מבנה מחשבים ספרתיים- תרגיל בית 4

זיכרון מטמון

1 תיאור התרגיל

בתרגיל הבית תממשו סימולאטור של הירארכיית זיכרון. ההירארכיה הנ"ל מורכבת משתי רמות של זיכרון מטמון וזיכרון ראשי. התוכנית תקבל כקלט 2 קבצים: הראשון יכיל את הגדרות המטמונים(פירוט בהמשך) . הקובץ השני יכיל סדרה של גישות לזיכרון. בנוסף, הסימולאטור יקבל שם של קובץ אליו הסימולאטור יצור הירארכיית זיכרון בהתאם לקובץ ההגדרות. עבור כל גישה לזיכרון , הסימולאטור יחזיר כפלט איזה רמה בהירארכיה מכילה את הכתובת הנתונה: L1HIT (מצב של L1 hit ) , L2HIT (מצב בו מתרחש L1 miss ו- L2 hit ) או MEMACC (גישה לזיכרון הראשי בעקבות L1 miss ו- L2 miss).

2 תיאור הקבצים המצורפים

cacheSim.cpp - התוכנית הראשית של הסימולטור.

MemHier.h - קובץ ההגדרות של המחלקה MemHier ,אותה אתם נדרשים לממש .

Cache.h - קובץ הגדרות של המחלקה Cache, אותה אתם נדרשים לממש.

def.txt - קובץ הגדרות לדוגמא עבור הירארכיה.

input.txt - קובץ גישות לדוגמא.

output.txt - קובץ פלט לדוגמא, עבור הגישות שמתוארות בקובץ input.

makefile - קובץ makefile עבור הסימולטור. נשתמש ב- makefile דומה בשביל לקמפל את הסימולטור בבדיקת התרגיל .

3 פורמט קבצי הקלט

הפורמט של קובץ ההגדרות הוא :

CACHE\_LINE\_SIZE, CACHE\_INCLUSIVE,

L1\_NUM\_WAYS, L1\_DATA\_SIZE, L2\_NUM\_WAYS, L2\_DATA\_SIZE

בקובץ ההגדרות עצמו כל הפרמטרים מופיעים בשורה יחידה, מופרדים באמצעות פסיקים . שימו לב שמדיניות הפינוי של כל זיכרון מטמון היא תמיד LRU ומדיניות הכתיבה היא write through , ולכן אינן מוזכרות בקובץ ההגדרות.

תיאור הפרמטרים:

CACHE\_LINE\_SIZE: גודל שורת מטמון עבור L1 ו- L2 .

CACHE\_INCLUSIVE: הערכים האפשריים הם TRUE או FALSE . ערך TRUE משמעותו שמתקיים עיקרון ההכלה ו – L1 מוכל ב- L2 .

עבור כל רמה של מטמון מוגדרים 3 הפרמטרים הבאים:

DATA\_SIZE : מספר הבתים של מידע בזיכרון המטמון (כלומר, לא כולל סיביות הנדרשות לניהול המטמון , כמו TAG ).

CACHE\_NUM\_WAYS : מספר ה-WAYS בזיכרון המטמון,שימו לב שעבור NUM\_WAYS=1 אז ה-Cache הוא direct map.

הנחות על הקלט:

* כל הערכים המספריים יהיו חזקות שלמות של 2 (להזכירכם, גם 1 היא חזקה של 2 ).
* כל הערכים המספריים "תקינים" . לדוגמא, מכפלת CACHE\_LINE\_SIZE ב- NUM\_WAYS קטן מ- DATA\_SIZE.
* גודל CACHE\_LINE\_SIZE הוא לפחות 2
* L2 מכיל יותר מידע מאשר L1

הפורמט של קובץ הגישות לזיכרון:

<ADDRESS1> , W\R

<ADDRESS2> , W\R

…………………..

<ADDRESSn> , W\R

כל שורה מכילה כתובת יחידה אליה ניגשים ודגל המתאר את אופי הגישה לזיכרון : R עבור קריאה מהזיכרון ו – W עבור כתיבה. מצורפים קבצים לדוגמא בהם תוכלו להיעזר .

4 הנחיות למימוש הסימולאטור

אתם תממשו סימולאטור של הירארכית זיכרון בעלת שתי רמות של זיכרון מטמון. גודל כתובת הוא תמיד 32 ביט ומדיניות הפינוי של שתי רמות המטמון היא תמיד LRU. כתיבה נחשבת גם היא כגישה עבור מדיניות ה-LRU . מדיניות הכתיבה היא Write through . מדיניות ה-write miss היא no write allocate . הניחו שבין כל שתי גישות לזיכרון נכתב המידע לכל הרמות הזיכרון.

בקובץ CacheSim.cpp קיימת התוכנית הראשית, אשר מבצעת parsing לקלט, הרצה של הסימולציה וכתיבת של תוצאת הסימולציה לקובץ. התוכנית הראשית מקבלת 3 פרמטרים: הפרמטר הראשון הוא שמו של קובץ ההגדרות. הפרמטר השני הוא שמו של קובץ הגישות לזיכרון . הפרמטר האחרון הוא שמו של הקובץ אליו ייכתב פלט הסימולאטור.

לכם נותר לממש את שתי מחלקות הבאות:

המחלקה Cache אשר מייצגת זיכרון מטמון בהירארכיה. אתם נדרשים לממש עבורה את בנאי המחלקה שחתימתו:

Cache(AddrType numWays,AddrType dataSize, AddrType lineSize)

כאשר AddrType הוא typedef לטיפוס המייצג כתובת. AddrType מוגדר בקובץ Cache.h .

הבנאי מקבל את מספר ה-ways בזיכרון המטמון,גודל ה-data , גודל שורה עבורה זיכרון המטמון.

המחלקה הנוספת אותה אתם נדרשים לממש היא MemHier אשר מייצגת את ההירארכיה עצמה. את נדרשים לממש את בנאי המחלקה שחתימתו:

MemHier( Cache\* L1,Cache\* L2,bool inclusive)

הבנאי מקבל מצביעים לשתי רמות מטמון L1,L2 מטיפוס Cache וכן משתנה בוליאני אשר אם ערכו true אז ההירארכיה מקיימת את עיקרון ההכלה ו - L1 מוכל ב- L2 .

בנוסף אתם נדרשים לממש את המתודה המייצגת גישה לזיכרון שחתימתה:

CacheStatus accessMemory(AddrType addr,bool toWrite)

CacheStatus הוא enum המוגדר בקובץ MemHer.h .

המתודה מקבלת כתובת זיכרון אליה ניגשים וכן דגל אשר ערכו true אם הגישה היא כתיבה לזיכרון (אחרת זוהי קריאה מהזיכרון). המתודה תבצע סימולציה של גישה למטמון ותחזיר כפלט CacheStatus לפי הרמה בהירארכיה אשר הכילה את הזיכרון: L1HIT , L2HIT או MEMACC.

דגשים והנחיות נוספות:

1. אתם רשאים להוסיף מתודות כרצונכם למחלקות MemHier ו- Cache אך אינכם ראשיים לשנות את ההגדרות והחתימות הנתונות בקבצים המצורפים.

2. אתם רשאים להשתמש ב- STL לצורך המימוש.

3. עליכם לקמפל את הקוד באמצעות ה- makefile המצורף.

4.התרגיל ייבדק עם הגדרות נוספות עבור ההירארכיה מעבר לקובץ ההגדרות המצורף, לכן מומלץ לבצע טסטים על מגוון של הגדרות.

5 הוראות הגשה

אתם נדרשים להגיש בקובץ zip המכיל את הקבצים הבאים **בלבד**: Cache.h,Cache.cpp,MemHier.h,MemHier.cpp . שם הקובץ יהיה <id1>\_<id2> , עבור מספרי הסטודנט של הסטודנטים המגישים.