

מבנה מחשבים - תרגיל #6

אריאל סטורמן

קבוצה 02

(1)

אין כל צורך להוסיף / לשנות דבר. להלן הסבר ל-*control* עבור הפקודה *lwR*:

- *RegDst*: יהיה 1, רוצים לכתוב לרגיסטר *rd* לכן נרצה לקבלו.
- *Branch*: יהיה 0, כי לא מתבצע *branch*.
- *MemRead*: יהיה 1, קוראים מהזיכרון את תוכן הכתובת $rt + rs$.
- *MemtoReg*: יהיה 1, כותבים מהזיכרון לרגיסטר *rd*.
- *ALUOp*: יהיה 01, רוצים לבצע פעולת חיבור בין הרגיסטרים $rs, rt \Leftarrow$ *funct*: יהיה *xx0000*, לביצוע פעולת חיבור.
- *MemWrite*: יהיה 0, לא כותבים דבר לזיכרון.
- *ALUSrc*: יהיה 0, רוצים לחבר שני רגיסטרים rs, rt .
- *RegWrite*: יהיה 1, רוצים לכתוב לרגיסטר *rd*.

ההגדרות הנ"ל יביאו לביצוע הפקודה הרצויה: קריאת rs, rt , חיבורם, קריאת תוכן הכתובת שהתקבלה מהזיכרון והשמה ברגיסטר *rd*.

(2)

נכניס את פקודת ה-*jr* באופן הבא:

- נוסיף קו בקרה *jr* שיקבל 1 עבור פקודת *jr* ו-0 אחרת.
- את ה-*MUX* היוצא אל ה-*PC* נעביר דרך *MUX* חדש: כניסת ה-0 תהיה זו שחושבה, וכניסת ה-1 תהיה קו חדש היוצא מ-1 *Read data*, הוא הרגיסטר המכיל את הכתובת הרצויה (כתובת אבסולוטית, ללא צורך ב-*shift left 2*). כך, אם הפעולה היא *jr* נקפוץ אל הכתובת הרצויה ב-*PC*, ואם לא נמשיך לפי הפקודה המבוצעת.

Instruction	RegDst	ALUSrc	Memto-Reg	Reg-Write	Mem Read	Mem Write	Branch	ALUOps1	ALUOp0	JR
R-format	1	0	0	1	0	0	0	1	0	0
lw	0	1	1	1	1	0	0	0	0	0
sw	X	1	X	0	0	1	0	0	0	0
Beq	X	0	X	0	0	0	1	0	1	0
jr	X	X	X	0	X	0	X	X	X	1

הסבר :

כאשר קופצים לכתובת לפי הרגיסטר הראשון (1 *read data*), לא משנה מה נשלח לשאר קווי הבקרה, פרט לאלו המשנים את המצב במערכת, שהם *MemWrite*, *RegWrite*. אליהם נשלח 0 כדי שלא נשנה את מצב המערכת. כיוון שחיברנו את ה-*MUX* – *jr* אחרי ה-*MUX* האחרון המחשב את ה-*PC*, אז גם ה-*Branch* לא משנה.

(3)

א. כאשר הפקודה הראשונה ב-*MEM* והפקודה השנייה נמצאת ב-*EX*: ה-*forwarding unit* תעביר 0 לשני ה-*MUX*, היות והשינוי של רגיסטר \$2 בפקודה הראשונה אינו משפיע על הפקודה השנייה.
 ב. כאשר הפקודה הראשונה ב-*WB*, השנייה ב-*MEM* והשלישית ב-*EX*: הפקודה השלישית משתמשת בערך של רגיסטר \$4, המתעדכן בפקודה השנייה. לכן ה-*forwarding unit* תעביר 1 ל-*MUX* המתאים על מנת שהפקודה השלישית תתבצע עם הערך המעודכן של \$4.

(4)

```
add $2, $5, $4
add $4, $2, $5
sw $5, 100($2)
add $3, $2, $4
```

להלן פירוט הבעיות, כאשר כולן ייפתרו ע"י *forwarding*:

א. הפקודה הראשונה ב-*MEM*, הפקודה השנייה ב-*EX*: בפקודה הראשונה כותבים לרגיסטר \$2, אך בפקודה השנייה משתמשים בו.
 ב. הפקודה הראשונה ב-*WB*, הפקודה השנייה ב-*MEM*, הפקודה השלישית ב-*EX*: בפקודה הראשונה כותבים לרגיסטר \$2, אך בפקודה השלישית משתמשים בו.
 ג. הפקודה השנייה ב-*WB*, הפקודה השלישית ב-*MEM*, הפקודה הרביעית ב-*EX*: בפקודה השנייה כותבים לרגיסטר \$4, אך בפקודה הרביעית משתמשים בו.

(5)

הרגיסטרים הנקראים הם \$1, \$6, והנכתבים הם \$2.