



מערכות ספרתיות ומבנה המחשב (044252)

סמסטר אביב תשפ"א

בחינה סופית – מועד א

2 באוגוסט 2021

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 ולמחשבון.

הנחיות והוראות:

- הבחינה כתובה על גבי 20 עמודים כולל עמוד זה (בדקו בתחילת הבחינה שלא חסרים לכם עמודים). בסה"כ ישנן 15 שאלות: 10 שאלות אמריקאיות, ו- 5 שאלות פתוחות מרובות סעיפים.
- 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 הפתוחות בגוף המבחן, **במלבנים המסומנים לכך בלבד**.
- אין לתלוש או להפריד דפים מחוברת הבחינה, ממחברות הטיוטה ומדפי העזר.
- יש לכתוב את התשובות באמצעות עט שחור או כחול בלבד. אין לכתוב או לצייר בעט אדום.
- רשמו את מספר הסטודנט שלכם על חוברת הבחינה (בראש עמוד זה). **ודאו כי על מחברת הבחינה ועל טופס התשובות האמריקאי מודבקת מדבקת הנבחן שלכם.**
- לא מורדות נקודות (אין "קנס") בגין תשובה שגויה. לכן, בשאלות האמריקאיות כדאי לסמן תשובה כלשהי לכל שאלה.
- ציון השאלות האמריקאיות 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 מלבד מחשבון. 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 SystemVerilog הבא:

```
module my_mod(
    input logic clk,
    input logic rst,
    input logic start,
    output logic [31:0] vec
);

logic [1:0] z;
logic [1:0] b;
logic x;
logic y;

fsm fsm_inst(
    .clk(clk),
    .rst(rst),
    .start(start),
    .z(z)
);

always_comb begin
    vec = 32'hABCD;

    x = z[1] ^ z[0];
    y = z[1] | z[0];

    case (b)
        2'b00: begin
            vec = 32'h3579;
        end
        2'b01: begin
            vec = 32'hFFFF;
        end
        2'b10: begin
            vec = 32'h1248;
        end
        2'b11: begin
            vec = 32'h8421;
        end
    endcase
end

assign b = {x,y};

endmodule
```

1010-A
1011-B
1100-C
1101-D
1110-E
1111-F

```
module fsm(
    input logic clk,
    input logic rst,
    input logic start,
    output logic [1:0] z
);

typedef enum {
    idle_st,
    state_1,
    state_2
} sm_type;

sm_type current_state;
sm_type next_state;

// FSM synchronous procedural
block.
always_ff @(
    posedge clk,
    posedge rst) begin
    if (rst == 1'b1) begin
        current_state <= idle_st;
    end
    else begin
        current_state <= next_state;
    end
end

always_comb begin
    next_state = current_state;
    z = 2'b0;

    case (current_state)
        idle_st: begin
            if (start == 1'b1) begin
                next_state = state_1;
            end
        end
        state_1: begin
            next_state = state_2;
            z = 2'h1;
        end
        state_2: begin
            next_state = idle_st;
            z = 2'd3;
        end
    endcase
end

endmodule
```

b=00

b=11

0=01/10



בנוסף, נתון הכניסות ל- `my_mod` וארבע דיאגרמות אפשריות ליציאה `vec`.

<code>clk</code>							
<code>rst</code>							
<code>start</code>							
<code>vec1</code>	00003579		0000ffff		00001248		00003579
<code>vec2</code>	00003579				00008421		0000ffff
<code>vec3</code>	00003579		00001248		00003579		00008421
<code>vec4</code>	00003579				0000ffff		00001248

סמן את התשובה הנכונה ביותר:

א. `vec` יראה כמו דיאגרמה `vec1` בסימולציה.

ב. `vec` יראה כמו דיאגרמה `vec2` בסימולציה.

ג. `vec` יראה כמו דיאגרמה `vec3` בסימולציה.

ד. `vec` יראה כמו דיאגרמה `vec4` בסימולציה.

ה. אף תשובה לא נכונה



שאלה 2 (5 נקודות):

נגדיר משתנה כניסה טרינארי של פונקציה כמשתנה שיכול לקבל את אחד מן הערכים $\{0,1,2\}$ (משתנה בבסיס 3).

נגדיר מוצא אוקטלי של פונקציה כמוצא שיכול לקבל אחד מתוך הערכים $\{0,1,2,3,4,5,6,7\}$ עבור קלט נתון (מוצא בבסיס 8).

נסמן ב- f פונקציה בעלת n משתני כניסה טרינארים ויציאה אוקטלית אחת:

$$f: \{0,1,2\}^n \rightarrow \{0, \dots, 7\}$$

נגדיר פונקציה קבועה כפונקציה שמוצאה זהה עבור כל קלט.

$$8^n - 8$$

in	out
00	1
01	1
02	
10	1
20	
11	1
21	
12	1
22	1

כמה פונקציות שונות f שאינן קבועות קיימות?

א. 3^{3^n}

ב. $3^{8^n} - 3$

ג. $3^{8^n} - 8$

ד. $8^{3^n} - 3$

ה. $8^{3^n} - 8$

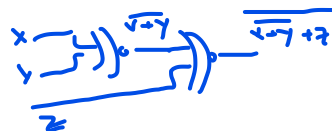
שאלה 3 (5 נקודות):

$$F(x,y,z) = xz' + yz$$

ממשו את הפונקציה בעזרת שערי NOR עם 2 כניסות בלבד.

מהו מספר שערי NOR המינימלי שיש צורך להשתמש בהם?

$$\overline{\overline{(x \cdot y)} + z} = (x \cdot y) + z$$



א. 1

ב. 2

ג. 3

ד. 4

ה. 5



שאלה 4 (5 נקודות):

נתונה הפונקציה הבאה:

$$f(a, b, c, d, e) = \sum (6, 7, 22, 23, 31) + \sum_{\phi} (18, 30)$$

סטודנט בקורס מעוניין לממש את הפונקציה.

לרשות הסטודנט יש שער OR יחיד עם 32 כניסות, הקבועים '0' ו-'1', ובנוסף רכיבים כמתואר בטענות הבאות:

✓ טענה 1: ניתן לממש את הפונקציה בעזרת מפענח 3->8 ושער AND יחיד בעל שתי כניסות בלבד.

✓ טענה 2: ניתן לממש את הפונקציה בעזרת מפענח 4->16 יחיד, וללא שערים לוגיים נוספים בלבד.

✓ טענה 3: ניתן לממש את הפונקציה בעזרת מפענח 5->32 יחיד בלבד.

כאשר לכל המפענחים יש כניסת Enable כנלמד.

הערה: בתשובות להלן כאשר מצוין ששתי טענות נכונות, הכוונה היא שכל אחת מהן נכונה בנפרד.

סמנו את התשובה הנכונה ביותר:

א. כל הטענות לא נכונות. ✗

ב. טענה 3 נכונה, וטענות 1 ו-2 לא נכונות. ✗

ג. טענות 2 ו-3 נכונות, וטענה 1 לא נכונה. ✗

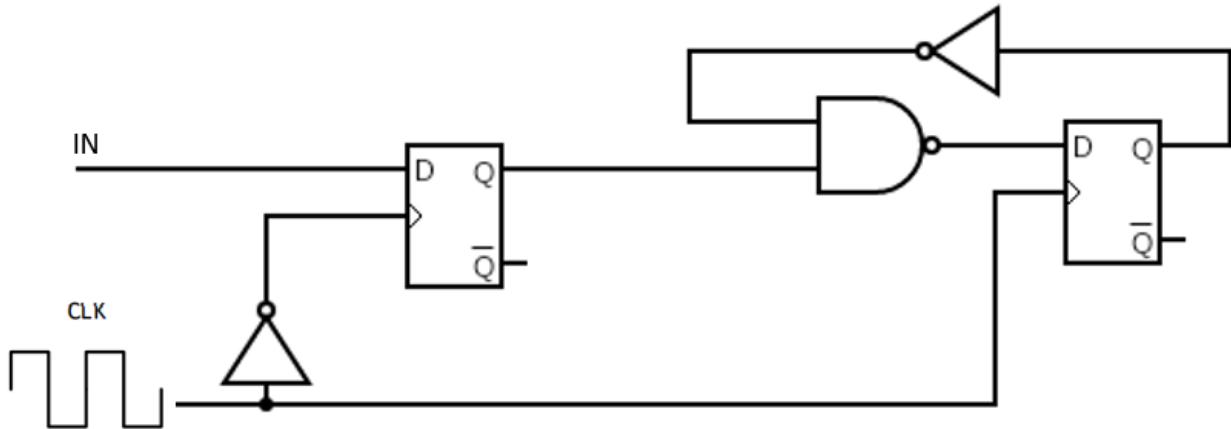
ד. טעות 1 ו-3 נכונות, וטענה 2 לא נכונה. ✗

ה. כל הטענות נכונות. (✓)



שאלה 5 (5 נקודות):

נתון המעגל הבא:



שימו לב שהשערים הלוגים בשרטוט הינם NAND ו-NOT בלבד.

זמני ההשהיה של הרכיבים מופיעה בטבלה הבאה (נתונים ב ns):

2-2: $t_{clk} = t_{pd}(FF) + t_{pd}(NAND) + t_{pd}(NAND) + t_{pd}(FF) = 10 + 5 + 5 + 10 = 30$
 $33 \leq 30$

	$t_{pd}/t_{pc \rightarrow Q}$	t_{setup}
FF	10	X
NOT	2	-
NAND	5	-

1-2: $t_{pd}(FF) + t_{pd}(NAND) + t_{pd}(NAND) + t_{pd}(FF) = 10 + 5 + 5 + 10 = 30$
 $33 \leq 30$

נתון ש- $t_{cd}(NOT) = t_{pd}(NOT)$

זמן המחזור של המעגל הינו $T = 50ns$, כאשר הזמן שבו השעון שווה '1' שווה לזמן שבו השעון שווה '0' ($Duty Cycle = 50\%$).

הניחו כי הכניסה IN עומדת בתנאי setup ו-hold, ותנאי hold במערכת מתקיימים.

מהו ערכו המקסימלי האפשרי של X (זמן setup של ה FF) כך שהמערכת עומדת בתנאי התזמון?

א. $5ns$

ב. $8ns$

ג. $17ns$

ד. $33ns$

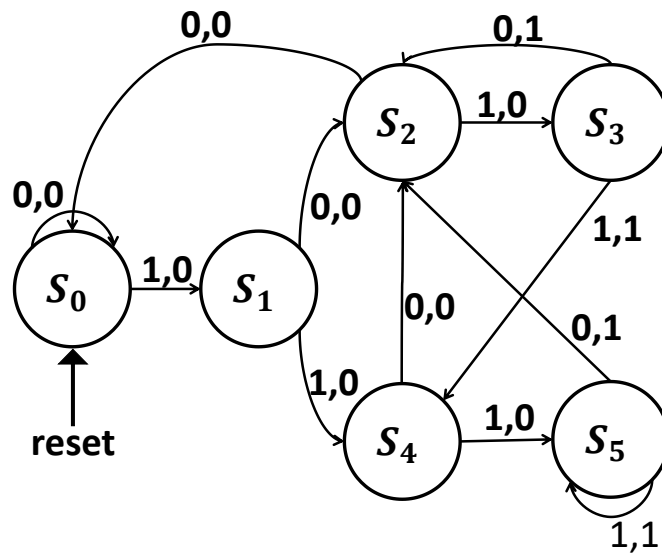
ה. המערכת לא יכולה לעמוד בתנאי setup ללא תלות בערכו של X.



שאלה 6 (5 נקודות):

נתונה דיאגרמת המצבים של מערכת Mealy הבאה.

הסימון על החיצים הינו: $input, output$.



קלט המערכת עבור כל מחזור הינו:

cycle	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
input	1	0	1	1	1	1	1	0	1	0	1	0	1	1	1	0

בתחילת מחזור 0 מכונת המצבים נמצאת במצב S_0 .

מה יהיה פלט המערכת (הסיבית השמאלית ביותר הינה המוצא במחזור מספר 0)?

א. 0001010100001000

ב. 0001001100010110

ג. 0001011101010101

ד. 0001010101001001

ה. 1001000101010101



שאלה 7 (5 נקודות):

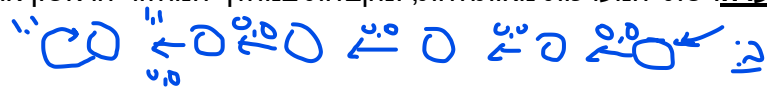
מהנדס חשמל מעוניין לממש שתי מערכת עקיבה מסוג מילי בעלות סיבית כניסה אחת וסיבית מוצא אחת. שתי המערכות מתייחסות לרצף הסיביות שנקלטו עד כה כמספר בינארי X (למשל אם סדרת הקלט הינה 0101, אז $X=5$).

מערכת א': מתייחסת לסיבית הנקלטת בכל מחזור כסיבית ה- LSB של המספר X (כלומר הסיבית הראשונה שנקלטה הינה סיבית ה- MSB של המספר).

מערכת ב': מתייחסת לסיבית הנקלטת בכל מחזור כסיבית ה- MSB של המספר X (כלומר הסיבית הראשונה שנקלטה הינה סיבית ה- LSB של המספר).

שתי המערכות מוציאות 1 אמ"מ המספר שהתקבל עד כה מתחלק ב-16 ללא שארית.

הערה: שתי המערכות מאותחלות, ומקבלות במהלך המחזור הראשון את הקלט הראשון.



ממשו את מכונות המצבים כך שיכילו מספר מצבים מינימלי אפשרי, וסמנו את התשובה הנכונה ביותר:

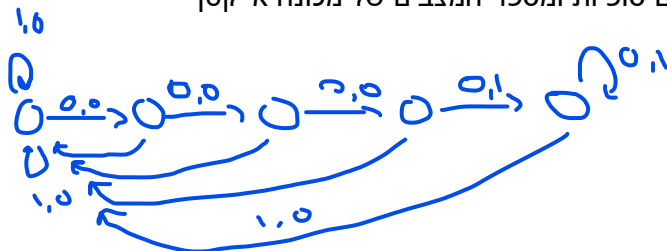
☒ א. אחת מהמכונות לא ניתנות למימוש כמכונות מצבים סופית.

☒ ב. שתי המכונות לא ניתנות למימוש כמכונות מצבים סופית.

ג. שתי המכונות ניתנות למימוש כמכונות מצבים סופיות ומספר המצבים של מכונה א' גדול ממספר המצבים של מכונה ב'.

☒ ד. שתי המכונות ניתנות למימוש כמכונות מצבים סופיות ומספר המצבים של מכונה א' שווה למספר המצבים של מכונה ב'.

ה. שתי המכונות ניתנות למימוש כמכונות מצבים סופיות ומספר המצבים של מכונה א' קטן ממספר המצבים של מכונה ב'.





שאלה 8 (5 נקודות):

נתון מעבד *Single Cycle RISC-V*.

מהנדסת מעוניינת להריץ את הפקודה *BeqMemReg* על המעבד כפקודה אמיתית (לא פסאודו פקודה).

$$val (address, (rs1) - imm) = reg(rs2)$$

פורמט הפקודה הינו:

BeqMemReg rs1, rs2, imm

הפקודה בודקת האם $Mem[Reg[rs1]+imm] = Reg[rs2]$. אם כן, אז הקפיצה תתבצע לפי $PC=Reg[rs2]$, ואם לא אז $PC=PC+4$.

בחרו את התשובה המחמירה ביותר שמאפשרת עבודה תקינה של המעגל עבור הפונקציה המוצעת, ולא פוגעת בביצוע שאר הפונקציות שנלמדו.

הערה: אם תשובה X מוכלת בתוך תשובה Y , אז תשובה X מחמירה יותר.

☒ ניתן לממש את הפקודה ללא שינוי של ה- *Datapath* כלל.

☒ ניתן לממש את הפקודה ע"י חיווט מחדש ב- *Datapath* בלבד.

☒ ניתן לממש את הפקודה ע"י חיווט מחדש ב- *Datapath*, הוספת בורר יחיד, והוספת סיגנלי בקרה בהתאם.

☒ ניתן לממש את הפקודה ע"י חיווט מחדש ב- *Datapath*, הוספת בורר יחיד, הרחבת בורר יחיד והוספת סיגנלי בקרה בהתאם.

☒ כל התשובות אינן נכונות.



שאלה 9 (5 נקודות):

מעוניינים להוסיף מימוש של הפקודה *RMW* במעבד *multicycle RISC-V*. פקודה זו קוראת מילה מהזכרון בכתובת ששמורה ברגיסטר *rs1* ושומרת אותה ברגיסטר *rd*. לאחר מכן כותבת מילה מרגיסטר *rs2* לזכרון לאותה הכתובת. ניתן לבצע שינויים במעבד שכוללים הוספה של חיוטים והרחבה או הוספה של בוררים.

פקודה זו בעלת הפורמט:

rmw rd, rs1, rs2

lw rd, rs1

$reg[rd] \leftarrow Mem[reg[rs1]]$

$Mem[reg[rs1]] \leftarrow reg[rs2]$

sw rs1, rs2

הערה: רגיסטר *rd* שונה מרגיסטרים *rs1* ו-*rs2* (הכוונה למספר הרגיסטר ולא לתוכנו).

מה מספר המחזורים המינימלי הנדרש לביצוע פקודה זו?

4. א.

5. ב.

6. ג.

7. ד.

8. ה.



שאלה 10 (5 נקודות):

התוכנית הבאה מורצת במעבד RISC-V Pipelined בעל הנתונים הבאים:

- כולל יחידת Forwarding מלא, כלומר בשלב ה-EXE כל רכיב יכול לקבל את ערכו העדכני ביותר של כל רגיסטר. ה-Forwarding מתבצע בין השלבים: $WB \rightarrow ID, WB \rightarrow EXE, MEM \rightarrow EXE$.
- לא קיימת יחידת Load Hazard Detection.
- עבור פקודות branch, המעבד מניח שקפיצות אינן נלקחות, ובמידה שכן, לא מתבצע Flush ע"י המעבד.
- ההחלטה על ביצוע Branch מתקבלת בשלב ה-EXE.

עליכם לדאוג להוסיף פקודות nop במקומות הדרושים על מנת שהקוד ירוץ כהלכה על המעבד הנתון.

1. `add t1, x0, x0`
2. `lw t2, 0(t1)` **# initialized t2 to be 1**
3. `lw t3, 4(t1)` **# initialized t3 to be 3**
4. `addi t5, x0, 1`
5. `Loop: sub t3, t3, t2`
6. `slt t4, x0, t3`
7. `add t10, t4, x0`
8. `beq t4, t2, loop`
9. `add t5, t10, t5` **← 2 nops**

10 trash

מה מספר ה-nops המינימלי שצריך להוסיף כך שהקוד ירוץ כמתוכנן?

א. nop יחיד.

ב. 2 nop-ים

ג. 3 nop-ים.

ד. 4 nop-ים.

ה. התוכנית רצה בצורה תקינה בלי nops.



החל מהעמוד הבא מתחיל החלק של השאלות פתוחות

(שאלות 11 – 15)



שאלה 11 (10 נקודות):

מהנדס בונה פונקציה עבור משחק. הקלט לפונקציה הינו מספר דו-ספרתי בבסיס 4 בין $(1)_4$ ל- $(33)_4$, כולל (כלומר הקלט $(0)_4$ לא יכול להתקבל).

הפונקציה מוציאה 1 אם המספר מתחלק ב-3 או מכיל את הספרה 3.

לדוגמא, אם הקלט הוא $(12)_4$ הפלט הוא 1 (כי $(6)_{10} = (12)_4$ מתחלק ב-3). אם הקלט הוא $(13)_4$ הפלט הוא 1 (כי אחת הספרות היא 3).

המספר מתקבל כקלט בינארי $(a_3a_2a_1a_0)_2$, כאשר a_1a_0 היא מייצגת את הספרה הימנית של המספר בבסיס 4, a_3a_2 הינה הספרה השמאלית.

$$xy = x \cdot 4 + y \cdot 1$$

סעיף א' – מלאו את מפת הקרנו הבא בהתאם לפונקציה הנדרשת:

00 0
01 0
02 0
03 1
10 0
11 1
20 0
21 1
22 0
23 1
30 1
31 1
32 1
33 1

$a_3a_2 \backslash a_1a_0$	00	01	11	10
00			1	
01			1	1
11	1	1	1	1
10		1	1	

a_3a_2 00 01 11 10
 a_1a_0 00 01 11 10
 00 0 1 1 1
 01 0 1 1 1
 11 1 1 1 1
 10 0 1 1 0

סעיף ב' – צמצמו את הפונקציה כסכום מכפלות. מה הביטוי המינימלי?

$$f(a_0, a_1, a_2, a_3) = a_0a_1 + a_2a_3 + a_1a_2 + a_0a_3$$



שאלה 12 (10 נקודות):

נתונה המערכת הצירופית הבאה. המספרים מציינים את השהיית הרכיב בנוי שניות.

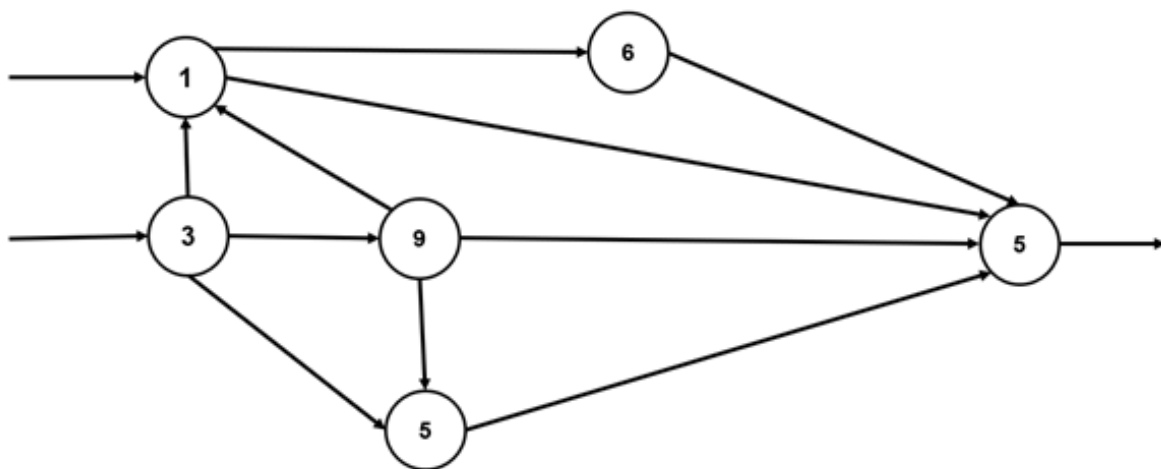
באפשרותכם להשתמש בשני סוגים של רגיסטרים:

▪ רגיסטרים אידאליים, $T_{PCQ} = 0ns$, $T_{setup} = 0ns$, אשר עלותם גבוהה.

▪ רגיסטרים בעלי $T_{PCQ} = 1ns$, $T_{setup} = 1ns$, ללא עלות.

הניחו שתנאי HOLD מתקיימים במעגל.

עליכם לצנר את המערכת הנתונה כדי לקבל תפוקה מקסימלית בעדיפות ראשונה, השהייה מינימלית בעדיפות שניה ועלות מינימלית בעדיפות שלישית.



א. מהי התפוקה המקסימלית?

$\frac{1}{9}$

ב. בכמה רגיסטרים אידאליים תשתמשו?

4

-סעיף ג' בעמוד הבא-



ג. כעת אין אפשרות להשתמש ברגיסטרים אידאליים, אלא רק באלו עם השהייה. מהי התפוקה המקסימלית כעת?





שאלה 13 (10 נקודות):

משדר ומקלט עובדים עפ"י פרוטוקול תקשורת טורית הדומה לזה שנלמד בקורס: תחילה משודרת סיבית התחלה (*Start Bit*), לאחר מכן משודרות X סיביות נתון, ולבסוף סיבית סיום (*Stop Bit*). ידוע כי זמן מחזור השעון של המשדר הוא $4.8ns$, וזמן מחזור השעון של המקלט הוא $5ns$. מבחינת המשדר והמקלט, זמן הסיבית (*Tbit*) הוא חמישים מחזורי שעון פנימיים ($N_T = N_R = 50$). הניחו כי המקלט מזהה באופן מידי את סיבית ההתחלה (*Start Bit*), כמו כן נדרש שגם סיבית הסיום תידגם בצורה תקינה. מהו ה- X (כמות סיביות המידע) המקסימאלי? פרטו את הדרך בקצרה.

דרך:

תשובה סופית:

10



שאלה 14 (10 נקודות):

נתונים מעבדי *single-cycle RISC-V* ו-*multicycle RISC-V* כפי שנלמדו בכיתה.

בנוסף נתון מעבד *RISC-V Pipelined* בעל הנתונים הבאים:

- כולל יחידת Forwarding מלא, כלומר בשלב ה-*EXE* כל רכיב יכול לקבל את ערכו העדכני ביותר של כל רגיסטר. ה-*Forwarding* מתבצע בין השלבים: $WB \rightarrow ID, WB \rightarrow EXE, MEM \rightarrow EXE$.
- קיימת יחידת Load Hazard Detection.
- עבור פקודות branch, המעבד מניח שקפיצות אינן נלקחות, ומבצע Flush במידה וכן.
- ההחלטה על ביצוע Branch מתקבלת בשלב ה-*EXE*.

מעבד ה-*single-cycle RISC-V* עובד עם מחזור שעון T_{single} .

מעבד ה-*multicycle RISC-V* עובד עם מחזור שעון T_{multi} .

מעבד ה-*pipelined RISC-V* עובד עם מחזור שעון $T_{pipelined}$.

נתונה התוכנית הבאה:

0x1AA0 000C		<i>addi s0, x0, 4</i>	4	} $\times 4$
0x1AA0 0010	Loop:	<i>add s1, s2, s2</i>	4	
0x1AA0 0014		<i>lw s3, 4(s4)</i>	5	
0x1AA0 0018		<i>add s2, s3, s4</i>	4	
0x1AA0 001C		<i>addi s0, s0, -1</i>	4	
0x1AA0 0020		<i>bne s0, x0, Loop</i>	3	
0x1AA0 0024	Exit			

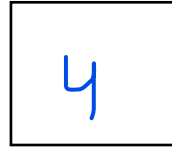
ניתן להניח שהתוכנית רצה ללא תקלות וללא חריגות.

סיג: 21 mult: 4



א. מה התחום שבו צריך להיות היחס $\frac{T_{single}}{T_{multi}}$ כדי שמעבד ה-*multicycle* יסיים את ריצת התוכנית מהר יותר. רשמו את תשובתכם במרובעים, כאשר המרובע מכיל **מספר** והמרובע השמאלי מכיל **רק** סימן יחס (אחד מן הסימנים הבאים: $<, >, \leq, \geq, =, \neq$).

$$\frac{T_{single}}{T_{multi}}$$



$$T_m < 2T_s$$

ב. מה התחום שבו צריך להיות היחס $\frac{T_{pipelined}}{T_{multi}}$ כדי שמעבד ה-*multicycle* יסיים את ריצת התוכנית מהר יותר. רשמו את תשובתכם במרובעים, כאשר המרובע מכיל **מספר** והמרובע השמאלי מכיל **רק** סימן יחס (אחד מן הסימנים הבאים: $<, >, \leq, \geq, =, \neq$).

$$\frac{T_{pipelined}}{T_{multi}}$$





שאלה 15 (10 נקודות):

נתון מעבד *pipelined RISC-V* ללא מנגנון forwarding וללא *hazard detection unit*.

כמו כן נתון הקוד הבא:

```
0x1AA0 0000    main :    lw t2, 0(s1)
0x1AA0 0004                sw t2, 4(s1)
0x1AA0 0008                addi t2, a0, -1
0x1AA0 000C                addi t1, t2, 0
0x1AA0 0010                add a1, t2, t2
```

סעיף א' -

עליכם למלא בטבלה הבאה את כמות פקודות ה-NOP המינימאלית הנדרשת לריצה תקינה של הקוד על המעבד הנתון.

לדוגמה: אם לדעתכם צריך להוסיף 3 פקודות NOP בין פקודות 5 ל-6 אז תמלאו באופן הבא:

כמה NOP להוסיף	אחרי פקודה מספר	לפני פקודה מספר
3	5	6

מלאו או הטבלה הבאה (אין הכרח למלא את כולה):

כמה NOP להוסיף	אחרי פקודה מספר	לפני פקודה מספר

-סעיף ב' עמוד הבא-



סעיף ב' -

כעת הוחלט להוסיף forwarding מסוג MEM->EXE.

מלאו או הטבלה הבאה איך תשתנה כמות פקודות ה-NOP המינימלית הנדרש לריצה תקינה של המעבד הנתון לאחר השינויים שנוספו.

לפני פקודה מספר	אחרי פקודה מספר	כמה NOP להוסיף