

שאלון X

חלק א: (48 נקודות)

בחלק זה שתיים עשרה 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					
9					
10					
11					
12					

נתונה תמונת סגמנט הקוד הבאה שנלקחה מה-MARS:

Source	Basic	Code	Address
0x00456708	0x240b000d	addiu \$11,\$0,0x0000000d	1 li \$t3,0xd
0x0045670c	0x240c0056	addiu \$12,\$0,0x00000056	2 li \$t4,0x56
0x00456710	0x016c5824	and \$11,\$11,\$12	3 and \$t3,\$t3,\$t4
0x00456714	0x000b5900	sll \$11,\$11,0x00000004	4 sll \$t3,\$t3,4
0x00456718	0x20010064	addi \$1,\$0,0x00000064	5 bgt \$t3,100,moran
0x0045671c	0x002b082a	slt \$1,\$1,\$11	
0x00456720	0x1420000c	bne \$1,\$0,0x0000000c	

שאלות 1 ו 2 מתייחסות לתמונה זו.

שאלה 1

- מה ניתן להגיד על ההוראה בשורה 5 בקובץ המקור `bgt $t3,100,moran`?
- זוהי כמובן פסיאודו פקודה, ניתן לדעת שלא תתבצע קפיצה לתווית moran.
 - זוהי כמובן פסיאודו פקודה, ניתן לדעת שתתבצע קפיצה לתווית moran.
 - זוהי כמובן פסיאודו פקודה, לא ניתן לדעת האם תתבצע קפיצה לתווית moran.
 - זוהי כמובן פקודה בסיסית, לא ניתן לדעת האם תתבצע קפיצה לתווית moran.
 - זו לא פקודה בסיסית של ה MIPS וגם לא פסיאודו פקודה הנתמכת בסביבת העבודה MARS, ולכן התכנית לא תעבור הידור ותתקבל הודעת שגיאה שלא ניתן להשתמש בקבוע בפקודת bgt.

שאלה 2

על סמך הנתונים בתמונת סגמנט הקוד, מה כתובת התווית moran?

- 0x0045670c
- 0x00456750
- 0x00456754
- 0x00456768
- לא ניתן לדעת על סמך הנתונים בשאלה.

שאלה 3

בישיבה שערכו ב IEEE החליטו לשנות את תקנה 754 לייצוג מספרים בנקודה צפה ב 32 סיביות כך שמהערך בשדה ה exponent נפחית רק 100 במקום 127.

מה נכון להגיד לגבי שינוי זה ?

- א. נוכל לייצג מספרים קטנים יותר גם בהצגה הלא מנורמלת.
- ב. אלגוריתם החיבור חיסור צריך להשתנות עקב השינוי.
- ג. לא ניתן לבנות חומרה שתתמודד בשינוי זה ולכן השינוי אינו ישים.
- ד. נוכל לייצג מספרים גדולים יותר עד כמעט 2^{155} .
- ה. נוכל להציג מספרים בדיוק רב יותר אחרי הנקודה.

שאלה 4

במעבד העובד בתדר 2.5GHz נמדדו הנתונים שלהלן עבור הרכב פקודות בתכנית מסוימת :
 $8 \cdot 10^7$ פקודות אריתמטיות לוגיות $5 \cdot 10^6$ פקודות store, 10^7 פקודות load, ו- $8 \cdot 10^6$ פקודות branch.
 בהנחה שפקודות אריתמטיות לוגיות לוקחות שני מחזורי שעון, גישה לזיכרון (load, store) ארבעה מחזורי שעון ו- branch שלושה מחזורים.
 מה יהיה זמן הביצוע **בשניות** של תכנית זו?

- א. 0.0976
- ב. 0.0896
- ג. 0.0880
- ד. 0.0656
- ה. 0.0336

שאלה 5

נתונה מכונה בה 30% מהמערכת לא ניתן למקבל, לכן לא ניתן לשפר את הביצועים של חלק זה של המכונה. את השאר ניתן למקבל.
 מה מדד ההאצה (speedup) עם מקבול של פי 4?
 הערה : ההנחה היא שהמקבול ישפר את הביצועים של המכונה באותו היחס.

- א. 1
- ב. 2.10526
- ג. 2.222
- ד. 3.10526
- ה. 4

שאלה 6

נתון מעבד MIPS חדש עם ארכיטקטורת-פקודות (ISA) שונה במקצת ממעבד ה MIPS אותו למדנו בסמסטר זה. במעבד החדש, כל הפקודות בשפת מכונה מקודדות בעזרת 32 סיביות (כמו ה MIPS שלמדנו) כאשר החלוקה לשדות בפורמטים השונים זהה למעבד אותו למדנו. בשונה מהמעבד שלמדנו, במעבד החדש יש 32 אוגרים ברוחב 64 סיביות. בהנחה שלא הקטנו את כמות הפקודות הנתמכות, מבין הפקודות מסוג R איזה פקודה אינה נתמכת במלואה?

א. slt

ב. sll

ג. sub

ד. and

ה. הפקודות בסעיפים א-ד נתמכות בצורה מלאה.

שאלה 7

בתרשים 4.60 בנספח לבחינה בכניסה לאוגר הצגת ID/EX נכנס פעמיים הערך IF/ID.RegisterRt. מדוע יש צורך להציג סימון זה פעמיים?

א. היות ובתרשים נשמט המרבב של ALUSrc (ראו תרשים 4.51 בנספח) אז למעשה במקור השכפול של ערך Rt היה אמור להיכנס למרבב זה.

ב. בפקודות sw יש צורך לדעת את אוגר המקור rt, לצורך מימוש סיכוני נתונים.

ג. על מנת להדגיש את ההבדל בין הפורמטים השונים R ו I שבהן אוגר היעד לכתובה למקבץ האוגרים הוא שונה.

ד. למעשה קו אחד זה הערך של האוגר כלומר \$rt והקו השני משמעותו הוא מספר האוגר.

ה. על מנת להדגיש שיש צורך בערך זה להמשך, לכמה רכיבים. יחידת העברה קדימה, המרבב של Regdst ויחידת ה HDU.

שאלה 8

איזה משפט מבין המשפטים הבאים אינו נכון לגבי מוסכמת עבודת המחסנית (stack) בתכנות באסמבלי של MIPS?

א. המחסנית הינה מוסכמת תוכנה בלבד, ואין לה תמיכה בחומרה.

ב. המחסנית מתמלאת מכתובות גבוהות בזיכרון כלפי כתובות נמוכות.

ג. מצביע ראש מחסנית הינו אוגר מספר 29.

ד. גודל איבר במחסנית הינו 16 סיביות.

ה. לא קיימות הפקודות push pop באסמבלי של ה MIPS.

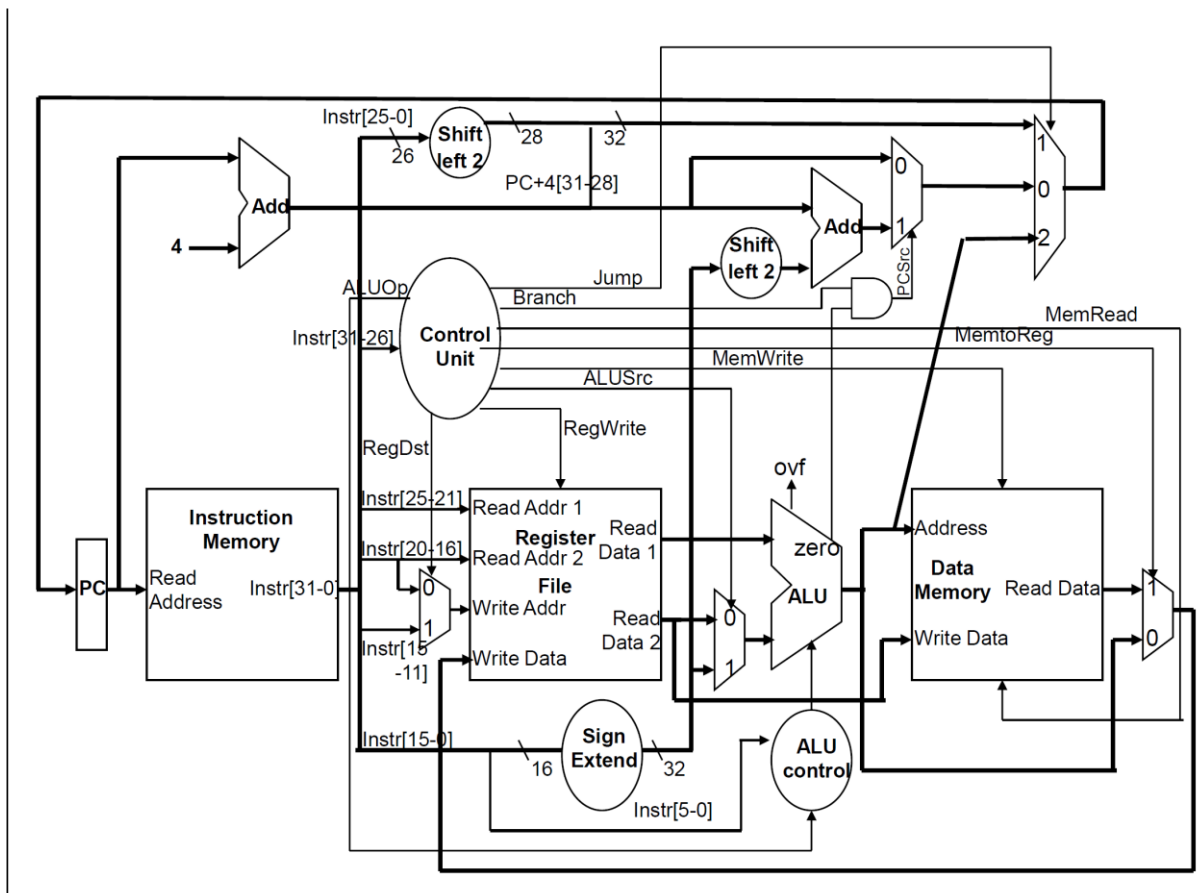
שאלה 9

מצורף תרשים של מעבד MIPS החד-מחזורי עם הרחבת המרביב של Jump, כך שבכניסה הנוספת (כניסה מספר 2) מועבר ערך היוצא מה-ALU (ALUResult). עבור פקודה כלשהי העובדת על-פי תרשים זה נתונים קווי הבקרה האלה:

RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp	Jump
0	1	0	0	0	0	1	01	2

מה תבצע הפקודה?

- הפקודה תבצע את פעולת beq. (כלומר אם ערך $rs = rt$, תתבצע קפיצה לפי חישוב ה-branch target, אם לא, נתקדם לפקודה הבאה)
- תתבצע קפיצה לכתובת, שהיא הסכום של ערכי rs ו- rt (בהנחה שכתובת זו מתחלקת בארבע).
- תתבצע קפיצה לכתובת, שהיא ההפרש בין ערך rs לבין הערך המידי לאחר הרחבת סימן (בהנחה שכתובת זו מתחלקת בארבע).
- תתבצע קפיצה לכתובת, שהיא ההפרש בין ערך rs לבין ערך rt (בהנחה שכתובת זו מתחלקת בארבע).
- תתבצע קפיצה לכתובת שהיא הסכום של ערך rs והערך המידי לאחר הרחבת סימן (בהנחה שכתובת זו מתחלקת בארבע).



שאלה 10

נתון מעבד MIPS, העובד בתדר של 2Ghz עם זיכרון מטמון ברמה אחת (L1 cache), המחולק למטמון לפקודות ומטמון לנתונים. 2% מהפקודות מוחטאות, וכן מוחטאים 4% מהנתונים. 40% מהפקודות מבצעות גישה לזיכרון המטמון לנתונים, וזמן ההחטאה (miss penalty) הוא 400 מחזורי שעות (לפקודות ולנתונים). הניחו כי ללא החטאות של זיכרון המטמון, ה-CPI הממוצע של המעבד הוא 1.4 [cc/ins]. מה יהיה זמן הריצה של תכנית עם 10^9 פקודות בנתוני שאלה זו?

א. אין אפשרות לחשב בנתוני השאלה (חסר CPI כללי).

ב. 7.7[sec]

ג. 7.9[sec]

ד. 12.7[sec]

ה. 15.8[sec]

שאלה 11

נתון זיכרון מטמון בעל המאפיינים הבאים:

- גודל **1Mbyte**, כלומר 2^{20} byte. (הכוונה כמובן רק לאיזור ה Data)
- גודל בלוק **128 סיביות**.
- אורך כתובת לגישה בזיכרון הראשי **28 סיביות**.
- הזיכרון מנוהל בשיטת מיפוי ישיר Direct Mapped.

כמה סיביות valid יש בזיכרון זה?

א. 2^{14}

ב. 2^{15}

ג. 2^{16}

ד. 2^{17}

ה. לא ניתן לדעת על סמך הנתונים.

שאלה 12

נתון זיכרון מטמון 4 אסוציאטיבי (4-Way Set associative), עם כתובות בגודל 32 סיביות המציינות כתובות בבתים. משתמשים בסיביות 0 עד 4 כדי לציין את ההיסטים (Offset), בסיביות 5 עד 11 משתמשים לאינדקס (Index) ובסיביות 12 עד 31 משתמשים לתגית (Tag). כמה סיביות יש בבלוק? (הכוונה לנתונים בלבד)

א. 32

ב. 64

ג. 128

ד. 256

ה. 512

חלק ב (52 נקודות) שאלות פתוחות.

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

את התשובות לחלק זה יש לכתוב על גבי שאלון הבחינה, במקום המיועד.

שאלה 13 : (36 נקודות)

נתון קטע הקוד שלהלן באסמבלי של ה mips :

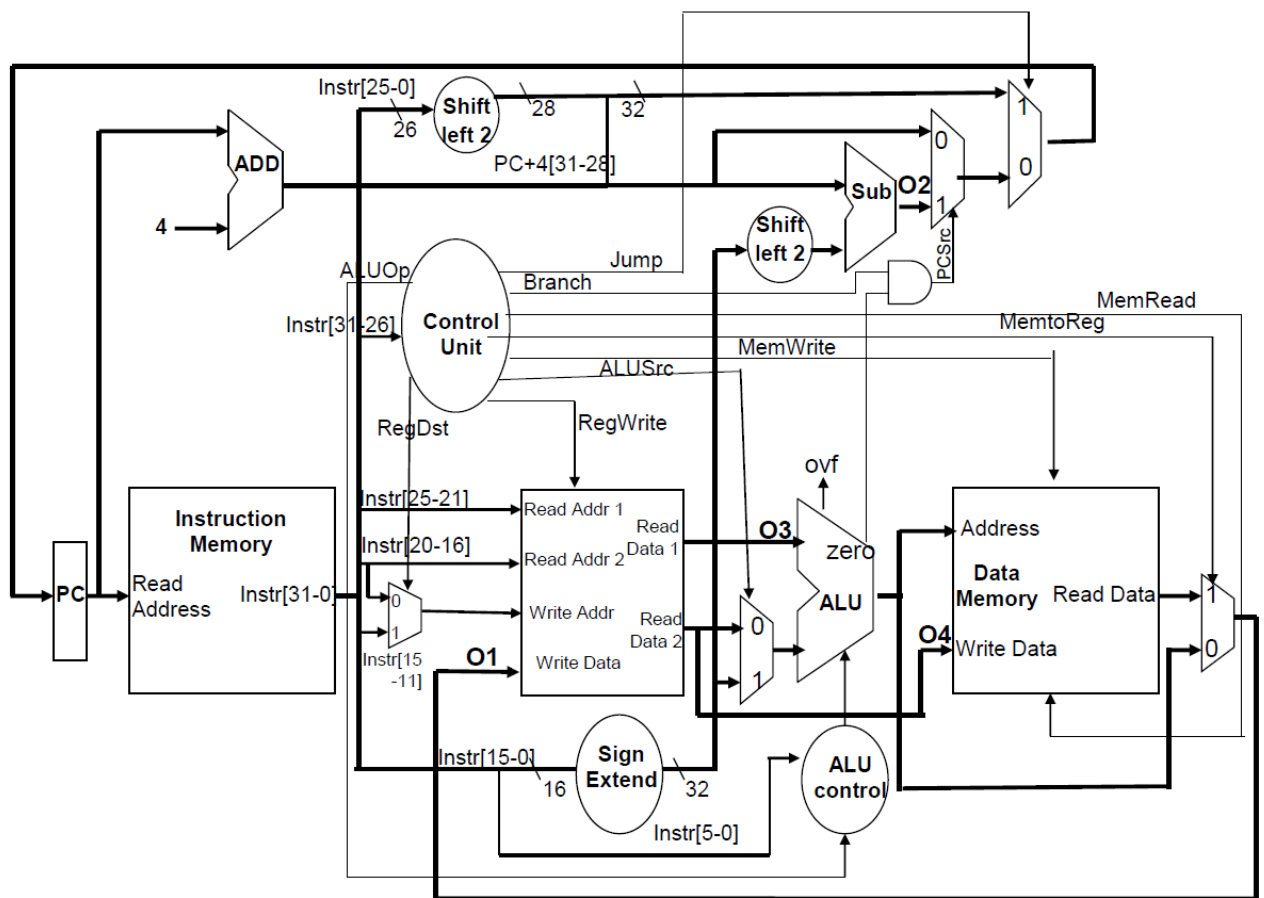
Address	Code	Basic	Source
0x00aa889c	0x8e090300	lw \$9,0x00000300(\$16)	1 lw \$9,0x300(\$16)
0x00aa88a0	0x02296020	add \$12,\$17,\$9	2 add \$12,\$17,\$9
0x00aa88a4	0x01895022	sub \$10,\$12,\$9	3 sub \$10,\$12,\$9
0x00aa88a8	0x1128fffe	beq \$9,\$8,0xfffffffffe	4 beq \$9,\$8,loop

ערך כל אוגר בתחילת ביצוע הקוד הינו מספרו כפול 0x30 (כלומר אוגר 1 ערכו 0x30 אוגר 2 ערכו 0x60 אוגר 10 ערכו 0x1E0 וכן הלאה).

- ניתן לגשת לכל מרחב הזיכרון בפקודות lw sw (כל עוד הכתובת מתחלקת ב 4).
- ערך 0x1000 המילים הנמוכות בזיכרון מאוחל באופן שכל מילה מאוחלת לכתובתה במילים.
- לדוגמא : מילה בכתובת 0 במילים (בתים 0-3 בזיכרון) מאוחלת לערך 0.
- מילה בכתובת אחד במילים (בתים 4-7 בזיכרון) מאוחלת לערך 1.
- מילה בכתובת 0x100 במילים (בתים 0x400-0x403 בזיכרון) מאוחלת לערך 0x100.

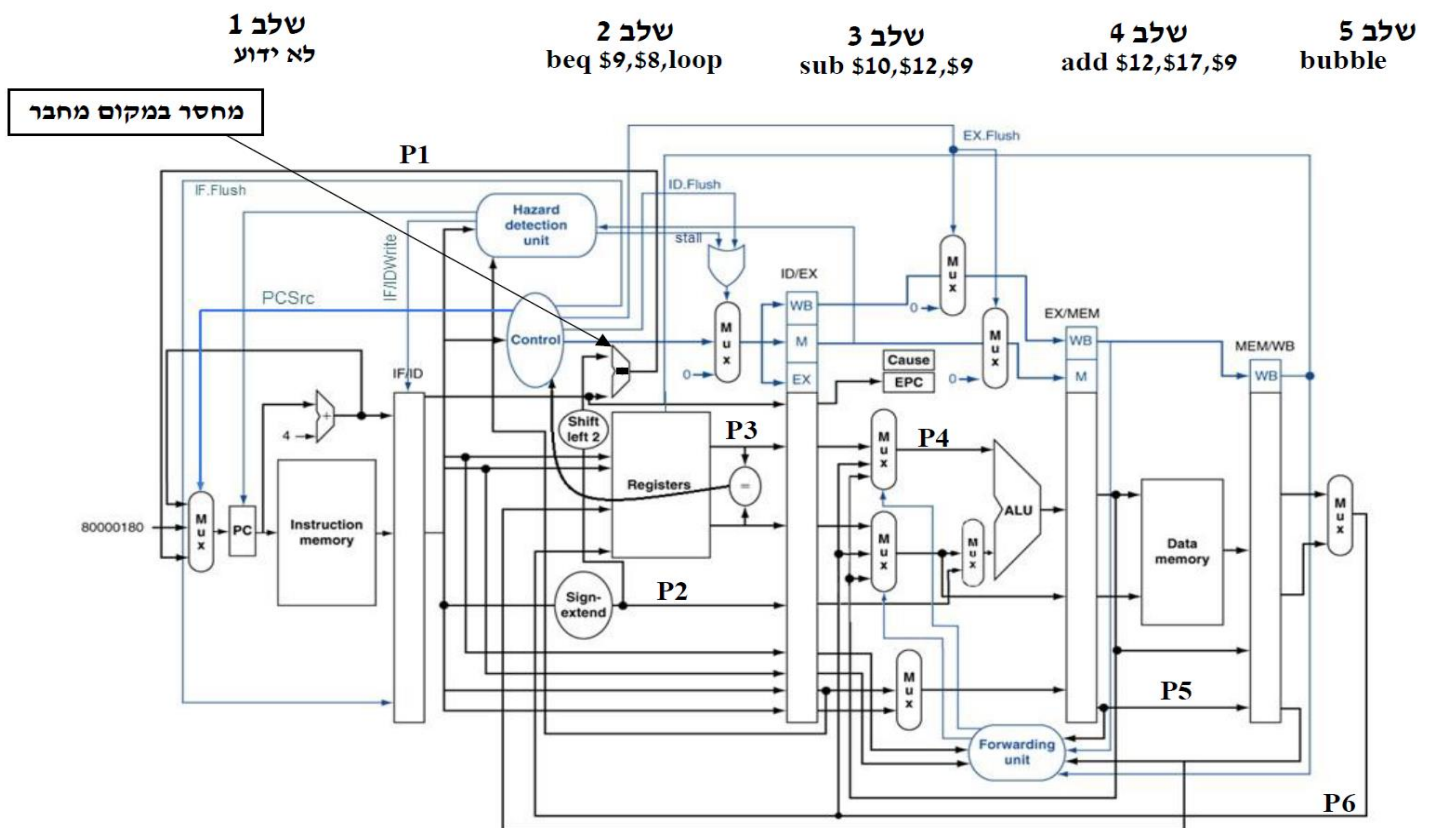
נתונים אלו תקפים לכל סעיפי השאלה.

13.1: (12 נק') בתרשים שלהלן מעבד חד-מחזורי שבו חישוב ערך ה **branch target** מתבצע עם רכיב מחסר (במקום מחבר). סדר החיסור הוא של הערך $PC + 4$ פחות הערך המידי (לאחר הזזה שמאלה פעמיים). בתרשים 4 סימונים O1-O4, על קווים מסוימים על המעבד. עליכם לחשב את הערכים העוברים על קווים אלו, בהנחה שהמידע נבדק לקראת סוף פעימת השעון בביצוע הממוספרת (1) בקוד `.lw $9,0x300($16)`. את תשובתכם יש למלא בטבלה שבתחתית העמוד, יש להציג **בבסיס הקסא** באמצעות הסימון `0x` (3 נק' לסימון). ערך לא ידוע יש לסמן ב X.



O1		O3	
O2		O4	

13.2 (18 נק') התרשים שלפניכם בעמוד זה מבוסס על תרשים 4.66 המשופר בנספח לבחינה. בתרשים חישוב ערך ה- **branch target** מתבצע עם רכיב מחסר (סדר החיסור כמתואר בסעיף 13.1). החץ בתרשים מסמן רכיב זה. בתרשים מתוארת סוף פעימת השעון ה- **השישית** בביצוע הקוד שבשאלה, **על פי הנתונים המופיעים בתחילת השאלה**. (בפעימה זו נמצאת בועה 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. (לדוגמא הבועה בשלב 5 נוצרה כתוצאה מסיכון נתונים).
בתרשים 6 סימונים P1-P6 על קווים מסוימים על המעבד. יש למלא בטבלה שבתחתית העמוד את הערכים של קווים אלו. יש להציג **בבסיס הקסא** (3 נק' לסימון), ערך לא ידוע יש לסמן ב- X.



P1		P4	
P2		P5	
P3		P6	

במידה ופקודה 3 בקוד תשתנה להיות sub \$9,\$12,\$9 וכעת הקוד יראה :

Address	Code	Basic	Source
0x00aa889c	0x8e090300	lw \$9,0x00000300(\$16)	1 lw \$9,0x300(\$16)
0x00aa88a0	0x02296020	add \$12,\$17,\$9	2 add \$12,\$17,\$9
0x00aa88a4	0x01848022	sub \$9,\$12,\$9	3 sub \$9,\$12,\$9
0x00aa88a8	0x1128fffe	beq \$9,\$8,0xfffffffffe	4 beq \$9,\$8,loop

בהנחה שהקוד מורץ על תרשים צנרת 4.66 המשופר כמתואר בסעיף הקודם, איזה סיכון נתונים חדש קיים בין פקודות 3 ל 4 בקוד ?

מהו הפתרון האפשרי לסיכון נתונים זה? (יש לתת קווים כלליים לפתרון).

[illegible]

על סמך הפתרון שהצעת מהו הערך המתקבל בסוף פעימת השעון השישית על הקו המסומן ב P3 בתרשים בסעיף הקודם (13.2) ? יש להסביר את התשובה..

[illegible]

שאלה 14 (16 נק')

נתונה פרוצדורה בשם **Que_tal**. הפרוצדורה מקבלת כפרמטר באוגר \$a1 ערך כלשהו.

(באוגר \$a0 לא מועבר ערך והוא משמש לצורך עבודה בהדפסות עם syscall בתוך הפרוצדורה)

Que_tal:

```
li $t2,0
loop:
    andi $t0,$a1,1
    add $t2,$t2,$t0
    srl $a1,$a1,1
    bnez $a1,loop
#####
li $v0,1
move $a0,$t2
syscall
beqz $t2,exit #14.4
andi $t0,$t2,1
beqz $t0,print
addi $t2,$t2,-32#
sub $t2,$0,$t2
print:
li $v0,11 #
li $a0,'\n' #new line
syscall
li $a0,'*' #astrix/star
astrix:
addi $t2,$t2,-1
syscall
bnez $t2,astrix
exit:
jr $ra
```

הבהרות לגבי עבודה עם syscall 1

מתבצעת הדפסת המספר המיוצג ב 32 סיביות באוגר \$a0 כמספר עם סימן (משלים ל-2). ההדפסה כמובן לפי בסיס 10.

הבהרות לגבי עבודה עם syscall 11

מתבצעת הדפסת התו לפי הערך קוד האסקי הנמצא בבית הנמוך (סיביות במיקום 7..0) באוגר \$a0.

וודאו שהבנתם את מימוש הפרוצדורה לפני שאתם עונים על השאלות בעמוד הבא:

14.1 (4 נק')

במידה והערך באוגר \$a1 הוא 5 מה תדפיס הפרוצדורה ?

14.2 (4 נק')

במידה והערך באוגר \$a1 הוא 1024 מה תדפיס הפרוצדורה ?

14.3 (4 נק')

במידה והערך באוגר \$a1 הוא -2 (מינוס 2) מה תדפיס הפרוצדורה ?

14.4 (4 נק') במידה ונוריד את שורת הקוד `beqz $t2,exit` (מסומנת כהערה 14.4 בקוד) איזו בעיה עלולה להיווצר ?

בהצלחה !

טיוטא:

טיוטא:

טיוטא:

פתרון X

חלק א: (48 נקודות)

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

חלק ב (52 נקודות) שאלות פתוחות.

שאלה 13 : (36 נקודות)

13.1: (12 נק')

O1	$0x00000180^1$ ($\text{mem}[0x600] = 0x180$)	O3	$0x00000300$ ($\$16 = 0x10 * 0x30$)
O2	$0x00aa7CA0^2$	O4	$0x000001B0$ ($\$9 = 0x9 * 0x30$)

1 לפי נתוני השאלה הפקודה lw ניגשת לכתובת $0x600$ בבתים ($\$16 + 0x300 = 0x10 * 0x30 + 0x300$)
בכתובת זו יש את ערך המילה שזו כתובתה כלומר הערך $0x180$. ($0x600/4 = 0x180$)
2 חישוב ה branch target מבוצע הפעם עם פעולת חיסור $pc + 4 - \text{se}(\text{imm}) * 4$ כלומר:
 $0x00aa88a0 - 0x300 * 4 = \mathbf{AA\ 7CA0}$

13.2: (18 נק')

P1	$0x00aa88b4^1$	P4	$0x000004B0^3$
P2	$0xffffffff$ se(imm) of beq	P5	$0x0c$ (register RD = 12)
P3	$0x00000180^2$ ($\$9$)	P6	$0x00000930^4$

1 חישוב ה branch target מבוצע בפעולת חיסור $pc + 4 - \text{se}(\text{imm}) * 4$ כאשר הערך $0xffffffff$ הוא מינוס 2
כלומר:

$$0x00aa88ac - (-2 * 4) = 0x00aa88ac + 8 = \mathbf{0x00aa88b4}$$

2 ערך אוגר 9 נכתב למקבץ האוגרים (בפעימה החמישית בביצוע הקוד) ע"י פקודת ה lw שכבר לא נמצאת
במעבד בפעימה השישית בצנרת, ערך זה חושב בסימון O1 להיות $0x180$

3 מבוצע זיהוי של מקרה 1a בהעברה קדימה. ($\text{Ex/Mem.RegisterRD} = \text{ID/EX.RegisterRs} = 12$)
ומועבר הערך שחושב ע"י הפקודה $\text{add } \$12, \$17, \$9$ עבור אוגר 12 כלומר חיבור של ערך אוגר 17 עם אוגר
9. שימו לב שערך אוגר 9 עודכן בגלל ה load use לערך $0x180$ כמתואר בסימון O1 (ע"פ נתוני השאלה) לכן
יתבצע:

$$\$17 + \$9 = 0x330 + 0x180 = \mathbf{0x4B0}$$

למי שחישב עם הערך המקורי של אוגר 9 וקיבל $0x4E0$ התקבל ניקוד חלקי.

4 בהנחה שהבקורות של הבועה שנוצרה כתוצאה מה load use הן כולן אפס (כפי שלמדנו) אז הבועה היא
למעשה הערכים של הפקודה $\text{add } \$12, \$17, \$9$ עם בקורות אפס. כשהבועה הייתה בשלב ה EX מבוצע
חישוב של חיבור ($\text{ALUOP} = 00$) של $\$rt$ ($\text{ALUSrc} = 0$) אבל בנקודה זו יש זיהוי של מקרה 1b בין הבועה
לבין פקודת ה lw ומועברת הכתובת שחישבה ה lw כערך $\$rt$ כלומר $0x600$ (ראו בהסבר עבור O1 בסעיף
הקודם) ומתבצע חיבור $\$17 + \$9 = 0x330 + 0x600 = \mathbf{0x930}$ והיות שגם מרוב memtoreg הוא אפס אז
הבועה מוציאה ערך זה על הקו המסומן P6. נציין שעריך זה לא יכתב למקבץ האוגרים מאחר שהערך של
Regwrite=0 בבועה.

למי שסימן X בהנחה שעריך memtoreg במצב בועה הוא don't care (כמוסבר בספר הקורס ב
Elaboration (עמוד 328 במהדורות שישית או עמוד 316 מהדורה חמישית שבפועל ניתן לאפס את בקורות
האפשרור RegWrite MemWrite MemRead ואין חשיבות לבקורות המהוות סלקטור למרוב במהלך
יצירת הבועה או לבקורות (ALUOP תשובתו התקבלה חלקית.

16.3 (6 נק)

בהנחה שהקוד מורץ על תרשים צורת 4.66 המשופר כמתואר בסעיף הקודם, איזה סיכון נתונים חדש קיים בין פקודות 3 ל 4 בקוד ?

נוצר כאן סיכון נתונים שיחידות החומרה שלמדנו עליהן לא מסוגלות לטפל. מאחר ובהקדמת ה branch לשלב 2 מתבצעת השוואה של ערכי \$rs \$rt בשלב 2 ביציאה ממקבץ האוגרים. בקוד המתואר בשאלה כאשר הפקודה sub הנמצאת בשלב 3 מתבצע חישוב ב ALU של ערך חדש לאוגר 9, באותה פעימת שעון בה הפקודה beq הנמצאת בשלב 2 משווה את הערך של אוגר 9 שנקרא ממקבץ האוגרים. כלומר הבדיקה של התנאי לקפיצה של ה beq הוא לא לערך המעודכן של אוגר 9. יחידת העברת ערכים (forwarding unit) אותה למדנו מטפלת רק בעדכון ערכים לפקודות הנמצאות בשלב 3 ולא בשלב 2.

מהו הפתרון האפשרי לסיכון נתונים זה ? (יש לתת קווים כלליים לפתרון).

פתרון אפשרי הוא בניית יחידת העברה קדימה ספציפית למקרה זה של beq המבצעת את השוואת ערכי אוגרי המקור בשלב 2 מול פקודה המחשבת ב ALU ערך חדש לאחד מאוגרי המקור. הבעייתיות של פתרון זה שאילוץ הזמן מגדיר שרק לאחר פעולת ה ALU יועבר הערך דרך מרבב המתאים למשווה. בהנחה שהמרבבים בכניסה ל ALU +זמן חישוב ALU+מרבב של העברה קדימה + משווה קצרים מזמן גישה לזיכרון (נתונים או פקודות בשלבים 1 ו 4) אז הפתרון לא מאלץ להאריך את השעון של המעבד

במידה והפתרון היה מאלץ להאריך את השעון של המעבד אז יכול להיות שהיה עדיף לפתור סיכון זה בהשהיה (ע"י הוספת HDU ייעודי לבעיה זו) והעברה קדימה משלב 4 ל 2 (שכמובן מחייב חומרה נוספת)

הערה : יש לציין שאנחנו מתייחסים ספציפית לסיכון המתואר בשאלה של פקודות צמודות וקיימים סיכונים נתונים חדשים עקב הקדמת ה branch לשלב 2 שלא טופלו.

על סמך הפתרון שהצעת מהו הערך המתקבל בסוף פעימת השעון השישית על הקו המסומן ב P3 בתרשים בסעיף הקודם (13.2) ? יש להסביר את התשובה.

למי שפתר בהנחה שמנגנון העברה קדימה עובד מבחינת תזמון אז $9 = \$12 - \$9 = 0x4B0 - 0x180 = 0x330$ נשים לב שה add וה sub קיזזו את ערך אוגר 9. כלומר גם מי שהייתה לו טעות נגררת על ערך אוגר 9 תשובתו אמורה להיות נכונה.

למי שפתר את סיכון הנתונים עם השהיה (ע"י הוספת HDU ייעודי לבעיה זו) במהלך פעימת השעון השישית אז הערך של אוגר 9 עוד לא הספיק להתעדכן כלומר $0x00000180$ (הסבר חישוב בקווים O1 ו P3 בסעיפים הקודמים בשאלה).

שאלה 14 (16נקודות)

14.1 (4 נק')

הפרוצדורה תדפיס:

2

**

הסבר:

בחלק הראשון של הפרוצדורה הלולאה loop סופרת באוגר \$t2 את מספר סיביות 1 בערך שבאוגר \$a1. בדוגמא שלנו יש 2 סיביות 1 (5=101) ערך זה מודפס למסך בעזרת 1 syscall. לאחר מכן נבדק האם ערך זה הוא זוגי בעזרת הפקודה andi \$t0,\$t2,1 היות וערך זה זוגי מודפסת ירידת שורה ומודפסים כלולאת astrix כוכביות כמספר ב \$t2 כלומר כמספר סיביות ה 1 באוגר \$a1.

14.2 (4 נק')

1

הסבר:

בחלק הראשון של הפרוצדורה הלולאה loop סופרת באוגר \$t2 את מספר סיביות 1 בערך שבאוגר \$a1. בדוגמא שלנו יש 1 סיביות 1 (1024=1000000000) ערך זה מודפס למסך בעזרת 1 syscall. לאחר מכן נבדק האם ערך זה הוא זוגי בעזרת הפקודה andi \$t0,\$t2,1 היות וערך זה אינו זוגי מבוצע חיסור של ערך זה מ 32 כלומר מחושבים מספר סיביות האפס באוגר \$a1. מודפסת ירידת שורה ומודפסים כלולאת astrix כוכביות כמספר 32-\$t2 כלומר כמספר סיביות ה 0 באוגר \$a1. 31=32-1 יודפסו 31 כוכביות.

14.3 (4 נק')

31

*

הסבר:

בחלק הראשון של הפרוצדורה הלולאה loop סופרת באוגר \$t2 את מספר סיביות 1 בערך שבאוגר \$a1. בדוגמא שלנו יש 31 סיביות 1 (-2=11111111111111111111111111111110) ערך זה מודפס למסך בעזרת 1 syscall. לאחר מכן נבדק האם ערך זה הוא זוגי בעזרת הפקודה andi \$t0,\$t2,1 היות וערך זה אינו זוגי מבוצע חיסור של ערך זה מ 32 כלומר מחושבים מספר סיביות האפס באוגר \$a1. מודפסת ירידת שורה ומודפסים כלולאת astrix כוכביות כמספר 32-\$t2 כלומר כמספר סיביות ה 0 באוגר \$a1. 1=32-31 תודפס כוכבית אחת.

14.4 (4 נק')

במקרה קצה של הערך אפס באוגר \$a1 נקבל גם באוגר \$t2 את הערך 0 לאחר ביצוע ה andi יחזר 0 ב \$t0 ותתבצע קפיצה לתווית print עם הערך אפס באוגר \$t2 תתבצע ירידת שורה ואז תתבצע הדפסה של מספר עצום של כוכביות (בפועל כ 4 מיליארד או עד לנפילת התכנית) היות ומתבצעת כניסה ללולאת for (astrix) עם הערך 0 במונה (\$t2) כאשר הפעולה הראשונה המתבצעת זה חיסור 1 מערך המונה שהוא אפס).