

אוניברסיטת תל-אביב
מדעי המחשב
סמסטר א, תשס"ו

מבחן במבנה המחשב
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 לטייטה בלבד ולא תיבדקנה.

משך המבחן: 3 שעות

מלא/י את תעודת הזהות בכל העמודים!

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

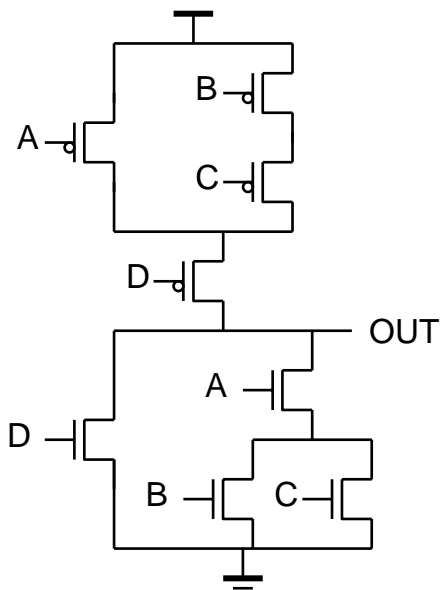
מועד המבחן: 24.2.2006

חומר עזר מותר בשימוש: כל חומר עזר כתוב

בהצלחה !

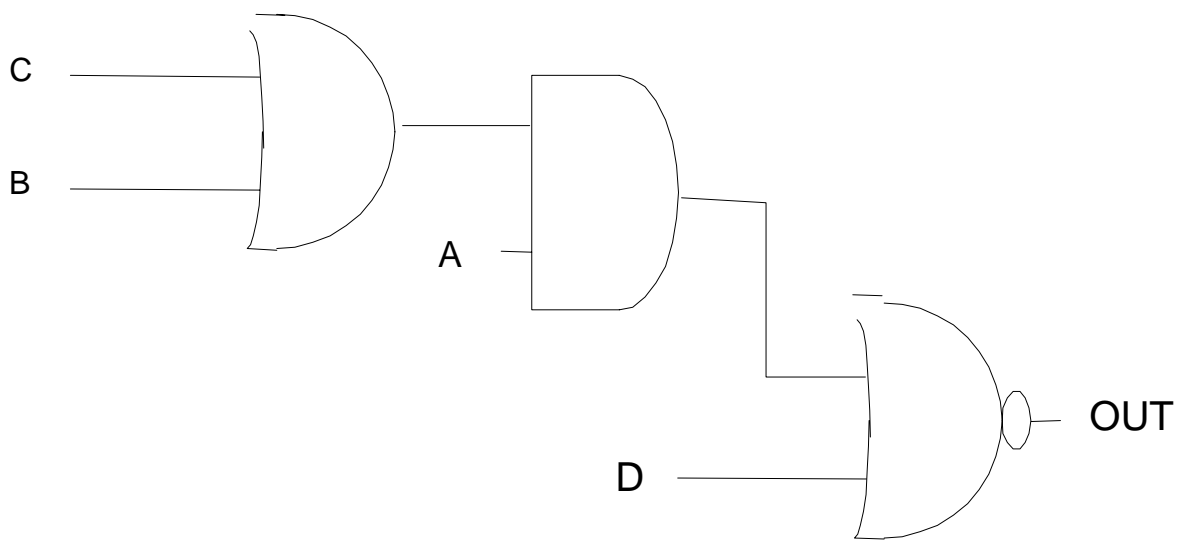
שאלה 1 (15%):

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 האמת 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.



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.

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_loguc_vector(1 downto 0);
        selected: OUT std_logic_vector (3 downto 0);
  END MUX16X4;
```

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 D/A (digital-to-analog), וממיר A/D (analog-to-digital).

על המעגל להיות פשוט ככל הניתן.

שאלה 4 (25%) : Pipeline

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	<u>R1</u> , 100(R4)
SW	R1, 200(<u>R5</u>)

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

בין שתי הפקודות חייב לבוא stall (bubble) יחיד, בהתאם לשרטוט.
ה-stall נוצר ע"י יחידת hazard-detection המזהה תלות ברגיסטר R1 בפקודה שאחרי LW.
במקרה כזה מוכנס תמיד מחזור stall יחיד.

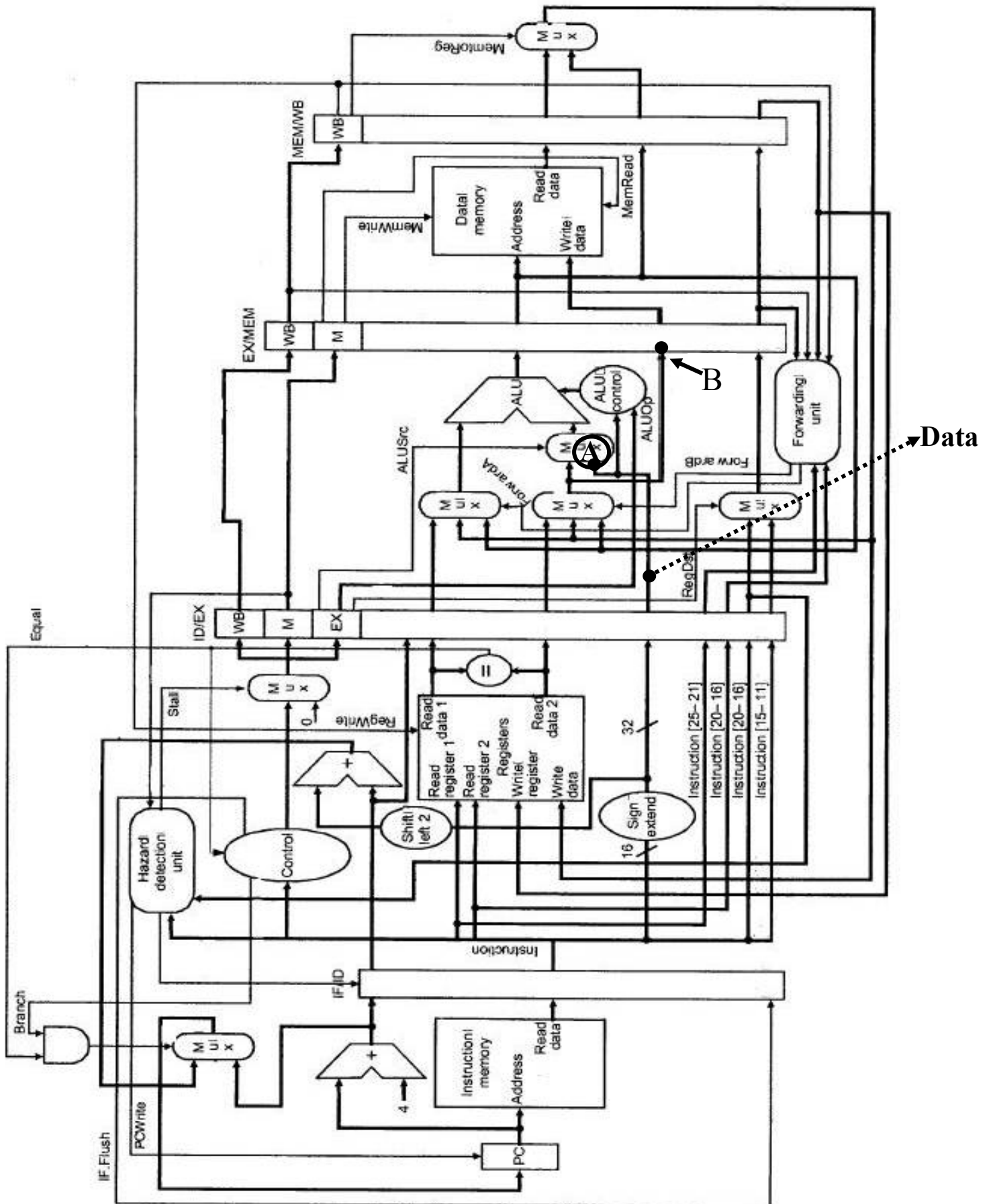
ב. (5%) נניח שהקו המסומן בשרטוט "Data" היה מחובר לנקודה B במקום לנקודה A. איזו בעיה היתה מתעוררת הסבר ?

לא ניתן היה לבצע פקודת SW. הנתון עבור פקודת SW צריך להגיע משדה Rt, ולכן הוא מועבר דרך קו Data אל ה-latch שבין שלבי EX/MEM. במקרה זה שדה Rt לא יועבר ובמקומו יועבר ערך ה - offset בכתובת של פקודת ה-SW.

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 forwarding, Hazard Detection



ג1. (3%) תאר בקצרה את הפתרון המוצע :

הפתרון הוא לבצע קידום (forwarding) של מוצא הזכרון (Memory out) אל כניסת הזכרון (Memory Data In)

ג2. (2%) הסבר מילולי קצר המתאר את החומרה המוצעת ותפקידה :

נוסיף יחידה דומה ל – forwarding אבל בשלב MEM. יחידה זו תזהה את המקרה היחיד בו נדרש קידום מסוג כזה (LW ואחריו SW) ותבצע את הקידום הנדרש. היחידה כוללת בקרה (לזיהוי) mux יחיד, הבורר בין שתי האפשרויות לאחסון נתון בזכרון (Memory Data In) : אחת רגילה (מקו-Data in) והשניה אחרי קידום מ-LW (מיציאת Memory Out).

ג3. (2%) קווי הנתונים הנוספים/ששוננו (גם צייר אותם על השרטוט) :

שם הקו / הנתון עליו	רוחב	יוצא מ....	נכנס אל	הסבר והערות
WB_result	32	יציאת MEM/mux	MUX חדש	קידום תוצאת LW
SW_rt	5	ID/EX latch	EX/MEM latch	שמירת רגיסטר המקור (rt) של פקודת SW

ג4. (2%) קווי הבקרה הנוספים/ששוננו (גם לציין בשרטוט):

שם קו הבקרה	תוצאה כאשר האות = 0	תוצאה כאשר האות = 1	קידוד האות מאותות מוכרים
MEM_forward	נכתב לזכרון הערך מקו Data	נכתב לזכרון הערך מקו WB_result	MEM/WB[MemRead] == 1 and EX/MEM[MemWrite] == 1 and

ג5. (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)

כן. יש למנוע הכנסת stall (bubble) אחרי LW אם הפקודה שאחריה היא SW. (המנגנון בצורתו הרגילה מכניס stall אחרי כל פקודת LW, מבלי לבדוק איזו פקודה באה אחריה).

ג6. (4%) האם כתוצאה מהוספת החומרה בדרך שתארת נדרשים שינויים במנגנון הקידום

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

לא. היחידה החדשה עובדת בשלב MEM ומנגנון ה-forwarding עובד בשלב EX.

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 $t_e=t_{tlb}=10nSec$
- 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:

1. אין Page Faults במהלך ביצוע הקוד (כל נתון מבוקש, נמצא בזיכרון הראשי).
2. בתחילת הביצוע, ה-Data-Cache וה-TLB ריקים.
3. Data-Cache עובד עם כתובות פיסיות והוא Fully Associative.
4. גודל ה Block ב cache הוא 4 בתים, או מילה אחת.
5. החיפוש ב-Data-Cache, החיפוש ב-TLB, החיפוש בזיכרון וחישוב הכתובת מבוצעים במידת האפשר במקביל (כך שלדוגמה 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.
6. עצם הכנסת נתון חדש ל-Data-Cache או ל-TLB אינה דורשת זמן.
7. קיים מנגנון לחישוב כתובת כניסה 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 $t_m, t_e, t_{tlb}, t_{cal}$), 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 ($t_e, t_{cal}, t_{tlb}, t_m$)	תאור	תזמון (לאחר/במקביל לשלב קודם)
1	קריאת byte ב $3FF_H$ (A[0])			
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 לטבלת דפים	t_m	מתקבל תרגום כתובת דף וירטואלי לפיסי	לאחר קודם
1.4	תרגום כתובת וירטואלית ב $3FF_H$ ב TLB (המעודכן)	t_{tlb}	מתקבלת כתובת פיסית של $3FF$	לאחר קודם
1.5	גישה ל cache לכתובת פיסית	t_e	miss	במקביל
1.6	הבאת הנתון מהזכרון ועדכון cache במקביל	t_m		במקביל
				סה"כ: $nSec_220$
2	קריאת byte ב 400_H (A[1])			
2.1	חיפוש כתובת וירטואלית ב 400_H ב TLB ותרגומה	t_{tlb}	תוצאה: miss (כיוון שגודל דף 512 בתים)	לאחר השלב הקודם
2.2	חישוב כתובת הכניסה	t_{cal}	מתקבלת כתובת פיזית EP2.	לאחר השלב הקודם

			המתאימה בטבלת הדפים של התהליך (נקרא לה EP2)	
לאחר קודם	מתקבל תרגום כתובת דף וירטואלי לפיסי וה TLB מתעדכן	t_m	גישה לזכרון לטבלת דפים לקריאת EP2	2.3
לאחר קודם	מתקבלת כתובת פיסי של 400	t_{tlb}	תרגום כתובת וירטואלית ב TLB (המעודכן) 400_H	2.4
במקביל	Miss מכיוון שזה לא אותו block	t_c	גישה ל cache לכתובת פיסי שהתקבלה	2.5
במקביל		t_m	הבאת הנתון מהזכרון ועדכון cache במקביל	2.6
סה"כ: nSec_220_				
			קריאת byte ב 401_H (A[2])	3
לאחר השלב הקודם	תוצאה: Hit (אותו דף כמו הקודם) מתקבלת הכתובת הפיסי	t_{tlb}	חיפוש כתובת וירטואלית ב TLB ותרגומה 401_H	3.1
לאחר השלב הקודם	Hit	t_c	גישה ל cache לכתובת פיסי שהתקבלה	3.2
סה"כ: nSec_20_				
			קריאת byte ב 404_H (A[5])	4
לאחר השלב הקודם	תוצאה: Hit	t_{tlb}	חיפוש כתובת וירטואלית ב TLB 404_H	4.1
לאחר השלב הקודם	Miss מכיוון שזה לא אותו block	t_c	גישה ל cache לכתובת פיסי שהתקבלה	4.2
לאחר קודם		t_m	הבאת הנתון מהזכרון ועדכון cache במקביל	4.3
סה"כ: nSec_110_				
			קריאת byte ב $3FF_H$ (A[2])	5
לאחר השלב הקודם	תוצאה: Hit (ראשון) מתקבלת הכתובת הפיסי	t_{tlb}	חיפוש כתובת וירטואלית ב TLB ותרגומה $3FF_H$	5.1
לאחר השלב הקודם	Hit	t_c	גישה ל cache לכתובת פיסי שהתקבלה	5.2
סה"כ: nSec_20_				

פתרון כאשר ה cache הוא בכתובות virtual בעמוד הבא (כלומר הנחה שה- TLB בא אחרי ה cache בניגוד למה שנלמד בכיתה).

פתרון כאשר ה cache הוא בכתובות virtual (כלומר הנחה שה- TLB בא אחרי ה cache בניגוד למה שנלמד בכיתה).

מס.	מהות השלב	זמן (tc,teal, ttlb,tm)	תאור	תזמון (לאחר/במקביל לשלב קודם)
1	קריאת byte ב 3FFH (A[0])			
1.1	חיפוש כתובת וירטואלית ב 3FFH ב cache	tc	תוצאה: miss	
1.2	חיפוש כתובת הנתון ב TLB	ttl	תוצאה: miss	במקביל לקודם
1.3	חישוב כתובת הכניסה המתאימה בטבלת הדפים של התהליך (נקרא לה EP)	teal	מתקבלת כתובת פיזית EP.	לאחר השלב הקודם
1.4	גישה לזכרון לטבלת דפים	tm	מתקבל תרגום כתובת דף וירטואלי לפיסי	לאחר קודם
1.5	תרגום כתובת וירטואלית ב 3FFH ב TLB (המעודכן)	ttl	מתקבלת כתובת פיזית של 3FF	לאחר קודם או במקביל
1.6	הבאת הנתון מהזכרון ועדכון ב cache במקביל	tm		במקביל
				סה"כ: nSec_220_
2	קריאת byte ב 400H (A[1])			
2.1	חיפוש כתובת וירטואלית ב 400H ב cache	tc	תוצאה: miss (כיוון ש block שונה)	לאחר השלב הקודם
2.2	חיפוש כתובת הנתון ב TLB	tTL	תוצאה miss מכיוון שגודל דף 512 בתים.	במקביל
2.3	חישוב כתובת הכניסה המתאימה בטבלת הדפים של התהליך (נקרא לה EP)	teal	מתקבלת כתובת פיזית EP.	לאחר קודם
2.4	גישה לזכרון לטבלת דפים	tm	מתקבל תרגום כתובת דף וירטואלי לפיסי	לאחר קודם
2.5	תרגום כתובת וירטואלית ב 400H ב TLB (המעודכן)	ttl	מתקבלת כתובת פיזית של 400	לאחר או במקביל לקודם
2.6	הבאת הנתון מהזכרון ועדכון ב cache במקביל	tm		במקביל
				סה"כ: nSec_220_
3	קריאת byte ב 401H (A[2])			
3.1	חיפוש הבית המבוקש ב 401H ב cache וירטואלית	tc	תוצאה: Hit (אותו block כמו הקודם)	
				סה"כ: nSec_10_
4	קריאת byte ב 404H (A[5])			
4.1	גישה ל cache לכתובת וירטואלית	tc	Miss מכיוון שזה לא אותו block	
4.2	חיפוש כתובת וירטואלית ב 404H ב TLB	ttl	תוצאה: Hit	במקביל לשלב הקודם
4.3	הבאת הנתון מהזכרון ועדכון ב cache במקביל	tm		לאחר קודם
				סה"כ: nSec_110_
5	קריאת byte ב 3FFH (A[0])			
5.1	גישה ל cache להבאת הנתון מכתובתו הוירטואלית	tc	Hit	
				סה"כ: nSec_10_