

## מבנה מחשבים ספרתיים 83-301

### תשע"ה סמס' ב' מועד א'

|            |                                                                                                                                                                                                             |          |         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------|
| שם הקורס   | מבנה מחשבים ספרתיים                                                                                                                                                                                         |          |         |
| מספר הקורס | 83-301                                                                                                                                                                                                      |          |         |
| שם המרצה   | פרופ' שמואל וימר + מר עמית מנדלבאום                                                                                                                                                                         |          |         |
|            | תשע"ה                                                                                                                                                                                                       | סמסטר ב' | מועד א' |
| משך הבחינה | שלוש שעות                                                                                                                                                                                                   |          |         |
| חומר עזר   | שני ספר הקורס של המחברים Patterson & Hennessy + שקפי הקורס והתרגול בלבד (כולל הערות על גביהם). כל חומר אחר, כולל תרגילים, אמצעים אלקטרוניים, מחשבוני, וכו', אסורים בשימוש. יש לצרף את שאלוני הבחינה למחברת. |          |         |
|            | יש לענות על כל השאלות. כל תשובה יש לנמק ולהסביר. כל תשובה מספרית מחייבת את הצגת דרך החישוב. סה"כ הנקודות האפשריות 110. ציון הבחינה לא יעלה על 100. יש לכתוב בעט בלבד. כתיבה בעפרון לא תיבדק.                |          |         |

### בהצלחה!

#### שאלה א' (25 נק')

1. (2 נק') ידוע כי הוספת Cache כמתווך בין ה-CPU לזיכרון הראשי מגדילה את זמן קבלת המידע ישירות מהזיכרון הראשי, מדוע בכל זאת הוספת Cache מקטינה את זמן הגישה לזיכרון?  
**תשובה:** אמנם זמן קבלת המידע מהזיכרון עולה אך לאחר מכן, בזכות השימוש בעיקרון הלקאליות, מידע שהיה באותו הבלוק שהובא מהזיכרון ל-Cache נקרא שוב ללא גישות נוספות לזיכרון הראשי ולכן בסך הכל זמן הגישה לזיכרון מתקצר.  
**טעות נפוצה:** לא לאזכר את עקרון הלקאליות

2. (5 נק') נתונים שני מעבדים הפועלים באותו תדר ומריצים אותה תכנית בדיוק. במעבד הראשון זמן הגישה לזיכרון הראשי במקרה של CACHE MISS הינו 30 ננושניות, ואילו CACHE HIT אורך 3 ננושניות. ידוע ש CACHE HIT קורה ב 80% מהזמן.

במעבד השני זמן הגישה לזכרון הראשי במקרה של CACHE MISS הינו 50 ננושניות, ואילו CACHE HIT אורך 2 ננושניות. ידוע ש CACHE HIT קורה ב 90% מהזמן.

חשב בצורה מלאה את ביצועי הזכרון של שני המעבדים. לאיזה מהם ביצועים טובים יותר?

**Answer:**

In the 1<sup>st</sup> processor 80% of the time the access will be 3 ns while the rest of the time (20%) the access time will be 30 ns. The average is

$$(0.8 \times 3 \text{ ns}) + (0.2 \times 30 \text{ ns}) = 2.4 \text{ ns} + 6 \text{ ns} = 8.4 \text{ ns}$$

In the 1<sup>st</sup> processor 90% of the time the access will be 2 ns while the rest of the time (10%) the access time will be 50 ns. The average is

$$(0.9 \times 2 \text{ ns}) + (0.1 \times 50 \text{ ns}) = 1.8 \text{ ns} + 5 \text{ ns} = 6.8 \text{ ns}$$

**טעות נפוצה:** חישוב Hit-time ב- 100% מהמקרים

3. (3 נק') איזה מבין שלשת סוגי זכרון המטמון (N-WAY, FULLY ASSOCIATIVE, DIRECT, REPLACEMENT ALGORITHM) מדוע?

**תשובה:** ב-Cache מסוג Direct כל כתובת ממופה לתא אחד בלבד ב-cache ולכן כאשר יש צורך להחליף מידע בתא מסוים אין "בחירה" איזה תא להחליף ולכן לא נדרש אלגוריתם החלפה.

4. (15 נק') לפניך חלק מתוך CACHE של מעבד כלשהו. בכל אחד מהסעיפים שלהלן צרף לתשובתך הסבר מלא וחשוב מדויק.

| Tag<br>(binary values) | Set ID<br>(binary values) | Word within block |                  |                  |                  |
|------------------------|---------------------------|-------------------|------------------|------------------|------------------|
|                        |                           | 00                | 01               | 10               | 11               |
| 10100001001001         | 01101101                  | 00 <sub>16</sub>  | 61 <sub>16</sub> | C2 <sub>16</sub> | 23 <sub>16</sub> |
| 11100001100100         | 01101101                  | 10 <sub>16</sub>  | 71 <sub>16</sub> | D2 <sub>16</sub> | 33 <sub>16</sub> |
| 11001011010110         | 01101110                  | 20 <sub>16</sub>  | 81 <sub>16</sub> | E2 <sub>16</sub> | 43 <sub>16</sub> |
| 11100101101011         | 01101110                  | 30 <sub>16</sub>  | 91 <sub>16</sub> | F2 <sub>16</sub> | 53 <sub>16</sub> |
| 11110110110100         | 01101111                  | 40 <sub>16</sub>  | A1 <sub>16</sub> | 02 <sub>16</sub> | 63 <sub>16</sub> |
| 10100111010101         | 01101111                  | 50 <sub>16</sub>  | B1 <sub>16</sub> | 12 <sub>16</sub> | 73 <sub>16</sub> |
| 10101010111110         | 01110000                  | 84 <sub>16</sub>  | E5 <sub>16</sub> | 46 <sub>16</sub> | A7 <sub>16</sub> |
| 10101010010011         | 01110000                  | 94 <sub>16</sub>  | F5 <sub>16</sub> | 56 <sub>16</sub> | B7 <sub>16</sub> |
| 01110001001000         | 01110001                  | A4 <sub>16</sub>  | A5 <sub>16</sub> | 66 <sub>16</sub> | C7 <sub>16</sub> |
| 00001101101101         | 01110001                  | B4 <sub>16</sub>  | 15 <sub>16</sub> | 76 <sub>16</sub> | D7 <sub>16</sub> |
| 01011010010010         | 01110010                  | C4 <sub>16</sub>  | 25 <sub>16</sub> | 86 <sub>16</sub> | E7 <sub>16</sub> |
| 10101111001011         | 01110010                  | D4 <sub>16</sub>  | 35 <sub>16</sub> | 96 <sub>16</sub> | F7 <sub>16</sub> |

א. מאיזה סוג ה-CACHE הנ"ל? (N-WAY, FULLY ASSOCIATIVE, DIRECT)

**Answer:** It is a 2-way set associative cache, i.e., there are two lines per set.

**טעות נפוצה:** 4-way כי יש 4 בתים בבלוק

ב. מה גודל מרחב הכתובות של המעבד?

**תשובה:** ישנן 24 ביטים בכתובת סה"כ (2 + 8 + 14) לכן 2<sup>24</sup> Bytes

ג. כמה בלוקים יש במרחב הכתובות?

תשובה: בכל בלוק ישנם 4 בתים, לכן  $2^{22}$  Blocks

ד. מה גדלו של ה CACHE בביתים? (BYTES)

**Answer:** Since it is a 2-way set associative cache, and since there is an 8-bit tag, there are  $2^{8+1} = 512$  lines. Each line has 4 bytes, so the cache has a total of 2048 bytes.

**טעות נפוצה:** חישוב לפי הגודל שמופיע בטופס הבחינה (למרות שכתוב במפורש שזה רק חלק).

ה. נדרש לקרוא ממרחב הזיכרון את הבית (BYTE) הנמצא בכתובת 0x7121C5. האם ניתן לדעת מה ערכו של הנתון בכתובת הנ"ל? ובמידה וכן, מהו ערכו?

**Answer:**

First, convert 7121C5 to binary.  $7121C5_{16} = 11100010010000111000101_2$

The last two bits, 01, identify the word position within the block. 01 means that it is in the second column of data.

The next eight bits, 01110001, should identify the set. It identifies the set consisting of the third and fourth rows from the bottom.

The fourth row from the bottom has the matching tag of 0111000100100.

Therefore, the data is in the second column of the fourth row from the bottom, i.e., **A5**.

### שאלה ב' (50 נק')

מהם מצבי טבלאות ה RS, ROB, וה FP RF (FLOATING POINT REGISTER FILE) המצורפות עבור מעבד ספקולטיבי המממש את אלגוריתם TOMASULO, אשר מריץ את התכניות שלהלן, תחת ההנחות הבאות:

- רק פקודה אחת יכולה לצאת (ISSUE) בכל מחזור שעון.
- ל ROB ישנן 8 כניסות (SLOTS) והוא משמש גם בתור חוצץ (BUFFER) ל LOAD ול STORE.
- כל היחידות הפונקציונליות (FU) הינן FULLY PIPELINED.
- ישנן 2 יחידות RS לכפל (FP MULTIPLY), 3 יחידות לחבור (FP ADD) ו 3 יחידות RS לשלמים, המשמשות גם ל LOAD ו STORE.
- חריגות (EXCEPTIONS) אינן קורות.
- השיהוי בבצוע (EXECUTION) פעולות בשלמים הינו מחזור שעון אחד.
- להוציא STRUCTURAL HAZARD, פעולת LOAD יוצאת לדרך במחזור שעון אחד, מבוצעת בזה שאחריו, וכותבת בשלישי. פעולה התלויה ב LOAD יכולה להתבצע (EXECUTE) ברביעי.
- השיהוי בבצוע (EXECUTION) חבור FP הינו 2 מחזורי שעון וכפל FP 4 מחזורי שעון.
- במקרה של קונפליקט בכתיבה ל CDB, הקדימות (PRIORITY) לפקודה שיצאה לדרך מוקדם יותר.
- בצוע (EXECUTION) של פקודות תלויות יכול להתחיל במחזור השעון לאחר שנתונייהן משודרים ב CDB.
- מניחים שטבלאות ה RS, ROB, וה FP RF הינן ריקות ופנויות בעת תחילת בצוע התכניות שלהלן.
- העמודה "VALUE" מתעדכנת בעת שידור הערך ב CDB.

**דרישה חשובה בתשובות לשאלה:** בכל שינוי ממצב "BUSY" ל "NOT BUSY", נדרש כמובן לעדכן את העמודה המתאימה, אבל לא למחוק את תוכן שאר העמודות (אלא אם כן הדבר נדרש כתוצאה מפקודה אחרת התופסת מקום זה).

#### **טעויות כלליות נפוצות:**

- א. רישום שדה Value גם עבור פקודות שלא עשו Write.
  - ב. רישום שדות Q עבור פקודות שעשו (או במהלך) Execute
  - ג. מחיקת רגיסטרים מה-Register status כאשר הם לא Busy
  - ד. חוסר עדכון של שדות ה- Vj ו- Vk כאשר הערך שלהם אמור להתעדכן.
- סעיף ב' טעות נפוצה מאוד – פקודה לפני אחרונה ב- Execute במקום ב- Issue

א. (25 נקודות) רשום באופן מלא את הטבלאות המצורפות בעת השגור של הפקודה האחרונה בקוד שלהלן

|     |       |            |
|-----|-------|------------|
| Lp: | LD    | F0, 0(R1)  |
|     | LD    | F2, 0(R2)  |
|     | MULTD | F4, F0, F2 |
|     | ADDD  | F6, F0, F0 |
|     | SUBI  | R1, R1, 8  |
|     | SUBI  | R2, R2, 8  |
|     | ADDI  | R3, R3, 1  |

| Reservation stations |          |       |                |                |                |                |      |
|----------------------|----------|-------|----------------|----------------|----------------|----------------|------|
| Name                 | Busy     | Op    | V <sub>j</sub> | V <sub>k</sub> | Q <sub>j</sub> | Q <sub>k</sub> | Dest |
| Add1                 | <i>N</i> | ADDD  | Mem[R1]        | Mem[R1]        |                |                | #4   |
| Add2                 |          |       |                |                |                |                |      |
| Add3                 |          |       |                |                |                |                |      |
| Mult1                | Y        | MULTD | Mem[R1]        | Mem[R2]        |                |                | #3   |
| Mult2                |          |       |                |                |                |                |      |
| Int1                 | Y        | SUBI  | R1             | 8              |                |                | #5   |
| Int2                 | Y        | SUBI  | R2             | 8              |                |                | #6   |
| Int3                 | Y        | ADDI  | R3             | 1              |                |                | #7   |

| Reorder buffer |      |                  |         |             |                |
|----------------|------|------------------|---------|-------------|----------------|
| Entry          | Busy | Instruction      | State   | Destination | Value          |
| 1              | N    | LD F0, 0(R1)     | Commit  | F0          | Mem[0(R1)]     |
| 2              | N    | LD F2, 0(R2)     | Commit  | F2          | Mem[0(R2)]     |
| 3              | Y    | MULTD F4, F0, F2 | Execute | F4          |                |
| 4              | Y    | ADDD F6, F0, F0  | Write   | F6          | <i>F0 + F0</i> |
| 5              | Y    | SUBI R1, R1, 8   | Execute | R1          |                |
| 6              | Y    | SUBI R2, R2, 8   | Execute | R2          |                |
| 7              | Y    | ADDI R3, R3, 1   | Issue   | R3          |                |
| 8              |      |                  |         |             |                |

| FP register status |    |    |    |    |    |     |     |     |     |
|--------------------|----|----|----|----|----|-----|-----|-----|-----|
| Field              | F0 | F2 | F4 | F6 | F8 | F10 | F12 | ... | F30 |
| Reorder #          | 1  | 2  | 3  | 4  |    |     |     | ... |     |
| Busy               | N  | N  | Y  | Y  |    |     |     | ... |     |

ב. (25 נקודות) רשום באופן מלא את הטבלאות המצורפות בעת השגור של הפקודה האחרונה בקוד שלהלן

MULTD F0, F2, F4  
 ADDD F6, F6, F0  
 ADDD F2, F2, F8  
 LD F4, 0(R2)  
 ADDI R1, R1, 8  
 MULTD F8, F10, F12  
 ADDD F4, F4, F10  
 ADDI R2, R2, 1

| Reservation stations |      |       |       |       |    |              |      |
|----------------------|------|-------|-------|-------|----|--------------|------|
| Name                 | Busy | Op    | Vj    | Vk    | Qj | Qk           | Dest |
| Add1                 | Y    | ADDD  | F6    | F2xF4 |    | <i>blank</i> | #2   |
| Add2                 | N    | ADDD  | F2    | F8    |    |              | #3   |
| Add3                 | Y    | ADDD  | $F_4$ | F10   |    |              | #7   |
| Mult1                | N    | MULTD | F2    | F4    |    |              | #1   |
| Mult2                | Y    | MULTD | F10   | F12   |    |              | #6   |
| Int1                 | N    | LD    | R2    | 0     |    |              | #4   |
| Int2                 | Y    | ADDI  | R1    | 8     |    |              | #5   |
| Int3                 | Y    | ADDI  | R2    | 1     |    |              | #8   |

| Reorder buffer |      |                     |                |             |              |
|----------------|------|---------------------|----------------|-------------|--------------|
| Entry          | Busy | Instruction         | State          | Destination | Value        |
| 1              | N    | MULTD F0, F2, F4    | <i>Commit</i>  | F0          | F2×F4        |
| 2              | Y    | ADDD F6, F6, F0     | <i>Execute</i> | F6          |              |
| 3              | Y    | ADDD F2, F2, F8     | <i>Write</i>   | F2          | $F2+F8$      |
| 4              | Y    | LD F4, 0(R2)        | <i>Write</i>   | F4          | $Mem[0(R2)]$ |
| 5              | Y    | ADDI R1, R1, 8      | <i>Execute</i> | R1          |              |
| 6              | Y    | MULTD F8, F10, F12  | <i>Execute</i> | F8          |              |
| 7              | Y    | ADDD F4 Mem[R2] F10 | <i>Issue</i>   | $F_4$       |              |
| 8              | Y    | ADDI R2, R2, 1      | <i>Issue</i>   | R2          |              |



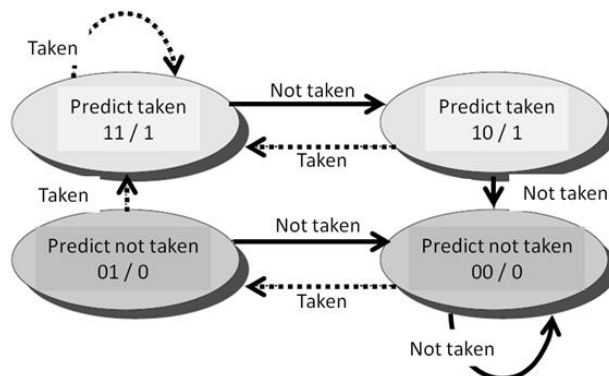
| FP register status |    |    |    |    |    |     |     |     |     |
|--------------------|----|----|----|----|----|-----|-----|-----|-----|
| Field              | F0 | F2 | F4 | F6 | F8 | F10 | F12 | ... | F30 |
| Reorder #          | 1  | 3  | 7  | 2  | 6  |     |     | ... |     |
| Busy               | N  | Y  | Y  | Y  | Y  |     |     | ... |     |

## שאלה ג' (35 נקודות)

נתונה התכנית שלהלן

```
loop:
    LW    R4, 0(R3)
    ADDI  R3, R3, 4
    SUBI  R1, R1, 1
b1:
    BEQZ  R4, b2
    ADDI  R2, R2, 1
b2:
    BNEZ  R1, loop
```

נתונים ערכי ההתחלה הבאים:  $R1 = n$ ,  $R2 = 0$ ,  $R3 = p$  (מצביע לראש מערך של מספרים שלמים). ברשותנו מעבד MIPS בעל מנגנון חיזוי קפיצות המתואר להלן. הניחו ש  $b1$  ו  $b2$  אינם מתנגשים בטבלת היסטורית הקפיצות.



א. (5 נקודות) מה בדיוק מבצעת התכנית הנ"ל? מהו ערכו של  $R2$  בעת היציאה מהלולאה?

**תשובה:** התוכנית מקבלת מערך  $P$  וסופרת כמה מבין  $n$  האיברים הראשונים בו שונים מ-0.

בסיום התוכנית  $R2$  מכיל את כמות האיברים שונים מ-0

**טעות נפוצה:** לא מתייחסים ל- $n$

ב. (10 נקודות) נניח שהקלט לתכנית הינו  $n = 8$  ו  $p[0] = 1$ ,  $p[1] = 0$ ,  $p[2] = 1$ ,  $p[3] = 0, \dots$

**מלא את טבלה א' המצורפת במלואה.** העמודה PC שבתבלה מציינת את

הפקודה בה נמצא ה PC. העמודות  $b1$  ו  $b2$  מציינות את מצב מכוונות המצבים המתאימות,  $N$  מצוין NOT TAKEN, ואילו T מצוין TAKEN. מצב מכוונות המצבים בעת קבלת הניחוש

מצוין במודגש, ואילו המצב הבא מצוין באלכסון. **מהו מספר חיזויי הקפיצה השגויים?**

**תשובה:** (ראה טבלה ליד הטבלה המקורית) מספר החיזויים השגויים : 7

**טעות נפוצה (גם בסעיף ג') לא שמים לב ש- $R1$  יורד ב-1 לפני הקפיצות ולכן ממלאים את 2**

**השורות האחרונות בטבלה (למרות שאמורות להישאר ריקות)**

[illegible]

| System state |       | Predictor |         | Branch behavior |        |
|--------------|-------|-----------|---------|-----------------|--------|
| PC           | R3/R4 | B1 bits   | B2 bits | Predicted       | Actual |
| B1           | 4/1   | 00        | 00      | N               | N      |
| B2           | 4/1   | 00        | 00      | N               | T      |
| B1           | 8/0   | 00        | 01      | N               | T      |
| B2           | 8/0   | 01        | 01      | N               | T      |
| B1           | 12/1  | 01        | 11      | N               | N      |
| B2           | 12/1  | 00        | 11      | T               | T      |
| B1           | 16/0  | 00        | 11      | N               | T      |
| B2           | 16/0  | 01        | 11      | T               | T      |
| B1           | 20/1  | 01        | 11      | N               | N      |
| B2           | 20/1  | 00        | 11      | T               | T      |
| B1           | 24/0  | 00        | 11      | N               | T      |
| B2           | 24/0  | 01        | 11      | T               | T      |
| B1           | 28/1  | 01        | 11      | N               | N      |
| B2           | 28/1  | 00        | 11      | T               | T      |
| B1           | 32/0  | 00        | 11      | N               | T      |
| B2           | 32/0  | 01        | 11      | T               | N      |

טבלה ב'

| System State |       |             | Branch Predictor |       |         |       | Behavior  |        |
|--------------|-------|-------------|------------------|-------|---------|-------|-----------|--------|
| PC           | R3/R4 | history bit | b1 bits          |       | b2 bits |       | Predicted | Actual |
|              |       |             | set 0            | set 1 | set 0   | set 1 |           |        |
| b1           | 4/1   | 0           | 00               | 00    | 00      | 00    | N         | N      |
| b2           | 4/1   | 1           | 00               | 00    | 00      | 00    | N         | T      |
| b1           | 8/0   | 0           | 00               | 00    | 00      | 01    |           |        |
| b2           | 8/0   |             |                  |       |         |       |           |        |
| b1           | 12/1  |             |                  |       |         |       |           |        |
| b2           | 12/1  |             |                  |       |         |       |           |        |
| b1           |       |             |                  |       |         |       |           |        |
| b2           |       |             |                  |       |         |       |           |        |
| b1           |       |             |                  |       |         |       |           |        |
| b2           |       |             |                  |       |         |       |           |        |
| b1           |       |             |                  |       |         |       |           |        |
| b2           |       |             |                  |       |         |       |           |        |
| b1           |       |             |                  |       |         |       |           |        |
| b2           |       |             |                  |       |         |       |           |        |
| b1           |       |             |                  |       |         |       |           |        |
| b2           |       |             |                  |       |         |       |           |        |
| b1           |       |             |                  |       |         |       |           |        |
| b2           |       |             |                  |       |         |       |           |        |
| b1           |       |             |                  |       |         |       |           |        |
| b2           |       |             |                  |       |         |       |           |        |

| System state |       | History bit | Predictor |    |         |    | Branch behavior |        |
|--------------|-------|-------------|-----------|----|---------|----|-----------------|--------|
| PC           | R3/R4 |             | B1 bits   |    | B2 bits |    | Predicted       | Actual |
| B1           | 4/1   | 0           | 00        | 00 | 00      | 00 | N               | N      |
| B2           | 4/1   | 1           | 00        | 00 | 00      | 00 | N               | T      |
| B1           | 8/0   | 0           | 00        | 00 | 00      | 01 | N               | T      |
| B2           | 8/0   | 1           | 01        | 00 | 00      | 01 | N               | T      |
| B1           | 12/1  | 1           | 01        | 00 | 00      | 11 | N               | N      |
| B2           | 12/1  | 1           | 01        | 00 | 00      | 11 | T               | T      |
| B1           | 16/0  | 0           | 01        | 00 | 00      | 11 | N               | T      |
| B2           | 16/0  | 1           | 11        | 00 | 00      | 11 | T               | T      |
| B1           | 20/1  | 1           | 11        | 00 | 00      | 11 | N               | N      |
| B2           | 20/1  | 1           | 11        | 00 | 00      | 11 | T               | T      |
| B1           | 24/0  | 0           | 11        | 00 | 00      | 11 | T               | T      |
| B2           | 24/0  | 1           | 11        | 00 | 00      | 11 | T               | T      |
| B1           | 28/1  | 1           | 11        | 00 | 00      | 11 | N               | N      |
| B2           | 28/1  | 1           | 11        | 00 | 00      | 11 | T               | T      |
| B1           | 32/0  | 0           | 11        | 00 | 00      | 11 | T               | T      |
| B2           | 32/0  | 1           | 11        | 00 | 00      | 11 | T               | N      |