



מערכות ספרתיות ומבנה המחשב (044252)

סמסטר אביב תשע"ט

בחינה סופית – מועד א 4 ביולי 2019

טור 1

--	--	--	--	--	--	--	--	--	--

מספר סטודנט

משך המבחן: 3 שעות (180 דקות). **תכננו את זמנכם היטב.**

חומר עזר: אין להשתמש בכל חומר עזר בכתב, מודפס או אלקטרוני, פרט לדפי העזר שיחולקו במהלך הבחינה.

הנחיות והוראות:

- הבחינה כתובה על גבי 18 עמודים כולל עמוד זה (בדקו בתחילת הבחינה שלא חסרים לכם עמודים).
- בתחילת הבחינה תקבלו חוברת בחינה, מחברת טיוטה, דפי עזר וטופס תשובות ממוחשב. בסיום הבחינה, החזירו את חוברת הבחינה וטופס התשובות הממוחשב בלבד.
- יש לענות על כל השאלות בגוף המבחן.
- אין לתלוש או להפריד דפים מחוברת הבחינה, ממחברות הטיוטה ומדפי העזר.
- רשמו את מספר הסטודנט שלכם על חוברת הבחינה (בראש עמוד זה), על דפי העזר, ועל כל מחברות הטיוטה.
- לא מורדות נקודות (אין "קנס") בגין תשובה שגויה. לכן, כדאי לסמן תשובה כלשהי לכל שאלה.
- ציון השאלות רב הברירה ייקבע על סמך סריקה ממוחשבת של טופס התשובות בלבד. **לא לשכוח לסמן בטופס התשובות הממוחשב את מספר הטור שלכם (מופיע בראש עמוד זה).**
- אסור שימוש בכל חומר חיצוני. אסורה העברת חומר כלשהו בין הנבחנים, ואסורה כל תקשורת עם אנשים אחרים או כל מקור מידע. האיסור חל על כל צורות התקשורת – מילולית, חזותית, כתובה, אלקטרונית, אלחוטית, טלפתית, או אחרת. בפרט, אין להחזיק בטלפון סלולארי.

בהצלחה!



שאלה 1 (5 נקודות)

נתון קוד ה-System Verilog הבא:

```
module test(  
  input logic clk,  
  input logic [3:0] a,  
  output logic [3:0] z  
);  
  always_comb begin  
    z = (a << 2) + 1;  
  end  
endmodule
```

איזה קטע קוד ייצור חומרה זהה לחומרה שתיווצר כתוצאה מהקוד הנ"ל?

א-

```
module test(  
  input logic clk,  
  input logic [3:0] a,  
  output logic [3:0] z  
);  
  always_ff @(posedge clk) begin  
    z <= (a * 4) + 1;  
  end  
endmodule
```

ב-

```
module test(  
  input logic [3:0] clk,  
  input logic [3:0] a,  
  output logic [3:0] z  
);  
  assign z = {a[1:0], {2{1'b1}}};  
endmodule
```

ג-

```
module test(  
  input logic clk,  
  input logic [3:0] a,  
  output logic [3:0] z  
);  
  assign z[3] = a[1];  
  assign z[2] = a[0];  
  assign z[1:0] = 2'b01;  
endmodule
```

ד- תשובות א' ו-ב' נכונות.
ה- תשובות ב' ו-ג' נכונות.



שאלה 2 (5 נקודות)

שני סטודנטים להנדסת חשמל מעוניינים לתקשר ביניהם באמצעות מילים השייכות לקוד 2 out of 4 עליו למדו בקורס. בקוד זה, המילים הקיימות הן מילים בעלות 4 ביטים כך שמספר הסיביות במילה שערך הוא '1' הינו 2 בדיוק. למשל, המילה: 0101, נמצאת בקוד. שכן, 0101 הינה מילה בעלת 4 ביטים, ושני ביטים הם '1'.

סטודנט א' שולח מילים לסטודנט ב' מתוך הקוד. סטודנט ב' בודק האם המילים שקיבל אכן שייכות לקוד. סטודנט ב' יודע שקו התקשורת בין הסטודנטים רועש מעט באופן שבו כל מילה שקיבל עלולה להכיל **עד היפוך סיבית אחת לכל היותר**. לכן, סטודנט ב' מעוניין לממש מערכת צירופית, שמקבלת מילה אחת (4 ביטים) ובודקת האם היא שייכת לקוד או לא. במידה וכן, המערכת תוציא '1', אחרת '0'.

לסטודנט נתונים שערי AND, OR, XOR, בעלי **שתי כניסות**, ושערי NOT.

איך ניתן לממש את המערכת הצירופית הנדרשת?

- א- ניתן לממש את הפונקציה בעזרת שערי XOR בלבד.
- ב- ניתן לממש את הפונקציה בעזרת שערי OR בלבד.
- ג- ניתן לממש את הפונקציה בעזרת שערי AND בלבד.
- ד- כדי לממש את הפונקציה צריך גם שערי AND וגם- OR, אך אין צורך בשערים נוספים.
- ה- תשובות א' – ד' אינן נכונות מכיוון שהשערים הנתונים בהן אינם מהווים מערכת פעולות שלמה.



שאלה 3 (5 נקודות)

בהמשך לשאלה הקודמת, סטודנט א' שולח לסטודנט ב' מילים על הקו באופן סדרתי. סטודנט ב' מעוניין לבדוק האם קו השידור בין הסטודנטים אמין. לשם כך הוא משתמש ביציאת המערכת מהסעיף הקודם שמדווחת האם המילה הנוכחית נמצאת בקוד. אם הקלט הינו '0' אז המילה איננה בקוד ואם '1' אז המילה אכן בקוד. אם מתוך 3 המילים האחרונות שהתקבלו, לפחות 2 לא בקוד, הקו מוגדר כלא תקין.

סטודנט ב' מעוניין לבנות מכונת מצבים מסוג **MOORE** שמקבלת כניסה סיבית המסמלת האם המילה הנוכחית תקינה או לא. במידה והקו נמצא כלא תקין, המכונה תוציא '1', ללא תלות בקלט עתידי.

למשל:

Cycle	1	2	3	4	5	6	7	8	9	10	11
IN	1	1	0	1	1	0	1	0	1	0	1
OUT	X	0	0	0	0	0	0	0	1	1	1

במהלך מחזורים מספר 6 ו-8 התקבל בכניסה '0'. כלומר במשך שלושה מחזורים קיבלנו שתי מילים שאינן בקוד, ולכן החל מהמחזור הבא, המערכת תוציא את הפלט '1', מבלי להתחשב בכניסות.

יש להניח כי המערכת מתחילה את פעולתה במצב כאילו קיבלה כמות גדולה של מילים חוקיות.

כמה מצבים יהיו במכונת **המצבים המצומצמת ביותר** המממשת את המערכת הנדרשת?

- א- 4 מצבים או פחות.
- ב- 5 מצבים.
- ג- 6 מצבים.
- ד- 7-8 מצבים.
- ה- 9 ומעלה



שאלה 4 (5 נקודות)

יהי X מספר בינארי ברוחב $N > 0$ ביטים, ויהי ψ_X קידוד גריי שלו.
נתונות הטענות הבאות:

$$(1) \quad X \oplus \psi_X \neq 0, \text{ לכל } X$$

$$(2) \quad X \leq Y \text{ אם"ם } \psi_X \leq \psi_Y$$

$$(3) \quad X \oplus \psi_X = Y \oplus \psi_Y \text{ אם"ם } X = Y$$

הערות:

- $\psi_X \leq \psi_Y$ משמעותו שהערך המיוצג בבסיס בינארי (ללא סימן) כ- ψ_X קטן או שווה

לערך המיוצג בבסיס בינארי (ללא סימן) כ- ψ_Y .

למשל, עבור $\psi_Y = 1000$, $\psi_X = 0111$ מתקיים ש- $\psi_X < \psi_Y$ מכיוון ש- $7 \leq 8$.

כנ"ל לגבי $X \leq Y$.

- $Z = X \oplus Y$ מוגדרת להיות פעולת bitwise XOR בין כל שני ביטים מ- X ומ- Y

באופן הבא:

$$Z_i = X_i \oplus Y_i, \quad \forall i: 0 \leq i < N$$

מבין התשובות הבאות, בחרו את התשובה הנכונה:

א- רק טענה 1 נכונה

ב- רק טענה 2 נכונה

ג- רק טענה 3 נכונה

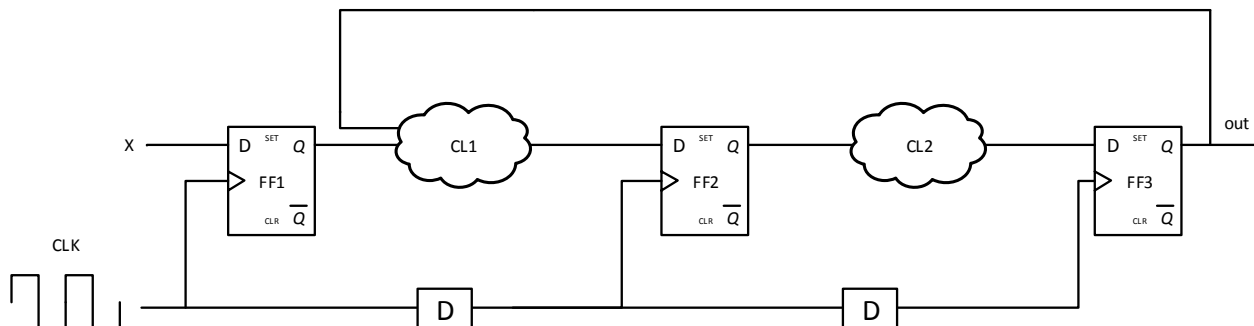
ד- מבין הטענות 1-3 יש רק 2 טענות נכונות

ה- כל הטענות 1-3 לא נכונות



שאלה 5 (5 נקודות)

נתון המעגל הבא:



נתון כי הדלגלים דוגמים בעליית שעון. בנוסף, נתון כי כניסת המערכת X מתוזמנת בהתאם לתנאי ה-*hold* וה-*setup* של הדלגלג FF1.

להלן זמני ההשהיה:

$$\begin{aligned} t_{pd}(CL1) &= 5 \text{ ns} \\ t_{pd}(CL2) &= 6 \text{ ns} \\ t(D) &= 2 \text{ ns} \\ t_{pcQ}(FF) &= 7 \text{ ns} \\ t_{setup}(FF) &= 3 \text{ ns} \end{aligned}$$

בשאלה זו ניתן להניח שתנאי *hold* מתקיים (לא צריך לבדוק אותו).

מבין התשובות הבאות, מהו אורך מחזור השעון **המינימלי** שיאפשר למעגל לפעול בצורה תקינה?

- א- 14 ns
- ב- 15 ns
- ג- 16 ns
- ד- 17 ns
- ה- גדול מ- 17 ns



שאלה 6 (5 נקודות)

מעוניינים לממש מערכת צירופית המכפילה מספר בינארי X בן n סיביות פי 5, אך ממומשת בעזרת רכיבי FullAdder וקבועים ('0' ו-'1') **בלבד**.

הערות:

- 1- המספר X מיוצג בשיטת ה-unsigned.
- 2- שימו לב שיתכן שצריך יותר מ- n ביטים כדי לייצג את התוצאה.

מבין התשובות הבאות, מהי הכמות **המינימלית** של רכיבי FullAdder בה אפשר לממש את המערכת?

- א- $2n$
- ב- $4n$
- ג- n^2
- ד- $5n$
- ה- n



שאלה 7 (5 נקודות)

נתון קטע הקוד הבא, המתחיל לרוץ מ-main (כתובת 0x0001 0000).

```

[0x0001 0000] main:   addi sp, sp, -4
[0x0001 0004]         addi a0, x0, 3
[0x0001 0008]         addi a1, x0, 2
[0x0001 000c]         sw ra, 0(sp)
[0x0001 0010]         jal ra, func
[0x0001 0014]         lw ra, 0(sp) ←
[0x0001 0018]         addi sp, sp, 4
[0x0001 001c]         jalr x0, ra

[0x0001 0100] func:   addi sp, sp, -4
[0x0001 0104]         beq a1, x0, done
[0x0001 0108]         addi a1, a1, -1
[0x0001 010c]         sw ra, 0(sp)
[0x0001 0110]         jal ra, func
[0x0001 0114]         add a0, a0, a0
[0x0001 0118]         lw ra, 0(sp)
[0x0001 011c] done:   addi sp, sp, 4
[0x0001 0120]         jalr x0, ra

```

מה יהיה הערך ברגיסטר a0 מיד לפני שמריצים את הפקודה בכתובת [0x0001 0014] (הפקודה המסומנת בחץ)?
בחרו את התשובה המתאימה:

א- התוכנית לעולם לא תגיע לפקודה בכתובת [0x0001 0014]

ב- 6

ג- 9

ד- 12

ה- 15



שאלה 8 (5 נקודות)

בפרוטוקול תקשורת חדש שנקרא XUART, מועברים X סיביות מידע בכל שידור, כאשר X הינו מספר ידוע וקבוע (אינו משתנה משידור לשידור).
שאר המאפיינים של פרוטוקול זה זהים ל- UART עליו למדנו.

נתונים:

$$T_{cycle}(Tx) = 10nsec$$

$$T_{bit} = 50nsec$$

$$T_{cycle}(Rx) = 13nsec$$

בנוסף נתון שהמקלט מזהה את ה- Start Bit באופן מיידי.

יש להניח שלאחר שהמקלט קבע מהו ה- N_R שאיתו יעבוד, הוא קבוע עד לסוף קליטת השידור ולא ניתן לשינוי.

מבין התשובות הבאות, מה המספר **המקסימלי** של **סיביות מידע** שניתן לשדר ולקלוט בלי שגיאה?

הערה: סיביות Start-bit ו- Stop-bit אינן סיביות מידע.

א- 8

ב- 9

ג- 10

ד- 13

ה- 14



שאלות 9 (5 נקודות)

נתונה טבלת המעברים הבאה:

PS	NS(x=0), z	NS(x=1), z
A	A,0	C,0
B	E,1	A,0
C	B,0	E,1
D	B,1	C,0
E	F,1	D,0
F	D,0	F,1

מבין הסדרות הבאות, לאיזו סדרת קלט היציאה תהיה '1' במהלך המחזור החמישי (האחרון), כאשר נתון שמתחילים ממצב A והסיביות נקלטות משמאל לימין בכל אחת מהסדרות?

א-

cycle	1	2	3	4	5
x	1	0	1	0	0

ב-

cycle	1	2	3	4	5
x	0	1	1	1	1

ג-

cycle	1	2	3	4	5
x	1	1	0	1	0

ד-

cycle	1	2	3	4	5
x	1	1	1	1	0

ה-

cycle	1	2	3	4	5
x	0	0	1	0	0



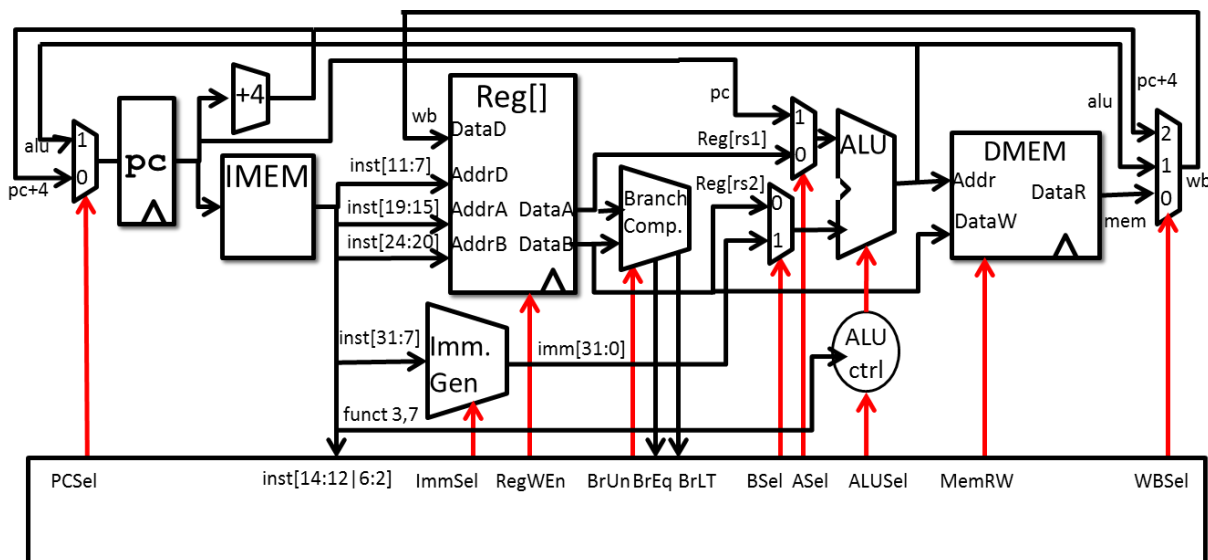
שאלה 100 (8 נקודות)

במעבד מסוג Single Cycle RISC-V, רוצים להוסיף תמיכה בפקודה החדשה המיוצגת בפורמט S-type:

sPC rs1, rs2

הפקודה sPC (store Program Counter), שומרת את ערך ה-PC בזיכרון בכתובת $\text{Mem}[\text{Reg}[\text{rs1}] + \text{Reg}[\text{rs2}]]$. כלומר ערך PC יכתב ל- $\text{Mem}[\text{Reg}[\text{rs1}] + \text{Reg}[\text{rs2}]]$. אין לשנות את הזיכרון או את ה-Register File, אך ניתן להוסיף בוררים.

בצעו את השינויים הנדרשים במסלול הנתונים של המעבד כך שיתמוך בפקודה בשרטוט הבא (ציירו על השרטוט):



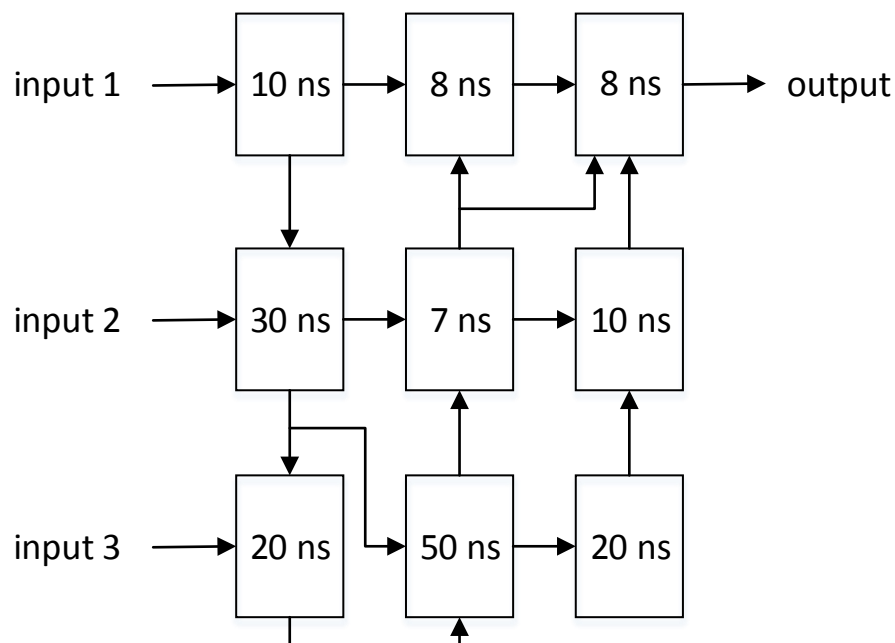
כתבו מהם קווי הבקרה לביצוע הפקודה (אם הוספתם בוררים/ים, הגדירו גם מהם קווי הבקרה שלו/שלהם ב- NewSel1 ו/או NewSel2 וסמנו אותם בשרטוט):

ImmSel =	<input type="checkbox"/>	BSEL =	<input type="checkbox"/>	WBSel =	<input type="checkbox"/>
RegWEn =	<input type="checkbox"/>	ALUSel =	<input type="checkbox"/>	NewSel1 =	<input type="checkbox"/>
ASel =	<input type="checkbox"/>	MemRW =	<input type="checkbox"/>	NewSel2 =	<input type="checkbox"/>
PCSel =	<input type="checkbox"/>				



שאלה 101 (7 נקודות)

נתונה המערכת הצירופית הבאה



המספרים בתוך המלבנים מסמנים את זמני ההשהיה של הרכיבים השונים במערכת.

להלן זמני ההשהיה של הרגיסטרים:

$$t_{pc-q}(FF) = 4 \text{ ns}$$

$$t_{cc-q}(FF) = 1 \text{ ns}$$

$$t_{setup}(FF) = 2 \text{ ns}$$

$$t_{hold}(FF) = 1 \text{ ns}$$

צנרו את המעגל לקבלת Throughput מקסימלי. לשם כך נדרש להשתמש במספר רגיסטרים מינימלי אפשרי. מהו מספר הרגיסטרים, ה-Latency וה-Throughput של המעגל המצונר?

מספר רגיסטרים

זמן ההשהיה (Latency)

ספיקה (Throughput)



שאלה 102 (8 נקודות)

לאחר ייצור מעבד *Single Cycle RISC-V* כפי שנלמד בכיתה. התגלתה תקלה בסיגנל הזיכרון *MemRW* מסוג *'1' stuck at*. כלומר $MemRW = Write$ כל הזמן ולא רק עבור פקודות כתיבה לזיכרון. שימו לב שלמעבד יש סיגנל בקרה אחד לזיכרון, אם הערך שלו הוא 0 מתבצעת קריאה מתוך הזיכרון, אחרת מתבצעת כתיבה. המצב ההתחלתי של הרגיסטרים הוא $RegFile[reg_addr] = reg_addr$ והמצב ההתחלתי של הזיכרון הוא $Mem[mem_addr] = mem_addr$. כאשר reg_addr הוא מספר הרגיסטר, ו- mem_addr הוא כתובת תחילת המילה בזיכרון.

לדוגמא:

התוכן של הרגיסטר $x2$ הוא $0x2$, התוכן של הרגיסטר $x31$ הוא $0x31$ וכו'.
התוכן של הזיכרון:

<i>mem_addr</i>	<i>value</i>
0x0	0x0
0x4	0x4
...	...
0x2000	0x2000
0x2004	0x2004
...	...

מעוניינים להריץ את הפקודה הבאה על המעבד

addi x5, x8, 0x104

לאחר ביצוע הפקודה לעיל, האם יהיה שינוי במצב הרגיסטרים ו/או מצב הזיכרון? אם כן, כיתבו את כל השינויים, אחרת כיתבו בטבלה "אין שינוי".

הערה: הכוונה בשינוי במצב הרגיסטרים/זיכרון היא שהתוכן של אחד או יותר מהרגיסטרים/כתובות הזיכרון השתנו לאחר ביצוע הפקודה. בתשובה שלכם ציינו את כל הרגיסטרים/כתובות שהשתנו, את הערך הישן שלהם (לפני ביצוע הפקודה) ואת הערך החדש שלהם (לאחר סיום ביצוע הפקודה).
אין צורך לרשום את הרגיסטרים/כתובות שהערך שלהם לא השתנה.

רגיסטרים

<i>Register number</i>	<i>Old value</i>	<i>New value</i>

זיכרון

<i>Memory address</i>	<i>Old value</i>	<i>New value</i>



שאלה 103 (7 נקודות)

נתונים מעבדי *Single Cycle RISC-V* ו-*Multi Cycle RISC-V* כפי שנלמדו בכתה.

מעבד ה-*single-cycle* פועל עם מחזור שעון T_{SC} .

מעבד ה-*multi-cycle* פועל עם מחזור שעון T_{MC} .

מה התחום שבו צריך להיות היחס $\frac{T_{SC}}{T_{MC}}$ כדי שמעבד ה-*multi-cycle* יריץ מהר יותר ממעבד ה-*single-cycle* את התוכנית הבאה:

```
addi x1, x0, 0xA
loop: lw x2, 0(x1)
      sw x2, 4(x1)
      subi x1, x1, 0x1
      beq x1, x0, loop
exit:
```

רשמו את תשובתכם במרובעים מטה, כך שבמרובע הימני (הגדול) מופיע רק מספר וברובע השמאלי (הקטן) מופיע רק סימן יחס (סימן היחס יכול להיות רק אחד מהסימנים הבאים: $<, >, \leq, \geq, =, \neq$).

רשמו את המספר וסימן היחס כך שהביטוי המתקבל עם השבר $\frac{T_{SC}}{T_{MC}}$ יהווה את התשובה לשאלה.

$\frac{T_{SC}}{T_{MC}}$



שאלה 104 (8 נקודות)

מהנדס משתמש במעבד מסוג MultiCycle RISC-V. זמן המחזור של המעבד הינו 100ns ובסה"כ ירוצו N פקודות על המעבד. מתוך כל הפקודות שירוצו, אחוז הפקודות מכל סוג מפורט בטבלה הבאה:

סוג הפקודה	אחוז מכלל הפקודות
Load	25%
Store	15%
R-type	35%
Branch	15%
Jump	10%

- 1) ייעול הקוד והפחתת מספר הפקודות הכולל ל- $0.75N$. ייעול זה אינו משנה את אחוז ביצוע סוגי הפקודות.
- 2) פיצול חלק ה- Execute שהינו צוואר הבקבוק ב- Datapath של מעבד זה. הפיצול יגרום לשיפור קצב השעון והפחתת זמן המחזור מ- 100ns ל- 80ns .
ייעול זה יגרור הגדלת שלב ה- Execute ממחזור אחד לשניים עבור כל הפקודות.
- 3) שיפור פקודת Load, והפחתת מספר המחזורים לביצוע הפקודה מ- 5 ל-4, ללא שינוי זמן המחזור.

א- מה יהיה זמן הריצה הכולל כתלות ב- N , על המעבד ללא שיפורים כלל?

Total runtime =

ב- מה יהיה זמן הריצה הכולל כתלות ב- N , על המעבד עם שיפור 1?

Total runtime =

ג- מה יהיה זמן הריצה הכולל כתלות ב- N , על המעבד עם שיפור 2?

Total runtime =

ד- מה יהיה זמן הריצה הכולל כתלות ב- N , על המעבד עם שיפור 3?

Total runtime =



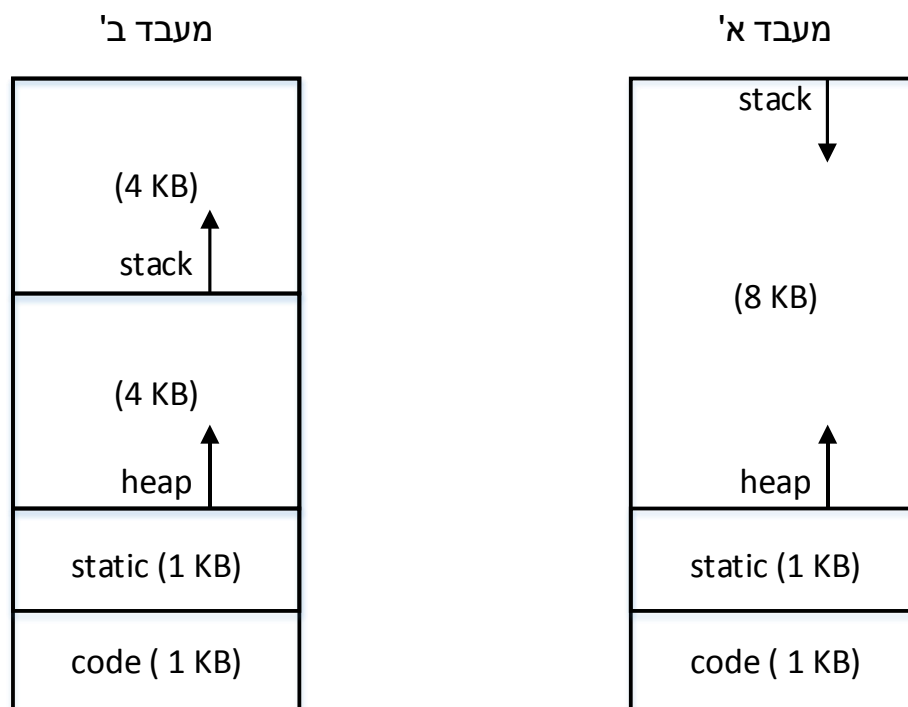
שאלה 105 (10 נקודות)

בהינתן הקוד הבא הכתוב בשפת C (שימו לב שהקוד דומה לקוד שראיתם בתרגול עם שינוי בפונקציה main):

```
int fact (unsigned int n) {
    if (n == 0)
        return 1;
    return n * fact(n-1);
}

int main() {
    int x = 1;
    return x + fact(x);
}
```

מעוניינים להריץ את הקוד על שני מעבדים מסוג RISC-V. שני המעבדים זהים לחלוטין חוץ מחלוקת הזיכרון (Memory Layout), כפי שמתואר בצירוף:



הערה: ($x \text{ KB}$) מסמן את גודל כל segment ב-KB כאשר $1 \text{ KB} = 2^{10} \text{ Bytes}$.

נגדיר את החריגה **stack overflow**: חריגה המתרחשת כאשר תוכנית קוד קריאה לשגרה מנסה לייצר frame חדש במחסנית, אבל חורג מגודל המקסימלי המותר למחסנית.



ענו על השאלות א'-ד' תחת **ההנחות** הבאות:

- i. אין הקצאות דינאמיות (לא משתמשים ב-heap).
- ii. קוד ה-assembly מקיים את קונבנציית הקריאה לפונקציות ומשתמש אך ורק ברגיסטרים: ra , sp , $a0$, $a1$.

א- סטודנט בקורס כתב קוד אסמבלי שמממש את הפונקציות הנתונות בשאלה על מעבד א' בצורה תקינה. המתרגל הריץ את הקוד של הסטודנט ללא שינוי על מעבד ב' וקיבל stack overflow. מדוע קרתה החריגה?

כעת, נניח שהקוד שרץ על שני המעבדים הוא תקין ורץ כהלכה. בנוסף, X הוא משתנה המתקבל כקלט בפונקציה main (ולא מוגבל לערך 1).

ב- מהו גודל ה-frame (בבתים) של הפונקציה main?

ג- מהו גודל ה-frame (בבתים) של הפונקציה fact?

ד- תנו דוגמה לערך של המשתנה X (המוגדר בפונקציה main) **שלא** גורם לחריגה מסוג stack overflow אם מריצים את התוכנית על מעבד א' אבל **גורם** לחריגה מסוג stack overflow אם מריצים את התוכנית על מעבד ב' (בהנחה שאין לנו את הבעיה מסעיף א').



שאלה 106 (7 נקודות)

במחשב עם מעבד מסוג Pipeline RISC-V הורץ הקוד הבא:

```
lw s1, 12(s0)
addi s0, s1, 0
sub s1, s0, s1
sw s1, 4(s0)
```

נתון שהמעבד בעל forwarding מלא ויחידת hazard detection.
(בין השלבים: $MEM \rightarrow EX, WB \rightarrow EX, WB \rightarrow ID$)

נתון שהפקודה הראשונה נמצאת בשלב ה-IF במחזור שעון מספר 1.
מלאו את הטבלה הבאה, ע"פ המעקפים שיתבצעו במהלך ריצת התוכנית (יתכן שיש פחות מעקפים ממספר השורות):

מס' מעקף	מחזור שעון	שם רגיסטר	מאיזה שלב	לאיזה שלב
1				
2				
3				
4				
5				
6				