הטכניון – מכון טכנולוגי לישראל

הפקולטה למדעי המחשב

תאריך הגשה: 5.6.2016 שעה 11:50am

הוראות הגשה: ההגשה בזוגות . הוסיפו שמות, ת.ז., אי-מייל, תא אליו יש

להחזיר את התרגיל ואת תשובותיכם לתרגיל, הדפיסו והגישו לתא הקורס

בקומה 1. עבור הגשות באיחור יש להגיש לתא של יוני.

מבנה מחשבים ספרתיים – תרגיל יבש 3

|  |  |  |
| --- | --- | --- |
| *שם* | *ת.ז.* | *מייל* |
| עידן אטיאס | 201368958 | idan2k3@gmail.com |
| ליאור בן עמי | 201182839 | liorb\_a@hotmail.com |

להחזיר לתא: 39

שאלה 1 – Two level Branch Prediction

נתון מעבד בעל מבנה ה-pipeline הבא:

Mem

Write

back

Exe

Dec 2

Dec 1

Inst fetch

1. חיזוי הקפיצה נמשך שני מחזורי שעון, כך שהוא מבוצע בסוף שלב dec1 .
2. במעבד קיים branch predictor מסוג Local, שבו לכל branch היסטוריה באורך 3 המצביעה על מערך של 8 מונים (bimodal counters). ההיסטוריה מאותחלת ל-0 והמונים מאותחלים ל-2 (weakly taken) .
3. בסוף שלב ה-mem ידוע האם יש לקפוץ או לא.
   1. כמה penalty (במחזורי שעון) נשלם עבור חיזוי בכל אחד מהמקרים הבאים?

חיזוי taken \ בפועל taken

תשובה: מחזור שעון אחד

חיזוי taken \ בפועל not taken

תשובה: ארבעה מחזורי שעון

חיזוי not taken \ בפועל taken

תשובה: ארבעה מחזורי שעון

חיזוי not taken \ בפועל not taken

תשובה: ללא תשלום

* 1. המעבד מריץ תוכנית ובה פקודת קפיצה בודדת בעלת סידרת הקפיצות הבאה: ...01101 01101 (החל משמאל). השלם את הטבלה הבאה בהנחה שההיסטוריה והמונים מתעדכנים בסוף שלב ה-Mem ושבין קפיצה לקפיצה עובר מספיק זמן, כך שההיסטוריה והמונים מתעדכנים עוד לפני שנדרש לבצע חיזוי נוסף.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| `Taken/  not-taken | ערך ההיסטוריה  לפני הקפיצה | ערך המונה לפני הקפיצה | החיזוי  (0/1) | החיזוי נכון\ שגוי |
| 0 | 000 | 2 | 1 | F |
| 1 | 000 | 1 | 0 | F |
| 1 | 001 | 2 | 1 | T |
| 0 | 011 | 2 | 1 | F |
| 1 | 110 | 2 | 1 | T |
| 0 | 101 | 2 | 1 | F |
| 1 | 010 | 2 | 1 | T |
| 1 | 101 | 1 | 0 | F |
| 0 | 011 | 1 | 0 | T |
| 1 | 110 | 3 | 1 | T |

* 1. עבור כל אחת מהסדרות הבאות ענו האם החזאי בשאלה מסוגל לאחר מספיק חזרות לחזות אותן בצורה מושלמת. אם לדעתך החזאי אינו מסוגל לחזות את הדפוס בצורה מושלמת, נמק מדוע, וקבע מה אורך ההיסטוריה המינימאלי הנדרש על מנת לתת תשובה נכונה.

א.100010111 100010111 100010111….

תשובה: החזאי אינו יכול לחזות בצורה מושלמת, אורך ההיסטוריה המינימלי עבור סדרה זאת הוא 4. הסבר: נשים לב כי עבור רצף ההיסטוריה 111 יש שני חיזויים ברצף: 1 ו -0. בפעם הראשונה החיזוי הוא WT=2 ולכן יהיה נכון ויעבור ל ST=3. אך מיד לאחר מכן לא תתבצע קפיצה ולכן יהיה שגוי ויעבור ל WT שוב. בפעם השלישית נחזור למקרה הראשון וחוזר חלילה. עבור שאר הרצפים באורך 3 החזאי תקין לכן צריך היסטוריה באורך 4 ומעלה כדי שהחיזוי יהיה תקין תמיד.

ב. 0101111 0101111 0101111…..

תשובה: החזאי אינו יכול לחזות בצורה מושלמת, אורך ההיסטוריה המינימלי עבור סדרה זאת הוא 4. הסבר: זהה לסעיף הקודם.

ג. 1011001 1011001 1011001…

תשובה: החזאי אינו יכול לחזות בצורה מושלמת, אורך ההיסטוריה המינימלי עבור סדרה זאת הוא 5. הסבר: עבור הרצף 110 בפעם הראשונה הוא החיזוי אמור לתת 0 אך יתן 1 WT=2, ואז יעבור ל WNT=1. בפעם הבאה ייתן לכן 0 אך אמור לתת 1 וחוזר חלילה. עבור היסטוריה באורך 4, לא יסתדר הפעם מכיוון שהרצף 0110 בעל אותו מבנה ( 0 ואז 1). הפעם צריך היסטוריה באורך 5 ומעלה כדי שהחיזוי יהיה מושלם.

4.מריצים benchmark בעל מספר פקודות גדול על המעבד. 20% מהפקודות הן פקודות קפיצה. 60% מהחיזויים הם חיזוי Taken . 75% מחיזוי ה-not taken ו- 50% מחיזוי ה- taken נכונים . כל ה- data hazards בקוד נפתרים באמצעות forwarding . אין החטאות במטמון במהלך ריצת התוכנית. מהו ה- cpi של התוכנית?

תשובה

שאלה 2 – זיכרון וירטואלי

א.מנה שני יתרונות לכך שהדפים הוירטואלים והפיסיים מיושרים(aligned) יחסית לגודל של הדף (כלומר - עבור גודל דף 4KB , לדוגמא, כתובותיהם יתחילו ב 12 אפסים. )

תשובה:

* 1. מנגנון התרגום מכתובת וירטואלית לפיסית מתרגם VPN 🡺 PPN ולכן עבור כל כתובת וירטואלית, שכאמור מיושרת לפי גודל דף, זמן התרגום מתקצר כיוון שאין צורך לתרגם log(|page|) ביטים תחתונים.
  2. נוכל למקבל גישה ל-Cache ול- TLB:

מכיוון ש- log(|page|) ביטים תחתונים (המבטאים את ה-offset בתוך הדף המבוקש) אינם צריכים תרגום נוכל לחלץ בעזרתם את מספר הסט המתאים ב Cache וע"י כך לקבל את ה – tag המתאים של ה – cache line. בו בזמן, נוכל לקחת את שאר הביטים (העליונים) של הכתובת ולחפש מיפוי של VPN 🡺 PPN ב-TLB. במידה וקיבלנו TLB HIT אזי כל שנותר לעשות הוא להשוות האם ה-tag של ה PPN שקיבלנו שווה ל tag של ה cache line ואם כן אז קיבלנו cache hit.

\*נשים לב שנוכל לעשות זאת רק כיוון שידוע לנו כי מספר מסוים של ביטים תחתונים לא צריך תרגום. (וזה כאמור נובע מכך שכל הדפים מיושרים יחסית לגודל דף)

ב. בארכיטקטורת x86 עם גודל כתובת של 32 ביט, אילו סוגי טבלאות תרגום חייבים להימצא בזיכרון הראשי?

תשובה:

טבלאות ה-PGD של כל תהליך חייבות להימצא ב-RAM.

טבלת הPGD היא הטבלה הראשונה בהיררכית טבלאות התרגום בארכיטקטורה הנ"ל. ישנו רגיסטר CR3 שתמיד מצביע על הPGD של התהליך הנוכחי וההגדרה היא שתמיד הטבלה (PGD) שמוצבעת ע"י כתובת זו - מוקצית.





