



## ת"ב 4 – זיכרון מטמון Cache

### קצר ולעניין

בתרגיל זה תממשו סימולטור Cache משלכם, בדומה לנלמד בכיתה. תצורת ה-cache תהיה גמישה ותוגדר בתחילת הריצה באמצעות פרמטרים. בנוסף, הסימולטור שלכם יקרא קובץ קלט שבו יהיו מפורטות הגישות לזיכרון. הסימולטור שלכם יצטרך לחשב את ה-Hit/Miss rate ואת זמן הגישה הממוצע לזיכרון עבור התצורה שנקבעה ועבור ה-cache (קובץ הקלט).

### מאפייני סימולטור ה-cache

- הסימולטור ידמה גישות data בלבד
- הסימולטור יכיל שתי דרגות (L1 ו-L2)
- שני ה-cache יעבדו במדיניות Write Back & Write Allocate
- ה-cache יעבוד על פי עקרון ההכלה (inclusive)
- מדיניות הפינוי הינה LRU
- בתחילת הריצה ה-cache יהיה ריק
- כל הגישות הן בגודל 4 בתים ומיושרות לפי גבול של ארבעה בתים (שתי הסיביות התחתונות של הכתובת תהיינה תמיד 00)

המאפיינים שניתן לקבוע באמצעות פרמטרים לאיתחול:

- גודל ה-cache (ברזולוציה של בתים)
- גודל הבלוק (ברזולוציה של בתים)
- רמת האסוציאטיביות
- זמני הגישה (במחזורי שעון)

### מבנה קובץ הקלט

השורות של קובץ הקלט מכילות עקבות (trace) של גישות לזיכרון 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. הכתובת ממנה קוראים או כותבים (ב-hexa)

לדוגמא:

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

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

### אז מה תכל'ס צריכים לעשות?

התכנית שלכם תקרא cacheSim וההרצה שלה תהיה כזו:

```
./cacheSim <input file> --mem-cyc <# of cycles> --bsize <block log2(size)>
--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 ה-cache, גודל הבלוק ורמת האסוציאטיביות הינם בחזקות של 2 (מספר cycles אינו חזקה של 2). כל המספרים הינם שלמים.

למשל, אם נרצה לדמות זמן גישה של 100 cycles לזיכרון הראשי, גודל בלוק של 32B, L1 cache בגודל 64KB ו-8 ways עם זמן גישה של cycle בודד ובנוסף L2 cache בגודל 1M ו-16 ways עם זמן גישה של 5 cycles. עבור trace ששמו example.in, נשתמש בפקודה הבאה:



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

זמני הגישה עבור כל רמה כוללים את הגישות לרמות הקודמות וכן את כל תקורת ההחלפות, אם נדרשת. כלומר, זמני הגישה למקרים השונים יהיו:

Hit in L1 → L1 cycles

Miss in L1 + Hit in L2 → L2 cycles

Miss in L1 + L2 → Main memory cycles

למשל, עבור גישה לכתובת הראשונה ב-trace זמן הגישה יהיה בדיוק זמן הגישה לזיכרון הראשי (מכיוון שה-cache ריק בהתחלה).

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) בדיוק של 3 ספרות אחרי הנקודה בדיוק. השדה <avg. acc. time> יהיה ממוצע זמן גישה על פני כל הגישות במחזורי שעון בדיוק של 3 ספרות אחרי הנקודה בדיוק. יש לעגל את המספרים לרמת הדיוק המבוקשת על פי כללי עיגול רגילים (לקרוב ביותר).

יש להוסיף סימן שורה חדשה (\n) בסוף השורה הנ"ל (ראו קובץ פלט לדוגמא המצורף לתרגיל).

### שימו לב !!!

- בדיקת התכניות שלכם תתבצע באופן אוטומטי. באחריותכם לספק פלט זהה לזה שהוגדר.
- עליכם לכתוב את התכנית ב-C או ב-C++ בלבד ולספק קובץ makefile בשביל הקומפילציה.

### דרישות ההגשה

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

### מועד ההגשה: עד יום ה' 16.06.2016 בשעה 23:55.

עליכם להגיש קובץ ZIP בשם hw4\_ID1\_ID2.zip כאשר ID1 ו-ID2 הם מספרי ת.ז. של המגישים. לדוגמה: hw4\_012345678\_987654321.zip. ה-ZIP יכיל את הקבצים הבאים:

- קוד המקור של סימולטור ה-cache שלכם
- קוד המקור חייב להכיל תיעוד פנימי במידה סבירה על מנת להבינו
- makefile בשביל הקומפילציה. שימו לב כי התכנית (קובץ הריצה) שתייצרו תקרא cacheSim

דגשים להגשה:

1. המימוש שלכם חייב להתקמפל בהצלחה ולרוץ בסביבת השרת הטכניוני T2 (או TX). יש להשתמש בקומפיילר G++/GCC ברירת המחדל בשרת.
2. שימו-לב לסנקציות במקרה איחור כמפורט באתר הקורס. מאוד לא מומלץ לאחר את מועד ההגשה שהוגדר לתרגיל. בנסיבות מיוחדות יש לקבל אישור **מראש** מהמתרגל האחראי לאיחור.
3. מניסיונם של סטודנטים אחרים: הקפידו לוודא שהקובץ שהעלתם ל"מודל" הוא אכן הגרסה שהתכוונתם להגיש. לא יתקבלו הגשות נוספות לאחר מועד ההגשה שנקבע בטענות כמו "משום מה הקובץ במודל לא עדכני ויש לנו גרסה עדכנית יותר שלא נקלטה".

**בהצלחה!**