

פרויקט סימולטור מטמון

בפרויקט זה נממש סימולטור של מטמון נתונים. הפרויקט יתבצע בזוגות, להגשה עד 22.1.2017.

- הסימולטור תומך בשתי קונפיגורציות שונות של המעבד: בקונפיגורציית הראשונה קיים מטמון L1 בלבד, המחובר ישירות לזיכרון הראשי, ובקונפיגורציית השנייה יש שתי דרגות מטמון: L1 ו-L2.
- המעבד הינו מעבד 32 סיביות עם address space של 24 סיביות. לא נממש סימולטור של המעבד אלא נעבוד עם קבצי trace שיכילו שורת טקסט עבור כל גישה של המעבד. המעבד מחובר למטמון L1 עם bus ברוחב 32 סיביות עבור ה-data, ו-24 סיביות עבור ה-address. כתובות המעבד הינם בבתים.
- בקונפיגורציית המכילה את L2, ה-bus בין L1 לבין L2 הינו ברוחב 256 סיביות.
- הזיכרון הראשי בגודל 16MB, ורוחב ה-bus אליו הינו 64 סיביות (מתחבר ל-L2 בקונפיגורציית של שתי דרגות מטמון, אחרת ישירות אל L1). הניחו שזמן הגישה לזיכרון הראשי הינו 100 מחזורי שעון עבור המעבד הראשון על ה-bus, ומחזורי שעון נוסף עבור כל מעבר על ה-bus.
- שני המטמונים הינם מסוג write-back ו-write-allocate. מדיניות ההחלפה במטמון L2 הינה LRU.
- מטמון L1 הינו direct-mapped בגודל 4KB, וגודל הבלוק ב-L1 הינו חזקה שלמה של 2, אשר ניתן לקינפוג בין 4 לבין 128 בתים. זמן ה-hit time ב-L1 הינו מחזורי שעון בודד.
- מטמון L2 הינו two-way-set-associative בגודל 32KB בתים, וגודל הבלוק ב-L2 הינו גם כן חזקה שלמה של 2, אשר ניתן לקינפוג בין 4 לבין 1024 בתים (גודל מגודל הבלוק של L1). זמן ה-hit time ב-L2 הינו 4 מחזורי שעון.
- בתחילת העבודה המטמונים ריקים ותוכן הדאטא בתוכם מאותחל לאפס. הזיכרון מאותחל לתוכן הקובץ meminit.txt.

1 הרצה וקבצים:

הפרויקט יבנה אל תוך command line application שנקרא sim, ויורץ עם רשימת פרמטרים:

```
sim levels b1 b2 trace.txt meminit.txt memout.txt l1.txt l2way0.txt l2way1.txt stats.txt
```

הפרמטר levels הינו 1 עבור סימולציית של L1 בלבד, או 2 עבור סימולציית של L1 ו-L2.

הפרמטרים b1 ו-b2 הינם מספרים דצימליים (בסיס 10), המתארים את גודל הבלוק במטמונים L1 ו-L2 בהתאמה, בבתים (בקונפיגורציית של L1 בלבד ערכו של b2 אינו משפיע).

הקובץ trace.txt הינו קובץ טקסט המתאר את רצף הבקשות של המעבד לקריאות וכתובות מהמטמון, כאשר כל שורה מתארת בקשת קריאה או כתיבה למטמון בפורמט תואם לדוגמא הבאה:

```
7 L CA2C34
3 S BC12A8 127FEABD
```

- השדה הראשון מציין את מספר ההוראות בקוד שאינם הוראות LOAD או STORE, שהמעבד ביצע בין הגישה הקודמת לזיכרון לבין הגישה שתהיה עקב השורה הנוכחית. הניחו כי כל הוראה בלתי תלויה כזו מתבצעת במחזורי שעון בודד.

- השדה השני הינו L עבור הוראת LOAD, ו-S עבור הוראת STORE.

- השדה השלישי כולל 6 ספרות הקסאדצימליות (בבסיס 16) המתאר כתובת בת 24 סיביות במרחב הזיכרון.

- השדה הרביעי מופיע בהוראות STORE בלבד, וכולל 8 ספרות הקסאדצימליות שהן ה־ DATA שאותו כותבים.

ניתן להניח שכל גישות המעבד הינן לכתובות aligned ל־ 32 ביטים, כלומר כל הכתובות בקבצי ה־ trace מתחלקות ב־ 4. ה־ data נכתב לזיכרון בפורמט little endian, למשל עבור הדאטא 127FEABD, הבית BD יכתב לכתובת הנמוכה מתוך ה־ 4, והבית 12 לכתובת הגבוהה.

הקובץ memin.txt הינו קובץ טקסט המכיל את תוכן הזיכרון הראשי לפני ריצת התוכנית. כל שורה מכילה 2 ספרות הקסאדצימליות ומתארת תוכן בית בודד בזיכרון. הקובץ נטען אל הזיכרון החל מכתובת 0. במידה ומספר השורות בקובץ קטן מגודל הזיכרון, תוכן שאר הזיכרון הראשי (שלא אותחל ע"י הקובץ) יאותחל לאפס.

בסיום ריצת הסימולטור יש ליצור את הקבצים הבאים:

הקובץ memout.txt הינו קובץ טקסט בפורמט זהה ל־ memin.txt, שמכיל את תוכן הזיכרון הראשי בסיום ריצת התוכנית.

הקובץ l1.txt הינו קובץ טקסט בפורמט זהה ל־ memin.txt, המכיל את תוכן מטמון L1.

הקובץ l2way0.txt הינו קובץ טקסט בפורמט זהה ל־ memin.txt, המכיל את תוכן way 0 של מטמון L2. יש לכתוב לקובץ רק במידה ומסמלצים שני מטמונים.

הקובץ l2way1.txt הינו קובץ טקסט בפורמט זהה ל־ memin.txt, המכיל את תוכן way 1 של מטמון L2. יש לכתוב לקובץ רק במידה ומסמלצים שני מטמונים.

הקובץ stats.txt מכיל הינו קובץ טקסט המכיל סטטיסטיקות לגבי ריצת התוכנית. כל שורה מכילה מספר בודד, בסדר הבא:

```
program running time in cycles
number of read hits in L1
number of write hits in L1
number of read misses in L1
number of write misses in L1
number of read hits in L2
number or write hits in L2
number of read misses in L2
number of write misses in L2
L1 local miss rate
global miss rate
AMAT
```

השורות הראשונות מכילות מספרים שלמים דצימלים (בבסיס 10). שלושת השורות האחרונות מכילים מספרים עשרוניים עם לפחות 4 ספרות לאחר הנקודה העשרונית. בשורות המתאימות אל L2 יש לכתוב 0 במידה ומסמלצים קונפיגורציה עם L1 בלבד.

2 דוקומנטציה:

הקפידו שהקוד יהיה קריא, ומכיל comments לגבי מבני הנתונים והפונקציות. כמו כן יש להגיש דוקומנטציה חיצונית המתארת באופן כללי את הפרויקט.

3 בדיקות:

הפרויקט שלכם יבדק בן השאר ע"י קבצי קלט שלא תקבלו מראש. לכן חשוב מאוד לבדוק נכונות ע"י בנייה של קבצי קלט שונים, וכמו כן בדיקה עם פרמטרים שונים עבור גדלי הבלוקים.

4 סביבות תכנות והוראות הגשה:

ניתן לממש את הפרויקט בשפת C, או C++, או Java. ניתן לממש על סביבת Windows או Linux. אם משתמשים ב־ Visual Studio, יש להגיש את כל ספריית ה־ Solution כך שנוכל לקמפל ע"י build solution.

ב־ Linux יש לכלול Makefile שבונה את הפרויקט כאשר מריצים make.

עבור Java יש לכלול קובץ jar המכיל את הפרויקט.

בכל מקרה יש לכלול הוראות קומפילצייה והרצה מפורטות.

בנוסף להגשת הקוד והדוקומנטצייה, יש גם להגיש שתי ספריות בדיקה, אשר כל אחת כוללת קובץ קלט, שורת הרצה עם גדלי בלוקים שתבחרו, וקבצי הפלט שנוצרו מהרצת הסימולטור. הסבירו כיצד קבצי הקלט שהכנתם אכן בודקים באופן טוב את נכונות הסימולטור. כמו כן יש להגיש ולהסביר שלושה גרפים עבור כל ספריית בדיקה (6 גרפים סה"כ):

1. עבור קונפיגורצייה של L1 בלבד (levels=1), יש להגיש גרף של L1 miss rate כפונקצייה של גודל הבלוק, כאשר משנים את גודל הבלוק בין 4 לבין 128 בכפולות של 2.

2. עבור קונפיגורצייה של 2 מטמונים (levels=2), עבור גודל בלוק L2 של 128 בתים, יש להגיש גרף של זמן ריצת הקוד כפונקצייה של גודל בלוק L1, כאשר משנים את גודל הבלוק בין 4 לבין 128 בכפולות של 2.

3. עבור קונפיגורצייה של 2 מטמונים (levels=2), עבור גודל בלוק L1 של 8 בתים, יש להגיש גרף של ה־ AMAT כפונקצייה של גודל בלוק L2, כאשר משנים את גודל הבלוק בין 8 לבין 256 בכפולות של 2.