

חלק א': (24 נקודות)

בחלק זה 6 שאלות רב ברירה. יש לבחור את התשובה הנכונה מבין התשובות האפשריות. משקל כל שאלה 4 נקודות. יש לסמן באופן ברור ב-X, בטבלה שלפניכם, את התשובה הנכונה.
הערה: יש לסמן רק אפשרות אחת לכל שאלה! ייבדקו רק הסימונים בטבלה. כל רישום ליד השאלה עצמה הינו בחזקת טיוטה, ולא ייבדק!!!

שאלה	א	ב	ג	ד	ה
1			X		
2	X				
3					X
4		X			
5				X	
6					

שאלה 1

נתונים שני מעבדים שונים: P1, P2. בשני המעבדים קיים אותו אוסף פקודות. ניתן לחלק את הפקודות ל-4 קבוצות שונות: A, B, C ו-D.

קבוצה	% פקודות	CPI במעבד P1	CPI במעבד P2
A	20%	6	6
B	30%	6	4
C	10%	6	2
D	40%	3	2

נתון כי קצב שעון של P1 הוא 3GHz

נתון כי קצב שעון של P2 הוא 2GHz

בטבלה משמאל נתון מספר מחזורי שעון לכל קבוצת פקודות, עבור כל מעבד. כמו כן, נתון בטבלה אחוז יחסי של הפקודות, עבור כל קבוצת פקודות, מכלל פקודות התוכנית.

בהינתן תוכנית X בת 10^8 פקודות, המחולקת לקבוצות הפקודות בהתאם ל "% הפקודות" הרשום בטבלה: מה מדד ההאצה, ובאיזה מעבד התוכנית תרוץ מהר יותר?

א. לא ניתן לדעת על בסיס הנתונים בשאלה

ב. מדד ההאצה הינו 1.000, על בסיס הנתונים בשאלה זו – יש זהות בין המעבדים

ג. מדד ההאצה הינו 1.063, על בסיס הנתונים בשאלה זו – מעבד P1 מהיר יותר

ד. מדד ההאצה הינו 1.063, על בסיס הנתונים בשאלה זו – מעבד P2 מהיר יותר

ה. מדד ההאצה הינו 0.824, על בסיס הנתונים בשאלה זו – מעבד P2 מהיר יותר

$$\begin{aligned} \text{CPI}_1 &= 6 \cdot 0.2 + 6 \cdot 0.3 + 6 \cdot 0.1 + 3 \cdot 0.4 = 1.2 \\ \text{CPI}_2 &= 6 \cdot 0.2 + 4 \cdot 0.3 + 2 \cdot 0.1 + 2 \cdot 0.4 = 0.85 \\ \text{CPUtime}_1 &= \text{IC} \cdot \text{CPI}_1 / (1 / \text{CR}) = 10^8 \cdot 1.20 / (3 \cdot 10^9) = 0.0400 \\ \text{CPUtime}_2 &= \dots\dots\dots = 10^8 \cdot 0.85 / (2 \cdot 10^9) = 0.0425 \\ \text{SpeedUp} &= \text{cpu(s)}/\text{cpu(f)} = 0.0425 / 0.0400 = 1.063 \end{aligned}$$

שאלות 2 ו-3 מתייחסות לתמונת סגמנט הקוד שלהלן, שנלקחה מה-MARS:

Address	Code	Basic	Source
0x00400000	0x3c080b00	lui \$8,0x000000b00	12: lui \$t0, 0xb00
0x00400004	0x00084e02	srl \$9,\$8,0x00000018	13: srl \$t1, \$t0, 24
0x00400008	0x00084200	sll \$8,\$8,0x00000008	14: sll \$t0, \$t0, 8
0x0040000c	0x01094025	or \$8,\$8,\$9	15: or \$t0, \$t0, \$t1
0x00400010	0x310800f3	andi \$8,\$8,0x000000f3	16: andi \$t0, \$t0, 0xf3
0x00400014	0x20010003	addi \$1,\$0,0x00000003	17: bne \$t0, 3, somewhere1
0x00400018	0x14280007	bne \$1,\$8,0x00000007	

שאלה 2

מה ניתן להגיד על הפקודה בשורה 17 בקוד המקור: **bne \$t0, 3, somewhere1**?

א. זו פקודת פסיאודו, ניתן לדעת שלא תתבצע קפיצה לתווית somewhere1

ב. זו פקודת אסמבלר בסיסית, ניתן לדעת שלא תתבצע קפיצה לתווית somewhere1

ג. זו פקודת פסיאודו, ניתן לדעת שתתבצע קפיצה לתווית somewhere1

ד. זו פקודת אסמבלר בסיסית, ניתן לדעת שתתבצע קפיצה לתווית somewhere1

ה. זו פקודת פסיאודו, לא ניתן לדעת לפי נתוני קטע הקוד האם תתבצע קפיצה לחווית somewhere1

שאלה 3

ערכו של אוגר \$t0 לאחר פקודה 16 הוא 3, ומאחר ומדובר בפקודת BNE, וידוע ערכו של אוגר \$t0 – לא תתבצע קפיצה

12: \$t0 = 0x0b000000

על בסיס הנתונים בתמונת סגמנט הקוד, מה כתובת תווית somewhere1?

13: \$t1 = 0x0000000b

15: \$t0 = 0x00000000

15: \$t0 = 0x0000000b

16: \$t0 = 0x00000003

חישוב התווית, מבוצע לפי הפקודה בכתובת 0x00400018:

$$0x00400018 + 4 + 7*4 = 0x0040001c + 0x1c = 0x00400038$$

א. 0x00400018

ב. 0x0040001c

ג. 0x00400030

ד. 0x00400034

ה. 0x00400038

שאלה 4

נתון זיכרון מטמון עם המאפיינים הבאים:

- מאורגן בשיטת 4-Way Set Associative
- גודל כל בלוק נתונים (data) הינו 64 בתים
- נתון כי יש 16,384 (2^{14}) בלוקים של נתונים במטמון
- הזיכרון ממפה כתובות של 32 סיביות בזיכרון הראשי

מה גודלם של שדות ה- Index וה- Tag במטמון זה?

א. גודל שדה ה- Index הוא 14 סיביות, גודל שדה ה- Tag הוא 10

ב. גודל שדה ה- Index הוא 12 סיביות, גודל שדה ה- Tag הוא 14

ג. גודל שדה ה- Index הוא 12 סיביות, גודל שדה ה- Tag הוא 16

ד. גודל שדה ה- Index הוא 14 סיביות, גודל שדה ה- Tag הוא 14

ה. לא ניתן לדעת על בסיס נתוני השאלה

נתון שיש 2^{14} בלוקים של נתונים, ב- 4 ways set associative, ולכן בכל way יש 2^{12} בלוקים (חילוק ב- 4, 2^2). מכאן... גודל אינדקס הוא 12 סיביות.

נתון שיש 64 מילים בבלוק, כלומר, 16 מילים (64/4). מכאן... גודל Block Offset הוא 4 סיביות (2^4 מילים).

וכמובן, לא לשכוח – 2 סיביות של Byte Offset

חישוב סופי לגודל ה- Tag:

$$32 - 12 - 4 - 2 = 14$$

שאלה 5

נתון הערך -6.325

$$-6.325 = 2^2 + 2^1 + 2^{-2} + 2^{-3} = - (2^2 * (1 + 2^{-1} + 2^{-4} + 2^{-5}))$$

- 2 (1) 10011

- 2+127 (1) 10011

- 129 (1) 10011

1 10000001 1001100000000000....

1100000011001100000000000000000000

1100 0000 1100 1100 0000 0000 0000 0000 = 0xC0CC0000

מהו ערך ההקסה המייצג מספר רציונלי float בהתאם ל- IEEE 754.

א. 0x014C0000

ב. 0x814C0000

ג. 0x40CC0000

ד. 0xC0CC0000

ה. 0xC0E60000

שאלה 6

על סמך התרשים הבא (תרשים B.5.12 בספר בנספח B) איזה מידע לא ניתן לדעת רק על סמך המידע היוצא מקופסה ALU31, לגבי ביצוע תוצאות של פעולות אריתמטיות ב-ALU:

א. את ערך Result31

ב. האם תוצאת הפעולה שבוצעה היא שלילית (במשמעות של המסלים 2)

ג. האם תוצאת הפעולה שבוצעה היא 0 (כלומר, נדלק חיווי Zero)

ד. האם הייתה גלישה אריתמטית בפעולה שבוצעה

ה. את ערך ה MSB (Most Significant Bit) בתוצאה המתקבלת.

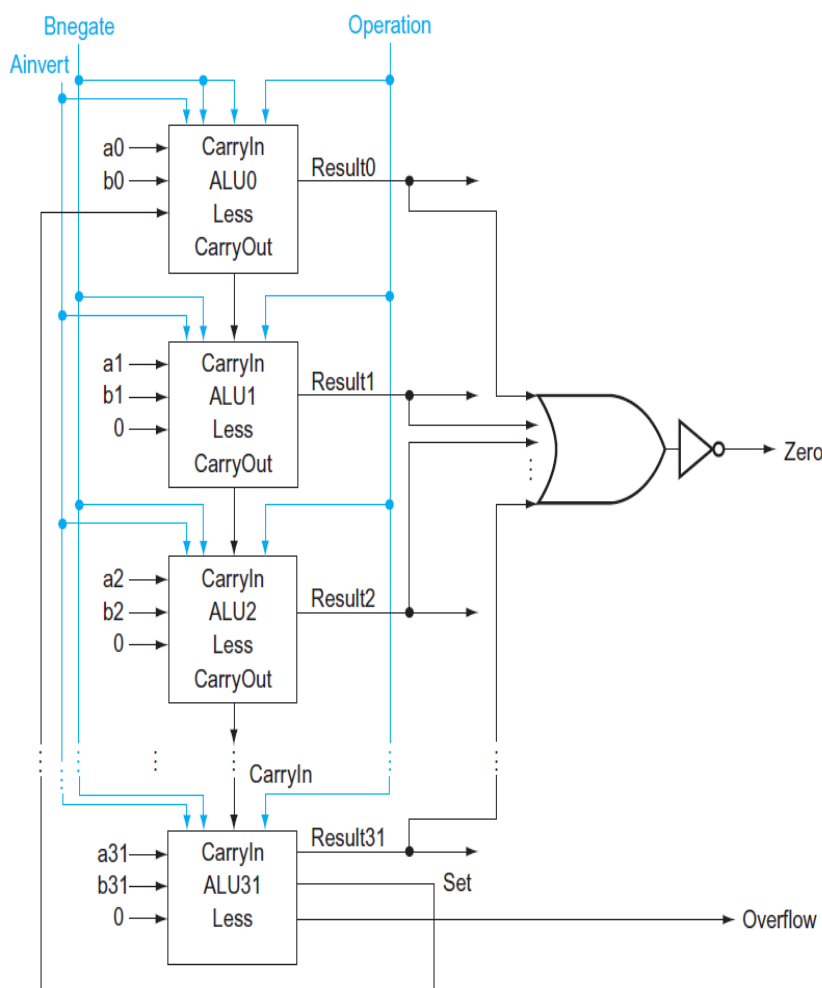


FIGURE B.5.12 The final 32-bit ALU. This adds a Zero detector to Figure B.5.11.

חלק ב' – שאלות פתוחות (76 נקודות)

בחלק זה 2 שאלות. יש לענות על שאלות אלה בטופס המבחן בהתאם להנחיות בסעיפים השונים.

שאלה 7 (51 נקודות)

נתון קטע הקוד שלהלן, באסמבלר של ה-MIPS:

Address	Code	Basic	Source
0x50400010	0x01094822	sub \$9,\$8,\$9	34 sub \$9, \$8, \$9
0x50400014	0x21280200	addi \$8,\$9,0x00000200	35 addi \$8, \$9, 0x200
0x50400018	0x8d090200	lw \$9,0x00000200(\$8)	36 lw \$9, 0x200(\$8)
0x5040001c	0x01e98820	add \$17,\$15,\$9	37 add \$17, \$15, \$9
0x50400020	0x00895022	sub \$10,\$4,\$9	38 sub \$10, \$4, \$9

- ערך כל אוגר בתחילת ביצוע הקוד הינו הערך 0x200 בתוספת מספר האוגר כפול 0x20
 - אוגר 1: ערכו 0x00000220 ($0x200 + 0x1 \cdot 0x20$)
 - אוגר 2: ערכו 0x00000240
 -
 - אוגר 10: ערכו 0x00000340 ($0x200 + 0xa \cdot 0x20$)
 - ...
 - אוגר 16: ערכו 0x00000400 ($0x200 + 0x10 \cdot 0x20$)
 - ...
- נתון שניתן לגשת לכל מרחב הזיכרון בפקודות SW ו-LW (כמובן, כל עוד הכתובות מתחלקות ב-4)
- ערך 0x1000 **המילים** הנמוכות בזיכרון, מאותחל באופן שכל מילה מאותחלת לכתובתה **במילים**:
 - מילה בכתובת 0 במילים (בתיים 0-3 בזיכרון) מאותחלת לערך 0
 - מילה בכתובת 1 במילים (בתיים 4-7 בזיכרון) מאותחלת לערך 1
 -
 - מילה בכתובת 0x100 במילים (בתיים 0x400-0x403 בזיכרון) מאותחלת לערך 0x100
 - ...
 - מילה בכתובת 0xFFF במילים (בתיים 0x3ffc-0x3fff בזיכרון) מאותחלת לערך 0xfff

נתונים אלה תקפים לכל סעיפי שאלה 7

7.1 (2 נקודות):

מה ערכו של אוגר \$8 לאחר פקודה בשורה 35 (פקודת addi בשורה השנייה)?

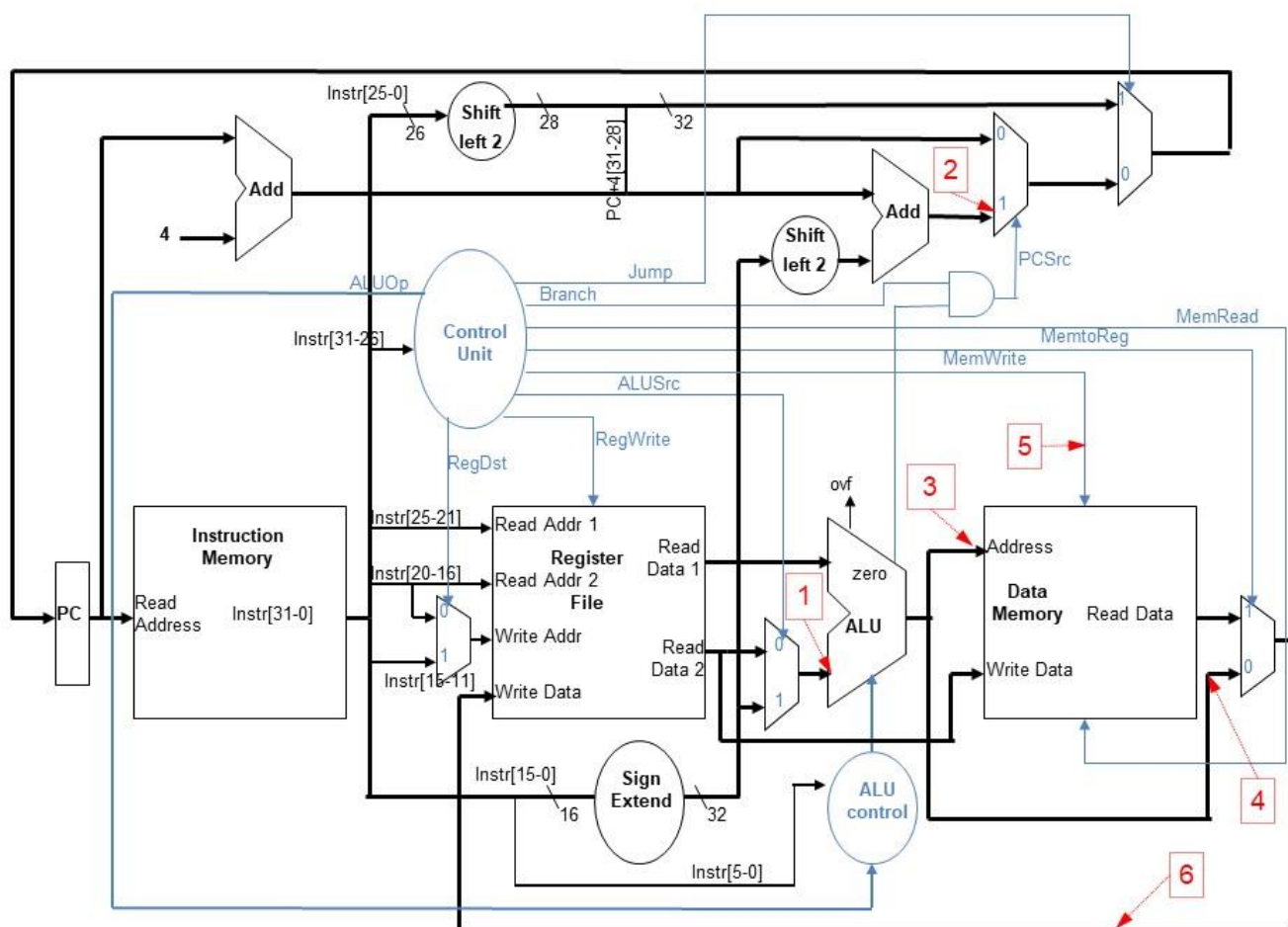
0x000001e0

7.2 (18 נקודות):

בתרשים שלהלן מעבד חד-מחזורי. על התרשים 6 סימונים (ממוספרים מ-1 עד 6, עם חץ המצביע על קו נתונים במעבד).

עליכם לסמן בטבלה שבעמוד זה את הערכים העוברים על הקווים המסומנים **בבסיס הקסה** (XX נק' לסימון), בהנחה שהמידע נבדק לקראת סוף פעימת השעון בביצוע הפקודה השלישית בקוד (שורה 36):

lw \$9, 0x200(\$8)



ערך לא ידוע יש לסמן ב-X. ניתן להיעזר בנתונים מטבלאות 4.12, 4.18 בספר הלימוד.

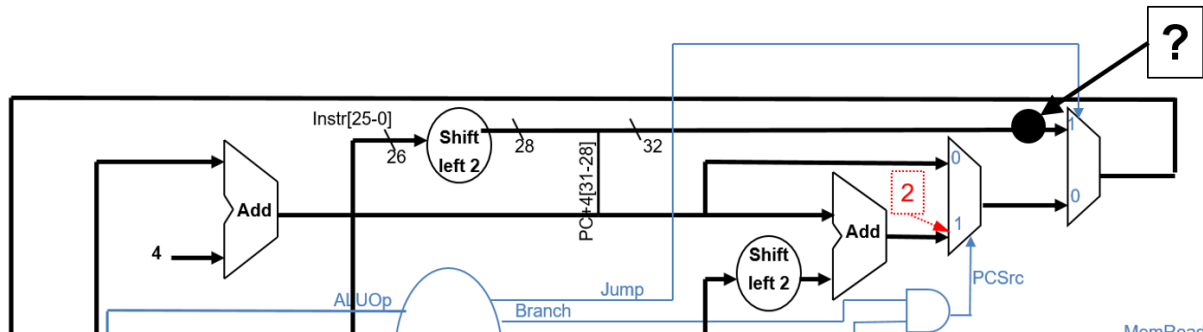
3 נקודות לכל סימון

נקודה	ערך (בהקסה)
1	0x00000200
2	$pc+4+imm*4 = 0x5040081c$ $0x50400018+4+0x200*4$
3	0x000003e0

נקודה	ערך (בהקסה)
4	0x000003e0 (תוצאת ALU שנכנסת כ address מגיעה גם לכאן)
5	0 (זר פקודת LW, ולכן memWrite שווה 0)
6	0x000000f8 (ערך המילה בכתובת שווה לערך נקודה 3 מחולק ב-4)

7.3 (4 נקודות)

בהמשך לסעיף 7.2, בעת ביצוע הפקודה השלישית (LW) – מה יהיה הערך על הקו בנקודה השחורה המסומנת עם "סימן השאלה"?



0x54240800 (מחושב לפי סיביות 0-25 בפקודה 0x1090200, הזזה ב-2 מקומות שמאלה, והשלמת 4 סיביות 28-31 : 5)

האם תתבצע קפיצה לכתובת זו? יש לנמק להסביר בקצרה מה מאפשר/מונע את הקפיצה

	לא תתבצע קפיצה.
	כתובת זו הינה כתובת החישוב של Jump. מאחר ודגל Jump כבוי
	בפקודה זו, ה mux אליו נכנס ערך זה – לא "יבחר" את הערך הזה,
	אלא זה שחושב ב $pc+4$

7.4 (4 נקודות)

כעת, בזמן ביצוע פקודת LW בלבד, עקב תקלה ב-2 יחידות Shift Left 2 (לפני חישוב כתובת Jump, ולפני חישוב כתובת דילוג Branch – ראו בתרשים בסעיף 7.3), יחידות אלא מבצעות Shift Left 3

עקב תקלה זו: מה ההשפעה הישירה של התקלה? האם תהיה פגיעה בביצוע פקודת ה LW? יש לנמק את ההשפעה.

האם יש השפעה על תשובותיך בסעיפים 7.2 ו-7.3? מה ההשפעות? יש לציין איזו נקודה בדיוק הושפעה, לנמק עם התייחסות לערכים המשתנים, היכן משתנים, וההשפעות המלאות שלהן.

אין לתקלה זו השפעה על ביצוע פקודת LW. זאת מאחר, והנתונים הרלוונטיים לביצוע פקודת LW לא משתמשים בערכים העוברים על גבי הקווים עם התקלה. דגלי הבקרה של הפקודה מונעים מהערכים "הלא רלוונטיים" לעבור את ה MUX-ים אליהם נכנסים, ולפיכך הפקודה לא נפגעת.
בסעיף 7.2: נקודה (2) תקבל ערך אחר. הערך יהיה: $pc+4+imm*4 = 0x5040101c$
$0x50400018+4+0x200*8$
בסעיף 7.3: $0x58481000$ (מחושב לפי סיביות 0-25 בפקודה $0x1090200$, הזזה ב-3 מקומות שמאלה, והשלמת 4 סיביות גבוהות 28-31 : 5)

7.5 (23 נקודות)

התרשים שלפניכם בעמוד הבא מבוסס על תרשים 4.60 בספר. התרשים, מתאר את מצב מעבד הצנרת בסוף פעימת השעון החמישית בביצוע הקוד בשאלה, על פי הנתונים המופיעים בתחילת השאלה (שימו לב, אירוע התקלה בסעיף 7.4 אינו רלוונטי לסעיף זה!)

תזכורת לקטע הקוד:

Address	Code	Basic	Source
0x50400010	0x01094822	sub \$9,\$8,\$9	34 sub \$9, \$8, \$9
0x50400014	0x21280200	addi \$8,\$9,0x00000200	35 addi \$8, \$9, 0x200
0x50400018	0x8d090200	lw \$9,0x00000200(\$8)	36 lw \$9, 0x200(\$8)
0x5040001c	0x01e98820	add \$17,\$15,\$9	37 add \$17, \$15, \$9
0x50400020	0x00895022	sub \$10,\$4,\$9	38 sub \$10, \$4, \$9

7.5.1 (3 נקודות)

השלימו, על גבי התרשים בעמוד הבא, במקום המיועד לכך (חלק עליון של התרשים), את פריסת הפקודות במעבד, נכון למחזור שעון 5 של ביצוע רצף הפקודות הנ"ל.

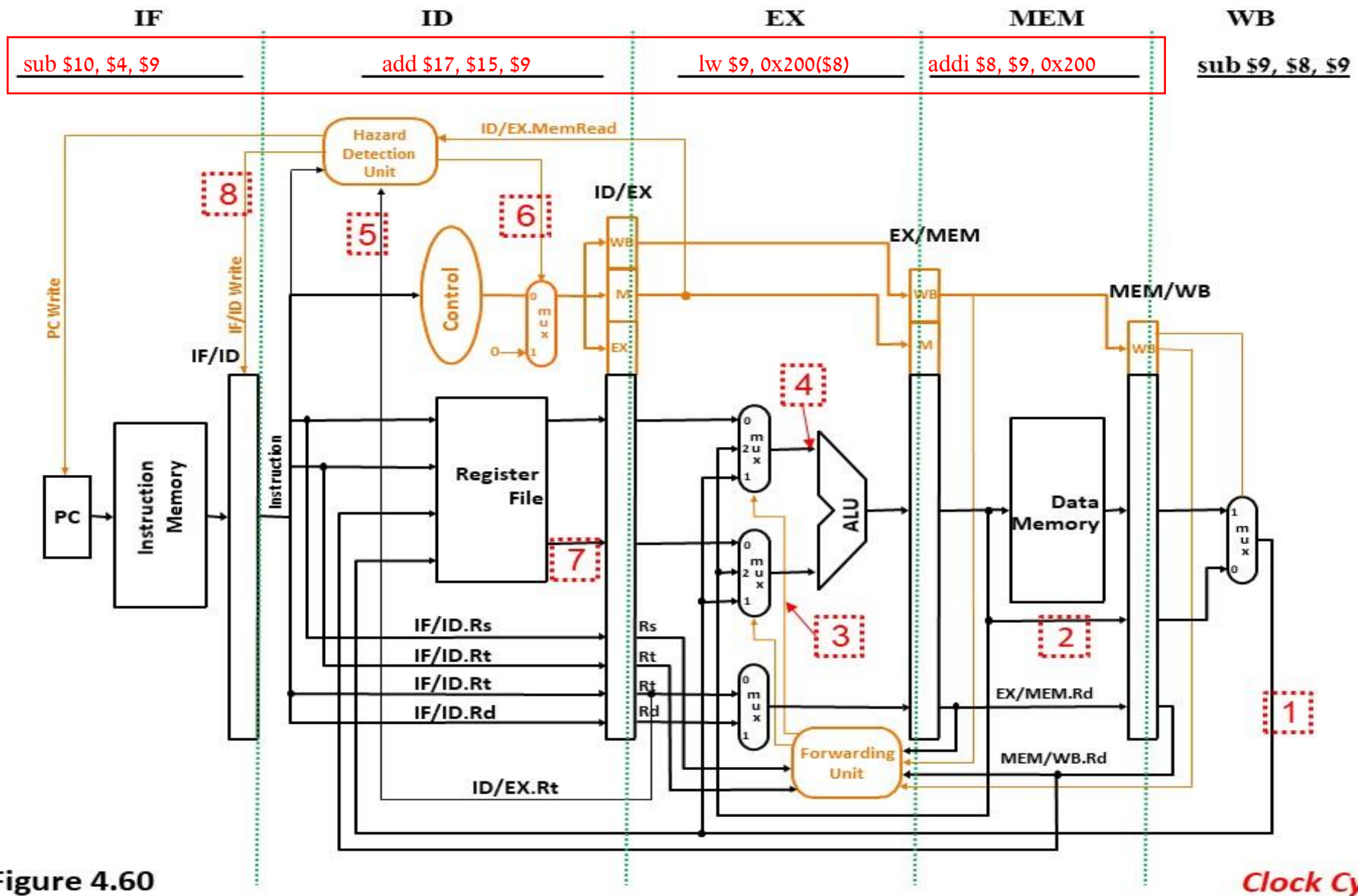
ניתן להניח סיכוני הנתונים בהרצת הקוד נפתרים על ידי יחידת העברה קדימה ויחידת איתור הסיכונים (Forwarding Unit, HDU) וחצית מקבץ האוגרים.

7.5.2 (16 נקודות)

על גבי התרשים מסומנות 8 נקודות. יש למלא בטבלה את הערכים של קווים אלה.

יש לרשום את הערכים בבסיס הקסה (2 נקודות לכל סימון).

נקודה	ערך (בהקסה)	נקודה	ערך (בהקסה)
1	0xfffffe0	5	0x9 (מספר אוגר מטרה של פקודת ה LW לבדיקת ה HDU)
2	0x000003e0	6	0x1 (כדי לבחור את היציאה עם ערכי 0, לאיפוס הדגלים ויצירת bubble במחזור שעון 6)
3	0x2 (יחידת forwarding Unit מוציאה 2 כדי לקדם ערך משלב ה mem)	7	0xfffffe0 (ערך אוגר \$9 מפקודת sub שב-wb, עקב "חציית מקבץ האוגרים")
4	0x000003e8 (ערך מקודם משלב ה mem עבור אוגר \$8)	8	0x0 (למניעת כתיבה על אוגר צנרת IF/ID, כדי ליצור bubble במחזור שעון 6)



7.5.3 (4 נקודות)

מה תהיה פריסת הפקודות במעבד, בשלבים השונים שלו, במחזור שעון 6 (מחזור שעון אחד לאחר מחזור השעון שבסעיף 7.5.2)? יש להשלים את פריסת הפקודות בתרשים המצ"ב.

IF	ID	EX	MEM	WB
<u>sub \$10, \$4, \$9</u>	<u>add \$17, \$15, \$9</u>	<u>bubble</u>	<u>lw \$9, 0x200(\$8)</u>	<u>addi \$8, \$9, 0x200</u>

כעת, מה יהיו הערכים בנקודות (5), (6) ו-(7) שעל גבי התרשים בסעיף קודם, עבור מחזור שעון 6? יש לציין את הערך בכל נקודה, ולהוסיף משפט הסבר מדוע זה הערך.

האם הערך בנקודה (7) הינו הערך איתו תבוצע הפקודה, שרשמת בשלב ה ID, כאשר תבצע בשלב ה EX? יש לציין האם כן/לא, ולהסביר את תשובתך בהתייחסות למה שיקרה במחזור שעון 7, באופן מפורט לערך זה. יש להתייחס לכל ההיבטים הרלוונטיים של יחידת forwarding unit (כולל מקרי הבדיקה כפי שמפורט בנספח, יחידת HDU, בהתאם למה שיתרחש בפועל במחזור שעון 7.

נקודה (5): 0x11 (מספר אוגר \$17) שהוא אוגר המטרה של פקודת add שעבר לשלב ה EX עם ה bubble

נקודה (6): 0x0 (במחזור שעון זה הדגלים של פקודת add עוברים במלואם)

נקודה (7): ללא שינוי - 0xffffffe0 (ערך אוגר \$9 כפי שחושב בפקודת sub שיצאה מהצנרת, עדיין אין ערך מעודכן לאוגר זה – בהתאם לצפוי בפקודת LW)

הערך בנקודה (7) אינו הערך הסופי עימו תבוצע פקודת ה ADD כאשר תגיע לשלב EX, במחזור שעון 7. במחזור שעון 7, פקודת LW תהיה ב WB, ופקודת ה ADD תהיה בשלב EX. במחזור שעון זה, תזדה יחידת ה forwarding unit צורך לקדם את ערך אוגר \$9 משלב WB אל שלב EX (בין ה LW לבין ADD). יחידת forwarding unit תזזה את ה forwarding לפי מקרה 2b, ו-mux control של ForwardB יקבל את הערך 01.

שאלה 8 (25 נקודות)

כתבו פונקציה בשם `maxLengthInRowCol` המקבלת 4 פרמטרים:

- `$a0` - כתובת תחילת מטריצה ריבועית (כל איבר בגודל Byte)
- `$a1` - מספר שלם N , המייצג את ממדי המטריצה ($N \times N$)
- `$a2` - מספר שלם R – המייצג שורה במטריצה
- `$a3` - מספר שלם C – המייצג עמודה במטריצה

הפונקציה תספור, החל ממיקום $[R][C]$ במטריצה – את אורך רצף ערכים זהים ממקום זה (כולל) לאורך השורה (כלומר, מהמקום עצמו וימינה), את אורך רצף ערכים זהים ממקום זה בעמודה (כלומר, מהמקום עצמו ולמטה) X .

הפונקציה תחזיר את אורך הרצף הגדול מבין השניים (אם שווה, אז יוחזר גודלו של הרצף הזהה)

	0	1	2	3	4
0	1	2	3	4	1
1	0	2	2	2	1
2	1	2	9	0	1
3	2	0	2	4	1
4	1	2	1	1	1

דוגמה – עבור המטריצה משמאל:

- עבור מיקום $[0][4]$: יוחזר הערך 5 (ממקום זה ימינה יש פעם אחת את הערך 1, ממקום זה למטה יש חמש פעמים 1).
- עבור מיקום $[4][2]$: יוחזר הערך 3 (ממקום זה ימינה יש שלוש פעמים 1, ממקום זה למטה יש פעם אחת 1)
- עבור מיקום $[1][1]$: יוחזר הערך 3 (ממקום זה ימינה יש שלוש פעמים 2, ממקום זה למטה יש שתי פעמים 2)

הנחיות לפתרון:

- יש להקפיד שלא לחרוג מגבולות המערך
- יש לכתוב את הפרוצדורה, בלבד (אין צורך ולא נדרש לכתוב תוכנית ראשית!)
- יש לכתוב בהתאם לכללי העבודה שלמדנו בכתיבת פרוצדורות.
- ניתן להשתמש בכל הפקודות כפי ש MARS מקבל (כלומר...עובר קומפילציה ב MARS).
- הקפידו על כך, והקפידו "שלא להמציא" פקודות שאינן קיימות.
- הפתרון חייב להשתמש אך ורק בפרמטרים שהתקבלו, **ללא** כל שימוש/תוספת של משתנים במקטע הנתונים (.data).
- הפתרון צריך להתאים לכל גודל מערך דו ממדי ריבועי (ולא רק למערך של 5×5 שניתן בדוגמה).
- ניתן להניח שהערכים בפרמטרים (אוגרים `$a0.... $a3`) תקינים – כלומר, **אין צורך** לבצע בדיקות תקינות.

פתרון שאלה 8:

the solution

maxLengthInRowCol:

\$a0 - matrix address (a square matrix)

\$a1 - matrix size - n (nxn)

\$a2 - row index

\$a3 - col index

mul \$t0, \$a2, \$a1 # how much bytes

add \$t0, \$t0, \$a0

add \$t0, \$t0, \$a3 # start position

move \$t9, \$t0 # save start position

li \$t1, 0 # initialize counter for row: from start position and right

move \$t2, \$a3 # current column

lb \$t3, 0(\$t0) # value of start position for comparing

countRow:

bge \$t2, \$a1, endCountRow # row ended - break the loop

lb \$t4, 0(\$t0) # get current value

bne \$t3, \$t4, endCountRow # not equal - break the loop

addi \$t1, \$t1, 1 # increment row counter

addi \$t2, \$t2, 1 # increment column #: for verify row end

addi \$t0, \$t0, 1 # next row position

j countRow

endCountRow:

li \$v0, 0 # initialize counter for column: from start position and down

move \$t2, \$a2 # current row

countColumn:

bge \$t2, \$a1, endCountCol # column ended - bread the loop

lb \$t4, 0(\$t9) # get current value

bne \$t3, \$t4, endCountCol # not equal - break the loop

addi \$v0, \$v0, 1 # increment column counter

addi \$t2, \$t2, 1 # increment row #: for verify column end

add \$t9, \$t9, \$a1 # add row size

j countColumn

endCountCol:

bge \$v0, \$t1, maxLengthEnd # if column counter bigger - return this counter

move \$v0, \$t1 # row counter is bigger - return this counter

maxLengthEnd:

jr \$ra

[illegible]

חלק א': (24 נקודות)

בחלק זה 6 שאלות רב ברירה. יש לבחור את התשובה הנכונה מבין התשובות האפשריות. משקל כל שאלה 4 נקודות. יש לסמן באופן ברור ב-X, בטבלה שלפניכם, את התשובה הנכונה הערה: יש לסמן רק אפשרות אחת לכל שאלה! ייבדקו רק הסימונים בטבלה. כל רישום ליד השאלה עצמה הינו בחזקת טיוטה, ולא ייבדק!!!

שאלה	א	ב	ג	ד	ה
1					
2					
3					
4					
5					
6					

שאלה 1

נתונים שני מעבדים שונים: P1, P2. בשני המעבדים קיים אותו אוסף פקודות. ניתן לחלק את הפקודות ל-4 קבוצות שונות: A, B, C ו-D.

קבוצה	% פקודות	CPI במעבד P1	CPI במעבד P2
A	20%	6	6
B	30%	6	4
C	10%	6	2
D	40%	3	2

נתון כי קצב שעון של P1 הוא 3GHz

נתון כי קצב שעון של P2 הוא 2GHz

בטבלה משמאל נתון מספר מחזורי שעון לכל קבוצת פקודות, עבור כל מעבד. כמו כן, נתון בטבלה אחוז יחסי של הפקודות, עבור כל קבוצת פקודות, מכלל פקודות התוכנית.

בהינתן תוכנית X בת 10^8 פקודות, המחולקת לקבוצות הפקודות בהתאם ל "% הפקודות" הרשום בטבלה: מה מדד ההאצה, ובאיזה מעבד התוכנית תרוץ מהר יותר?

א. לא ניתן לדעת על בסיס הנתונים בשאלה

ב. מדד ההאצה הינו 1.000, על בסיס הנתונים בשאלה זו – יש זהות בין המעבדים

ג. מדד ההאצה הינו 1.063, על בסיס הנתונים בשאלה זו – מעבד P1 מהיר יותר

ד. מדד ההאצה הינו 1.063, על בסיס הנתונים בשאלה זו – מעבד P2 מהיר יותר

ה. מדד ההאצה הינו 0.824, על בסיס הנתונים בשאלה זו – מעבד P2 מהיר יותר

שאלות 2 ו-3 מתייחסות לתמונת סגמנט הקוד שלהלן, שנלקחה מה-MARS:

Address	Code	Basic	Source
0x00400000	0x3c080b00	lui \$8,0x000000b00	12: lui \$t0, 0xb00
0x00400004	0x00084e02	srl \$9,\$8,0x00000018	13: srl \$t1, \$t0, 24
0x00400008	0x00084200	sll \$8,\$8,0x00000008	14: sll \$t0, \$t0, 8
0x0040000c	0x01094025	or \$8,\$8,\$9	15: or \$t0, \$t0, \$t1
0x00400010	0x310800f3	andi \$8,\$8,0x000000f3	16: andi \$t0, \$t0, 0xf3
0x00400014	0x20010003	addi \$1,\$0,0x00000003	17: bne \$t0, 3, somewhere1
0x00400018	0x14280007	bne \$1,\$8,0x00000007	

שאלה 2

מה ניתן להגיד על הפקודה בשורה 17 בקוד המקור: **?bne \$t0, 3, somewhere1**

- זו פקודת פסיאודו, ניתן לדעת שלא תתבצע קפיצה לתווית somewhere1
- זו פקודת אסמבלר בסיסית, ניתן לדעת שלא תתבצע קפיצה לתווית somewhere1
- זו פקודת פסיאודו, ניתן לדעת שתתבצע קפיצה לתווית somewhere1
- זו פקודת אסמבלר בסיסית, ניתן לדעת שתתבצע קפיצה לתווית somewhere1
- זו פקודת פסיאודו, לא ניתן לדעת לפי נתוני קטע הקוד האם תתבצע קפיצה לחווית somewhere1

שאלה 3

על בסיס הנתונים בתמונת סגמנט הקוד, מה כתובת תווית somewhere1?

- 0x00400018
- 0x0040001c
- 0x00400030
- 0x00400034
- 0x00400038

שאלה 4

נתון זיכרון מטמון עם המאפיינים הבאים:

- מאורגן בשיטת 4-Way Set Associative
- גודל כל בלוק נתונים (data) הינו 64 בתים
- נתון כי יש 16,384 (2^{14}) בלוקים של נתונים במטמון
- הזיכרון ממפה כתובות של 32 סיביות בזיכרון הראשי

מה גודלם של שדות ה- Index וה- Tag במטמון זה?

- גודל שדה ה- Index הוא 14 סיביות, גודל שדה ה- Tag הוא 10
- גודל שדה ה- Index הוא 12 סיביות, גודל שדה ה- Tag הוא 14
- גודל שדה ה- Index הוא 12 סיביות, גודל שדה ה- Tag הוא 16
- גודל שדה ה- Index הוא 14 סיביות, גודל שדה ה- Tag הוא 14
- לא ניתן לדעת על בסיס נתוני השאלה

שאלה 5

נתון הערך 6.325-

מהו ערך ההקסה המייצג מספר רציונאלי float בהתאם ל- IEEE 754.

- 0x014C0000
- 0x814C0000
- 0x40CC0000
- 0xC0CC0000
- 0xC0E60000

שאלה 6

על סמך התרשים הבא (תרשים B.5.12 בספר בנספח B) איזה מידע לא ניתן לדעת רק על סמך המידע היוצא מקופסה ALU31, לגבי ביצוע תוצאות של פעולות אריתמטיות ב-ALU:

א. את ערך Result31

ב. האם תוצאת הפעולה שבוצעה היא שלילית (במשמעות של המסלים 2)

ג. האם תוצאת הפעולה שבוצעה היא 0 (כלומר, נדלק חיווי Zero)

ד. האם הייתה גלישה אריתמטית בפעולה שבוצעה

ה. את ערך ה MSB (Most Significant Bit) בתוצאה המתקבלת.

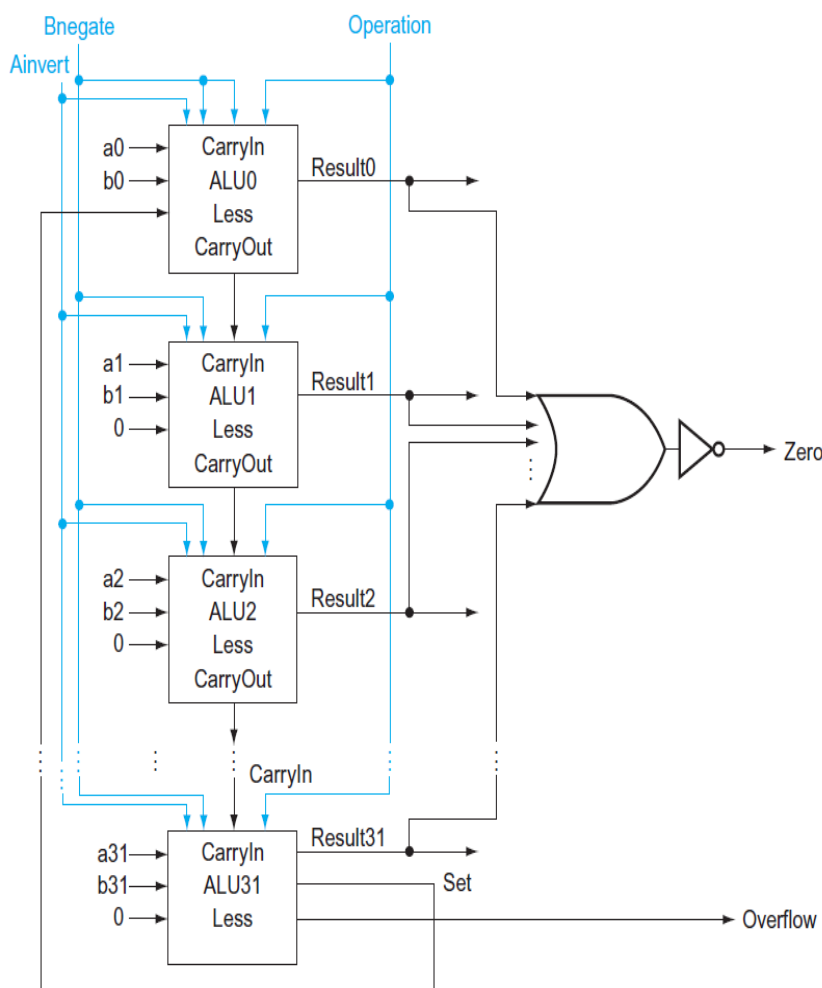


FIGURE B.5.12 The final 32-bit ALU. This adds a Zero detector to Figure B.5.11.

חלק ב' – שאלות פתוחות (76 נקודות)

בחלק זה 2 שאלות. יש לענות על שאלות אלה בטופס המבחן בהתאם להנחיות בסעיפים השונים.

שאלה 7 (51 נקודות)

נתון קטע הקוד שלהלן, באסמבלר של ה-MIPS:

Address	Code	Basic	Source
0x50400010	0x01094822	sub \$9,\$8,\$9	34 sub \$9, \$8, \$9
0x50400014	0x21280200	addi \$8,\$9,0x00000200	35 addi \$8, \$9, 0x200
0x50400018	0x8d090200	lw \$9,0x00000200(\$8)	36 lw \$9, 0x200(\$8)
0x5040001c	0x01e98820	add \$17,\$15,\$9	37 add \$17, \$15, \$9
0x50400020	0x00895022	sub \$10,\$4,\$9	38 sub \$10, \$4, \$9

- ערך כל אוגר בתחילת ביצוע הקוד הינו הערך 0x200 בתוספת מספר האוגר כפול 0x20
 - אוגר 1: ערכו 0x00000220 ($0x200 + 0x1 \cdot 0x20$)
 - אוגר 2: ערכו 0x00000240
 -
 - אוגר 10: ערכו 0x00000340 ($0x200 + 0xa \cdot 0x20$)
 - ...
 - אוגר 16: ערכו 0x00000400 ($0x200 + 0x10 \cdot 0x20$)
 - ...
- נתון שניתן לגשת לכל מרחב הזיכרון בפקודות SW ו-LW (כמובן, כל עוד הכתובות מתחלקות ב-4)
- ערך 0x1000 **המילים** הנמוכות בזיכרון, מאותחל באופן שכל מילה מאותחלת לכתובתה **במילים**:
 - מילה בכתובת 0 במילים (בתיים 0-3 בזיכרון) מאותחלת לערך 0
 - מילה בכתובת 1 במילים (בתיים 4-7 בזיכרון) מאותחלת לערך 1
 -
 - מילה בכתובת 0x100 במילים (בתיים 0x400-0x403 בזיכרון) מאותחלת לערך 0x100
 - ...
 - מילה בכתובת 0xFFF במילים (בתיים 0x3ffc-0x3fff בזיכרון) מאותחלת לערך 0xfff

נתונים אלה תקפים לכל סעיפי שאלה 7

7.1 (2 נקודות):

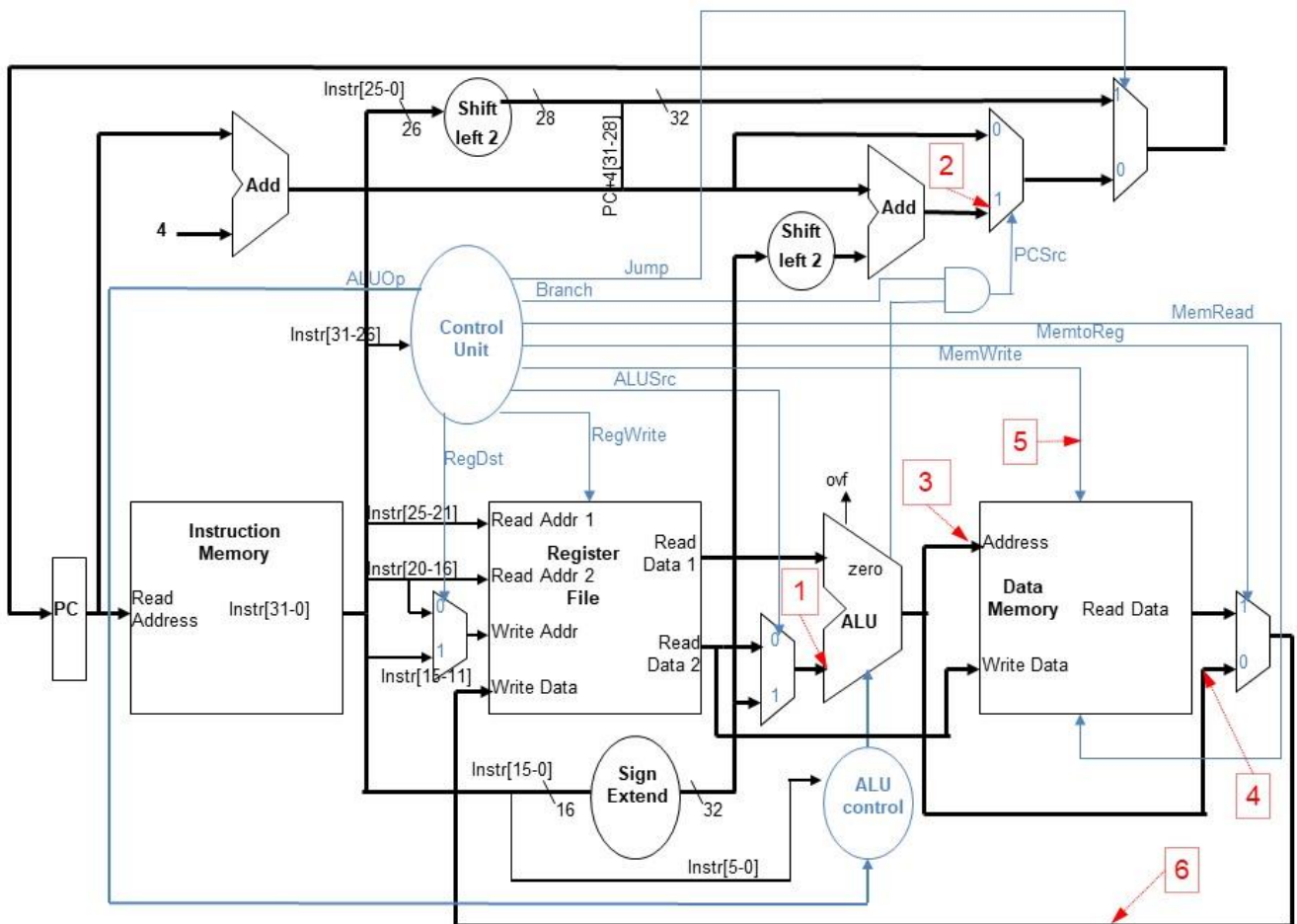
מה ערכו של אוגר \$8 לאחר פקודה בשורה 35 (פקודת addi בשורה השנייה)?

7.2 (18 נקודות):

בתרשים שלהלן מעבד חד-מחזורי. על התרשים 6 סימונים (ממוספרים מ-1 עד 6, עם חץ המצביע על קו נתונים במעבד).

עליכם לסמן בטבלה שבעמוד זה את הערכים העוברים על הקווים המסומנים **בבסיס הקסה** (XX נק' לסימון), בהנחה שהמידע נבדק לקראת סוף פעימת השעון בביצוע הפקודה השלישית בקוד (שורה 36):

lw \$9, 0x200(\$8)



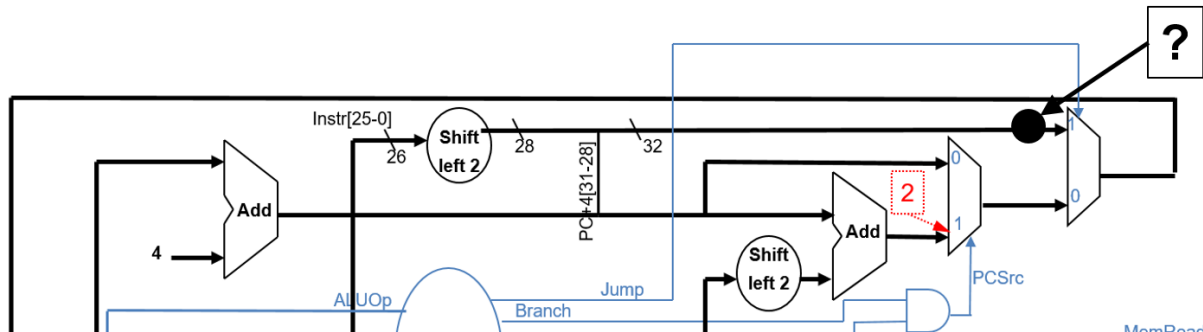
ערך לא ידוע יש לסמן ב-X. ניתן להיעזר בנתונים מטבלאות 4.12, 4.18 בספר הלימוד.

3 נקודות לכל סימון

נקודה	ערך (בהקסה)
1	
2	
3	

נקודה	ערך (בהקסה)
4	
5	
6	

בהמשך לסעיף 7.2, בעת ביצוע הפקודה השלישית (LW) – מה יהיה הערך על הקו בנקודה השחורה המסומנת עם "סימן השאלה"?



האם תתבצע קפיצה לכתובת זו? יש לנמק להסביר בקצרה מה מאפשר/מונע את הקפיצה

כעת, **בזמן ביצוע פקודת LW בלבד**, עקב **תקלה** ב- 2 יחידות Shift Left 2 (לפני חישוב כתובת Jump, ולפני חישוב כתובת דילוג Branch – ראו בתרשים בסעיף 7.3), יחידות אלא מבצעות **Shift Left 3**

עקב תקלה זו: מה ההשפעה הישירה של התקלה? האם תהיה פגיעה בביצוע פקודת ה-LW? יש לנמק את ההשפעה.

האם יש השפעה על תשובותיך בסעיפים 7.2 ו- 7.3? מה ההשפעות? יש לציין איזו נקודה בדיוק הושפעה, לנמק עם התייחסות לערכים המשתנים, היכן משתנים, וההשפעות המלאות שלהן.

7.5 (23 נקודות)

התרשים שלפניכם בעמוד הבא מבוסס על תרשים 4.60 בספר. התרשים, מתאר את מצב מעבד הצנרת בסוף פעימת השעון החמישית בביצוע הקוד בשאלה, על פי הנתונים המופיעים בתחילת השאלה (שימו לב, אירוע התקלה בסעיף 7.4 אינו רלוונטי לסעיף זה!)

תזכורת לקטע הקוד:

Address	Code	Basic	Source
0x50400010	0x01094822	sub \$9,\$8,\$9	34 sub \$9, \$8, \$9
0x50400014	0x21280200	addi \$8,\$9,0x00000200	35 addi \$8, \$9, 0x200
0x50400018	0x8d090200	lw \$9,0x00000200(\$8)	36 lw \$9, 0x200(\$8)
0x5040001c	0x01e98820	add \$17,\$15,\$9	37 add \$17, \$15, \$9
0x50400020	0x00895022	sub \$10,\$4,\$9	38 sub \$10, \$4, \$9

7.5.1 (3 נקודות)

השלימו, על גבי התרשים בעמוד הבא, במקום המיועד לכך (חלק עליון של התרשים), את פריסת הפקודות במעבד, נכון למחזור שעון 5 של ביצוע רצף הפקודות הנ"ל.

ניתן להניח סיכוני הנתונים בהרצת הקוד נפתרים על ידי יחידת העברה קדימה ויחידת איתור הסיכונים (Forwarding Unit, HDU) וחצית מקבץ האוגרים.

7.5.2 (16 נקודות)

על גבי התרשים מסומנות 8 נקודות. יש למלא בטבלה את הערכים של קווים אלה.

יש לרשום את הערכים בבסיס הקסה (2 נקודות לכל סימון).

נקודה	ערך (בהקסה)	נקודה	ערך (בהקסה)
1		5	
2		6	
3		7	
4		8	

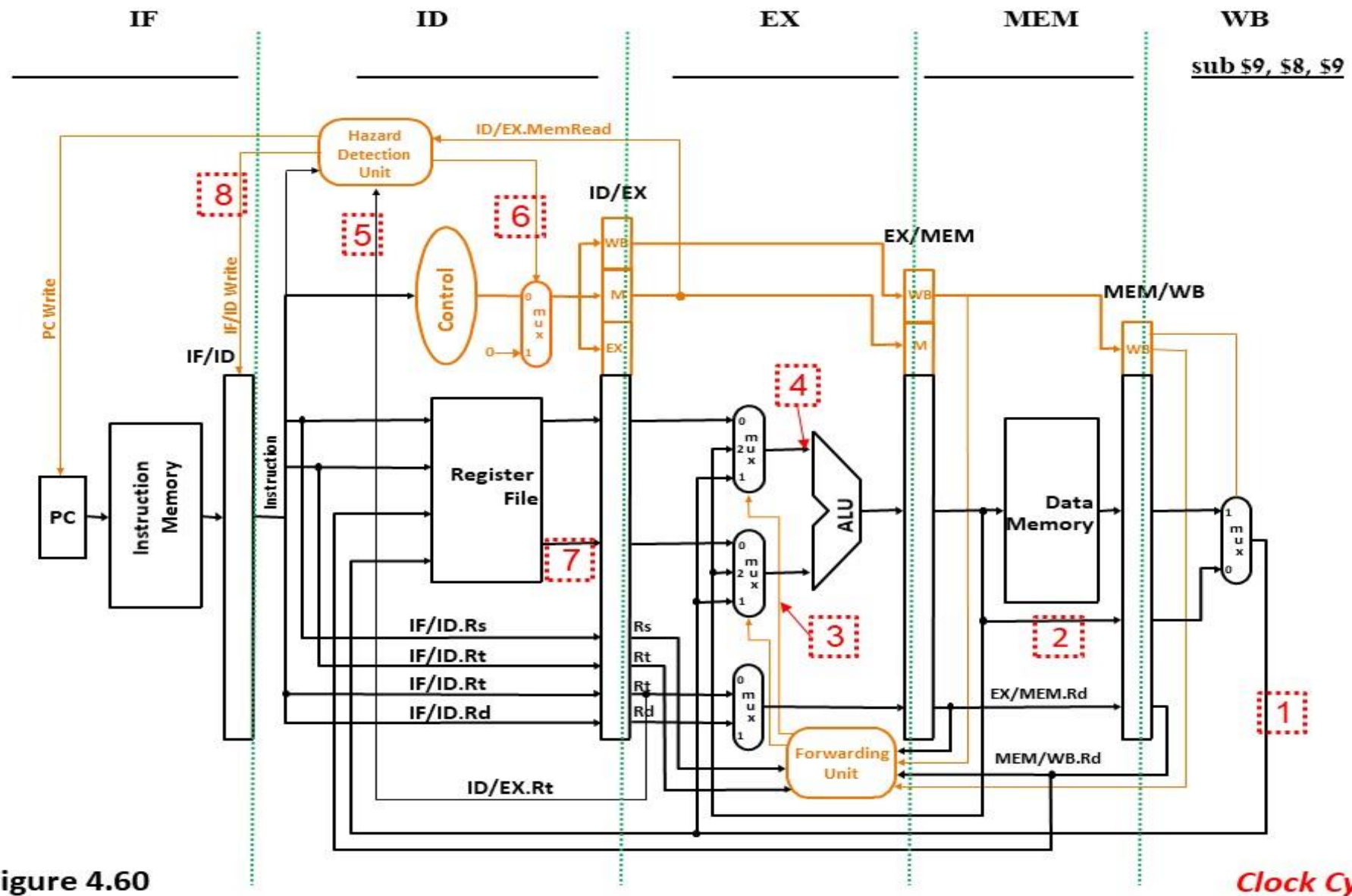


Figure 4.60

Clock Cycle 5

מה תהיה פריסת הפקודות במעבד, בשלבים השונים שלו, במחזור שעון 6 (מחזור שעון אחד לאחר מחזור השעון שבסעיף 7.5.2)? יש להשלים את פריסת הפקודות בתרשים המצ"ב.

שאלון 10124 - X ארכיטקטורת חומרת ושפת סף, 2023, סמסטר א

שאלה 8 (25 נקודות)

כתבו פונקציה בשם `maxLengthInRowCol` המקבלת 4 פרמטרים:

- `$a0` - כתובת תחילת מטריצה ריבועית (כל איבר בגודל Byte)
- `$a1` - מספר שלם N , המייצג את ממדי המטריצה ($N \times N$)
- `$a2` - מספר שלם R – המייצג שורה במטריצה
- `$a3` - מספר שלם C – המייצג עמודה במטריצה

הפונקציה תספור, החל ממיקום $[R][C]$ במטריצה – את אורך רצף ערכים זהים ממקום זה (כולל) לאורך השורה (כלומר, מהמקום עצמו וימינה), את אורך רצף ערכים זהים ממקום זה בעמודה (כלומר, מהמקום עצמו ולמטה) X .

הפונקציה **תחזיר** את אורך הרצף הגדול מבין השניים (אם שווה, אז יוחזר גודלו של הרצף הזהה)

	0	1	2	3	4
0	1	2	3	4	1
1	0	2	2	2	1
2	1	2	9	0	1
3	2	0	2	4	1
4	1	2	1	1	1

דוגמה – עבור המטריצה משמאל:

- עבור מיקום $[0][4]$: יוחזר הערך 5 (ממקום זה ימינה יש פעם אחת את הערך 1, ממקום זה למטה יש חמש פעמים 1).
- עבור מיקום $[4][2]$: יוחזר הערך 3 (ממקום זה ימינה יש שלוש פעמים 1, ממקום זה למטה יש פעם אחת 1)
- עבור מיקום $[1][1]$: יוחזר הערך 3 (ממקום זה ימינה יש שלוש פעמים 2, ממקום זה למטה יש שתי פעמים 2)

הנחיות לפתרון:

- יש להקפיד שלא לחרוג מגבולות המערך
- יש לכתוב את הפרוצדורה, בלבד (אין צורך ולא נדרש לכתוב תוכנית ראשית!)
- יש לכתוב בהתאם לכללי העבודה שלמדנו בכתיבת פרוצדורות.
- ניתן להשתמש בכל הפקודות כפי ש MARS מקבל (כלומר...עובר קומפילציה ב MARS).
- הקפידו על כך, והקפידו "שלא להמציא" פקודות שאינן קיימות.
- הפתרון חייב להשתמש אך ורק בפרמטרים שהתקבלו, **ללא** כל שימוש/תוספת של משתנים במקטע הנתונים (.data).
- הפתרון צריך להתאים לכל גודל מערך דו ממדי ריבועי (ולא רק למערך של 5×5 שניתן בדוגמה).
- ניתן להניח שהערכים בפרמטרים (אוגרים `$a0.... $a3`) תקינים – כלומר, **אין צורך** לבצע בדיקות תקינות.

פתרון שאלה 8:

```
##### the solution #####
```

[illegible]

[illegible]