



מערכות ספרתיות ומבנה המחשב (044252)
סמסטר חורף תש"פ

בחינה סופית – מועד א
3 במרץ 2020
פתרון
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 13 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-SystemVerilog הבא:

```
module my_module (
    input logic clk,
    input logic rst,
    input logic a,
    output logic out
);

typedef enum {first_st, second_st} sm_type;

sm_type current_state;
sm_type next_state;
logic w;

always_ff @(posedge clk, posedge rst)
begin
    if (rst == 1'b1) begin
        current_state <= first_st;
    end
    else begin
        current_state <= next_state;
    end
end

always_comb begin
    next_state = current_state;
    w = 1'b0;
    case (current_state)
        first_st:
            if (a == 1'b1) begin
                next_state = second_st;
                w = 1'b1;
            end
        second_st:
            if (a == 1'b0) begin
                next_state = first_st;
                w = 1'b1;
            end
    endcase
end

assign out = w & (~a);

endmodule
```

ניתן להניח כי הכניסה a מסונכרנת עם עליית השעון וכי כל הרכיבים אידיאליים (וכן כי הכניסות עומדות במשטר הזמנים). מתי הסיגנל out בעל ערך '1'?

- א. בכל זמן שבו הסיגנל a בעל ערך '1'
- ב. תמיד. הסיגנל out הוא הערך הקבוע '1'
- ג. במחזור שעון בו ישנה עליה של הסיגנל a
- ד. במחזור שעון בו ישנה ירידה של הסיגנל a
- ה. במחזור שעון בו ישנו שינוי כלשהו של הסיגנל a (עליה או ירידה)



תשובה: ד'

פתרון:

הקוד הנתון דומה לקוד של EdgeDetector שראינו בסדנה, עם השינויים הבאים:

```
module my_module (
    input logic clk,
    input logic rst,
    input logic a,
    output logic out
);

typedef enum {first_st, second_st} sm_type;

sm_type current_state;
sm_type next_state;
logic w;

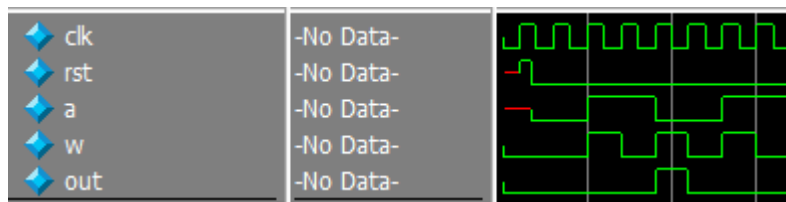
always_ff @(posedge clk, posedge rst)
begin
    if (rst == 1'b1) begin
        current_state <= first_st;
    end
    else begin
        current_state <= next_state;
    end
end

always_comb begin
    next_state = current_state;
    w = 1'b0;
    case (current_state)
        first_st:
            if (a == 1'b1) begin
                next_state = second_st;
                w = 1'b1;
            end
        second_st:
            if (a == 1'b0) begin
                next_state = first_st;
                w = 1'b1;
            end
    endcase
end

assign out = w & (~a);

endmodule
```

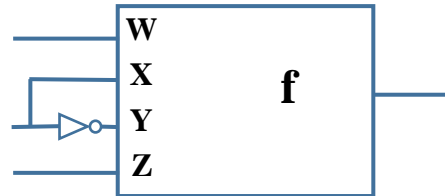
הסיגנל w מזהה עליה וירידה של הסיגנל a משום שכל פעם שעוברים בין שני המצבים, הוא מקבל את הערך '1'. לעומת זאת, הסיגנל out הוא תוצאת ה-AND של w ושל הערך ההפוך (NOT) של a. לכן, רק כאשר a יהיה בעל ערך '0' יש מעבר בין המצבים, שזה שקול לירידה של הסיגנל a, הסיגנל out יקבל את הערך '1'.
להלן תוצאות סימולציה של הקוד:





שאלה 2 (5 נקודות)

נתון הרכיב הצירופי הבא:



ל- $f(W, X, Y, Z)$ ישנן 4 כניסות ויציאה אחת. נתון כי המוצא של f הוא '1' אם"מ בכניסות של הרכיב ישנם לפחות שלושה '0'-ים.

מהו הייצוג המינימלי של f כסכום מכפלות עבור הרכיב הספציפי הנ"ל?

א. $W'XY'Z' + W'X'YZ'$

ב. $W'Z'$

ג. $X'Y'$

ד. $W'Y'Z' + W'YZ'$

ה. $W'X'Z' + W'XZ'$

תשובה: ב'

פתרון:

מפת הקרנו עבור הבעיה הינה:

wx \ yz	00	01	11	10
00	0	1		0
01	0			0
11		0	0	
10	1	0	0	

ולכן הייצוג המינימלי הוא $W'Z'$.

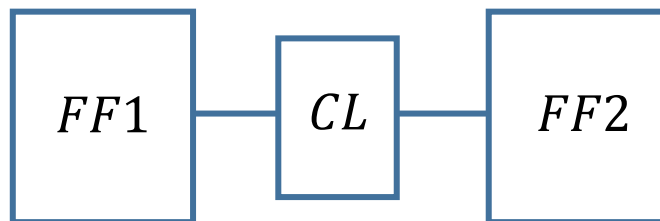
לחלופין ניתן להבין כי X ו- Y תמיד הפוכים, ולכן האפשרות היחידה לקבלת 1 לוגי במוצא היא כאשר $W = Z = 0$ (ואז בהכרח יהיו 3 אפסים), כלומר הפונקציה הינה $W'Z'$.



שאלה 3 (6 נקודות)

תופעת ה-Jitter הינה תופעה אשר מתארת סטייה של זמן עליית השעון בכניסה ל-FF ביחס לזמן העלייה המתוכנן. בהינתן כי אות השעון בכניסה ל-FF סובל מ-Jitter של t_{jitter} יחידות זמן ($t_{jitter} \geq 0$), זמן העלייה של השעון עלול להתרחש t_{jitter} יחידות זמן לפני או אחרי זמן העלייה של אות השעון המקורי. כלומר, ה-Jitter עלול לגרום ל-skew חיובי או שלילי ביחס לעליית השעון המקורית, כאשר בכל עליית שעון כיוון הסטייה הינו אקראי. בנוסף, התופעה עלולה להשפיע על כל אחד מהרכיבים מבוססי השעון במערכת, כאשר בכל עליית שעון כיוון הסטייה לא דווקא זהה עבור כל ה-FF-ים.

עבור המערכת הבאה, מצאו את זמן המחזור המינימלי T_{min} , ואת $t_{hold}(FF2)$ במקרה הגרוע ביותר (המקסימלי) אשר יבטיחו עבודה תקינה של המערכת:



	t_{pd}	t_{cd}	t_{setup}	t_{hold}
FF1	6ns	5ns	3ns	3ns
FF2	7ns	4ns	2ns	?
CL	10ns	7ns		

כאשר $t_{jitter} = 3ns$.

- א. $T_{min} = 24ns$, $t_{hold} = 6ns$
- ב. $T_{min} = 21ns$, $t_{hold} = 9ns$
- ג. $T_{min} = 21ns$, $t_{hold} = 6ns$
- ד. $T_{min} = 24ns$, $t_{hold} = 9ns$
- ה. $T_{min} = 25ns$, $t_{hold} = 9ns$

תשובה: א'

פתרון:

תופעת ה-Jitter היא מעיין skew אשר כיוונו יכול להשתנות. במקרה הגרוע ביותר ה-skew יהיה בכיוונים מנוגדים (הכיוונים במקרה הגרוע משתנים כתלות בתנאי הנבדק). טווח הסטיות האפשרי מופיע בציור העליון, כאשר הקו השחור מציין את זמן עליית השעון המקורי.

עבור תנאי ה-hold (ציור תחתון), במידה ואות השעון יגיע אל FF1 בסטייה של t_{jitter} מוקדם מהמתוכנן ואל FF2 הוא מגיע בסטייה של t_{jitter} מאוחר מהמתוכנן, התנאי אשר נדרש לעמוד בוא הינו:

$$\begin{aligned}
 t_{cd}(FF1) + t_{cd}(CL) &\geq t_{hold}(FF2) + 2t_{jitter} \\
 5 + 7 &\geq t_{hold} + 2 \cdot 3 \\
 5 + 7 - 6 &= 6ns \geq t_{hold}(FF2) \rightarrow t_{hold}(FF2) = 6ns
 \end{aligned}$$

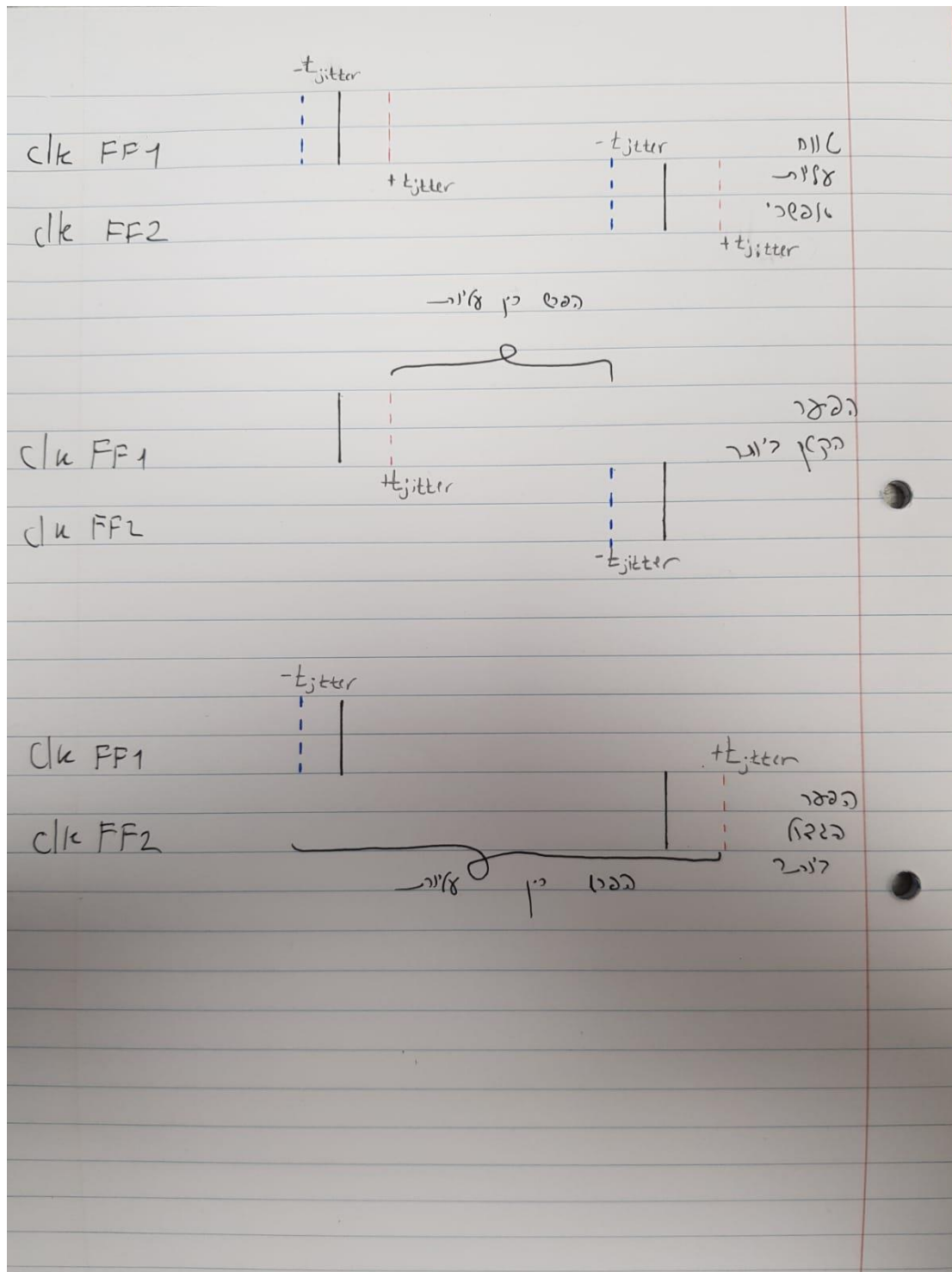


עבור זמן המחזור (ציור אמצעי), המקרה הגרוע ביותר מתרחש כאשר אות השעון אשר מגיע אל FF1 בסטייה של t_{jitter} מאוחר מהמתוכנן ואל FF2 הוא מגיע בסטייה של t_{jitter} מוקדם מהמתוכנן. התנאי הדרוש הינו:

$$t_{pd}(FF1) + t_{pd}(CL1) + t_{setup}(FF2) + 2t_{jitter} \leq T_{min}$$

$$6 + 10 + 2 + 2 \cdot 3 \leq T_{min}$$

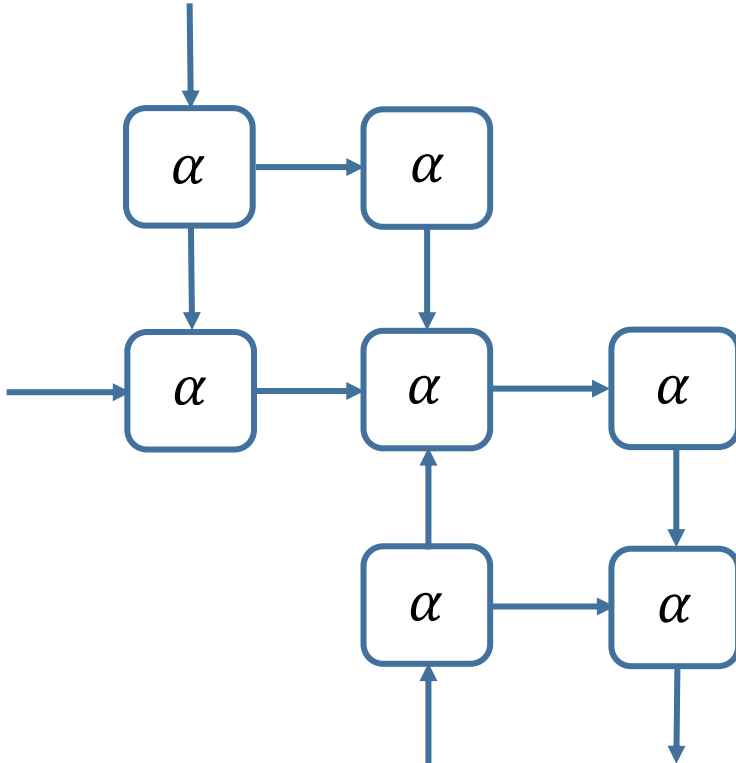
$$24ns \leq T_{min} \rightarrow T_{min} = 24ns$$





שאלה 4 (6 נקודות)

נתונה המערכת הצירופית הבאה, אשר מורכבת מ-7 יחידות צירופיות מסוג α , מכניסות ומיציאות. כל היחידות הצירופיות הן זהות.



עליכן לבצע צינור למערכת הנתונה. שימו לב כי הרגיסטרים אינם אידיאליים, והניחו כי כניסות המעגל עומדות במשטר הזמנים הדינאמי. מהי כמות הרגיסטרים המינימלית אשר דרושה על מנת לקבל תפוקה (Throughput) מקסימלית במערכת ומהי תפוקה זו, כאשר תנאי משטר הזמנים הדינאמי נשמרים?

לרשותכן עומדים גם חוצצים בכמות בלתי מוגבלת (רכיבים צירופיים אשר יציאתם זהה לכניסתם $Buffer(a) = a$).

פרמטרי הרכיבים הינם:

$$t_{pd}(\alpha) = t_{cd}(\alpha) = 2ns$$

$$t_{pd}(buffer) = t_{cd}(buffer) = 3ns$$

$$t_{hold}(Reg) = 2ns, \quad t_{su}(Reg) = t_{pd}(Reg) = t_{cd}(Reg) = 1ns$$

א. 13 רגיסטרים, $Throughput = 0.25$

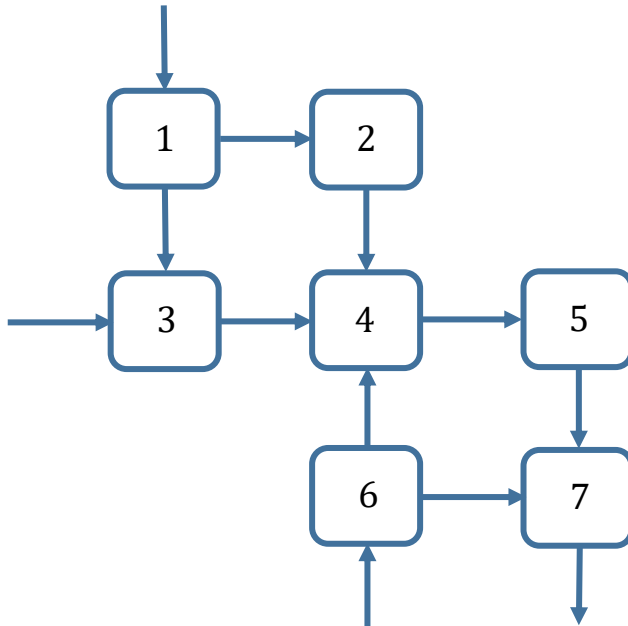
ב. 13 רגיסטרים, $Throughput = 0.2$

ג. 12 רגיסטרים, $Throughput = 0.25$

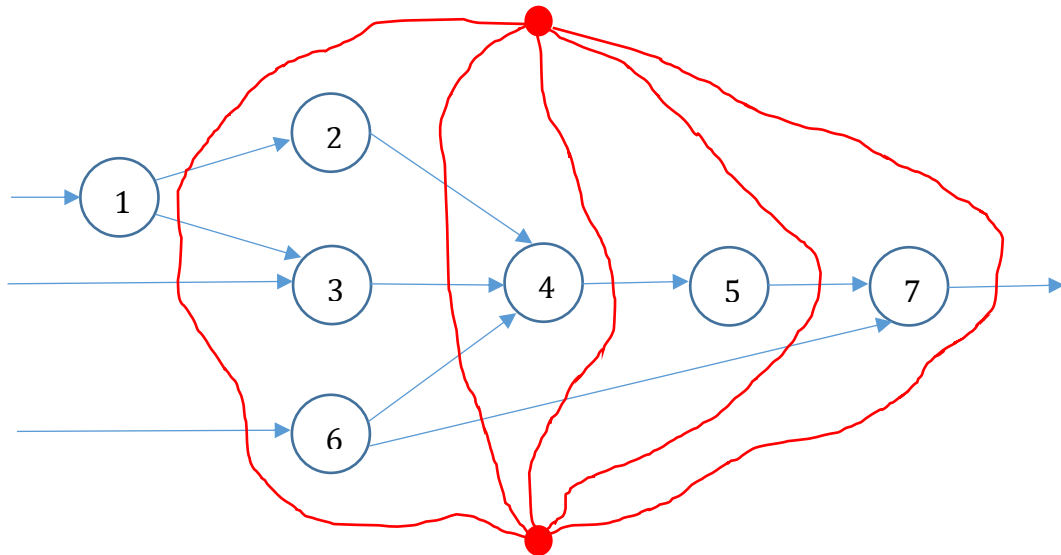
ד. 12 רגיסטרים, $Throughput = 0.2$

ה. 14 רגיסטרים, $Throughput = 0.25$

תשובה: ב'
נמספר את רכיבי המערכת:



נשרטט את המערכת מחדש ונעבוד לפי חוקי הצינור אשר נלמדו בכיתה. מכיוון שכל היחידות החישוביות זהות ונדרשנו לחשב תפוקה מקסימלית, נרצה תחילה לחצוץ בין כל שתי יחידות מסוג α ולבדוק לאחר מכן את תנאי משטר הזמנים.



כלומר, קיבלנו כי מספר הרגיסטרים המינימלי הוא 13. כעת עלינו לוודא עמידה במשטר הזמנים הדינאמי. על פי הנחות השאלה, כניסות המעגל עומדות במשטר הזמנים ולכן אין צורך לבדוק אותן ואת המסלולים אשר עוברים דרך רכיב מספר 1 ומגיעים לדרגת הרגיסטרים הראשונה. נותר לבדוק כי כל המסלולים בין כל שני רגיסטרים עומדים במשטר הזמנים. עבור כל מסלול בין שני רגיסטרים אשר מכיל יחידת α נקבל כי:

$$t_{pd}(Reg) + t_{pd}(\alpha) + t_{su}(Reg) = 1 + 2 + 1 = 4ns \leq T$$

$$t_{cd}(Reg) + t_{cd}(\alpha) = 1 + 2 = 3ns \geq 2ns = t_{hold}(Reg)$$

עבור כל מסלול בין 2 רגיסטרים ללא יחידת α בניהם נקבל:



$$t_{pd}(Reg) + t_{su}(Reg) = 1 + 1 = 2ns \leq T$$

$$t_{cd}(Reg) = 1 = 1ns \not\geq 2ns = t_{hold}(Reg)$$

כלומר תנאי ה- $hold$ מופר. בכדי לתקן בעיה זו נוכל להוסיף חוצץ בכל מסלול בין שני רגיסטרים אשר אינו מכיל יחידת α , ואז נקבל:

$$t_{pd}(Reg) + t_{pd}(buffer) + t_{su}(Reg) = 1 + 3 + 1 = 5ns \leq T$$

$$t_{cd}(Reg) + t_{cd}(buffer) = 1 + 3 = 4ns \geq 2ns = t_{hold}(Reg)$$

ישנם 2 מסלולים כאלו, ולכן נזדקק לשני חוצצים. נשים לב כי הוספת החוצצים גורמת לזמן המחזור לעלות, לכן מסלול זה הוא שמגדיר אותו.

$$Throughput = \frac{1}{T} = \frac{1}{5} = 0.2$$



שאלה 5 (5 נקודות)

משדר ומקלט עובדים ע"י פרוטוקול תקשורת טורית הדומה לזה שנלמד בקורס: תחילה משודרת סיבית התחלה (Start Bit), לאחר מכן משודרות 14 סיביות מידע, ולבסוף סיבית סיום (Stop Bit). ידוע כי זמן מחזור השעון של המשדר הוא 100ns. מבחינת המשדר והמקלט, זמן הסיבית (Tbit) הוא שישה מחזורי שעון פנימיים (N=6). נתון שסיבית הסיום נדגמת בצורה תקינה, בדיוק בסוף שידורה. מהו מחזור השעון של המקלט בהנחה ואנחנו מזחים את Start bit בדיוק ברגע הירידה? יש לעגל את התשובות כלפי מטה.

- א. 101ns
- ב. 102ns
- ג. 103ns
- ד. 104ns
- ה. 105ns

שאלה 6 (5 נקודות)

כל נתוני השאלה הקודמת נותרים זהים עבור שאלה זו. בהינתן שסיבית ההתחלה מתקבלת בדיוק בעליית שעון המקלט (כלומר נגלה את תחילת השידור מחזור שעון מאוחר יותר), מהו המחזור השעון של המקלט? יש לעגל את התשובות כלפי מטה.

- א. 101ns
- ב. 102ns
- ג. 100ns
- ד. 99ns
- ה. 103ns

פתרון:

שאלה 5: תשובה: ג'

$$T_{bit}(T) = 600ns$$

לכן, סיבית הסיום תהיה בין גבולות הזמן:

$$9000ns = 600 \cdot 15 \leq t \leq 16 \cdot 600 = 9600ns$$

דגימה ראשונה של המקלט מתרחשת ב $1.5T_{bit}(R)$ ומשם כל דגימה נוספת לאחר $1T_{bit}(R)$

כלומר הדגימה ה-15 היא בזמן: $1.5T_{bit}(R) + (n-1) \cdot T_{bit}(R) = 15.5T_{bit}(R)$ ומתקיים:

$$15.5T_{bit}(R) = 9600$$

$$T_{bit}(R) = 619.35ns \rightarrow T_{cycle}(R) = 103ns$$

שאלה 6: תשובה: ב'

כעת זמן הדגימה משתנה, הוא מתחיל כבר ב $T_{cycle}(R)$ מאוחר יותר. לכן נקבל:

$$15.5T_{bit}(R) + T_{cycle}(R) = 9600$$

$$94T_{cycle}(R) = 9600ns \rightarrow T_{cycle} = 102ns$$



שאלה 7 (5 נקודות)

נתונים זמני המחזור של כל אחד מהמעבדים השונים:

$$T_{single\ cycle} = 6ns$$

$$T_{multi\ cycle} = 1.25ns$$

$$T_{pipeline} = 1ns$$

במידה ונריץ את פקודת *And* על כל אחת מן הארכיטקטורות, מה הוא יחס הזמנים בין ה-latency של הארכיטקטורות השונות?

א. $T_{single\ cycle} > T_{pipeline} > T_{multi\ cycle}$

ב. $T_{multi\ cycle} > T_{single\ cycle} = T_{pipeline}$

ג. $T_{multi\ cycle} > T_{single\ cycle} > T_{pipeline}$

ד. $T_{single\ cycle} > T_{multi\ cycle} > T_{pipeline}$

ה. $T_{single\ cycle} > T_{multi\ cycle} = T_{pipeline}$

תשובה: ה'

פתרון:

נחשב את הזמן הדרוש עבור כל אחת מן הארכיטקטורות:

$$Latency_{single\ cycle} = T_{single\ cycle} = 6ns$$

$$Latency_{multi\ cycle} = T_{multi\ cycle} \cdot \#cycles(R - type) = 1.25 \cdot 4 = 5ns$$

$$Latency_{pipeline} = T_{pipeline} \cdot 5 = 1 \cdot 5 = 5ns$$

$$T_{single\ cycle} > T_{multi\ cycle} = T_{pipeline}$$

ולכן מתקיים

שאלה 8 (6 נקודות)

נתון מעבד pipedRISCv בעל כל יחידות Forwarding שראינו בתרגול.

נניח כי: $Mem[a1 + 4] = 0x2$ וכי: $Mem[a0] = [0x2, 0x3, 0x4, 0x5]$

מריצים עליו את הקוד הבא:

(הניחו כי פקודת *lwm* היא פקודת R רגילה והחישוב מתבצע ביחידת הALU)

```
(1)add t3, x0, a0
(2)addi t2, x0, 1
(3)lw t0, 4(a1)
(4)label: lw t1, 0(t3)
(5)mul t2, t2, t1
(6)sw, t2, 0(t3)
(7)addi t3, t3, 4
(8)subi t0, t0, 1
(9)bne t0, x0, label
(10-15)nop
```



כמה מחזורי שעון ייקחו עד לסיום פקודה 9 בה הקפיצה אינה נלקחת (לאחר שפקודה 9 סיימה את שלב ה-WB)?

א. 21

ב. 23

ג. 14

ד. 13

ה. 32



פתרון: ב'

Cycle	Fetch	Decode	Execute	Memory	WriteBack	Explain?
1	1	X	X	X	X	
2	2	1	X	X	X	
3	3	2	1	X	X	
4	4	3	2	1	X	
5	5	4	3	2	1	
6	6	5	4	3	2	
7	6	5	nop	4	3	5 wants to use the value read in 4. But 4 is in the execute stage, we need to stall the pipe one clock.
8	7	6	5	nop	4	Sw uses the value of t2 just in the Memory stage. We can forward it from the WB.
9	8	7	6	5	nop	
10	9	8	7	6	5	
11	10	9	8	7	6	We assume always not taken, therefore we fetch instruction 10.
12	11	10	9	8	7	The decision on the branch is taken place when 9 is in the execute stage.
13	4	nop	nop	9	7	Branch is taken. We flush 10&11.
14	5	4	nop	nop	9	
15	6	5	4	nop	nop	
16	6	5	nop	4	nop	Again, 5 wants to use the value read in 4.
17	7	6	5	nop	4	
18	8	7	6	5	nop	
19	9	8	7	6	5	
20	10	9	8	7	6	We assume not taken, therefore we fetch 10.
21	11	10	9	8	7	
22	12	11	10	9	8	The branch is not taken.
23	13	12	11	10	9	Done ;-)



שאלה 9 (5 נקודות)

נתון מעבד Pipeline RISC-V ללא hazard detection unit וללא forwarding בכלל. המעבד מניח כי ה-branch אינו מתרחש ועושה flush במקרה שכן. ההחלטה על ה branch מתקבלת בשלב MEM. על מנת שהקוד המצורף ירוץ כהלכה, יש להוסיף פקודות סח. מהו צירוף השיפורים במעבד שיגרמו לכך שמספר פקודות ה-סח שיש להוסיף יהיה מינימלי?

- הוספת forwarding מ MEM ל EXE, ו- forwarding מ WB ל EXE
- הוספת forwarding מ MEM ל EXE, ו- hazard detection unit
- הוספת forwarding מ WB ל EXE, ו- hazard detection unit
- הוספת forwarding מ WB ל EXE, ו- forwarding מ WB ל DEC
- הוספת forwarding מ MEM ל EXE, ו- forwarding מ WB ל DEC

```

1      addi t1, x0, 9
2      sub t2, x0, x0
3      add s0, t1, t1
4      loop: lw t3, 0(s1)
5          lw t4, 4(s1)
6          xori s3, t3, 256
7          addi s3, s3, 1024
8          sw t3, 8(s1)
9          addi s2, s2, 1
10     bne s2, t1, loop
11     add t5, x0, x0
12     addi t6, t6, 0
13     add t7, x0, x0
    
```

פתרון:
תשובה א'

```

1      addi t1, x0, 9
2      sub t2, x0, x0
3      nop
4      nop
5      add s0, t1, t1
6      loop: lw t3, 0(s1)
7          lw t4, 4(s1)
8          nop
9          nop
10         nop
11         nop
12         xori s3, t3, 256
13         nop
14         nop
15         nop
16         addi s3, s3, 1024
    
```



```
8      sw  t3, 8(s1)
9      addi s2, s2, 1
      nop
      nop
      nop
10     bne s2, t1, loop
11     add t5, x0, x0
12     addi t6, t6, 0
13     add t7, x0, x0
```

שאלה 10 (6 נקודות)

נתונה התוכנית הבאה אשר רצה על מעבד multicycle RISC-V, התומך בטיפול בכל החריגות שנלמדו בקורס:

```
sub x15, x0, 1
lw x16, 0xFF(x15)
add x2, x0, x3
unknown
add x2, x0, t4
add x2, x0, t5
```

- כל פונקציות הטיפול בחריגות מפסיקות את ריצת התוכנית כאשר מתגלה חריגה והתוכנית לא חוזרת להמשיך את ריצתה.
- הפקודה unknown היא בעלת opcode לא חוקי (מתגלה בשלב ה ID) ולכן גורמת לחריגה.
- לפני ריצת התוכנית ערכי הרגיסטרים הם לפי מספרם. לדוגמה רגיסטר x4 מכיל את הערך 4.

הערה: הפקודה lw מתייחסת למידע ברגיסטר המכיל את הכתובת כמספר חיובי.

מה ערכו של רגיסטר x2 לאחר סיום ריצת התוכנית (סיום התוכנית או הפסקתה)?

- א. 0
- ב. 2
- ג. 3
- ד. 4
- ה. 5



פתרון:

תשובה ב'.

ישנה חריגת גלישה ב w לכן התוכנית תפסיק את הריצה וערך הרגיסטר לא ישתנה.

שאלה 11 (6 נקודות)

להלן קוד שאמור להתבצע על אחד ממעבדי RISC-V:

```
00000010000011100010100000000011
000000010000110000000011000110011
```

נתון כי לפני ביצוע שתי הפקודות הנ"ל:

- התוכן של כל רגיסטר ב-register file שווה למספר הרגיסטר, לדוגמה רגיסטר $x4$ מכיל את הערך 4.
- התוכן של כל בית בזיכרון הנתונים שווה לבית התחתון של כתובתו, לדוגמה בכתובת $0x12345678$ נמצא הערך $0x78$.

מהו הערך ברגיסטר $X12$ לאחר ריצת 2 הפקודות?

- א. 4
- ב. 12
- ג. 32
- ד. 84
- ה. 94

תשובה: ד'

```
000000100000 11100 010 10000 0000011
0000000 10000 11000 000 01100 0110011
```

הפקודה הראשונה היא פקודת w שטוענת מילה מכתובת שהיא ברגיסטר 28 (תוכן 28) ועוד היסט 32, לרגיסטר 16

הפקודה השנייה היא פקודת ADD בין רגיסטר 16 לרגיסטר 24 לתוך רגיסטר 12
לכן התשובה 84.



שאלה 12 (10 נקודות)

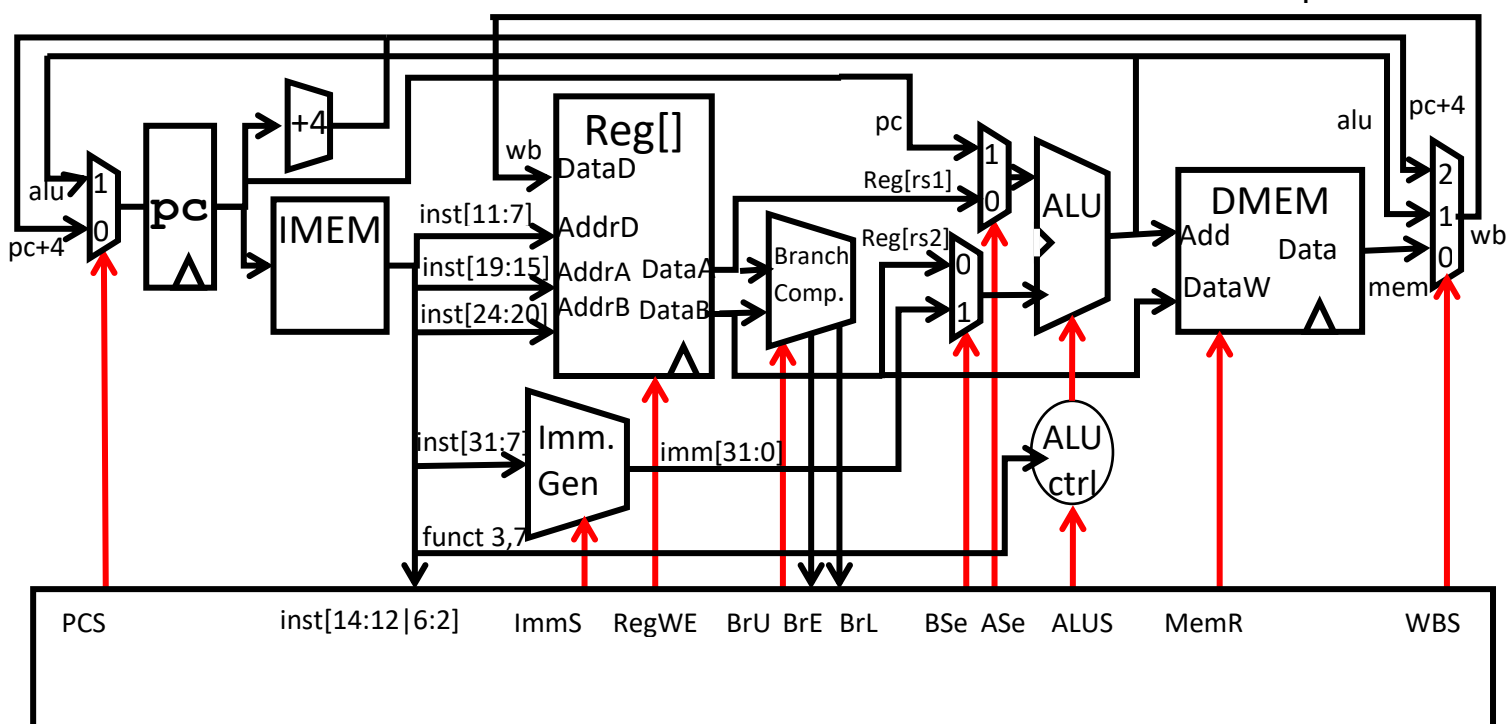
המהנדס הנודע ארכיטקטור ה-Single cycle RISC5 להוסיף לארכיטקטורה את הפקודה החדשה הבאה:

StorePC *rs1, imm*

פקודה זו משתמשת בכתובת הזיכרון אשר שמורה ברגיסטר *rs1*, ומאחסנת בכתובת זו את הערך $PC + imm$. לצורך התמיכה בפקודה ניתן להוסיף בוררים וסיביות בקרה בלבד. הניחו כי הפקודה היא בעלת Opcode חדש, אשר מגדיר פעולה בעלת מבנה דומה לפקודות מסוג S (הניחו כי סיביות בקרה קיימות ניתנות להתאמה בהתאם לדרוש עם Opcode זה וניתן לבצע כל שינוי בבקר).

[31:25]	[24:20]	[19:15]	[14:12]	[11:7]	[6:0]
Imm[11:5]	00000	Rs1	101	Imm[4:0]	OpCode=1111111

ציירו את השינויים אשר נדרשים במערכת עבור תמיכה בפקודה החדשה (ניתן לצייר גם מתחת לשרטוט ולציין את שמות החוטים אליהם מתחברים).



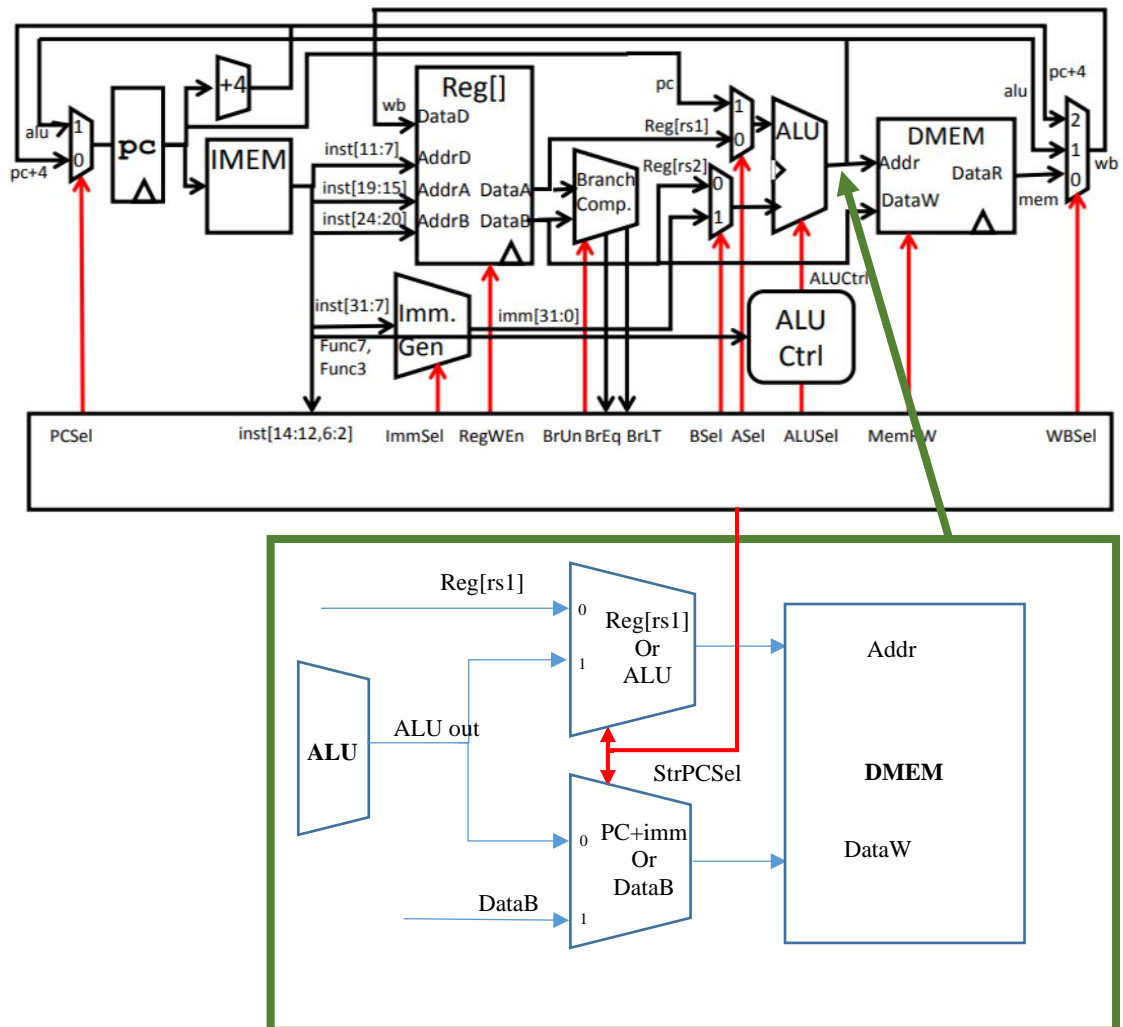
ניתן לשרטט גם כאן:



תשובה: ד'

פתרון:

עלינו להוסיף 2 בוררים בין ה-ALU לבין הזיכרון (החלק הירוק), אשר אליהם נחבר סיביות בקרה חדשה (אותה סיביות לשני הבוררים).





שאלה 13 (10 נקודות)

שירי החליטה להכין עוגת גבינה. על מנת שהעוגה תתייצב בהצלחה, עליה לשמור עליה ב: $-2 \leq temp$.
לרשותה מקרר שבו טווח הטמפרטורה הוא: $-6 \leq temp \leq 6$. המקרר מוציא חיווי לטמפרטורה ע"י 4 יציאות $xyzw$ (כשערך הטמפרטורה מחושב בשיטת מספרים מכוונים, כלומר המשלים ל2).
עליכם לממש מערכת צירופית $F(x, y, z, w)$ שתעזור לשירי, ותוציא את הערך '1' כאשר טמפ' המקרר יורדת מתחת ל-2.
רשמו את הפונקציה המצומצמת ביותר אשר מתארת את התנהגות המערכת והשלימו את מפת הקרנו הבאה:

ZY	XY			
	00	01	11	10
00				
01				
11				
10				

פתרון:

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

	V	X	y	z	w	f
0	0	0	0	0	0	
1	1	0	0	0	1	
2	2	0	0	1	0	
3	3	0	0	1	1	
4	4	0	1	0	0	
5	5	0	1	0	1	
6	6	0	1	1	0	
7	7	0	1	1	1	Φ
8	8-	1	0	0	0	Φ
9	7-	1	0	0	1	Φ
10	6-	1	0	1	0	1
11	5-	1	0	1	1	1
12	4-	1	1	0	0	1
13	3-	1	1	0	1	1
14	2-	1	1	1	0	
15	1-	1	1	1	1	

ZY	XY			
	00	01	11	10
00			1	Φ
01			1	Φ
11		Φ		1
10				1

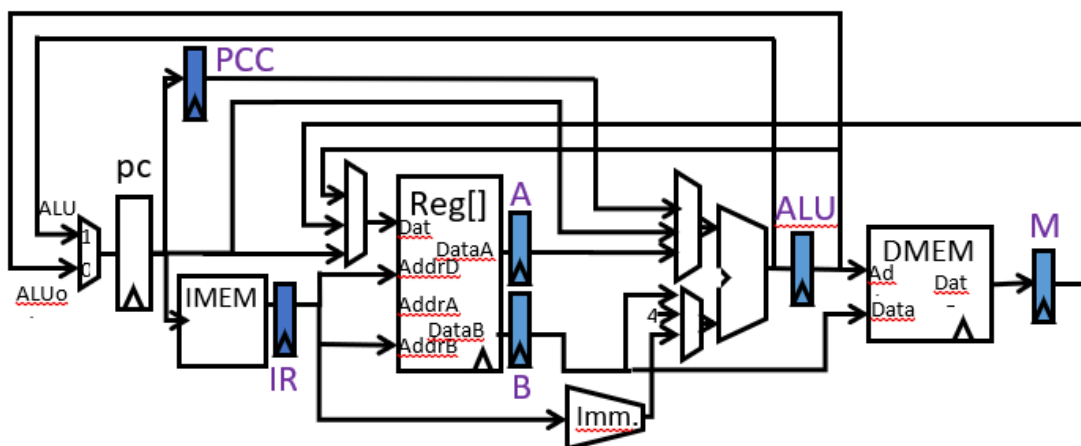
שאלה 14 (9 נקודות)

נרצה להוסיף למעבד Multicycle Risc את הפקודה MAC (multiply and accumulate) פורמט הפקודה:

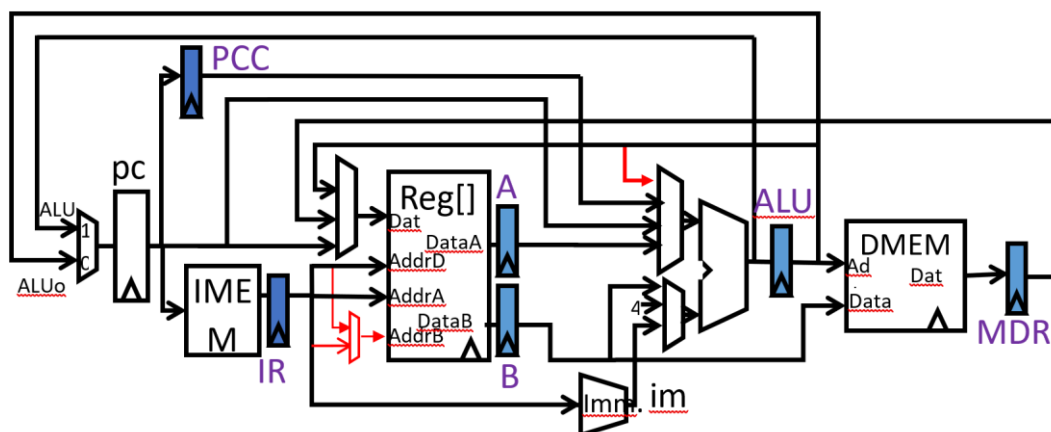
MAC rd, rs1, rs2

$rd = rd + rs1 * rs2$

ציירו בשרטוט מטה את השינויים הנדרשים על-מנת לתמוך בפקודה (ניתן להוסיף רכיבי אטום או להוסיף כניסות לרכיבי אטום קיימים **בלבד**). הניחו כי הפקודה MUL נתמכת ע"י המעבד ב-ALU.



פתרון:



נוסיף חיבור מיציאת רגיסטר ה-ALU אל ה-MUX שמחובר ל-ALU.
בנוסף, נחבר את כניסת AddrB אל יציאת MUX הבוחר בין rd ל- $rs1$.
כך, במחזור השעון הראשון נחשב את תוצאת הכפל בין $rs1$ ל- $rs2$ ובמחזור השעון הבא, נחשב את הסכום של rd עם תוצאת הכפל שחישבנו, ונטען אותה לזכרון.



שאלה 15 (5 נקודות)

עבור מעבדי RISC-V המצורפים בנספח, נדרש להוסיף מימוש של הפקודה `lrsi` (כפקודת מכונה אמיתית), אשר שומרת תוכן של מילה מהזיכרון (מכתובת שמורכבת מחיבור של רגיסטרים `rs1` ו `rd`) ועוד `imm`, ברגיסטר `rd`. שינויים מותרים במסלולי הנתונים: ניתן להוסיף ולהרחיב MUXs, לחוות ולהתאים את זמן מחזור השעון מחדש. שינויים מותרים בבקרים: ללא הגבלה. נדרש כמובן לא לפגוע בפקודות אחרות.
פקודה זו בעלת הפורמט:

`lrsi rd, rs1, imm`

המבצעת את הפעולה הבאה:

`rd ← Mem[rs1+rd] +imm`

א. (2 נק') מה מבנה הפקודה (מבין 6 המבנים)?

ב. באיזה מבין מעבדי RISC-V ניתן לממש פקודה זו כפקודת מכונה? הקף:

(1 נק') Single Cycle: אפשר / אי אפשר

(1 נק') Multi Cycle: אפשר / אי אפשר

(1 נק') Pipeline: אפשר / אי אפשר

פתרון:

א. הפקודה מקודדת כפקודת I-type.

ב. ב Multicycle ניתן לבצע את חישוב הכתובת והמידע בשלבים תוך שימוש ב ALU. ב SC וב Pipeline לא ניתן מכיוון שיש רק ALU יחיד.



שאלה 16 (6 נקודות)

נתונה הפונקציה: $f(w, x, y, z) = \sum(2,3,4,5,6,7,8,10,12,13)$.

א. (2 נק') האם ניתן לממש אותה באמצעות בוררים $1 \rightarrow 4$ והקבועים 0 ו 1 בלבד?
סמן:
כן / לא

ב. (4 נק') האם ניתן לממש אותה באמצעות בורר $1 \rightarrow 4$ יחיד, מספר בלתי מוגבל של שערי not והקבועים 0 ו 1 בלבד? סמן:
כן / לא

פתרון:

	W	X	Y	Z	f
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

א. כן
ב. כן