



מערכות ספרתיות ומבנה המחשב (044252)

סמסטר אביב תשפ"א

בחינה סופית – מועד ב - פתרון

20 באוקטובר 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 X עמודים כולל עמוד זה (לא רלוונטי עבור קובץ הפתרון) (בדקו בתחילת הבחינה שלא חסרים לכם עמודים). בסה"כ ישנן 16 שאלות: 12 שאלות אמריקאיות, ו-4 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, טלפנית, או אחרת. בפרט, אין להחזיק בטלפון סלולארי.

בהצלחה!



שאלה 1 (5 נקודות):

נתונה מכונת המצבים הבאה:

```
module fsm (  
    output logic [1:0] z,  
    input logic clk,  
    input logic rst,  
    input logic x  
);  
typedef enum {S0, S1, S2, S3} sm_type;  
sm_type current;  
sm_type next;  
  
always_ff @(posedge clk, posedge rst) begin  
    if (rst) begin  
        current <= S0;  
    end  
    else begin  
        current <= next;  
    end  
end  
  
always_comb begin  
    next = current;  
    z = 2'd1;  
    case (current)  
        S0: begin  
            z = 2'b00;  
            if (x == 1'b0) begin  
                next = S1;  
            end  
        end  
        S1: begin  
            if (x == 1'b1) begin  
                next = S3;  
            end  
            else begin  
                next = S2;  
            end  
        end  
        S2: begin  
            z = 2'b10;  
            if (x == 1'b1) begin  
                next = S0;  
            end  
        end  
        S3: begin  
            z = 2'h3;  
            if (x == 1'b0) begin  
                next = S1;  
            end  
        end  
    end  
endcase  
end  
endmodule
```



לאחר reset, הכניסה x מקבלת את סדרת הערכים הבאה בכל עליית שעון (משמאל לימין):

101000101

האם מכונה זו היא מכונת Mealy או Moore? ומה יהיה הערך של היציאה z לאחר שהמכונה מגיבה לכל סדרת הערכים של x?

סמנו את התשובה הנכונה ביותר (שימו לב שבשאלה זו ישנן 6 תשובות אפשריות):

א. מכונת Mealy, $z = 2'b00$

ב. מכונת Mealy, $z = 2'b11$

ג. מכונת Mealy, $z = 2'b10$

ד. מכונת Moore, $z = 2'b10$

ה. מכונת Moore, $z = 2'b11$

ו. מכונת Moore, $z = 2'b01$

פתרון:

ניתן לראות שהכניסה X רק משפיעה על המצב הבא, ולא משפיעה ישירות על היציאה, z ולכן זו מכונת Moore.

טבלה של המכונה:

current	Next, x=0	Next, x =1
S0	S1	S0
S1	S2	S3
S2	S2	S0
S3	S1	S3

ולכן לאחר סדרת הכניסות אם עוקבים לפי הטבלה נקבל שהמכונה תהיה במצב S3, כלומר התשובה הנכונה היא ה'.



שאלה 2 (5 נקודות):

סטודנט בקורס קיבל מחברתו לקורס מספר אשר מיוצג בפורמט נקודה צפה לפי הנלמד בכיתה. מתוך סקרנות, הסטודנט החליט לתרגם את המספר לערכו העשרוני ולאחר מכן לבדוק כיצד נראה הייצוג של מספר זה לאחר המרתו לקוד גריי.

המספר שקיבל הסטודנט בפורמט נקודה צפה הוא:

0x41900000

מהו קוד גריי בן 5 סיביות עבור אותו מספר? שימו לב כי ההמרה מתבצעת לאחר תרגום לערך העשרוני.

א. 11001

ב. 10010

ג. 10110

ד. 11011

ה. המספר שקיבל הסטודנט הוא שלילי, ולכן לא ניתן להמירו לקוד גריי.

פתרון:

תשובה ד'.

נבצע המרה של המספר מייצוג נקודה צפה אל ייצוג עשרוני. לאחר מכן נתרגם את המספר אל ייצוג בינארי בכדי שנוכל להמירו לקוד גריי:

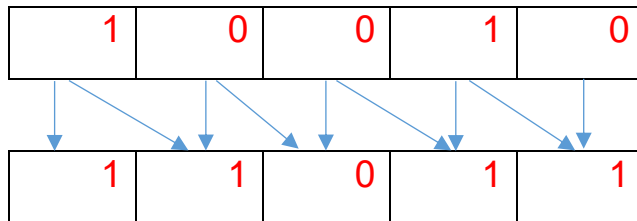
$$0x41900000 \rightarrow (0100\ 0001\ 1001\ 0000\ \dots\ 0000)_{FP} = (18)_{10}$$

קיבלנו כי המספר העשרוני הוא 18. הייצוג הבינארי של 18 הוא:

$$(18)_{10} = (10010)_2$$



כעת נמיר לקוד גריי:



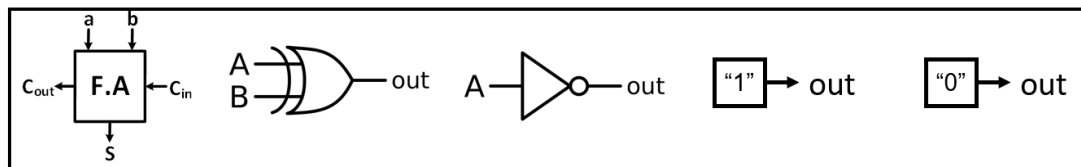
קיבלנו כי הקידוד של המספר לקוד גריי הוא 11011.

שימו לב כי תשובה ה' לא נכונה. גם אם המספר הנתון היה שלילי, לא הייתה כל מניעה להמיר את רצף הביטים שמייצג אותו לקוד גריי.



שאלה 3 (5 נקודות):

נתונים הרכיבים הבאים:



הרכיבים הינם: רכיב שמייצר "1" לוגי, רכיב שמייצר "0" לוגי, שער *not*, שער *xor* ורכיב *FA* (full adder).

רוצים לממש רכיב שמחסר בין שתי סיביות.

הרכיב מקבל כקלט שתי הסיביות: I_1, I_2 , מבצע חיסור ביניהן ומוציא כפלט את שתי הסיביות: O_1, O_2 , המייצגות את תוצאת החיסור $I_2 - I_1$ בייצוג משלים ל-2 (סיבית ה- *MSB* הינה O_2).

מהו מספר הרכיבים המינימלי שאנו צריכים כדי לממש את הרכיב?

א. 2

ב. 3

ג. 4

ד. 5

ה. אי אפשר לממש את הרכיב בעזרת הרכיבים הנתונים.



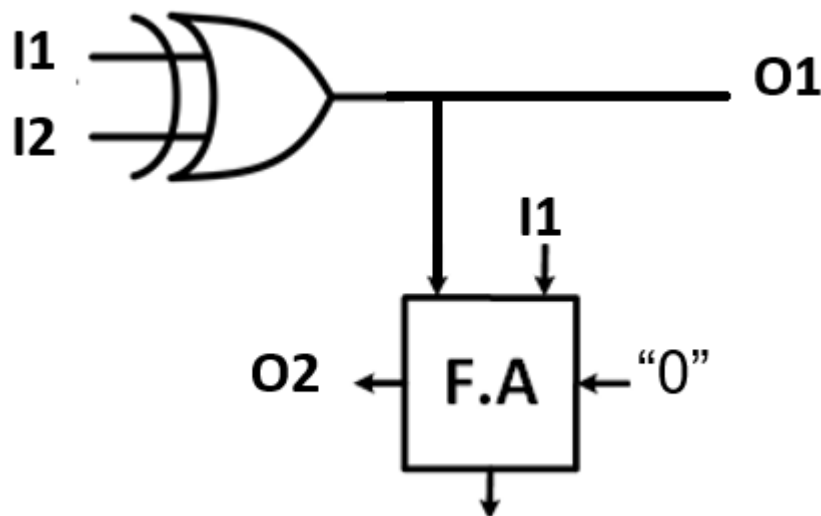
פתרון:

תשובה ב.

טבלת אמת עבור השאלה:

I_2	I_1	O_2	O_1
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

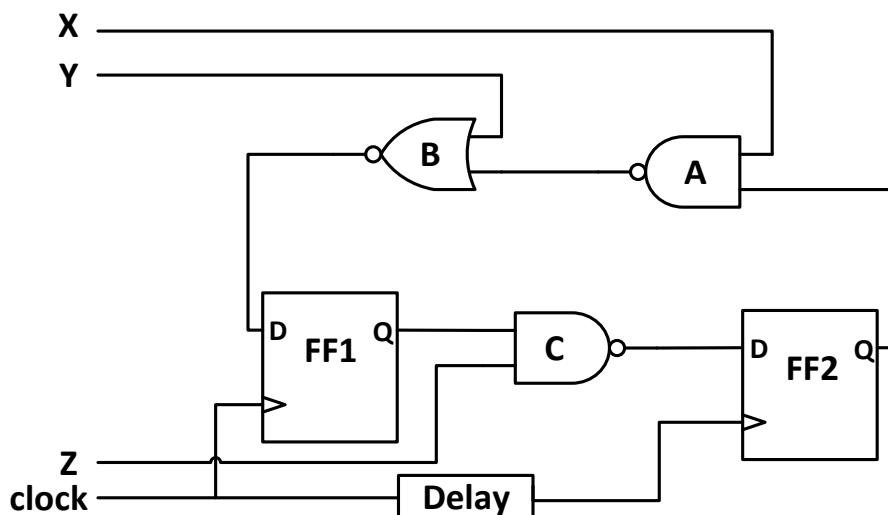
$$O_1 = I_1 \oplus I_2, \quad O_2 = \bar{I}_2 I_1$$



ולכן צריכים 3 רכיבים: מייצר '0', FA ושער XOR.

שאלה 4 (5 נקודות):

עבור המעגל הסינכרוני הבא נתונים הזמנים:



	t_{pCQ}/t_{pd}	t_{ccQ}/t_{cd}	t_{setup}	t_{hold}
Nand	8	2		
Nor	6	2		
FF1/FF2	20	3	7	7

מעוניינים לקבוע את זמן המחזור למינימלי ביותר שיאפשר פעולה תקינה של המעגל. לשם כך הוסיפו השהייה (delay) בכניסה לשעון ב- FF2.

בנוסף נתון שהכניסות מקיימות תנאי setup ו- hold.

מבין האפשרויות הבאות מה ערכה של ההשהייה שתאפשר פעולה תקינה עם מחזור השעון הקצר ביותר?

א. Delay = 3

ב. Delay = -1

ג. Delay = -3

ד. כל ערך של delay שונה מאפס, מאריך את זמן המחזור, לעומת מעגל ללא תוספת delay.

ה. המעגל לא יפעל באופן תקין לכל ערך delay (כולל delay=0)



פתרון:

תשובה ה'

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

$$t_{skew} \leq t_{cd}(FF1) + t_{cd}(C) - t_{hold}(FF2) = 3 + 2 - 7 = -2$$

$$t_{hold}(FF1) \leq t_{cd}(FF2) + t_{cd}(A) + t_{cd}(B) + t_{skew}$$

$$t_{skew} \geq t_{hold}(FF1) - t_{cd}(FF2) - t_{cd}(A) - t_{cd}(B) = 7 - 3 - 2 - 2 = 0$$

אין אף ערך delay עבורו המעגל יעבוד באופן תקין (כולל delay=0)



שאלה 5 (5 נקודות):

סטודנטים בקורס נדרשו לתכנן מערכת עקיבה מסוג Mealy בעלת כניסה יחידה ויציאה יחידה המפיקה $Z=1$ אמ"מ האינדקס של מחזור השעון הנוכחי הוא אי זוגי ומספר האחדים עד כה הוא אי זוגי.

דוגמה:

Cycle	1	2	3	4	5	6	7	8	9
Input	0	0	1	1	0	1	0	0	0
Output	0	0	1	0	0	0	1	0	1

כמה מצבים יהיו במכונה המצומצמת?

יש להניח כי המערכת מתחילה ממצב של רצף אפסים באורך זוגי וממחזור שעון בעל אינדקס אי זוגי.

א. 3

ב. 4

ג. 5

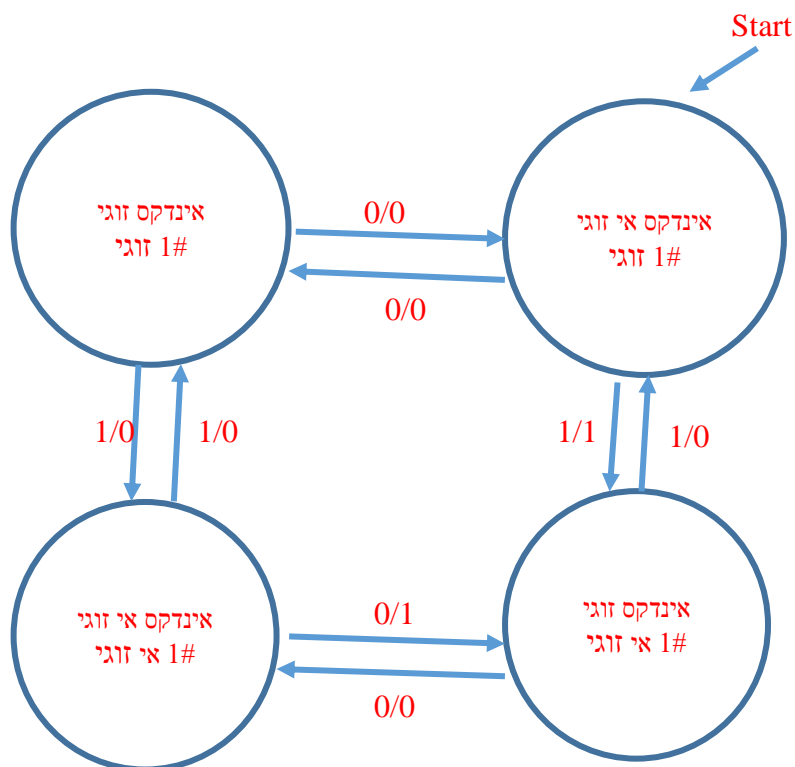
ד. 6

ה. 8



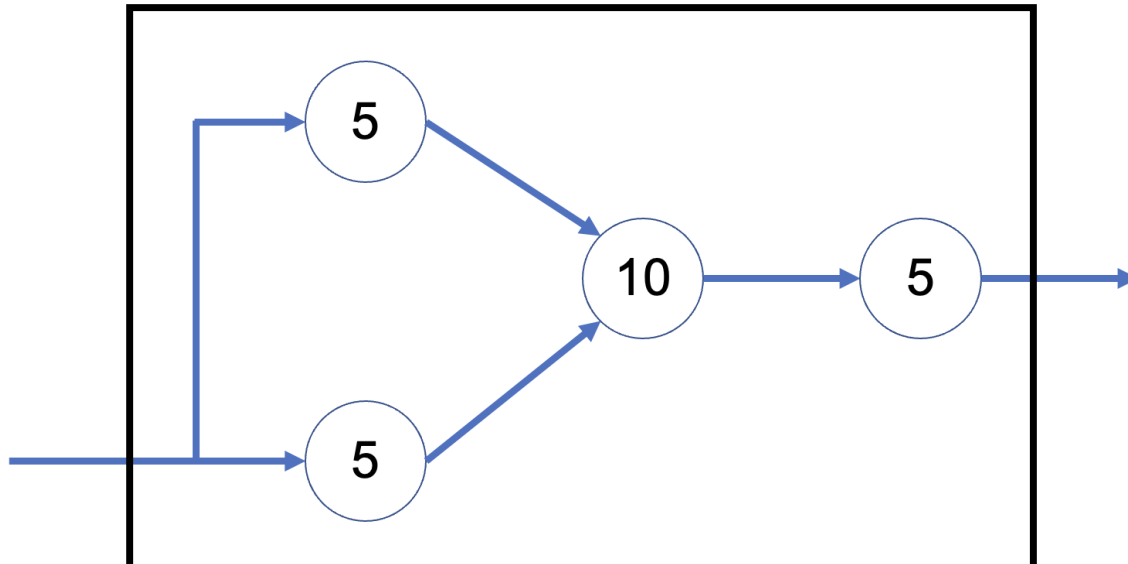
פתרון:

תשובה ב'. להלן שרטוט של המכונה המבוקשת.



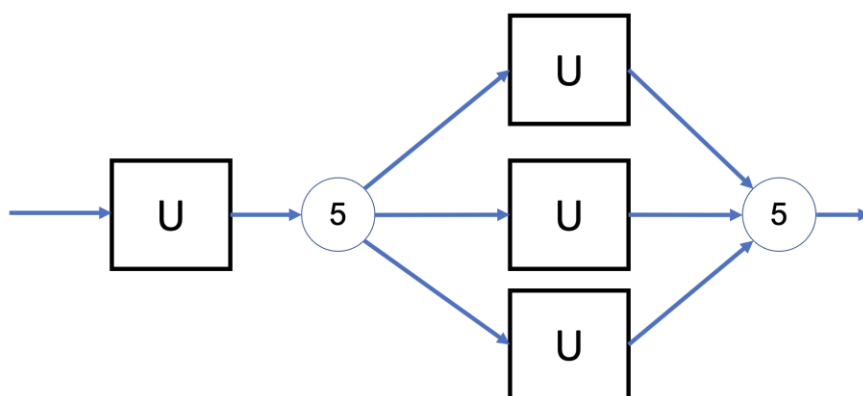
שאלה 6 (5 נקודות):

נתון הרכיב U הבא, בעל כניסה אחת ויציאה אחת.



מהנדס מעוניין לצנר את הרכיב U כך שיקבל throughput (תפוקה) מקסימלית בעדיפות ראשונה, ו-Latency (השהייה) מינימלית בעדיפות שנייה.

מהנדס אחר מעוניין לקנות מספר רכיבי U מצוורים, הוא מודע למימוש הפנימי של הרכיבים, ואין לו יכולת לשנות את מימוש הרכיבים. בהמשך, הוא מעוניין לחבר אותם במעגל לרכיבים צירופיים נוספים באופן הבא:



המהנדס מעוניין לצנר את המעגל לעיל לפי אותן עדיפויות. בנוסף נתון שכלל הרגיסטרים בשאלה זו אידיאליים.



מה מספר הרגיסטרים הכולל (כולל את הרגיסטרים בתוך רכיבי U) הנדרשים לצינור המעגל?

א. 16

ב. 17

ג. 20

ד. 24

ה. לא ניתן לצנר את המעגל כנדרש.

פתרון:

תשובה ב'.

כל רכיב U מצונר ע"י 4 רגיסטרים. בנוסף נצטרך עוד רגיסטר יחיד במוצא.

שימו לב שאין צורך לשים רגיסטר לאחר רכיבי U, שכן הם כבר מצונרים. בנוסף, שימו לב שלאחר רכיב "5" הראשון במעגל, אין צורך להוסיף רגיסטר כי בכניסה לרכיב U יש שימוש ברכיב "5" נוסף, והמסלול בין רגיסטרים נשאר 10, כמו המסלול הארוך ביותר.

לעומת זאת, לפני ה- "5" השני יש רגיסטר מיותר שלא ניתן לשנותו משום שרכיבי U לא ניתנים לשינוי.

סה"כ נזדקק ל- 17 רגיסטרים.



שאלה 7 (5 נקודות):

משדר ומקלט עובדים עפ"י פרוטוקול תקשורת טורית הדומה לזה שנלמד בקורס: תחילה משודרת סיבית התחלה (Start Bit), לאחר מכן משודרות X סיביות מידע, ולבסוף סיבית סיום (Stop Bit).

ידוע כי זמן מחזור השעון של המשדר הוא 60ns , וזמן מחזור השעון של המקלט הוא 125ns . מבחינת המשדר, זמן הסיבית היא 4 מחזורי שעון פנימיים ($N_T = 4$). מבחינת המקלט, זמן הסיבית (Tbit) הוא 2 מחזורי שעון פנימיים ($N_R = 2$).

בנוסף נתון שהמקלט מזהה את ה- start bit מידית.

בהינתן שנדרש שגם סיבית הסיום תידגם בצורה תקינה, מהו ה- X המקסימאלי?

א. 8

ב. 9

ג. 10

ד. 11

ה. 12

פתרון:

תשובה ג'.

נשים לב כי $T_{bit} = 4 \cdot 60 = 240$. עבור X סיביות מידע, זמן תחילת השידור של סיבית הסיום הינו $(X + 1) \cdot T_{bit}$, וסיום שידור סיבית הסיום הינו $(X + 2) \cdot T_{bit}$.

נניח כי המקלט רואה מיידית את סיבית ההתחלה, לכן הדגימה של סיבית הסיום נעשת בזמן $3 \cdot 125 + X \cdot 2 \cdot 125$.

בגלל שנדרש שסיבית הסיום נדגם בצורה תקינה, נדרוש

$$(X + 1) \cdot 240 \leq 375 + 250 \cdot X \leq (X + 1) \cdot 240 + 240$$

$$375 + 250 \cdot X \leq 240 \cdot X + 480$$

$$10 \cdot X \leq 105$$



שאלה 8 (5 נקודות):

לאור התפתחות עולם הקריפטו (מטבעות דיגיטליים), חברה מעוניינת בהוספת יכולת חישוב אריתמטית ליכולות מעבד *Single Cycle RISC-V*.

החישוב מתבצע ע"י חומרה חדשה בשם *hash*.

שיש לה כניסה אחת (I_{in}) ויציאה אחת (O_{out}) ומבצעת $O_{out} = hash(I_{in})$.

הוחלט בחברה לבחון שתי אפשרויות לחישוב פעולת ה-*hash*.

1. פקודת *SHA3* בעלת הפורמט

SHA3 rd, rs1

המבצעת את החישוב:

$$Reg[rd] = hash[Reg[rs1]]$$

2. פקודת *SHA3mem* בעלת אותו פורמט:

SHA3mem rd, rs1

אך מבצעת את החישוב על ערך השמור בזיכרון באופן הבא:

$$Reg[rd] = hash[Mem[Reg[rs1]]]$$

איזו פקודה מבין שתי הפקודות דורשת פחות שינויים במסלול הנתונים ובבקר?

שימו לב, שינוי בקר משמעותו הוספה או הרחבה של סיגנלי בקרה.

הניחו שזמן המחזור של המעבד לא נפגע כתוצאה משימוש ביחידה החדשה.

א. ניתן לממש את שתי הפקודות ללא שינויי בקר, וכמות השינויים

הנדרשת עבור שתי הפקודות במסלול הנתונים זהה.

ב. עבור שתי הפקודות נדרשים שינויי בקר ומסלול הנתונים, וכמות

השינויים הנדרשת עבור שתי הפקודות זהה.

ג. עבור שתי הפקודות נדרשים שינויי בקר ומסלול הנתונים, כמות

השינויים הנדרשת עבור *SHA3* קטנה מ-*SHA3mem*.

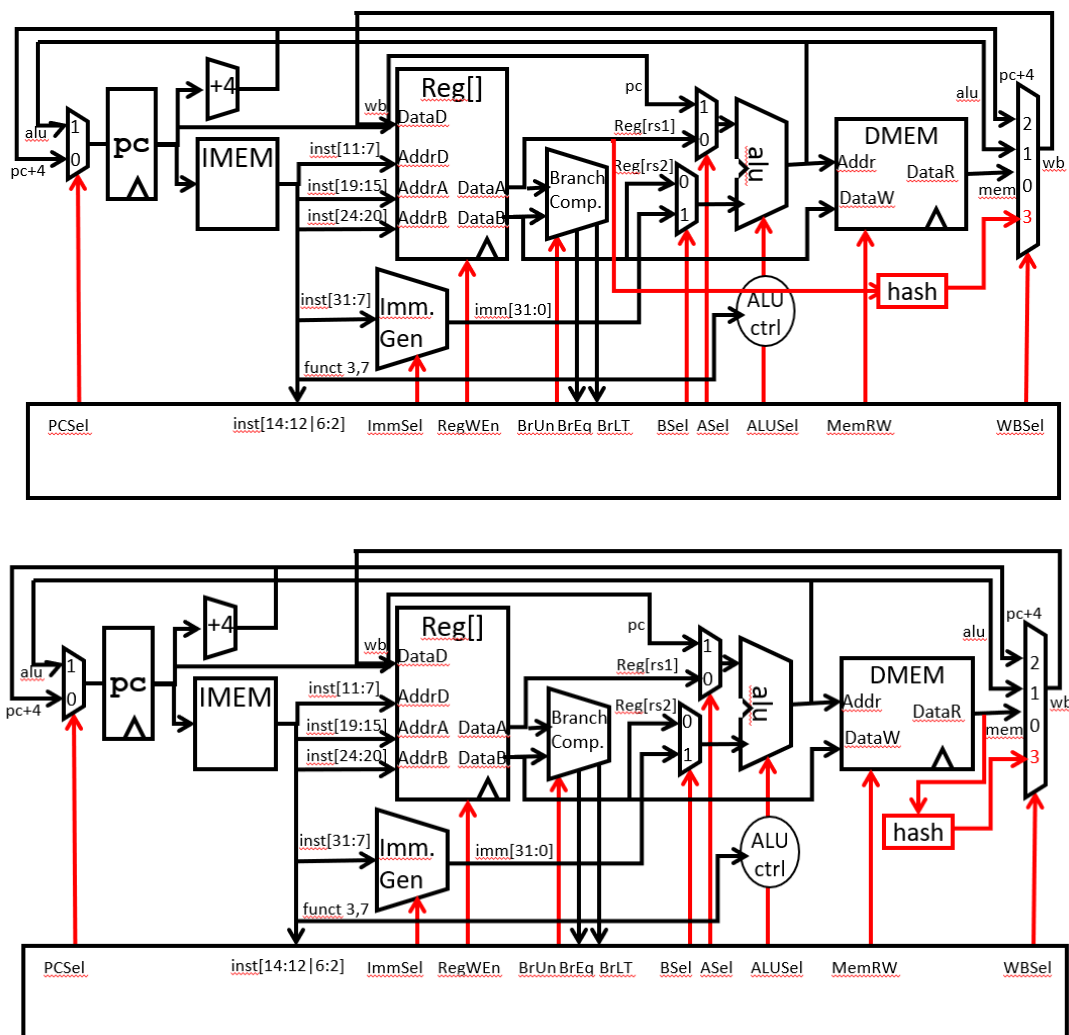
ד. עבור שתי הפקודות נדרשים שינויי בקר ומסלול הנתונים, כמות

השינויים הנדרשת עבור *SHA3* גדולה מ-*SHA3mem*.

ה. לא ניתן לממש את אחת הפקודות.

פתרון:

א. תשובה א'.



במימוש של *SHA3* מחברים את הרכיב החדש כך שיקבל בכניסה שלו את ערך הרגיסטר *rs1*.

במימוש של $SHA3mem$ מחברים הרכיב החדש כך שיקבל בכניסה שלו את ערך הנקרא מהזיכרון. במימוש זה יש צורך לבצע חיבור של $rs1$ עם 0 ב- ALU . שימו לב שאין צורך לעשות זאת, ואפשר לקודד את הפקודה כך שה- $Immediate$ שלה יהיה 0.

בשני המימושים מחברים את היציאה ל-*WBSel*. סלקטור זה במקור היה בעל 3 כניסות מידע, כלומר 2 כניסות בקרה. משמעות הדבר היא שיש כניסת מידע נוספת שאיננה בשימוש. כלומר, נוכל לחבר חוט נוסף לכניסת מידע של הסלקטור מבלי להגדיל את מספר קווי הבקרה שלו.



שאלה 9 (5 נקודות):

נתון מעבד מסוג Multi-Cycle RISC-V כפי שנלמד בקורס. סטודנט רצה לבדוק את יכולותיו והחליט להוסיף את הפקודה UpdMem למעבד.

הפקודה קוראת שני ערכים מהזיכרון, מבצעת כפל ביניהם ושומרת לרגיסטר:

UpdMem rd, rs1, rs2

$\text{Reg}[\text{rd}] = \text{Mem}[\text{Reg}[\text{rs1}]] * \text{Mem}[\text{Reg}[\text{rs2}]]$

הסטודנט החמיץ את תרגול 11, עזרו לסטודנט לקבוע מהם הרכיבים שעליו להוסיף ומהו מספר המחזורים המינימלי לביצוע הפקודה.

ניתן לחוות מחדש את המעבד, להוסיף קבועים, להוסיף בוררים, להגדיל בוררים קיימים ולהוסיף סיגנלי בקרה כנדרש.

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

א. 4

ב. 5

ג. 6

ד. 7

ה. אף תשובה אינה נכונה.



פתרון:

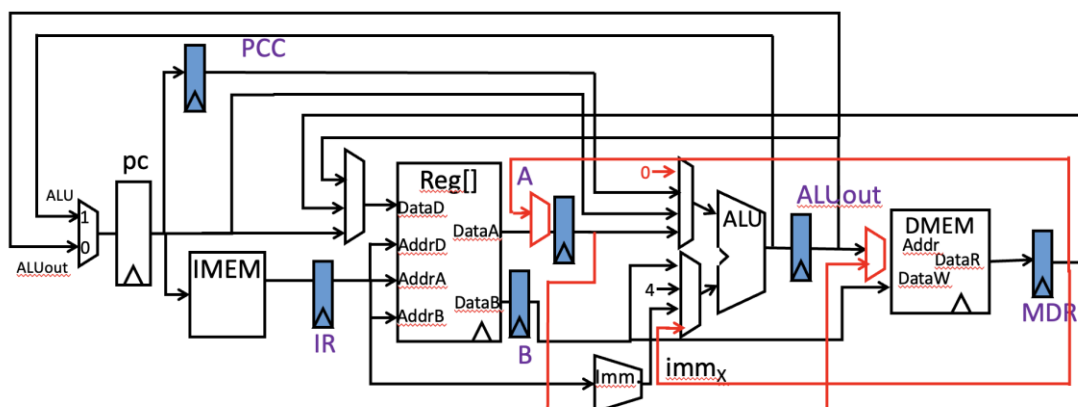
תשובה ג' - מימוש ב 6 מחזורים.

מחזורים ראשונים DECODE, FETCH.

במחזור השלישי קוראים מהזיכרון לפי הכתובת השמורה ב- $rs1$ ושומרים את התוצאה ב- MDR. במקביל מחשבים את הכתובת $rs2+0$, ושומרים את התוצאה ב- ALUout.

במחזור הרביעי קוראים מהזיכרון לפי הכתובת השמורה ב- $rs2$, ושומרים ב- MDR, במקביל את התוכן השמור ב- MDR מעבירים לרגיסטר A. במחזור החמישי מבצעים את פעולת הכפל של התוכן של רגיסטר A ושל תוכן רגיסטר MDR.

במחזור השישי אנו כותבים את התוצאה לרגיסטר rd.





שאלה 10 (5 נקודות):

בהמשך לשאלה הקודמת, הוחלט לממש את הפקודה UpdMem כפסאודו-פקודה על מעבד Multy-Cycle RISC-V, ללא שינויי מעבד.

לצורך המימוש ניתן להשתמש ברגיסטר זמני יחיד t0.

מה מספר המחזורים המינימלי הנדרש למימוש הפקודה?

א. 3

ב. 5

ג. 12

ד. 14

ה. לא ניתן להוסיף את הפקודה.

פתרון:

תשובה ד'.

פסאודו קוד מתאים (נשתמש ב- t0 כרגיסטר זמני עבור הקוד):

lw t0, 0(rs1)

lw rd, 0(rs2)

mul rd, rd, t0

מספר המחזורים הנדרש הינו: $14 = 4 + 5 + 5$.



שאלה 11 (5 נקודות):

נתון RISC-V Multi Cycle כנלמד בקורס. עקב ריבוי גישות למערכים בלולאות, מהנדס התבקש להוסיף את הפקודה `lwr`, בעלת הפורמט הבא:

`lwr rd,rs1(rs2)`

שמבצעת:

$$Reg[rd] = MEM[Reg[rs1] + Reg[rs2]]$$

מהו המספר המינימלי של מצבים ואותות בקרה שיש להוסיף למערכת על מנת לתמוך בפעולה?

עבור הפעולה ניתן לעדכן שורות ולהוסיף מצבים ל-`dispatch rom` 1 ו-2, אבל לא ניתן להוסיף `dispatch rom` חדש. **בפרט ניתן לשנות את מספרי המצבים הקיימים במכונת המצבים שנלמדה בקורס.**

- א. אין צורך להוסיף מצבים או אותות בקרה
- ב. אין צורך להוסיף מצבים, אבל נדרש אות בקרה 1
- ג. נדרש עוד מצב 1, ולא נדרשים אותות בקרה
- ד. נדרש עוד מצב 1, ונדרש אות בקרה 1
- ה. נדרשים עוד 2 מצבים, ואות בקרה 1

פתרון:

תשובה ג'

עבור החיבור ניתן להשתמש באותות הבקרה ממצב 6 ו-2, אך לא ניתן להשתמש במצב 6, ולכן נצטרך מצב חדש. לאחר מכן להשתמש במצבים 3 ו-4. כדי לעשות זאת מבלי להוסיף עוד `dispatch rom`, צריך לשנות את המספר של מצב 2 למספר אחר, ולמספר את המצב החדש במספר 2.



שאלה 12 (5 נקודות):

נתון מעבד Pipelined RISC-V בדומה לנלמד בקורס, בו ההחלטה על קפיצות מתבצעת בשלב ה-Mem.

למעבד מוסיפים יחידת branch prediction. היחידה מקבלת את ה-PC הנוכחי, ואם הפקודה הנוכחית הינה פקודת branch, היחידה מנסה לחזות בשלב ה-Decode האם תתבצע קפיצה או לא.

- אם היחידה חוזה שתתבצע קפיצה: הפקודה הבאה שתטען למעבד הינה הפקודה לאחר הקפיצה והיחידה מבצעת Flush לפקודה שנמצאת כרגע ב-Fetch.
- אם היחידה חוזה שלא תתבצע קפיצה: הפקודה הבאה שתטען הינה הפקודה הבאה בקוד, ולא מתבצע בשלב זה flush לפקודה שבשלב ה-fetch.

שימו לב שהחישוב האמיתי לגבי האם צריכה להתבצע קפיצה או לא מתבצע רק בשלב ה-MEM. אם היחידה טעתה בחיזוי, המעבד מבצע flush ל-pipe. בשאלה זו, אין כלל data hazard בקוד המורץ על המעבד.

משווים מערכת עם המנגנון, לבין מערכת בלעדיו. סמנו את המשפט הנכון:

- א. במערכת עם branch prediction יהיו בהכרח יותר flush.
- ב. מחזור שעון המערכת עם branch prediction בהכרח ארוך יותר ממערכת בלי.
- ג. זמן ביצוע פקודות מסוג R-type יהיה קצר יותר במערכת עם יחידת branch prediction.
- ד. ייתכנו פחות flush במערכת עם branch prediction.
- ה. כל התשובות לא נכונות.



פתרון:

תשובה ד'

א' אינו נכון - שהרי ייתכן שהניחושים מצויינים.

ב' אינו נכון - ייתכן שהגישה לזכרון הינה ארוכה מאד ובכך כל לוגיקה צירופית היא זניחה.

ג' – אין השפעה על פקודות שהם לא branch.

ד' תשובה נכונה - ייתכן שיהיו פחות flush עם הניחוש.



החל מהעמוד הבא מתחיל החלק של
השאלות פתוחות
(שאלות 13 – 16)



שאלה 13 (10 נקודות):

סטודנט מעוניין לגבות את ספר הטלפונים שלו ע"י שליחת מספרי הטלפון לשרת גיבוי. כל מספר טלפון מורכב מ- 4 ספרות עשרוניות, וכל ספרה מיוצגת ע"י קוד BCD.

בשאלה זו, הניחו בכל הסעיפים שכל מספר טלפון מהווה מילת קוד.

סעיף א' –

תחילה הסטודנט מעוניין לגבות את כל הספר הטלפונים שלו שמכיל את כל האפשרויות של 4 הספרות.

מה מרחק הקוד?

סעיף ב' –

נגדיר שני מספרים כ-"שונים" זה מזה, אם כל זוג ספרות במקום ה- i שונות זו מזו (i מייצג את מיקום הספרה). למשל: 1579 ו- 2794 שונים זה מזה מכיוון שהספרה הראשונה בשני המספרים שונה, הספרה השניה שונה וכן הלאה.

סטודנט הציע לשמור רק מספרים שונים זה מזה. מה מרחק הקוד כעת?

סעיף ג' –

בנוסף לסעיף ב', סטודנט נוסף הציע להגדיל את מרחק הקוד על ידי שמירת מספרי טלפון שבהם מופיעות הספרות 1, 2, 4 ו- 8 בלבד (למשל, המספר 2147 אינו יישמר כי מופיעה בו הספרה '7').

מה מרחק הקוד כעת?



פתרון:

כל ספרה נבדלת לכל הפחות בסיבית אחת (למשל הספרה 0 מיוצגת ע"י הסיביות 0000, והספרה 2 מיוצגת ע"י הסיביות 0010). לכן, כשנסתכל על כל המספר כמילת קוד, נקבל שמרחק הקוד הינו 1. למשל שתי מילות הקוד המתוארות ע"י המספר 2000 והמספר 6000 נבדלות בסיבית אחת בלבד.

בסעיף ב', אנחנו יודעים שכל המספרים בספר הטלפונים שונים זה מזה, ולכן בין שני מספרים כלשהם, יש מרחק של לפחות סיבית אחת לכל ספרה, כלומר מרחק הקוד גדל ל- 4.

בסעיף ג', הספרות האפשריות הן 1, 2, 4, ו- 8. כלומר, המספרים הבינאריים הינם 0001, 0010, 0100 ו- 1000. המרחק בין כל זוג ספרות הינו 2, וכל הספרות שונות זו מזו, אז מרחק הקוד גדל ל- 8.



שאלה 14 (10 נקודות):

א. נתונה מפת קרנו של פונקציה. סמנו את הגוררים אשר נדרשים לצורך מימוש מינימלי כסכום מכפלות עבור הפונקציה:

$yz \backslash wx$	00	01	11	10
00				
01			1	1
11		1	1	
10				

ב. האם יתכן כי מתרחש הבהוב סטטי (הזארד) במימוש המצומצם (הניחו כי כל השערים בעלי זמני השהייה זהים)? במידה וכן, הוסיפו גוררים שימנעו את התרחשות ההבהוב.

כן / לא

$yz \backslash wx$	00	01	11	10
00				
01			1	1
11		1	1	
10				



פתרון:

מפת הקרנו של הפונקציה:

$yz \backslash wx$	00	01	11	10
00				
01			1	1
11		1	1	
10				

ייתכן כי מתרחש הבהוב סטטי במידה ומבצעים את המעבר הקלט הבא:

$$(w, x, y, z) = (1, 1, 1, 1) \rightarrow (1, 1, 0, 1)$$

בכדי למנוע הבהוב זה, עלינו להוסיף את המכפלה הכתומה.



שאלה 15 (10 נקודות):

נתון הקוד הבא באסמבלי:

שימו לב שכל הרגיסטרים מאותחלים לערכים לא ידועים כלשהם.

1	addi t0, t0, 0x400	
2	beq t8, t0, far	
3	addi t8, t8, 1	} 2000 ₁₀ addi t8,t8,1 operations
4	addi t8, t8, 1	
5	addi t8, t8, 1	
	
1999	addi t8, t8, 1	
2000	addi t8, t8, 1	
2001	addi t8, t8, 1	
2002	far: addi t8, t8, 1	
2004	#Finish	

מהנדס שם לב שפקודת הקפיצה beq עלולה לא להצליח משום שיעד הקפיצה רחוק מדי. בנוסף המעבד עליו רץ הקוד לא תומך בפקודות מסוג J.

על מנת שהקוד ירוץ כראוי, אילו פקודות יש להוסיף / לשנות בקוד? מלאו בטבלה להלן. בנוסף, ניתן להשתמש באופן חופשי ברגיסטרים שאינם מופיעים בקוד.

דוגמה למילוי הטבלה: נניח שנרצה להוסיף את הפקודה add x0, x0, x0 מיד לאחר פקודה בשורה מס' 5, נמלא את הטבלה באופן הבא:

תיאור פקודה	אחרי / במקום שורה מס'
Add x0, x0, x0	5



הערה: שינוי או הוספת label, נחשבת שינוי פקודה.

רמז: מותר ואף רצוי לשנות את מיקום ה-labels, או להוסיף labels חדשים.

תיאור פקודה	אחרי שורה מס'



פתרון:

ראשית נציין שהסיבה לכך שהקוד יכשל היא שבפורמט הפקודה beq ישנם 12 ביטים לייצג מרחק הקפיצה [12:1], imm[12:1], (מניחים שביט imm[0] שווה לאפס) ולכן אפשר לקפוץ לפקודות שכתובתן במרחק של כ- $\pm 2^{12}$ בתים מהפקודה הנוכחית. כל פקודה מורכבת מ 4 בתים, ולכן אפשר לקפוץ רק כ- $\pm 2^{10}$ (שהם כ- ± 1024 פקודות). זו הסיבה שהקוד לעיל לא יוכל לרוץ כהלכה.

תיאור פקודה	אחרי שורה מס'
addi t10, t8, 0	1
beq t8, t0, mid	1
mid: beq t10, t0, far	כל תשובה בטווח 977-1026

זאת דרך אחת לפתרון:

addi t0, t0, 0x400

addi t10, t8, 0

beq t8, t0, mid

addi t8, t8, 1

addi t8, t8, 1

.....

mid: beq t10, t0, far

addi t8, t8, 1

addi t8, t8, 1

.....

far: addi t8, t8, 1

#Finish

הרבה תשובות יצירתיות התקבלו וגם תשובות שהם לא הכי אופטימליות.



טעויות נפוצות :

הרבה פתרונות השתמשו ב- t_8 גם בחישובי הקפיצות הבאות (אלו שהתווספו לאמצע הקוד) במקום להשתמש ברגיסטר זמני אחר. בהנחה שהקפיצה הראשונה לא נלקחה, (כלומר t_8 לא שווה ל- t_0), הפקודות הבאות משנות את t_8 , ואז יכול להיות שנגיע לאמצע הקוד, ותנאי הקפיצה לסיום כן יתקיים.

כמה פתרונות הזיזו את ה- `#finish` לאמצע התוכנית. זה כמובן לא נכון משום שהתוכנית ממשיכה להתקדם ולא עוצרת אחרי ה- `label` הזה.



שאלה 16 (10 נקודות):

נתון מעבד Pipelined RISC-V בעל התכונות הבאות:

- ללא Hazard Detection Unit ובעל forwarding הבא בלבד:
WB ל- Decode.

- עבור פקודת קפיצה מותנית branch, המעבד מניח כי הקפיצה לא מתרחשת ובמידה וכן, עושה flush כנדרש. ההחלטה על הקפיצה מתבצעת בשלב Execute.

נתון קטע הקוד הבא, כאשר הרגיסטרים מאותחלים לערכים המתאימים להרצת הקוד:

```
main: addi t0, x0, 0
      addi t1, x0, 20
      addi t2, x0, 0
label: lw t3, 0(s0)
      lw t4, 4(s0)
      add t3, t3, t3
      add t4, t4, t4
      sw t3, 0(s0)
      sw t4, 4(s0)
      addi s0, s0, 8
      add t2, t2, t3
      addi t0, t0, 1
      bne t0, t1, label
```




סעיף א':

עליכם להוסיף מספר מינימלי של פקודות NOP על מנת שהתוכנית תרוץ כהלכה (קבלת תוצאות זהות לאלו שהיו מתקבלות במעבד single cycle תקין). סמנו בקוד הבא היכן תוסיפו פקודות סח: הערה: אין לשנות את סדר הפקודות בקוד.

```
main: addi t0, x0, 0
```

```
      addi t1, x0, 20
```

```
      addi t2, x0, 0
```

```
label: lw t3, 0(s0)
```

```
      lw t4, 4(s0)
```

```
      add t3, t3, t3
```

```
      add t4, t4, t4
```

```
      sw t3, 0(s0)
```

```
      sw t4, 4(s0)
```

```
      addi s0, s0, 8
```

```
      add t2, t2, t3
```

```
      addi t0, t0, 1
```

```
      bne t0, t1, label
```



סעיף ב':

כעת ניתן לשנות את מיקומה בקוד של פקודה יחידה כך שתוצאות ההרצה צריכות להיות זהות לאלו של הרצת הקוד המקורי.
מהו מספר פקודות ה-NOP המינימלי שיש להוסיף על מנת שירוצן כהלכה?
לאיזה פקודה שיניתם את המיקום על מנת להגיע למספר ה-nop המינימלי אותו ציינתם בסעיף זה?

מהו מספר ה-nop?

הפקודה שהזזתם:



פתרון:

סעיף א': 4

```
addi t0, x0, 0
addi t1, x0, 20
addi t2, x0, 0
label: lw t3, 0(s0)
      lw t4, 4(s0)
      nop
      add t3, t3, t3
      add t4, t4, t4
      nop
      sw t3, 0(s0)
      sw t4, 4(s0)
      addi s0, s0, 8
      add t2, t2, t3
      addi t0, t0, 1
      nop
      nop
      bne t0, t1, label
```



סעיף ב': 1

```
addi t0, x0, 0
addi t1, x0, 20
addi t2, x0, 0
label: lw t3, 0(s0)
      lw t4, 4(s0)
      addi t0, t0, 1
      add t3, t3, t3
      add t4, t4, t4
      nop
      sw t3, 0(s0)
      sw t4, 4(s0)
      addi s0, s0, 8
      add t2, t2, t3
      bne t0, t1, label
```