



הטכניון - מכון טכנולוגי לישראל

## מבנה מחשבים (236267)

מבחן מסכם מועד א'

10 פברואר 2020

**מרצים:** ליהוא רפופורט, עדי יועז.

**מתרגלים:** פרנק סלה, איתי רביד.

מס. ת.ז. :
------------

- משך הבחינה: שלוש שעות.
- מותר כל חומר עזר.
- יש לכתוב את התשובות בטופס הבחינה בלבד ובמקום המיועד לתשובה.
- יש לכתוב בקיצור ככל האפשר, אך יש לנמק כל תשובה.
- בדקו שבטופס שבידכם יש 14 עמודים כולל עמוד זה.
- המבחן כולל 5 שאלות, יש לענות על כולן.
- יש לכתוב בעט בלבד (לא בעיפרון!), ולא בעט בצבע אדום.

שאלה 1	OOO	/ 32
שאלה 2	BP	/ 10
שאלה 3	power	/ 8
שאלה 4	VM/cache	/ 40
שאלה 5	MESI	/ 10
ציון סופי		/100

**בהצלחה !**

## שאלה 1 – Out-Of-Order Execution (32 נק')

א. (26 נק') יש למלא את הטבלה שבהמשך. לכל פקודה יש לרשום:

- |                                       |   |  |
|---------------------------------------|---|--|
| <b>מולאו בחלקם, יש להשלים את החסר</b> | { | • R3, R2, R1 – ערכי הרגיסטרים הארכיטקטוניים לאחר commit של הפקודה. עבור פקודה ממסלול שגוי, הערך שחושב עבור הרגיסטר הפיזי בזמן execute. |
|                                       |   | • addr – כתובת הגישה לזיכרון – עבור פקודות load ו-store בלבד.  |
|                                       |   | • data – ערך זיכרון שנקרא או נכתב – עבור פקודות load ו-store בלבד.   |

- T alloc: הזמן בו מבוצעת אלוקציה: עד 3 פקודות בכל מחזור, החל מ-  $t=1$ . ניתן לבצע אלוקציה רק כאשר **לכל הפקודות** שמספק ה-frontend במחזור יש מקום ב-ROB וב-RS. ROB יש 8 **כניסות**. כל פקודה (כולל פקודות Store) תופסת מקום אחד ב-ROB ומקום אחד ב-RS.
- src1, src2: מספרי הרגיסטרים המשמשים כ-sources לפקודה:  $P_i$  עבור רגיסטר פיזי, ו- $R_i$  במידה וקוראים ישירות את הרגיסטר הארכיטקטוני. עבור store: src1 – הרגיסטר המשמש לחישוב הכתובת. src2 – הרגיסטר המכיל את הנתון.
- T src1 ready, T src2 ready: הזמן בו מוכן כל אחד ערכי ה-sources לפקודה. אם ה-src כבר מוכן בזמן האלוקציה, אז זמן זה יהיה שווה לזמן האלוקציה. אחרת, זמן זה שווה ל-T data ready של הפקודה שמחשבת את הערך של ה-src.
- T exe: הזמן בו הפקודה נשלחת לביצוע. הניחו כי ישנן אינסוף יחידות ביצוע.
  - פקודה יכולה להיכנס לביצוע לכל המוקדם במחזור שלאחר האלוקציה.
  - פקודה נכנסת לביצוע במחזור השעון שלאחר המחזור בו כל ה-src-ים מוכנים. פקודת store נכנסת לביצוע במחזור השעון שלאחר המחזור בו src1 (המשמש לחישוב הכתובת) מוכן.
- Load block code: עבור load שנשלח לביצוע בזמן  $t=T_{exe}$ , או שהוסר עבורו תנאי חסימה קודם בזמן  $t$ , תנאי החסימה נבדקים בזמן  $t+1$  לפי הסדר (רישמו את כל תנאי החסימה לפי הסדר):
  1. unknown store address (התנאי מוסר בזמן  $t = T_{addr\ ready}$  של ה-store החוסם)
  2. waiting for store data (התנאי מוסר בזמן  $t = T_{data\ ready}$  של ה-store החוסם)
- T addr ready: ממלא עבור פקודות load ו-store בלבד:  **$T_{exe}+1$** 
  - עבור load ו-store המבוצעים באותו זמן  $t$ : תנאי החסימה של ה-load נבדקים בזמן  $t+1$ , בזמן זה הכתובת של ה-store ידועה, ולכן ה-load לא נחסם על unknown store address ע"י ה-store.
- T data ready:
  - עבור פקודות ALU: Add, Sub,  **$T_{exe}+1$** : Mul,  **$T_{exe}+5$** .
  - עבור load שהוסרו עבורו כל תנאי החסימה בזמן  $t$  או שבוצע בזמן  $t$  ולא נחסם:
    - במידה וה-load פוגע ב-cache, או שיש store to load forwarding: בזמן  $t+4$ .
    - אחרת, במידה ובוצע load/store לאותה שורה ב-cache בזמן  $t' < t$ : בזמן  **$\max(t'+8, t+4)$** .
    - אחרת, בזמן  **$t+8$** .
  - עבור store:  **$T_{src2\ ready} + 1$**
- עבור פקודת Jump עם חיזוי שגוי, מבוצע flush בזמן  $T_{exe}+1$ , והפקודות מהמסלול הנכון מבצעות אלוקציה החל מזמן  **$T_{exe}+6$**  (במידה ואין סיבה אחרת שמעכבת את האלוקציה).

- T commit: הזמן בו הפקודה מבצעת commit. ניתן לבצע commit לעד 3 פקודות בכל מחזור. פקודה (למעט store) יכולה לבצע commit החל מזמן  $T \text{ data ready} + 1$ . פקודת store יכולה לבצע commit החל מזמן  $\max(T \text{ addr ready}, T \text{ data ready}) + 1$ . פקודה מוצאת מה-ROB במחזור שלאחר ה-commit שלה, וכבר במחזור זה פקודה חדשה יכולה לבצע alloc ולהשתמש במקום שהתפנה.
- #ROB entries: מספר הכניסות התפוסות ב-ROB לאחר שהפקודה ביצעה אלוקציה. יש להתחשב גם בפקודות שהוצאו מה-ROB כתוצאה מ-commit.

• הנחות:

- בטבלה רשומות הפקודות שמבצעות אלוקציה, כולל פקודות מהמסלול השגוי.
  - הכתובות הן פיזיות (אין צורך בתרגום). כל הערכים המספריים הם בבסיס 16.
  - L1 data cache: גודל שורה  $32_{10}B = 20_{16}B$ . write allocate. ריק בתחילת הביצוע.
  - בתחילת הביצוע כתובת N בזכרון מכילה את הערך N.
  - פקודה המסומנת ב-\* בטבלה אינה מסופקת ע"י ה-frontend באותו מחזור של הפקודה הקודמת לה, אלא במחזור הבא. בכל מקרה אחר, ה-frontend מספק 3 פקודות למחזור.
- ב. (6 נק') עבור ריצת התוכנית שמצאתם בסעיף א', מלאו לכל מחזור שעון שבטבלה שבסעיף זה את נתוני ה-Top level breakdown. (מחזורי השעון והנתונים הממולאים מתייחסים לזמן האלוקציה).
- מחזורי אלוקציה במהלך מסלול שגוי שבהם לא ניתן לבצע אלוקציה כתוצאה ממספר לא מספיק של כניסות ב-ROB יש לסמן כ-Backend bound. במידה ובמחזור האלוקציה האחרון נותרו בתוכנית פחות מ-3 פקודות, יש להחשיבו כ-Frontend bound.
  - ייתכן שיש בטבלה עמודות מיותרות, אם כן יש להשאירן ריקות ולא להתייחס אליהם.

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13
Back-end Stall													
Alloc Slot 0													
Alloc Slot 1													
Alloc Slot 2													
Frontend Bound													
Backend Bound													
Retiring													
Bad Speculation													

\_\_\_\_\_ Frontend Bound ארוך  
 \_\_\_\_\_ Bad Speculation ארוך  
 \_\_\_\_\_ Backend Bound ארוך  
 \_\_\_\_\_ Retiring ארוך

Pdst	instruction	R1	R2	R3	addr	data	src1	src2	T alloc	#ROB entries	T src1 ready	T src2 ready	T exe	T addr ready	T data ready	block code	T commit
1	mul R2 $\leftarrow$ R1 $\times$ 2	10	20	30													
2	store m[R2+20] $\leftarrow$ R3	10	20	30	40	30											
3	if (R3>0) PC $\leftarrow$ 100 חיצוי נכון	10	20	30													
4	load R3 $\leftarrow$ m[R3+20]	10	20	50	50	50											
5	store m[R2+50] $\leftarrow$ R3	10	20	50	70	50											
6	load R2 $\leftarrow$ m[R1+60] *	10	50	50	70	50											
7	if (R1>0) PC $\leftarrow$ 200 חיצוי שגוי	10	50	50													
8	load R1 $\leftarrow$ m[R1+10]																
9	sub R1 $\leftarrow$ R1 – 10																
10	add R3 $\leftarrow$ R2 + R1																
11	sub R2 $\leftarrow$ R3 – R2																

**שאלה 2 – חיזוי קפיצות (10 נק')**

נתונה סידרת הקפיצות המחזורית הבאה:  $1100011 \ 1100011 \ \dots$ . נניח חזאי מסוג local, שבו ההיסטוריה באורך 2 מצביעה אל מערך שבו בכל כניסה מונה 2 סיביות עם רוויה.

א. (7 נק') האם אחוז החיזוי במצב היציב תלוי במצב ההתחלתי של המונים? אם כן, הסבירו מהו אחוז החיזוי במצב היציב בכל אחת מהאפשרויות השונות. אם לא, הסבירו מדוע לא, וכן הסבירו מהו אחוז אחוז החיזוי במצב היציב.

This image shows a full page of blank graph paper. The grid consists of thin, light gray horizontal and vertical lines that intersect to form a uniform pattern of small squares across the entire surface. There are no margins, text, or other markings on the paper.

ב. (3 נק') מהו אורך ההיסטוריה המינימלי המאפשר חיזוי מושלם של הסידרה הנתונה במצב היציב?  
הסבירו.

[illegible]

### **שאלה 3 – Power/Performance (8 נק')**

ליבה קטנה	ליבה גדולה	
1mm <sup>2</sup>	3mm <sup>2</sup>	שטח
0.2W	0.5W	Leakage Power
500pF	1000pF	Cdyn עבור אפליקציה A
2	3	IPC עבור אפליקציה A

נתונים שני סוגי ליבות:  
ליבה גדולה וליבה קטנה,  
המריצים אפליקציה A.

בטבלה הבאה נתונות נקודות מתח ותדר אפשריות עבור הליבה הגדולה והקטנה (העמודות הנוספות בטבלה לשימוכם בפתרון):

מחבר Volt	ליבה קטנה תדר GHz	ליבה גדולה תדר GHz					
0.9	1.7	2.2					
1.0	2.2	2.8					
1.1		3.5					

יש לתכנן מעבד המכיל ליבות גדולות או קטנות (אך לא שילוב של ליבות גדולות וקטנות). המעבד יריץ את אפליקציה A, על-ידי מספר רב של משתמשים בו-זמנית. במהלך ריצה של האפליקציה מבוצעות עד  $10^9 \times 60$  פקודות. יש לתכנן מעבד שיאפשר לכמה שיותר משתמשים להריץ את האפליקציה במקביל, ובנוסף להבטיח שעבור כל משתמש האפליקציה תסתיים תוך לכל היותר 8 שניות. על-מנת להגביל את עלות המעבד, נדרש שהשטח שלו לא יעלה על  $30\text{mm}^2$ . על ההספק הכולל להיות נמוך מ-25W. תארו את מבנה המעבד האופטימלי, פרטו והסבירו את החישוב.

This image shows a full page of blank graph paper. The grid consists of small, uniform squares formed by thin, light gray lines. There are no margins, text, or other markings on the page.

## שאלה 4 – זיכרון וירטואלי ו-cache (40 נק')

נתון מעבד דמוי x86 העובד במבנה הכתובת הבא:

63	47 46	36 35	25 24	14 13	0
Sign Ext	PML3	PML2	PTE	offset	

- כל טבלת דפים (בכל הרמות) היא בגודל דף.

המעבד כולל:

- TLB בגודל של 128 כניסות, 8 ways, מדיניות LRU. זמן גישה (ל-hit או miss) של TLB miss פונים ל-PMH.
- PMH הכולל translation caches, PML2\$ ו-PML3\$, שהגישה אליהם מתבצעת במקביל. זמן גישה (ל-hit או miss) 3 מחזורים. בכל אחד מהם 2 כניסות Fully Associative במדיניות LRU.
- L1 cache: 32KB, 8 ways, 64B בשורה, מדיניות LRU. זמן גישה (ל-hit או miss) 4 מחזורים.
- L2 cache: 512KB, 8 ways, 64B בשורה, מדיניות LRU. זמן גישה (ל-hit או miss) 10 מחזורים.
- זמן גישה לזיכרון הוא 100 מחזורים.
- לאחר ביצוע תרגום, מידע התרגום המעודכן מוכנס לכל המבנים שאליהם ניגשו במהלך התרגום.

נתונה התוכנית הבאה (הקידומת 0x מציינת מספר בבסיס 16):

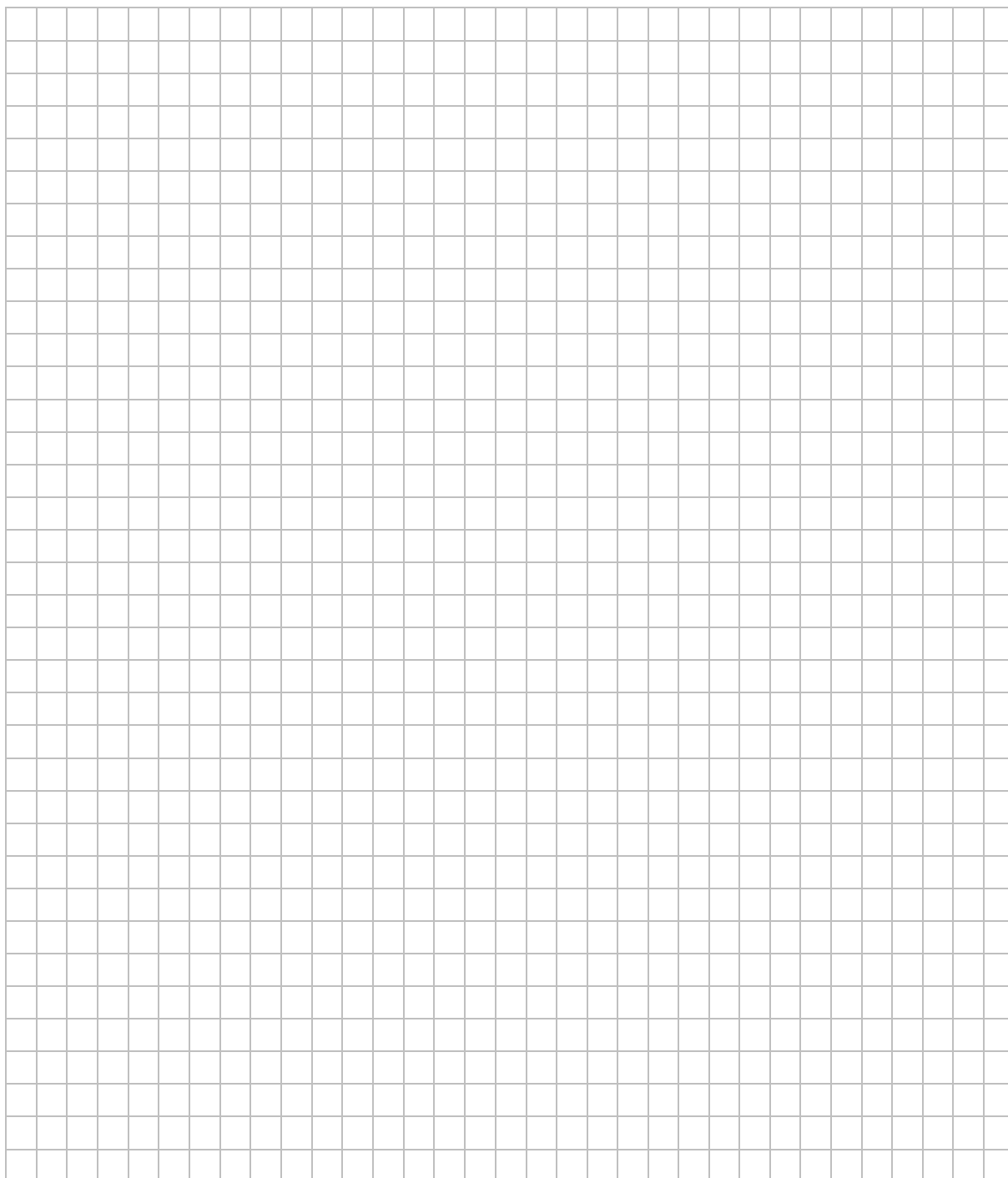
```
for (int j=0; j< 0x1,000; j++)
    for (int i=0; i< 0x1,000; i++)
        S+= A[i][j] ^ (1/i+1/j);
```

- המערך  $A[0x1,000][0x1,000]$  מתחיל בכתובת 0x400,000.
- המערך פרוש בקטע רציף בזיכרון לפי שורות: תחילה כל האיברים של שורה  $i=0$ , אחריה כל האיברים של שורה  $i=1$ , וכו'. כל איבר במערך הוא בגודל 4 בתים.
- $S, i, j$  מאוחסנים כל אחד ברגיסטר.
- מערכת ההפעלה מקצה דפים עבור המערך בזמן ההכרזה על המערך. הדפים מוקצים באזור רציף בזיכרון הפיזי, החל מכתובת 0x1,000,000.
- מערכת ההפעלה מקצה את טבלאות התרגום הנדרשות למיפוי המערך  $A$  באזור רציף בזיכרון הפיזי, החל מכתובת 0x10,000,000, שבה מתחילה טבלת PML3. לאחר מכן טבלאות PML2 הדרושות למיפוי  $A$ , ולבסוף טבלאות התרגום הדרושות למיפוי  $A$  לפי הסדר.
- בשאלה זו נתעלם מגישות לצורך הבאת הקוד עצמו. בתחילת הביצוע כל זיכרונות המטמון ריקים.
- שימו לב ש-L1 משמש הן עבור הנתונים והן עבור כניסות בטבלת הדפים של-PMH מביא מהזיכרון.

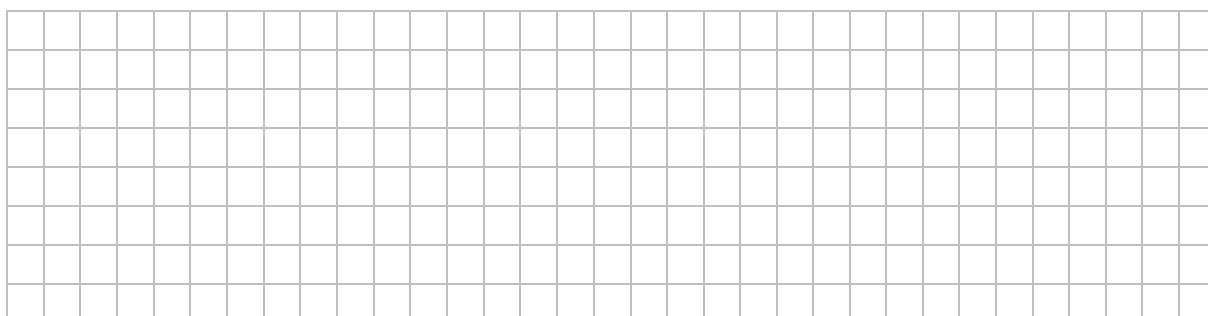
This image shows a full page of blank graph paper. The grid consists of thin, light gray horizontal and vertical lines that intersect to form small squares across the entire surface. There are no margins, text, or other markings on the paper.

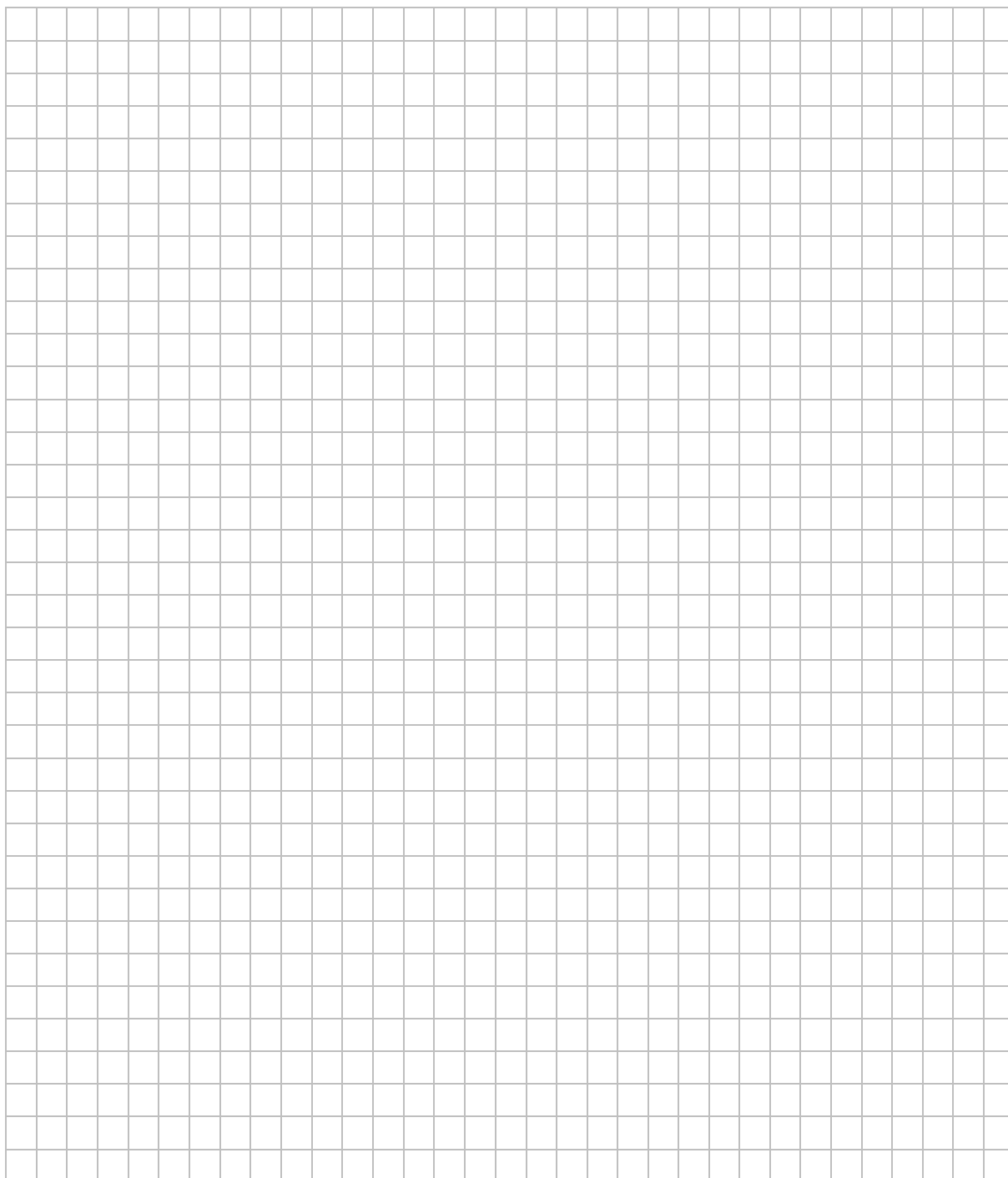
[illegible]



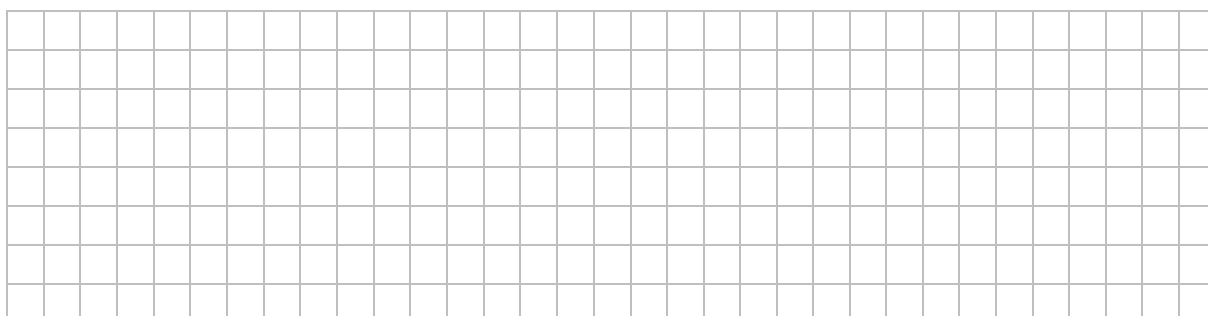


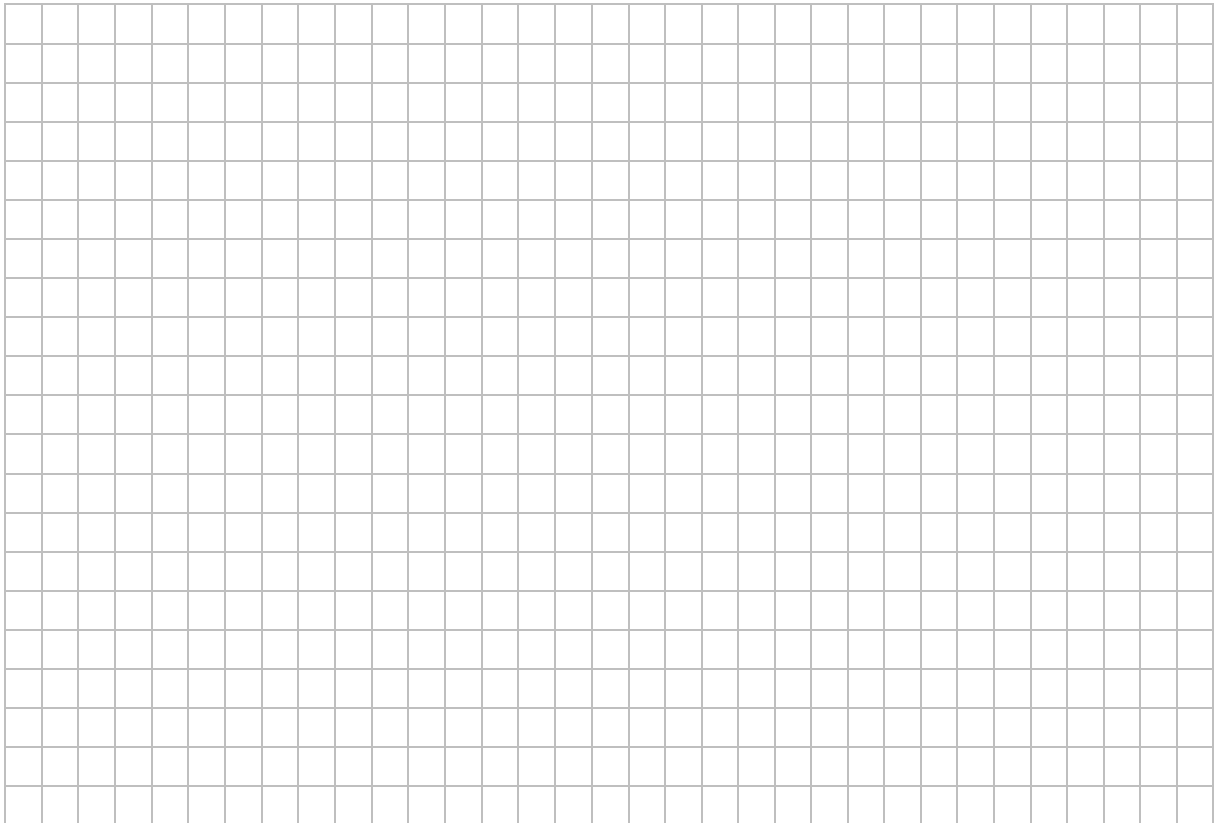
ג. (8 נק') מהו משך הזמן הכולל המדויק הנדרש לביצוע תרגום כל הכתובות באיטרציה הראשונה של  $j$  ( $j=0$ ) ? הסבירו.





ד. (4 נק') מהו משך הזמן הכולל המדויק הנדרש לביצוע תרגום כל הכתובות באיטרציה השנייה של  $j$  ( $j=1$ ) ? הסבירו.

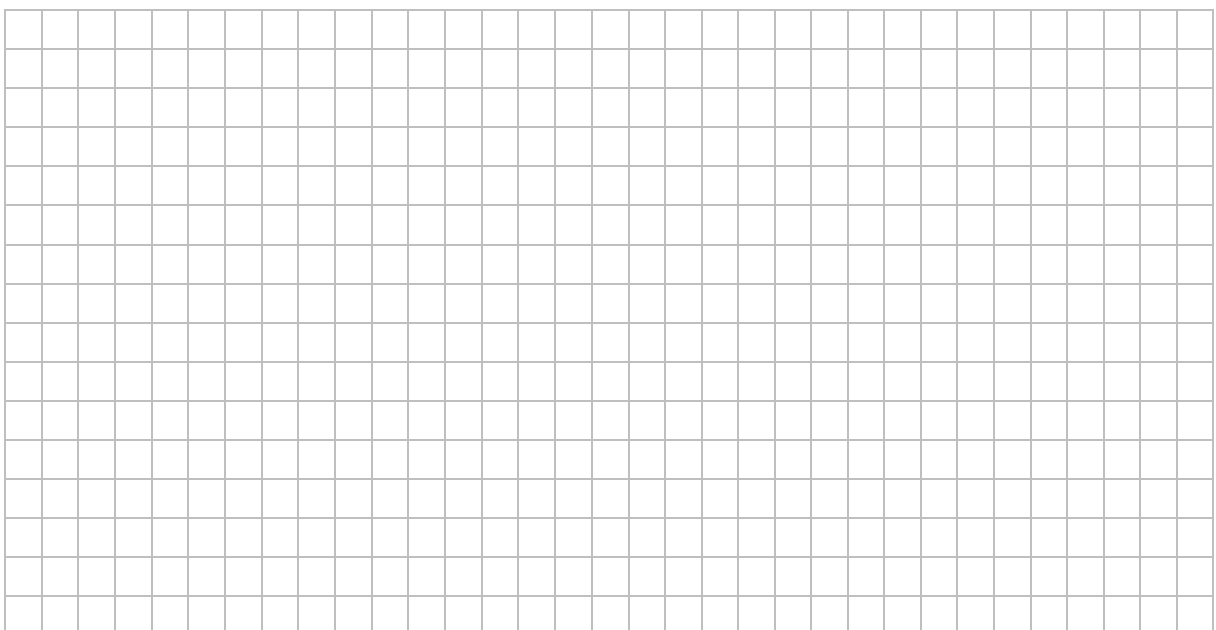


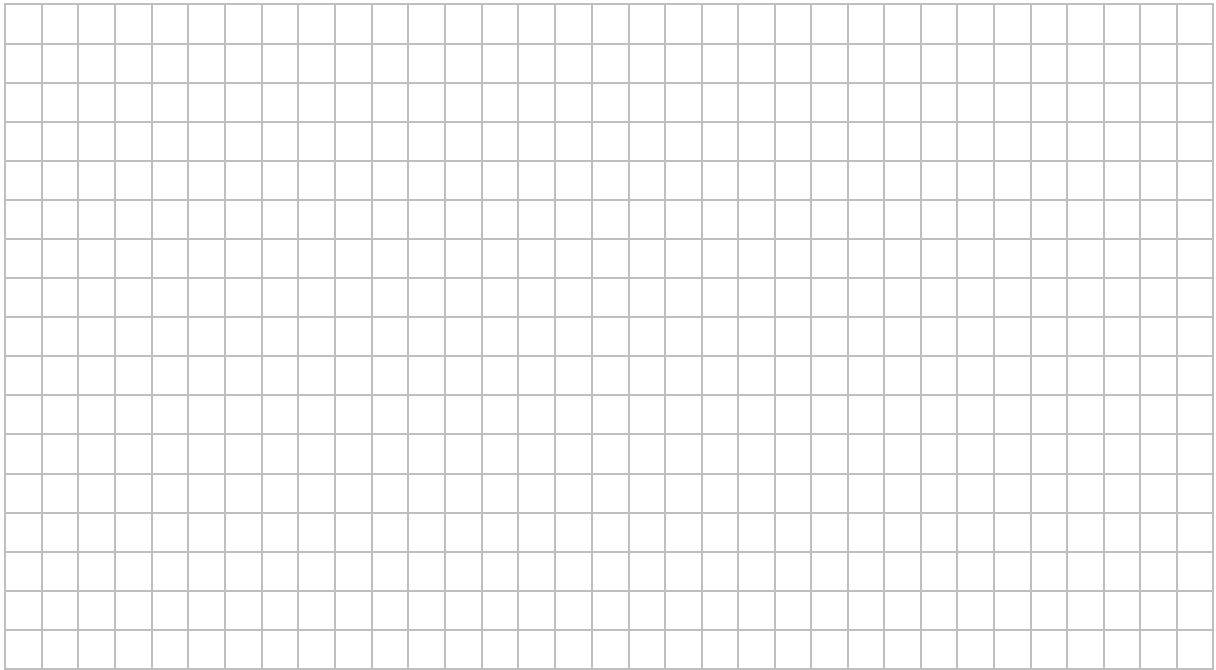


הסעיפים הבאים בשאלה מתייחס לתוכנית הבאה:

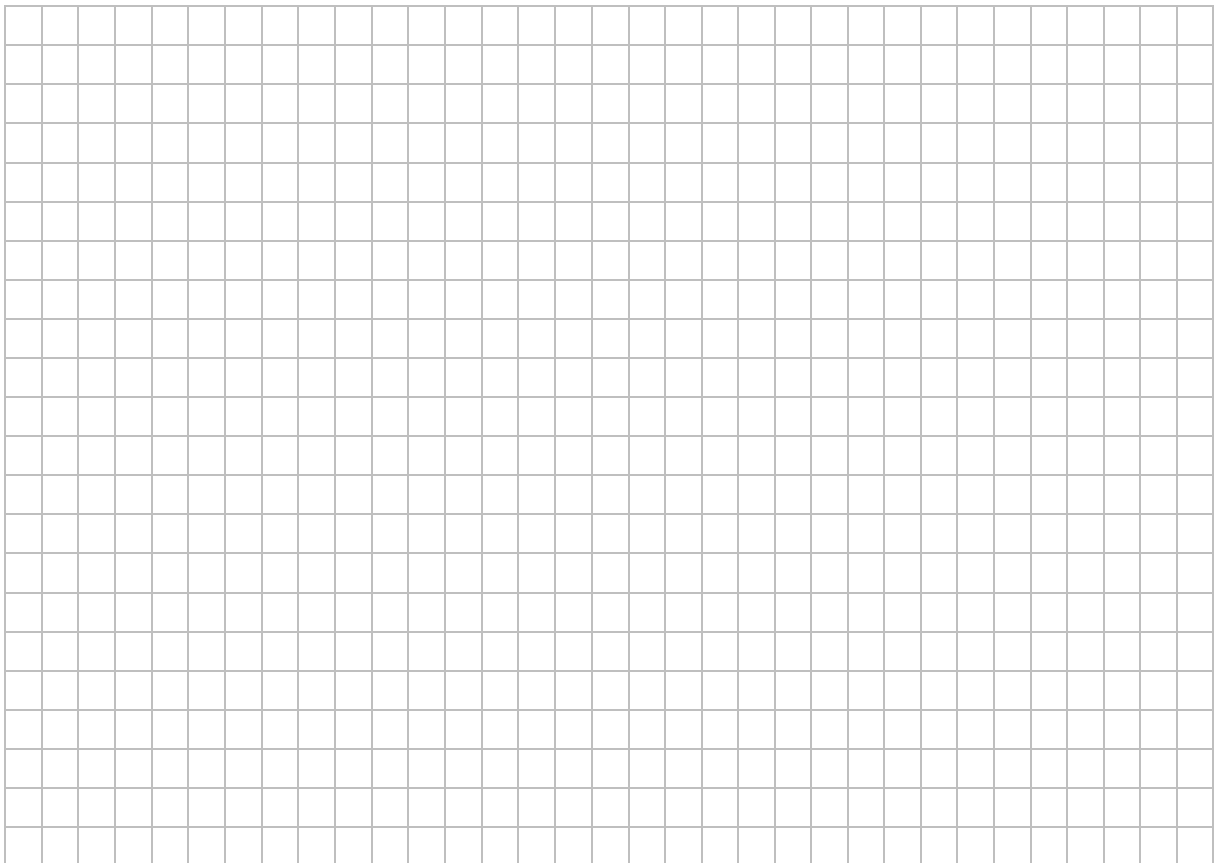
```
for (int k=0; k< 0x10; k++)  
    for (int i=0; i< 0xC; i++)  
        for (int j=0; j< 0x40; j++)  
            S+= A[i][j] ^ (1/i+1/j+1/k);
```

ה. (6 נק') מהו משך הזמן הכולל המדויק הנדרש לביצוע תרגום כל הכתובות בתוכנית? הסבירו.

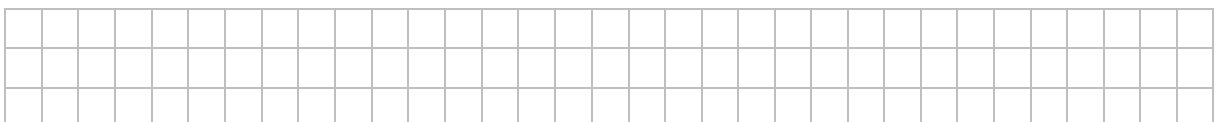


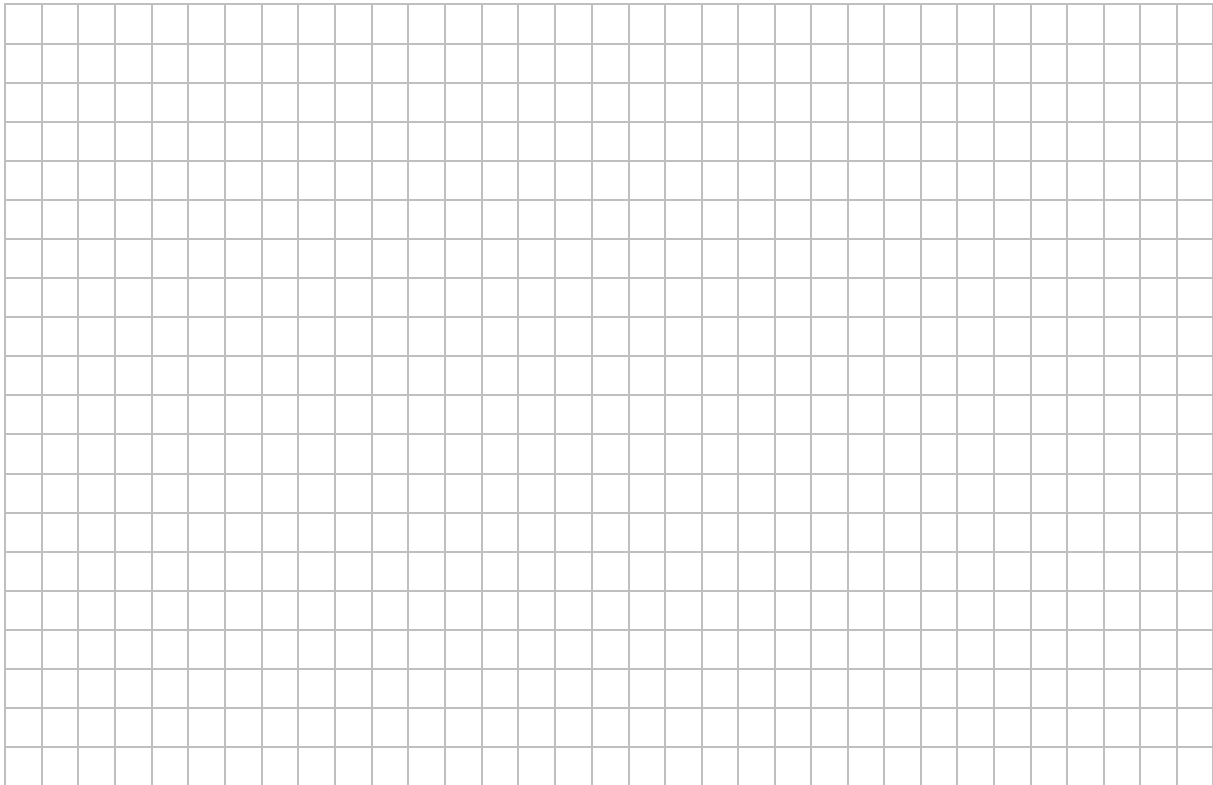


ו. (5 נק') מהו ה-L1 hit rate במהלך התוכנית כולה עבור פניות לנתונים? הסבירו



ז. (5 נק') מהו ה-L2 hit rate במהלך התוכנית כולה עבור פניות לנתונים? הסבירו





### **שאלה 5 – פרוטוקול MESI (10 נק')**

נתונה מערכת עם שלושה מעבדים (P1, P2, P3) בפרוטוקול MESI. לכל מעבד יש L1 data cache משלו, וכל המעבדים חולקים L2 cache משותף. מתקיימת הכלה בין L2 לבין כל ה-L1s. ה-cache-ים עובדים במדיניות write allocate + write back.

L1 שולח ל-L2 את סוגי ההודעות הבאות:

- Read (A): בקשה לקבלת ה-data שבכתובת A.
- RFO (A) (Read for ownership): בקשה לקבלת ה-data שבכתובת A, הכוללת גם ownership על ה-data.
- SR (A) (Snoop response) – אישור על קבלת snoop – נשלח רק במידת הצורך: או שנדרשת שליחת data (ואז כולל גם data), או במקרה ש-L2 לא יכול להבין בעצמו את המצב.
- WB (A) (Write Back) – נשלח במקרה של פינוי שורה שהייתה במצב M, כולל ה-data.

L2 שולח את סוגי ההודעות הבאות:

- Data (A): Data שנשלח אל L1 בתגובה ל-Read או RFO, כולל גם את ה-MESI state.
- Snoop(A): נשלח אל L1, כולל את מצב ה-MESI שאליו המעבד נדרש לעבור (I או S). עשוי לדרוש אישור שכולל במידת הצורך גם Data.
- Read (A): נשלח לזיכרון במקרה של L2 miss לכתובת A.
- WB(A): כתיבת Data בחזרה לזיכרון.

הזיכרון שולח הודעה מסוג Data (A): Data – Data שנשלח אל ה-L2.

הודעה מ-L1 אל L2 או מ-L2 אל L1 אורכת 10ns, הודעה בין הזיכרון מ/אל L2 אורכת 100ns. במידה ואין תלות בין הודעות, ההודעות נשלחות במקביל. רק הודעות חיוניות נשלחות (ששולח הודעה מסיק שההודעה מוסיפה למקבל ההודעה מידע חיוני). L2 יכול לשלוח הודעות למספר מעבדים במקביל.

א. (6 נק') נתונה סידרת פעולות על כתובת A. מלאו את הטבלה במצב ה-MESI של כתובת A ב-L1 cache בכל אחד מהמעבדים, ובמצב ה-Core Valid Bits שב-L2 cache לאחר ביצוע הפעולה. כמו כן מלאו את סכ"ה הזמן לשליחת ההודעות. במקרה שנשלחות מספר הודעות במקביל יש להחשיב כזמן של הודעה אחת. במקרה ולא נשלחה אף הודעה, מלאו אפס.

	L1 State for A			L2 CVBs			Total message time
	P1	P2	P3	P1	P2	P3	
Initial State	I	I	E				
L2 evicts A							
P1 reads A							
P2 reads A							
P3 writes A							
P2 writes A							

ב. (4 נק') נתונה מערכת שבה הכתיבות הן נדירות ולעומת זאת מצב שבו מספר מעבדים קוראים את אותה הכתובת הוא נפוץ. הציעו שינוי בפרוטוקול שיקטין את משך זמן ההודעות הממוצע במערכת זו. כיצד יתבטא השינוי בטבלה שבסעיף א' ? הסבירו.

This image shows a full page of blank graph paper. The grid consists of thin, light gray horizontal and vertical lines that intersect to form a uniform pattern of small squares across the entire surface. There are no margins, text, or other markings on the paper.