

חלק א': (24 נקודות)

בחלק זה 6 שאלות רב ברירה. יש לבחור את התשובה הנכונה מבין התשובות האפשריות (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					X

שאלה 1

נתון מעבד אשר זמן מחזור שעון שלו הינו $2 \cdot 10^{-10}$.

CPI	% פקודות	קבוצה
6	20%	A
4	30%	B
2	40%	C
8	10%	D

במעבד יש 4 קבוצות של פקודות: A, B, C ו- D. בטבלה משמאל נתון, לכל אחת מקבוצת הפקודות: מספר מחזורי שעון הנדרש לפקודות בקבוצה זו, וכן את החלק היחסי של הפקודות בתוכנית X. נתון שיש בתוכנית X לולאה עם 300 פקודות (הלולאה מכילה פקודות מקבוצות הפקודות משמאל, בהתאם לאחוז המפורט).

נתון כי משך הזמן לביצוע הלולאה הינו 24 שניות.

כמה פעמים עובדת הלולאה?

א. לא ניתן לדעת על בסיס הנתונים בשאלה

ב. הלולאה עובדת 100,000,000 פעמים (10^8)

ג. הלולאה עובדת 80,000,000 פעמים ($8 \cdot 10^7$)

ד. הלולאה עובדת 120,000,000 פעמים ($1.2 \cdot 10^8$)

ה. הלולאה עובדת 10,000,000 פעמים (10^7)

$$CPI = 6 \cdot 0.2 + 4 \cdot 0.3 + 2 \cdot 0.4 + 8 \cdot 0.1 = 4$$

$$CPU_{time} = IC \cdot CPI_1 \cdot CCT$$

$$24 = (300 \cdot X) \cdot 4 \cdot (2 \cdot 10^{-10})$$

$$24 / (300 \cdot 4 \cdot (2 \cdot 10^{-10})) = X$$

$$X = 100,000,000 = 10^8$$

Address	Code	Basic	Source
0x9040001c	0x3c086100	lui \$8,0x00006100	20: lui \$t0, 0x6100
0x90400020	0x00084602	srl \$8,\$8,0x00000018	21: srl \$t0, \$t0, 24
0x90400024	0x310800df	andi \$8,\$8,0x000000df	22: andi \$t0, \$t0, 0xdf
0x90400028	0x20090041	addi \$9,\$0,0x00000041	23: addi \$t1, \$zero, 'A'
0x9040002c	0x11090001	beq \$8,\$9,0x00000001	24: beq \$t0, \$t1, noWhere
0x90400030	0x08100007	j guessMe	25: j guessMe

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

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

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

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

ה. לא ניתן לבצע השוואה בין אוגר \$t0 לאוגר \$t1 מאחר ולאוגר \$t1 יש ערך SCII
\$t0 יש ערך בהקסה, ולכן לא תתבצע קפיצה

ערכו של אוגר \$t0\$ לאחר פקודה 22 הוא 0x41, ערכו של אוגר \$t1\$ לאחר פקודה 23 הוא 0x41, ולפיכך בפקודה בשורה 24 תבוצע קפיצה לתווית noWhere

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

0x90400030 .N

0x0040001c .b

0x90100007 .λ

0x9040001c .T

חישוב התווית, מבוצע לפי ניתוח עמודת ה code של פקודת ה- jump בשורה 25:

0x08100007 → 000010 000001000000000000000000111

ה. לא ניתן לדעת מהנתונים הקיימים

000001000000000000000000111 shift left 2: 0000010000000000000000001100

נוסיף כעת 4 סיביות גבוהות של PC+4, שהן $0x9 \leftarrow 1001$ ונקבל:

1001 0000 0100 0000 0000 0000 0001 1100 = 0x9040001c

נתון זיכרון מטמון עם המאפיינים הבאים:

נתון שיש 2^{13} בלוקים של נתונים, ב- 2
way set associative, ולכן בכל way יש
 2^{12} בלוקים (חילוק ב- 2, 2). מכאן ... גודל
אינדקס הוא 12 סיביות.

לא לשכוח – 2 סיביות של Byte Offset

מכאן, גודל שדה ה- Block Offset הינו:

$$32 - 13 - 12 - 2 = 5$$

כלומר, בבלוק יש 2^5 מילים, 32 מילים
(128 בתים – 2^7)

גודל המטמון בבתים הינו:

$$2^7 * 2^{13} = 2^{20} = 1,048,576$$

- מאורגן בשיטת 2-Way Set Associative
- נתון כי יש 8,192 (2^{13}) בלוקים של נתונים במטמון
- גודל שדה ה- Tag הינו 13 סיביות

מה גודל הנתונים (data) במטמון, בבתים (Bytes)?

א. $2^{19} (524,288)$ בתים

ב. $2^{20} (1,048,576)$ בתים

ג. $2^{21} (2,097,152)$ בתים

ד. $2^{22} (4,194,304)$ בתים

ה. לא ניתן לדעת על בסיס נתוני השאלה

$$0x41350000 = 0100\ 0001\ 0011\ 0101\ 0000\dots$$

$$= 0\ 10000010\ 011010100000\dots$$

נתון ערך הקסה 0x41350000 המייצג מספר רציונאלי float בהתאם ל-IEEE 754.

$$= +\ 130\ (1) + 2^{-2} + 2^{-3} + 2^{-5} + 2^{-7}$$

$$= +\ (130-127) * (1 + 2^{-2} + 2^{-3} + 2^{-5} + 2^{-7})$$

$$= +\ 2^3 * (1 + 2^{-2} + 2^{-3} + 2^{-5} + 2^{-7})$$

$$= +\ 2^3 + 2^1 + 2^0 + 2^{-2} + 2^{-4}$$

$$= +\ 8 + 2 + 1 + 0.25 + 0.0625$$

$$= 11.3125$$

שאלה 5

כאשר נמיר ערך זה לעשרוני, איזה ערך הוא מייצג?

א. 3.3125

ב. -3.3125

ג. 11.3125

ד. -11.3125

ה. 8.3125

שאלה 6

ב MIPS העובד בטכנולוגיית צנרת קו הבקרה IF.Flush (ניתן לראות בתרשימים 4.65 ו-4.66 בספר הקורס) וקו הבקרה IF/ID.Write (ניתן לראות בתרשים 4.60 בספר) פועלים שניהם על אוגר הצנרת IF/ID. מה ההבדל בין קווים אלו?

א. הקו IF.Flush מטפל בסיכוני נתונים וקו IF/ID.Write בסיכוני בקרה.

ב. הקו IF.Flush מנקה את תוכן האוגר IF/ID, מוחק את הפקודה הנמצאת שם והופך אותה לקידוד 0, ואילו הקו IF/ID.Write מאפס רק את ערכי הבקרה של הפקודה, והופך אותה לבועה.

ג. הקו IF/ID.Write מבצע חיסור של 4 מ-PC, ובכך משנה את חישוב כתובת branch target (ולכן נכנס לחלק העליון של IF/ID), ואילו הקו IF.Flush יכול למנוע כתיבה לאוגר IF/ID בסוף פעימת שעון.

ד. אין הבדל ביניהם. זה פשוט כינוי שונה לאותה בקרה בתרשימים שונים בצנרת, המבצעים פעולה של מחיקת אוגר הצנרת IF/ID.

ה. הקו IF.Flush מנקה את תוכן האוגר IF/ID, מוחק את הפקודה הנמצאת שם והופך אותה לקידוד 0, ואילו הקו IF/ID.Write יכול למנוע כתיבה לאוגר IF/ID בסוף פעימת שעון.

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
0x00400200	0x010a3024	and \$6,\$8,\$10	28 and \$6, \$8, \$10
0x00400204	0x8cca0200	lw \$10,0x0000020c(\$6)	29 lw \$10, 0x20c(\$6)
0x00400208	0x01446025	or \$12,\$10,\$4	30 or \$12, \$10, \$4
0x0040020c	0x118efffc	beq \$12,\$14,0xffffffff	31 beq \$12, \$14, someWhere

ערך כל אוגר בתחילת ביצוע הקוד הינו מספר האוגר כפול $0x80000$ (2^{19})

- אוגר 1: ערכו $0x00080000$
- אוגר 2: ערכו $0x00100000$
-
- אוגר 10: ערכו $0x00500000$ (אוגר $10 \leftarrow 0xa \leftarrow$ כפול $0x80000$)
- ...
- אוגר 16: ערכו $0x00800000$
- ...

כמו כן, נתון שניתן לגשת לכל מרחב הזיכרון בפקודות SW ו-LW (כמובן, כל עוד הכתובות מתחלקות ב-4)

נתונים אלה תקפים לכל סעיפי שאלה 7

7.1 (2 נקודות):

מה ערכו של אוגר \$6 לאחר פקודה בשורה 28?

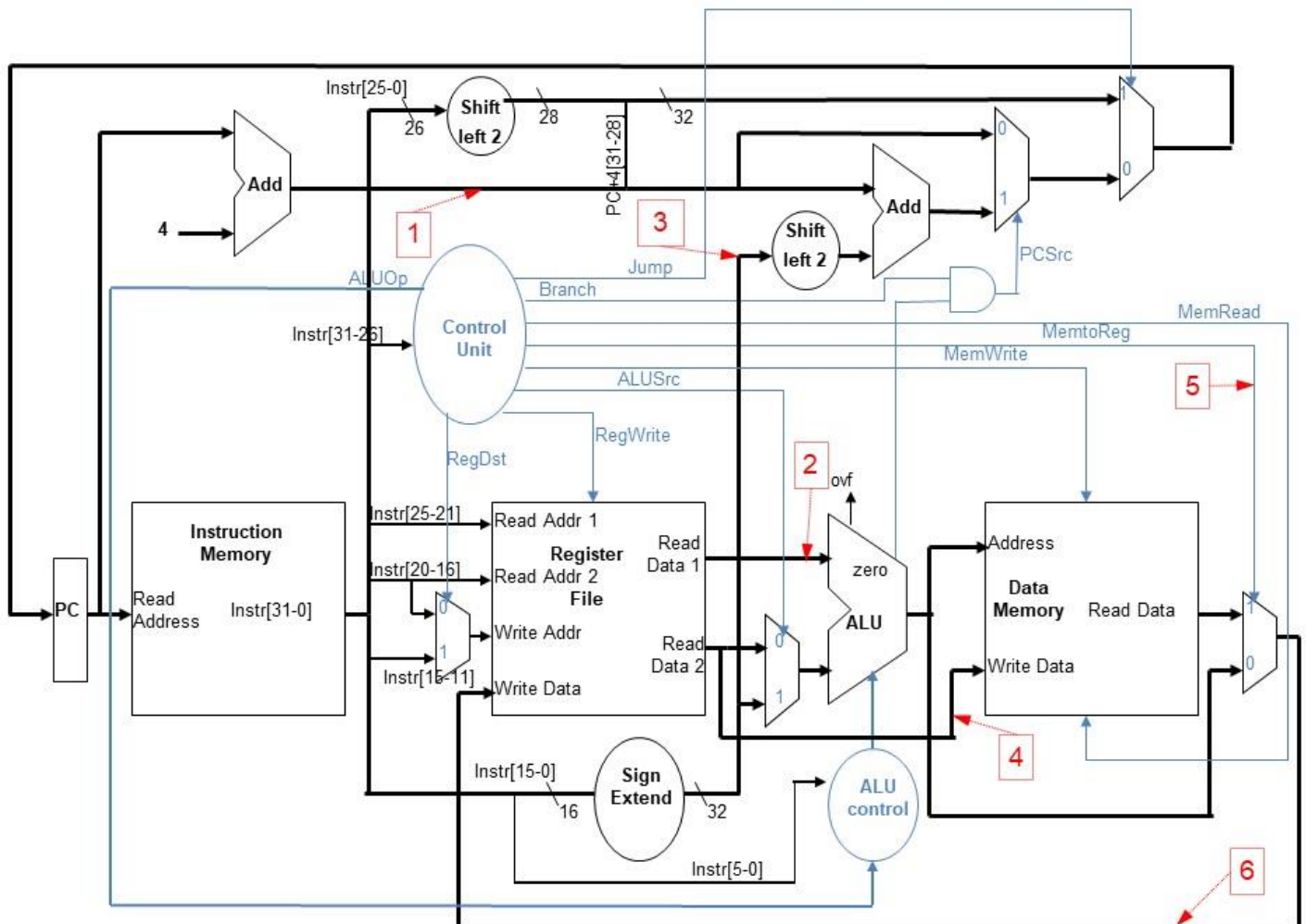
0x00400000

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 השנייה בקוד (שורה 29):

lw \$10, 0x20c(\$6)



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	0x00400208
2	0x00400000
3	0x0000020c

נקודה	ערך (בהקסה)
4	0x00800000
5	1
6	0x118efffc (ערך בזיכרון בכתובת 0x0040020c, ה-Code של beq)

7.3 (4 נקודות)

מה כתובת התווית somewhere בפקודת ה-BEQ (הפקודה הרביעית בשורה 31 בקוד?

$$PC + 4 + imm * 4 = 0x0040020c + 4 + (-4) * 4 = 0x00400200$$

האם תתבצע קפיצה לתווית somewhere כאשר הקוד יגיע לפקודה זו? יש לנמק עם התייחסות לערכי האוגרים בפקודה זו

לא תתבצע קפיצה.
ערכו של אוגר \$12 לאחר פקודת ה or בשורה 30: 0x11aeffff
ערכו של אוגר \$14, לפי נתוני השאלה, הינו 0x00a00000
לא מתקיים שוויון, ולכן, אין קפיצה

7.4 (4 נקודות)

כעת, בזמן ביצוע פקודת LW בלבד, עקב תקלה בקו המחבר את המרביב RegDst עם כניסת Write Addr של מקבץ האוגרים, הערך העובר על קו הקו ונכנס למקבץ האוגרים משובש, וכל הביטים שעל הקו זזים מקום אחד ימינה (shift right 1).

עקב תקלה זו: מה ההשפעה הישירה של התקלה?

האם יש השפעה על תשובותיך בסעיפים 7.2 ו-7.3?

מה ההשפעות? יש לנמק עם התייחסות לערכים המשתנים, היכן משתנים, וההשפעות המלאות שלהן

ההשפעה הישירה של תקלה זו הינה, שבפקודת LW אוגר המטרה יהיה בפועל אוגר \$5 מאחר ומספר האוגר משובש, ובמקום הערך 10 כמספר אוגר המטרה, ייכנס הערך 5 כמספר אוגר המטרה.
בסעיף 7.2 אין שינוי בערכי הנקודות של גבי התרשים.
מאידך, יש השפעה בהמשך. בעת ביצוע פקודת ה OR, ערכו של אוגר \$10 לא השתנה בפועל בפקודת ה LW (מאחר והתוצאה נרשמה באוגר \$5). ערכו של אוגר \$10 נשאר כפי שהיה לפי נתוני השאלה – 0x00800000. פקודת ה OR תתבצע עם אוגר \$4 שערכו 0x00200000.
תוצאת ה OR בין שני ערכים אלה תהיה 0x00a00000, וזו התוצאה שתישמר באוגר \$12.
כעת, בפקודת BEQ – בעת ההשוואה עם אוגר \$14 שערכו 0x00a00000 – יתקיים השוויון, ולכן תתבצע קפיצה לתווית someWhere

7.5 (23 נקודות)

התרשים שלפניכם בעמוד הבא מבוסס על תרשים 4.60 בספר. התרשים, שאינו מלא כעת, 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
0x00400200	0x010a3024	and \$6,\$8,\$10	28 and \$6, \$8, \$10
0x00400204	0x8cca0200	lw \$10,0x0000020c(\$6)	29 lw \$10, 0x20c(\$6)
0x00400208	0x01446025	or \$12,\$10,\$4	30 or \$12, \$10, \$4
0x0040020c	0x118efffc	beq \$12,\$14,0xffffffffc	31 beq \$12, \$14, someWhere

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.

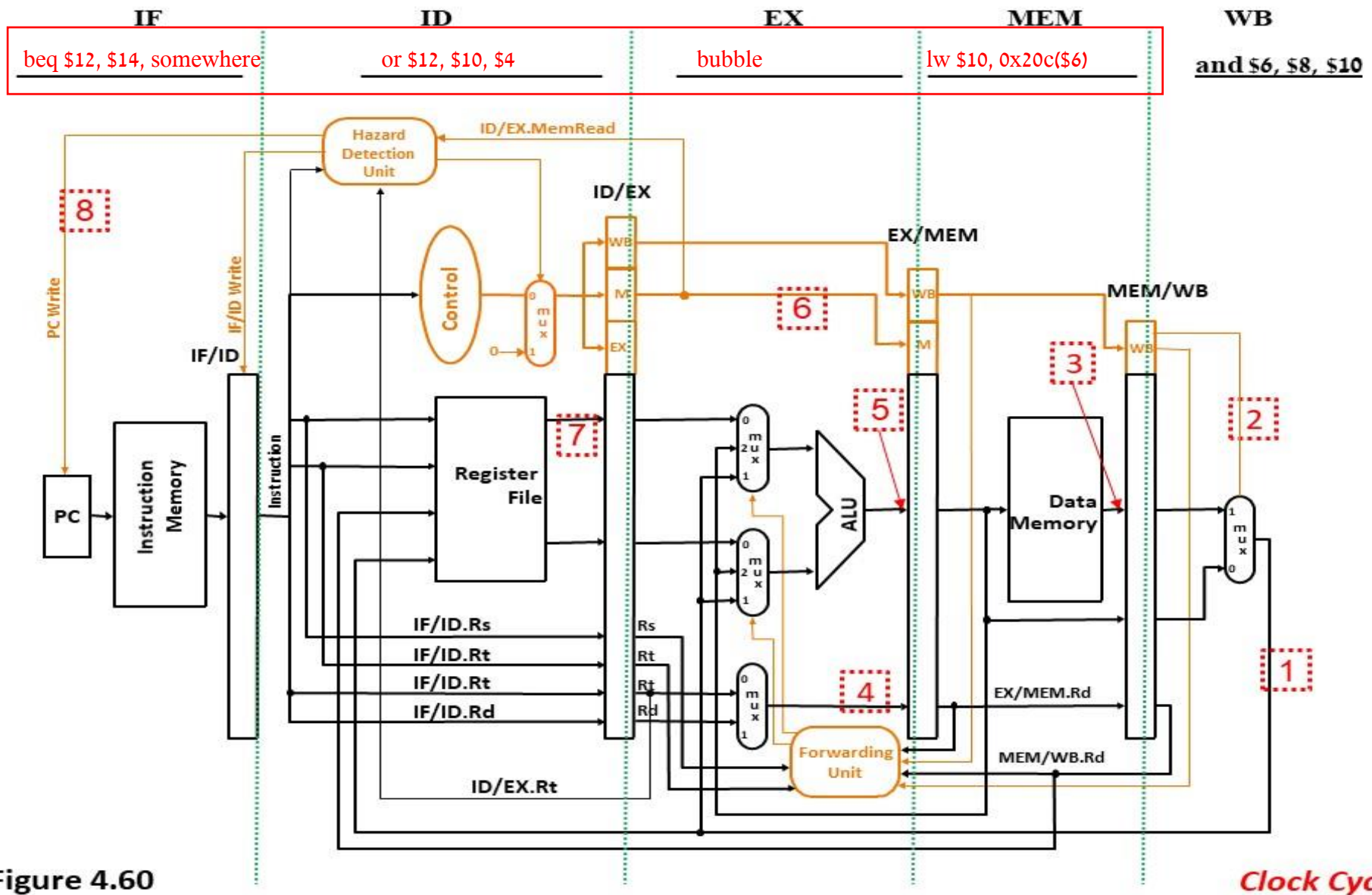
ניתן להניח סיכוני הנתונים בהרצת הקוד נפתרים על ידי יחידת העברה קדימה ויחידת איתור הסיכונים (Forwarding Unit, HDU) וחצית מקבץ האוגרים.

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	0x00400000 (תוצאה לאוגר \$6)	5	0x00700000 (תוצאת ALU של פקודת OR שעברה עם ה-bubble, בהתבסס על ערך מקורי של \$10 ואוגר \$4, לערך זה יש משמעות בפועל מאחר וכל הדגלים של הפקודה מאופסים, כי זה למעשה bubble)
2	0 (דגל MemToReg)	6	0x0 (ב bubble כל הדגלים מאופסים)
3	0x118efffc (ערך בזיכרון בכתובת 0x0040020c, ה-Code של beq)	7	0x00500000 (ערכו "המקורי" של אוגר \$10 לפני העדכון של פקודת LW)
4	0xc (מספר אוגר המטרה של פקודת OR "שעברה" עם ה-bubble)	8	1 (מאפשר כתיבה לאוגר PC, זה מחזור שעון לאחר הטיפול ב-LW ל bubble)



7.5.3 (4 נקודות)

בתרשים בעמוד הבא, מסומן עיגול שחור ● בשלב EX, על גבי הקו שבין מרבב ה RegDst לאוגר הצנרת Ex/Mem. עיגול זה מתאר אותה תקלת חומרה המתוארת בסעיף 7.4 (shift right 1). הפעם, זו תקלה קבועה המתרחשת בכל מחזורי השעון של המעבד. לאור זאת, האם וכיצד תשתנה תשובתך לסעיף 7.5.1?

IF	ID	EX	MEM	WB
<u>beq \$12, \$14, somewhere</u>	<u>or \$12, \$10, \$4</u>	<u>bubble</u>	<u>lw \$10, 0x20c(\$6)</u>	<u>and \$6, \$8, \$10</u>

יש לנמק ולהסביר מדוע לדעתך יש/אין שינוי? לא נדרש הסבר כמותי, אלא הסבר עקרוני וקצר.

<p>7.5.1 אין שינוי בתשובה בהתייחסות לסעיף 7.5.1</p> <p>הסיבה לכך הינה שבעת מעבר פקודת LW בשלב ה EX, מספר אוגר המטרה של פקודת ה LW, אכן משתבש והופך ל- 5. אבל, הקו ID/Ex.Rt היוצא משלב ה EX אל ה HDU, מכיל את הערך <u>לפני השיבוש</u>. כלומר, כלומר מספר אוגר \$10. לכן, <u>למרות</u> התקלה בהמשך, אל ה HDU מגיע מספר אוגר 10, והוא מבצע השוואה עם האוגר בפקודת OR וזיהוי הצורך ב bubble נשאר בתוקף</p>

שאלה 8 (25 נקודות)

כתבו פונקציה בשם isDiagonalKing המקבלת 3 פרמטרים:

- \$a0 – כתובת תחילה של נתוני מטריצה מלבנית, ערכי המטריצה בגודל Byte אחד
 - Byte ראשון – מספר שורות במטריצה
 - Byte שני – מספר עמודות במטריצה
 - Byte שלישי ואילך – ערכי המטריצה, שורה אחרי שורה
- \$a1 – מספר שלם R – המייצג שורה במטריצה
- \$a2 – מספר שלום C – המייצג עמודה במטריצה

הפונקציה תבדוק שהערך במקום [R][C] הינו הערך הגדול ביותר באלכסון העובר שך מקום זה. כלומר, האלכסון ממקום זה כלפי – שמאלה/למעלה, והאלכסון ממקום זה כלפי ימינה/למטה.

הפונקציה **תחזיר** את הערך 1 (מייצג true) במידה והערך במקום [R][C] הינו הגדול. 0 (מייצג false) בכל מקרה אחר. הבהרה: עם קיים ערך זהה באלכסון – זה מוגדר כתקין!

	0	1	2	3	4	5	6
0	96	81	56	82	61	13	24
1	40	6	54	85	90	41	18
2	52	29	17	77	67	69	91
3	10	7	1	39	95	79	30
4	49	28	12	43	86	35	36

דוגמה – עבור המטריצה משמאל:

- עבור מיקום [3][4]: יוחזר הערך 1 (95 הוא הערך הגדול ביותר באלכסון שמאלה/למעלה עד תא [0][1], וכן באלכסון ימינה/למטה עד תא [4][5]).
- עבור מיקום [1][5]: יוחזר הערך 0 (41 הוא לא הערך הגדול ביותר באלכסון אליו שייך – ממקום [0][4] עד [2][6]).
- עבור מיקום [2][0]: יוחזר הערך 1 (52 הוא הערך הגדול ביותר באלכסון אליו שייך – ממקום [2][0] עד [4][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 מלבני (ולא רק למערך של 5x7 שניתן בדוגמה).
- ניתן להניח שהערכים בפרמטרים (אוגרים \$a0.... \$a2) תקינים – כלומר, **אין צורך** לבצע בדיקות תקינות.

פתרון שאלה 8:

the solution

isDiagonalKing:

\$a0 - matrix address: first Byte is number of rows
second Byte is number of columns
from 3rd Byte - matrix values
\$a1 - row index
\$a2 - column index

```
lb $t0, 0($a0) # get number of rows
lb $t1, 1($a0) # get number of columns (and also the row size)

mul $t2, $a1, $t1
add $t2, $t2, $a2
addi $t2, $t2, 2
add $a0, $a0, $t2 # a0 - is the check position address
lb $t2, 0($a0) # the value to check if it is the diagonal king
addi $t3, $t1, 1 # $t3 - number of bytes to skip thru the diagonal
```

```
li $v0, 1 # set return value to true
move $t8, $a1 # start row
move $t9, $a2 # start col
move $t7, $a0 # start address
```

upLeftLoop:

```
addi $t8, $t8, -1 # next diagonal row
addi $t9, $t9, -1 # next diagonal column
blt $t8, 0, downRightLoop # check for matrix border
blt $t9, 0, downRightLoop # check for matrix border
sub $t7, $t7, $t3 # next diagonal value address
lb $t6, 0($t7)
bgt $t6, $t2, notKing # current value is bigger? if yes, not a king
j upLeftLoop
```

downRightLoop:

```
addi $a1, $a1, 1 # next diagonal row
addi $a2, $a2, 1 # next diagonal column
bge $a1, $t0, isKing # check row matrix border, if reach - it is the king
bge $a2, $t1, isKing # check column matrix border, if reach - it is the king
add $a0, $a0, $t3 # next diagonal value address
lb $t6, 0($a0)
bgt $t6, $t2, notKing # current value is bigger? if yes, not a king
j downRightLoop
```

notKing:

```
li $v0, 0
```

isKing:

```
jr $ra
```

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

חלק א': (24 נקודות)

בחלק זה 6 שאלות רב ברירה. יש לבחור את התשובה הנכונה מבין התשובות האפשריות (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

נתון מעבד אשר זמן מחזור שעון שלו הינו $2 \cdot 10^{-10}$.

CPI	% פקודות	קבוצה
6	20%	A
4	30%	B
2	40%	C
8	10%	D

במעבד יש 4 קבוצות של פקודות: A, B, C ו-D. בטבלה משמאל נתון, לכל אחת מקבוצת הפקודות: מספר מחזורי שעון הנדרש לפקודות בקבוצה זו, וכן את החלק היחסי של הפקודות בתוכנית X. נתון שיש בתוכנית X לולאה עם 300 פקודות (הלולאה מכילה פקודות מקבוצות הפקודות משמאל, בהתאם לאחוז המפורט).

נתון כי משך הזמן לביצוע הלולאה הינו 24 שניות.

כמה פעמים עובדת הלולאה?

- לא ניתן לדעת על בסיס הנתונים בשאלה
- הלולאה עובדת 100,000,000 פעמים (10^8)
- הלולאה עובדת 80,000,000 פעמים ($8 \cdot 10^7$)
- הלולאה עובדת 120,000,000 פעמים ($1.2 \cdot 10^8$)
- הלולאה עובדת 10,000,000 פעמים (10^7)

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

Address	Code	Basic	Source
0x9040001c	0x3c086100	lui \$8,0x00006100	20: lui \$t0, 0x6100
0x90400020	0x00084602	srl \$8,\$8,0x00000018	21: srl \$t0, \$t0, 24
0x90400024	0x310800df	andi \$8,\$8,0x000000df	22: andi \$t0, \$t0, 0xdf
0x90400028	0x20090041	addi \$9,\$0,0x00000041	23: addi \$t1, \$zero, 'A'
0x9040002c	0x11090001	beq \$8,\$9,0x00000001	24: beq \$t0, \$t1, noWhere
0x90400030	0x08100007	j guessMe	25: j guessMe

שאלה 2:

מה ניתן להגיד על הפקודה בשורה 24 בקוד המקור: **beq \$t0, \$t1, nowhere**:

- זו פקודת פסיאודו, ניתן לדעת שלא תתבצע קפיצה לתווית noWhere
- זו פקודת אסמבלר בסיסית, ניתן לדעת שלא תתבצע קפיצה לתווית noWhere
- זו פקודת פסיאודו, ניתן לדעת שתתבצע קפיצה לתווית noWhere
- זו פקודת אסמבלר בסיסית, ניתן לדעת שתתבצע קפיצה לתווית noWhere
- לא ניתן לבצע השוואה בין אוגר \$t0 לאוגר \$t1 מאחר ולאוגר \$t1 יש ערך ASCII בעוד שב-\$t0 יש ערך בהקסה, ולכן לא תתבצע קפיצה

שאלה 3

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

- 0x90400030
- 0x0040001c
- 0x90100007
- 0x9040001c
- לא ניתן לדעת מהנתונים הקיימים

שאלה 4

נתון זיכרון מטמון עם המאפיינים הבאים:

- מאורגן בשיטת 2-Way Set Associative
- נתון כי יש 8,192 (2^{13}) בלוקים של נתונים במטמון
- גודל שדה ה-Tag הינו 13 סיביות

מה גודל הנתונים (data) במטמון, בבתים (Bytes)?

- 2^{19} (524,288) בתים
- 2^{20} (1,048,576) בתים
- 2^{21} (2,097,152) בתים
- 2^{22} (4,194,304) בתים
- לא ניתן לדעת על בסיס נתוני השאלה

שאלה 5

נתון ערך הקסה $0x41350000$ המייצג מספר רציונאלי float בהתאם ל-IEEE 754.
כאשר נמיר ערך זה לעשרוני, איזה ערך הוא מייצג?

- א. 3.3125
- ב. -3.3125
- ג. 11.3125
- ד. -11.3125
- ה. 8.3125

שאלה 6

ב MIPS העובד בטכנולוגיית צנרת קו הבקרה IF.Flush (ניתן לראות בתרשימים 4.65 ו-4.66 בספר הקורס) וקו הבקרה IF/ID.Write (ניתן לראות בתרשים 4.60 בספר) פועלים שניהם על אוגר הצנרת IF/ID. מה ההבדל בין קווים אלו?

- א. הקו IF.Flush מטפל בסיכוני נתונים וקו IF/ID.Write בסיכוני בקרה.
- ב. הקו IF.Flush מנקה את תוכן האוגר IF/ID, מוחק את הפקודה הנמצאת שם והופך אותה לקידוד 0, ואילו הקו IF/ID.Write מאפס רק את ערכי הבקרה של הפקודה, והופך אותה לבועה.
- ג. הקו IF/ID.Write מבצע חיסור של 4 מ-PC, ובכך משנה את חישוב כתובת branch target (ולכן נכנס לחלק העליון של IF/ID), ואילו הקו IF.Flush יכול למנוע כתיבה לאוגר IF/ID בסוף פעימת שעון.
- ד. אין הבדל ביניהם. זה פשוט כינוי שונה לאותה בקרה בתרשימים שונים בצנרת, המבצעים פעולה של מחיקת אוגר הצנרת IF/ID.
- ה. הקו IF.Flush מנקה את תוכן האוגר IF/ID, מוחק את הפקודה הנמצאת שם והופך אותה לקידוד 0, ואילו הקו IF/ID.Write יכול למנוע כתיבה לאוגר IF/ID בסוף פעימת שעון.

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
0x00400200	0x010a3024	and \$6,\$8,\$10	28 and \$6, \$8, \$10
0x00400204	0x8cca0200	lw \$10,0x0000020c(\$6)	29 lw \$10, 0x20c(\$6)
0x00400208	0x01446025	or \$12,\$10,\$4	30 or \$12, \$10, \$4
0x0040020c	0x118efffc	beq \$12,\$14,0xffffffff	31 beq \$12, \$14, someWhere

ערך כל אוגר בתחילת ביצוע הקוד הינו מספר האוגר כפול $0x80000$ (2^{19})

- אוגר 1: ערכו $0x00080000$
- אוגר 2: ערכו $0x00100000$
-
- אוגר 10: ערכו $0x00500000$ (אוגר $10 \leftarrow 0xa \leftarrow$ כפול $0x80000$)
- ...
- אוגר 16: ערכו $0x00800000$
- ...

כמו כן, נתון שניתן לגשת לכל מרחב הזיכרון בפקודות SW ו-LW (כמובן, כל עוד הכתובות מתחלקות ב-4)

נתונים אלה תקפים לכל סעיפי שאלה 7

7.1 (2 נקודות):

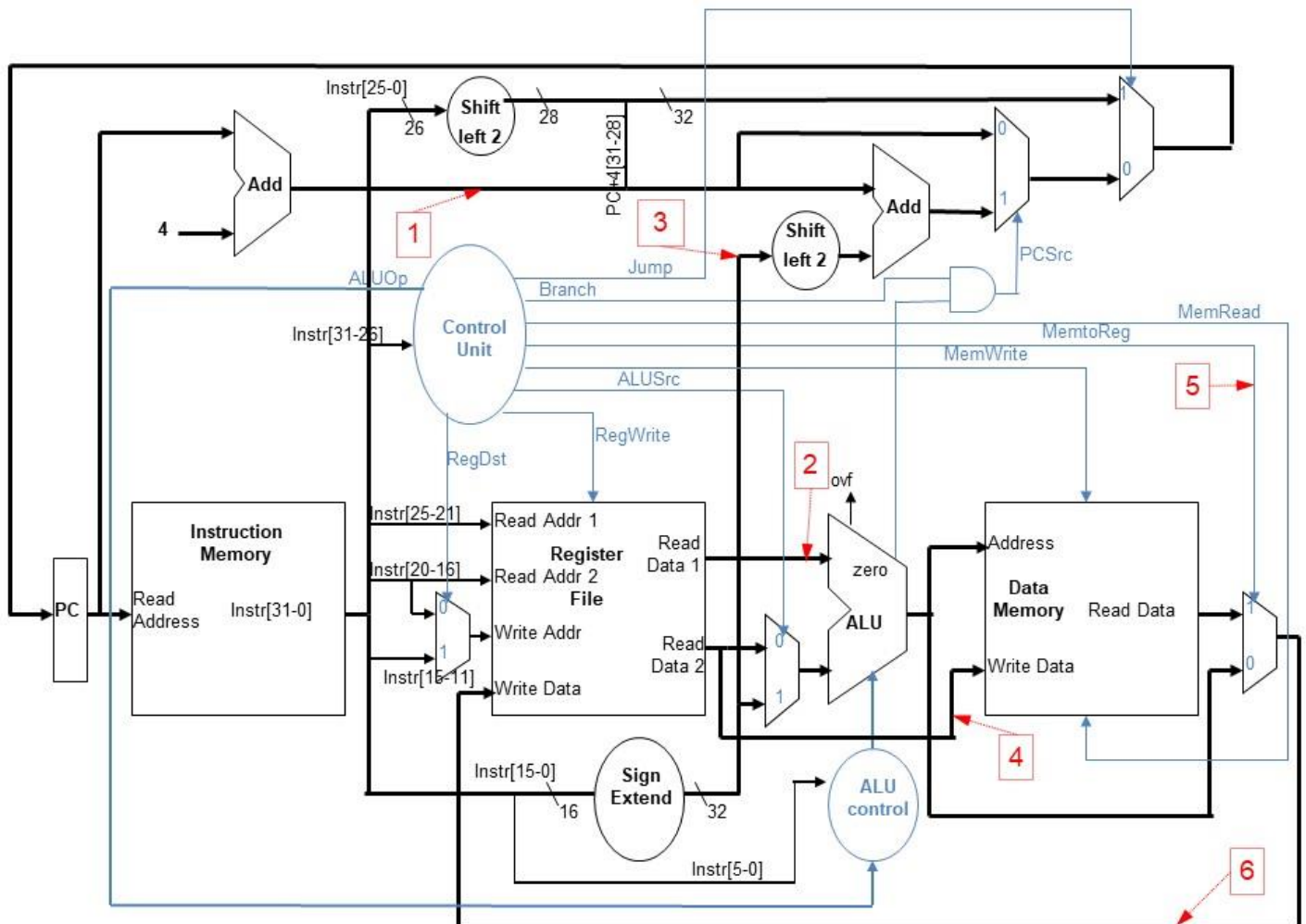
מה ערכו של אוגר \$6 לאחר פקודה בשורה 28?

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 השנייה בקוד (שורה 29):

`lw $10, 0x20c($6)`



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	

מה כתובת התווית **somewhere** בפקודת ה-BEQ (הפקודה הרביעית בשורה 31 בקוד)?

[illegible]

7.5 (23 נקודות)

התרשים שלפניכם בעמוד הבא מבוסס על תרשים 4.60 בספר. התרשים, שאינו מלא כעת, 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:

<u>Address</u>	<u>Code</u>	<u>Basic</u>	<u>Source</u>
0x00400200	0x010a3024	and \$6,\$8,\$10	28 and \$6, \$8, \$10
0x00400204	0x8cca0200	lw \$10,0x0000020c(\$6)	29 lw \$10, 0x20c(\$6)
0x00400208	0x01446025	or \$12,\$10,\$4	30 or \$12, \$10, \$4
0x0040020c	0x118efffc	beq \$12,\$14,0xffffffffc	31 beq \$12, \$14, someWhere

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.

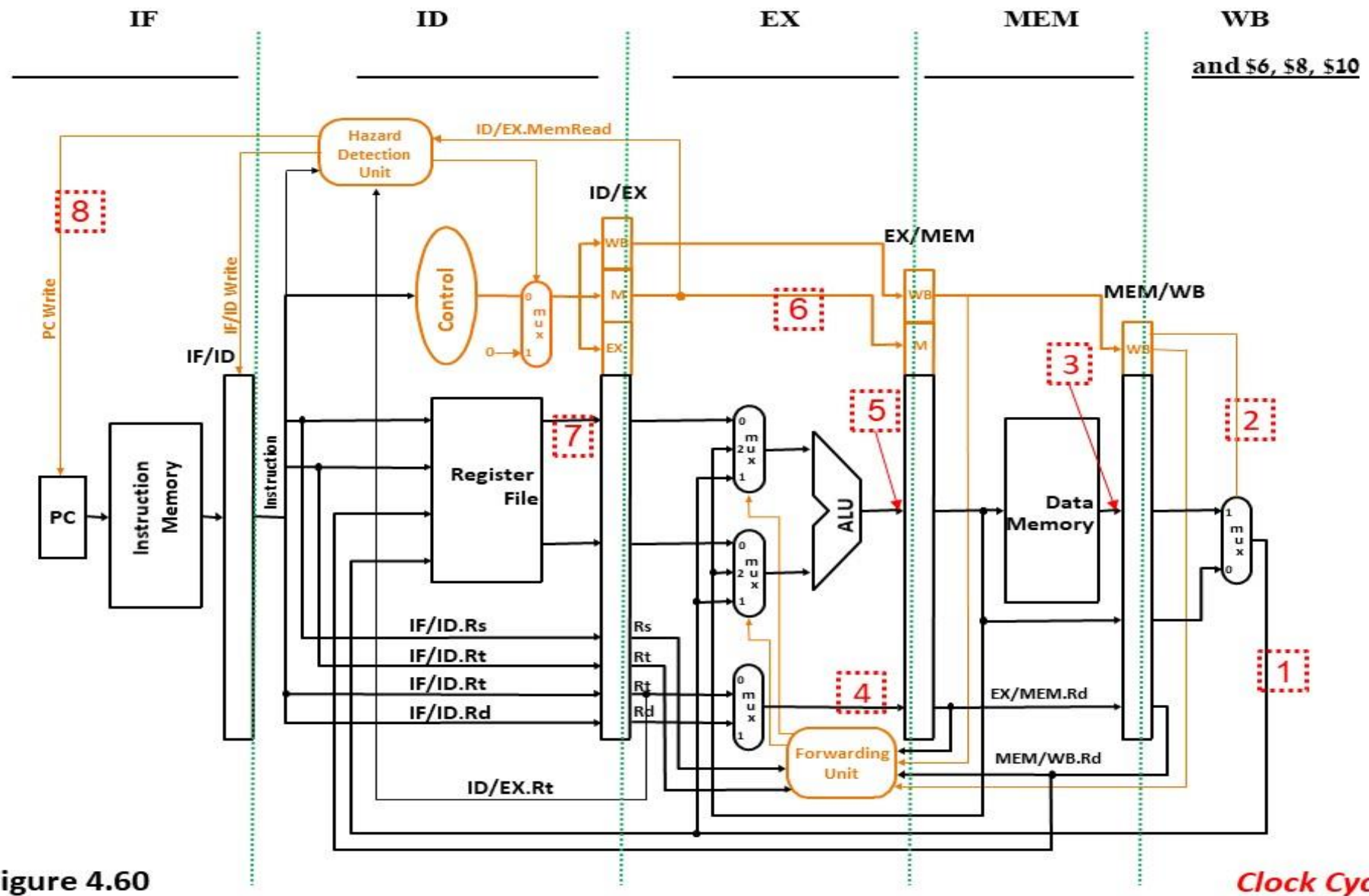
ניתן להניח סיכוני הנתונים בהרצת הקוד נפתרים על ידי יחידת העברה קדימה ויחידת איתור הסיכונים (Forwarding Unit, HDU) וחצית מקבץ האוגרים.

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	



בטרשים בעמוד הבא, מסומן עיגול שחור ● בשלב EX, על גבי הקו שבין מרבב ה RegDst לאוגר הצנרת Ex/Mem. עיגול זה מתאר אותה תקלת חומרה המתוארת בסעיף 7.4 (shift right 1). הפעם, זו תקלה קבועה המתרחשת בכל מחזורי השעון של המעבד. לאור זאת, **האם וכיצד** תשתנה תשובתך לסעיף 7.5.1?

יש לנמק ולהסביר מדוע לדעתך יש/אין שינוי? לא נדרש הסבר כמותי, אלא הסבר עקרוני וקצר.

[illegible]

שאלה 8 (25 נקודות)

כתבו פונקציה בשם isDiagonalKing המקבלת 3 פרמטרים:

- \$a0 – כתובת תחילה של נתוני מטריצה מלבנית, ערכי המטריצה בגודל Byte אחד
 - Byte ראשון – מספר שורות במטריצה
 - Byte שני – מספר עמודות במטריצה
 - Byte שלישי ואילך – ערכי המטריצה, שורה אחרי שורה
- \$a1 – מספר שלם R – המייצג שורה במטריצה
- \$a2 – מספר שלום C – המייצג עמודה במטריצה

הפונקציה תבדוק שהערך במקום [R][C] הינו הערך הגדול ביותר באלכסון העובר שך מקום זה. כלומר, האלכסון ממקום זה כלפי – שמאלה/למעלה, והאלכסון ממקום זה כלפי ימינה/למטה.

הפונקציה **תחזיר** את הערך 1 (מייצג true) במידה והערך במקום [R][C] הינו הגדול. 0 (מייצג false) בכל מקרה אחר. הבהרה: עם קיים ערך זהה באלכסון – זה מוגדר כתקין!

	0	1	2	3	4	5	6
0	96	81	56	82	61	13	24
1	40	6	54	85	90	41	18
2	52	29	17	77	67	69	91
3	10	7	1	39	95	79	30
4	49	28	12	43	86	35	36

דוגמה – עבור המטריצה משמאל:

- עבור מיקום [3][4]: יוחזר הערך 1 (95 הוא הערך הגדול ביותר באלכסון שמאלה/למעלה עד תא [0][1], וכן באלכסון ימינה/למטה עד תא [4][5]).
- עבור מיקום [1][5]: יוחזר הערך 0 (41 הוא לא הערך הגדול ביותר באלכסון אליו שייך – ממקום [0][4] עד [2][6]).
- עבור מיקום [2][0]: יוחזר הערך 1 (52 הוא הערך הגדול ביותר באלכסון אליו שייך – ממקום [2][0] עד [4][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 מלבני (ולא רק למערך של 5x7 שניתן בדוגמה).
- ניתן להניח שהערכים בפרמטרים (אוגרים \$a0.... \$a2) תקינים – כלומר, **אין צורך** לבצע בדיקות תקינות.

פתרון שאלה 8:

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

[illegible]

[illegible]