

BAR-ILAN UNIVERSITY (RA)

Faculty of Engineering

Ramat-Gan 52900, Israel



Tel: 03-5317722

engbi@mail.biu.ac.il

אוניברסיטת בר-אילן (ע"ר)

הפקולטה להנדסה

רמת-גן 52900

מבנה מחשבים ספרתיים

תשע"ט סמסטר ב' מועד ג'

83-301

מרצה: פרופ' שמואל וימר

מתרגל: מר בנימין פרנקל

- יש לקרוא היטב את ההוראות.
- חובה לענות על כל השאלות.
- ציון מקסימלי בבחינה: 100 נקודות.
- יש לנמק היטב את כל תשובותיכם, לפרט חישובים ולהסבירם.
- יש להקפיד על כתב יד קריא!
- יש לרשום תשובות בתוך הטבלאות המצורפות במקום שנדרש.
- חומר עזר מותר בשימוש: מחשבון, ספר הקורס ושקפי ההרצאות בלבד! כל חומר עזר אחר אסור בהחלט!
- משך הבחינה: שלוש שעות.
- יש לצרף את שאלוני הבחינה למחברת!

בהצלחה!

שאלה א' 55 נק' (זכרון מטמון)

1. נתונים שני מעבדים הפועלים באותו תדר ומריצים אותה תכנית בדיוק.
 במעבד הראשון miss penalty הינו $0.2\mu Sec$, ואילו cache hit אורך $2.0nSec$. ידוע ש hit קורה ב 90% מהזמן.
 במעבד השני miss penalty הינו $0.5\mu Sec$, ואילו cache hit אורך $1.0nSec$. ידוע ש hit קורה ב 95% מהזמן.
- א. (5 נק') חשב בצורה מלאה את ביצועי הזכרון של שני המעבדים. לאיזה מהם ביצועים טובים יותר?
- ב. (5 נק') עבור איזה cache hit ביצועי המעבדים משתווים?

תשובה:

In the 1st processor 90% of the time the access will be 2 ns while the rest of the time (10%) the access time will be 200 ns. The average is
 $(0.9 \times 2ns) + (0.1 \times 200ns) = 1.8ns + 20ns = 21.8ns$
 In the 2nd processor 95% of the time the access will be 1ns while the rest of the time (5%) the access time will be 500 ns. The average is
 $(0.95 \times 1ns) + (0.05 \times 500ns) = 0.95ns + 25ns = 25.95 ns$

2. לפניך חלק מתוך CACHE של מעבד כלשהו. בכל אחד מהסעיפים שלהלן צרף לתשובתך **הסבר מלא וחשוב מדויק**.

TAG	ID	BYTE IN BLOCK			
		11	10	01	00
10101000110010101011	10101010	0x00	0x61	0xC1	0x23
10111000110010101011	10101011	0x10	0x71	0xC2	0x45
10101110101000111110	10101100	0x20	0x81	0xC3	0x67
10100100001100101011	10101101	0x30	0x91	0xC4	0x56
00111110001100101011	10101110	0x40	0xA1	0xC3	0xF7
11011010001100101011	10101111	0x50	0xB1	0xC2	0xF6
00110100101000111011	10110000	0x60	0xC1	0xC1	0xEE
11101001000011000101	10110001	0x70	0xD1	0xC0	0xDE
10101110001100101011	10110010	0x84	0xE1	0x24	0x3A
10101110001010101011	10110011	0x94	0xF1	0x23	0x4B
10101110100011111011	10110100	0xA4	0x55	0x22	0x5C
10100100110010101011	10110101	0xB4	0x45	0x21	0x6D
00111000110010101011	10110110	0xC4	0x35	0x20	0x7E
11101000110010101011	10110111	0xD4	0x25	0x19	0x8F
00110010100011111011	10111000	0xE4	0x15	0x18	0x90
11101001001100100101	10111001	0xF4	0x05	0x17	0xA1

- א. (5 נק') מאיזה סוג ה-CACHE הנ"ל?
 מהו גודל מרחב הכתובות של המעבד?
- תשובה:** Directly mapped וישנן 30 ביטים בכתובת סה"כ $(2 + 8 + 20)$ לכן 2^{30} Bytes

ב. (5 נק') .

כמה בלוקים יש במרחב הכתובות?

מהו גדלו של ה-CACHE בבתים (Bytes) ?

תשובה: בכל בלוק ישנם 4 בתים, $2^8 = 256 \times 4 = 1024 \text{ Bytes}$

ג. (10 נק') נדרש לקרוא ממרחב הזיכרון את הבית (Byte) הנמצא בכתובת 0x01BA3ED1 האם

ניתן לדעת מה ערכו של הנתון בכתובת הנ"ל? ובמידה וכן אז מהו?

תשובה:

0x01BA3ED1=101011101000111110111011010001 ולכן:

Byte offset = 01

Index = 10110100

Tag=10101110100011111011

ולכן הכתובת מופיעה ב-cache והערך הנמצא בה הוא 0x22.

ד. (10 נק') בקוד שרץ על המעבד הנ"ל ישנה לולאה בת n חזרות ובתוכה רצף גישות למלים

הנמצאות בכתובות הבאות:

11110000001111111000 10110100 00

11110000001111111000 10110101 00

11110000001111111000 10110110 00

11110000001111111000 10110111 00

00001111000000000111 10110100 00

00001111000000000111 10110101 00

00001111000000000111 10110110 00

00001111000000000111 10110111 00

00111001100011111111 10110100 00

00111001100011111111 10110101 00

00111001100011111111 10110110 00

00111001110011111111 10110111 00

מה יהיה מספר ה-misses ?

ה. (10 נק') הוצע להפוך את ה-cache ל 2-way set associative וזאת מבלי לשנות את גדלו.

האם עבור הלולאה שבסעיף הקודם הדבר יועיל? ואם כן, מה יהיה מספר ה-misses וזאת

בהנחה שבתחילת הלולאה אף אחת מהכתובות הנ"ל לא הופיע ב-cache.

ו. (5 נק') כעת הוצע להפוך את ה-cache ל 4-way set associative וזאת מבלי לשנות את גדלו.

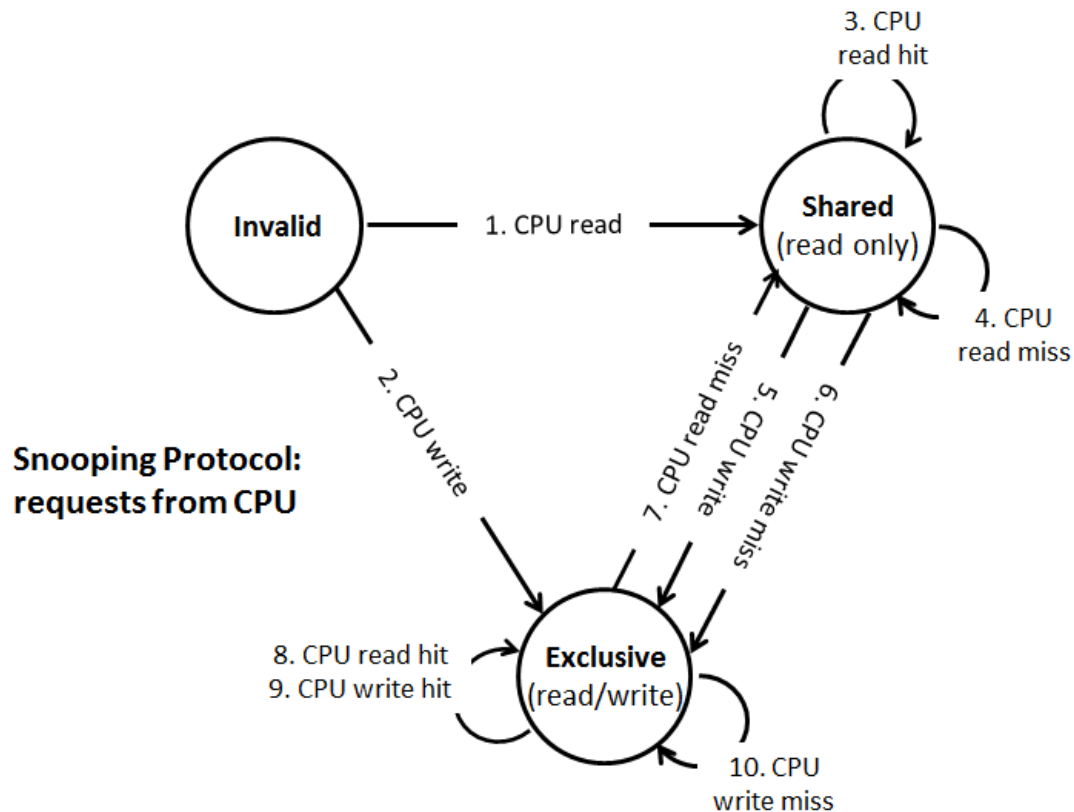
האם עבור הלולאה שבסעיף הקודם הדבר יועיל? ואם כן, מה יהיה מספר ה-misses וזאת

בהנחה שבתחילת הלולאה אף אחת מהכתובות הנ"ל לא הופיע ב-cache.

שאלה ב' (55 נקודות)

נתון מעבד בעל ארבע ליבות בארכיטקטורת 32 סיביות ו shared memory. לכל ליבה זכרון מטמון משלה עם bus משותף. פרוטוקול הקוהרנטיות של הזכרונות הינו write ,snooping protocol .invalidate. וזכרונות המטמון פועלים בשיטת writeback.

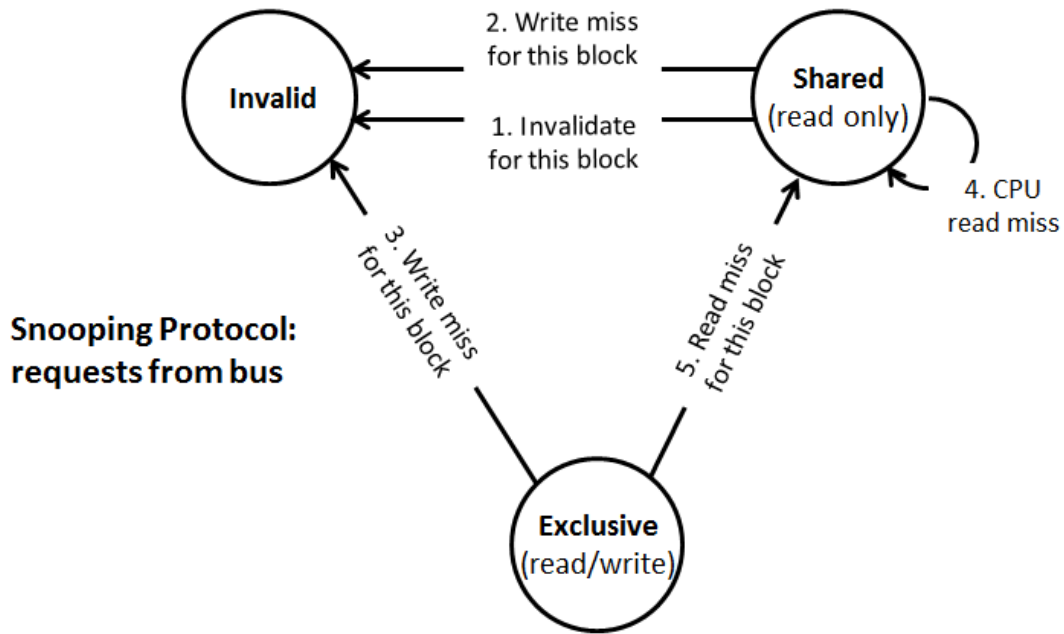
1. (15 נקודות) מכונת המצבים הנענית לבקשות של המעבד מתוארת להלן. לכל אחד מהמעברים רשום בטבלה בכך/לא האם נדרשת פעולה על ה bus. במדה וכן פרט במקום המתאים בטבלה מהי בדיוק הפעולה הנדרשת. במדה ולא נמק במקום המתאים בטבלה מדוע.



	Bus required?	Detailed operation
1		1. במדה זה block נמצא, מאחר והמצב הנוכחי invalid, קריאת נתונים איננה יכולה להיענות מה cache המקומי. 2. יתכן גם והבלוק איננו נמצא. יש להודיע read miss על ה bus. זה גורר פעולת cache רגילה של החלפת הבלוק. 1. הבאתו מה cache שנמצא במצב exclusive, או 2. הבאתו מהזיכרון הראשי. בכל מקרה מצבו של ה block משתנה ל shared.
2	כן	1. במדה זה block נמצא, מאחר והמצב הנוכחי invalid, כתיבת נתונים איננה יכולה להיעשות ב cache המקומי. 2. יתכן גם והבלוק איננו נמצא. יש להודיע write miss על ה bus. זה גורר פעולת cache רגילה

		של החלפת הבלוק. 1. הבאתו מה cache שנמצא במצב exclusive, או 2. הבאתו מהזיכרון הראשי. בכל מקרה מצבו של ה block משתנה ל exclusive.
3	לא	קרא מהזיכרון המקומי. מצב הזכרון נשאר shared.
4	כן	מאחר ומצב ה block הינו shared, פירוש הדבר שזהו miss רגיל כמו ב uniprocessor. צריך להודיע read miss על ה bus, להביא את הבלוק מהזיכרון הראשי, ולבצע replacement רגיל. מצב ה block עובר ל shared.
5	כן	מאחר והייתה כתיבה ל block (עקב hit), מצבו הופך ל exclusive, ויש להודיע על ה bus לכל הזכרונות האחרים המחזיקים באותו block (במדה וישנם כאלה) שעלים להפוך ל invalid.
6	כן	מאחר וה block במצב shared, פירוש הדבר שזהו miss רגיל. יש להודיע write miss על ה bus ולהביא את ה block המבוקש מהזיכרון. מאחר ומדובר בכתיבה, מצבו נקבע ל exclusive.
7	כן	מאחר ומצב ה block הינו exclusive, פירוש הדבר שזהו miss רגיל כמו ב uniprocessor. צריך להודיע write miss על ה bus, לבצע writeback לזיכרון הראשי (במדה וה block מוחלף) ולהביא את הבלוק מהזיכרון הראשי, ולבצע replacement רגיל. מצב ה block עובר ל shared.
8	לא	זהו read hit רגיל. הבקשה נענית מה block המקומי משום שהוא במצב exclusive. מצבו נשאר exclusive.
9	לא	זהו write hit רגיל. הכתיבה נעשית ל block המקומי משום שהוא במצב exclusive. מצבו נשאר exclusive.
10	כן	מאחר וה block במצב exclusive והתרחש write miss, יש להביא את ה block מהזיכרון. יש כמוכן לבצע לפניכן writeback לזיכרון הראשי (במדה וה block מוחלף). מצב ה block נקבע ל exclusive.

2. (10 נקודות) נתונה מכונת מצבים הנענית לבקשות של bus, בנתונים של סעיף 1. רשום בטבלה שלהלן לכל אחד מהמעברים כן/לא האם נדרשות פעולות הקשורות לזכרון הן למעבד הנוכחי והן לזכרון הראשי. במדה וכן, פרט במקום המתאים בטבלה מהי בדיוק הפעולה הנדרשת. במדה ולא, נמק במקום המתאים בטבלה מדוע.



בטבלה השתמשנו במושג cache ו block במשמעות דומה.

	Mem operation required?	Detailed operation
1	לא	זו פעולת coherence. ה block היה במצב shared. עקב write hit ב block אחר, מצב הבלוק הנוכחי הופך ל invalid.
2	לא	זו פעולת coherence. ל block הנוכחי נכתב ב cache אחר עובדה ששודרה על ה bus בהודעת ה write miss. מצבו על כן הופך ל invalid.
3	כן	זו פעולת coherence. על ה bus מופיע write miss שקרה ב cache אחר עבור ה block הנוכחי (במקום האחר ה block היה invalid). ה cache הנוכחי מספק את ה miss ע"י writeback והפניה לזיכרון הראשי שהופיעה על ה bus מבוטלת. מאחר והכתיבה נעשתה ב cache אחר, ה cache הנוכחי הופך ל invalid.
4	לא	על ה bus מופיע read miss שקרה ב cache אחר ל block שאיננו הנוכחי. ה miss טופל במקום האחר וזה לא נוגע ל block הנוכחי, כך שלא נדרשת שום פעולת זכרון וגם לא שינוי מצב.
5	כן	זו פעולת coherence. על ה bus מופיע read miss שקרה ב cache אחר עבור ה block הנוכחי (במקום האחר ה block היה invalid). ה cache הנוכחי מספק את ה miss ע"י writeback (כל ה caches האחרים מסונכרנים) והפניה לזיכרון הראשי שהופיע על ה bus מבוטלת. מצב כל ה caches הופך ל shared.

3. (5 נקודות) נתון מעבד בעל ארבע ליבות, ולכל ליבה זכרון מטמון משלה עם bus משותף. זכרון המטמון הינו 4KByte מסוג direct map, גודל בלוק 8 בתים. המערכת שומרת על קוהרנטיות הזכרון ע"י שימוש בפרוטוקול שבסעיפים א' וב'. בזמן מסויים מעבדים P1 ו P2 מכילים את הבלוק

- שכתובתו 0xAB7121C0 ואילו המעבדים P3 ו P4 מכילים את הבלוק שכתובתו 0xAC5121C0.
- מצב מכונות המצבים ככתוב בטבלה שלהלן כאשר לכל הכניסות אינדקס משותף.
- השלימו את ערכי ה tag בטבלה והסבירו את תשובתכם.
 - כמה סיביות בסה"כ נדרשות למימוש מכונות המצבים הנ"ל?

P1 cache		P2 cache		P3 cache		P4 cache	
state	tag	state	tag	state	tag	state	tag
shared	0xAB712	shared	0xAB712	shared	0xAC512	shared	0xAC512

הסבר: 8 בתים בבלוק דורשים 3 סיביות כתובת. לזיכרון 4KByte וזה גורר 512 בלוקים, כלומר 9 סיביות לאינדקס. זה מותיר $20 = 32 - (9 + 3)$ סיביות ל tag.

- לפניכם סדרה של פניות לזיכרון. יש לפרט את המתרחש במערכת. בכל אחד מהסעיפים נדרש:
- למלא את הטבלה המתארת את מצב הזכרונות בסיום כל פעולות הזכרון הנדרשות בהתאם ולנמק בשורות שמתחת לטבלה את התשובה.
 - לציין במפורש האם נדרש write-back ואם כן, על ידי איזה מעבד.

4. (5 נקודות) בהמשך פנה מעבד P3 וביקש לכתוב את המילה בכתובת 0xAB7121C4.

P1 cache		P2 cache		P3 cache		P4 cache	
state	tag	state	tag	state	tag	state	tag
invalid	0xAB712	invalid	0xAB712	exclusive	0xAB712	shared	0xAC512

הסבר: הבלוק הנדרש כתוצאה מהמלה בכתובת 0xAB7121C4 איננו ב P3, אבל פנייה על ה bus מוצאת אותה ב P1 ו P2. מאחר והבלוק שהיה ב P3 איננו "מלוכלך", לא נדרש write-back. הבלוק שהיה ב P3 מועף ואת מקומו תופס הבלוק שכתובתו 0xAB7121C0 וה tag מעודכן בהתאם. מאחר ו P3 ליכלך אותו, מצבו עובר ל exclusive, ואילו העותקים ב P1 ו P2 הופכים ל invalid.

5. (5 נקודות) בהמשך פנה מעבד P2 וביקש לכתוב את המילה ב כתובת 0xAB7121C8.

P1 cache		P2 cache		P3 cache		P4 cache	
state	tag	state	tag	state	tag	state	tag
invalid	0xAB712	exclusive	0xAB712	invalid	0xAB712	shared	0xAC512

הסבר: כתובת 0xAB7121C8 אמנם נמצאת ב P2 אבל הבלוק invalid. הבלוק המעודכן נמצא ב P3, שחייב כעת לבצע write-back ל bus בלבד (שימו לב שאין צורך לעדכן את הזכרון הראשי, מדוע?) ועובר למצב invalid. P2 מקבל את הבלוק המעודכן על ה bus, כותב לתוכו והופך לבעל הבית, ולכן מצבו עובר ל exclusive.

P1 cache		P2 cache		P3 cache		P4 cache	
state	tag	state	tag	state	tag	state	tag
invalid	0xAB712	exclusive	0xAB712	invalid	0xAB712	shared	0xAC512
הסבר: מאומה לא יקרה! מדובר בכלל באינדקס אחר.							

P1 cache		P2 cache		P3 cache		P4 cache	
state	tag	state	tag	state	tag	state	tag
invalid	0xAB712	exclusive	0xCC712	invalid	0xAB712	shared	0xAC512

הסבר: יש מצב של write miss ונדרש לפנות מקום בcache למידע החדש, ולכן מתבצע back של המידע לזיכרון הראשי כיוון שזה המידע המעודכן שלא נמצא במקום אחר. נשים לב שהמצב של הזיכרונות נשאר זהה כיוון שמידע החדש מופיע עדיין רק ב P2.

P1 cache		P2 cache		P3 cache		P4 cache	
state	tag	state	tag	state	tag	state	tag
invalid	0xAB712	shared	0xCC712	invalid	0xAB712	shared	0xCC712

הסבר: עכשיו המידע של הכתובת 0xCC7121C8 מופיע בשני מקומות ולכן לא ניתן לעשות בו שינוי ללא הודעה לbus על כך ולכן מעבירים גם את P2 למצב של קריאה בלבד .shared.