקודות שבקוד, לפי שלבי העבודה של מעבד הצנרת בתחילת מחזור שעון 7. יש לרשום, בטבלה כאן, לכל שלב בצנרת: **איזו פקודה** תהיה בכל שלב במחזור שעון 7. יש להוסיף הסבר קצר המסביר מדוע זה מיקום הפקודות בצנרת.

שלב 1 - IF	שלב 2 - ID	שלב 3 - EXE	שלב 4 - MEM	שלב 5 - EX
				bne \$16, \$4, cont

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

בתרשים, המבוסס על תרשים 4.66, מסומנות 6 נקודות (1-6). עליכם לסמן **בטבלה**, שבעמוד הבא, את הערכים העוברים על הקווים המסומנים **בבסיס הקסה** (3 נקודות לסימון), בהנחה שהמידע נבדק לקראת סוף פעימות השעון השביעית (7).

שימו לב: העתיקו את סדר הפקודות שרשמתם לחלק העליון של התרשים, כפי שרשמתם בסעיף 9.2, כדי להקל עליכם את המעקב. בהתאם לנקודות המסומנות ובהתאם לפקודות בכל שלב ושלב, השלימו את הערכים בטבלה.

במקרה של טעות בסעיף 9.2, ובהנחה שיש תאימות מלאה בין תשובתכם בסעיף 9.2 לסעיף 9.3, הפתרון לא יפסל. כל ערך התואם את הטעות יקבל 2 מתוך 3 הנקודות של הסעיף (טעות נגררת).

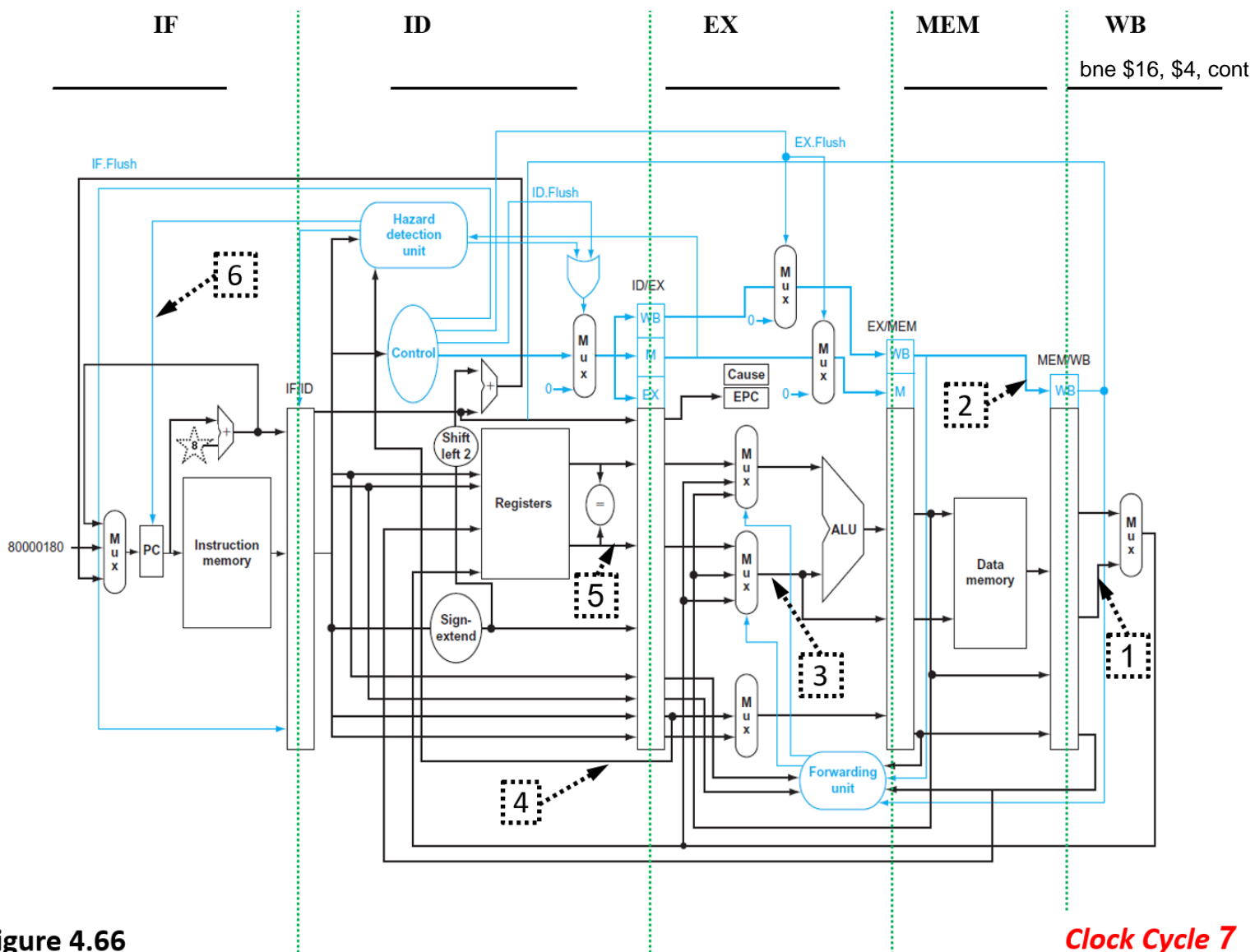


Figure 4.66

Clock Cycle 7

1	
2	
3	

4	
5	
6	

9.4 מעבד צנרת (3 נקודות)

בהמשך לסעיפים 9.2 ו- 9.3, רשמו בטבלה כאן, בדומה לסעיף 9.2, עבור מחזור שעון 8 – איזו פקודה תהיה בכל שלב במחזור שעון זה. יש להוסיף הסבר קצר (2-3 משפטים) המסביר מודע זה מיקום הפקודות בצנרת.

שלב 1 - IF	שלב 2 - ID	שלב 3 - EXE	שלב 4 - MEM	שלב 5 - EX

שאלה 10 (20 נקודות)

נתונה פרוצדורה בשם whatDoldo. הפרוצדורה מקבלת פרמטר אחד:

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

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

str: .asciiz "a4D]h{Q>w^Z"

בתוכנית הראשית – אופן זימון הפרוצדורה:

```
la    $a0, str1
jal   whatDoldo
```

קוד הפרוצדורה:

```
1      whatDoldo:
2      #    #a0 - null terminated string address
3          xor    $v0, $v0, $v0
4      loopStart:
5          lbu     $t0, 0($a0)
6          beq     $t0, $zero, loopEnd
7          blt     $t0, 'A', cont1
8          ble     $t0, 'Z', cont2
9          blt     $t0, 'a', cont1
10         bgt     $t0, 'z', cont1
11         andi    $t0, $t0, 0xdf
12         j      cont3
13     cont1:
14         addi    $v0, $v0, 1
15         j      cont4
16     cont2:
17         ori     $t0, $t0, 0x20
18     cont3:
19         sb      $t0, 0($a0)
20     cont4:
21         addi    $a0, $a0, 1
22         j      loopStart
23     loopEnd:
24         jr      $ra
```

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, בוצע שינוי של הגדרת המחרוזת. במקטע הנתונים מופיע ההגדרה הבאה של המחרוזת `str`:
`str: .ascii "aBcD$&)?QRs"`

לא בוצע כל שינוי בזימון הפרוצדורה וכן לא בוצע כל שינוי בפרוצדורה עצמה. לאור זאת:
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. מה יהיה בזימון, במקטע הנתונים (`.data`) ערך המחרוזת שהוגדרה (בהתייחסות, כמובן, לתשובתכם בתת סעיף א')

ג. מה יוחזר באוגר `$v0` (בהתייחסות, כמובן, לתשובתכם בתת סעיף א')

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
			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
0xa0400000	0x2010a040	addi \$16,\$0,0x0000a040	13 addi \$s0, \$zero, 0xa040
0xa0400004	0x00108400	sll \$16,\$16,0x00000010	14 sll \$s0, \$s0, 16
0xa0400008	0x22100008	addi \$16,\$16,0x00000008	15 addi \$s0, \$s0, 8
0xa040000c	0x02000008	jr \$16	16 jr \$s0
0xa0400010	0x22100004	addi \$16,\$16,0x00000004	17 addi \$s0, \$s0, 0x4
0xa0400014	0x02000008	jr \$16	18 jr \$s0
0xa0400018	0x2402000a	addiu \$2,\$0,0x0000000a	19 li \$v0, 10 # Exit
0xa040001c	0x0000000c	syscall	20 syscall

שאלה 1

כאשר הקוד יבצע בפעם הראשונה את הפקודה JR **בשורה 16**: באיזו כתובת נמצאת הפקודה הבאה שתבוצע (כלומר, הפקודה שתבוצע אחרי פקודת JR)?

א. לא ניתן לדעת, לא נתונה תווית בקוד ולא ברור לאן יבוצע הדילוג

ב. ניתן לדעת, הכתובת אליה יבוצע הדילוג היא 0xa0400008 (שורה 15 בקוד)

ג. ניתן לדעת, הכתובת אליה יבוצע הדילוג היא 0xa0400000 (שורה 13 בקוד)

ד. ניתן לדעת, הכתובת אליה יבוצע הדילוג היא 0xa0400018 (שורה 19 בקוד)

ה. ניתן לדעת, הכתובת אליה יבוצע הדילוג היא 0xa0400010 (שורה 17 בקוד)

שאלה 2

בעת הרצת הקוד, איזה מהמשפטים הבאים מסביר מה יקרה?

א. יש בעיה בקוד - לא ניתן להבין מה קורה, במיוחד עקב פקודות JR בשורות 16 ו- 18, שלא ברור מה יעשו

ב. בפעם ראשונה שהקוד יגיע לפקודת JR, בשורה 16, יבוצע דילוג לכתובת 0xa0400008 (שורה 15), ולמעשה תהיה לולאה אין סופית בין פקודת JR בשורה 16 לבין הפקודה בשורה 15.

ג. בפעם ראשונה שהקוד יגיע לפקודת JR, בשורה 16, יבוצע דילוג לכתובת 0xa0400008 (שורה 15), ובפעם השנייה שיגיע לפקודת JR (בשורה 16) יבצע דילוג לכתובת 0xa0400010 (שורה 17), וכאשר יגיע לפקודת JR בשורה 18 יכנס ללולאה אין סופית כאשר ידלג שוב ושוב לכתובת 0xa0400014 (שורה 18)

ד. בפעם ראשונה שהקוד יגיע לפקודת JR, בשורה 16, יבוצע דילוג לכתובת 0xa0400008 (שורה 15), ובפעם השנייה שיגיע לפקודת JR (בשורה 16) יבצע דילוג לפקודה בכתובת 0xa0400010 (שורה 17), וכאשר יגיע לפקודת JR בשורה 18 ידלג לכתובת 0xa0400018 (שורה 19) ויגיע לסיום הקוד בשורה 20.

ה. בפעם ראשונה שהקוד יגיע לפקודת JR, בשורה 16, יבוצע דילוג לכתובת 0xa0400010 (שורה 17), וכאשר יגיע לפקודת JR בשורה 18 ידלג לכתובת 0xa0400018 (שורה 19) ויגיע לסיום הקוד בשורה 20.

שאלה 3

תוכנית רצה, על מעבד שתדירותו 3GHz, במשך 12 שניות. לצורך שיפור ביצועים, הוחלט להחליף את כל פקודות הכפל ב- 3 על ידי 2 פקודות חיבור, כלומר:

$\text{mul } x, x, 3 \rightarrow \begin{array}{l} \text{add } \$at, x, x \\ \text{add } x, \$at, x \end{array}$

נתון כי ה- CPI של פקודת כפל הוא 6, ושל פקודת חיבור הוא 1. לאחר קומפילציה מחדש התוכנית רצה ב- 10.5 שניות. כמה פקודות כפל הוחלפו?

א. לא ניתן לחשב מאחר ולא ידוע לנו מהו ה IC הכולל של התוכנית

ב. $1.125 * 10^8$

ג. $1.125 * 10^9$

ד. $0.888 * 10^8$

ה. $0.888 * 10^9$

נגדיר X – מספר פקודות כפל
נתון CPI של פקודת כפל – 6, CPI של חיבור – 1
אפשר לומר, על בסיס נתוני השאלה כי זמן ריצה של פקודות הכפל פחות זמן ריצה של פקודות החיבור שהחליפו את פקודת הכפל שווה ל- 1.5 שניות, שהרי זה הזמן שנחסך

$$6 * X / (3 * 10^9) - 2 * X * 1 / (3 * 10^9) = 1.5$$

$$6x - 4x = 1.5 * 3 * 10^9 \rightarrow 4x = 4.5 * 10^9 \rightarrow X = 1.125 * 10^9$$

שאלה 4

נתון זיכרון מטמון שהיקף/גודל הנתונים (data) הינו 64 Kbytes. המטמון ממפה מרחב כתובות של 32 סיביות בזיכרון הראשי

נתון כי המיפוי הינו 4 ways set associative

נתון כי במטמון יש 1024 (2^{10}) סיביות Valid

על בסיס נתונים אלה – מה גודלו של שדה ה- Tag?

א. 14 סיביות

ב. 16 סיביות

ג. 18 סיביות

ד. 20 סיביות

גודל המטמון הנתון הוא 64K, כלומר $2^{16} \text{ bytes} = 2^{10} * 2^6$. במטמון יש 2^{10} סיביות Valid, מכאן ניתן להבין שיש כמות זהה של בלוקים בכל המטמון, שהרי יש סיביות Valid אחת הצמודה לכל בלוק (ללא תלות בכמות ה- ways), ומכאן שגודל בלוק הוא 64 בתים: $2^{16} / 2^{10} = 2^6$, כלומר, יש 2 סיביות byte offset, ו- 4 סיביות Block offset
כמו כן, מאחר ונתון 4 ways set associative, ניתן לדעת שכמות השורות בכל way הינה $2^{10} / 2^2 = 2^8 = 256$.
וכעת ניתן למצוא את גודל ה- Tag: $32 - 8 - 4 - 2 = 18$

ה. לא ניתן לחשב מאחר ואין נתון של גודל הבלוק במטמון

שאלה 5

הסיבה לכך ששיטת המשלים לשתיים לייצוג מספרים עם סימן היא השימושית ביותר במחשבים היא:

א. בשיטה זו אין הבדל בין הלוגיקה של חיבור מספרים חיוביים לזו של חיבור שליליים

ב. התנאי לגלישה מהתחום (overflow) זהה לזו שבשיטת ייצוג מספרים ללא סימן ולכן אותו חייווי גלישה (overflow) מגדיר גלישה בשתי שיטות הייצוג ללא סימן ומשלים לשתיים

ג. זוהי הדרך היחידה לייצג מספרים שליליים וחיוביים

ד. בשיטה זו ההבדל היחיד בין שני מספרים בעלי אותו ערך מוחלט מתבטא בתוכן סיבית יחידה

ה. אין הסבר מיוחד לכך, פשוט החליטו כך

שאלה 6

נתון המעגל הלוגי הבא (משמאל):

מה מחזירה הפונקציה F ?

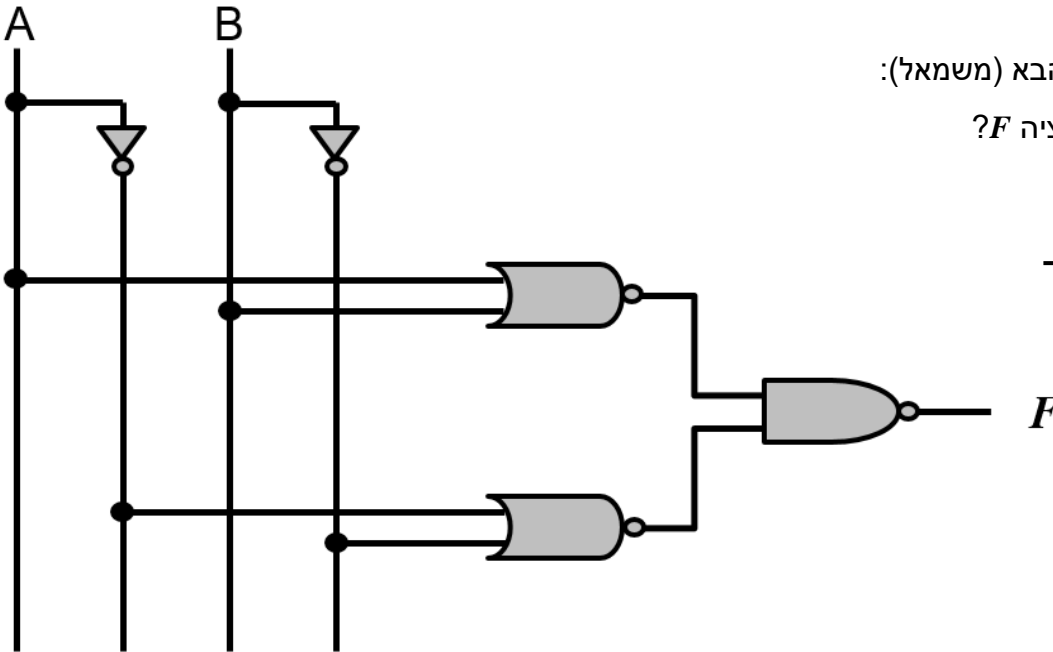
א. $A \text{ XOR } B$

ב. $\overline{A \text{ XOR } B}$

ג. $A \text{ OR } B$

ד. 0

ה. 1



שאלה 7

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

א. יחידה זו מקבלת את נתון דגל PCSrc שמחושב בפועל בשלב MEM של ה-Branch, ובאמצעות

Predicted Direction של יחידת ה-BTB (שעבר לאורך הצנרת) מזוהה האם החיזוי (שבוצע

בשלב ה-IF של פקודת ה-Branch) תקין או שגוי. במידה וזוהה כשגוי מבוצע טיפול בניקוי

הפקודות שיש לנקות מהצנרת, עדכון PC בכתובת הנכונה, ותיקון התחזית ב-BTB

ב. יחידה זו מקבלת את נתון דגל PCSrc שחושב בפועל בשלב MEM, ובאמצעות Predicted

Direction של יחידת ה-BTB (שעבר לאורך הצנרת) מזוהה האם החישוב תקין או שגוי. במידה

וזוהה כשגוי מבוצע תיקון של דגל PCSrc שחושב, על בסיס התחזית. כמו כן, מבוצע טיפול

בניקוי הפקודות שיש לנקות מהצנרת.

ג. יחידה זו אחראית לביצוע flush של פקודות מהצנרת. על בסיס נתוני החיזוי המועברים מה-BTB

דרך קווי הצנרת – אם החיזוי הוא branch taken, מבוצע ניקוי של הפקודות השגויות מתוך

הצנרת.

ד. יחידה זו מקבלת את מספר אוגר המטרה של הפקודה ואת דגל Mem Read. בנוסף, מקבלת

את מספרי האוגרים שבשימוש בפקודה שלאחר פקודה נוכחית. היה ויש שימוש באוגר המטרה

בפקודה שלאחר הפקודה הנוכחית, מזוהה צורך ליצור bubble – ובאמצעות מגנון של דגלי ה

flush היוצאים מיחידה זו – נוצרים ה bubble במקומות הנדרשים.

ה. כל התשובות (א' עד ד') נכונות

שאלה 8

ב-MIPS העובד בטכנולוגיית צנרת: קו הבקרה IF Flush (ניתן לראות בתרשימים 4.65 ו-4.66 בנספח החומרה) וקו הבקרה IF/ID.Write (ניתן לראות בתרשימים 4.60 בנספח החומרה) – פועלים שניהם על אוגר הצנרת IF/ID. מה ההבדל בין קווים אלה?

א. אין הבדל בין ביניהם. זה פשוט כינוי שונה לאותה בקרה בתרשימים שונים בצנרת, המבצעים פעולה של מחיקת אוגר הצנרת IF/ID

ב. קו IF.Flush נוצר על ידי ה-Control ומטרתו למחוק כל פקודה לא חוקית שנכנסה לאוגר הצנרת IF/ID, ואילו קו IF/ID Write נוצר על ידי HDU כאשר מזוהה שימוש באוגר מטרה של LW בפקודה שלאחריו

ג. קו IF Flush נוצר על ידי ה-Control כאשר מזוהה שוויון בבדיקת BEQ בשלב ID, ומיועד לנקות את אוגר הצנרת מהפקודה שלאחר BEQ באמצעות מחיקת הפקודה והפיכת הקידוד שלה ל-0, ואילו קו IF/ID Write נוצר על ידי HDU כאשר מזוהה שימוש באוגר מטרה של LW בפקודה שלאחריו והופך את הפקודה שב IF/ID לבועה (Bubble)

ד. קו IF Flush נוצר על ידי ה-Control כאשר מזוהה שוויון בבדיקת BEQ בשלב ID, ומיועד לנקות את אוגר הצנרת מהפקודה שלאחר BEQ באמצעות מחיקת הפקודה והפיכת הקידוד שלה ל-0, ואילו קו IF/ID Write נוצר על ידי HDU כאשר מזוהה שימוש באוגר מטרה של LW בפקודה שלאחריו ומיועד לעכב את הפקודה שב IF/ID מחזור שני, באמצעות מניעת כתיבה על IF/ID.

ה. קו IF/ID Write נוצר על ידי ה-Control כאשר מזוהה שוויון בבדיקת BEQ בשלב ID, ומיועד לנקות את אוגר הצנרת מהפקודה שלאחר BEQ באמצעות מחיקת הפקודה והפיכת הקידוד שלה ל-0, ואילו קו IF Flush נוצר על ידי HDU כאשר מזוהה שימוש באוגר מטרה של LW בפקודה שלאחריו ומיועד לעכב את הפקודה שב IF/ID מחזור שני, באמצעות מניעת כתיבה על IF/ID.

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
0x00400400	0x20104004	addi \$16,\$0,0x00004004	13 addi \$16, \$zero, 0x4004
0x00400404	0x8e08000c	lw \$8,0x0000000c(\$16)	14 lw \$8, 0xc(\$16)
0x00400408	0x00108380	sll \$16,\$16,0x0000000e	15 sll \$16, \$16, 14
0x0040040c	0xae080000	sw \$8,0x00000000(\$16)	16 sw \$8, 0(\$16)
0x00400410	0x1604ffff	bne \$16,\$4,0xfffffff	17 bne \$16, \$4, cont
0x00400414	0x01454822	sub \$9,\$10,\$5	18 sub \$9, \$10, \$5

בתחילת ביצוע הקוד: נתון שערכו של כל אוגר שווה למספרו הוא. כלומר, אוגר \$1 שווה 1, אוגר \$2 שווה 2, ..., אוגר \$10 שווה 10 ... וכך הלאה עד אוגר \$31.

נתון כי באזור (סגמנט) הנתונים (.data), אשר מתחיל בכתובת 0x1001000 מוגדרים נתונים באופן הבא:

Values: .word 0x11111111, 0x22222222, 0x33333333, 0x44444444, 0x55555555

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. שורה 13: $16 = 0x00004004$

2. שורה 15: $16 = 0x10010000$ (הקוד מבצע כל פקודה "שנייה" עקב דילוג של 8 בתים ב PC)

3. שורה 17: אוגר 4 שווה 4 בהתאם להגדרות השאלה, ולכן לא מתקיים שוויון. לאור זאת, מתבצע הדילוג לפי החישוב הבא:

$$Pc+8 + imm*4 = 0x00400410 + 0x8 + 0xFFFFFFF*4 = 0x00400418 + 0xFFFFFEC = 0x00400404$$

כתובת זו הינה הכתובת של הפקודה בשורה 14, פקודת ה LW

4. שורה 14: פקודת ה LW מבצעת Load מכתובת לפי ערכו של אוגר 16 (שחושב בשורה 15) בתופסת 0xc, כתובת

0x1001000c. זו כתובת באזור הנתונים, של המערך Values, ומבוצע load לערך הרביעי במערך: 0x44444444.

5. שורה 16: מבוצע sw לכתובת 0x10010000, כך שהערך שבאוגר \$8, שכפי שקיבל בשורה 14, ישמר במקום הראשון של המערך המוגדר בזיכרון.

6. שורה 18: אוגר 5 = 10 - 5 = \$9

9.1 מעבד חד מחזורי (24 נקודות)

בתרשים שלהלן מעבד חד-מחזורי (המבוסס על תרשים 4.24, עם תוספת טיפול ב-bne). במעבד שבתרשים בוצע שינוי לפני המחבר של PC. במקום הערך 4, נכנס הערך 8 (מסומן ב-☆). ערכו של PC מקודם בהתאמה לשינוי זה.

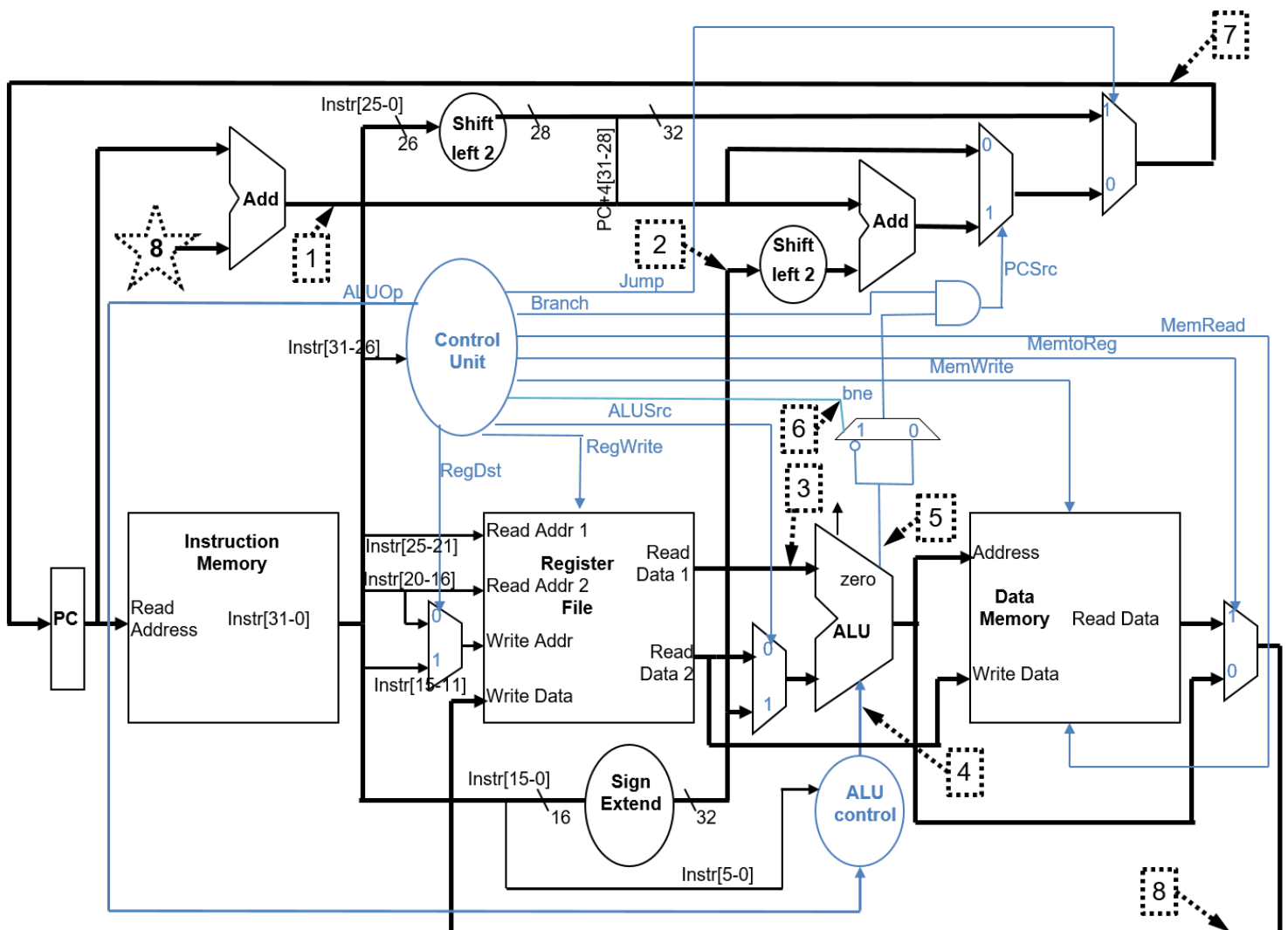
הניחו שהפקודה שמתבצעת בפעימת השעון הראשונה הינה פקודת ADDI שבשורה 13.

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 BNE (שורה 17 בקוד שבעמוד קודם).

bne \$16, \$4, cont

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	0x00400418 (כתובת פקודת bne + 8)	5	0x0 – דגל zero, מאחר ואין שוויון, הדגל הוא 0
2	0xFFFFFFFF (סיביות 0-15 לאחר sign Extend)	6	0x1 – דגל bne המציין שזו פקודת bne ויש לבחור את היציאה של not zero ב-mux כדי שיבוצע דילוג כאשר "לא שווה"
3	0x10010000 – ערכו של אוגר \$16 לאחר ביצוע הפקודה בשורה 15	7	0x00400404 – זו תוצאת החישוב של חיבור נקודה (1) עם תוצאת הכפלת נקודה (2) ב-4
4	0110 = 0x6 – סיביות ה-ALU Control – הנחיה ל-ALU לבצע חיסור להשוואה	8	X – לא מוגדר ולא ניתן לדעת, זו פקודת bne ואין הגדרה של memToReg ל-branch בטבלה 4.18

9.2 מעבד צנרת (3 נקודות)

בעמוד הבא נתון תרשים 4.66 הכולל טיפול בפקודות branch בשלב ה-ID. הניחו שטיפול זה כולל גם את פקודות BNE ולא רק את BEQ. כמו כן, התרשים כולל את שינוי החומרה המתואר במעבד החד-מחזורי בסעיף 9.1 (גם כאן, מודגש השינוי בעזרת "כוכב").

רשמו בטופס התשובות את מיקום הפקודות שבקוד, לפי שלבי העבודה של מעבד הצנרת בתחילת מחזור שעון 7. יש לרשום, בטבלה כאן, לכל שלב בצנרת: איזו פקודה תהיה בכל שלב במחזור שעון 7. יש להוסיף הסבר קצר המסביר מדוע זה מיקום הפקודות בצנרת.

שלב 1 - IF	שלב 2 - ID	שלב 3 - EXE	שלב 4 - MEM	שלב 5 - EX
sub \$9, \$10, \$5	sw \$8, 0(\$16)	lw \$8, 0xc(\$16)	bubble / nop	bne \$16, \$4, cont

יש לשים לב לסדר הפקודות המושפע מ $pc+8$: כלומר – רצף ביצוע הפקודות הוא 13, 15, 17 ואז בעקבות ה-BNE פקודות 14, 16 ו-18
 כאשר פקודת bne (שורה 17) היתה בשלב ID, זוהי הצורך לבצע דילוג, ולכן צריך היה לנקות את הפקודה שנכנסה לצנרת (כתובת 0x00400418, נקודה 1 בסעיף 9.1), ולכן בוצע flush באמצעות if/flush. זו הסיבה שיש nop בין פקודת BNE לפקודת LW

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

בתרשים, המבוסס על תרשים 4.66, מסומנות 6 נקודות (1-6). עליכם לסמן **בטבלה**, שבעמוד הבא, את הערכים העוברים על הקווים המסומנים **בבסיס הקסה** (3 נקודות לסימון), בהנחה שהמידע נבדק לקראת סוף פעימות השעון השביעית (7).

שימו לב: העתיקו את סדר הפקודות שרשמתם לחלק העליון של התרשים, כפי שרשמתם בסעיף 9.2, כדי להקל עליכם את המעקב. בהתאם לנקודות המסומנות ובהתאם לפקודות בכל שלב ושלב, השלימו את הערכים בטבלה.

במקרה של טעות בסעיף 9.2, ובהנחה שיש תאימות מלאה בין תשובתכם בסעיף 9.2 לסעיף 9.3, הפתרון לא יפסל. כל ערך התואם את הטעות יקבל 2 מתוך 3 הנקודות של הסעיף (טעות נגררת).

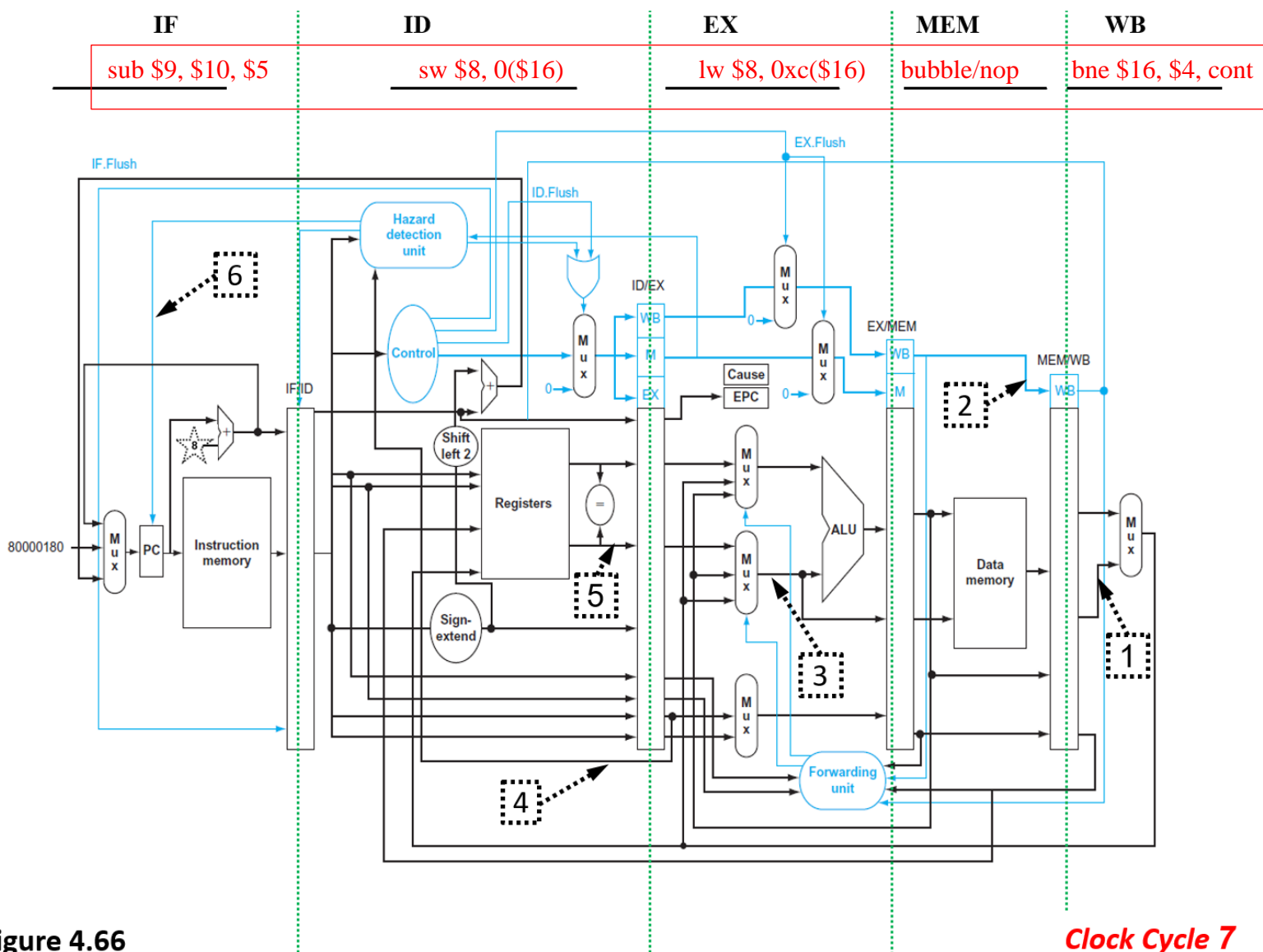


Figure 4.66

Clock Cycle 7

1	0x1000FFFC – תוצאת החיסור של פקודת ה bne (0x10010000 – 0x4)
2	0 – כל הדגלים מאופסים – זו פקודה מאופסת – bubble/nop
3	0x0000000c – הערך המיידי של פקודת ה LW

4	0x8 – מספר אוגר rt של פקודת LW – הנכנס ל HDU לאפשר זיהוי load use
5	0x8 – ערך התחלתי של אוגר \$8, הערך לפני שפקודת LW מתבצעת ויעודכן במחזור שעות מאוחר יותר
6	0x0 – ערך דגל IF Flush כדי למנוע כתיבה על אוגר PC כדי שפקודת ה sub תשאר בשלב IF. מתרחש במקביל לעצירת פקודת SW – כדי לאפשר בועה בין LW ו-SW

9.4 מעבד צנרת (3 נקודות)

בהמשך לסעיפים 9.2 ו-9.3, רשמו בטבלה כאן, בדומה לסעיף 9.2, עבור מחזור שעון 8 – איזו פקודה תהיה בכל שלב במחזור שעון זה. יש להוסיף הסבר קצר (2-3 משפטים) המסביר מודע זה מיקום הפקודות בצנרת.

שלב 1 - IF	שלב 2 - ID	שלב 3 - EXE	שלב 4 - MEM	שלב 5 - EX
sub \$9, \$10, \$5	sw \$8, 0(\$16)	bubble	lw \$8, 0xc(\$16)	bubble / nop

יש לשים לב לסדר הפקודות המושפע מ pc+8: כלומר – רצף ביצוע הפקודות הוא 13, 15, 17 ואז בעקבות ה BNE פקודות 14, 16 ו-18
בהמשך לסעיף 9.2 ו-9.3, ה HDU זיהה במחזור שעון 7 – סוגיה של load use – עקב אוגר \$8 שפקודת ה LW שומרת בו ערך, ופקודת ה SW משתמשת בתוכנו מיידית לאחר מכן. עקב סוגיה data hazard על אוגר זה, נדרש bubble

שאלה 10 (20 נקודות)

נתונה פרוצדורה בשם whatDoldo. הפרוצדורה מקבלת פרמטר אחד:

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

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

str: .asciiz "a4D]h{Q>w^Z"

בתוכנית הראשית – אופן זימון הפרוצדורה:

```
la    $a0, str1
jal   whatDoldo
```

קוד הפרוצדורה:

```
1      whatDoldo:
2      #    #a0 - null terminated string address
3          xor    $v0, $v0, $v0
4      loopStart:
5          lbu     $t0, 0($a0)
6          beq     $t0, $zero, loopEnd
7          blt     $t0, 'A', cont1
8          ble     $t0, 'Z', cont2
9          blt     $t0, 'a', cont1
10         bgt     $t0, 'z', cont1
11         andi    $t0, $t0, 0xdf
12         j       cont3
13     cont1:
14         addi    $v0, $v0, 1
15         j       cont4
16     cont2:
17         ori     $t0, $t0, 0x20
18     cont3:
19         sb      $t0, 0($a0)
20     cont4:
21         addi    $a0, $a0, 1
22         j       loopStart
23     loopEnd:
24         jr      $ra
```


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 5. זו כמות התווים שאינם אותיות (לא אות גדולה ולא אות קטנה). כאשר התו "הנוכחי" בלולאה אינו אות, הקוד מגיע לתווית cont1, שם מגדילים את המונה באוגר \$v0 ב- 1 (עבור כל תו השונה מאות)

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 אותה מחרוזת כאשר כל אות קטנה במקור – תהייה כעת אות גדולה, ולהיפך – כל אות גדולה במקור – תהייה כעת אות קטנה. אין שינוי בתווים שאינם אותיות.

המחרוזת המקורית תיראה כעת: A4dJH{q>W^z

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 מחזורת מסוג (asciiz) null terminated: הפרוצדורה עוברת על כל התווים במחזורת ומבצעת 3 פעולות: כל אות קטנה במחזורת, מוחלפת בזיכרון באות גדולה, כל אות גדולה במחזורת מוחלפת בזיכרון באות קטנה. היה והתו אינו אות (גדולה/קטנה) – הוא נשאר ללא שינוי, והפרוצדורה סופרת כמה תווים כאלה קיימים במחזורת. עם סריקת המחזורת, מוחזר ערך המציין את כמות התווים שאינם אותיות.

10.4 (5 נקודות)

כעת, בוצע שינוי של הגדרת המחזורת. במקטע הנתונים מופיע ההגדרה הבאה של המחזורת `str`:
`str: .ascii "aBcD$&)?QRs"`

לא בוצע כל שינוי בזימון הפרוצדורה וכן לא בוצע כל שינוי בפרוצדורה עצמה. לאור זאת:

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)

הלוגיקה של ניהול הלולאה בפרוצדורה מתבססת על כך שהמחזורת הינה null terminated. כלומר, יש NULL בסוף המחזורת. בסעיף זה הגדרת המחזורת הינה ללא null בסוף המחזורת – המשמעות – הלולאה לא תסתיים כמו שצריך בסוף המחזורת, והלולאה תמשיך לעבוד: לכאורה לולאה אין סופית, או עד "שבמקרה" יהיה ערך 0 או NULL בזיכרון ואז יסיים.

ב. מה יהיה בזיכרון, במקטע הנתונים (data). ערך המחזורת שהוגדרה (בהתייחסות, כמובן, לתשובתכם בתת סעיף א')

בשלב ראשון, בחלק בו מוגדרת המחזורת, יעבוד בהתאם למשימת הפרוצדורה וחלק זה יהפוך ל:
`AbCd$&)?qrS`. לאחר חלק זה הדברים תלויים במה שיש בפועל בזיכרון

ג. מה יוחזר באוגר \$v0 (בהתייחסות, כמובן, לתשובתכם בתת סעיף א')

יש 2 אפשרויות: היה ויש לולאה אין סופית – לא יחזור ערך. היה ובמקרה יש ערך 0 או NULL הפרוצדורה תחזיר ערך על בסיס כמות התווים שאינם אותיות במחזורת (4 בתוספת מה שימצא לאחר מכן