

BAR-ILAN UNIVERSITY (RA)

Faculty of Engineering

Ramat-Gan 52900, Israel



אוניברסיטת בר-אילן (ע"ר)

הפקולטה להנדסה

רמת-גן 52900

Tel: 03-5317722

engbi@mail.biu.ac.il

מבנה מחשבים ספרתיים

תש"ף סמסטר ב' מועד ג'

83-301

מרצה: פרופ' שמואל וימר

מתרגל: מר בנימין פרנקל

- הבחינה נערכת ב-ZOOM. חובה לפתוח מצלמות וידאו. אי פתיחת מצלמה תגרור פסילת הבחינה.
- פתרון הבחינה חייב להיות בפורמט PDF. יש להעלותו בתוך חלון הזמן שארכו כמשך הבחינה + 15 דקות. אי העלאת קובץ הפתרון בזמן תחשב כאי הגשה.
- בבחינה שתי שאלות ומשקלן שווה. סה"כ הניקוד 120, ציון מקסימלי 100 נקודות.
- יש להקפיד על כתב יד ברור וקריא.
- מותר שימוש בכל חומר עזר.
- משך הבחינה: שעותיים.
- בראש דף הפתרון יש להעתיק ולחתום על ההצהרה הבאה. ללא הצהרה וחתימה הבחינה לא תיבדק. אני מתחייב(ת) בזאת לשמור על טוהר הבחינה, לפתרה בכוחות עצמי בלבד ולא לעזור לשום גורם אחר שהוא. ידוע לי כי חשד כלשהו בטוהר הבחינה יאפשר לדרוש ממני להגן על פתרון הבחינה בעל-פה.

שם התלמיד(ה): _____ חתימה: _____

בהצלחה!

שאלה מספר 1 – Branch Prediction (60 נקודות)

בשאלה זו נעסוק במעבד המכיל חזאי קפיצות (branch predictor).

על המעבד מריצים את התוכנית באה:

```
ADDI R3, R8, 0
ADDI R1, $zero, N
Loop: LW R4, 0(R3)
      ADDI R5, R4, 0
      SRL R5, R5, 1
      SLL R5, R5, 1
      ADDI R3, R3, 4
      BNE R4, R5, Cond --br1
      ADDI R2, R2, 1
      SRL R5, R5, 2
      SLL R5, R5, 2
      SUBI R1, R1, 1
      BNE R4, R5, Cond --br2
      ADDI R6, R6, 1
Cond: BNE R1, $zero, Loop --br3
```

תזכורת:

- SLL - Shift Left Logical
- SRL - Shift Right Logical
- פקודות ה- shift אינן ציקליות

התוכנית עוברת על מערך של מספרים ובודקת חלוקה ב-2 וב-4 ללא שארית. נתון כי R8 מצביע לראש מערך המכיל מספרים בין 1 ל-N לפי הסדר, דהיינו: $[1, 2, 3, \dots, N]$, כאשר N הינו מספר טבעי המתחלק ב-4 ללא שארית. בכל סעיפי השאלה התוכנית מורצת אינסוף פעמים, ואנו נתעניין במצב המתמיד בלבד (ולא בשלבי הלימוד של חזאי הקפיצות).

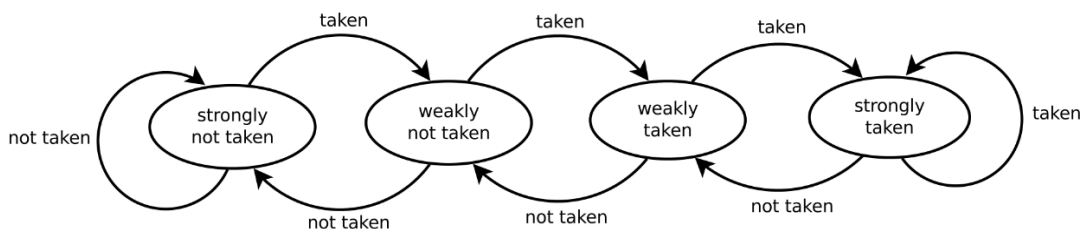
א. עליכם לכתוב את סדרת האירועים המתרחשים בפועל (לא החיזוי) במהלך ריצת התוכנית כתוצאה מפקודות הקפיצה המותנות (T/NT) עבור כל פקודת קפיצה בנפרד ($br1, br2, br3$). אם קיימת מחזוריות – ציינו אותה. התייחסו למצב המתמיד בלבד! (8 נקודות)

עבור $br1 - NT, T$ מחזורי.

עבור $br2 - NT, T$ מחזורי. יש לשים לב שאם $br1$ נלקח אז $br2$ לא ירוץ בכלל.

עבור $br3 - N-1$ פעמים T ולבסוף NT (מחזורי, כיוון שהתוכנית רצה אינסוף פעמים).

- נתון כי חזאי הקפיצות הקיים במעבד הינו מבוסס *2-bit-saturated counters* (כל *counter* הינו בעל מצבים: *SNT*, *WNT*, *WT*, *ST*). מכונת המצבים הבסיסית של כל-*counter* מתוארת בתרשים הבא:



החזאי משתמש ב- $k=3$ סיביות היסטוריה. ההיסטוריה הינה פרטית (לוקאלית), ומכונות המצבים פרטיות (לכל פקודת *branch* יש את החזאים שלה). עבור כל אירוע בהיסטוריה נסמן: $NT = 0$ ו- $T = 1$. נתון כי אוגרי ההיסטוריה מאותחלים באפסים וכי ערכי היסטוריה חדשים מוכנסים ל-LSB. שימו-לב כי המצב ההתחלתי של מכונות המצבים לא נתון. לכן, אם קיימת תלות במצב זה, יש לציין זאת, ולענות על השאלה כתלות במצב ההתחלתי.

ב. נריץ את התוכנית הנתונה על מעבד בעל החזאי הנתון.

i. מהם ערכי ההיסטוריה השונים של כל קפיצה בנפרד, ומהם מצבי המכונה והחיזויים בכל

ערך היסטוריה? התייחס גם להשפעתו של הערך N על התוצאה. (6 נקודות)

ii. מהו ה- *misprediction rate* לכל קפיצה בנפרד? מהו ה- *total misprediction rate*? (6 נקודות)

(נקודות)

br1		br2		br3	
History	SM	History	SM	History	SM
010	ST - T	010	ST - T	110	ST - T
101	SNT - NT	101	SNT - NT	101	ST - T
				011	ST - T
				111	If $N=4$: SNT - NT If $N>4$: ST (mostly) - T WT (once every iteration) - T

פקודות br1 ו- br2 לא טועות לעולם, ולכן: $MR_{br1} = MR_{br2} = 0$

לגבי br3: אם $N=4$ אזי br3 לא טועה לעולם, ולכן $MR_{br3} = 0$. אם $N>4$ אזי br3 טועה באחד

מתוך N חיזויים, ולכן: $MR_{br3} = \frac{1}{N}$.

סה"כ: פקודות br1 ו-br3 רצות פעם אחת בכל איטרציה, ואילו br2 רצה ב- 1/2 מהאיטרציות. ולכן, אם $N=4$ אז: $\text{total misprediction rate} = 0$.

$$\text{total misprediction rate} = \frac{\text{mispredictions}}{\text{branches}} = \frac{1}{N + \frac{N}{2} + N} = \frac{2}{5N} \quad \text{אם } N > 4$$

- מכאן ואילך נניח כי $N=8$.

ג. מהו הערך k (מספר סיביות ההיסטוריה) הנמוך ביותר שיאפשר

$$\text{total misprediction rate} = 0\% \quad ? \quad (8 \text{ נקודות})$$

בתוכנית זו רק br3 תושפע מ- k . על מנת לאפשר $\text{MR}_{\text{br3}} = 0\%$ עלינו להיות במצב בו עבור כל ערך היסטוריה החיזוי יהיה נכון. אם $k = N - 1 = 7$ אז יהיה ערך היסטוריה ייחודי לכל איטרציה של הלולאה (מה שאומר שיש מכונה ייעודית לכל שלב בלולאה), ולכן החיזוי יהיה מושלם. כל k קטן מזה גורם לכך שקיים שיתוף של אותה מכונת מצבים בין מצב שבו צריך לקפוץ ובין מצב שבו לא צריך לקפוץ (מכונת המצבים שמתאימה לערך היסטוריה '11111' תהיה משותפת, ולכן לא ייתכן חיזוי מושלם).

- התוכנית רצה על מעבד מצונר (pipelined) בעל 5 שלבים. נתון כי החיזוי מתבצע בשלב הראשון, וכי ה-branch resolution הינו בשלב השני (שלב ה-ID). הניחו כי אין עיכובים של מערכת הזיכרון, וכי כל תלויות המידע מטופלות ע"י מנגנוני קידום כך שאין אובדן מחזורי שעון בשל כך ($\text{CPI}_{\text{ideal}} = 1$).

ד. מהו החלק היחסי של פקודות הקפיצה המותנית בפילוג הדינאמי של פקודות התוכנית? עבור $k=3$, מהו ה-CPI האפקטיבי של המעבד עבור התוכנית הנתונה? כתבו ביטוי כללי תחילה, ולאחר מכן הציבו מספרים. (10 נקודות)

באיטרציות בהן R4 מכיל מספר אי-זוגי ישנן 7 פקודות, מתוכן שתי פקודות branch.
באיטרציות בהן R4 מכיל את המספר 2 או את המספר 6 ישנן 12 פקודות, מתוכן 3 פקודות branch.
באיטרציות בהן R4 מכיל את המספר 4 או את המספר 8 ישנן 13 פקודות, מתוכן 3 פקודות branch.

בנוסף, יש את פקודת האתחול של הרגיסטרים R3 ו-R1, המתבצעות פעם אחת ב- 8 איטרציות. סה"כ: 20 פקודות branch מתוך 80 פקודות ולכן: $\%br = 25\%$.
ה-MR כבר חושב בסעיף ב'. ה-BR בשלב השני בצינור, ולכן שגיאת חיזוי גוררת אובדן ביצועים של מחזור שעון אחד.

$$\text{CPI} = \text{CPI}_{\text{ideal}} + \%br \times \text{MispredictionRate} \times \text{MispredictionPenalty} = 1 + \frac{20}{80} \times \frac{2}{5 \times 8} \times 1 = 1.0125$$

ה. על-מנת לחסוך במשאבי מערכת, הוחלט להשאיר את ההיסטוריות פרטיות, אך להשתמש במכונות מצבים משותפות לכל פקודות ה-branch. עבור $k=3$, מהו עתה ה-misprediction rate לכל קפיצה בנפרד? מהו ה-total misprediction rate? נמק היטב את צעדיך. (12 נקודות)

נתבונן במצבי ההיסטוריה ונזהה מצבים משותפים ומצבים ייחודיים:
 מצבים '110', '011' ו-'111' הם מצבים ייחודיים ל-br3 ולכן מכונות אלו יתפקדו כפרטיות ל-br3. מכונות המצבים המתאימות להיסטוריה '110', '011' תמיד יתנו חיזוי מושלם (תמיד יחזו T), ואילו מכונות המצבים המתאימה להיסטוריה '111' תחזה תמיד T, אך זה יהיה נכון רק ב-4 פעמים מתוך 5 פעמים שניגשים למכונות מצבים זו במחזור שלם של ריצת התוכנית (8 איטרציות של הלולאה).

מצב '010' הינו משותף ל-br1 ול-br2, כאשר גם br1 וגם br2 תמיד יקפצו במצב הזה, ולכן החיזוי של מכונות המצבים המתאימה למצב זה תמיד יהיה נכון (תמיד החיזוי יהיה T).

מצב '101' הינו משותף ל-br1, br2 ול-br3, כאשר br3 תמיד תקפוץ במצב הזה, ואילו br1 ו-br2 תמיד לא יקפצו במצב הזה. בכל מחזור של ריצת התוכנית "נבקר" במצב זה 7 פעמים סה"כ, כאשר רק ביקור אחד יהיה ע"י פקודה br3, ולכן, המכונה הזאת תמיד תחזה NT (במצב היציב), ולכן, התרומה שלה ל-misprediction rate לא תלויה באתחול של מכונות המצבים. יוצא, שהחיזוי של מכונה זו יהיה תמיד נכון עבור br1 ו-br2, ויהיה תמיד שגוי עבור br3 (וזאת בנוסף לטעות שיש למכונות המצבים של ההיסטוריה '111' באיטרציה השמינית של הלולאה).

לסיכום:

$$MispredictionRate_1 = 0$$

$$MispredictionRate_2 = 0$$

$$MispredictionRate_3 = \frac{2}{8} = \frac{1}{4} = 25\%$$

$$Total\ MispredictionRate = \frac{\text{mispredictions}}{\text{branches}} = \frac{0+0+2}{8+4+8} = 10\%$$

1. בנוסף לחיסכון במשאבי המערכת המתואר בסעיף הקודם (שימוש במכונות מצבים משותפות לכל פקודות ה-branch), הוחלט לצמצם גם את מספר סיביות ההיסטוריה ולהסתפק ב-k=2 סיביות היסטוריה עבור כל פקודת branch. מהו עתה ה-misprediction rate לכל קפיצה בנפרד? מהו ה-total misprediction rate? נמק היטב את צעריך. (10 נקודות)

למעשה, עבור br1 ו-br2 אין כל שינוי מהותי. פקודות אלו ישתמשו במכונות המצבים המתאימה למצב '10' ולמצב '01', כאשר עבור שתי הפקודות החיזוי של המכונות הנ"ל יהיה מושלם.

מצב '11' הוא מצב ייחודי ל-br3, אשר תמיד יחזה T, והפעם החיזוי יהיה נכון ב-5 פעמים מתוך 6 הפעמים שמבקרים במצב זה במהלך מחזור אחד של ריצת התוכנית.

מצב '10' יהיה משותף גם ל-br3 (בנוסף ל-br1 ו-br2), כאשר בכל מחזור של ריצת התוכנית "נבקר" במצב זה 7 פעמים סה"כ. אמנם רק ביקור אחד יהיה ע"י פקודה br3, אך במקרה הזה החיזוי של המכונה (T) יתאים גם ל-br3, ולכן המכונה הזאת תמיד תיתן חיזוי נכון.

מצב '01' יהיה משותף גם הוא ל-br3 (בנוסף ל-br1 ו-br2), כאשר br3 תמיד תקפוץ במצב הזה, ואילו br1 ו-br2 תמיד לא יקפצו במצב הזה. בכל מחזור של ריצת התוכנית "נבקר" במצב זה 7 פעמים סה"כ, כאשר רק ביקור אחד יהיה ע"י פקודה br3, ולכן, המכונה הזאת תמיד תחזה NT (במצב היציב), ולכן, התרומה שלה ל-misprediction rate לא תלויה באתחול של מכונות

המצבים. יוצא, שהחיזוי של מכונה זו יהיה תמיד נכון עבור br1 ו-br2, ויהיה תמיד שגוי עבור br3 (וזאת בנוסף לטעות שיש למכונת המצבים של ההיסטוריה '11' באיטרציה השמינית של הלולאה).

ולכן, התשובה הסופית לסעיף זה זהה לתשובה הסופית של הסעיף הקודם.

שאלה מספר 2 – Memory (60 נקודות)

נתון מעבד בעל מרחב כתובות ווירטואלי של 40 סיביות, רוחב מילה bit – 64 וזיכרון פיזי 16GB, וכן נתון כי רזולוציית המיעון היא per-byte. במעבד קיים 2^W – way set associative cache בגודל 64KB. גודלו של בלוק הוא 16B וזיכרון המטמון הינו physically tagged. החלפת הבלוקים נעשית בשיטת LRU. נתון שגודל frame הינו 16KB.

א. מהו מספר ה-sets, מהו גודל ה-tag ומהו מספר הכולל של סיביות ה-tag בזיכרון המטמון? (6 נקודות)

ה- block offset הינו 4 סיביות, ומספר ה- blocks ב- cache הינו $2^{12} = \frac{64KB}{16}$. בנוסף, מאחר ומדובר ב- 2^W – way set מכל 2^W בלוקים, ולכן מספר ה- sets הינו 2^{12-W} . מכאן נסיק כי גודל ה-tag הוא: $(12 - W) - 4 - 34$. סיביות. בסה"כ בכל ה- cache ישנן: $2^{12} \times (18 + W)$ סיביות של tag.

ב. במהלך התכן של המעבד התברר שמסלול ההשהיה הקריטי עובר במערכת הזיכרון ואורכו 1.4nSec, כאשר המסלול מורכב מתרגום כתובת ווירטואלית לכתובת פיזית האורך 800ps, וגישה ל- cache האורכת 600ps. על מנת להאיץ את ביצועי המחשב הציע מהנדס צעיר בוגר בר-אילן לצנר את מערכת הזיכרון. האם הדבר ניתן לביצוע? אם כן, כיצד? בהזנחת זמני ההשהיה ברגיסטרים ובלוגיקה, ובהינתן כי המסלול הארוך ביותר מחוץ למערכת הזיכרון אורך 500ps, מהו התדר המרבי אליו ניתן להאיץ את פעולת המחשב? (6 נקודות)

ניתן לביצוע ע"י הפרדת התרגום מכתובת ווירטואלית לפיזית (TLB) והגישה ל cache לשני שלבים נפרדים בצינור. זמן ההשהיה לשם תרגום הכתובת (גישה ל TLB) הוא זמן ההשהיה המירבי, ועל כן התדר המרבי הינו: $1.25 \times 10^9 = 1.25GHz = \frac{1}{800 \times 10^{-12}}$.

ג. מהנדס בכיר בחברה אמר כי ניתן להגיע להאצה המושגת בסעיף ב' ללא צורך בהעמקת הצינור, וזאת ע"י גישה ל- TLB ולזיכרון המטמון באופן מקבילי. לשם כך, כדי שהגישה באופן מקבילי תוכל להתבצע, המהנדס אמר כי יש צורך לבצע שינוי בגודלו של ה- frame ו/או בגודלו של ה- cache, והמליץ לבצע את השינוי דווקא בגודל ה- frame, משום ששינוי בגודלו של ה- cache יפגע ב- hit-rate שלו. הסבר את דבריו של המהנדס הבכיר. האם הדבר משפיע מספר הכניסות של ה- TLB? האם ישנה דרך להימנע משינוי גודלו של ה- frame? נמקו היטב את תשובתכם. (6 נקודות)

הדבר אכן ניתן להשגה וזאת בתנאי שיהיו לנו מספיק סיביות מה- page-offset כדי לגשת את זיכרון המטמון. במקרה זה ה- page offset גדול שווה ל- cache index (ביחד עם ה- block-offset), ואין צורך בתרגום של TLB בכדי לגשת ב- cache לאינדקס המתאים. על מנת שהתנאי יתקיים (כנראה ש: $w < 2$) יש צורך או להגדיל את ה- frame או להקטין את ה- cache, אכן, הקטנת גודל ה- cache עשויה לפגוע באופן טבעי ב- hit-rate שלו (כי יש בו פחות מידע), ולכן בחר המהנדס הבכיר להגדיל את גודל ה- frame.

להצעתו של המהנדס אין כל השפעה על מספר הכניסות של ה-TLB, שמספרם נקבע משיקולי זמני גישה ושיקולי hit/miss rate של מערכת הזיכרון הווירטואלי.

ישנה דרך להימנע משינוי גודלו של ה-frame, וזאת ע"י הגדלת האסוציאטיביות (דרושה דרגת אסוציאטיביות של לפחות 4-way).

ד. יהי $\text{frame size} = 2^F B$. בהנחה שהצעתו של המהנדס הבכיר מסעיף ג' אכן התבצעה, האם הדבר משפיע על מספר סיביות ה-tag הכולל בזיכרון המטמון? נמקו היטב את תשובתכם, ואם התשובה חיובית רשמו ביטוי מפורש למספרם במערכת הזיכרון הנתונה בסעיף א'. (5 נקודות)

אם נקבל את הצעתו של המהנדס מסעיף ג' ונגדיל את גודל ה-frame, לא תהיה לכך שום השפעה על מספר סיביות ה-tag בזיכרון המטמון. (אמנם, אם נגדיל את מידת האסוציאטיביות של זיכרון המטמון תהיה לכך השפעה על מספר סיביות ה-tag שהוא: $2^{12} \times (18 + W)$ סיביות של tag בזיכרון המטמון כולו).

ה. מהנדס שלישי בוגר אונ' בן-גוריון הציע לוותר לגמרי על תרגום כתובות ווירטואליות לפיזיות לשם חיפוש בזיכרון המטמון. האם הדבר אפשרי? אם כן, מהן הסכנות בויתור על התרגום? האם ניתן ע"י ויתור על התרגום להגדיל את תדר השעון מעבר להצעה בסעיף ב'? נמקו היטב את תשובתכם. (6 נקודות)

הדבר אכן ניתן להשגה וזאת ע"י הפיכת ה-cache ל-virtually indexed virtually tagged, שבו משתמשים גם בגישה ל-cache וגם ב-tag בכתובת ווירטואלית.

הסכנות שבשימוש ב-virtual tag הינן בעיות של aliasing ושל homonyms.

על פי נתוני הזמנים בסעיף ב', הרי שעל ידי ויתור על הגישה ל-TLB ניתן לעבוד בתדר גבוה יותר, שכן זמן ההשהיה הארוך ביותר כעת הוא זמן הגישה ל-cache, ועל כן התדר המרבי כעת הינו: $1/600 \times 10^{-12} = 1.667 \times 10^9 = 1.667 \text{GHz}$.

ו. האם ההצעה בסעיף ה' משפיעה על גודל ה-tag השמור בזיכרון המטמון? אם כן, כמה סיביות ידרשו בסה"כ למימוש החדש? בהנחת אסוציאטיביות 2^W -way ובהתעלם מהסיביות הנדרשות לניהול ה-cache (dirty וכו') מהו היחס בגדלו הכולל של ה-cache בסעיף זה לזה שבסעיף א'? האם הגדלת האסוציאטיביות מגדילה, איננה משנה, או מקטינה את היחס הזה? (6 נקודות)

מאחר והכתובת הווירטואלית הינה בת 40 סיביות ואלו הפיזית בת 34 סיביות, ה-tag של כל בלוק יגדל מ- $40 - 4 - (12 - W)$ ל- $34 - 4 - (12 - W)$. מאחר ומספר הבלוקים איננו משתנה, וכל block entry מכיל הן data והן tag, היחס בגודל הוא:

$$\frac{40 - 4 - (12 - W) + 128}{34 - 4 - (12 - W) + 128} = \frac{152 + W}{146 + W}$$

הגדלות האסוציאטיביות תקטין את היחס הנ"ל, בהתאם לביטוי.

ז. עבור נתוני סעיף א' מהו גודלו של page table? כמה כניסות ישנן בו, ומה גודלו הכולל ב-Bytes? הנח שהכתובת הפיזית יחד עם הסיביות הנדרשות לניהול הדפים מאוחסנות ב-64 סיביות. מהו המספר המקסימלי של תהליכים היכולים להתקיים בו זמנית בזיכרון הפיזי? (6 נקודות)

מאחר ועובדים במרחב כתובות ווירטואלי, $total\ pages = \frac{2^{40}}{16KB} = \frac{2^{40}}{2^{14}} = 2^{26}$, כלומר, למערכת הזיכרון הווירטואלי ישנם 64M דפים, ולכן זהו מספר הכניסות הנדרש ב-page table. כל כניסה מייצגת כתובת של frame שיחד עם סיביות ניהול הדף תופסות 64 סיביות, או 8 בתים. על-כן, הגודל הכולל של טבלת דפים הוא: $512MB = 2^{26+3}Byte$.

ה-page tables מאוכסנים בזיכרון הפיזי שגודלו $16GB = 2^{34}Byte$. על כן לכל היותר יכולים להתנהל בו זמנית בזיכרון $\frac{2^{34}}{2^{29}} = 32$ תהליכים. כמובן שזהו חסם תיאורטי, משום שזה לא מותיר בזיכרון שום מקום ל-data ול-instructions.

ח. על מנת שמערכת ההפעלה תנצל טוב יותר את הזיכרון ותוכל למקם את ה-page tables באופן יעיל יותר, הוחלט לפצל את הכתובת הלוגית לשלושה שדות כדלהלן: $d = 14$, $p_1 = 13$ ו- $p_2 = 13$, כאשר p_1 משמש את ה-outer table ו- p_2 את ה-inner tables. לאחר שינוי זה, מהו שטח הזיכרון המינימלי שיש להקצות על מנת להיות מסוגלים להריץ 100 תהליכים שונים בו זמנית? (6 נקודות)

על מנת שיהיה ניתן לתרגם כתובות, כל תהליך נדרש למינימום של טבלה אחת מתוך ה-inner tables, וכמובן שיש צורך להחזיק בזיכרון את ה-outer table של כל תהליך. גודלה של כל טבלה כזאת הוא: $2^{13+3}Byte$, ולכן נפח הזיכרון המינימלי הנדרש עבור ה-page tables של כל תהליך הינו $2 \times 2^{16}Byte$. לכן, כדי שיהיה ניתן להריץ 100 תהליכים שונים בו זמנית יש צורך להקצות בזיכרון מינימום של $100 \times 2^{17}Byte$.

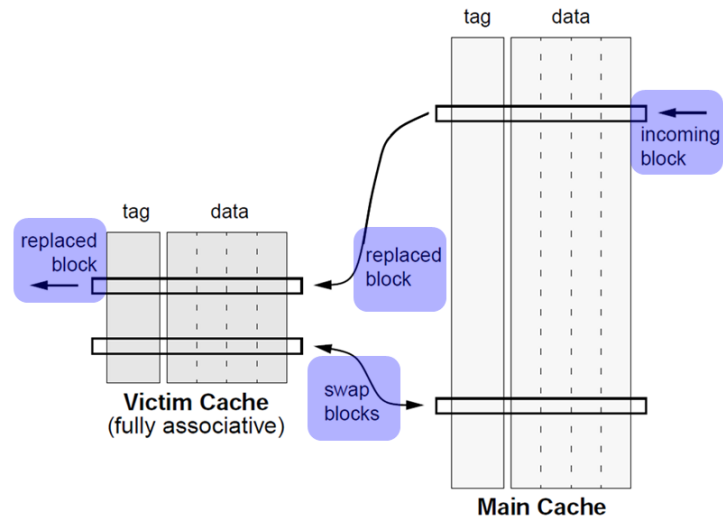
ט. ב-cache שבסעיף א' נקבע $W = 2$. בתוכנית שרצה במחשב הופיע רצף הגישות לכתובות הבאות (hexadecimal) שחזר על עצמו ברציפות 1001 פעמים:

0x2F012C800, 0x1EF01C808, 0x0DEF0F800, 0x2CDEFF808, 0x1BCDE8800, 0x0ABCD8800, 0x112344808, 0x001234800.

בהתעלם מהאיטרציה הראשונה, האם הרצף הנ"ל גורם להחטאות? אם כן, לכמה? (7 נקודות)

כאמור בסעיף א', ה-block offset הינו 4 סיביות. מאחר ו- $W = 2$, ה-cache הינו 4-way set associative, כלומר בכל set ישנם 4 בלוקים. אם כך, כמות ה-sets הינה: $\frac{64KB}{16B \times 4} = 2^{10}$, ולכן גודל האינדקס לצורך גישה ל-set הינו 10 סיביות. ב-6 מתוך 8 הכתובות הנ"ל כל 10 הסיביות של שדה האינדקס זהות, ולכן כתובות הבלוקים המתאימים ממופות לאותו set. בתום האיטרציה הראשונה ארבעת הבלוקים המכילים את הכתובות: 0x1EF01C808, 0x2F012C800, 0x001234800, 0x112344808 יהיו מצויים ב-cache (ב-set מספר 128), ומשם ואילך כל פניה ל-set זה גורמת ל-miss. לעומת זאת, שתי הכתובות 0x2CDEFF808, 0x0DEF0F800 ימופו לסט אחר (ל-set מספר 896), ולכן בגישות לכתובות אלו לא תהיה אף החטאה. לכן, בסה"כ ישנם $6000 = 1000 \times 6$ החטאות.

י. מהנדסת צעירה בוגרת בר-אילן נתבקשה לפתור את הבעיה והציעה להצמיד ל-cache שבסעיף א' cache קטן נוסף מסוג fully associative, כך ששני הזיכרונות יהיו אקסקלוסיביים, כלומר שמילה תימצא רק באחד משניהם, ואם תימצא בקטן, אזי תתבצע החלפה של מילה בין הקטן לגדול, כך שהמילה המבוקשת תעבור מהקטן לגדול, ובמקומה תעבור מילה מסוימת מהגדול לקטן (על-פי מדיניות LRU). המערכת פועלת ע"פ המתואר בציור שלהלן:



עבור רצף הגישות מהסעיף הקודם, מלאו את מספר ה- misses בטבלה הבאה. הסבירו במדויק את הערכים שמילאתם בטבלה. (6 נקודות)

Victim cache size [Blocks]	1	2	4	8	16
misses	6000	0	0	0	0

הוספת victim cache בגודל של בלוק אחד איננה תפתור את בעיית ההחטאות המתוארת בסעיף הקודם. הוספה של victim cache בגודל של 2 בלוקים ומעלה – תאפס את מספר ה- misses עבור רצף הגישות הנ"ל, משום שזיכרון המטמון יחזיק כל הזמן את 8 הבלוקים הנתונים ברצף הגישות. ה- victim cache ישרת את set מספר 128, ובכך ימנע את ההחטאות. סט מספר 896 איננו זקוק לעזרת ה- victim cache.