**A picture containing knife

Description automatically generatedהמחלקה להנדסת חשמל ומחשבים**

**קורס מבנה המחשב**.

**פרויקט מס. 1**

**מטרת הפרויקט:**

1. למידה מתקדמת של נושא ה- pipeline
2. למידה מעמיקה של נושאי ה – pipeline hazards
3. למידת השפעת ה- pipeline על ביצועי המעבד.

**תיאור הפרויקט:**

נדרש לכתוב סימולטור ל- pipeline של מעבד ה – riscv.

על הסימולטור לקרוא קובץ trace המכיל את רצף הפקודות שבוצעו ע״י המעבד.

נדרש לתאר את מצב ה- pipline בכל מחזור שעון כפי שיתואר.

**הנחיות הגשה:**

1. את הסימולטור יש לכתוב בשפת C ולהגיש אותו בקובץ יחיד riscv.c.
2. בנוסף, יש להעתיק את כל התוכנית לקובץ WORD אחד. יש לציין את מספר השאלה לפני כל תכנית.
3. **את הפלט של הסימולטור יש להגיש בקובץ נפרד לכל הרצה.**
4. את כל הקבצים יש לרכז בקובץ ZIP (לא קובץ rar) אחד ולהעלות לאתר
5. יש להקפיד ולתעד בקוד על ידי הוספת הערות

**דרישות הפרויקט:**

1. קלט – הקלט לסימולטור נתון על ידי קובץ trace המתאר את הפקודות שבוצעו בתכנית. פורמט כל שורה בקובץ נתון להלן:

<instruction address> <opcode> <operand1> <operand2> <operand3>

<instruction address> - כתובת הפקודה בזיכרון

<opcode> - ה- opcode של הפקודה. ה- opcodes הנתמכים ע״י הסימולטור הם:

add, sub, or, and, addi, subi, ori, andi, lw, sw, j, beq, bneq

<operand1-3> הינם האופרנדים של הפקודה בהתאם לפורמט שלה. האופרנד הכי שמאלי הוא האופרנד שנכתב ע״י הפקודה.

כל השדות בכל קורה מופרדים ע״י רווח. למשל:

100 add x1 x2 x3 //x1=x2+x3

104 lw x2 100 x4 //x2=Mem[100+x4]

108 subi x3 x10 1 //x3=x10-1

109 beq x3 x0 label //if(x3==x0) goto label

1. סימולציה של ה- RISC-V pipeline – על הסימולטור להציג בכל מחזור שעון את הפקודה הנמצאת בכל שלב ב- pipeline של ה- RISC-V. במידה ויש stall על הסימולטור להציג זאת בפלט. בסוף הסימולציה על הסימולטור לחשב את ה-CPI של התכנית.

נתונים:

* גישות לזיכרון לנתונים ולפקודות הן בשלב ה- MEM וה- Fetch בהתאמה ואורכות מחזור שעון יחיד.
* ביצוע הפקודות בשלב ה- Ex דורש מחזור שעון יחיד.
* המעבד משתמש בחיזוי סטטי של predict not taken
* הסימולטור מקבל 2 פרמטרים בשורת ההרצה:
  + פרמטר ראשון: 0 הריצה ללא forwarding, 1 המעבד תומך בכל סוגי ה- forwarding האפשריים.
  + פרמטר שני: 0 ה- branch resolution מתבצע בשלב ה- MEM, 1 ה- branch resolution מתבצע בשלב ה- ID.

למשל ההרצאה mips.exe 1 1 תריץ את הסימולטור במוד של תמיכה ב- forwarding וה- branch resolution יתבצע בשלב ה- ID.

1. פלט – על הפלט להיות בפורמט על פי הדוגמא הבאה לכל מחזור שעון:

Cycle 30:

Fetch instruction: lw x10 2 x8

Decode instruction: sw x8 12 x0

Execute instruction: subi x9 x0 -25

Memory instruction: subi x8 x0 -10

Writeback instruction: stall

Cycle 31:

Fetch instruction: stall

Decode instruction: lw x10 2 x8

Execute instruction: sw x8 12 x0

Memory instruction: subi x9 x0 -25

Writeback instruction: subi x8 x0 -10

CPI: 1

4. הרצה – את הסימולטור יש להריץ על פי הטבלה הנתונה להלן:

|  |  |  |  |
| --- | --- | --- | --- |
| הרצה | קובץ trace | Forwarding | Branch resolution |
| 1 | trace1.txt | 0 | 0 |
| 2 | trace1.txt | 1 | 0 |
| 3 | trace1.txt | 0 | 1 |
| 4 | trace1.txt | 1 | 1 |
| 5 | trace2.txt | 1 | 1 |

קבצי הקלט trace1.txt ו- trace2.txt נתונים באתר המודל.

**את הפלט של הסימולטור יש להגיש בקובץ נפרד לכל הרצה.**

בהצלחה!!!