

מחברת מספר: 27 1

מס' ת.ז.: 066251227

אוניברסיטת תל-אביב

15 | 1

מדעי המחשב

20 | 2

סמסטר א, תשס"ו

15 | 3

25 | 4

מבחן במבנה המחשב

25 | 5

מועד א'

המבחן כולל 11 עמודים (כולל עמוד זה) עם 5 שאלות.

הנחיות מיוחדות:

עליך לענות על \* כל \* השאלות !

יש לענות על השאלות על גבי טופס המבחן בלבד.

המחברות הן לטיוטה בלבד ולא תיבדקנה.

משך המבחן: 3 שעות

מלא/י את תעודת הזהות בכל העמודים!

פרופ' יהודה אפק, פרופ' נתן אינטרטור, אורי שלו

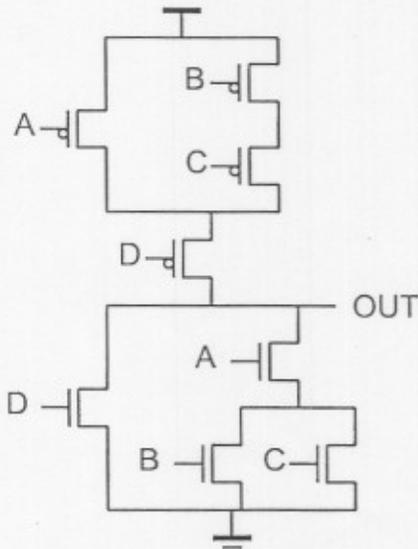
מועד המבחן: 24.2.2006

חומר עזר מותר בשימוש: כל חומר עזר כתוב

**בהצלחה !**

שאלה 1 (15%):

נתון המעגל הבא:



A B \ C D	00	01	11	10
00	1	1	0	1
01	0	0	0	0
11	0	0	0	0
10	1	1	0	0

א. (10%) מלא את טבלת האמות של המעגל. לפחות לשתי כניסות בטבלה הסבר ונמק מדוע מלאת את הערכים.  
 ב. (5%) צייר מעגל לוגי משערי AND, OR ו- NOT שמממש את אותה פונקציה.

A	D	C	D
0	0	0	1

f. ציור בטבלה

בדיקה של ה"ד" המעגל הכיוון ה-D הוא קלט ב-A עבור האופן של D המעגל  
 ומצד שני המעגל של D האובי קלט כדור סימול > OUT פוט כדור  
 האופן האובי סימול (אובי) של D (בטור) של בטור אובי בטור (D=1)

A	B	C	D
1	0	0	0

דכי בטבלה

הכיוון ה-D, C, B, A צריכים להיות 1 - 1 - 1 - 1.  $\sqrt{out}$  הכיוון ה-D, C, B, A.  
 הכיוון ה-D, C, B, A צריכים להיות 0 - 0 - 0 - 0. ~~כדור סימול~~  
 הכיוון ה-D, C, B, A צריכים להיות 0 - 0 - 0 - 1.  $\sqrt{out}$  הכיוון ה-D, C, B, A

$$\begin{aligned} \bar{A}\bar{D} + A\bar{B}\bar{C}\bar{D} &= \bar{A}\bar{D} \\ &= \bar{D}(\bar{A} + A\bar{B}\bar{C}) = \bar{D}(\bar{A} + B\bar{C}) = \bar{D}A + \bar{D}B\bar{C} \\ &= \bar{D}(A + B\bar{C}) \end{aligned}$$

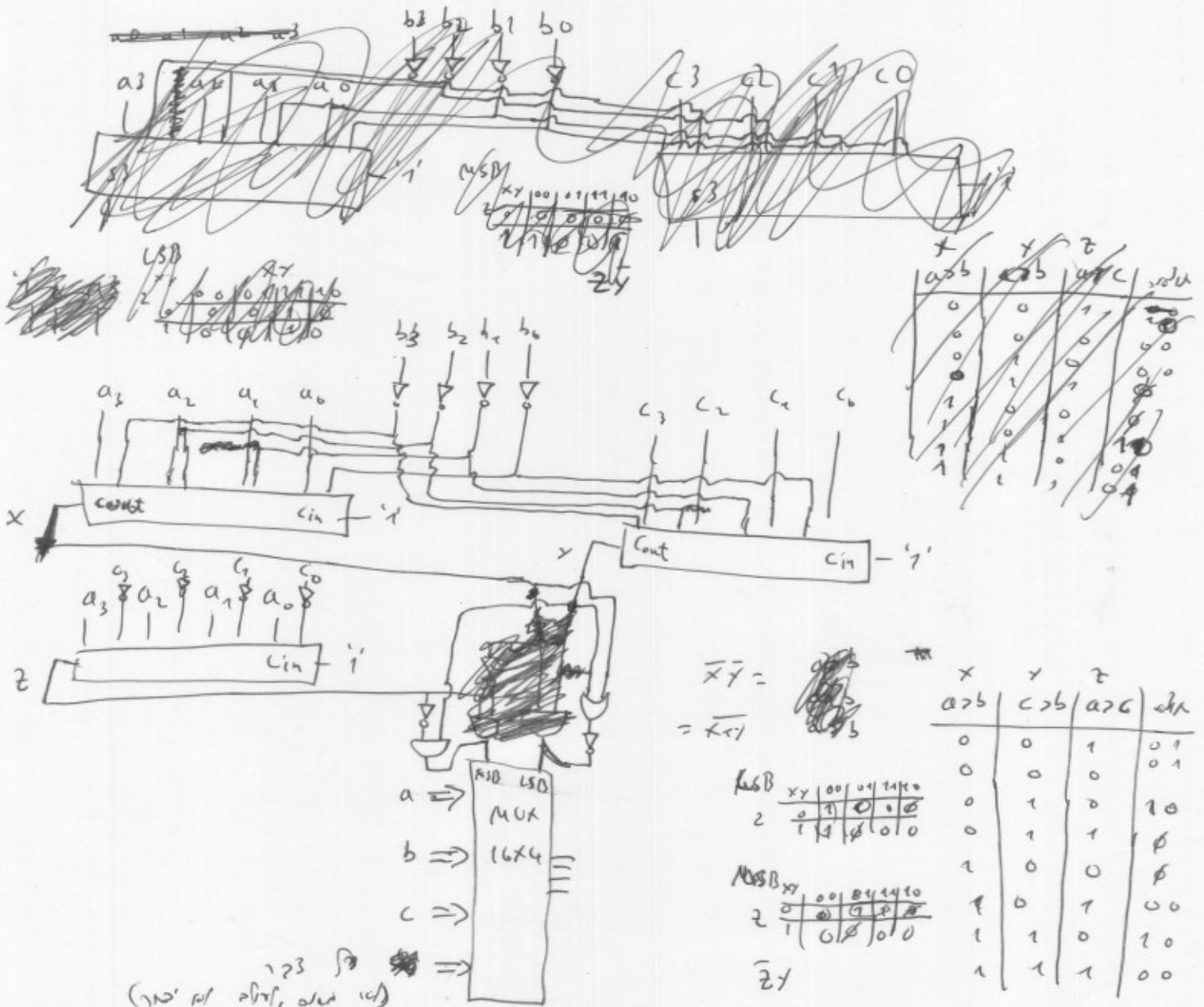


$$\bar{x}\bar{y} = \bar{y}\bar{x} = \bar{y}(\bar{x} + \bar{y}) = \bar{y}(\bar{A} + B + C) =$$

שאלה 2 (20%)

א. תכנן מעגל המקבל שלושה מספרים בני 4 ביט (unsigned) ומוציא כפלט את המספר הגדול מביניהם. למעגל 12 קווי קלט ו-4 קווי פלט. לרשותך:

יחידות Four-bit-adder בעלי יציאות carry ו-overflow, שערים לוגיים, ויחידה אחת מסוג MUX16X4. על המעגל להיות פשוט ככל שניתן.



בסעיף א' יש לבנות מעגל המקבל שלושה מספרים בני 4 ביט (unsigned) ומוציא כפלט את המספר הגדול מביניהם. למעגל 12 קווי קלט ו-4 קווי פלט. לרשותך: יחידות Four-bit-adder בעלי יציאות carry ו-overflow, שערים לוגיים, ויחידה אחת מסוג MUX16X4. על המעגל להיות פשוט ככל שניתן.

המעגל יבנה באמצעות יחידות Four-bit-adder בעלי יציאות carry ו-overflow, שערים לוגיים, ויחידה אחת מסוג MUX16X4. על המעגל להיות פשוט ככל שניתן.

המעגל יבנה באמצעות יחידות Four-bit-adder בעלי יציאות carry ו-overflow, שערים לוגיים, ויחידה אחת מסוג MUX16X4. על המעגל להיות פשוט ככל שניתן.

ב. ממש את המעגל הנ"ל כרכיב בשפת VHDL. הנח כי הארכיטקטורות של רכיבי ה-Four-bit-adder וה-MUX16X4 ממומשות, להלן הגדרות ה-entities שלהם:

```
ENTITY FourBitAdder IS
  PORT (a, b: IN std_logic_vector(3 downto 0);
        Cin : IN std_logic;
        sum: OUT std_logic_vector (3 downto 0);
        Cout, overflow: OUT std_logic);
  END FourBitAdder;

ENTITY MUX16X4 IS
  PORT (i0, i1, i2, i3: IN std_logic_vector(3 downto 0);
        select: IN std_logic_vector(1 downto 0);
        selected: OUT std_logic_vector (3 downto 0);
  END MUX16X4;
```

```
ENTITY Largest IS
  PORT (a, b, c: IN std_logic_vector(3 downto 0);
        out: OUT std_logic_vector(3 downto 0))
  END Largest;
```

ARCHITECTURE structure OF Largest IS

COMPONENT FourBitAdder

ENDCOMPONENT;

COMPONENT MUX16X4

ENDCOMPONENT;

SIGNAL notA, notB, ~~notC~~, temp: std\_logic\_vector (3 downto 0);

SIGNAL x, y, z: std\_logic;

SIGNAL t1, t2, t3: std\_logic;

SIGNAL t01, t02, t03: std\_logic\_vector;

SIGNAL sel: std\_logic\_vector (2 downto 0);

BEGIN

notB(0) <= not B(0);    notA(0) <= not A(0);    z(0) <= not (x(0) or y(0));  
 notB(1) <= not B(1);    notA(1) <= not A(1);    z(1) <= not (x(1) or y(1));  
 notB(2) <= not B(2);    notA(2) <= not A(2);    z(2) <= not (x(2) or y(2));  
 notB(3) <= not B(3);    notA(3) <= not A(3);    z(3) <= not (x(3) or y(3));

FBA1: FourBitAdder PORT (a, notB, '1', t01, x, t1);

FBA2: FourBitAdder PORT (c, notB, '1', t02, y, t2);

FBA3: FourBitAdder PORT (a, notC, '1', t03, z, t3);

MUX: MUX16X4 PORT (a, b, c, temp, sel, t0y);

sel(1) <= x and not z;

sel(0) <= not(x or y);

END

שאלה 3 (15%):

תכנן מעגל המקבל כקלט שני ערכים אנלוגיים בין 0 ל-1 ברזולוציה של 16 רמות, הפולט סיגנל אנלוגי שהוא מכפלת שני הקלטים, גם הוא באותה רזולוציה. למעגל גם כניסת שעון.

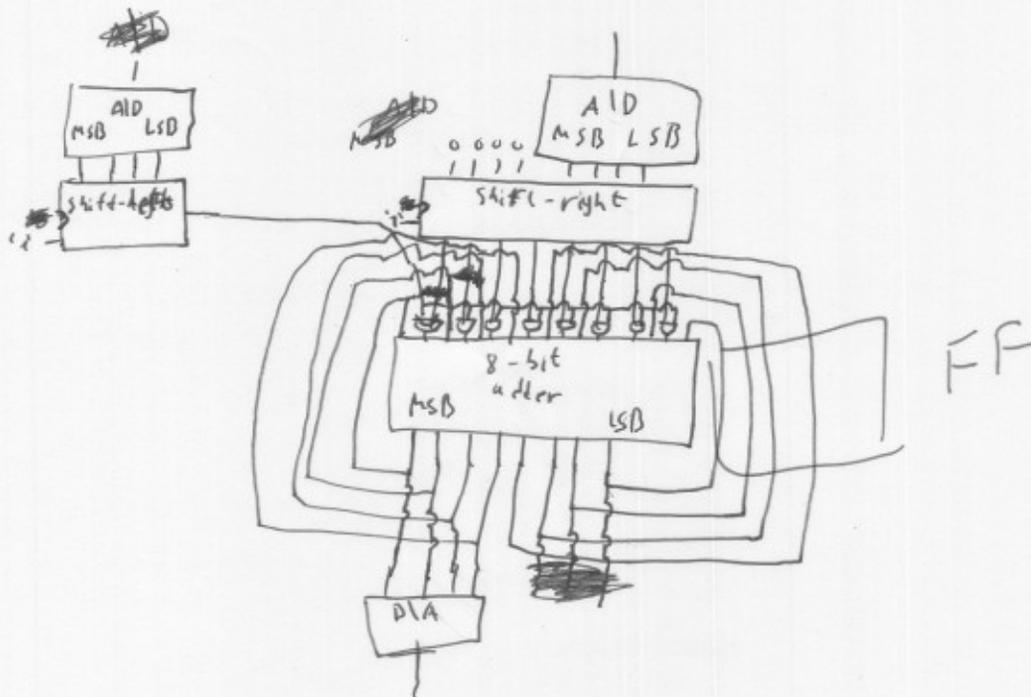
עליך לממש את המכפלה באופן סדרתי, כך שערך המכפלה יחושב תוך ארבעה מחזורי שעון. לרשותך:

אוגרי הזזה (shift-left וגם shift-right בכמות ובגדלים כרצונך), מחברי-8 ביט, יחידות FF, שערים לוגיים, ממיר (digital-to-analog) D/A, וממיר (analog-to-digital) A/D.

על המעגל להיות פשוט ככל הניתן.



מס' - קלט שני ערכים אנלוגיים בין 0 ל-1 ברזולוציה של 16 רמות, הפולט סיגנל אנלוגי שהוא מכפלת שני הקלטים, גם הוא באותה רזולוציה. למעגל גם כניסת שעון. עליך לממש את המכפלה באופן סדרתי, כך שערך המכפלה יחושב תוך ארבעה מחזורי שעון. לרשותך: אוגרי הזזה (shift-left וגם shift-right בכמות ובגדלים כרצונך), מחברי-8 ביט, יחידות FF, שערים לוגיים, ממיר (digital-to-analog) D/A, וממיר (analog-to-digital) A/D. על המעגל להיות פשוט ככל הניתן.



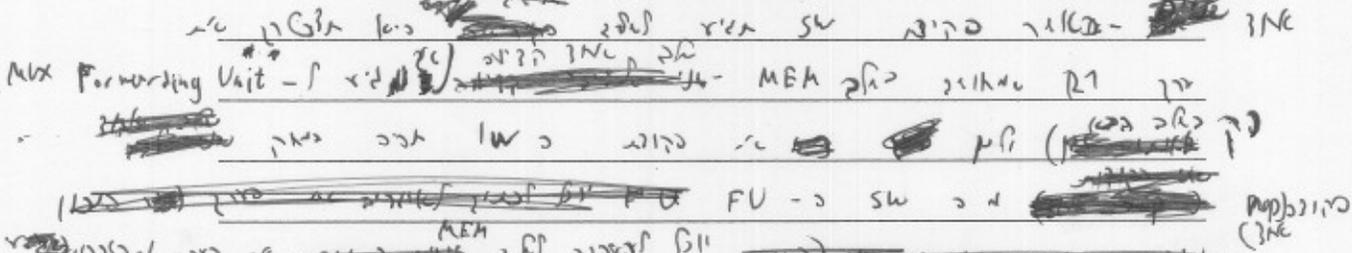
Pipeline : (25%) 4 שאלה

נתייחס למעבד MIPS כפי שנלמד בכיתה, עם Forwarding, Hazard Detection Unit ויחידת Register File שמאפשרת קריאת ערך חדש מרגיסטר באותו מחזור שעון בו הוא מתעדכן. ובהתאם לשרטוט המצורף.

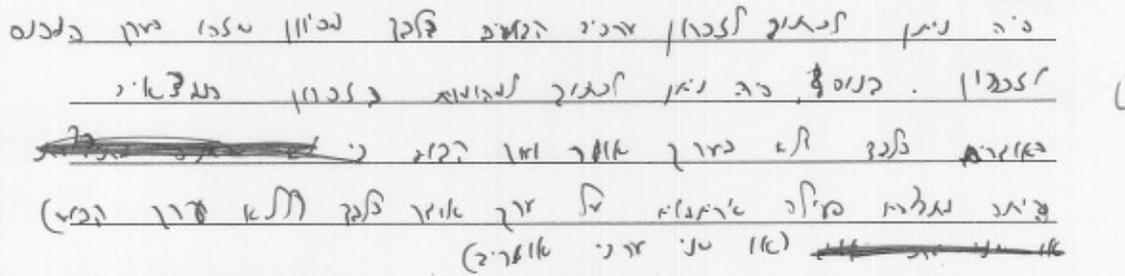
הנח שכל הקוד והנתונים מובאים מהמטמונים המתאימים (100% פגיעה). נתון קטע הקוד הבא (שתי פקודות):

```
LW R1, 100(R4)
SW R1, 200(R5)
```

א. (5%) מה יהיה מספר ה-stalls בין שתי פקודות אלה? הסבר.



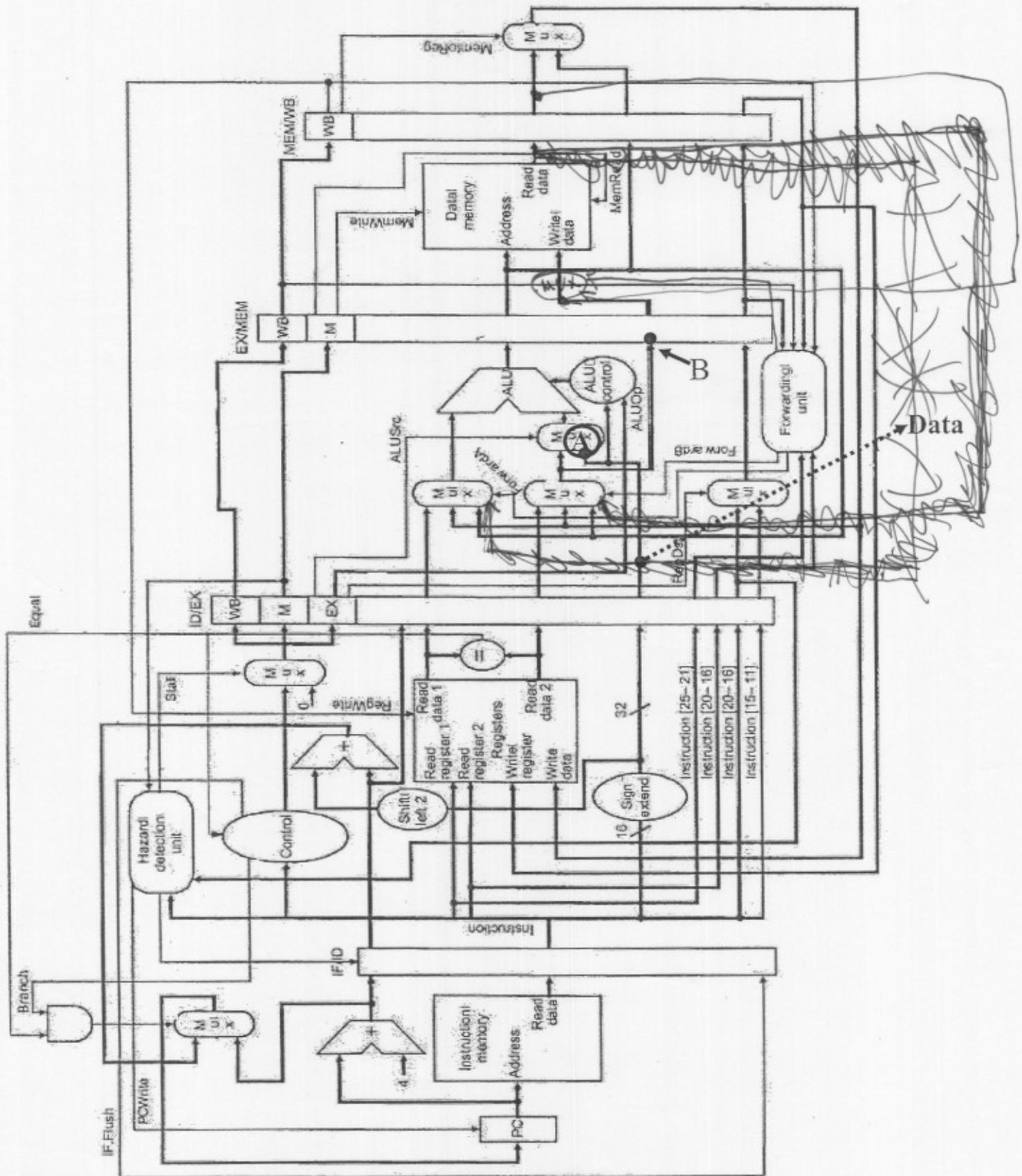
ב. (5%) נניח שהקו המסומן בשרטוט "Data" היה מחובר לנקודה B במקום לנקודה A. איזו בעיה היתה מתעוררת הסבר?



ג. (15%) הקו Data מחובר לנקודה A. רוצים לבצע שינוי בחומרה כך שלא יהיה צורך בכלל במחזורי stall במקרים כמו בסעיף א'.

תאר במילים ועל השרטוט המצורף איזו חומרה צריך להוסיף למעבד כדי לתמוך בשיפור זה. שים לב לפרטים, תאר בנפרד קווי מידע (data) ובקרה (control) חדשים שיש להוסיף או לשנות. את תפקיד הקווים החדשים/ששונו תאר בטבלאות שבעמודים הבאים. קווי בקרה חדשים יש לקודד עפ"י קווי בקרה מוכרים וקיימים ב-MIPS. נסה להסתפק בשינויים קלים ככל האפשר. אם נדרשים שינויים ביחידות קיימות ב-MIPS תאר את השינויים, הקפד לא לפגוע ביישום פקודות קיימות.

שרטוט של ה- DataPath של ה-MIPS כולל Hazard Detection, forwarding.



1.1 (3%) תאר בקצרה את הפתרון המוצע:

~~הפתרון המוצע הוא להשתמש ב- MEM ו- EX כדי לשפר את הביצועים של המערכת.~~

~~השימוש ב- MEM יאפשר לנו לשמור את הנתונים בצורה יעילה יותר, בעוד ש- EX יאפשר לנו להשתמש במעבד בצורה יעילה יותר.~~

~~הפתרון המוצע הוא להשתמש ב- MEM ו- EX כדי לשפר את הביצועים של המערכת.~~

~~השימוש ב- MEM יאפשר לנו לשמור את הנתונים בצורה יעילה יותר, בעוד ש- EX יאפשר לנו להשתמש במעבד בצורה יעילה יותר.~~

~~הפתרון המוצע הוא להשתמש ב- MEM ו- EX כדי לשפר את הביצועים של המערכת.~~

~~השימוש ב- MEM יאפשר לנו לשמור את הנתונים בצורה יעילה יותר, בעוד ש- EX יאפשר לנו להשתמש במעבד בצורה יעילה יותר.~~

2.2 (2%) הסבר מילולי קצר המתאר את החומרה המוצעת ותפקידה:

~~החומרה המוצעת היא מעבד MEM ו- EX.~~

~~המעבד MEM יאפשר לנו לשמור את הנתונים בצורה יעילה יותר, בעוד ש- EX יאפשר לנו להשתמש במעבד בצורה יעילה יותר.~~

~~המעבד MEM יאפשר לנו לשמור את הנתונים בצורה יעילה יותר, בעוד ש- EX יאפשר לנו להשתמש במעבד בצורה יעילה יותר.~~

~~המעבד MEM יאפשר לנו לשמור את הנתונים בצורה יעילה יותר, בעוד ש- EX יאפשר לנו להשתמש במעבד בצורה יעילה יותר.~~

3.3 (2%) קווי הנתונים הנוספים/ששוננו (גם צייר אותם על השרטוט):

שם הקו / הנתון עליו	רוחב	יוצא מ...	נכנס אל ...	הסבר והערות
<del>קווי הנתונים</del>	32	Hard Disk	MEM	
קווי נתונים	32	MEM	MEM	קווי נתונים
קווי נתונים	32	MEM	MEM	קווי נתונים
קווי נתונים	32	MEM	MEM	קווי נתונים

4. (2%) קווי הבקרה הנוספים/ששונו (גם לציין בשרטוט):

שם קו הבקרה	תוצאה כאשר האות = 0	תוצאה כאשר האות = 1	קידוד האות מאותות מוכרים
קריית צור לבני	ש"ב 2102	WB / 2102	ש"ב 2102, WB / 2102

~~ש"ב 2102 - WB / 2102~~  
~~קריית צור לבני~~

ג. (2%) האם כתוצאה מהוספת החומרה בדרך שתארת נדרשים שינויים פנימיים במנגנון

זיהוי התלויות (Hazard Detection Unit) הקיים? מהם? (הסבר מילולי מפורט אך קצר, אין צורך לממש בלוגיקה)

הסבר מילולי מפורט אך קצר, אין צורך לממש בלוגיקה  
 Hazard al <sup>subtle</sup> ~~מסוכן~~ ~~מסוכן~~ ~~מסוכן~~  
~~מסוכן~~ ~~מסוכן~~ ~~מסוכן~~

ג. (4%) האם כתוצאה מהוספת החומרה בדרך שתארת נדרשים שינויים במנגנון הקידום

הקיים (forwarding/bypass)? מהם? (הסבר מילולי מפורט אך קצר, אין צורך לממש בלוגיקה)

~~מסוכן~~ ~~מסוכן~~ ~~מסוכן~~  
~~מסוכן~~ ~~מסוכן~~ ~~מסוכן~~  
~~מסוכן~~ ~~מסוכן~~ ~~מסוכן~~

קריית צור לבני  
 מסוכן מסוכן מסוכן  
 מסוכן מסוכן מסוכן

שאלה 5 (25%) : TLB זכרון מטמון וזמן גישה לזכרון

נתון מעבד שעובד בשיטת ה Virtual Memory עם TLB's ו- Cache (נפרד לפקודות ולנתונים) שעבורו נתון כי :

- גודל דף הוא 512 בתים.
- זמן גישה לזיכרון הראשי הוא  $t_m = 100nSec$
- זמן גישה ל-Data-Cache וזמן גישה ל-TLB הוא  $t_c = t_{tlb} = 10nSec$
- הזמן הנדרש לחישוב כתובת הוא  $t_{cal} = 10nSec$

נתון קטע הקוד הבא :

```
for I=0 to 2 do
    read A[I];
end do;
read A[5];
read A[0];
```

כאשר A הוא מערך של בתים, ו-  $A = 000003FF_H$  (כלומר זאת כתובת הבית הראשון). שים לב, טבלת הדפים המעודכנת נמצאת בזכרון הראשי ועדכון ה TLB מתבצע ע"י קריאה מהטבלה.

- הנח כי :
- אין Page Faults במהלך ביצוע הקוד (כל נתון מבוקש, נמצא בזיכרון הראשי).
  - בתחילת הביצוע, ה-Data-Cache וה-TLB ריקים.
  - ה-Data-Cache עובד עם כתובות וירטואליות והוא Fully Associative.
  - גודל ה Block ב cache הוא 4 בתים, או מילה אחת.
  - החיפוש ב-Data-Cache, החיפוש ב-TLB, החיפוש בזיכרון וחישוב הכתובת מבוצעים במידת האפשר במקביל (כך שלדוגמא במקרה של Cache miss או TLB miss, כבר התחלנו לחפש את הנתון המבוקש ב-TLB או בזיכרון). גישה לזיכרון במקביל ל-Data-Cache או ה-TLB ניתנת לביטול במידת הצורך.
  - עצם הכנסת נתון חדש ל-Data-Cache או ל-TLB אינה דורשת זמן.
  - קיים מנגנון לחישוב כתובת בטבלת הדפים במקרה של TLB miss וקריאת ערך מאותה כניסה שאינו משתמש ב TLB או ב Cache, אבל עדיין צורך זמן קריאה מזכרון ראשי.

- זמן חישוב הכתובת  $t_{cal}$  הוא הזמן הדרוש לחישוב כתובת גישה לטבלה כלשהי.

בקטע הקוד מבוצעות חמש קריאות של נתונים מהמערך A. לכל קריאה, פרט את השלבים השונים הנדרשים לביצוע הקריאה, את הזמן שאורך כל שלב (במונחים של  $t_m, t_c, t_{tlb}, t_{cal}$ , ואת סכ"ה הזמן (ב-nSec). מלא טבלה כלהלן (לצורך הבהרה בלבד התחלנו את מילוי הטבלה, עליך למשיך ולמלא את כל הטבלה):

מס.	מהות השלב	זמן ( $t_c, t_{cal}, t_{tlb}, t_m$ )	תאור	תזמון (לאחר/במקביל לשלב קודם)
1	קריאת byte ב $3FF_H$			
1.1	חיפוש כתובת וירטואלית ב $3FF_H$ ב TLB	$t_{tlb}$	תוצאה: miss	לאחר השלב הקודם
1.2	חישוב כתובת הכניסה המתאימה בטבלת הדפים של התהליך (נקרא לה EP)	$t_{cal}$	מתקבלת כתובת פיזית EP.	לאחר השלב הקודם
1.3	גישה לזכרון...	$t_m$		
1.4		$t_m$		
1.5				
1.6				
				סה"כ: $270 nSec$
2	קריאת byte ב $400_H$			
2.1	חיפוש כתובת וירטואלית ב TLB	$t_{tlb}$	miss	
	חיפוש כתובת וירטואלית ב Data-Cache	$t_c$	miss	
	חיפוש כתובת וירטואלית ב זיכרון	$t_m$		
				סה"כ: $220 nSec$

מס'	מהות השלב	זמן (t <sub>calc</sub> , t <sub>tbl</sub> )	תאור	תזמון (לאחר/במקביל לשלב קודם)
3	היטענות -> 401			
3.1	אטל כניסה ו' (אילוף)	t <sub>tbl</sub>	הטל פאונד	לרא
3.2	קריאה -> cache	t <sub>c</sub>	הטל כניסה מילוי	לרא 2045 300
<del>4</del>	<del>קריאה 402 byte</del>			
<del>4.1</del>	<del>אילוף כניסה ו' (אילוף)</del>	<del>t<sub>tbl</sub></del>	<del>הטל כניסה</del>	
<del>4.2</del>	<del>קריאה -&gt; cache</del>	<del>t<sub>c</sub></del>	<del>הטל כניסה (מילוי)</del>	
4	היטענות מ 403			
4.1	אילוף כניסה > TLB	t <sub>tbl</sub>	הטל כניסה (מילוי)	לרא 2045 300
4.2	קריאה -> cache	t <sub>c</sub>	הטל כניסה	
5	קריאה מ 3FF			
5.1	אילוף כניסה > TLB	t <sub>tbl</sub>	הטל כניסה	לרא
5.2	קריאה -> cache	t <sub>c</sub>	הטל כניסה (מילוי)	לרא 2045 300

מילוי 400-403 (מילוי) -> cache מילוי  
 אילוף כניסה / TLB -> מילוי כניסה  
 כניסה אילוף כניסה מילוי כניסה  
 קריאה כניסה -> TLB (מילוי) כניסה  
 200-3FF  
 9/1/0

**שאלה 5 (25%): TLB זכרון מטמון וזמן גישה לזכרון**

- נתון מעבד שעובד בשיטת ה Virtual Memory עם TLB's ו- Cache (נפרד לפקודות ולנתונים) שעבורו נתון כי:
- גודל דף הוא 512 בתים.
  - זמן גישה לזיכרון הראשי הוא  $t_m = 100nSec$
  - זמן גישה ל-Data-Cache וזמן גישה ל-TLB הוא  $t_c = t_{tlb} = 10nSec$
  - הזמן הנדרש לחישוב כתובת הוא  $t_{cal} = 10nSec$

נתון קטע הקוד הבא:

```
for I=0 to 2 do
    read A[I];
end do;
read A[5];
read A[0];
```

כאשר A הוא מערך של בתים, ו-  $A = 000003FF_H$  (כלומר זאת כתובת הבית הראשון). שים לב, טבלת הדפים המעודכנת נמצאת בזכרון הראשי ועדכון ה TLB מתבצע ע"י קריאה מהטבלה.

**הנח'כי:**

- אין Page Faults במהלך ביצוע הקוד (כל נתון מבוקש, נמצא בזיכרון הראשי).
  - בתחילת הביצוע, ה-Data-Cache וה-TLB ריקים.
  - Data-Cache עובד עם כתובות וירטואליות והוא Fully Associative.
  - גודל ה Block ב cache הוא 4 בתים, או מילה אחת.
  - החיפוש ב-Data-Cache, החיפוש ב-TLB, החיפוש בזיכרון וחישוב הכתובת מבוצעים במידת האפשר במקביל (כך שלדוגמא במקרה של Cache miss או TLB miss, כבר התחלנו לחפש את הנתון המבוקש ב-TLB או בזיכרון). גישה לזיכרון במקביל ל-Data-Cache או ה-TLB ניתנת לביטול במידת הצורך.
  - עצם הכנסת נתון חדש ל-Data-Cache או ל-TLB אינה דורשת זמן.
  - קיים מנגנון לחישוב כתובת בטבלת הדפים במקרה של TLB miss וקריאת ערך מאותה כניסה שאינה משתמש ב TLB או ב Cache, אבל עדיין צורך זמן קריאה מזכרון ראשי.
- זמן חישוב הכתובת  $t_{cal}$  הוא הזמן הדרוש לחישוב כתובת גישה לטבלה כלשהי.

בקטע הקוד מבוצעות חמש קריאות של נתונים מהמערך A. לכל קריאה, פרט את השלבים השונים הנדרשים לביצוע הקריאה, את הזמן שאורך כל שלב (במונחים של  $t_m, t_c, t_{tlb}, t_{cal}$ ), ואת סכ"ה הזמן (ב-nSec).

מלא טבלה כלהלן (לצורך הבהרה בלבד התחלנו את מילוי הטבלה, עליך למשיך ולמלא את כל הטבלה):

מס.	מהות השלב	זמן ( $t_c, t_{cal}, t_{tlb}, t_m$ )	תאור	תזמון (לאחר/במקביל לשלב קודם)
1	קריאת byte ב $3FF_H$			
1.1	חיפוש כתובת וירטואלית ב $3FF_H$ ב TLB	$t_{tlb}$	תוצאה: miss	לאחר השלב הקודם
1.2	חישוב כתובת הכניסה המתאימה בטבלת הדפים של התהליך (נקרא לה EP)	$t_{cal}$	מתקבלת כתובת פיזית EP.	לאחר השלב הקודם
1.3	גישה לזכרון ראשי	$t_m$		במקביל
1.4	חישוב כתובת זיכרון	$t_{cal}$		במקביל
1.5	גישה ל-Data-Cache	$t_c$		במקביל
1.6	חישוב כתובת זיכרון	$t_{cal}$		במקביל
				סה"כ: 220 nSec
2	קריאת byte ב $400_H$			
2.1	חיפוש כתובת זיכרון	$t_{tlb}$	miss	במקביל
2.2	חישוב כתובת זיכרון	$t_{cal}$		במקביל
2.3	גישה לזיכרון ראשי	$t_m$		במקביל
2.4	חישוב כתובת זיכרון	$t_{cal}$		במקביל
2.5	גישה ל-Data-Cache	$t_c$		במקביל

