



מס' נבחן

פתרון X



מס' נבחן

חלק א: (32 נקודות)

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

חלק ב (68 נקודות) שאלות פתוחות

שאלה 9 (42 נקודות)
 9.1 (18 נקודות)

D1	0x23 (Opcode lw)	D4	0x00009000 (\$9=9*0x1000)
D2	0x00 (Instruction 11-15) ¹	D5	0x00006100 ³
D3	0x00006500 ²	D6	0x00895022 ⁴

1 סיביות 11-15 הן 5 הסיביות הגבוהות משדה הערך המיידי שערכו 0x0100 (בפקודה lw) ולכן 5 הסיביות ערכן 0.

2. חישוב הערך ה branch target נעשה הפעם ע"י $\text{branch target} = \text{PC} - 4 + \text{s.e(imm)} * 4$

$$0x00006104 - 4 + 0x00000100 * 4 = 0x00006500$$

3. מהפקודה הראשונה $\$10 = \$3 + \$3 = 0x00006000$

בפקודה lw הערך המחושב ב ALU הוא: $0x00006100 = \$10 + 0x100$ וזו הכתובת אליה ניגש בזיכרון.

4. במקום בזיכרון 0x00006100 נמצאת הפקודה sub \$10,\$4,\$9 (ממוספרת 2) קידוד הפקודה כמופיע בעמודת ה

code בתמונת הזיכרון. **0X00895022**



9.2 (24 נק')

Z1	0x000060f8 (pc-4 of the or \$6,\$7,\$8)	Z5	0x09 (rt of the bubble)
Z2	0x0A (Rd of add \$10,\$3,\$3)	Z6	0x00004000 (\$rs=\$4=4*0x1000)
Z3	0x00004000 (\$rs=\$4=4*0x1000)	Z7	0x0 ²
Z4	0x00009000 ¹ (\$rt=\$9=9*0x1000)	Z8	0x3 ³ (lw: 1=RegWrite 1=Memtoreg)

1 זה עדיין לא הערך המעודכן של \$9 שיועבר קדימה ע"י הפקודה lw רק בפעימת השעון הבאה.

2 הבקורות של הבועה מאופסות

3 למי שהתייחס רק ל RegWrite וכתב 1 תשובתו התקבלה.

10 (26 נק')

\$a0= string null terminate address

Reverse_word:

```
li $v0,11 #syscall 11 print char
move $t0,$a0 #backup $a0 to $t0 ($a0 use in syscall 11)
li $t1,0x20 # $t1=' '=space
li $t2,0 # $t2= number of char in word
```

get_word:

```
lbu $t3,($t0)
addi $t2,$t2,1 #char counter ++
addi $t0,$t0,1 #next char address
beq $t3,$t1,End_Of_Word # $t3 = space
bnez $t3, get_word #if not null terminate
```

End_Of_Word:

```
addi $t4,$t0,-1 # $t4 point to the end of the reverse word
addi $t2,$t2,-1 # $t2 the number of char in the word
```

print_reverse:

```
addi $t4,$t4,-1 #go reverse
lbu $a0,($t4) #get char
andi $a0,$a0,0xDf # reset bit 6 for capital letter
syscall
addi $t2,$t2,-1 #char counter --
bnez $t2,print_reverse
move $a0,$t1 #print space
syscall
beq $t3,$t1 get_word #in case of space there are more word.
jr $ra
```



מס' נבחן

שאלון X

חלק א: (32 נקודות)

בחלק זה שמונה שאלות יש לבחור את התשובה הנכונה מבין האפשרויות. משקל כל שאלה 4 נקודות. יש לסמן באופן ברור ב-X בטבלה שלפניכם.
הערה: יש לסמן רק אפשרות אחת לכל שאלה!

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

שאלה 1

מה יהיה ערכו של \$2 בסיום קוד זה ?

xor \$1, \$1, \$2
xor \$2, \$1, \$2

- א. ערכו יהיה 0.
ב. מתבצע היפוך סימן משלים לשתיים לערך המקורי של \$2. כלומר ערך חיובי יהפוך לשלילי וערך שלילי יהפוך לחיובי.
ג. ערכו יהיה תוצאת החיסור של \$2 המקורי מ \$1 המקורי.
ד. \$2 יקבל את ערכו המקורי של \$1.
ה. ערכו יהיה תוצאת החיסור של \$1 המקורי מ \$2 המקורי.

**שאלה 2**

באסמבלי המורחב של מעבד MIPS, פסיאודו הפקודה `la $a0, L1` טוענת את הכתובת של התווית `L1` לאוגר `$a0`. נרצה לממש את הפקודה `jal func` בעזרת קוד חלופי המשתמש בפסיאודו פקודה זו. איזה קטע קוד מממש את הפקודה `jal func`?

א.

```
j func  
la $ra, L1
```

`L1:`

ב.

```
la $31, L1  
jr $ra
```

`L1:`

ג.

```
la $ra, func  
j func
```

`L1:`

ד.

```
la $ra, L1  
j func
```

`L1:`

ה.

```
la $ra, func  
j L1
```

`L1:`**שאלה 3**

על מעבד מסוים מריצים לולאה, המכילה 3 סוגי פקודות, A, B ו-C. קצב השעון של המעבד הוא 2GHz. הלולאה מכילה 10 פקודות, ומורצת שלושה מיליארד ($3 \cdot 10^9$) פעם. התפלגות הפקודות כמופיע בטבלה:

סוג הפקודה	מספר הפקודות	ה-CPI של סוג הפקודה
A	50%	1
B	30%	3
C	20%	4

מה יהיה זמן ריצת הלולאה?

א. 3.3[sec]

ב. 11[sec]

ג. 33[sec]

ד. 40[sec]

ה. 120[sec]

**שאלה 4**

איזה מהטיעונים הבאים לגבי הממד Speedup (מדד ההאצה) נכון? (קיים רק טיעון אחד נכון)

- א. אם Speedup גדול מ-1, פירושו כי שופרו הביצועים.
- ב. אם Speedup גדול מ-0, פירושו כי שופרו הביצועים.
- ג. אם Speedup קטן מ-1, פירושו כי שופרו הביצועים.
- ד. Speedup חייב להיות מספר שלם (Integer).
- ה. מדד ה-Speedup ניתן ביחידות של שניה.

שאלה 5

נתונה פקודה באסמבלי של ה-MIPS, שקידודה לשפת מכונה הוא 0x0BEFFF20. נריץ פקודה זו על מעבד חד-מחזורי, כמתואר בתרשים 4.24 בנספח לבחינה, איזה ערך ייכנס בכניסה התחתונה ל-ALU (הכוונה לערך המגיע מהיציאה של המרכב של ALUsrc) במהלך ביצוע פקודה זו? ניתן להסתכל גם בטבלאות 4.12 ו-4.18 בנספח לבחינה.

- א. ערך אוגר 31.
- ב. ערך אוגר 15.
- ג. 0xffffffff20
- ד. 0x0000ff20
- ה. לא ניתן לדעת, מאחר שערך ALUsrc הינו don't care.



מס' נבחן

שאלה 6

נתון קטע קוד :

I1	add	\$1,\$2,\$3
I2	sw	\$3,40(\$1)
I3	label: \$\$\$	(Undefined Instruction)
I4	sub	\$2,\$4,\$3
I5	lw	\$3,16(\$4)

Exception_Handler:

0x80000180 sw \$26,1000(\$0)

I3 היא פקודה לא מוגדרת (undefined instruction).

sw \$26,1000(\$0) היא הפקודה הראשונה של התכנית המטפלת בחריגה (Exception Handler).

בהנחה שהקוד רץ על הצנרת המתוארת **בתרשים 4.66** בנספח לבחינה. ובהנחה שלאחר המעבר ל

Exception Handler לא חוזרים יותר לקוד זה, מבין הפקודות I1-I5 אילו פקודות מתבצעות עד סופן?

(הכוונה שהפקודה עוברת את כל השלבים עד לשלב החמישי שלב ה WB)

א. I1 I2

ב. I1 I2 I3

ג. I1 I2 I4

ד. I1 I3

ה. I1 I5

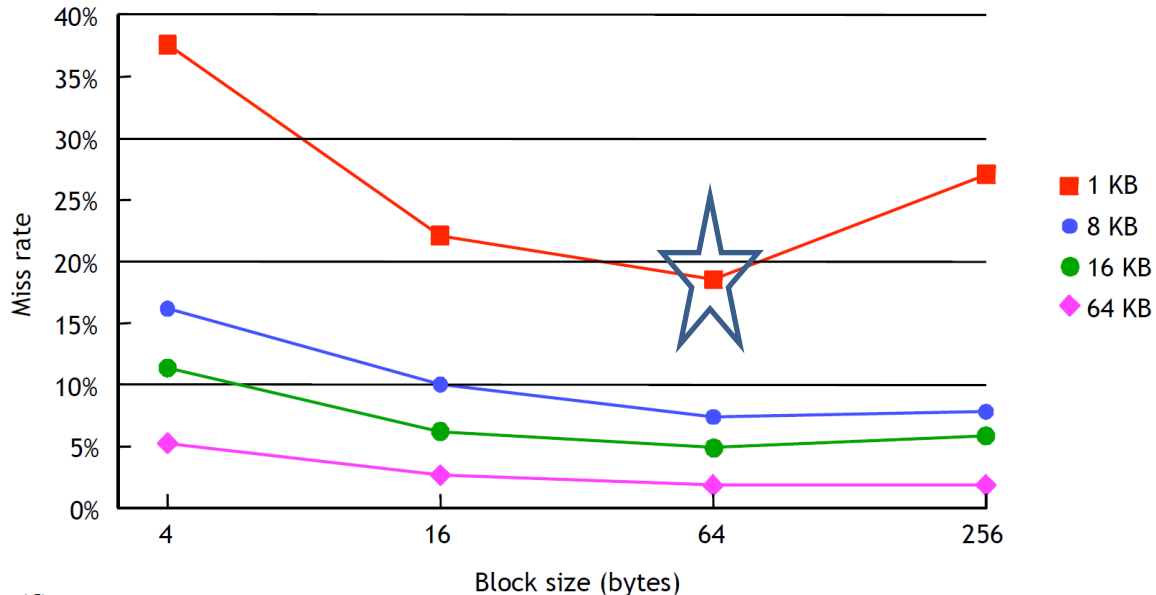


התרשים והנתונים הבאים הם עבור שאלות 7-8:
מצורף תרשים המתאר את שיעור ההחטאות (miss rate) בגישות למטמון כפונקציה של גודל הבלוק בבתים (byte). כל קו מחובר מציין גודל מסוים של מטמון, המופיע מימין בהתאמה: 1KB, 8KB וכו'. הכוונה בגודל הוא לאזור הנתונים בלבד של המטמון.

2-way set associative

- הזיכרונות בנויים בתצורת

- הזיכרון הראשי הממופה הוא בגודל 32 סיביות.



שאלה 7

כמה סיביות יש בשדה האינדקס בזיכרון המטמון (2-way) בגודל 1KB, שבו גודל בלוק מוגדר כ- 64 בתים, (הנקודה המוקפת בכוכב)?

- 2
- 3
- 4
- 5
- 6

שאלה 8

בתרשים בעמוד הקודם עבור המטמון של 1KB הגדלת הבלוקים מ-64 בתים ל-256 בתים גורמת לעלייה ב-miss rate. מה הסיבה לעלייה זו?

- ה-miss penalty הוא פי ארבע יותר גדול (לא מוצג בצורה ישירה בתרשים).
- בגודל בלוק של 256 בתים עוברים כבר לסכמת write back, שלא מוצגת בצורה ישירה בתרשים, ושם כמו כן ה-miss rate עולה באופן דרמטי.
- לוקליות במרחב (spatial locality).
- יש יותר miss Tag.
- יש יותר miss valid.

**חלק ב (68 נקודות) שאלות פתוחות.**

את התשובות לחלק זה יש לכתוב בשאלון הבחינה, במקום המיועד.

שאלה 9 : (42 נקודות)

נתונה תמונת הקוד שלהלן הלקוחה מה MARS באסמבלי של ה mips :

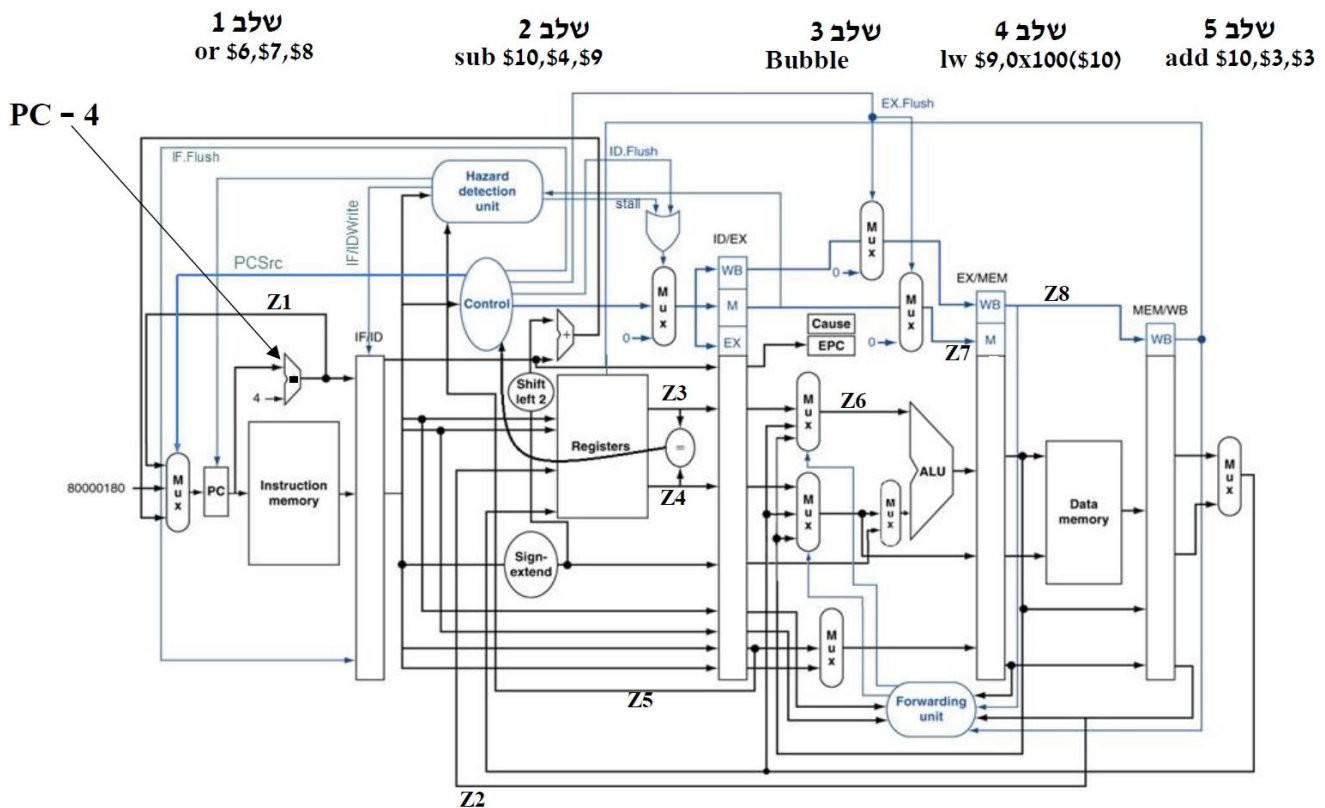
Address	Code	Basic	Source
0x000060fc	0x00e83025	or \$6,\$7,\$8	1 or \$6,\$7,\$8
0x00006100	0x00895022	sub \$10,\$4,\$9	2 sub \$10,\$4,\$9
0x00006104	0x8d490100	lw \$9,0x00000100(\$10)	3 lw \$9,0x100(\$10)
0x00006108	0x00635020	add \$10,\$3,\$3	4 add \$10,\$3,\$3

הפעם המעבד עובד בצורה סידרתית הפוכה בזיכרון כלומר סדר הפקודות בקוד זה הוא מכתובת גבוהה כלפי כתובת נמוכה (פקודה 4 ראשונה ופקודה 1 אחרונה בקוד) . לצורך זה יש מחסר של PC פחות ארבע במקום המחבר.

- ערך כל אוגר לפני ביצוע פקודה 4 בקוד הוא מספרו כפול 0x1000 (אוגר 1 ערכו 0x00001000 אוגר 2 ערכו 0x00002000 ערך אוגר 3 0x00003000 וכן הלאה)
- ניתן לגשת לכל מרחב הזיכרון בפקודות lw sw (כל עוד הכתובת מתחלקת ב 4) .
- **נתונים אלו תקפים לכל סעיפי השאלה.**



9.2 (24 נק') התרשים שלפניכם בעמוד זה מבוסס על תרשים 4.66 בספר בתוספת המרבב ALUSrc. לתרשים זה נחליף את רכיב המחבר של 4 ל PC ברכיב מחסר 4. בתרשים מתוארת סוף פעימת השעון החמישית בביצוע הקוד שבשאלה, על פי הנתונים המופיעים בתחילת השאלה. (בפעימה זו הפקודה add \$10,\$3,\$3 נמצאת בשלב 5 שלב ה-WB.) ניתן להניח שסיכוני הנתונים בהרצת הקוד נפתרים ע"י יחידת העברה קדימה, יחידת איתור סיכונים (HDU, Forwarding Unit) וחציית מקבץ האוגרים. בתרשים 8 סימונים Z1-Z8 על קווים מסוימים על המעבד. יש למלא בטבלה שבתחתית העמוד את הערכים על קווים אלו. יש להציג בבסיס הקסא (3 נק' לסימון), ערך לא ידוע יש לסמן ב X.



שימו לב סדר ביצוע הפקודות הוא הפוך !

Z1		Z5	
Z2		Z6	
Z3		Z7	
Z4		Z8	

הבהרה: שימו לב שהרכיב בשלב ה fetch מבצע חיסור של 4 מ PC (במקום חיבור)



שאלה 10 (26 נק')

עליכם לכתוב פרוצדורה בשם Reverse_word, הפרוצדורה מקבלת כפרמטר באוגר \$a0 כתובת של מחרוזת, שנגדיר אותה כמשפט.

משפט מורכב מאוסף של מילים (מילה = רצף של אותיות גדולות ו\או קטנות).
 המחרוזת מסתיימת בקוד האסקי null. (0x0 = '\0').
 המילים מופרדות ביניהן בתו רווח אחד (0x20 = ' ').

על הפרוצדורה להדפיס את המחרוזת כאשר האותיות בכל מילה רשומות בסדר הפוך ובאותיות גדולות בלבד.
 בהדפסה יש להקפיד על רווח אחד בין המילים ההפוכות.

לדוגמא עבור המחרוזת:

str: "Function allows you to evaluate".asciiz

יודפס למסך:

ETAULAVE OT UOY SWOLLA NOITCNUF

אין צורך לבדוק חוקיות משפט ניתן להניח מבנה מחרוזת חוקי כלומר:

- מלבד רווח (0x20) וקוד אסקי null (0x00) יש במחרוזת רק אותיות a-z (קוד אסקי 0x7A-0x61) ו A-Z (קוד האסקי 0x5A-0x41)
- תווי הרווח לא יופיעו בתחילה או סוף משפט ותו רווח בודד יפריד בין מילים בתוך המשפט.

נקודות לתשומת לב:

- יש לכתוב את הפרוצדורה בלבד ע"פ הכללים (מוסכמות) שלמדנו.
- לצורך הדפסה יש להשתמש ב syscall 11, המדפיסה תו בודד (שימו לב ש 11 syscall משתמש ב \$a0)
- יש לתעד את הפרוצדורה.
- מותר להשתמש בפסיאודו פקודות.
- שימו לב שאין לשנות את המחרוזת אלא רק להדפיס את המילים באותיות גדולות הפוך.
- נתון עזר: בהמרה לייצוג בינארי של קודי אסקי ההבדל בין אות גדולה (A-Z קוד האסקי 0x41-0x5A) לאות קטנה (a-z קוד אסקי 0x61-0x7A) הוא סיבית אחת (הסיבית השישית באינדקס 5 כלומר יש הפרש של 0x20). אפשר בעזרת פעולות לוגיות בסיסיות להפוך אות גדולה לקטנה או להיפך באמצעות שינוי סיבית זו.

את הפרוצדורה בלבד יש לכתוב בעמוד הבא:

פתרון שאלה 10:

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

[illegible]

טיוטא:



מס' נבחן



מס' נבחן

טיוטא:



מס' נבחן

טיוטא: