



תכן לוגי תרגיל יבש #3

הגשה עד: 3.1.19 בשעה 23:55.

המתרגל האחראי על התרגיל: אבירם אימבר
שאלותיכם במייל, (כולל עניינים מנהלתיים) יופנו רק אליו.

**כתבו בתיבת subject של
המייל: "יבש תכן לוגי".**

שאלות בעל-פה ייענו על
ידי כל מתרגל.



הוראות הגשה:

- לכל שאלה יש לרשום את התשובה במקום המיועד לכך.
- מידת הפירוט הרצויה בתשובות היא 2-3 משפטים לסעיף.
- יש לענות ע"ג טופס התרגיל.
- יש לצרף דף שער הכולל ברקודים של ת"ז המגישים.
- תמונת ברקוד אפשר לייצר לדוגמה בקישור :
<https://barcode.tec-it.com/en/Code128>
- את הפתרון לתרגיל מגישים בתא הקורס.
- על כל יום איחור או חלק ממנו, שאינו בתאום עם המתרגל האחראי על התרגיל, יורדו 5 נקודות.
- הגשות באיחור יש להניח בתא של המתרגל האחראי (נועם) בקומה 5 בציון שעת ותאריך ההגשה ובנוסף לשלוח מייל אל אחראי התרגיל כאשר הגשתם לתא, אחרת מועד האיסוף ייחשב כמועד ההגשה.
- ההגשה בזוגות בלבד.

שאלה 1 – MIPS ISA

א. תרגמו את הקוד הבא משפת C לשפת MIPS assembly:

```
for(int i = 0; i < 100; i++) {  
    arr[i] = i;  
}
```

השתמשו ברגיסטרים הבאים:

\$t0 – i עבור

\$t1 – מכליל בתחילת הריצה את כתובת תחילת המערך, ניתן לשנותו

\$t2 – עבור 100

ב. תרגמו את הקוד הבא משפת C לשפת MIPS assembly:

```
while(flag) {  
    ...  
}
```

Flag הינו ביטוי בוליאני, אין צורך לתרגם את מה שקורה בתוך הלולאה (סמנו את תוכן הלולאה ב"...").
השתמשו ברגיסטרים הבאים:

\$t0 – flag עבור

ג. נתון הקוד הבא בשפת MIPS assembly:

```
1.      beq $s1, $s2, label1
2.      add $t0, $s2, $s3
3.      add $t0, $t0, $t0
4.      lw $s1, 16($t0)
5.      j label2
6. label1: sub $t0, $s2, $s4
7.      add $t0, $t0, $t0
8.      add $t0, $t0, $s3
9.      lw $s1, 32($t0)
10. label2: ...
```

נתון שפקודה מספר 1 תיטען בכתובת 0xC0000010.
מה הוא הקידוד הבינרי של פקודה מספר 1 ופקודה מספר 5?
הראו בתשובה שלכם חלוקה של קידוד הפקודות, וציינו את השמות של החלקים השונים.

שאלה 2 – MIPS ISA

אלברט, תלמיד בקורס תכן לוגי, תכנן גרסת 64 ביט למחשב ה-Single-Cycle MIPS המוכר והאהוב. במחשב זה יש 64 רגיסטרים כל אחד מהן הוא בן 64 ביט. כל פקודה בגודל 64 ביט, "מילה" (שנקראת נכתבת מהזיכרון ע"י lw/sw) היא 64 ביט. הפקודות הן בדיוק אותן פקודות שהיו במעבד גרסת 32 ביט. הערה: עדיין לכל בית בזיכרון יש כתובת משלו.

א. מה גודל הזיכרון המקסימלי (בבתים) בו ניתן לתמוך במחשב?
הגודל המקסימלי הוא _____ בתים.
הסבר:

ב. הקף בעיגול את השינויים שצריך לשנות במסלול הנתונים של המחשב מבין הרשימה הבאה כדי לתמוך בשינוי (יכולה להיות יותר מתשובה אחת נכונה, וייתכנו שינויים נחוצים נוספים שאינם מופיעים ברשימה):

- יש להרחיב את ה-PC לאוגר ברוחב 64 ביט.
- יש להרחיב את מספר ביטי ה-opcode ל-7.
- בכל פעולה לקדם את ה-PC ב-8 במקום ב-4.
- בכל פעולה לקדם את ה-PC ב-16 במקום ב-4.
- לשנות את הרכיב שמבצע shift left 2 לרכיב שמבצע shift left 3.
- לשנות את הרכיב שמבצע shift left 2 לרכיב שמבצע shift left 4.

ג. מה טווח הקפיצה האפשרי בפקודות במחשב זה?

1. בעזרת פקודת קפיצה מותנת (branch):
ניתן לקפוץ סה"כ _____ פקודות, אחורה ו- _____ קדימה, כל אלו ביחס ל- _____.
הסבר קצר:

2. בעזרת קפיצה לא מותנת (J):
ניתן לקפוץ ל- _____ פקודות, אלו כל הפקודות ש- _____ סיביות ה-MSB של כתובתן זהות ל- _____ סיביות ה-MSB של ה-PC + _____, ו- _____ סיביות ה-LSB של כתובת הן _____.
הסבר קצר:

שאלה 3 – Single-Cycle MIPS

הניחו שמעבד Single-Cycle MIPS מבצע את קוד האסמבלי הבא: (המספרים כתובים בבסיס עשרוני)

```
lw $t0, 64($zero)
and $t0, $t0, $t2
add $t0, $t0, $gp
lw $t1, -20($t0)
```

הניחו ש-\$t2 מכיל את הערך ההקסדצימלי 80 0000, ו-\$gp מכיל את הערך ההקסדצימלי 1000 8000.

א. הערך שיימצא ברגיסטר \$t0 בסוף החישוב תלוי במילה המצויה בכתובת מסוימת בזיכרון. מהי הכתובת? כתבו את תשובתכם בבסיס **הקסדצימלי**.

(שימו לב: יש לכתוב את כתובת המילה בזיכרון. בפרט, ודאו שתשובתכם מתחלקת ב-4 וכתובה בבסיס **הקסדצימלי**.)

ב. הערך שיימצא ברגיסטר \$t0 בסוף החישוב תלוי אך ורק בביט בודד של המילה המצויה בכתובת שמצאתם בסעיף א'. מהו מספר הביט הזה?

(כאשר המספר 0 מייצג את ה-LSB, והמספר 31 מייצג את ה-MSB.)

ג. מהי הכתובת בזיכרון שממנה נקרא את הערך של \$t1 אם הביט הבודד שמצאתם בסעיף ב' שווה ל-0? כתבו את תשובתכם בבסיס **הקסדצימלי**.

ד. מהי הכתובת בזיכרון שממנה נקרא את הערך של \$t1 אם הביט הבודד שמצאתם בסעיף ב' שווה ל-1? כתבו את תשובתכם בבסיס **הקסדצימלי**.

כעת נניח את שתי ההנחות הבאות על המעבד שלנו: (הן מתקיימות באופן חלקי במספר מעבדים מודרניים)

1. נניח שאסור לתוכנה שלנו לקרוא את המידע המצוי בזיכרון בכתובת שמצאתם בסעיף א', כי זו כתובת ששמורה למערכת ההפעלה.

נניח שהמעבד מגלה שבוצעה קריאה אסורה מהזיכרון **50 מחזורי שעות** לאחר סיום ביצוע הקריאה, מציג הודעת שגיאה, ומבטל את כל הפקודות שביצעה התוכנה החל מהקריאה האסורה ועד עכשיו.

2. נניח שלאחר קריאת מילה מסוימת מהזיכרון, זמן הגישה לזיכרון לקריאת מילה זו פעם נוספת יהיה **קצר יותר**: קריאת מילה מהזיכרון בפעם הראשונה תיקח **5 מחזורי שעות**, וקריאת מילה מהזיכרון בפעם השנייה (ומעלה) תיקח רק **מחזור שעות אחד**.

זה נכון **אפילו** אם הפקודה המקורית לגישה לזיכרון בוטלה בגלל האיסור שבהנחה 1.

ה. נניח שלאחר ביצוע התוכנה, הודעת השגיאה, וביטול כל הפקודות שבוצעו, המעבד מבצע את הפקודה:

```
lw $t3, -20($gp)
```

מהו משך הקריאה מהזיכרון בפקודה זו אם הביט הבודד שמצאתם בסעיף ב' שווה ל-0? _____ מחזורי שעות

מהו משך הקריאה מהזיכרון בפקודה זו אם הביט הבודד שמצאתם בסעיף ב' שווה ל-1? _____ מחזורי שעות

ההבדל בין שתי התשובות שמצאתם בסעיף ה' הוא הרעיון הבסיסי של פרצת האבטחה **Meltdown**.

שאלה 4 – Single-Cycle MIPS

א. נוסף למעבד Single-Cycle Mips פקודות חדשות שמבצעות load באופן הבא:

LAdd Rd, Rs, Rt	$\ll Rd = Mem[Rs+Rt]$
LSub Rd, Rs, Rt	$\ll Rd = Mem[Rs-Rt]$
LMul Rd, Rs, Rt	$\ll Rd = Mem[Rs*Rt]$
LDiv Rd, Rs, Rt	$\ll Rd = Mem[Rs/Rt]$

ניתן להניח שהחלוקה היא חלוקה בשלמים, וניתן להניח שבכל המקרים תוצאת החישוב המתקבלת היא בהכרח כתובת חוקית.

סמנו את התשובה הנכונה הראשונה מבין התשובות הבאות לגבי מימוש הפקודות הנ"ל כאשר כל פקודה ממומשת **כפקודה יחידה** (ולא כפסאודו פקודה):

- (1) ניתן לממש באמצעות שינויים בבקר בלבד.
- (2) ניתן לממש באמצעות שינויים בבקר והוספת בורר יחיד.
- (3) ניתן לממש באמצעות שינויים בבקר והוספת כמה בוררים.
- (4) ניתן לממש באמצעות על השינויים של (3) ובנוסף שינוי מבנה הזיכרון או קובץ הרגיסטרים.
- (5) לא ניתן לממש את הפקודה כפקודה יחידה.

נימוק:

ב. נוסף שיטת מיעון חדשה המסומנת ב-(Rn). הכתובת האפקטיבית של שיטת המיעון היא $Mem[Rn]$, כלומר Rn משמש כמצביע לזיכרון. שיטת מיעון זו מותרת אך ורק בפקודות ALU (add, sub וכו'). מותר להשתמש בשיטה זו עבור אופרנד היעד, ועבור אופרנד מקור אחד לכל היותר. למשל, הפקודות הבאות הן חוקיות:

Add \$s1, \$s2, (\$s3)	$\ll \$s1 = \$s2 + Mem[\$s3]$
Add \$s1, (\$s2), \$s3	$\ll \$s1 = Mem[\$s2] + \$s3$
Add (\$s1), \$s2, \$s3	$\ll Mem[\$s1] = \$s2 + \$s3$
Add (\$s1), (\$s2), \$s3	$\ll Mem[\$s1] = Mem[\$s2] + \$s3$
Add (\$s1), \$s2, (\$s3)	$\ll Mem[\$s1] = \$s2 + Mem[\$s3]$

אבל הפקודה Add \$s1, (\$s2), (\$s3) והפקודה Add (\$s1), (\$s2), (\$s3) אינן חוקיות. נרצה לממש את כל הפקודות המתוארות ב-Single-Cycle Mips (ניתן להניח שלכל פקודה יש opcode פנוי כלשהו).

סמנו את התשובה הנכונה הראשונה מבין התשובות הבאות לגבי מימוש הפקודות הנ"ל כאשר כל פקודה ממומשת **כפקודה יחידה** (ולא כפסאודו פקודה):

- (1) המימוש אפשרי באמצעות שינויים בבקר בלבד.
- (2) המימוש אפשרי באמצעות שינויים בבקר ובחוטמים במסלול הנתונים בלבד.
- (3) המימוש אפשרי באמצעות כל השינויים לעיל + הוספת/הרחבת בוררים במסלול הנתונים.
- (4) המימוש אפשרי באמצעות כל השינויים לעיל + הוספת MEM למסלול הנתונים.
- (5) המימוש אפשרי באמצעות כל השינויים לעיל + הוספת ALU למסלול הנתונים.
- (6) המימוש כפקודה אחת אינו אפשרי ב-Single-Cycle Mips בעזרת השינויים מתשובות 1-5.

נימוק:

ג. גורי ניסה לממש את שיטת המיעון מסעיף ב' בתור פקודה יחידה במעבד, ונתקל בכמה קשיים. אסי הציע לממש אותה בתור **פסאודו-פקודה**. גורי לא יודע באיזו דרך מימוש לבחור, מה שגורם לו להיות מתוסכל, ולכן הוא פונה אליכם לעזרה. ממשו את $s1$, $s2$, $s3$ Add בתור **פסאודו-פקודה** שמורכבת משלוש פקודות מכונה. הפקודות יכולות לשנות כל רגיסטר שאינו $s2$, $s3$.



שאלה 5 – Single-Cycle MIPS

בשאלה זו שלושה סעיפים בלתי-תלויים. כולם עוסקים בפקודת אסמבלי חדשה, lti , שתוגדר באופן הבא:
 $lti Rj, Rk, constant$

הסמנטיקה של הפקודה היא:

$Rj \leftarrow constant$

$Rk \leftarrow constant + 4$

הניחו שמתקיים $j \neq k$. הניחו שהקבוע $constant$ הוא מספר (חיובי, שלילי, או אפס) ברוחב 16 ביט.

א. מעוניינים לממש את lti על Single-Cycle MIPS כפקודת מכונה (אחת).
סמנו את התשובה הנכונה הראשונה מבין התשובות הבאות:

- (1) המימוש אפשרי באמצעות שינויים בבקר בלבד.
- (2) המימוש אפשרי באמצעות שינויים בבקר ובחוטים במסלול הנתונים בלבד.
- (3) המימוש אפשרי באמצעות כל השינויים לעיל + הוספת/הרחבת בוררים במסלול הנתונים.
- (4) המימוש אפשרי באמצעות כל השינויים לעיל + שינוי מבנה הזיכרון במסלול הנתונים.
- (5) המימוש אפשרי באמצעות כל השינויים לעיל + שינוי מבנה ה-ALU במסלול הנתונים.
- (6) המימוש אפשרי באמצעות כל השינויים לעיל + שינוי מבנה קובץ הרגיסטרים במסלול הנתונים.
- (7) המימוש אפשרי באמצעות כל השינויים לעיל + הוספת ALU למסלול הנתונים.
- (8) המימוש כפקודת מכונה אחת אינו אפשרי ב-Single-Cycle MIPS.

ב. מעוניינים לממש את lti על Multi-Cycle MIPS כפקודת מכונה (אחת).
סמנו את התשובה הנכונה הראשונה מבין התשובות הבאות:

- (1) המימוש אפשרי באמצעות שינויים בבקר בלבד.
- (2) המימוש אפשרי באמצעות שינויים בבקר ובחוטים במסלול הנתונים בלבד.
- (3) המימוש אפשרי באמצעות כל השינויים לעיל + הוספת/הרחבת בוררים במסלול הנתונים.
- (4) המימוש אפשרי באמצעות כל השינויים לעיל + שינוי מבנה הזיכרון במסלול הנתונים.
- (5) המימוש אפשרי באמצעות כל השינויים לעיל + שינוי מבנה ה-ALU במסלול הנתונים.
- (6) המימוש אפשרי באמצעות כל השינויים לעיל + שינוי מבנה קובץ הרגיסטרים במסלול הנתונים.
- (7) המימוש אפשרי באמצעות כל השינויים לעיל + הוספת ALU למסלול הנתונים.
- (8) המימוש כפקודת מכונה אחת אינו אפשרי ב-Multi-Cycle MIPS.

ג. ממשו את lti כפסאודו-פקודה (עם הפרמטרים $Rj, Rk, constant$) באמצעות רצף של **שתי פקודות מכונה סטנדרטיות** שנלמדו בתרגול 7 (MIPS ISA):

שאלה 6 – Single vs. Multi

א. בסעיף זה נבדוק את היעילות של מימוש פקודות מכונה ופסאודו-פקודות על מימושים שונים של מעבד ה-MIPS.

מחזורי השעון של Single-Cycle MIPS ושל Multi-Cycle MIPS הם באורכים שונים זה מזה. כדי לדעת מהו משך הזמן בפועל לביצוע כל פקודה, נרצה למצוא את משך מחזור השעון. נניח שההשהייה של כל אחת מיחידות החישוב ה"ארוכות" שווה ל-T. כלומר, נדרש זמן T כדי לבצע כל אחת מהפעולות הבאות: (1) קריאה/כתיבה מקובץ הרגיסטרים, (2) קריאה/כתיבה מהזיכרון (מכל אחד מסוגי הזיכרונות: פקודות, נתונים, או משולב), ו-(3) חישוב ב-ALU.

כתבו בטבלה הבאה את משך הביצוע, הנמדד בכפולות של T, של פקודות המכונה ושל הפסאודו-פקודות הבאות בכל אחד מארבעת המימושים המופיעים בטבלה:

פקודת המכונה / הפסאודו-פקודה	משך ביצוע הפקודה במימוש Single-Cycle MIPS	משך ביצוע הפקודה במימוש Multi-Cycle MIPS
add \$t1, \$t0, \$s1		
beq \$s0, \$s1, label		
sw \$t1, -3(\$s3)		
li \$t5, 8398		
lw \$s5, 50(\$t0)		
addi \$t2, \$t1, -16		
li \$s0, 1051000		

הניחו שביצוע פקודת המכונה lui (לא li!) על Multi-Cycle MIPS דורש 4 מחזורי שעון. **שימו לב:**

- כאשר מבקשים לחשב את משך הביצוע של פסאודו-פקודות, עליכם לתרגם אותן למספר המינימלי האפשרי של פקודות מכונה ולחשב את משך הביצוע הכולל של פקודות המכונה.
- ישנן פסאודו-פקודות שניתנות לביצוע ללא פיצול לשתי פקודות מכונה (ומעלה)! במקרים כאלה עליכם לתרגם אותן לפקודת מכונה אחת בלבד.
- כל המספרים בפקודות האסמבלי שבטבלה נכתבים בבסיס עשרוני.
- פקודות האסמבלי שבטבלה אינן תלויות זו בזו.

ב. במחשב Multicycle MIPS שנלמד בכיתה, חלה טעות במימוש הבקר. בעקבות טעות זאת אותות הבקרה MemRead ו-MemWrite הוחלפו ביניהם. נגדיר "ריצה תקינה" בתור כזאת שמבצעת את הפעולות שנכתבו בתוכנית בתכנית המקורית. בחר בטענה הנכונה:

- (1) כל תכנית שאינה כוללת lw/sw תרוץ באופן תקין, ללא כל שינוי.
- (2) ניתן לשנות כל תכנית שכוללת lw (אך לא sw) כך שתרוץ באופן תקין.
- (3) ניתן לשנות כל תכנית שכוללת sw (אך לא lw) כך שתרוץ באופן תקין.
- (4) ניתן לשנות כל תכנית כך שתרוץ באופן תקין.
- (5) אף תכנית לא ניתנת לשינוי כך שתרוץ באופן תקין.

ג. בהמשך לסעיף קודם, במסגרת תיקון לבקר, הצליח מהנדס להחזיר את אות הבקרה MemWrite לערכו התקין, אבל MemRead עתה מקבל את ערכו של הסיגנל PCLoad. בחר בטענה הנכונה:

- (1) כל תכנית שאינה כוללת j תרוץ באופן תקין.
- (2) כל תכנית שאינה כוללת sw תרוץ באופן תקין.
- (3) כל תכנית שאינה כוללת lw תרוץ באופן תקין.
- (4) כל תכנית תרוץ באופן תקין.
- (5) אף תכנית לא ניתנת לשינוי כך שתרוץ באופן תקין.