



ת"ב 2 – זיכרון מטמון Cache

קצר ולענין

בתרגיל זה תמשו סימולטור זיכרון מטמון (Cache) משלכם, בדומה לנלמד בכיתה. תצורת זכרון המטמון תהיה גמישה ותוגדר בתחילת הריצה באמצעות פרמטרים. בנוסף, הסימולטור שלכם יקרא קובץ קלט שבו יהיו מפורשות הגישות לזכרון. הסימולטור שלכם יצריך לחשב את ה-Hit/Miss rate או זמן הגישה המוצע לזכרון עבור התצורה שנקבעה עבור ה-trace (קובץ הקלט).

מאפייני סימולטור ה-cache

- הסימולטור ידמה גישות למידע (Data) בלבד
- הסימולטור יכול שתי דרגות (L1 ו-L2)
- שני ה-cacheים יעבדו במדיניות Write Back ומדיניות Write Allocate או No Write Allocate
- זכרון המטמון יעבד על פי עקרון ההכללה (Inclusive)
- מדיניות הפיניinya LRU
- בתחלת הריצה זכרון המטמון ריק
- כל הגישות הן בגודל 4 בתים ומיושרות לפי גבול של ארבעה בתים (שתי הסיבות התוחנות של הכתובת תהינה תמיד 00)

המאפיינים שניתן לקבוע באמצעות פרמטרים לאיתחול:

- גודל זכרון המטמון (ברצולציה של בתים)
- גודל הבלוק (ברצולציה של בתים)
- רמת האסוציאטיביות
- זמן הגישה (במחזורי שעון)
- מדיניות Write Allocate או No Write Allocate

מבנה קובץ הקלט

השורות של קובץ הקלט מכילות עקבות (trace) של גישות לזכרון מריצת תוכנית כלשהי, כאשר כל שורה תכיל תיאור של גישה שכזאת במבנה של 2 שדות עם רווח ביניהם:

1. קריאה (r) או כתיבה (w)
2. הכתובתmana קוראים או כתבים (b-hexa)

לדוגמא:

```
w 0x1ff91ca8
r 0x20000cdc
```

כמו כן, מסופקת לכם דוגמא לקובץ קלט עם חומר התרגיל.

או מה תכליס צריכים לעשות?

התכנית שלם תקרא cacheSim והריצה שלה תהיה כזו:

```
./cacheSim <input file> --mem-cyc <# of cycles> --bsize <block log2(size)>
--wr-alloc <0: No Write Allocate; 1: Write Allocate>
--l1-size <log2(size)> --l1-assoc <log2(# of ways)> --l1-cyc <# of cycles>
--l2-size <log2(size)> --l2-assoc <log2(# of ways)> --l2-cyc <# of cycles>
```

כאשר גודל זכרון המטמון, גודל הבלוק ורמת האסוציאטיביות הינם בחזקות של 2 (מספר cycles אינו חזקה של 2). כל המספרים הינם שלמים. גודל זכרון המטמון מתיחס לחלק ה-data ולא מתייחס לגודל ה-tag directory (במילים אחרות, גודל ה-cache הנთון בארגומנטים מתיחס לכמות המידע שנייה לשמור ב-cache).



למשל, אם נרצה לדמות זמן גישה של 100 cycles לזיכרון הראשי, גודל בלוק של L1 cache, 32B ו- 8 ways עם זמן גישה של cycle בודד ובנוסף L2 cache בגודל 1M ו- 16 ways עם זמן גישה של 5 cycles, וכן שה-example.in trace שמו example.in, ניתן להשתמש בפקודה הבאה:

```
./cacheSim example.in --mem-cyc 100 --bsize 5 --wr-alloc 1  
--l1-size 16 --l1-assoc 3 --l1-cyc 1 --l2-size 20 --l2-assoc 4 --l2-cyc 5
```

זמן הגישה לרמות השונות אינם כוללים את הזמן הגישה לרמות הקודומות. לדוגמה, זמן הגישה במקרה של פספסות ב-L1 ו开会ה ב-L2 הוא:

$$t_{access} = t_{L1} + t_{L2}$$

במקרה של פספסות ב-L1 וב-L2, זמן הגישה יהיה:

$$t_{access} = t_{L1} + t_{L2} + t_{mem}$$

אין צורך להתחשב בתקורות של writeback (שקרוות בReLU בדרכ'כ, וכך בסופו של דבר לא משפיעות ישירות על זמן הגישה). לעומת זאת, לצורך התרגיל זה זמן הגישה לא משתנה אם שורה הינה dirty או לא.

גישה מ-L1 ל-L2 כתוצאה מ-writeback אינה תשפייע על חישוב miss rate של L2.

הפלט של התכנית יהיה:

```
L1miss=<L1 miss rate> L2miss=<L2 miss rate> AccTimeAvg=<avg. acc. time>  
השדות <L1/L2 miss rate> יהיו שברים עשרוניים בין 0 ל-1 (כלומר, לא באחוזים) בבדיקה של 3 ספרות לאחר  
הנקודה בבדיקה. השדה <avg. acc. time> יהיה ממוצע זמן גישה על פני כל הגישות במחזור שעון בבדיקה של 3  
ספרות אחרות הנקודת בבדיקה. יש לעגל את המספרים לרמת הבדיקה המבוקשת על פי כללי עיגול רגילים (לקרוב ביותר).  
יש להוסיף סימן שורה חדשה (\n) בסוף השורה הנ"ל (ראו קובץ פלט לדוגמא המצורף לתרגיל).
```

שים לב !!!

- בדיקת התכניות שלכם תבוצע באופן אוטומטי. באחריותכם לספק פלט זהה זהה שהוגדר.
- עליהם לכתוב את התכנית ב-C או ב-C++ בלבד ולספק קובץ makefile בשbill הקומpileציה.



דרישות ההגשה

הגשה אלקטרונית בלבד באתר הקורס ("מודל") מחשבונו של אחד הסטודנטים.

מועד הגשה: עד ה- 27.12.2025 בשעה 23:55

עליכם להגיש קובץ tar.gz בשם tar.gz_ID1-ID2.tar.gz כאשר ID1-ID2 הם מספרי ת.ז. של המציגים. לדוגמה:
.hw2_012345678_987654321.tar.gz

ה-tar יכול:

- קוד המקור של סימולטור cache שלכם.
- .הנכם רשאים להגיש עד שני קבצים, ביניהם קובץ cacheSim.cpp, שניית להויף לו גם כן קוד.
- קוד המקור חייב להכיל תיעוד פנימי במידה סבירה על מנת להבינו.
- בנוסף, עליכם להגיש את ה-makefile לצורך הקומpileציה. שימוש לב Ci התכנית (קובץ הריצה) שת יצורו תקרא cacheSim. הנכם רשאים לשנות את ה-makefile כל עוד שם קובץ הריצה לא ישנה.

dagshim להגשה:

1. השימוש שלכם חייב להתקמפל בהצלחה ולרוץ במכונה הווירטואלית שמסופקת לכם באתר הקורס. זהה סכיבת הבדיקה המחייבת לתרגום הבית. יש לוודא בניה מוצלחת במכונה הווירטואלית. **קוד שלא יתקמפל יגרור ציון 0!**
2. מניסיונים של סטודנטים אחרים: הקפידו לוודא שהקובץ שהעלתם ל"מודל" הוא אכן הגresa שהתקוונתם להגיש. לא יתקבלו הגשות נוספות לאחר מועד הגשה שנקבע בעוננות כמו "משום מה הקובץ במודל לא עדכני ויש לנו גרסה עדכנית יותר שלא נקלטה".
3. יש להעלות את קובץ ההגשה דרך החשבון של אחד השותפים בלבד.

העתיקות יטופלו בחומרה