



מערכות ספרתיות ומבנה המחשב (044252) סמסטר חורף תשפ"ב

בחינה סופית – מועד ב - פתרון

6 במרץ 2022

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 17 שאלות: 14 שאלות אמריקאיות, ו-3 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, ממחברת 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):

נתון קטע הקוד הבא:

```
module mymodule (
    input logic clk,
    input logic rst,
    input logic [3 : 0] compare,
    output logic out
);

logic [3: 0] ctr;

always_comb begin
    if (compare > ctr)
        out = 1'b1;
    else
        out = 1'b0;
    end

always_ff @(posedge clk) begin
    if (rst) begin
        ctr <= 1'b0;
    end
    else begin
        ctr <= ctr + 1'b1;
    end
end

endmodule
```

אות הכניסה clk הינו שעון בתדר 1Hz.

במהלך המחזור הראשון האות rst שווה לערך 1'b1 (rst = 1'b1) ובמחזורים הבאים שווה לערך 0'b0.

בעלייה השנייה האות compare מקבל את הערך 4'b0100 (compare = 4'b0100).

שימו לב:

1. ctr עובר ל-0 כאשר מוסיפים לו אחד בערכו מקסימלי.
2. out נשאר נמוך כאשר compare לא ידוע.



מה היחס בין הזמן שאות המוצא out שווה ל- '0' לבין הזמן שהוא שווה ל- '1'?
במשך 16 מחזורי שעון לאחר העלייה השנייה? כלומר:

$$\frac{T_{outOFF}}{T_{outOn}} = ?$$

- א. 2
- ב. 3
- ג. 3.33
- ד. 1
- ה. 2.5

פתרון:

תשובה ב'.

המודול הנתון משמש כ- PWM (Pulse Width Modulation). בעזרת האות compare ניתן לקבוע את ה DC (Duty cycle), ובגלל שהאות compare יכול לקבל 16 ערכים (ניתן לשלוט בערך הזה), ה- Duty cycle נקבע לפי ערכו של

$$compare \text{ חלקי } 16. \text{ במילים אחרות: } DC = \frac{compare}{16} = \frac{4}{16} = \frac{1}{4}$$

$$DC = \frac{T_{on}}{T_{total}} = \frac{T_{on}}{T_{on} + T_{OFF}} = \frac{1}{4} \rightarrow \frac{T_{outOFF}}{T_{outOn}} = 3$$

clock	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ctr	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
out	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0



שאלה 2 (5 נקודות):

נתונה המשוואה הבאה:

$$((24)_b - (6)_b)^2 + (14)_b = (150)_b \cdot (2)_b$$

תזכורת:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

המספרים במשוואה הינם בבסיס לא ידוע b .

מהו הבסיס b של המשוואה?

א. 7.

ב. 8.

ג. 9.

ד. יש יותר מבסיס אפשרי אחד למשוואה.

ה. המשוואה לעולם לא מתקיימת.

פתרון:

תשובה ב'.

$$(2b + 4 - 6)^2 + b + 4 = (b^2 + 5b) \cdot 2$$

$$4b^2 - 8b + 4 + b + 4 = 2b^2 + 10b$$

$$2b^2 - 17b + 8 = 0$$

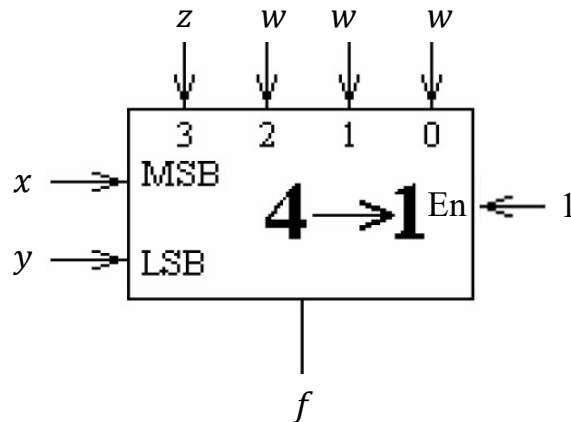
$$b = 8, b = 0.5$$

אך בסיס $b=0.5$ לא ייתכן, שכן יש ספרות גדולות מ-0.5.



שאלה 3 (5 נקודות):

נתון המעגל הבא, הבנוי מבורר $1 \rightarrow 4$:



יציאת המעגל היא הפונקציה $f(w, x, y, z)$.

מהי הפונקציה f כתלות במשתנים w, x, y, z ?

א. $x'w + y'w + xyz$

ב. $x'w + yw'$

ג. $x'w + wz + xyz$

ד. $y'w + yw + xyz$

ה. $wx' + xy'w + xy'z$

פתרון:

תשובה א'.

הפונקציה הממומשת על ידי הבורר היא:

$$\begin{aligned} f(x, y, z, w) &= x'y'w + x'yw + xy'w + xyz \\ &= x'y'w + x'y'w + x'yw + xy'w + xyz \\ &= x'w(y' + y) + y'w(x' + x) + xyz \\ &= x'w + y'w + xyz \end{aligned}$$



וכלן התשובה הנכונה היא א'.

שאלה 4 (5 נקודות):

נתונה פונקציית אסמבלי שמבצעת שימוש ומשנה את ערכי הריגסטרים s_0, t_0 , ועובדת לפי קונבנציית הקריאה לפונקציות שנלמדה בקורס.

נתונות שלוש טענות:

טענה 1: אם הפונקציה לא קוראת לפונקציה אחרת, אין צורך לשמור את הריגיסטר s_0 .

טענה 2: אם הפונקציה לא קוראת לפונקציה אחרת, אין צורך לשמור את הריגיסטר t_0 .

טענה 3: אם הפונקציה קוראת לפונקציה אחרת, חייבים לשמור את רגיסטר ra .

סמנו את התשובה הנכונה ביותר:

א. טענות 1 ו-3 נכונות, טענה 2 אינה נכונה.

ב. טענות 2 ו-3 נכונות, טענה 1 אינה נכונה.

ג. טענה 1 נכונה, שאר הטענות אינן נכונות.

ד. כל הטענות נכונות.

ה. כל הטענות אינן נכונות.

פתרון:

תשובה ב'.

טענה 1 לא נכונה: על מנת להחליט האם נדרש לשמור את s_0 , לא חשוב לנו האם הפונקציה קוראת או לא לפונקציה אחרת, אלא רק אם היא משתמשת בריגיסטר s_0 .

טענה 2 נכונה: הפונקציה אינה קוראת לפונקציה אחרת ולכן אין צורך לשמור את t_0 .



טענה 3 נכונה: הרגיסטר ra חייב להיות שמור מכיוון שהוא מכיל את כתובת
החזרה של הפונקציה הנוכחית.



שאלה 5 (5 נקודות):

נתונות זוג פונקציות `func_A`, `func_B`.

הפונקציות הנתונות קוראות רק לעצמן או אחת לשניה (ולא לפונקציות אחרות).

בכל קריאה לפונקציה נפתחת מסגרת זיכרון. כמה מסגרות זיכרון (frames) ייפתחו מהקריאה הראשונה של פונקציית `main` ל-`func_A`, ועד לחזרתה לפונקציה `main` שקראה לה (לא כולל המסגרת של ה-`main` והמסגרות הקודמות)?

א. 2

ב. כמספר הקריאות לפונקציה `func_A`

ג. כמספר הקריאות לפונקציה `func_B`

ד. כמספר הקריאות לפונקציה שנקראת יותר פעמים בין `func_B` או `func_A`

ה. כמספר הקריאות לפונקציה `func_A` ו-`func_B`

פתרון:

תשובה ה'.

עבור כל קריאה של פונקציה נפתחת מסגרת חדשה. לכן מספר המסגרות הוא כמספר הקריאות ל `func_A` ו `func_B`.



שאלה 6 (5 נקודות):

נתונים שני סוגים של רכיבים צירופיים: A ו-B, בעלי כניסה אחת ויציאה אחת. סטודנט בקורס חיבר באקראי יחידות רבות כאלו בסדר כלשהו (כך שמוצאה של כל יחידה מחובר לכניסה של יחידה אחרת כלשהי).

נתונים זמני ההשהייה של הרכיבים: $T_{pd}(A) = 3ns$, $T_{pd}(B) = 5ns$.

ע"מ לצנר את המערכת על מנת לקבל תפוקה מקסימלית, משתמש הסטודנט ברגיסטרים עבורם:

$$T_{setup} + T_{pCQ} = 2ns$$

עבור כל רגיסטר, ערכי T_{pCQ} ו- T_{setup} שונים (וכל אחד מהם יכול להיות שווה לאפס), הרגיסטרים נבחרו באקראי לצורך צינור המערכת. לא ניתן לדעת מהי חלוקת הזמנים בכל רגיסטר, אך הסכום נשאר קבוע.

לאחר הצינור, הסטודנט מעוניין להפעיל את המערכת. מהו זמן המחזור הקטן ביותר שיכול הסטודנט לבחור על מנת להבטיח שהמערכת תקיים את כללי התזמון?

א. $5ns$

ב. $7ns$

ג. $9ns$

ד. $10ns$

ה. $11ns$

פתרון:

תשובה ג'.

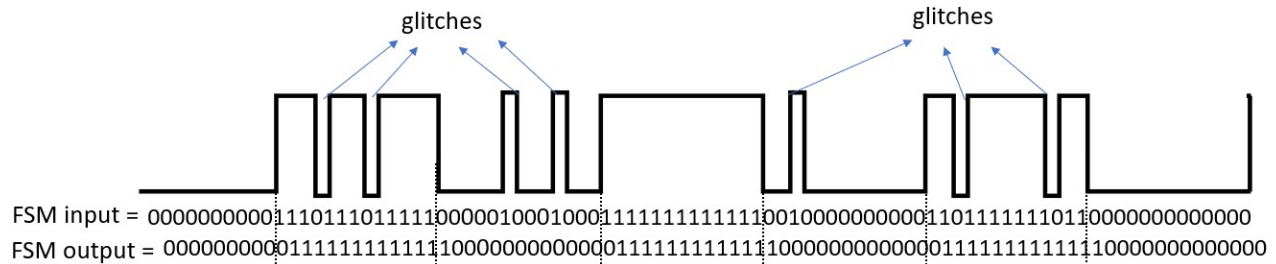
נצנר באמצעות הצבת רגיסטר בין כל רכיב לרכיב. כללי התיזמון נקבעים מרגיסטר לרגיסטר, והארוך ביותר יתקבל כאשר $T_{pCQ} = 2$ ו- $T_{setup} = 2$ עבור הרגיסטרים לפני / אחרי B בהתאמה.

$$T_{min} = T_{pd}(B) + T_{setup} + T_{pCQ} = 9$$



שאלה 7 (5 נקודות):

נתון אות שעון. מהנדס מעוניין להמיר את אות השעון לרצף סיביות, כאשר כל מחזור שעון יתורגם ל-N סיביות (N לא ידוע). לצערו של המהנדס, הוא שם לב שאות השעון מכיל הבהובים (הזארד) סטטיים, כמתואר בציור מטה. נרצה לעזור למהנדס ולתכנן מערכת MOORE ש"מנקה" את ההבהובים.



הערות:

- נתון $N \geq 4$, ושלפחות בשתי הסיביות לפני ואחרי עליות וירידות השעון אין הבהובים.
- בזמן עליות/ירידות השעון, נתון שלא יהיה הבהובים. כלומר אפשר להניח שרצף הסיביות לאחר המעבר מקבל לפחות שני אפסים או שני אחדים לפני שיקרה הבהוב, ושרצף הסיביות לפני המעבר מקבל לפחות שני אפסים או אחדים לפני המעבר.
- לא ייתכנו הבהובים אחד אחרי השני. כלומר, לא ייתכן שיתקבל הרצף 01010, אבל כן ייתכן שיתקבלו הרצפים 00100100 או 11011011.

יש לתכנן מכונת MOORE המקבלת את רצף הסיביות עם הבהובים, ומוציאה רצף סיביות בלי הבהובים.

כמה מצבים קיימים במכונה המצומצמת?

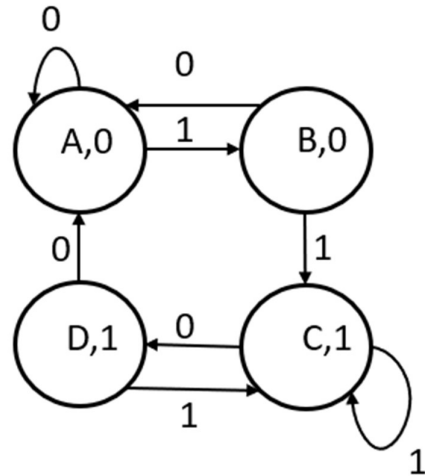
- 3
- 4
- 5
- 6
- 7



פתרון

תשובה ב', 4 מצבים:

אפשר להתחיל בכל אחד מהמצבים B,D, ללא תלות במצב ההתחלתי של השעון. כך אם מתחילים ממצב B למשל והשעון מתחיל ברצף של 1 אז עבור המחזור הראשון הפלט יהיה 0 וזה בסדר כך שאנחנו במערכת Moore.





שאלה 8 (5 נקודות):

להזכירכם, פרוטוקול התקשורת UART שנלמד בקורס:

1 start bit (= 0), 8 bits of data, 1 stop bit (= 1)

הוצעה שיטה לשיפור הפרוטוקול שנלמד בקורס, כך שבמקום לשדר 8 סיביות מידע בכל שידור, תשודרנה N סיביות מידע בכל שידור. בנוסף, בכל שידור נוספו N/4 (N מתחלק ב- 4) סיביות לגילוי שגיאות (ECC).

הפרוטוקול המשופר יראה כך:

1 start bit (= 0), N bits of data, N/4 bits for ECC, 1 stop bit (= 1)

נצילות פרוטוקול התקשורת מוגדרת כמספר סיביות המידע המועברות בכל שידור, מחולקות במספר הסיביות הכללי הדרוש להעברת השידור.

מבין הערכים הבאים, עבור איזה ערך של N הנצילות של הפרוטוקול המשופר יותר טובה מנצילות הפרוטוקול המקורי?

א. $N = 4$

ב. $N = 8$

ג. $N = 12$

ד. $N = 16$

ה. לא קיים N כזה

פתרון

תשובה ה'.

מקורי: שולחים $10 = 1 + 8 + 1$ ביטים מתוכם 8 ביטים של מידע. $8/10$

משופר: שולחים $1 + 1 + N/4 + N$ ביטים מתוכם N ביטים של מידע.

$$N / (N + N/4 + 2) > 8 / 10$$

$$10N > 8N + 2N + 16$$

$$0 > 16$$

וקיבלנו תנאי לא אפשרי על N.



שאלה 9 (5 נקודות):

מהנדס משתמש במעבד SingleCycle RISC-V, ושם לב כי הוא נדרש לבצע העברת ערך בין כתובות זיכרון פעמים רבות.

לטובת פעולה זו מימש את הפקודה החדשה $mva\ rs, rd$, אשר מעבירה את ערך הזיכרון השמור בכתובת $Reg[rs]$, לכתובת הזיכרון השמורה בכתובת $Reg[rd]$.

המהנדס מימש את הפקודה בשתי אופציות שונות: אופציה ראשונה כפסאודו פקודה ואופציה שנייה כפקודה מכונה ע"י שינוי ה-Datapath של המעבד.

את האופציה של מימוש ע"י פסאודו פקודה הוא מימש באופן הבא:

`lw x5, 0(rs)`

`sw x5, 0(rd)`

כאשר הפקודה ממומשת כפקודת מכונה, היא מאריכה את זמן המחזור פי c .

המהנדס מריץ את התוכנית הבאה ומודד זמן ריצה:

```
1      addi x6, x0, N
2  start: mva x10, x11
3      addi x10, x10, 4
4      addi x11, x11, 4
5      addi x6, x6, -1
6      bne x6, x0, start
```

נתון N גדול מאוד. מבין ערכי ה- c הבאים, מהו הערך המקסימלי שעבורו המהנדס יעדיף לממש את הפקודה כפקודת מכונה במקום להשתמש בפסאודו פקודה עבור הקוד הנתון?

א. 1.12

ב. 1.17

ג. 1.22

ד. 1.27

ה. לא קיים c כך שהמימוש כפקודת מכונה עדיף.



פתרון:

תשובה ב'

זמן הריצה עם המימוש ע"י הפסודו פקודה הוא $(1 + 6N)t_{ps}$. זמן הריצה עם המימוש ע"י החומרה הוא $(1 + 5N)t_{hw}$. נדרוש את האי-שוויון הבא:

$$(1 + 6N)t_{ps} > (1 + 5N)t_{hw} \Rightarrow \frac{t_{hw}}{t_{ps}} < \frac{1 + 6N}{1 + 5N} \Rightarrow c < 1.2$$

לכן גם 1.12 וגם 1.17 הן תשובות נכונות, ומהדרישה של מקסימליות נותר רק 1.17.



שאלה 10 (5 נקודות):

מהנדס מקבל מערך של מילים השמור בזיכרון, ומעוניין להכפיל את כל המילים במערך בערך immediate כלשהו.

על מנת לבצע פעולה זו ביעילות, המהנדס מעוניין לממש פקודת מכונה חדשה במעבד MultiCycle בשם `sw_mul`, בעלת הפורמט הבא:

`sw_mul rd, rs1, imm`

ופועלת באופן הבא:

$$Mem[Reg[rd]] = Mem[Reg[rs1]] \cdot imm$$

לצורך מימוש הפקודה החדשה, המהנדס יכול לבצע את השינויים הבאים ב-Datapath של המעבד: להוסיף בוררים חדשים, להרחיב בוררים קיימים, לשנות חיוטים ולהשתמש בקבועים.

שימו לב שלא ניתן לשנות את הרכיבים האחרים, או לשנות את זמן המחזור של המעבד.

הניחו שרכיב ה-ALUctrl מוציא את הערך הדרוש להפעלת ה-ALU.

מה מספר המחזורים המינימלי הנדרש לביצוע הפקודה?

א. 3

ב. 4

ג. 5

ד. 6

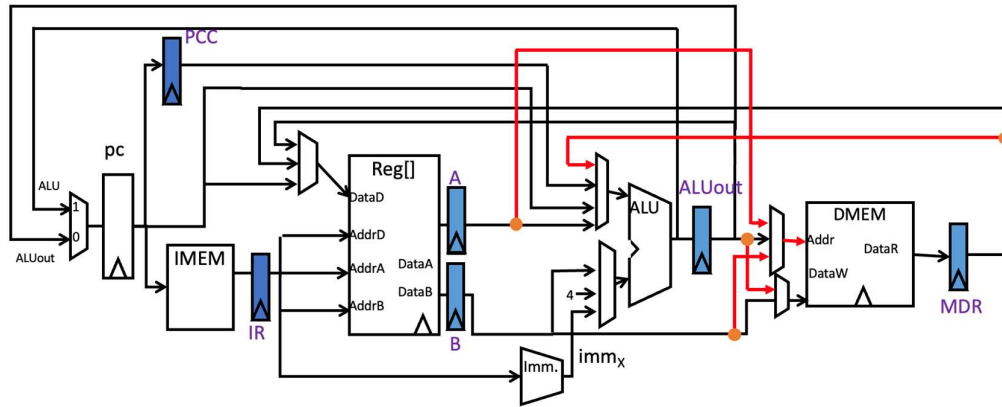
ה. לא ניתן לממש את הפקודה על המעבד הנתון.



פתרון:

תשובה ג'.

אחת האפשרויות לשינוי ה-Datapath:



במחזור הראשון נבצע Fetch ובשני Decode כרגיל. במחזור השלישי נבצע את הקריאה מהזיכרון ונשמור את הערך ב-MDR. במחזור הרביעי נבצע את פעולת הכפל הנדרשת ע"ג ה-ALU עם ה-imm הנתון. לבסוף, במחזור החמישי נבצע כתיבה לזיכרון. סה"כ 5 מחזורים.



שאלה 11 (5 נקודות):

בהמשך לשאלה הקודמת (וללא תלות בתוצאת השאלה הקודמת), המהנדס מעוניין כעת לממש את אותה פקודה על מעבד מסוג Pipeline, כנלמד בקורס. המעבד ממומש כ-Balanced Pipeline, כלומר זמן ההשהייה של כל שלבי ה-pipeline זהה, וזמן המחזור הוא המינימלי המתאים.

עבור אילו שינויים ב-Datapath של המעבד נוכל לממש את הפקודה, **ללא שינוי זמן המחזור?**

סמנו את התשובה הנכונה ביותר:

- א. ניתן לממש את הפקודה ללא שינוי ל-Datapath של המעבד.
- ב. ניתן יהיה לממש את הפקודה רק אם תהיה לנו גישה לרכיב ALU נוסף.
- ג. ניתן יהיה לממש את הפקודה רק אם תהיה לנו גישה לרכיב DMEM חדש במקום ה-DMEM הקיים, שיכול לקבל שתי כתובות – אחת לקריאה ואחת לכתיבה.
- ד. נוכל לממש את הפקודה רק אם תהיה לנו גישה לרכיב ALU נוסף **וגם** לרכיב DMEM חדש במקום ה-DMEM הקיים, שיכול לקבל שתי כתובות – אחת לקריאה ואחת לכתיבה.
- ה. כל התשובות אינן נכונות.

פתרון:

תשובה ה'.

סדר הפעולות אותו אנחנו צריכים לבצע הן: (1) קריאה מהזכרון, (2) הכפלת הערך בקבוע כלשהו ו- (3) כתיבת הערך לזיכרון.

למעשה הדרך היחידה שנוכל לבצע את סדר הפעולות כנדרש היא אם יהיה לנו גם רכיב ALU נוסף שנוכל להשתמש בו בשלב ה-MEM, וגם אם הייתה לנו גישה לזיכרון DMEM חדש שאפשר לקרוא ממנו ולכתוב אליו באותו מחזור. הבעיה היא שנהיה חייבים להגדיל את זמן המחזור כדי לאפשר קריאה-חישוב-כתיבה במהלך מחזור ה-MEM של המעבד.



שאלה 12 (5 נקודות)

נתונים המעבדים הבאים התומכים בפקודות add, addi, lw, sw, jal, beq:

- A. מעבד Pipeline RISC-V בתדר 3F, **ללא** forwarding **וללא** hazard detection unit. המעבד מניח שפקודות branch לא קופצות, ומבצע flush במידה וכן. ההחלטה על קפיצה מתרחשת בשלב השלישי.
- B. מעבד Pipeline RISC-V בתדר 3F עם forwarding בכל השלבים **ועם** hazard detection unit. המעבד מניח שפקודות branch לא קופצות, ומבצע flush במידה וכן. ההחלטה על קפיצה בשלב הרביעי.
- C. מעבד single cycle RISC-V בתדר 5F.
- D. מעבד Multi cycle RISC-V בתדר 8F.

שימו לב שלכל מעבד נתון **תדר** המעבד ולא זמן המחזור.

דרגו את המעבדים לפי הביצועים שניתן להגיע עליהם עבור תוכנית כלשהי שעבורה ניתן להגיע לביצועים המקסימליים (בעלת יותר מ 100 פקודות). כלומר, הביצועים הגבוהים ביותר האפשריים במונחי Instructions-per-Second. שימו לב למעבדים השונים יכולות להיות תוכניות שונות המביאות אותם לביצועים מקסימליים.

הבהרה: הסימון $B < A$ אומר **שתפוקת** מעבד A גבוהה יותר **מתפוקת** מעבד B.

א. $A=B=C=D$

ב. $A < B < C < D$

ג. $C > A = B > D$

ד. $C < D < A < B$

ה. $A = B < C < D$

פתרון:

תשובה ג'.

עבור תוכנית כלשהי הביצועים של מעבד C הם הטובים ביותר. עבור תוכנית ללא תלות מידע וללא קפיצות הביצועים של מעבדים A, B, והכי טובים מבין שאר המעבדים. עבור מעבד D כל פקודה דורשת לפחות 3 מחזורים לכן הביצועים הם הגרועים ביותר.



שאלה 13 (5 נקודות):

במהלך ייצור מעבד MultiCycle RISC-V, קרתה תקלה בה לא יוצרו הרגיסטרים PCC ו-IR כחלק מה-Datapath של המעבד. כלומר, כניסת הרגיסטרים עוברת ישירות למוצא הרגיסטרים.

סמנו את הטענה הנכונה:

- א. אפשר לשנות את מסלול הנתונים כדי שכל הפקודות יעבדו בצורה תקינה (מבלי להוסיף רגיסטרים)
- ב. אפשר להגדיל את זמן המחזור שעון כדי שכל הפקודות יעבדו בצורה תקינה
- ג. רק פקודות מסוג R-type יעבדו בצורה תקינה
- ד. אפשר לשנות את הבקר כדי שכל הפקודות יעבדו בצורה תקינה (ללא שינוי של מסלול הנתונים)
- ה. לא ניתן לשנות את המעבד כך שיעבוד בצורה תקינה.

פתרון:

תשובה ד'.

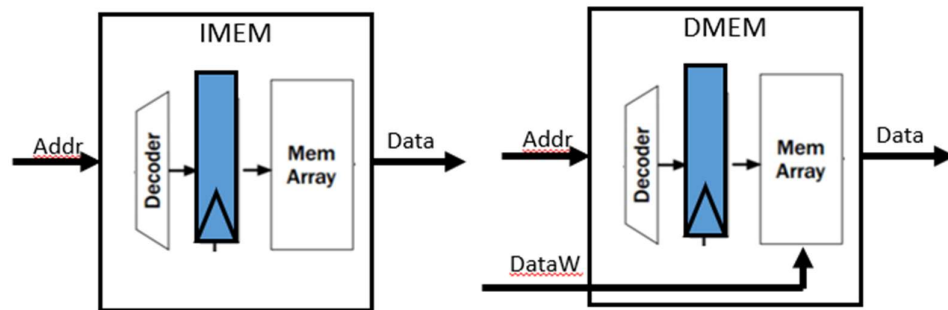
במעבד MULTI CYCLE אנחנו מוסיפים את רגיסטר PCC על מנת לשמור את ה-PC של הפקודה הנוכחית, שכן חישוב $PC+4$ נעשה במחזור הראשון (FETCH), ומוסיפים את רגיסטר IR על מנת לשמור את סיביות הפקודה הנוכחית.

לכן אף פקודה לא תעבוד בצורה תקינה מכיוון שלכל פקודה אנחנו זקוקים לסיביות הפקודה ששמורים ברגיסטר IR (שיתעדכן עם העדכון של PC).

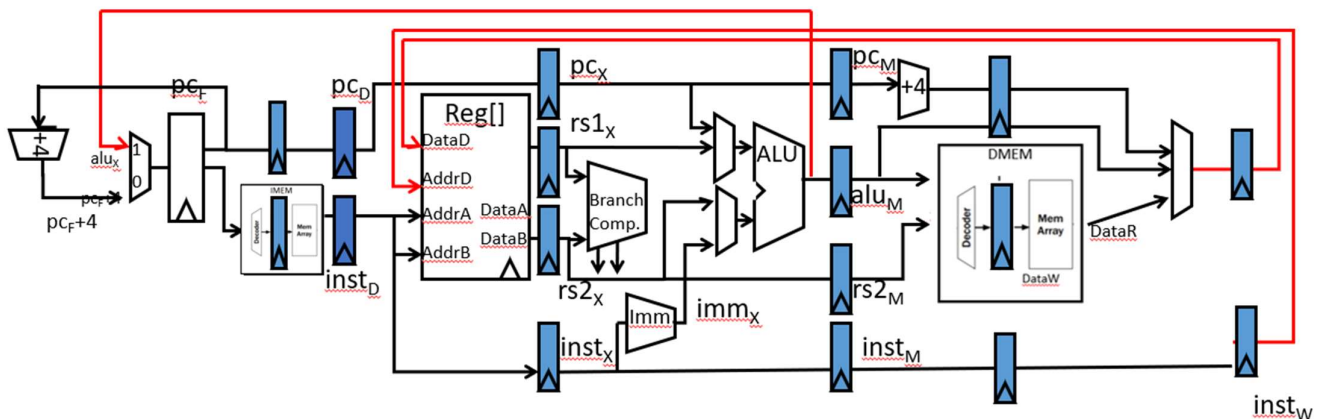
אם נשנה את הבקר כך שלא יעשה את החישוב של $PC+4$ במחזור הראשון, ויעשה אותו במחזור האחרון או במחזור נוסף (שנוסיף אחרי המחזור האחרון) אז נוכל לפתור את הבעיה. לכן התשובה הנכונה היא ד'.

שאלה 14 (5 נקודות):

מהנדסת שעובדת עם מעבד מסוג Pipelined RISC-V, מצאה כי שלבי ה-Fetch ו-Memory בעלי ההשהייה הארוכה ביותר, והחליטה לפצל אותם כך שרכיבי הזיכרון IMEM ו-DMEM, מפוצלים לשני שלבים, הכנת הכתובת הנדרשת בשלב הראשון, וביצוע קריאה או כתיבה בשלב השני.



הפתרון המוצע מגדיל את מספר השלבים במעבד, כך שכעת במקום 5 שלבים כפי שקיימים במעבד הרגיל, יש 7 שלבים. המעבד החדש יראה כך:



- שינוי זה מקצר את מחזור השעון מ 200ns ל 150ns.
- למעבד (המקורי והחדש) אין מנגנוני Forwarding או Hazard detection unit.
- המהנדסת מריצה קוד, כך ש 50% מהפקודות שלו מכילות data hazard. מספר הפקודות $N \gg 7$.



מהו מספר ה nops שצריך להוסיף בין שני פקודות שמכילות data hazard, ואיזה מהמעבדים המהנדסת תבחר על מנת להריץ את הקוד בזמן הקצר ביותר?

- א. 3 nops, עדיף להריץ על המעבד הישן.
- ב. 3 nops, עדיף להריץ על המעבד החדש.
- ג. 4 nops, עדיף להריץ על המעבד החדש.
- ד. 5 nops, עדיף להריץ על המעבד החדש.
- ה. 6 nops, עדיף להריץ על המעבד הישן.

פתרון:

תשובה ג'.

צריך להוסיף למעבד 4 nops כך ששלב ה WB ושלב ה decode לא יהיו באותו מחזור שעון,

אומנם שבשני המעבדים data hazard גורם ל 600ns של בזבז, אבל המעבד החדש התפוקה שלו גדולה יותר.



החל מהעמוד הבא מתחיל החלק של
השאלות פתוחות
(שאלות 15 – 17)



שאלה 15 (10 נקודות):

נתון קוד Gray שבו כל מילה היא בעלת ח סיביות.

א. כמה מילים שונות קיימות לכל היותר בקוד זה?

עבור סעיפים ב' ו-ג': הוחלט להוסיף סיבית זוגיות לכל מילה בקוד. להזכירכם, סיבית זוגיות שווה ל- 1 אם מספר ה- 1 במילה הינו אי-זוגי, ו-0 אם מספר ה- 1 במילה זוגי.

ב. מהו מרחק הקוד מסעיף א' לאחר הוספת סיבית הזוגיות?

ג. לרשותכן עומדות המילים המקוריות שקודדו לקוד Gray וכמות בלתי מוגבלת של שערי xor. עבור מילה כלשהי הנתונה בקוד Gray, כאשר ידוע מהי המילה התואמת לה בקוד המקורי, מהי כמות שערי ה-xor המינימלית לה נזדקק לצורך חישוב סיבית הזוגיות של המילה בקוד Gray?



פתרון:

- א. בדומה למקרים אחרים בהם נתקלנו בקורס, מספר המילים בקוד היא מספר המילים השונות שאפשר לייצר בעזרת n ביטים, כלומר 2^n .
- ב. קוד gray הוא לפי הגדרתו קוד בעל מרחק קוד של 1, הוספה של סיבית זוגיות תגדיל את המרחק הקוד ל-2.
- ג. מתקיים:

$$\begin{aligned} \text{parity bit} &= g_n \oplus g_{n-1} \dots \oplus g_0 = \\ &= b_n \oplus (b_n \oplus b_{n-1}) \oplus (b_{n-1} \oplus b_{n-2}) \oplus \dots \oplus (b_2 \oplus b_1) \oplus (b_1 \oplus b_0) = \\ &= b_n \oplus b_n \oplus b_{n-1} \oplus b_{n-1} \oplus b_{n-2} \dots \oplus b_2 \oplus b_1 \oplus b_1 \oplus b_0 = b_0 \end{aligned}$$

כלומר אין צורך בשערים לוגיים כלל בכדי לקבל את סיבית הזוגיות.



שאלה 16 (10 נקודות):

סעיף א':

מלאו '1' במקום המתאים, במפת קרנו של הפונקציה:

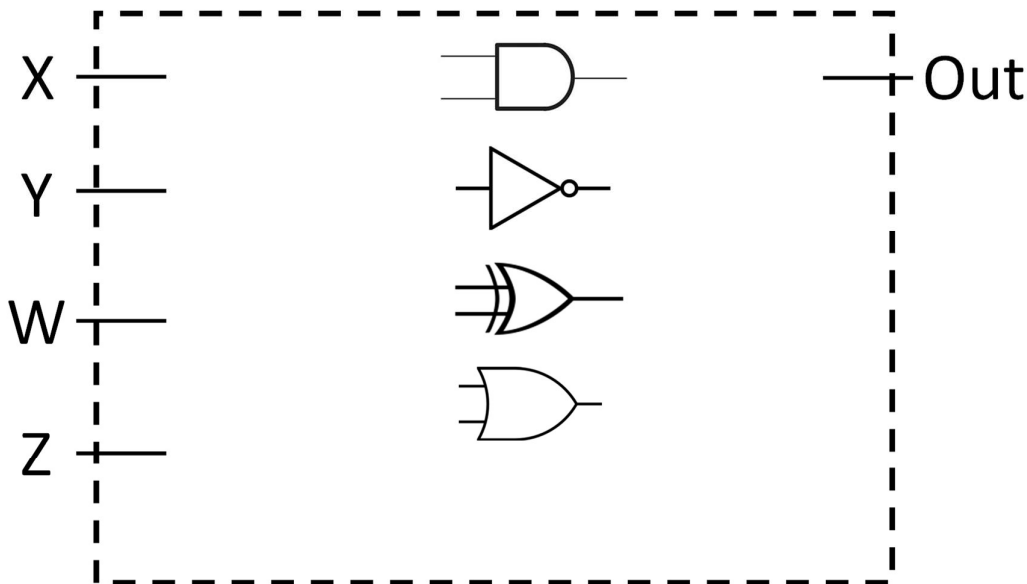
$$h(x, y, w, z) = x \oplus y \oplus w$$

xy \ wz	00	01	11	10
00				
01				
11				
10				

סעיף ב':

לרשותכם השערים הלוגיים Not, XOR, AND, OR , אחד מכל סוג, כמצוייר
בשרטוט. אין חובה להשתמש בכל השערים.

על גבי השרטוט להלן, ציירו את הפונקציה $x \cdot h(x, y, z, w)$.





סעיף ג:

נתונה הפונקציה הבאה:

$$f(x, y, z, t, w) = x \cdot h(y, z, t, w) + \bar{x} \cdot w \cdot h(t, w, y, z)$$

שימו לב לסדר המשתנים בתוך הפונקציה h .

מלאו '1' במקום המתאים, במפת קרנו של הפונקציה f (שימו לב לסדר הקלטים):

xyz \ tw	000	001	011	010	110	111	101	100
00								
01								
11								
10								

פתרון:

סעיף א

נמלא כל משבצת כאשר יש מספר אי זוגי של '1' בוקטור $wxyz$.

xy \ wz	00	01	11	10
00		1		1
01		1		1
11	1		1	
10	1		1	

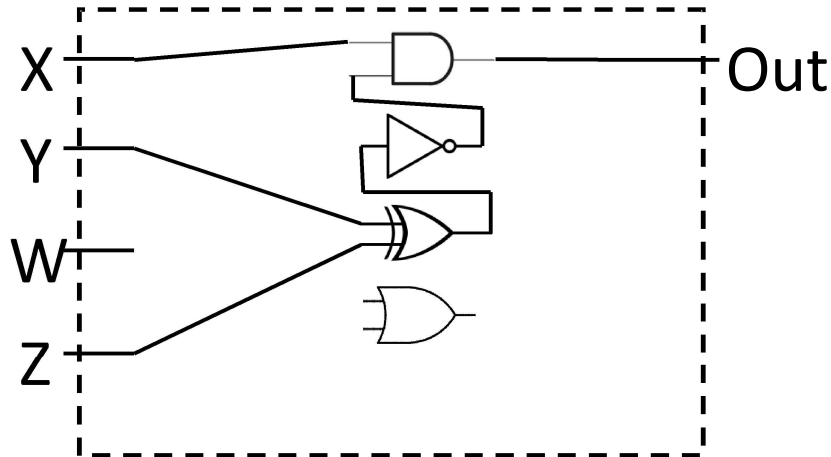
סעיף ב

נשים לב כי:



$$x \cdot x \oplus y \oplus z = x \cdot \overline{y \oplus z}$$

ולכן:



סעיף ג

נשים לב שניתן להפריד את המפת קרנו לשני חצאים, כאשר $x = 0$ וכאשר $x = 1$:

נקבל כי:

$$f(x, y, z, t, w) = x \cdot h(y, z, t, w) + \bar{x} \cdot w \cdot h(t, w, y, z)$$

כאשר $x = 0$, רק הביטוי הימיני משפיע. נעזר בסעיף ב' כדי לחשב:

$$w \cdot h(t, w, y, z) = w \cdot \overline{t \oplus y}$$

כאשר $x = 1$, רק הביטוי השמאלי ישפיע. ולכן נעתיק את המפת קרנו של $x \cdot h$ עד כדי שיקוף.

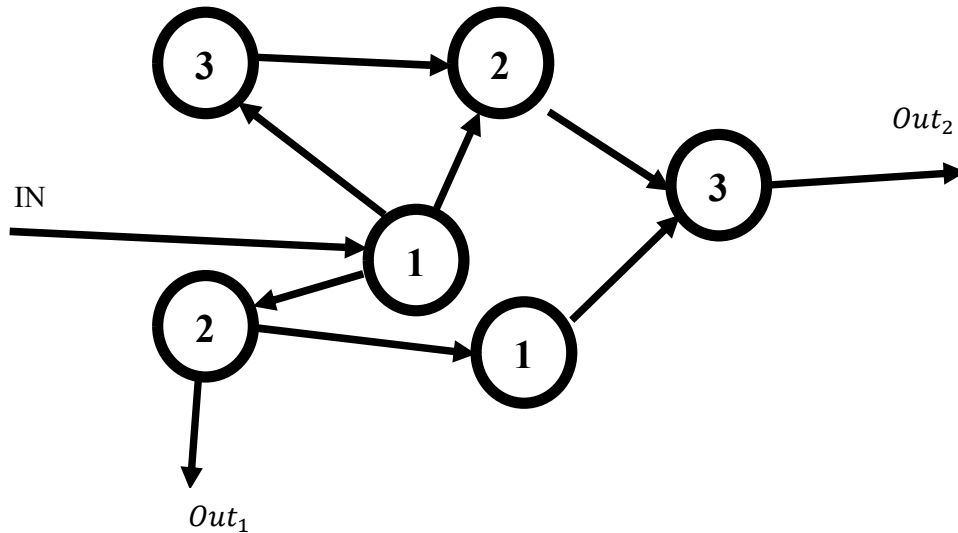
סה"כ נקבל:

xyz \ tw	000	001	011	010	110	111	101	100
00					1		1	
01	1	1			1		1	
11			1	1		1		1
10						1		1



שאלה 17 (10 נקודות):

נתונה המערכת המורכבת מיחידות צירופיות באופן הבא:



המספרים שבתוך הרכיבים מציינים את השהיית הרכיבים ב-ns.

סעיף א':

עליכם לצנר את המערכת בצורה אופטימלית מבחינת latency (הנמוך ביותר).

לרשותכם רגיסטרים אידאליים. מהו ה-latency המינימאלי האפשרי?

פתרון:

השהייה מינימלית מתקבלת כאשר אין צינור.
ההשהייה של המסלול הכי ארוך הוא 9 ns.



סעיף ב':

עבור אותה מערכת, כעת עליכם לצנר את המערכת בצורה אופטימלית מבחינת throughput (הגבוה ביותר) בעדיפות ראשונה, ובמספר רגיסטרים הנמוך ביותר בעדיפות שניה. לרשותכם רגיסטרים בעלי הנתונים:

$$t_{pcq} = 1ns$$

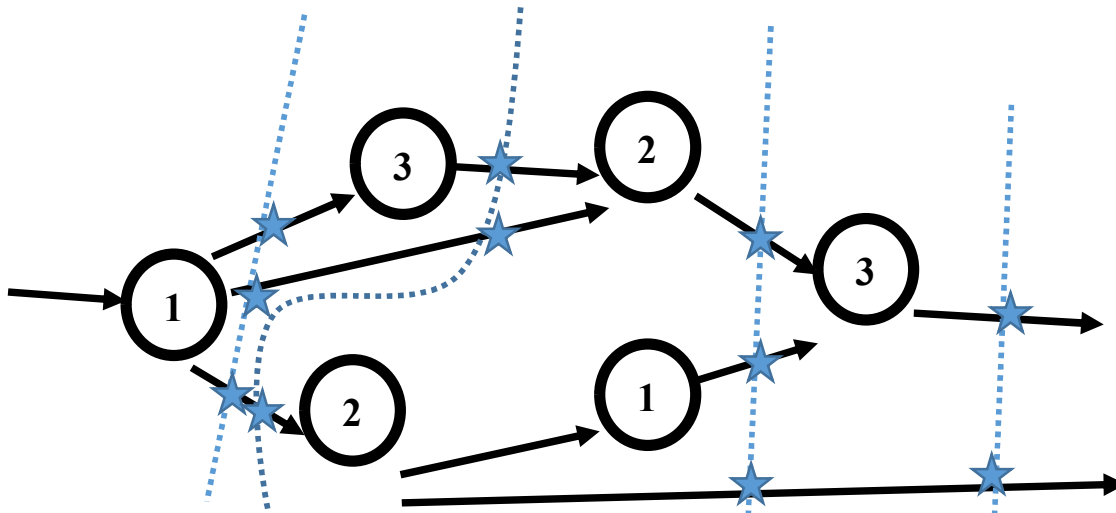
$$t_{setup} = 1ns$$

בכמה רגיסטרים תשתמשו, ומה יהיה ה-throughput (תפוקה)?

מס' רגיסטרים:

תפוקה

פתרון



11 רגיסטרים.

$$t_{clk} = t_{pd}(FF) + t_{pd}(Logic) + t_{setup}(FF) = 1 + 3 + 1 = 5$$

$$throughput = \frac{1}{t_{clk}} = \frac{1}{5}$$



סעיף ג':

כעת ברשותכם שני סוגים של רגיסטרים:

1. רגיסטרים אידאליים בעלות של 10\$ לאחד, עם הנתונים:

$$t_{pcQ}(FF) = 0, t_{setup} = 0$$

2. רגיסטרים מעשיים בעלות של 2\$ לאחד, עם הנתונים:

$$t_{pcQ}(FF) = 1.5, t_{setup} = 1$$

המטרה שלכם היא לקבל **תפוקה מקסימלית** ($max\ throughput$) בעדיפות ראשונה, במחיר הכי נמוך (מספר מינימלי של רגיסטרים אידאליים) בעדיפות שניה, וע"י שימוש במספר מינימלי של רגיסטרים בעדיפות שלישית.

מהו המספר המינימלי של רגיסטרים אידאליים שתשתמשו בהם כדי לקבל תפוקה מקסימלית שהיא 1/3?

מס' רגיסטרים:

פתרון:

7 רגיסטרים

