מערכות ספרתיות ומבנה המחשב (044252)

סמסטר חורף תשפ"א

בחינה סופית – מועד א - פתרון

4 בפברואר 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:

- יש לענות על הבחינה במערכת המודל בשני חלקים נפרדים - חלק אמריקאי וחלק לשאלות הפתוחות. בנוסף יש לענות על הבחינה כגיבוי המערכת הטפסים של מייקרוסופט - קישור באתר המודל. יש להעלות קובץ המכיל את טיוטת הפתרון לאתר המודל.
- במבחן זה ישנן 15 שאלות. 10 שאלות אמריקאיות, ו- 5 שאלות פתוחות מרובות סעיפים.
- לא מורדות נקודות (אין "קנס") בגין תשובה שגויה. לכן, בשאלות האמריקאיות, כדאי לסמן תשובה כלשהי לכל שאלה.
- בסיומו של המבחן יינתנו 15 דקות לצורך ביצוע סריקה של טיוטת הבוחן.
- אסור שימוש בכל חומר חיצוני מלבד מחשבון ודפי העזר. 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, או אחרת. בפרט, אין להשתמש בטלפון הסלולארי לכל שימוש שאינו צילום מסך המחשב, סריקת המבחנים או התנהלות מול הסגל והמשגיחים.
- עליכם להשאיר את המצלמה אשר מצלמת את פניכם במצב פעיל לאורך כל שלבי הבחינה.
- עליכם להשאיר את המצלמה אשר מצלמת את מסך המחשב במצב פעיל לאורך כל שלבי הבחינה.
- עליכם להשאיר את השמע של המחשב פעיל לכל אורך הבחינה.
- שימוש בטלפון הנייד יתאפשר לצורך וידוא נהלים/פתרון בעיות על ידי משגיח/איש סגל, לצורך ביצוע סריקות של מחברת הבחינה/טיוטה ולצורך צילום מסך המחשב בלבד. כל שימוש אחר בטלפון נייד בזמן הבחינה הוא אסור. בזמן הבחינה מכשיר הטלפון הנייד צריך להיות על מצב רטט.

בהצלחה!

שאלה 1 (5 נקודות)

נתונים שני הרכיבים הבאים func1, func2:

```
module func1 (
    input logic clk,
    output logic [5:0] a,
    output logic [1:0] b
);

    always_ff @(posedge clk) begin
        {a[1],a[4]} <= a[2:3];
        a[2:3] <= {a[0],a[5]};
        {a[0],a[5]} <= {a[1],a[4]};
    end

    assign b = {a[0],a[5]};

endmodule

module func2 (
    input logic clk,
    output logic [5:0] a,
    output logic [1:0] b
);

    always_ff @(posedge clk) begin
        {a[1],a[4]} = a[2:3];
        a[2:3] = {a[0],a[5]};
        {a[0],a[5]} = {a[1],a[4]};
    end

    assign b = {a[0],a[5]};

endmodule
```

מעוניינים להשוות בין הפעולה של שני הרכיבים בנפרד. קובעים ערך התחלתי של הווקטור a, ומחכים לעליית שעון אחת.
לפני עליית השעון ערך a הינו קבוע, ידוע וזהה לשני הרכיבים. עבור איזה ערך של a, נקבל בכל רכיב מוצא b שונה לאחר עליית השעון?

א. a = 6'd0
ב. a = 6'd25
ג. a = 6'd44
ד. a = 6'd7
ה. ערך היציאה b בשני הרכיבים תמיד יהיה זהה לאחר עליית השעון, לכל ערך של a לפני עליית השעון.

<u>פתרון:</u>

רכיב func1 עושה השמות של non-blocking, ולכן הערך שb יקבל לאחר עליית השעון הוא ערכי {a[1],a[4]} מלפני עליית השעון. לעומת זאת רכיב func2 עושה השמות blocking, ולכן הערך שb יקבל לאחר עליית השעון הוא ערכי a[2:3] מלפני עליית השעון. התשובה היחידה שבה ביטים אלו שונים לפני עליית שעון היא תשובה ג': a = 6'd44 = 6'b101100.

שאלה 2 (5 נקודות)

נתונים שני מספרים המיוצגים בשיטת FP, כאשר $B = 127$ (ראה דפי עזר).
מצא את סכומם $a + b$, כאשר:

$$a = 0x41840000$$
$$b = 0xC1000000$$

א. $0x02840000$
ב. $0xF2840000$
ג. $0x41080000$
ד. $0x41000000$
ה. $0x41C40000$

פתרון: תשובה ג'.

נשים לב כי $a = 16.5,\ b = -8$ ולכן סכומם: $a + b = 8.5$.

שאלה 3 (5 נקודות)

עליכם לתכנן מערכת עקיבה מסוג Mealy בעלת כניסה יחידה ומוצא יחיד z. המערכת תפיק z=1 כאשר המספר שנכנס עד כה החל מה MSB (כולל הכניסה הנוכחית המהווה את סיבית ה- LSB של המספר) הוא זוגי והכניסה הנוכחית, x_i, זהה לכניסה אשר התקבלה שני מחזורי שעון קודם לכן (כלומר כאשר $x_i = x_{i-2}$).

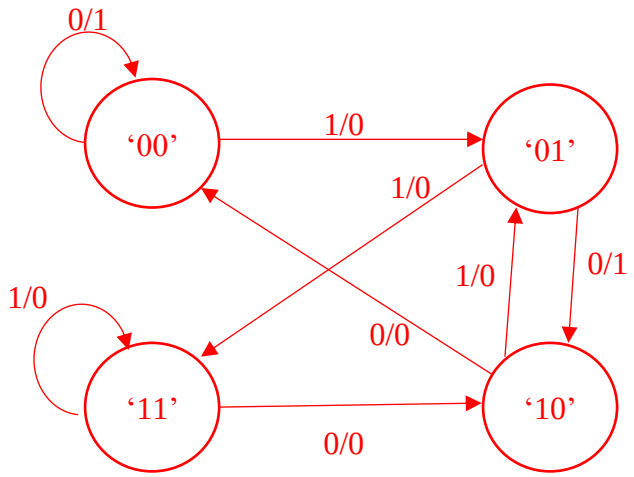
הניחו כי לפני שהתקבלה הכניסה הראשונה כל הכניסות היו בעלות הערך 0. התייחסו למספר 0 כאל מספר זוגי.

כמה מצבים קיימים במכונה המצומצמת אשר מקיימת את הדרישות?

א. 4
ב. 8
ג. בין 9 ל-15
ד. 16
ה. לא ניתן למימש מכונה זו בעזרת מספר מצבים סופי

פתרון:

תשובה א'.
רצף מספרים בינארי הוא זוגי רק אם הוא מסתיים ב-0. מכך נובע כי אנו מחפשים את אחד מן הרצפים '000' ו-'010':



שאלה 4 (5 נקודות)

נתונה הפונקציה: $f(x,y,z) = and(g_1(x,y,z), g_2(x,y,z))$.
כאשר: $g_1(x,y,z) = xz + y'z' + xyz'$
כאשר: $g_2(x,y,z) = y'z + yz'$

טענה A: ניתן לממש את הפונקציה $f(x,y,z)$ באמצעות בורר $\mathbf{2 \to 1}$ יחיד, שער XOR (של שתי כניסות) והקבועים 0 ו 1 בלבד.
טענה B: ניתן לממש את הפונקציה $f(x,y,z)$ באמצעות בורר $\mathbf{4 \to 1}$ יחיד, והקבועים 0 ו 1 בלבד.
טענה C: ניתן לממש את הפונקציה $f(x,y,z)$ באמצעות בורר $\mathbf{8 \to 1}$ יחיד והקבועים 0 ו 1 בלבד.

הערה: כלל הבוררים ללא רגלי Enable.

בחרו את המשפט הנכונה:

א. טענות A,B,C נכונות
ב. טענות B,C נכונות וטענה A אינה נכונה.
ג. טענות A,C נכונות וטענה B אינה נכונה.
ד. טענה C נכונה וטענות A,B אינן נכונות.
ה. טענות A,B,C אינן נכונות

פתרון:
סעיף א'.
טבלת האמת

X	Y	Z	g1	g2	f
0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	0	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	0	0

מימוש באמצעות בורר 2→1:

X=0, F=0
X=1, F=y xor z

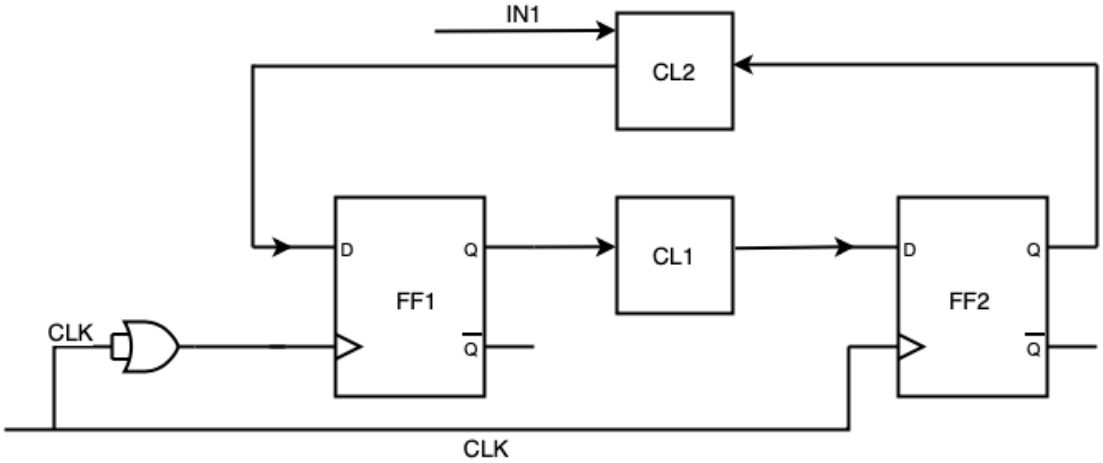
מימוש באמצעות $\mathbf{1} \rightarrow \mathbf{4}$:

Y	Z	out
0	0	0
0	1	X
1	0	X
1	1	0

לכן טענות **A,B** נכונות.
טענה **C** נכונה, מכיוון שניתן לממש כל פונקציה של 3 משתנים באמצעות בוררים $\mathbf{1} \rightarrow \mathbf{8}$ והקבועים 0 ו- 1. \לכן טענות **A,B,C** נכונות.

שאלה 5 (5 נקודות)

נתונה דיאגרמת המעגל הבאה:



זמן מחזור השעון במערכת, T, הוא $26ns$.

הרכיבים הם בעלי הפרמטרים הבאים:

	t_{cd}/t_{ccq}	t_{pd}/t_{pcq}	t_{hold}	t_{su}
$FF1$	?	?	$1ns$	$2ns$
$FF2$	$13ns$	$18ns$	$1ns$	$2ns$
$CL1$	$5ns$	$7ns$		
$CL2$	$4ns$	$6ns$		
OR	$2ns$	$2ns$		

נתון בנוסף שהמסלול In1->FF1 עומד בתנאי hold ו- setup של FF1.

מהם טווחי ה-t_{pcq} וה-t_{ccq} של $FF1$ אשר יאפשרו פעילות תקינה של המעגל?

א. $-4ns \le t_{ccq} \le t_{pcq} \le 14ns$
ב. $-6ns \le t_{ccq} \le t_{pcq} \le 15ns$
ג. $-2ns \le t_{ccq} \le t_{pcq} \le 19ns$
ד. $9ns \le t_{ccq} \le t_{pcq} \le 33ns$
ה. המעגל אינו פועל כשורה

פתרון:

תשובה ב'.

ראשית נבין כי המעגל הנתון בעצם שקול למעגל עם skew על השעון, כאשר השעון מגיע ל-$FF1$ מאוחר יותר. ערכו של ה-skew הוא:

$$skew = t_{pd}(or) = 2 = 2ns$$

מכאן נמשיך על ידי בדיקה של תנאי משטר הזמנים הדינאמיים במסלולים השונים. נשים לב כי מיקומו של ה-skew במשוואת הפוך מאלו אשר מופיעות בתרגול, כיוון שכיוון ה-skew במקרה הנתון הפוך:

$FF1 \rightarrow FF2$:

$$t_{pcq}(FF1) + t_{pd}(CL1) + t_{su}(FF2) + skew \leq T$$

$$t_{pcq}(FF1) + 7 + 2 + 2 \leq 26$$

$$t_{pcq}(FF1) \leq 15$$

$$t_{hold}(FF2) \leq t_{ccq}(FF1) + t_{cd}(CL1) + skew$$

$$1 \leq t_{ccq}(FF1) + 5 + 2$$

$$-6 \leq t_{ccq}(FF1)$$

דבר אשר מתקיים תמיד.

עלינו לבדוק כי התנאים מתקיימים גם במסלול השני:

$FF2 \rightarrow FF1$:

$$t_{pcq}(FF2) + t_{pd}(CL2) + t_{su}(FF1) \leq T + skew$$

$$18 + 6 + 2 = 26 \leq 28 = 26 + 2$$

$$t_{hold}(FF1) + skew \leq t_{ccq}(FF2) + t_{cd}(CL2)$$

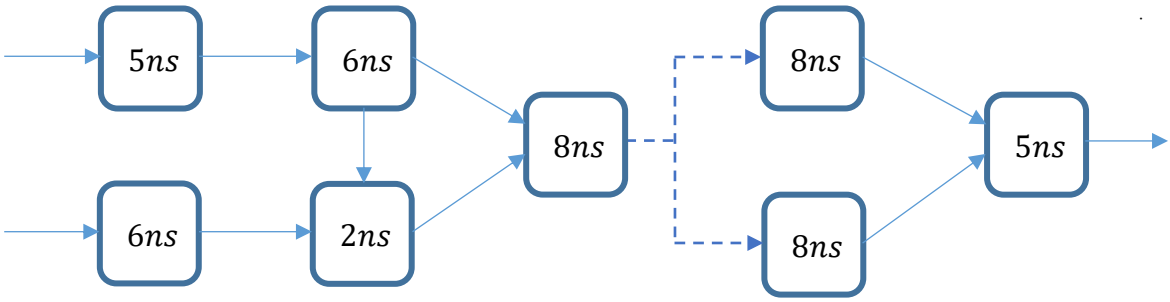
$$1 + 2 = 3 \leq 13 + 4 = 17$$

התנאים מתקיימים.

שאלה 6 (5 נקודות)

עליכם לצנר את המערכת הבאה כאשר $throughput$ גבוה ככול הניתן הוא בעדיפות עליונה ומספר רגיסטרים נמוך ככול הניתן הוא בעדיפות שנייה. זמני ההשהיה של כל רכיב רשומים בתוכו.

הערה: שימו לב כי הקו המקווקו מחזיק ערך זהה.



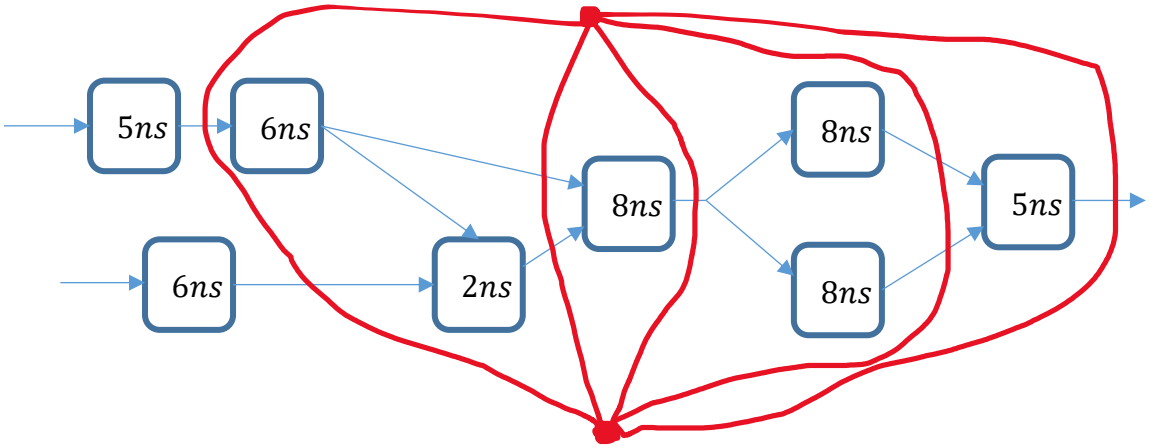
מהו מספר הרגיסטרים אשר דרוש לצורך צינור המערכת על פי סדר העדיפות הנתון?

א. 7 רגיסטרים
ב. 8 רגיסטרים
ג. 9 רגיסטרים
ד. 10 רגיסטרים
ה. 11 רגיסטרים

פתרון:

תשובה ב'.

נצייר את המעגל מחדש ונעזר בשיטת הקווים:



כלומר דרושים 8 רגיסטרים לצורך הצינור. כיוון שהיחידה האיטית ביותר היא בעלת השהייה של $8ns$, כלומר הספיקה במערכת היא $125MHz$.

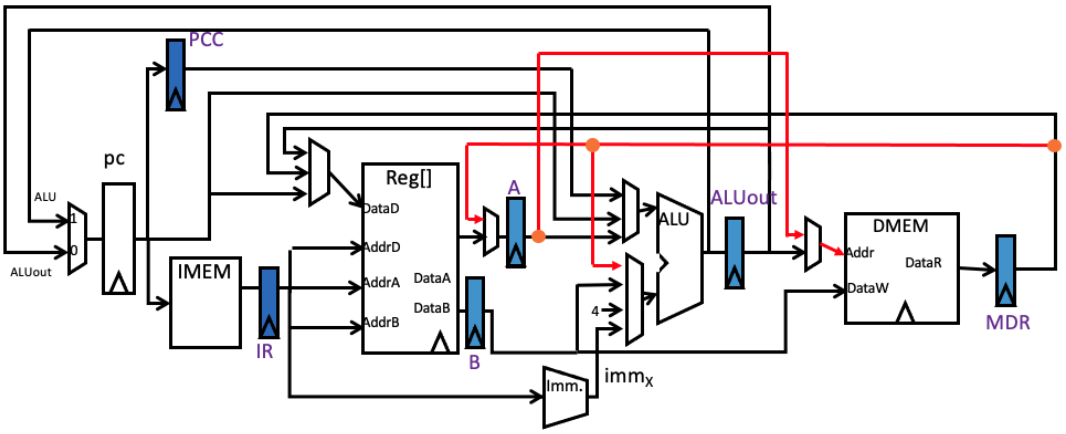
שאלה 7 (5 נקודות)

נתון מעבד מסוג Multi-Cycle RISC-V כפי שנלמד בקורס. סטודנט בקורס מעוניין לממש פקודה שתבצע את הפעולות הבאות:

AddMem rd, rs1, imm

Reg[rd] = Mem[Reg[rs1]] + Mem[Reg[rs1]+imm]

לצורך ביצוע הפקודה, הסטודנט שינה את ה- Datapath של המעבד באופן הבא:



שימו לב לשינויים המודגשים באדום:

1. נוסף בורר הכניסה לרגיסטר A, המקבל בנוסף את המידע שיוצא מרגיסטר MDR.
2. נוסף בורר בכניסת ה-addr של ה- DMEM, המקבל בנוסף את יציאת רגיסטר A.
3. נוספה כניסה לבורר שמחובר לקלט התחתון של ה- ALU. כניסה זו מקבלת את פלט ה- MDR.

מה מספר המחזורים המינימלי הדרושים על מנת להריץ את הפקודה על המעבד?

א. 3
ב. 4
ג. 5
ד. 6
ה. התשובות א' – ד' אינן נכונות.

פתרון:

תשובה ד'.

שני המחזורים הראשונים יבצעו Fetch, Decode ללא שינוי.

במחזור השלישי רגיסטר rs1 יעבור לכניסת ה- addr של ה- DMEM, ויקרא את תוכן הזיכרון שישמר ברגיסטר MDR. במקביל יחושב Reg[rs1]+imm ב- ALU והמידע ישמר ב- ALUout.

במחזור הרביעי המידע שנקרא מהזיכרון Mem[Reg[rs1]] ישמר ברגיסטר A, ובמקביל יקרא המידע Mem[Reg[rs1+imm]] וישמר ברגיסטר MDR. כלומר בסוף מחזור זה יהיה לנו את המידע הנדרש משתי כתובות הזיכרון.

במחזור החמישי נחשב את סכום הערכים שמגיעים מ- MDR, ומ- A ונשמור ב- ALUout.

במחזור השישי נשמור את המידע ב- rd.

שאלה 8 (10 נקודות)

נתונה טבלה המעברים של מערכת בעלת כניסה אחת X ושתי יציאות המסומנות ב Z.

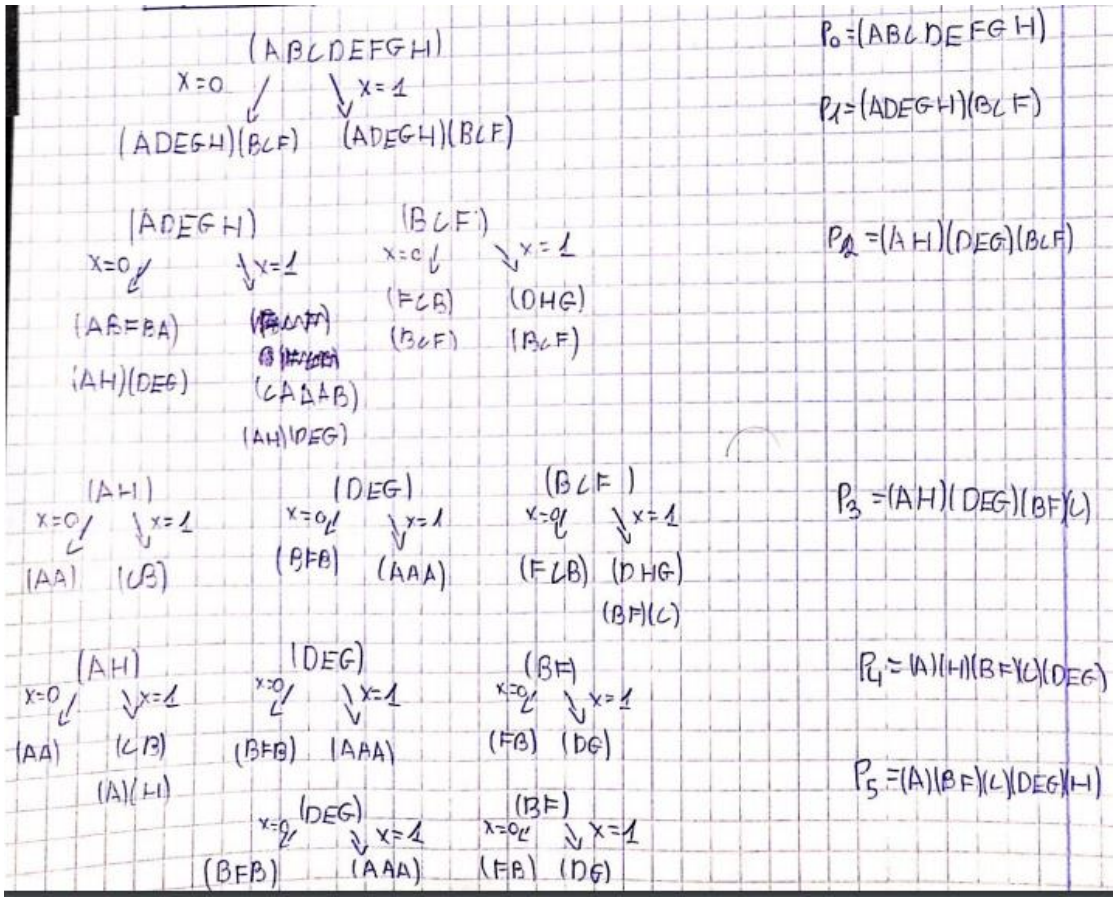
	$X = 0$		$X = 1$	
Present State	*Next State*	*Z*	*Next State*	*Z*
A	*A*	00	*C*	10
B	*F*	01	*D*	11
C	*C*	01	*H*	11
D	*B*	00	*A*	10
E	*F*	00	*A*	10
F	*B*	01	*G*	11
G	*B*	00	*A*	10
H	*A*	00	*B*	10

כמה מחלקות השקילות מתקבלות מצמצום מכונת המצבים הנתונה?

א. 2
ב. 3
ג. 4
ד. 5
ה. 6

פתרון:

תשובה ד'.



שאלה 9 (5 נקודות)

במעבד Multicycle RISC-V המצורף בנספח הבחינה מעוניינים להוסיף תמיכה בחריגת גישה (קריאה או כתיבה) לכתובת לא חוקית בזיכרון (Instruction / Data), אשר יכולה להתרחש **בכל שימוש של הזיכרון**, ומתגלה בעת הגישה לזיכרון. הבקר ממומש באמצעות Dispatch ROM כפי שנלמד בקורס.
על מנת לתמוך בחריגת הגישה הלא חוקית ניתן לשנות Dispatch קיים או להוסיף חדש בעת הצורך. לא ניתן לשנות את מספרי המצבים.

הערה שנאמרה במבחן: אפשר להניח שיש קו READ ENABLE כלומר, שימוש בזיכרון במצבים Fetch או Memory בלבד.

כמה Dispatch ROM חדשים יש להוסיף?

א. יש צורך להוסיף Dispatch יחיד.
ב. יש צורך להוסיף שני Dispatch.
ג. יש צורך להוסיף שלושה Dispatch.
ד. יש צורך להוסיף ארבעה Dispatch.
ה. אין צורך בהוספת Dispatch נוספים.

פתרון:

תשובה ג'.

יש להוסיף יציאה מכל אחד מהמצבים 0,3,5 עבור מצבים אלו יש להוסיף dispatch חדשים. סה"כ יש להוסיף 3 dispatch חדשים.

שאלה 10 (5 נקודות)

אחת החברות המובילות בייצור מעבדי RISC-V Multi Cycle החליטה להוסיף תמיכה בפקודה חדשה הנקראת *swi*. הפקודה שומרת לתוך הזיכרון את תוצאה המכפלה של רגיסטר וקבוע ונראית כך:

swi rd, imm(rs1)

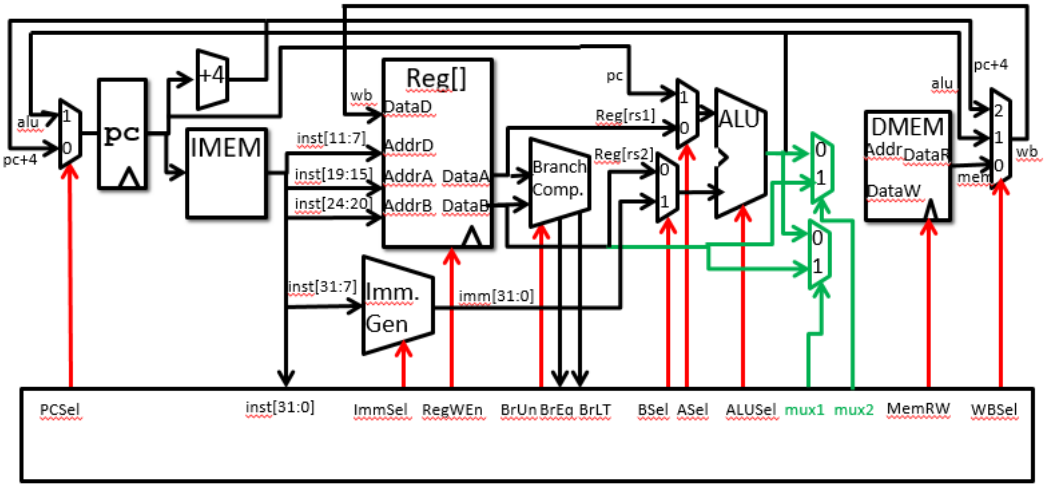
ומבצעת את הפעולה הבאה:

*Mem[Reg[rd]] = Reg[rs1]*imm*

בחרו בתשובה הנכונה ביותר:

א. ניתן לתמוך בפקודה ללא הוספת / שינוי מצבים ושינויים ב *datapath* .
ב. ניתן להוסיף תמיכה בפקודה באמצעות הוספת מצב יחיד למכונת המצבים וללא שינויים ב *datapath*.
ג. ניתן להוסיף תמיכה בפקודה באמצעות הוספת ארבעה מצבים חדשים למכונת המצבים וללא שינויים ב *datapath*.
ד. ניתן להוסיף תמיכה בפקודה באמצעות שינויים ב *datapath* ו/או במכונת המצבים.
ה. לא ניתן להוסיף תמיכה בפקודה כלל.

פתרון:
תשובה ד'.
ניתן להוסיף תמיכה בפקודה אך נשים לב שעלינו להוסיף מצב חדש, להוסיף בוררים למסלול הנתונים (וקווי בקרה מתאימים) ומצב נוסף.
מצורף המעבד והשינויים שבוצעו בו (מודגשים בירוק):



כמו כן עלינו להוסיף תמיכה במכונת המצבים, תמיכה זו יכולה להתווסף על ידי הוספת מצב יחיד שיגיע אחרי מצב 6 (כל החישובים עד לשלב זה זהים) והמצב הבא יהיה מצב שבו נבצע את הכתיבה לזיכרון.

שאלה 11 (12 נקודות)

נתונה הפונקציה הבאה הכתובה בשפת C המקבלת מערך, מבצעת עליו פעולה ושומרת במערך נוסף:

```
void apply(int arr[],int target[], int size){
        target[0] = arr[0];
        for (int i=1; i<size; i++){
                target[i] = target[i-1] + arr[i];
        }
}
```

מהנדס חרוץ מימש את הפונקציה באסמבלי, אך לרוע המזל, קבוצת האקרים הצליחו לפרוץ למחשבי החברה ולמחוק חלק מהקוד. עליכם לעזור למהנדס להשלים את הקוד.

ניתן להניח כי int תופס 4 בתים בזיכרון. כמו כן:

$s1 \leftarrow arr;\ s2 \leftarrow target;\ a0 \leftarrow size$.

<u>סעיף א' (7 נקודות):</u>

השלימו את הקוד:

```
0x1AA0 0000   main :   lw t2, 0(s1)
0x1AA0 0004            sw t2, 0(____)
0x1AA0 0008            addi t0, a0, -1
0x1AA0 000C            addi t1, x0, 0
0x1AA0 0010   loop:    add a1, t1, (____)
0x1AA0 0014            lw t2, 0(____)
0x1AA0 0018            addi t1, ____, 4
0x1AA0 001C            add a1, t1, (____)
0x1AA0 0020            lw t3, ____(____)
0x1AA0 0024            add ____, t2, t3
0x1AA0 0028            add a1, (____), (____)
0x1AA0 002C            sw t2, 0(_____)
0x1AA0 0030            addi t0, t0, -1
0x1AA0 0034            bne t0, x0, loop
0x1AA0 0038   exit:
```

<u>סעיף ב' (5 נקודות):</u>

כעת אנחנו רוצים לבדוק את התוכנית. לשם כך, נתון מעבד pipelined RISC-V בעל מנגנון forwarding מלא ($WB \rightarrow DEC, MEM \rightarrow EXE, WB \rightarrow EXE$) ו- hazard detection unit כפי שנלמדו בכיתה.

המעבד מניח כי פקודות הקפיצה אינן נלקחות, ומבצע flush אחרת. ההחלטה על הקפיצה מתבצעת בשלב ה-MEM.
המהנדס הראשי ביקש מכם לחשב מהו מספר מחזורי השעון שיעברו עד סיום ריצת התוכנית (עד יציאת הפקודה בכתובת 0x1AA00034 משלב הWB בפעם האחרונה).

נתון כי $size = 10$.

מלאו במלבן הבא את מספר מחזורי השעון:

פתרון:

להלן פתרון שתי השאלות, בשורה הראשונה יש לנו load hazard מכיוון שאנחנו מנסים לכתוב ערך שקראנו מהזיכרון בשורה קודמת.
עד תחילת הלולאה: 4 מחזורי שעון ראשונים למילוי הצינור. לאחר מכן, 4 מחזורי שעון עבור 4 פקודות ומחזור שעון נוסף עבור load hazard.

הלולאה עצמה מתבצעת 9 פעמים (size-1) ומכיוון שגם בLW השני בלולאה מתרחש load hazard אנחנו מוסיפים מחזור שעון בכל פעם שהלולאה רצה.

עבור פקודת הקפיצה, אנחנו מניחים שאנחנו לא קופצים אבל אנחנו כן נקפוץ ב-8 הפעמים הראשונות בהן אנחנו מגיעים לפקודת הקפיצה, ובכל טעות כזאת נזרוק 3 מחזורי שעון. בפעם ה-9 שהלולאה רצה, הקפיצה לא נלקחת, ולכן אין צורך לבצע Flush.

בסה"כ בכל ריצה של הלולאה, ירוצו 10 מחזורים עבור הפקודות, מחזור אחד עבור Load Hazard ועוד 3 מחזורים עבור Flush ל- Pipe, חוץ מריצת הלולאה בפעם האחרונה. כלומר, 14 מחזורים בכל ריצה של הלולאה, ו- 11 מחזורים בריצת הלולאה בפעם האחרונה.

סה"כ

$$Cycles = 4 + 5 + 8 * 14 + 1 * 11 = 132$$

0x1AA0 0000 main : lw t2, 0(s1)

=== Load Hazard + 1 ===

```
0x1AA0 0004          sw t2, 0(s2)
0x1AA0 0008          addi t0, a0, -1
0x1AA0 000C          addi t1, x0, 0
0x1AA0 0010   loop:  add a1, t1, s2
0x1AA0 0014          lw t2, 0(a1)
0x1AA0 0018          addi t1, t1, 4
0x1AA0 001C          add a1, t1, s1
0x1AA0 0020          lw t3, 0(a1)
                     === Load Hazard + 1*(size)  ===
0x1AA0 0024          add t2, t2, t3
0x1AA0 0028          add a1, t1, s2
0x1AA0 002C          sw t2, 0(a1)
0x1AA0 0030          addi t0, t0, -1
0x1AA0 0034          bne t0, x0, loop
                     === Assume not taken + 3*(size -1) ===
0x1AA0 0038   exit:
```

שאלה 12 (8 נקודות)

עליכם לתכנן פונקציה המקבלת 4 משתנים בינארים : w,x,y,z המתארים ספרה בבסיס 11, digit=wxyz.

לפונקציה יציאה בעלת ביט יחיד אשר מוציאה 1 עבור ספרות זוגיות, ו 0 אחרת.

א. מלאו את מפת קרנו של הפונקציה (יש לציין don't care במידה ויש):

wx yz	00	01	11	10
00				
01				
11				
10				

ב. מה הביטוי המינימלי כסכום מכפלות של פונקציה זו?

ג. כעת ניתן להביע את הפונקציה באמצעות הפעולות: nand, nor, xor
(בנוסף לפעולות: and, or, not).
האם ניתן לקבל ביטוי מינימלי יותר (פחות משתנים ופחות פעולות) מהתשובה לסעיף ב'?

כן / לא

פתרון:

טבלת אמת:

W	X	Y	Z	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	d
1	1	0	0	d
1	1	0	1	d
1	1	1	0	d
1	1	1	1	d

wx / yz	00	01	11	10
00	1	1	ϕ	1
01	0	0	ϕ	0
11	0	0	ϕ	ϕ
10	1	1	ϕ	1

ב. Z'

לא

שאלה 13 (10 נקודות)

שני מהנדסים מעוניינים לתקשר ביניהם בעזרת מנגנון UART כפי שנלמד בקורס, עם ההבדל היחיד שמשודרות X סיביות **מידע** במקום 8. אין שינוי בסיביות ההתחלה והסיום.

קצב השעון הפנימי של המקלט עובד בקצב 2.5MHz, אך קצב השעון הפנימי של המשדר וקצב השידור המקסימלי הנדרש על קו התקשורת הינו 0.3MHz.

את סיבית ההתחלה מזהה המקלט מיידית.

א. מה יהיה ההיסט המתקבל בקליטת סיבית המידע הראשונה D_0 במקלט?

ב. כעת נתון שהתשובה לסעיף א' הינה α. מה יהיה ההיסט המתקבל בקליטת סיבית המידע D_1 כפונקציה של α?

ג. כעת נתון שהתשובה לסעיף א' הינה α והתשובה לסעיף ב' הינה β.
מצאו חסם הדוק ביותר על המספר המקסימלי של סיביות מידע שניתן לשדר ללא שגיאה כפונקציה של α ו- β.
שימו לב שנרצה לקלוט גם את סיבית הסיום ללא שגיאה.

פתרון:

הערה כללית – הוחלט לא להוריד ניקוד על חוסר ביחידות, או שימוש ביחידות לא נכונות כמו ns, במקום μs.

$$f_R = 2.5MHz \rightarrow T(R_x) = 0.4\mu s$$

$$f_T = 0.3MHz \rightarrow T(T_x) = T_{bit} = 3.333\mu s$$

$$N_{R_x} = \frac{3.333}{0.4} = 8.3333 \rightarrow 8$$

כלומר את סיבית המידע הראשונה נקלוט לאחר 12 מחזורים ותתרחש לאחר $4.8\mu s$, במקום $5\mu s$. נוצר לנו היסט של $0.2\mu s$ בקליטת הסיבית הראשונה וקיבלנו $\alpha = 0.2\mu s$.

מכאן קליטת כל סיבית תתרחש בכל 8 מחזורים, כלומר כל $3.2\mu s$ במקום $3.333\mu s$, ונצבור היסט של $0.133\mu s$ לכל סיבית מידע.

היה אפשר להבין את השאלה בשתי דרכים, וקיבלנו את שתיהן. אפשרות אחת $\beta = 0.133\mu s$, והאפשרות השניה לפיה β הוא ההיסט המתקבל עבור סיבית D_1 וכולל בתוכו את α, ואז התשובה הינה $\beta = 0.133\mu s + \alpha$.

ההיסט עבור סיבית הסיום הינו, כאשר שידרנו X סיביות מידע:

$$\underbrace{\alpha}_{bias\ for\ D_0} + \underbrace{\beta \cdot (X-1)}_{bias\ for\ D_1 \to D_{X-1}} + \underbrace{\beta}_{bias\ for\ stop\ bit}$$

כל עוד הגודל לעיל קטן ממחצית זמן T_{bit} לא תתקבל שגיאה, כלומר:

$$\alpha + \beta \cdot X < 1.666\mu s$$

$$X < \frac{1.666\mu s - \alpha}{\beta}$$

נציב את המספרים בשאלה ונקבל:

$$\underbrace{0.2\mu s}_{bias\ for\ D_0} + \underbrace{0.133\mu s \cdot (X-1)}_{bias\ for\ D_1 \to D_{X-1}} + \underbrace{0.133\mu s}_{bias\ for\ stop\ bit} < 1.666\mu s$$

$$X < 11$$

כלומר ניתן לשדר 10 סיביות מידע עם הנתונים לעיל ללא שגיאה. אם נשדר 11 סיביות, קליטת סיבית הסיום תקרה בדיוק במעבר בין סיביות המידע האחרונה לסיבית הסיום מה שיכול לגרום לשגיאה.

עבור הפרשנות השניה (כמו בסעיף ב') התשובה היא:

$$\alpha + (\beta - \alpha) \cdot X < 1.666\mu s$$

$$X < \frac{1.666\mu s - \alpha}{\beta - \alpha}$$

כמובן שזה לא משנה את מספר הסיביות שיכולות להתקבל, אלא רק את התוצאות של α ו- β.

שאלה 14 (10 נקודות)

סטודנט חרוץ החליט לערוך השוואה בין מעבד RISC-V Single Cycle ובין מעבד RISC-V Multi Cycle. לצורך ההשוואה הוא פנה לחברה מוכרת וביקש מהם לקבל את שני המעבדים. זמן המחזור של מעבד ה Single Cycle RISC-V הוא $T_S = 250ns$. ואילו זמני ההשהיה של השלבים השונים של מעבד ה RISC-V Multi Cycle נתונים בטבלה הבאה:

IF	*ID*	*EXE*	*MEM*	*WB*
$40ns$	$60ns$	$70ns$	$80ns$	$60ns$

הסטודנט החליט להשוות בין זמני הריצה של שני המעבדים השונים, הוא החליט להריץ תוכנית זהה על שני המעבדים. הפקודות שיתבצעו בפועל על המעבד הן:

פקודה	כמות
JAL	12
BEQ	15
SW	5
LW	3
ADD	6

בשלושת הסעיפים הבאים ניתן להוסיף חישוב באורך של שורה אחת בלבד.

א. מהו זמן הריצה הכולל של התוכנית במעבד ה RISC-V Single Cycle?

_____

ב. מהו זמן המחזור של מעבד הRISC-V Multi Cycle?

_____

ג. מהו זמן הריצה הכולל של התוכנית במעבד ה RISC-V Multi Cycle?

_____

פתרון:

נחשב את זמן הריצה של מעבד ה *RISC-V Single Cycle*. אנו יודעים שכל פקודה רצה בדיוק מחזור שעון אחד ולכן זמן הריצה הכולל הוא מסר הפקודות כפול זמן המחזור:

$$T_{tot-Single}\ \#inst * T_s = (12 + 15 + 5 + 3 + 6) * 250ns = 41 * 250ns = 10250ns$$

כעת נחשב את זמן הריצה של מעבד ה *RISC-V Multi Cycle*. אנו יודעים כי כל פקודה רצה רצף אחר של שלבים, נחשב את זמן המחזור של המעבד שנקבע לפי השלב הארוך ביותר:

כלומר $T_{multi} = 80$

$$\#_{JAL} = \#_{BEQ} = IF + ID + EXE = 3$$

$$\#_{addi} = IF + ID + EXE + WB = 4$$

$$\#_{sw} = IF + ID + EXE + MEM = 4$$

$$\#_{lw} = IF + ID + EXE + MEM + WB = 5$$

$$T_{tot-Multi} = \left(12 * \#_{JAL} + 15 * \#_{BEQ} + 6 * \#_{addii} + 5 * \#_{sw} + 3 * \#_{lw}\right) * T_{multi} = 140 * 80 = 11200$$

שאלה 15 (10 נקודות)

נתון מעבד מסוג *Pipelined RISC-V*. המעבד בעל יחידת *Hazard Detection Unit* כפי שנלמד בכיתה ובעל מנגנון forwarding מלא ($WB \rightarrow DEC, MEM \rightarrow EXE, WB \rightarrow EXE$)

יחידת ה-*Branch comparator* במעבד זה ממוקמת בשלב ה-*DEC*. בכדי לתמוך בשינוי זה, מתכנני המעבד הוסיפו *Adder* בשלב ה-*Dec* אשר מסייע לחישוב כתובת הקפיצה במידה ויש בכך צורך. המעבד מניח תמיד כי הקפיצה לא נלקחת ומבצע *flush* במידה ומתברר כי יש לבצע את פקודת הקפיצה.

נתון קוד האסמבלי הבא:

הניחו כי בתוכנית לא מתרחשות חריגות.

	$0x00001000$	$lw\ \ t2, 0(s0)$
	$0x00001004$	$addi\ t2, x0, 5$
	$0x00001008$	$lw\ \ s3, 0(s1)$
	$0x0000100C$	$add\ \ s3, s3, s3$
	$0x00001010$	$add\ \ t1, x0, x0$
	$0x00001014$	$sub\ \ s4, s3, s4$
$Loop$:	$0x00001018$	$lw\ \ s1, 0(s0)$
	$0x0000101C$	$addi\ \ s2, s1, 1$
	$0x00001020$	$addi\ t2, t2, -1$
	$0x00001024$	$addi\ \ s3, s3, 1$
	$0x00001028$	$mul\ \ s2, s2, s3$
	$0x0000102C$	$bne\ t1, t2, Loop$
	$0x00001030$	NOP
	$0x00001034$	NOP
	$0x00001038$	NOP

(המשך בעמ' הבא)

א. ציינו את מספר מקרי ה-*fw* מכל סוג בטבלה הבאה:

$WB \rightarrow DEC$	$MEM \rightarrow EXE$	$WB \rightarrow EXE$

ב. מרגע כניסת הפקודה הראשונה לשלב ה-*Fetch*, כמה מחזורי שעון יעברו עד לסיום ריצת התוכנית (כלומר עד לרגע יציאת הפקודה בשורה $0x0000102C$ בפעם האחרונה משלב ה-*WB*)?

יש לפרט את החישוב בקצרה (לא יותר מ-2 שורות).

פתרון:

א. התקבלו שתי אפשרויות בסעיף זה. בראשונה המעקבים בלולאה נספרים פעם אחת (ללא קשר למספר האיטריציות – משמאל ל-/) ובשנייה הם נספרים כל איטרציה (מימין ל-/).

$WB \rightarrow DEC$	$MEM \rightarrow EXE$	$WB \rightarrow EXE$
$2/10$	$1/5$	$3/7$

	$0x00001000$	$lw\ \ t2, 0(s0)$	
	$0x00001004$	$addi\ t2, x0, 5$	
	$0x00001008$	$lw\ \ s3, 0(s1)$	$WB \rightarrow EXE$
		nop	
	$0x0000100C$	$add\ \ s3, s3, s3$	$WB \rightarrow EXE$
	$0x00001010$	$add\ \ t1, x0, x0$	
	$0x00001014$	$sub\ \ s4, s3, s4$	
$Loop$:	$0x00001018$	$lw\ \ s1, 0(s0)$	$WB \rightarrow EXE$
		nop	
	$0x0000101C$	$addi\ \ s2, s1, 1$	$WB \rightarrow DEC$
	$0x00001020$	$addi\ t2, t2, -1$	$WB \rightarrow DEC$
	$0x00001024$	$addi\ \ s3, s3, 1$	$MEM \rightarrow EXE$
	$0x00001028$	$mul\ \ s2, s2, s3$	
	$0x0000102C$	$bne\ t1, t2, Loop$	

ב. בין הפקודה השלישית לרביעית מתרחש *lw hazard* ולכן מוכנסת שם פקודת *NOP* אחת. בתוך הלולאה ישנו *lw hazard* נוסף. הלולאה מתבצעת *5* פעמים. סה"כ ישנם 6 *lw hazard* אשר גורמים להוספה של 6 פקודות. נשים לב כי בכל אחת מאיטרציות הלולאה עלינו לבצע *flush*. בעת ביצוע *flush* אנו זורקים פקודה יחידה, כלומר אנו מוסיפים עוד 4 פקודות *NOP* לתוכנית (הלולאה מתבצעת 5 פעמים, כאשר רק באיטרציה האחרונה הנחת המעבד נכונה). סה"כ ביצענו הוספה של 10 פקודות. בסה"כ מספר הפקודות אשר מתבצעות הוא:

$$6 + 5 \cdot 6 + 10 = 46$$

כלומר לאחר 50 מחזורי שעון התכונית תסיים לרוץ.