|  |  |  |
| --- | --- | --- |
| **BAR-ILAN UNIVERSITY**  Engineering Faculty |  | אוניברסיטת בר-אילן הפקולטה להנדסה |

תכן לוגי ומבוא להנדסת מחשבים

# תשע"ה סמס' א' מועד ב'

**83-253**

* מרצה : פרופ' שמואל וימר
* מתרגלים : מר יוסף לונדון, מר עמית מנדלבום
* **חומר עזר מותר:** ספר הקורס בלבד (אפשרי גם מודפס ונקי מכתב יד), שקפי הקורס מודפסים, קובץ פקודות אסמבלי מודפס.
* משך המבחן:

לסטודנטים משנה ב' הלומדים קורס תכן לוגי לראשונה – שעתיים וחצי.

לשאר הסטודנטים - שלוש שעות.

* לסטודנטים משנה ב' הלומדים קורס תכן לוגי לראשונה – חובה לענות על שתי השאלות הראשונות בלבד. אין לענות על שאלה 3.
* לשאר הסטודנטים – חובה לענות על כל שלושת השאלות.
* **יש לנמק את כל תשובותיכם**. אין צורך לפתח מחדש תוצאות שהוכחו בכיתה, אלא אם כן נאמר מפורשות לעשות כן.
* יש לשרטט דיאגראמות באופן ברור !
* הכתיבה בעט בלבד. כתיבה בעפרון לא תיבדק.
* יש לכתוב רק בחלק הדף המיועד לכך.
* סה"כ הניקוד הינו 100 (הניקוד בסוגריים לנבחני שנה ג' וד')

**בהצלחה!**

שאלה 1**– 45 נק' (37 לתלמידים שאינם בשנה ב')**

במעבד ה- Multi Cycle MIPS רוצים לשפר את ביצועי המחשב כך שבפקודות אשר לא עושות שימוש בזיכרון במחזור האחרון שלהם, יתבצע שלב Pre-fetch, כלומר Fetch של הפקודה הבאה.

1. באיזה סוגי פקודות לא ניתן לבצע פעולה זו, הסבר בפירוט מדוע.
2. האם נדרשים שינויים ב-DATA PATH על מנת לבצע שינוי זה? הסבר בפירוט.
3. בסוף השאלות מצורפת מכונת מצבים של Multi cycle MIPS, רשום את כל השינויים הנדרשים במכונה על מנת לתמוך בשינוי זה. שים לב, ניתן לשנות מצבים וקווים קיימים. **שימו לב:** יש לרשום את השינויים במחברת הבחינה ולא על גבי הטופס
4. לפניך קוד בשפת Assembly הממומש הן ב- Multy cycle MIPS התומך ב- Pre-fetch, והן ב- Pipelined MIPS, בעל תמיכה מלאה ב- Hazards , יש לרשום באיזה מחזור שעון תסתיים כל פקודה עבור כל אחד מה-MIPS:

ADD R1 R2 R3

SW R3 #5(R2)

LW R2 #3(R3)

ADD R1 R5 R7

ADDI R2 R2 #1

1. בתוכנית סטנדרטית ישנה החלוקה הבאה של פקודות:

|  |  |
| --- | --- |
| סוג פקודה | אחוזים מסך הפקודות |
| Load | 25% |
| Store | 10% |
| Branch | 11% |
| Jump | 2% |
| R-Type | 27% |
| I-Type | 25% |

יש לרשום CPI עבור Multi cycle MIPS עם ובלי תמיכה ב- Pre fetch

שאלה 2**55 נק' (43 נק' לתלמידים שאינם בשנה ב')**

נתונה פקודה חדשה – UNDO. כמו כל פקודה בISA , המתכנת באסמבלי כותב אותה.  
הפקודה מבטלת את הכתיבה לרגיסטר של פעולת **RTYPE** (בלבד) הקודמת לה, במידה ונכתב אל הרגיסטר המספר XXXXX שהינו מספר ש**צרוב** במערכת (אינו פרמטר – ערך ידוע למערכת שלכם). כלומר – במידה ולרגיסטר מסוים **נכתב** XXXXX בפעולת RTYPE שלפני הפקודה UNDO, הפקודה UNDO תשחזר את המידע שהיה ברגיסטר לפני כן.

לדוגמא – מתכנת כתב:

Add $r2,$r3,$r4  
UNDO

ערך הרגיסטר $r2 לאחר שתי הפקודות יחזור להיות מה שהיה במידה וסכום שני הרגיסטרים האחרים שווה לXXXXX.  
שימו לב – הפקודה איננה עוצרת כתיבה לרגיסטר אלא "מבטלת" כתיבה שקרתה. ממשו בהתאם.

ממשו את הפקודה בPIPELINED MIPS. לאורך כל השאלה התעלמו מבעיית OVERFLOW וכדו. כמו כן, שימו דגש על קווי בקרה וכן, שימו דגש על מינימיזציה של שינויים **בתוך** המערכת ליחידות מובנות כגון ALU וכדו' (תוספות מותר).

1. אלו שינויים צריכים להתבצע במערכת מבחינת חומרה (שינויים לקובץ הרגיסטרים, ALU, וכדו')? דגש – שימו לב, במידה ויש, לשינויים שהם אקוטיים ל**כל** תצורת פתרון עליהם יינתן דגש.
2. איזה פורמט פקודה מתאים ולמה? הציעו פורמט שמתאים למימוש שלכם.
3. האם יווצרו בעיות HAZARD חדשות? הסבירו פרטו ונמקו.
4. כעת, הניחו שהמספר XXXXX **אינו** צרוב אלא פרמטר שהמתכנת מכניס בעת **כתיבת** הפקודה. UNDO XXXXX.

ענו על כל הסעיפים לעיל מחדש עבור XXXXX כפרמטר. מהו הגודל המקסימלי הניתן לXXXXX?

שאלה 3  **(20 נק' לתלמידים שאינם בשנה ב')**



הגודל של כתובת הוא 32 סיביות. גודל בלוק הוא 8 בייט כפי שניתן לראות. גודל הindex הוא 10 ביטים.

1. מהו גודל שדה הTAG?
2. כמה מידע הCACHE יכול להכיל?
3. מהו הגודל של הCAHCE בביטים?
4. עבור CAHCE directly mapped בעל 4 בלוקים של 8 בייט מלא בטבלה הבאה לאלו ערכים יהיה HIT\MISS. כמו כן – עבור אלו עם HIT ציינו איזה לוקליות בא לידי ביטוי.

|  |  |  |
| --- | --- | --- |
| LOCALITY REASON | HIT\MISS | כתובת בינארית |
|  |  | 000000 |
|  |  | 010101 |
|  |  | 111111 |
|  |  | 010000 |
|  |  | 011111 |
|  |  | 000100 |
|  |  | 111111 |
|  |  | 110111 |
|  |  | 000111 |
|  |  | 111111 |
|  |  | 011100 |

|  |
| --- |
|  |

