

## תרגול 9

### Single Cycle Architecture

אלון שקלר - אוניברסיטת תל אביב

1

## מימוש פקודות - התהליך

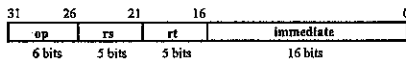
- ראשית, מתארים מה הפקודה עושה בשלבים:
  - מעבר נתונים (מקורות ועד) תוך שימת דגש על האופרנדים - זיכרון/רגיסטרים
  - פעולות אריתמטיות שנעשות
  - שנית, מממשים
  - מימוש מעבר הנתונים במערכת הקיימת:
    - האם הקווים הקיימים מספקים או צריך להוסיף קווים נוספים
  - מימוש הבקרה במערכת הקיימת:
    - קביעת ערכי הבקרה ביחידות הבקרה הקיימות
    - הוספת אטומים במידת הצורך והגדרת קווי הבקרה שלהם
    - עדכון פונקציית קווי הבקרה הקיימים על שילוב ערכי קווי הבקרה של הפקודה החדשה

אלון שקלר - אוניברסיטת תל אביב

2

## דוגמה - הפקודה addi

- addi \$t1, \$t2, immediate



תיאור:

- פקודה נלקחת מזיכרון הפקודות ורגיסטר ה-PC מקודם
- הרגיסטר \$t2 (rs) נקרא ממאגר הרגיסטרים (על-פי סיביות 21-25)
- immediate עובר הרחבה ל-32 סיביות ע"י מרחת הסימן
- ה-ALU מחשב את סכום הערך שנקרא ממאגר הרגיסטרים ו-immediate מורחב הסימן.
- תוצאת החישוב נכתבת אל מאגר הרגיסטרים לרגיסטר שמספרו נתון בסיביות 16-20 (rt).

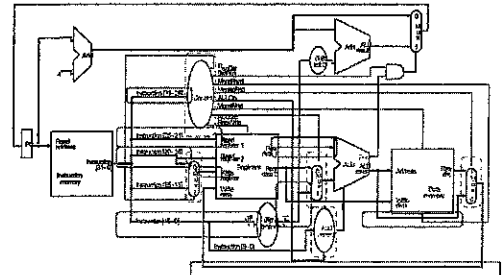
דמיון רב ל-lw אך ללא הגישה למאגר הזיכרון  
אלון שקלר - אוניברסיטת תל אביב

3

Figures taken from the book: "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson, John L. Hennessy, and Daniel A. Dreyer, Morgan Kaufmann Publishers Inc. - All rights reserved.

## addi

- האם קווי הנתונים הקיימים מספקים?
- האם יחידות הבקרה מספקות?



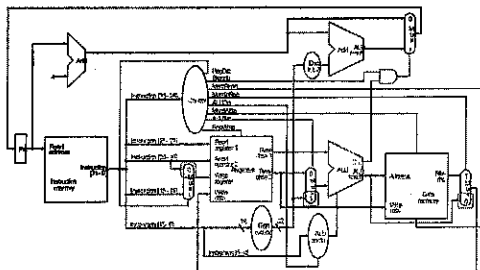
אלון שקלר - אוניברסיטת תל אביב

4

Figures taken from the book: "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson, John L. Hennessy, and Daniel A. Dreyer, Morgan Kaufmann Publishers Inc. - All rights reserved.

## קביעת ערכי הבקרה

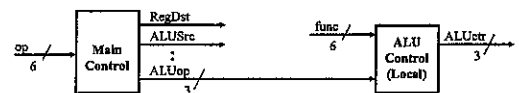
Instruction	RegDst	ALUSrc	MemWr	Branch	ALUOp	ALUOp
R-format	1	0	0	1	0	1
lw	0	1	1	1	0	0
addi	0	1	0	1	x	0



אלון שקלר - אוניברסיטת תל אביב

5

## עדכון פונקציית קווי הבקרה הקיימים



	000000	001001	100011	101011	000100	000010
	R-type	addi	lw	sw	beq	jump
RegDst	1	0	0	x	x	x
ALUSrc	0	1	1	1	0	x
MemWr	0	0	1	x	x	x
MemWrite	0	1	0	0	0	0
MemWrite	0	0	0	1	0	0
Branch	0	0	0	0	1	0
Jump	0	0	0	0	0	1
ExtOp	x	0	1	1	x	x
ALUOp (Symbolic)	"R-type"	Add	Add	Add	Subtract	xxx
ALUOp <1>	1	0	0	0	0	x
ALUOp <0>	0	0	0	0	1	x

אלון שקלר - אוניברסיטת תל אביב

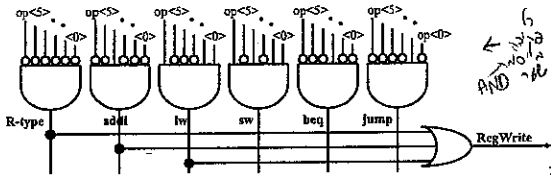
6

Figures taken from the book: "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson, John L. Hennessy and John L. Hennessy, Morgan Kaufmann Publishers Inc. - All rights reserved

### Implementation of RegWrite

op	00 0000	00 1000	10 0011	10 1011	00 0100	00 0010
	R-type	addi	lw	sw	beq	jump
RegWrite	1	1	1	0	0	0

RegWrite = R-type + addi + lw  
 =  $op\langle 5 \rangle \& \text{op}\langle 4 \rangle \& \text{op}\langle 3 \rangle \& \text{op}\langle 2 \rangle \& \text{op}\langle 1 \rangle \& \text{op}\langle 0 \rangle$  (R-type)  
 +  $op\langle 5 \rangle \& \text{op}\langle 4 \rangle \& \text{op}\langle 3 \rangle \& \text{op}\langle 2 \rangle \& \text{op}\langle 1 \rangle \& \text{op}\langle 0 \rangle$  (addi)  
 +  $op\langle 5 \rangle \& \text{op}\langle 4 \rangle \& \text{op}\langle 3 \rangle \& \text{op}\langle 2 \rangle \& \text{op}\langle 1 \rangle \& \text{op}\langle 0 \rangle$  (lw)



אלון שקלר - אוניברסיטת תל אביב

0000 10

op | address

### פקודת ה-jump

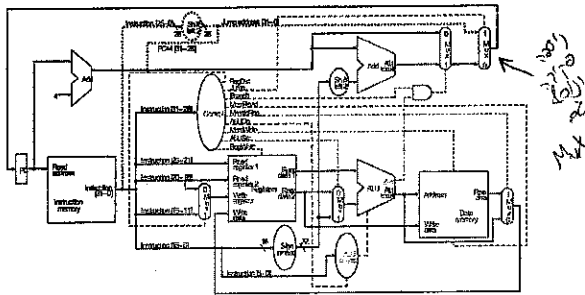
פירוש הפקודה: J 0101...10111011  
 הפקודה הבאה נמצאת בכתובת 0101...10111011

4 bits	26 bits	2 bits	
	101 ... 101111011		כתובת קפיצה במילים:
	101 ... 101111011	00	כתובת קפיצה בביטים:
0101	101 ... 101111011		תוספת 4 ביטים אחרונים:
0101	101 ... 101111011	00	הקפיצה הסופית:

אלון שקלר - אוניברסיטת תל אביב

Figures taken from the book: "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson, John L. Hennessy and John L. Hennessy, Morgan Kaufmann Publishers Inc. - All rights reserved

### Jump

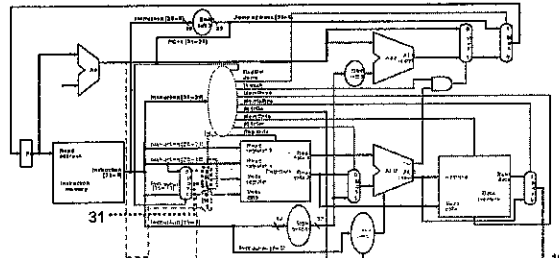


אלון שקלר - אוניברסיטת תל אביב

Figures taken from the book: "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson, John L. Hennessy and John L. Hennessy, Morgan Kaufmann Publishers Inc. - All rights reserved

### jal

הפקודה מורכבת משני חלקים:  
 - הכנסת PC+4 לאוגר \$ra שמספרו 31  
 - פעולת jump לכתובת שבפקודה



אלון שקלר - אוניברסיטת תל אביב

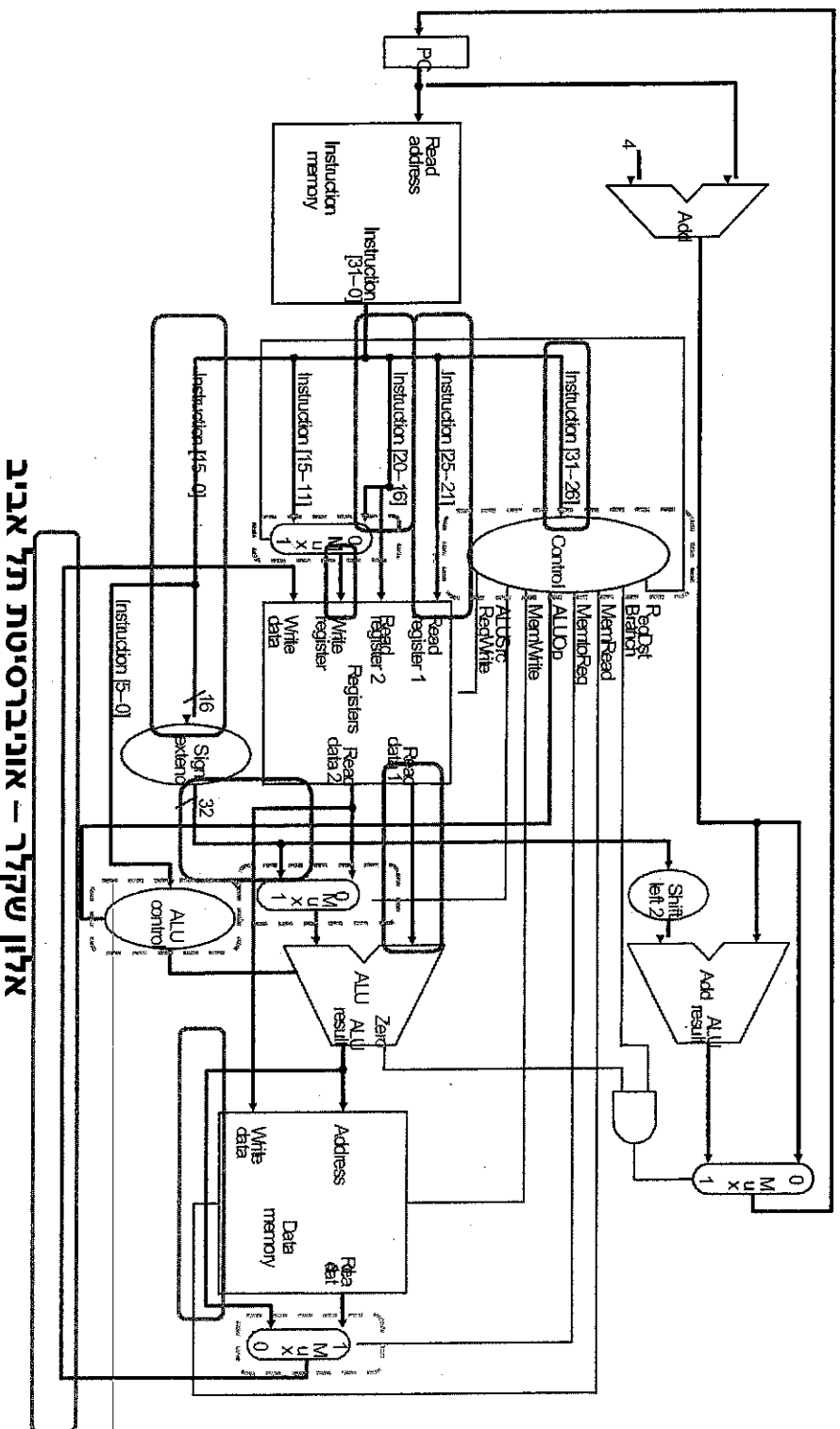
### עדכון קווי הבקרה והוספת קו חדש

op	00 0000	00 1000	10 0011	10 1011	00 0100	00 0010	00 0011
	R-type	addi	lw	sw	beq	jump	jal
RegDst	1	0	0	x	x	x	x
ALUSrc	0	1	1	1	0	x	x
MemtoReg	0	0	1	x	x	x	x
RegWrite	1	1	1	0	0	0	1
MemWrite	0	0	0	1	0	0	0
Branch	0	0	0	0	1	x	x
Jump	0	0	0	0	0	1	1
ExtOp	x	0	1	1	x	x	x
ALUSrc (Symbolic)	"R-type"	Add	Add	Subtract	xxx	xxx	xxx
ALUSrc <1>	1	0	0	0	0	x	x
ALUSrc <0>	0	0	0	0	1	x	x
jal	0	0	0	0	0	0	1

אלון שקלר - אוניברסיטת תל אביב

**addi**

- האם קווי המתונים הקיימים מספקים **[P]**
- האם יחידות הבקרה מספקות **[P]**



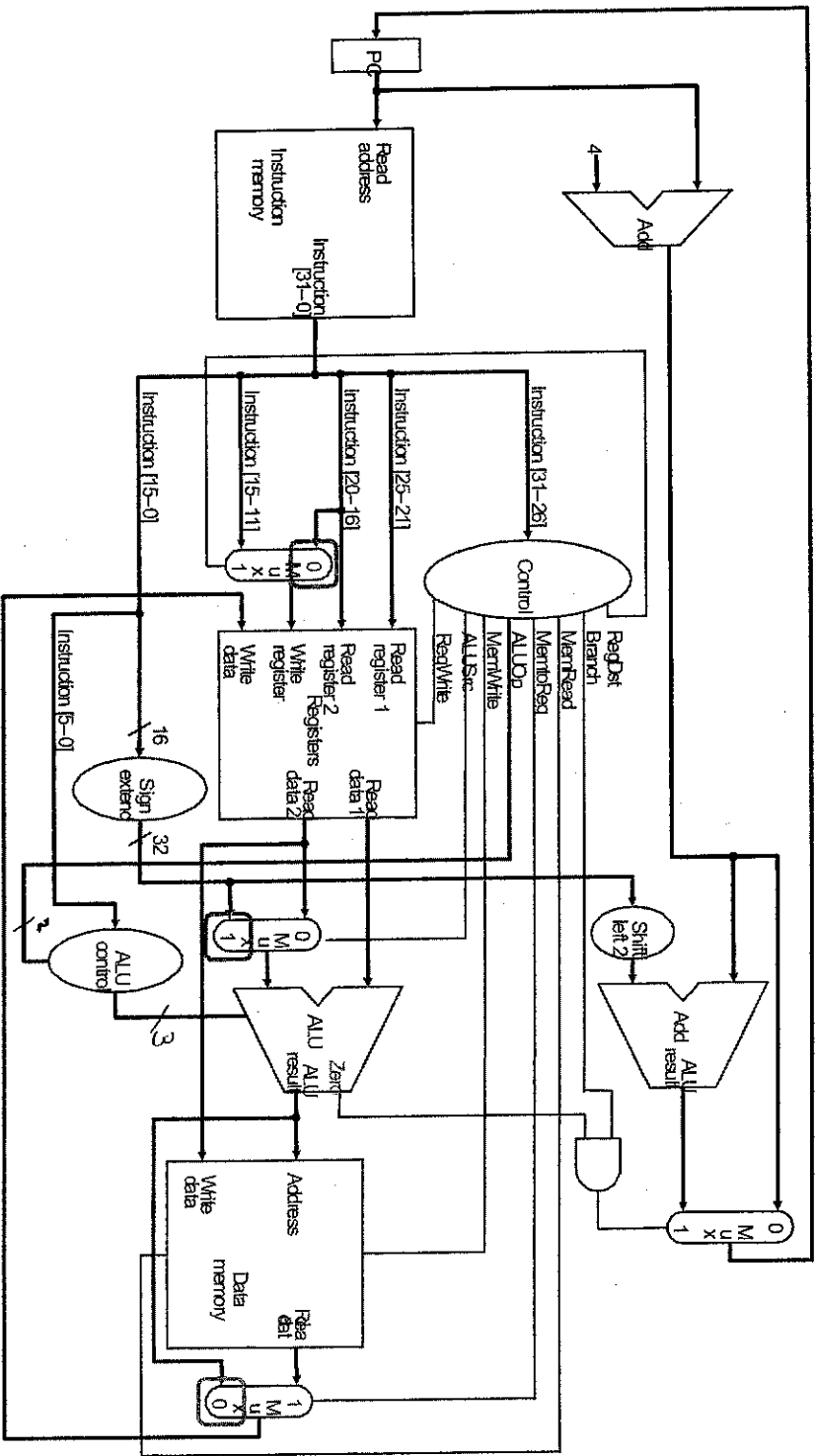
Rt/Re

SI: 26 25:21 20:16 15:11 10:6 5:0  
 OR VS r6 vd Shift funct | S:0

Figures taken from the book: "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson, John L. Hennessy and John L. Hennessy. Morgan Kaufmann Publishers Inc. - All rights reserved

# קביעת ערכי הביקורת

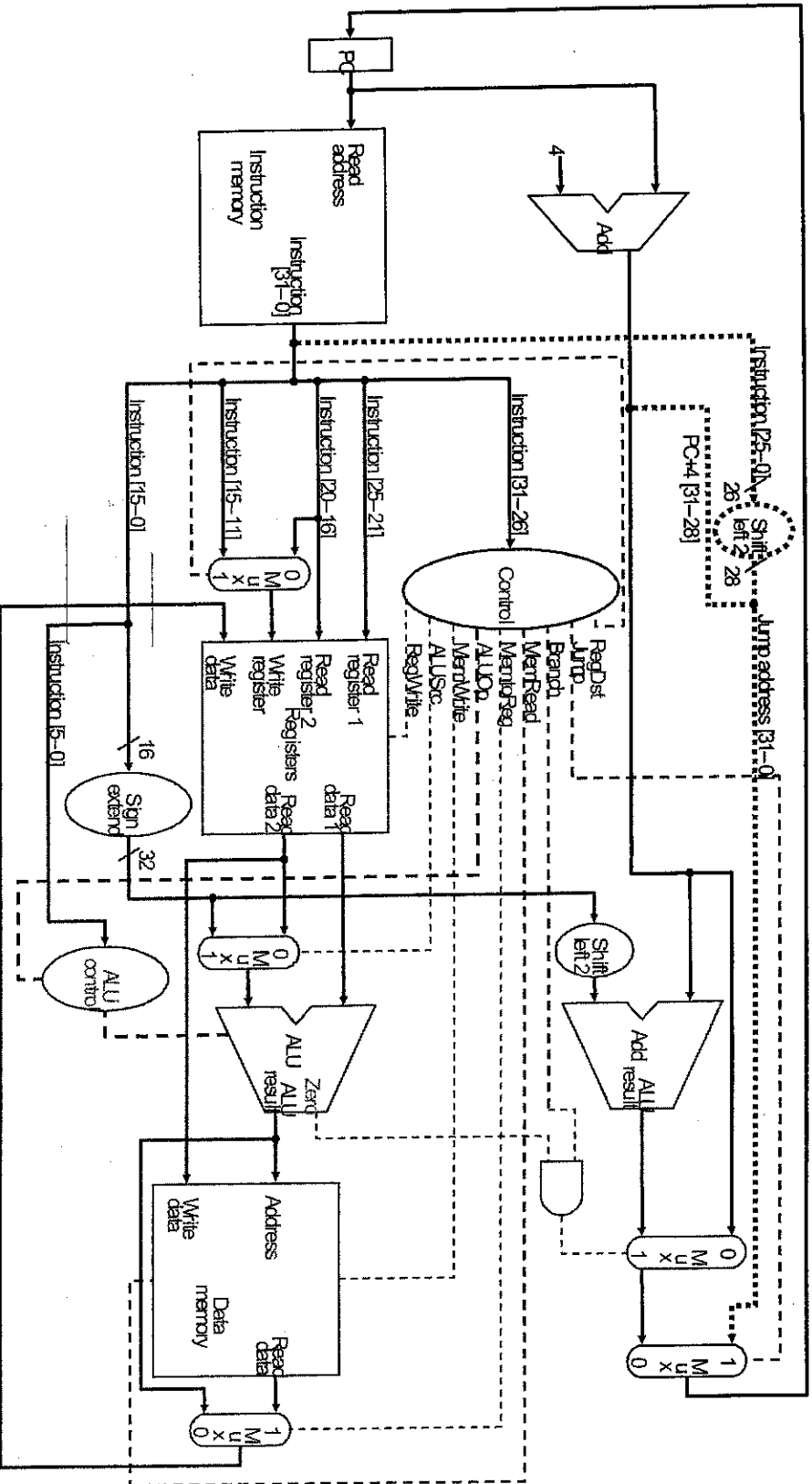
Instruction	RegDst	ALUSrc	RegWrite	MemRead	MemWrite	Branch	MemRead	MemWrite	ALUSrc	ALUOp	ALUOp
R-format	1	0	0	1	0	0	0	0	0	1	0
Lw	0	1	1	1	0	0	0	0	0	0	0
addi	0	1	0	1	x	0	0	0	0	0	0



אלון שקלר - אוניברסיטת תל אביב

Figures taken from the book: "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson, John L. Hennessy and John L. Hennessy, Morgan Kaufmann Publishers Inc. – All rights reserved

# Jump



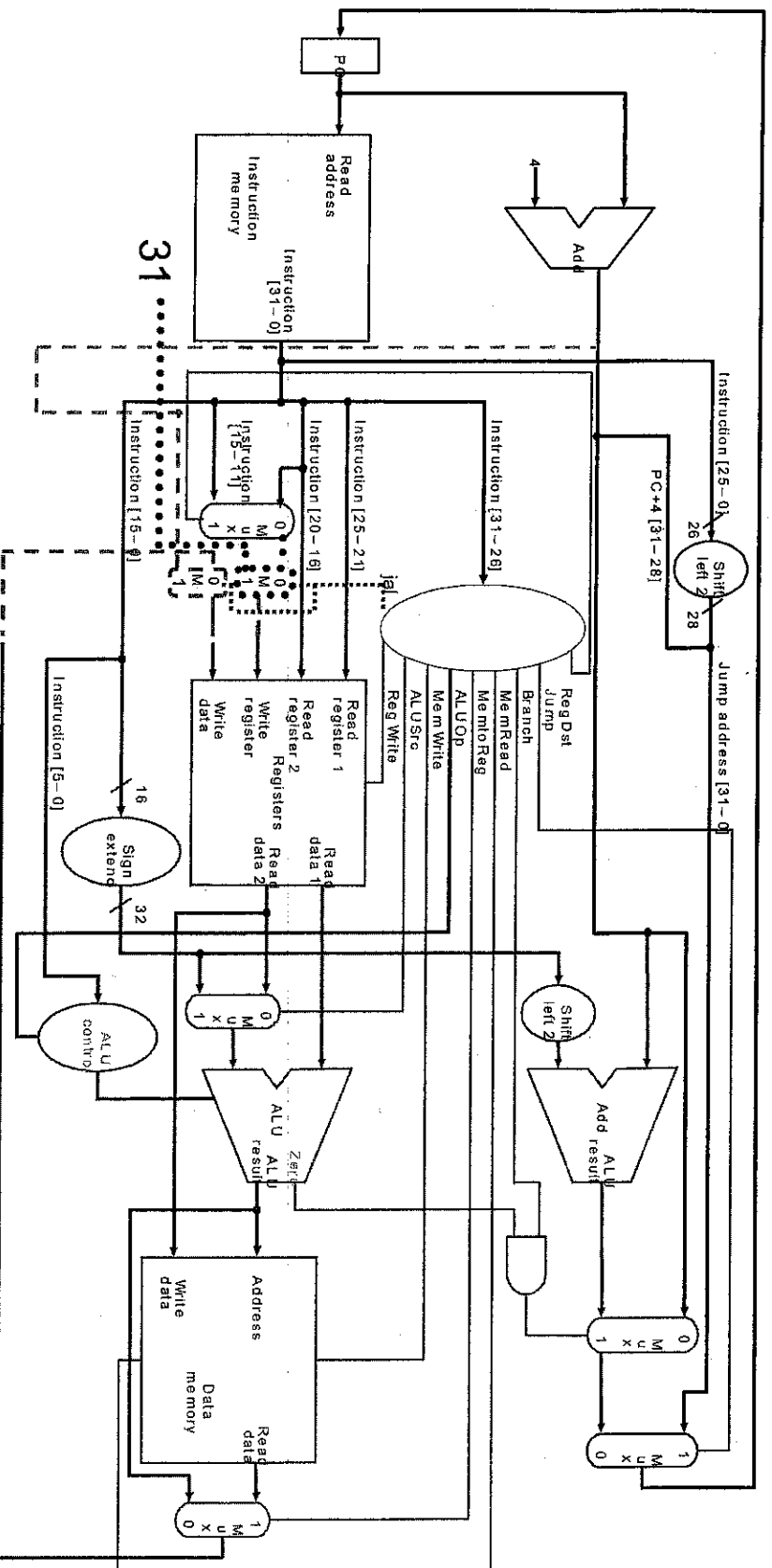
Figures taken from the book: "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson, John L. Hennessy and John L. Hennessy. Morgan Kaufmann Publishers Inc. – All rights reserved

jal

• הפקודה מורכבת משני חלקים:

– הכנסת PC+4 לאוגר \$ra שמספרו 31

– פעולת קמון לכתובת שבפקודה



אלון שקלר – אוניברסיטת תל אביב