



מערכות ספרתיות ומבנה המחשב (044252)

סמסטר חורף תש"פ

בחינה סופית – מועד ב

3 במרץ 2020

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 ולמחשבון.

הנחיות והוראות:

- הבחינה כתובה על גבי 15 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.
- 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), 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 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 ה-SystemVerilog הבא, כאשר counter הוא module המממש מונה בעל 2 ביטים (סופר מ-0 עד 3):

```
module my_module (
    input logic clk,
    input logic rst,
    input logic a,
    output logic q
);
    typedef enum { S0_st, S1_st, S2_st } sm_type;

    sm_type current_state;
    sm_type next_state;

    always_ff @(posedge clk, posedge rst) begin
        if (rst == 1'b1) begin
            current_state <= S0_st;
        end
        else begin
            current_state <= next_state;
        end
    end

    always_comb begin
        case (current_state)
            S0_st: begin
                next_state = S1_st;
                q = 1'b0;
            end
            S1_st: begin
                next_state = S2_st;
                q = 1'b0;
            end
            S2_st: begin
                next_state = S0_st;
                q = a;
            end
            default: begin
                next_state = S0_st;
                q = 1'b0;
            end
        endcase
    end
endmodule
```

```
module my_module2 (
    input logic clk,
    input logic rst,
    output logic out
);
    logic [1:0] cnt;
    counter cnt_inst(.clk(clk), .rst(rst), .cnt(cnt));
    my_module inst (.clk(clk), .rst(rst), .a(cnt[0] & cnt[1]),
        .q(out));
endmodule
```



```
module counter (  
    input logic clk,  
    input logic rst,  
    output logic [1:0] cnt  
);  
always_ff @(posedge clk, posedge rst)  
begin  
    if (rst == 1'b1) begin  
        cnt <= 2'b00;  
    end  
    else begin  
        cnt <= cnt + 1;  
    end  
end  
endmodule
```

Duty cycle מוגדר עבור אות מחזורי בתור החלק היחסי מתוך זמן מחזור האות שבו האות בעל ערך '1' (למשל, duty cycle של אות שעון סטנדרטי הוא $1/2$). מהו ה-duty cycle של הסיגנל out (היציאה של my_module2)?

- א. $1/12$
- ב. $1/6$
- ג. $1/4$
- ד. $1/3$
- ה. $1/2$

שאלה 2 (5 נקודות)

עבור פקודות האסמבלי הבאות, איזו פקודה לא ניתן לממש כפקודה אמיתית במעבד multi-cycle RISC-V (להזכירכם פקודה אסמבלי אמיתית רצה כפקודה אחת על המעבד)?

- ניתן לבצע שינויים בבקר והוספת muxes וחיווטים במסלול הנתונים של המעבד ולהתאים את זמן המחזור, אך **אסור** לבצע שינויים ביחידות Register file, Memory, ALU.
- א. פקודת swap rd, rs אשר מחליפה בין תוכנם של שני הרגיסטרים.
- ב. פקודת addi32 rd, rs, imm אשר מוסיפה ערך immediate בגודל 32 ביט לערך השמור ברגיסטר rs ושומרת ב-rd.
- ג. פקודת cp rs1, rs2 אשר מעתיקה מילה מהזכרון מהכתובת שנתונה ברגיסטר rs1 לכתובת בזיכרון אשר נתונה ברגיסטר rs2.
- ד. פקודת sub3 rd, rs1, rs2 אשר מבצעת את הפעולה $reg[rd] = reg[rd] - reg[rs1] - reg[rs2]$ (שומרת את התוצאה לרגיסטר rd).
- ה. ניתן לממש את כל הפקודות הנ"ל.



שאלה 3 (5 נקודות)

משדר ומקלט מתקשרים באמצעות קו יחיד לפי השיטה הנלמדת בקורס. נתון כי בעת השידור ישנו רעש אשר הופך באופן אקראי סיבית אחת (מתוך 8 סיביות המידע) בכל שידור ('0' לוגי הופך ל-'1' ולהפך). סיביות start ו-stop נשארות תקינות. מהו השינוי שיוכל להקטין בצורה המשמעותית ביותר את הסיכוי לשגיאה בשחזור המידע במקלט?

- א. שימוש בסיבית זוגיות אחת לכל 8 סיביות המידע.
- ב. שימוש בסיבית זוגיות אחת לארבעת סיביות המידע הראשונות וסיבית זוגית לארבעת הסיביות הבאות.
- ג. שידור כל סיבית פעמיים.
- ד. שידור כל סיבית שלוש פעמים.
- ה. לא ניתן לשחזר את המידע בכלל.

שאלה 4 (5 נקודות)

ממשו את הפונקציה $F(x,y,z)=xy+z'$ בעזרת שערי NAND עם 2 כניסות בלבד. מהו מספר שערי ה-nand המינימלי שיש צורך להשתמש בהם?

- א. 1
- ב. 2
- ג. 3
- ד. 4

ה. אין אפשרות לממש את הפונקציה הזו באמצעות שימוש בשערי nand

שאלה 5 (5 נקודות)

נדרש לתכנן מערכת עקיבה מסוג מילי בעלת כניסה יחידה ויציאה יחידה המפיקה $Z=1$ אמ"מ אינדקס הכניסה עד כה הוא זוגי ומספר האחדים עד כה הוא זוגי. יש להניח כי המערכת מתחילה ממצב של רצף אפסים באורך זוגי. כמה מצבים יהיו במכונה המצומצמת?

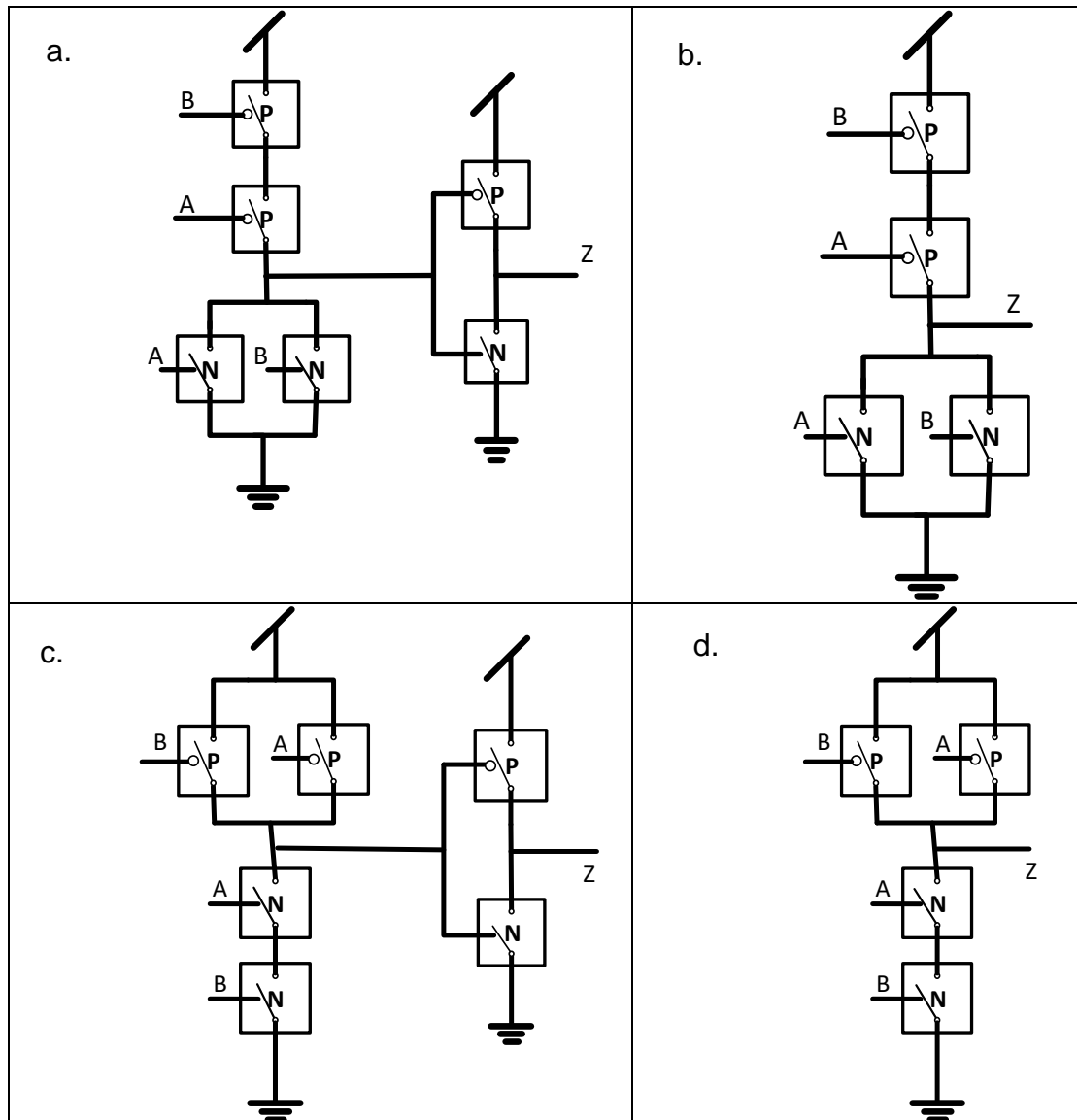
- א. 3
- ב. 4
- ג. 5
- ד. 6
- ה. 8



שאלה 6 (5 נקודות)

נתונים מספר שערים לוגים אשר מומשו על ידי מתגים.
מבין האפשרויות הבאות, מהו תכנון המתגים אשר מממש שער $Z = A + B$ or ?

תזכורת: מתג P מחובר כאשר כניסתו היא 0, ומתג N מחובר כאשר כניסתו היא 1.



- א. איור a
ב. איור b
ג. איור c
ד. איור d
ה. אף תשובה לא נכונה



שאלה 7 (5 נקודות)

כדי לחזק את יכולת גילוי השגיאות של קוד $Gray$ עבור מילה באורך n , כאשר n הוא חזקה שלמה של 2, הוצע להוסיף לכל מילה מקודדת ביט זוגיות אשר ימוקם בביט ה-LSB. לדוגמא, עבור המספר הבינארי 10011, הקידוד אשר יתקבל לאחר המרה לקוד $Gray$ יהיה 11010, ולאחר הוספה של ביט הזוגיות נקבל כי הקידוד הסופי יהיה 110101. הנכם מתבקשים לממש מעגל אשר יבצע את ההמרה מייצוג בינארי אל הייצוג בקוד $Gray$ בתוספת סיביות הזוגיות. לצורך המימוש ניתן להשתמש בשערי xor בעלי 2 כניסות בלבד. מהו מספר שערי ה- xor המינימלי (יש למצוא את פונקציית המיתוג המצומצמת ביותר) כתלות ב- n אשר דרוש בכדי שהמעגל יפעל בצורה תקינה?

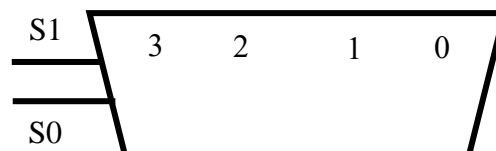
הערה: כדאי להיעזר בנוסחת ההמרה לקוד $Gray$ המופיעה בדף הנוסחאות.

- א. $n - 2$
- ב. $n - 1$
- ג. n
- ד. $2n - 3$
- ה. $3n - 4$

שאלה 8 (5 נקודות)

נתונים מספר מימושים של פונקציה בעזרת בוררים. מבין המימושים המוצגים (בעמוד הבא), אילו מימושים מבצעים המרה של מספר חיובי בן 2 סיביות מידע, a_1a_0 , המיוצג בייצוג ללא סימן, אל הייצוג שלו כמספר שלילי במשלים ל-2 (בעל 3 סיביות) $b_2b_1b_0$? עבור המספר 0 כקלט, הרכיב נדרש להוציא 0 בכל סיביות המוצא.

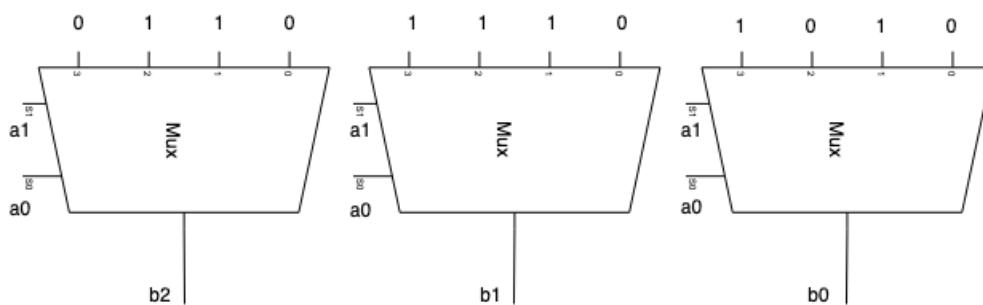
הערה: כל ה-Muxes אשר מופעים מטה, מתוכננים בצורה הבאה:



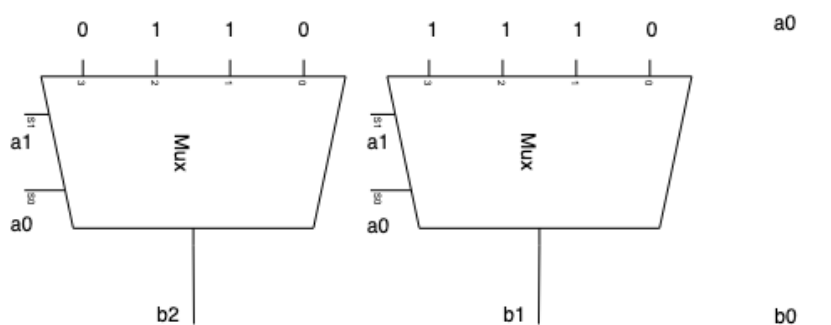
- א. מימוש a בלבד
- ב. מימוש b בלבד
- ג. מימוש c בלבד
- ד. מימושים a ו- b
- ה. מימושים c ו- d



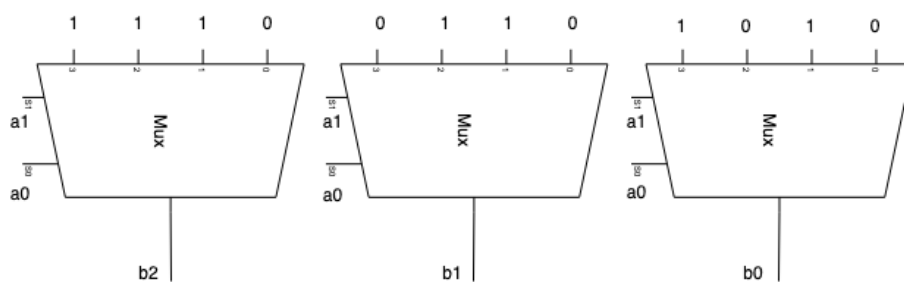
a.



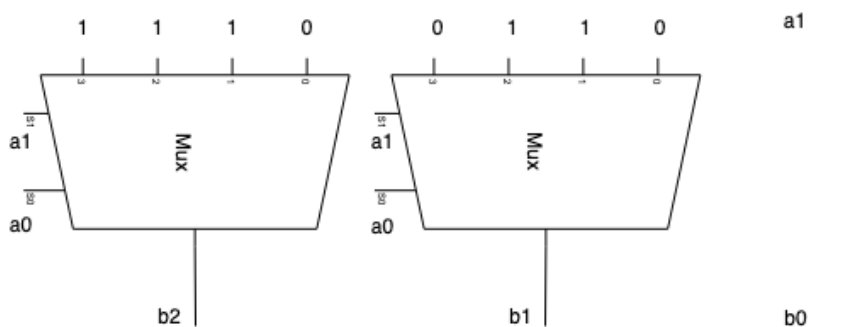
b.



c.



d.





שאלה 9 (5 נקודות)

נתון קטע הקוד הבא:

0x0111 0000	Func:	addi sp, sp, -12	
0x0111 0004		addi t1, x0, 2	
0x0111 0008		sw ra, 0(sp)	
0x0111 000C		sw a0, 4(sp)	
0x0111 0010		blt a0, t1, Exit	// Stop condition
0x0111 0014		addi a0, a0, -1	
0x0111 0018		jal ra, Func	// Func's output is stored in a0
0x0111 001C		add a1, a0, x0	
0x0111 0020		lw a0, 4(sp)	
0x0111 0024		sw a1, 8(sp)	
0x0111 0028		addi a0, a0, -2	
0x0111 002C		jal ra, Func	// Func's output is stored in a0
0x0111 0030		lw a1, 8(sp)	
0x0111 0034		add a0, a0, a1	
0x0111 0038	Exit:	lw ra 0(sp)	// Return value section
0x0111 004C		addi sp, sp, 12	
0x0111 0050		jr ra	
0x0112 0000	Main:	addi sp, sp, -4	
0x0112 0004		addi a0, x0, 2	
0x0112 0008		sw ra, 0(sp)	
0x0112 000C		jal ra, Func	// Func's output is stored in a0
0x0112 0010		lw ra, 0(sp)	
0x0112 0014		addi sp, sp, 4	
0x0112 0018		jr ra	

התוכנית מתחילה לרוץ מ-Main. מה יהיה ערכו של רגיסטר a0 בסיום ריצת התוכנית?
הערה: ניתן לפתור את השאלה גם ללא שימוש בטבלת מעקב.

- א. 1
- ב. 2
- ג. 3
- ד. 4
- ה. 0



שאלה 10 (5 נקודות)

נתון מעבד SingleCycle RISCv כפי שנלמד בכיתה התומך בפקודות הנלמדו בכיתה.
נתון שהמעבד עבר שינויים על מנת לתמוך גם בפקודה:

adMR – *add memory register*: *adMR rd, rs1, rs2*

שמבצעת את הפקודה הבאה:

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

נתון:

	Timing
Memory Access (Instruction or Data)	1 ns
Read a value from the register file	1 ns
ALU operation	2 ns
Write a value to the register file	1 ns

מהו משך זמן הביצוע המינימלי של הפקודה *addi rd, rs1, imm* אשר נתמכה לפני השינוי?

א. $T = 6ns$

ב. $T = 8ns$

ג. $T = 10ns$

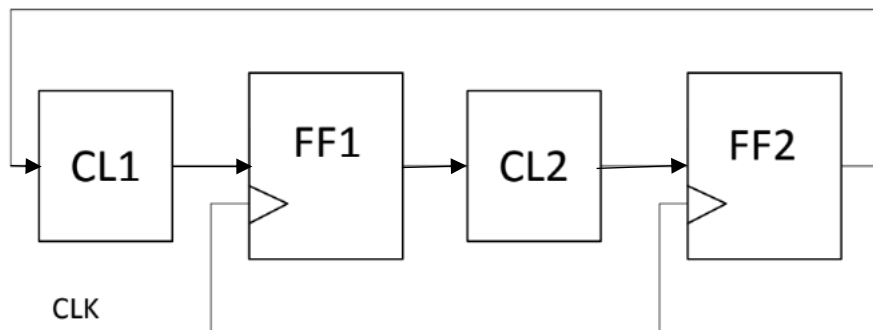
ד. $T = 5ns$

ה. $T = 4ns$



שאלה 11 (5 נקודות)

נתון המעגל הבא:



ונתונים גם:

	t_{hold}	t_{setup}	t_{pcQ}/t_{pd}	t_{cd}
FF1	$4ns$	$3ns$	$4ns$	$1ns$
FF2	$4ns$	$4ns$	$7ns$	$2ns$
CL1			$4ns$	$1ns$

מהם נתוני CL2 שיאפשרו תדר עבודה מקסימלי למעגל:

א. $t_{pd2} \leq 6ns$ $t_{cd2} \leq 2ns$

ב. $t_{pd2} \leq 6ns$ $t_{cd2} \geq 3ns$

ג. $t_{pd2} \leq 2ns$ $t_{cd2} \geq 3ns$

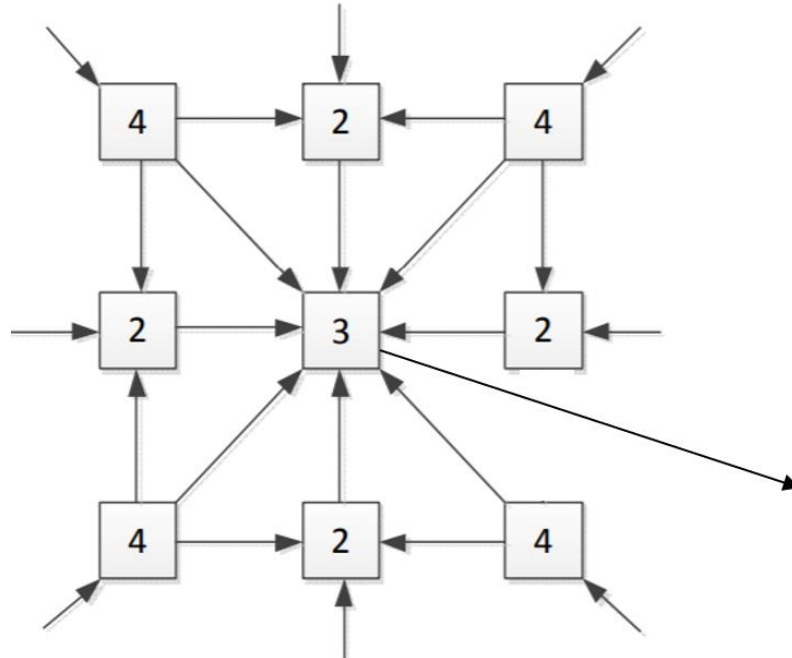
ד. $t_{pd2} \leq 4ns$ $t_{cd2} \leq 3ns$

ה. המעגל לא עומד במשטר הדינמי.



שאלה 12 (5 נקודות)

נתונה המערכת הבאה:



השהיית כל רכיב ב- ns רשומה בתוך הקובייה.

מוצא המערכת הוא מהיחידה המרכזית (השהייה של $3ns$).

$$t_{setup} = t_{pcq}(FF) = 1ns$$

נרצה לצנר את המערכת בצורה שתבטיח throughput מקסימלי (בעדיפות ראשונה) ו-latency מינימלי.

ניתן להניח כי הכניסות עומדות במשטר הזמנים הדינאמי.

מהו מספר הרגיסטרים שיבטיח תנאים אלו ומהו ה-latency?

א. Num =24 Latency =18ns

ב. Num =23 Latency =12ns

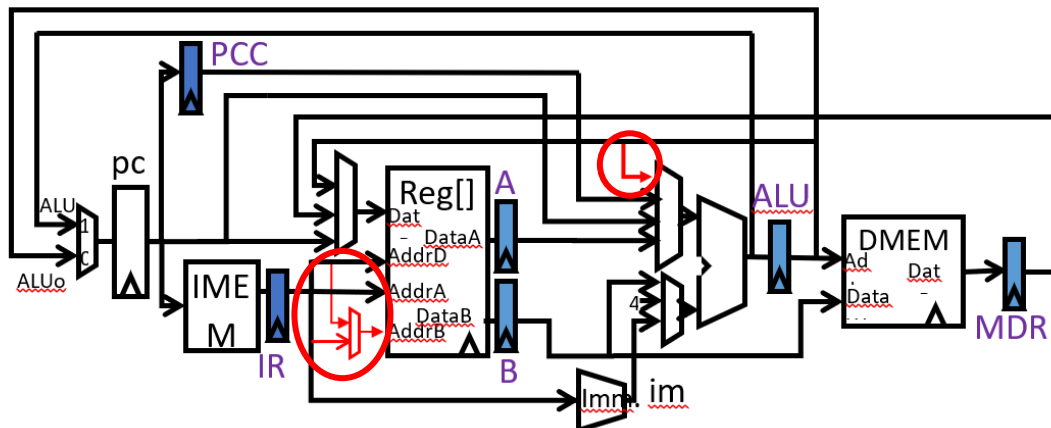
ג. Num =16 Latency =18ns

ד. Num =24 Latency =12ns

ה. Num =23 Latency =8ns

שאלה 13 (5 נקודות)

נתון מעבד ה-Multicycle Risc הבא אשר תומך בפקודות MAC (multiply and accumulate). השינויים במעבד מוקפים במעגל.



מלבד קו הבקרה אשר שייך ל-Mux בכניסת ה-Register File לא נוספו קווי בקרה נוספים. הפקודה מקודדת כפקודת R-type אך בעלת opcode ייחודי אשר שונה מכל הפקודות האחרות מסוג R-type. פורמט פקודת ה-Mac הינו:

MAC rd, rs1, rs2

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

תזכורת:

Dispatch ROM 2		
Op	Name	Value
x03	LW	3
x23	SW	5

Dispatch ROM 1		
Op	Name	Value
x33	R-type	6
x67	JAL	9
x63	BEQ	8
x03	LW	2
x23	SW	2

אילו שינויים נצטרך לבצע במכונת המצבים של הבקר על מנת לתמוך בפקודה ה-MAC? הניחו כי התמיכה בסיביות הבקרה של ה-muxes כבר נוספה לבקר.

ניתן ומומלץ להיעזר במכונת המצבים בדף הנוסחאות. שימו לב כי מספרי המצבים הישנים (אלו אשר מופיעים בדף הנוסחאות ונלמדו בכיתה) אינם משתנים.

- הוספת שורה ב-Dispatch ROM 1, והוספת מצב חדש למכונת המצבים.
- הוספת Dispatch ROM חדש ושני מצבים חדשים.
- הוספת שורה ב-Dispatch ROM 1, הוספת Dispatch ROM חדש והוספת מצב חדש למכונת המצבים.
- לא נצטרך לבצע שינויים במכונת המצבים כי המעבד כבר תומך בכלל ובחיבור.
- הוספת שורה ב-Dispatch ROM 1, הוספת Dispatch ROM חדש ושני מצבים חדשים.



שאלה 14 (10 נקודות)

נתון מעבד Multi cycle RISC-V אשר תומך בחריגות אשר מפורטות בטבלה מטה. הטיפול בחריגות מבוצע על ידי שיטת קוד הגורם לחריגה. הפעולות אשר מבצעת מערכת ההפעלה לצורך הטיפול בכל סוג של חריגה מתוארות בטבלה. בנוסף, הטבלה מתארת מהו הקידוד עבור כל חריגה.

סוג החריגה	קידוד	פעולה רצויה
חלוקה ב-0	1	השוואת רגיסטר המכנה ל-2
גלישה	4	השוואת הרגיסטרים אשר גורמים לגלישה ל-0

הניחו כי הגישה אל הרגיסטרים SCAUSE ו-SPEC מתבצעת בצורה זהה לשאר הרגיסטרים במערכת.

בשאלה זו הניחו כי ערכי הרגיסטרים בפעולות add מטופלים כמספרים בייצוג unsigned. פעולות addi עובדות כרגיל.

השלימו את הקוד הבא כך שירוצן כהלכה ויתמוך בכל סוגי החריגות עבור קטע ה-main הנתון:

```

0x1AA0 0000      Main:  addi s0, x0, 4
0x1AA0 0004      addi t0, x0, 4
0x1AA0 0008      addi s1, x0, -1
0x1AA0 000C      add s1, t0, s1
0x1AA0 0010      div t0, s0, s1
0x1AA0 0014      add s0, s0, t0
0x1AA0 0018      Exit:  j ra

0x1C09 0000      Interrupt handler: addi sp, sp, -4
0x1C09 0004      sw s0, _____
0x1C09 0008      addi s0, x0, 1
0x1C09 000C      addi s2, x0, 4
0x1C09 0010      beq SCAUSE, s0, Div0
0x1C09 0014      beq _____, s2, _____
0x1C09 0018      Done:  lw s0, 0(sp)
0x1C09 001C      addi sp, sp, 4
0x1C09 0020      jr, _____
0x1C09 0024      Div0:  addi _____, x0, _____
0x1C09 0028      j _____
0x1C09 002C      Overflow: addi s1, x0, x0
0x1C09 0030      addi _____, _____, _____
0x1C09 0034      j Done
0x1C09 0000      Interrupt handler: addi sp, sp, -4

```



שאלה 15 (10 נקודות)

נתון מעבד Pipeline RISC-V ללא hazard detection unit וללא forwarding בכלל. למעבד אין יכולת לבצע flush כלל.

א. (2 נקודות) כיצד משפיעה העובדה שלמעבד אין יכולת לבצע flush על ביצוע פקודות ?branch

ב. (8 נקודות) נתונה התוכנית:

```

1      addi t1, x0, 0
2      addi t2, t1, 2
3  loop: lw  t3, 0(s1)
4      lw  t4, 4(s1)
5      add t5, t3, t4
6      add t0, t0, t5
7      subi t2, t2, 1
8      bne t1, t2, loop
9      add t6, t6, s1
10     add t7, t7, s1
11     add t8, t8, s1
    
```

מלאו את הטבלה הבאה (ראו דוגמא כיצד למלא בשורה הראשונה) כך שתתאר את מספר ה-nops אשר צריך להוסיף לקוד על מנת שירוך באופן תקין. זכרו כי אין אפשרות לדעת מראש האם הקפיצה מתרחשת או לא.

בין פקודה מספר	לבין פקודה מספר	כמות nops
1	2	3



שאלה 16 (15 נקודות)

נגדיר 2 רצפים: רצף A – 1001, רצף B – 1100 (הרצפים מוגדרים משמאל לימין)

יש לממש מכונת מצבים סינכרונית מסוג מילי בעלת כניסה אחת ויציאה אחת, המפיקה 1 במוצא למשך מחזור שעון יחיד אם מתקבל בכניסה רצף A או רצף B. המימוש נדרש להיות מימוש מינימלי, כאשר עליכם לציין מהו מוצא המכונה בכל מעבר מצבים. הניחו כי במצב המכונה ההתחלתי המכונה קיבלה רצף של 0-ים.

דוגמא:

<i>clk</i>	1	2	3	4	5	6	7	8	9	10	11	12
<i>in</i>	0	1	0	0	1	1	0	0	1	1	0	0
<i>out</i>	0	0	0	0	1	0	0	1	1	0	0	1