

## תרגיל 2 – חזאי קפיצות

בתרגיל זה עליכם לדמות חזאי קפיצות בעל שני שלבים (2-Level Branch Predictor). החזאי יהיה ניתן להגדרה בזמן היצירה, כך שיתאפשר להגדיר האם החזאי משתמש בהיסטוריה לוקאלית או גלובלית והאם החזאי משתמש בטבלת מכונות מצבים לוקאלית או גלובלית. כמו כן, במקרה של טבלת מכונות מצבים גלובלית, האם החזאי משתמש ב-lshare או gshare. הסימלטר יקבל כקלט trace של פקודות branch והסימולטור ידפיס את החיזוי, T עבור קפיצה שנלקחה, N עבור קפיצה שלא נלקחה.

### הקבצים המצורפים:

הקובץ hw2 מפרש את קובץ הקלט. בקבצים predictor.hpp ו-predictor.cpp כתבו את הקוד שלכם. שנו את הקבצים הללו בלבד. בנייה מתבצעת באמצעות הקובץ makefile, עליכם לוודא שהתרגיל שלכם מתקמפל באמצעות קובץ זה. הרצה באמצעות קריאה ל-hw2 [trace-file]. לבסוף, ניתן קובץ קלט לדוגמא בשם trace.txt.

### מבנה של קובץ trace :

השורה הראשונה של קובץ ה-trace מגדירה את הפרמטרים של החזאי. סדר הפרמטרים הוא:

[local\_tables/global\_tables] [local\_history/global\_history] [history size] [history table size]  
[not\_using\_share/using\_share]

(הסבר על משמעות הפרמטרים יופיע בהמשך).

כל שורה לאחר מכן מתארת פקודת branch שהתרחשה בתוכנית. כל שורה מייצגת את כתובת של פקודת הקפיצה והאם היא נלקחה.

### מבנה החזאים:

כאשר ה-Predictor נוצר, מוגדרת צורת תפקודו. תכונות החזאים הן:

- האם ההיסטוריה לוקאלית או גלובלית (local\_history/global\_history)
- האם טבלת מכונות המצבים לוקאלית או גלובלית (local\_tables/global\_tables)
- גודל טבלת ההיסטוריות. תכונה זאת רלוונטית רק במקרה של היסטוריה לוקאלית. הניחו כי ערך זה קטן או שווה ל-32 (והינו חזקה של 2).
- גודל רגיסטר ההיסטוריה.
- הניחו כי ערך זה קטן או שווה ל-8.
- האם אנו משתמשים ב-lshare \ gshare. תכונה זאת רלוונטית רק במקרה של טבלת חיזוי גלובלית. עבור טבלאות לוקלאיות ערכה תמיד יהיה not\_using\_share

בנוסף, לכל החזאים שתדרשו לממש התכונות המשותפות הבאות:

- כל מכונות מצבים תהיה מכונת שני ביטים בדומה לנלמד בתרגול. המצב ההתחלתי של המכונות יהיה 01 weakly not taken ,

2. טבלת ההיסטוריות מאורגנת בדומה למטמון. הכניסות בטבלת ההיסטוריה מאורגנות כ- direct map . כתובות הפקודות במערכת אינן aligned ולכן קביעת ה-set (המיקום בטבלת ההיסטוריות) תואם ל-LSBs (Least Significant Bits).
3. גודל כתובת במעבד הוא 32 ביט.
4. tag mismatch , מצב בו ה-branch אותו אנו מבצעים לא נמצא בטבלת ההיסטוריה, יכול להתרחש רק במצב בו ההיסטוריה לוקאלית. במצב של היסטוריה גלובלית, הניחו ששמורות ב-BTB כל הכתובות של פקודות ה-branch בתוכנית.
5. כאשר משתמשים בהיסטוריה וטבלת חיזוי לוקאלית ומתרחש tag mismatch , נשתמש בחיזוי ברירת המחדל WNT .
6. לא ייבדק מצב בו ההיסטוריה גלובלית והטבלאות לוקאליות.

### מימוש החזאי

החזאי ייקרא עבור כל שורה של ההרצה (trace). ראשית נקראת המתודה predict ללא מידע באם הקפיצה קרתה בפועל. על המתודה להחזיר True במידה והחיזוי כי הקפיצה נלקחה. לאחר מכן נקראת המתודה update עם תוצאת הקפיצה המותנית בפועל, אשר מעדכנת את החזאי . עליכם לממש את שתי המתודות הללו.

הערות / הקלות:

1. אין צורך שהמימוש יהיה יעיל בזמן או במקום
2. התעלמו מחיזוי כתובות היעד.

### הוראות הגשה

יש להגיש את הקבצים predictor.cpp ו-predictor.hpp בקובץ zip שנקרא <id1>\_<id2> .  
על הקבצים להתקמפל באמצעות קובץ ה-makefile על סביבת לינוקס כדוגמת t2.