**הטכניון – מכון טכנולוגי לישראל**

**הפקולטה למדעי המחשב**

תאריך הגשה: 20.4.16

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 1**

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

**להחזיר לתא: 39**

**שאלה 1 – structural hazard pipeline**

בחברת dual-flop מייצרים מעבדים דמויי MIPS . כאשר בדקו בחברה בדקו את תקינות המעבדים, הם גילו להפתעתם שבקובץ הרגיסטרים יש בעיה שכתוצאה ממנה לא ניתן לכתוב ולקרוא מקובץ הרגיסטרים באותו מחזור שעון.

תזכורת: structural hazard מתרחש כאשר אין חומרה מספקת בשביל לבצע 2 פקודות במקביל.

1. באילו מצבים תתרחש במעבד הפגום בעיה? בתשובתך התייחס לסוג הפקודה. להזכירכם ישנם 3 סוגי פקודות במעבד MIPS : R-tye,I-type,J-Type .

**תשובה:**

* Type J – פקודה מסוג זה היא מהצורה הבאה:

|  |  |
| --- | --- |
| XI  26 bits | OP  6 bit |

לכן, אין לה שימוש בקובץ הרגיסטרים.

* Type R – פקודה מסוג זה יכולה לרשום תוצאה לרגיסטרים תו"כ מעקף של שלב ה memory. מצב זה בו היא כותבת במחזור שעון הרביעי לריצתה עלול לגרום ל structural hazard באופן הבא: פקודה Add R2,R3,R5 רוצה לכתוב ל R2 (בזמן בו היה אמור להתבצע שלב ה memory). בנוסף, פקודה נוספת Add R1,R4,R6 מנסה באותו מחזור שעון לקרוא את הרגיסטרים R4,R6 וזה כאמור לא יוכל להתבצע בגלל התקלה הנתונה.
* Type I - פקודה מסוג זה יכולה לרשום תוצאה לרגיסטרים לכן יכולה לגרום ל structural hazard באופן הבא: פקודת Load - LW R1, (30)R2 רוצה לרשום תוצאה בשלב ה WB לרגיסטר R1. באותו מחזור שעון, פקודה Add R1,R4,R6 אשר נכנסה למערכת שלושה מחזורי שעון אחריה, כלומר נמצאת כרגע בשלב ה -Decode מנסה לקרוא מהרגיסטרים R4,R6 ונכשלת בעקבות התקלה המתוארת.
* נשים לב שהתקלות אינן מתרחשות עקב תלות ברגיסטרים ספציפיים (אשר במקרה כזה הבקר יטפל כפי שנלמד בקורס), אלא בעקבות התקלה הנתונה בה לא ניתן לקרוא ולכתוב בו זמנית מקובץ הרגיסטרים.

1. מנהלי החברה לא הסכימו לייצר מחדש את קבצי הרגיסטרים בגלל העליות הייצור הגבוהות. אחד המהנדסים בחברה הציע לשכלל את ה HDU כדי לפתור את הבעיה. תאר את השינוי ב HDU.

**תשובה:**

עבור כל פקודה שנמצאת בשלב ה IF הבקר יבדוק האם היא צריכה לקרוא רגיסטרים. אם כן, נבדוק בעזרת הבקר האם במחזור שעון הבא צריכה להתבצע כתיבה לקובץ הרגיסטרים. אם כן, הבקר ישלח חיווי מתאים ל-HDU שבתורו יעכב את הפקודה שנמצאת בשלב ה-IF במחזור שעון אחד (stall).

1. בהנחה שאין data hazards ו- control hazards (hazards שנובעים מהסתעפות) מהו ה CPI המקסימלי המתקבל ? הדגימו באמצעות תכנית דוגמא ותיאור ה .PIPLINE

תשובה:

(1) LW $1<-(100)($2)

(2) LW $3<-(100)($4)

(3) LW $5<-(100)($6)

(4) LW $7<-(100)($8)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| WB | MEM | EXE | ID | IF | Clock |
|  |  |  |  | 1 | 1 |
|  |  |  | 1 | 2 | 2 |
|  |  | 1 | stall | 2 | 3 |
|  | 1 | stall | stall | 2 | 4 |
| 1 | stall | stall | stall | 2 | 5 |
| stall | stall | stall | 2 | 3 | 6 |
| stall | stall | 2 | 3 | 4 | 7 |
| stall | 2 | 3 | 4 |  | 8 |
| 2 | 3 | 4 |  |  | 9 |
| 3 | 4 |  |  |  | 10 |
| 4 |  |  |  |  | 11 |

**שאלה 2 - data hazards**

1. נתון מעבד pipeline דמוי MIPS בעל 7 שלבים, כך שהגישה לזיכרון מתרחשת במהלך שלושה שלבים (MEM1/MEM2/MEM3) . רק בסוף שלב MEM3 מסתיימת הגישה לזיכרון ,בין אם זאת הייתה קריאה או כתיבה. ציין באלו מקרים לא ניתן להימנע מ stall ע"י קידום מידע ( DATA FARWARDING ) ולמשך כמה מחזורי שעון.

**תשובה:**

* מקרה ראשון – כשתתבצע פקודת LW ואחריה פקודת חישוב אשר תשתמש ברגיסטר שאליו טענו את המילה בLW הקודמת. במקרה זה לא נוכל להימנע מ- 3 stalls (כלומר 2 stalls נוספים מעבר ל pipeline MIPS הרגיל).
* מקרה שני - כשתתבצע פקודת LW, אחריה **פקודה לא תלויה** ואחריה פקודת חישוב אשר תשתמש ברגיסטר שאליו טענו את המילה בLW הקודמת. במקרה זה נקבל 2 stalls.
* מקרה שלישי - כשתתבצע פקודת LW, אחריה **שתי** **פקודות לא תלויות,** ואחריהן פקודת חישוב אשר תשתמש ברגיסטר שאליו טענו את המילה בLW הקודמת. במקרה זה נקבל stallאחד.

1. נתון מעבד pipeline דמוי MIPS בעל 4 שלבים, שמבצע את שלבי ה-MEM וה-EXE באותו מחזור שעון. האם יהיה מקרה בו לא ניתן להימנע מ stall ע"י קידום מידע ( DATA FARWARDING ) ? הסבירו.

**תשובה:**

לא. בעזרת data farwarding נוכל להימנע מ- stall כאשר מדובר במערכת שמבצעת את שלבי ה mem ו exe ביחד. נשים לב כי במקרה הרגיל (MIPS רגיל) היינו מקבלים stall יחיד במידה והייתה תלות בין פקודת LW עם רגיסטר R ולאחריה פקודה שמשתמשת ברגיסטר R. עתה, כיוון שאיחדנו את שלבי ה MEM ו EXE, כעבור מחזור שעון יחיד הערך בו נרצה להשתמש כבר יהיה ולידי.

במקרים אחרים כמובן שלא יהיה stall מכיוון שגם שבמקרה הרגיל נוכל להימנע מ stall. (וכעת ה MIPS "משופר" יותר מבחינה זו).

**שאלה 3 - מדדים וחוק אמדל**

ישנו מעבד בו התדר הוא 2GHz. מריצים עליו את תוכנית המבחן הבאה:

|  |  |  |
| --- | --- | --- |
| סוג הפקודה | אחוז הפקודות בתכנית | CPI |
| פקודות INTEGER | 50% | 1 |
| חישובי FP | 10% | 3 |
| Branch | 20% | 2 |
| Load | 20% | 4 |

1. מהו ערך מדד ה-MIPS עבור תוכנית המבחן הזאת?

תשובה:

ב. משנים את המעבד כך שעבור פקודת LOAD יהיה CPI=2 . כתוצאה משינוי זה CPI של פקודות FP הוא כעת 4. **הראו באמצעות חוק אמדל** מהו ה-speedup הכולל של ה-benchmark לאחר השינוי ?

תשובה:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| סוג הפקודה | אחוז הפקודות בתכנית | CPI | New CPI | Speedup |
| פקודות INTEGER | 50% | 1 | 1 | 1 |
| חישובי FP | 10% | 3 | 4 | 3/4 |
| Branch | 20% | 2 | 2 | 1 |
| Load | 20% | 4 | 2 | 2 |

**אין קשר בין סעיף ג' לסעיפים הקודמים**

ג. מאיצים את פקודת החילוק במעבד נתון פי 1.5. עבור benchmark כלשהוא השגנו speedup כולל של 1.2 . מצאו באמצעות חוק אמדל מהו **אחוז מזמן** ריצת המעבד שבו ביצע מהעבד פעולות חילוק **לפני השיפור** ?

תשובה:

הסבר: המעבר הראשון (מהפרמטרים למספרים) נובע מהצבה בנוסחה. המעבר השני הוא על מנת לקבל את אחוז הזמן הנדרש.