**ממן 12 טל פלדמן ארגון המחשב 328441670**

**שאלה 1 (10%)**

1. X<Y, כאשר X ו-Y נחשבים כמספרים ללא סימן.

X<Y אם"ם:

נרשום בצורה של פונקציה לוגית:

ניתן להרחיב את הפונקציה גם עבור יותר סיביות:

1. X<Y, כאשר X ו-Y נחשבים כמספרים עם סימן (משלים ל-2)

X<Y אם"ם:

נרשום בצורה של פונקציה לוגית:

ניתן להרחיב את הפונקציה גם עבור יותר סיביות:

כך ש היא הפונ' מסעיף א כאשר Y<X

1. X=Y

X=Y אם"ם:

נרשום בצורה של פונקציה לוגית:

ניתן להרחיב את הפונקציה גם עבור יותר סיביות:

**שאלה 2 (15%)**

א. על סמך תרשים Figure B.5.10 התחתון , מה מיוחד בסיבית ה MSB (most significant bit) ב ALU איזה מידע נוסף היא מכילה ?

הALU מבצע פעולות הדורשות התייחסות לסיבית הסימן – סיבית הMSB. בשיטת המשלים ל2 הסיבית הזו קובעת אם המספר שלילי או חיובי ולכן היא מחייבת התייחסות מיוחדת. כאשר מחסרים את 2 המספרים והתוצאה שלילית הALU יחזיר חיווי בהתאם בסיבית הset (פקודת הslt) . כמו כן, כאשר מחברים 2 מספרים חיוביים גדולים והתוצאה גדולה מהגודל שניתן לאחסן נקבל חיווי לכך בoverflow.

ב. על סמך Figure B.5.12, איפה בא לידי ביטוי ההבדל בין פקודות sub ל subu אותן למדנו בפרק ד במדריך.

ההבדל בא לידי ביטוי אך ורק בoverflow. בsub כאשר הסיבית של הoverflow תהיה 1 נבצע trap, אך בsubu נתעלם מכך.

**שאלה 3 (40%)**

ענו על השאלות הבאות. **יש להתייחס לכל סעיף בנפרד ללא קשר לשאר הסעיפים** (למעט הקידוד של סעיף א ששימושי לכל הסעיפים).

1. מה קידוד הפקודה beq $t3,$t4,luxury בבסיס הקסא?

|  |  |  |  |
| --- | --- | --- | --- |
| immediate | rt | rs | opcode |
| )0x000040A4- 0x00004010-0x4 (/0x4  imm=0000000000100100 | 01100 | 01011 | 000100 |

1. מה תבצע הפקודה beq $t3,$t4,luxury ?

הפקודה בודקת האם הערכים ב2 הרגיסטרים בכתובות rs ו-rt שווים, אם כן היא מוסיפה לpc את הערך בimm\*4+4 (קופצת לתווית) ואם לא היא ממשיכה בקוד. במקרה שלנו, בt3 וt4 ערכים שונים ולכן לא נקפוץ.

1. מהו הערך של הקו המסומן A בשרטוט במהלך ביצוע הפקודה?
2. מהו הערך של הקו המסומן B בשרטוט במהלך ביצוע הפקודה?
3. מהו הערך של הקו המסומן C בשרטוט במהלך ביצוע הפקודה?
4. מהו הערך של הקו המסומן D בשרטוט במהלך ביצוע הפקודה?

*הערכים ברגיסטרים לא שווים ולכן הערך של הקו D יהיה*

1. מהו הערך של הקו המסומן E בשרטוט במהלך ביצוע הפקודה?

*אנו לוקחים את הopcode, הControl יוציא לנו את ה ALUOp של beq עבורו הALU control יוציא את הקוד 0110 וזה הערך של הקו E.*

1. אם קו RegWrite ניתקע על הערך 1, מה תבצע הפקודה ?(יש לבדוק את כל האפשרויות)

*ישנן מספר אפשרויות עבור MemtoReg, RegDest שיכולים להיות 0 או 1*

*1. MemtoReg=0 – תוצאת הALU זו שתיכתב -*

* *RegDest=0, התוצאה תיכתב לרגיסטר בשדה הrt של קידוד הפקודה(t4).*
* *RegDest=1, התוצאה תיכתב לרגיסטר בשדה הrd של קידוד הפקודה, אך במקרה שלנו השדה יכיל 00000 (רגיסטר $zero) ולכן לא תתבצע כתיבה כלל.*

*2. MemtoReg=1 – Read data של זיכרון הנתונים ייכתב, במקרה שלנו לא ניתן לדעת מה הוא יכיל.*

* *RegDest=0, הערך ייכתב לרגיסטר בשדה הrt של קידוד הפקודה(t4).*
* *RegDest=1, הערך ייכתב לרגיסטר בשדה הrd של קידוד הפקודה, אך במקרה שלנו השדה יכיל 00000 (רגיסטר $zero) ולכן לא תתבצע כתיבה כלל.*

1. אם ערכו של הקו memwrite נתקע על הערך 1, מה תבצע הפקודה ?

*נרשום את הערך ALU result לAddress בזיכרון של Read data 2(Instruction[20-16]=$t4 value). כלומר נרשום בכתובת בזיכרון במקרה שלנו.*

1. אם קו ALUSrc ניתקע על הערך 1 מה תהיה תוצאת ה ALU (ALU result) ?

*במקרה הזה נקבל בALU פעולת חיסור בין האופרנד המקורי $t3 לבין ערך הimmediate בגודל של 32-bit, כלומר - וזה יהיה הערך בALU result.*

1. אם ערכו של הקו aluop ניתקע על הערך 10(בינארי) מה יהיה הערך של הקווים E,C,D ?

עבור נקבל:

*הקו C ישאר ללא שינוי, נשים לב שהוא לא תלוי ב כלל -*

*הALU control יתרגם את ה ו לפעולת AND והערך של E יהיה בהתאם .*

*תוצאת הפעולה היא ולכן קו הZERO נדלק, ומכאן ערך הקו D יהיה עכשיו כערכו של C (כלומר, נבצע קפיצה) - .*

שאלה 4 (35%)

1. המרבבים של MemtoReg ו RegDst הורחבו בין אילו ערכים מנתבים מרבבים אילו ?

*המרבב של MemtoReg עכשיו בורר בין הערכים 0 ל-3, כלומר הוא מרבב 4 ל-1. נוספו הקו מהshift-left 16 והקו מהחישוב של pc+4.*

*המרבב של MemtoReg עכשיו בורר בין הערכים 0 ל-2, כלומר הוא מרבב 3 ל-1. נוסף קו המכיל את המספר 31 בדצימלי(10000).*

1. נתונה הפקודה שאלו הן קווי הבקרה שלה:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RegDst** | **ALUSrc** | **MemtoReg** | **RegWrite** | **MemRead** | **MemWrite** | **Branch** | **ALUOp** |
| **0** | **X** | **2** | **1** | **0** | **0** | **0** | **XX** |

איזו פקודה זו ? (בתוך הפקודות שלמדנו בפרק התכנות) מה מבצעת פקודה זו ?

*אנחנו כותבים לרגיסטר בInstruction[20-16] את Instruction[15-0] שיפט שמאלה 16 ביטים.*

*זה כמובן מתאים לפקודת lui.*

1. התרשים בעמוד הבא מציג את הערך המגיע לכניסה 2 במרבב MemtoReg עובר הרחבת סימן

(sign extend) . האם תשובתכם לסעיף ב תשתנה ? נמקו את תשובתכם במידה וכן האם ניתן לדעת מה תבצע הפקודה ?

*עבור מספר שלילי בגודל של 16 סיביות, בסעיף ב נטען אותו במו שהוא ב16 סיביות העליונות של הרגיסטר, אך במקרה הזה אנחנו קודם נרחיב אותו ל32 סיביות- כלומר סיבית הסימן בביט ה32 תהיה דלוקה, אך כאשר נעשה שיפט שמאלה 16 פעמים נאבד את סיבית הסימן הזו.*

*לכן בסעיף זה אנחנו נטען אמנם מספר ל16 סיביות העליונות של הרגיסטר אך לא את אותו המספר במקרה שהוא שלילי.*

1. נתונים קווי הבקרה הבאים.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **RegDst** | **ALUSrc** | **MemtoReg** | **RegWrite** | **MemRead** | **MemWrite** | **Branch** | **ALUOp** |
| **2** | **X** | **3** | **1** | **0** | **0** | **0** | **XX** |

איזה פעולה מתבצעת ? וכחלק מאיזה פקודה שלמדנו בפרק התכנות יכולה להיות פקודה זו ? מה חסר בשרטוט על מנת שהפקודה תתבצע במלואה ?

ייכתב לרגיסטר ra (מספר 31) הערך של pc+4. זה מתאים לפעולת הjal, אנחנו שמים את הערך העכשווי של הpc+4 וקופצים לערך בהתאם לפקודה מסוג J. החלק החסר בשרטוט על מנת שהפקודה תתבצע במלואה הוא חישוב הערך שאליו צריכים לקפוץ(לקחת את Instruction[27-0]) והקפיצה אליו (שהקו Branch יהיה דלוק).