**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 | T |
| 1 | 000 | 1 | 0 | F |
| 1 | 001 | 2 | 1 | T |
| 0 | 011 | 3 | 1 | F |
| 1 | 110 | 2 | 1 | T |
| 0 | 101 | 3 | 1 | F |
| 1 | 011 | 2 | 1 | T |
| 1 | 111 | 3 | 1 | T |
| 0 | 111 | 3 | 1 | F |
| 1 | 110 | 2 | 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.10010111 100010111 100010111….

ב. 0101111 0101111 0101111…..

ג. 1011001 1011001 1011001… 0

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.**

**שאלה 3- זיכרון וירטואלי**

נתון מעבד דמוי x86 העובד במוד של 64 ביט ומבנה הכתובת הבא:

11 12 23 24 31 32 39 40 47 48 63 0

offset

PTE

DIR

PDP

PML4

Sign ext

המעבד תומך הן בדפים בגודל קטן (המוצבעים ע"י כניסות ב-PTE) והן בדפים בגודל גדול (המוצבעים

ע"י כניסות ב-DIR). גודל כל כניסה בטבלאות הדפים בכל אחת מהרמות היא 8 bytes.

במעבד זה, טבלאות התרגום אינן בהכרח בגודל של דף קטן. במעבד קיימים TLB וכן PMH caches עבור כל הרמות PML4,PDP,DIR. בכל אחד מהם 4 כניסות, direct mapped. למעבד אין STLB .

א.מהם הגדלים של דף קטן ודף גדול במעבד?

ב. כמה סיביות אנו שומרים עבור שדה ה-tag ב- TLB וב PMH caches: PML4, PDP וDIR?

ג. נתונה סדרת פניות לזיכרון וירטואלי (בבסיס הקסדצימאלי) . עבור כל גישה , רשמו האם הייתה החטאה או פגיעה ב- TLB וב-PMH caches . הוסיפו הסבר מתאים. הניחו שמשתמשים בדפים קטנים בלבד, שהטבלאות התרגום כבר קימות בזיכרון ושה- TLB וה- PMH caches ריקים בתחילת התוכנית.

|  |  |  |
| --- | --- | --- |
| הסבר | TLBs(H/M) | כתובות |
|  | PML4:  PDP:  DIR:  TLB: | FFFF123456789ABC |
|  | PML4:  PDP:  DIR:  TLB: | FFFF123457789ABC |
|  | PML4:  PDP:  DIR:  TLB: | FFFF123456789BCD |
|  | PML4:  PDP:  DIR:  TLB: | FFFF123322788BCD |
|  | PML4:  PDP:  DIR:  TLB: | FFFF123456789BCD |

ד. זמן הגישה לזיכרון הוא 100 מ"ש(מחזורי שעון). זמן הגישה ל- TLBהוא 2 מ"ש (להחטאה או קבלת הנתון). לאחר TLB miss מתבצעת פנייה ל-PMH כדי שיבצע page walk להשגת התרגום. זמן הגישה ב-PMH בכל הרמות הוא 3 מ"ש עד לקבלת הנתון או זיהוי החטאה. למעבד יש זיכרון מטמון : גודלו 32 kb ,גודל שורה 64 בתים וארגון 8-way set associative עם מדיניות פינו LRU .זמן הגישה למטמון (לקבלת נתון או החטאה) הוא 4 מ"ש.

עבור כל אחת מהגישות לזיכרון שהוזכרו בסעיף ב', תוך כמה מ"ש התקבל תרגום לכתובת?

יש לחשב את הזמן הנדרש לקבלת תרגום עבור כל אחת מהכתובות, ולפרט את הגישות השונות שבוצעו לקבלת התרגום, את תוצאת כל גישה (hit או miss) ואת הזמן שהגישה ארכה.

|  |  |  |
| --- | --- | --- |
| הסבר | TLBs(H/M) | כתובות |
|  |  | FFFF123456789ABC |
|  |  | FFFF123457789ABC |
|  |  | FFFF123456789BCD |
|  |  | FFFF123322788BCD |
|  |  | FFFF123456789BCD |