

תרגול 13

Pipelining

אלון שקלר - אוניברסיטת תל אביב

איך Pipelining ?

- מחלקים כל פקודה לשלבי עבודה כאשר הפקודה ה"ארוכה" ביותר קובעת את מספר השלבים הכללי (IF, ID, EX, MEM, WB שלבים).
- היות ויש פקודות אשר זקוקות למידע שלהן (כמו רגיסטר יעד) עד סוף ביצוע יש לזכור מידע זה.
- מידע זה נשמר ברגיסטרים מיוחדים אשר שומרים בכל שלב את המידע שנחזק לשלב הבא.

אלון שקלר - אוניברסיטת תל אביב

למה Pipelining ?

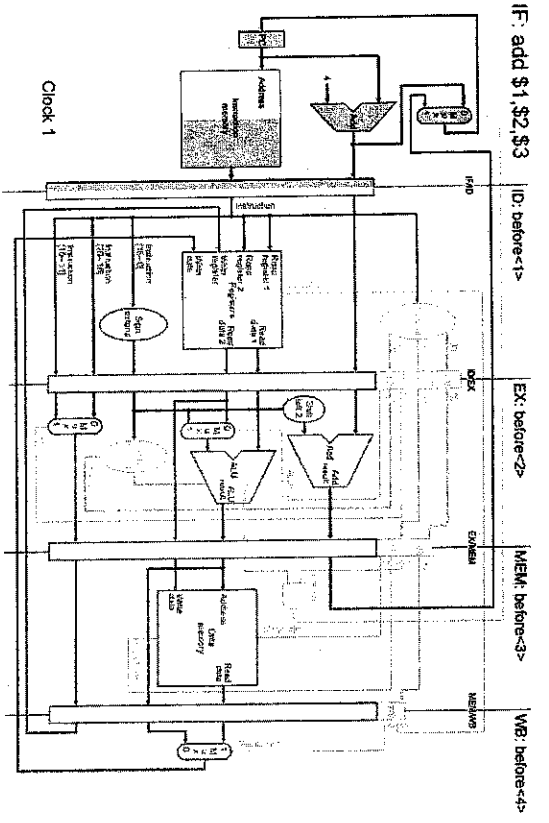
- נרצה שכל רכיב לא יישאר ללא עבודה במהלך מחזור השעון.
- בהנחה שאנו עובדים ב-single cycle לדוגמא, ה-ALU פעיל רק בחלק קטן ממחזור השעון. נרצה לנצל לטובת פקודות אחרות
- הפתרון: Pipelining - עבודה במקביל על מספר פקודות כאשר רכיב שסיים עבודתו במסגרת פקודה אחת יכול להשתתף בביצוע הפקודה הבאה.

אלון שקלר - אוניברסיטת תל אביב

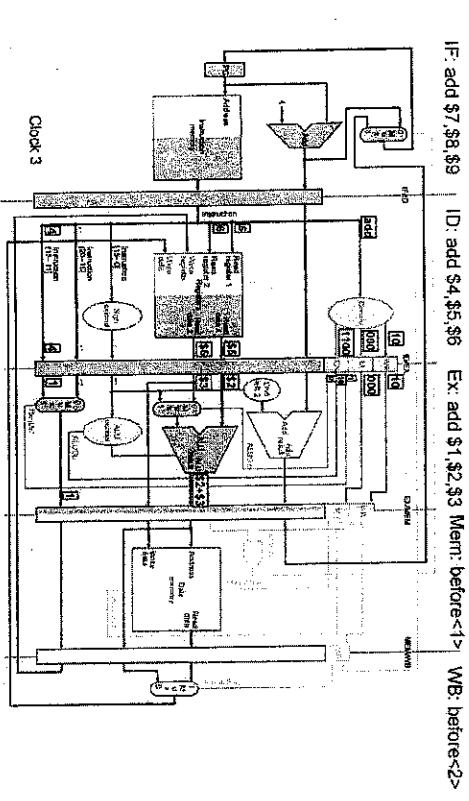
דוגמא

- נתון הקוד הבא:
- add \$1, \$2, \$3
- add \$4, \$5, \$6
- add \$7, \$8, \$9
- add \$10, \$11, \$12
- add \$13, \$14, \$15
- בסוף ה-cycle החמישי אילו רגיסטרים נכתבים ואילו נקראים ?

אלון שקלר - אוניברסיטת תל אביב

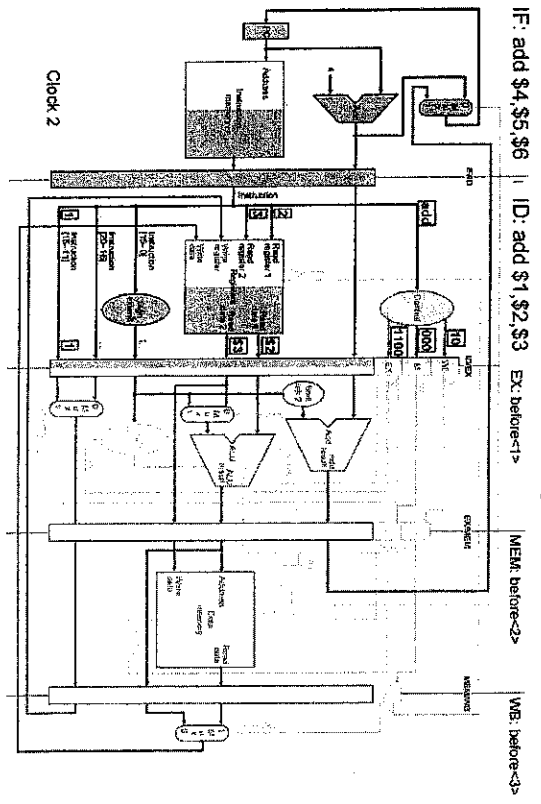


אלו שקלר - אוברטטת תל אביב

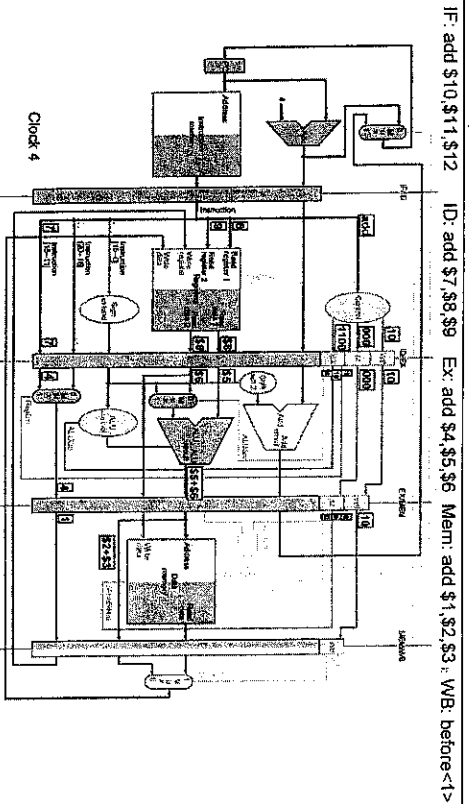


אלו שקלר - אוברטטת תל אביב

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



אלו שקלר - אוברטטת תל אביב

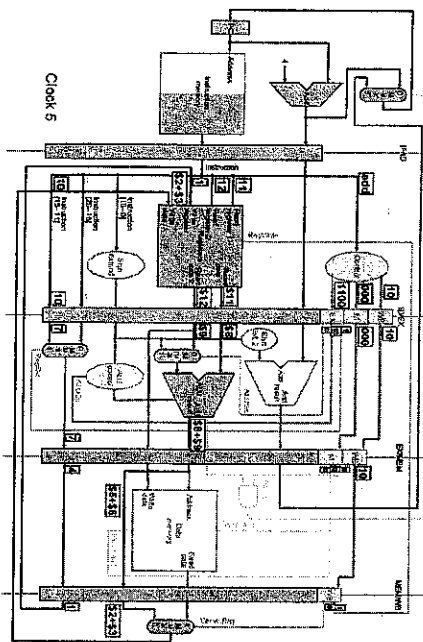


אלו שקלר - אוברטטת תל אביב

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

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

IF: add \$13,\$14,\$15 ID: add \$10,\$11,\$12 EX: add \$7,\$8,\$9 MEM: add \$4,\$5,\$6 WB: add \$1,\$2,\$3



אלון שקלר - אוניברסיטת תל אביב

המשך דוגמא

- בהתייחס לתכנית הקודמת: מה תעשה יחידת ה- forwarding במהלך ה-סאקס החמישי. אילו השוואות היא תבצע?

פתרון:

- היחידה בודקת ב-סאקס החמישי האם עליה לבצע forwarding ע"י בדיקה הפקודות הרביעית והחמישית אם הן מתכוונות לכתוב למאגר הרגיסטרים והאם הרגיסטר שייכתב משמש כקליט ל-ALU מהפקודה השלישית.
- ההשוואות שתבצענה הן:
 - $9 \neq 24$
 - $9 \neq 21$
 - $8 \neq 24$
 - $8 \neq 21$

אלון שקלר - אוניברסיטת תל אביב

ערכי הבקרה עבור כל שלב

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

Instruction	Execution/Address Calculation stage Control lines			Memory access stage Control lines			Write-back stage Control lines		
	Reg Dest	ALU Op1	ALU Op0	ALU Src	Branch	Mem Read	Mem Write	Reg Write	Mem to Reg
R-format	1	1	0	0	0	0	0	1	0
Lw	0	0	0	1	0	1	0	1	1
Sw	X	0	0	0	1	0	1	0	X
Beq	X	0	0	1	0	0	0	0	X

אלון שקלר - אוניברסיטת תל אביב

המשך דוגמא

- בהתייחס לתכנית הקודמת: מה תעשה יחידת ה- hazard detection במהלך ה-סאקס החמישי. אילו השוואות היא תבצע?

פתרון:

- היחידה בודקת ב-סאקס החמישי האם הפקודה שמתבצעת באותו הרגע ב-ALU היא SA והאם הפקודה שבשלב ה-ID קוראת מהרגיסטר שאליו הפקודה ה-SA כותבת.
- אם כן יש צורך ב-stall.
- אם הייתה הפקודה השלישית SA אז היחידה הייתה בודקת אם הרגיסטר היעד שלה הוא 11 או 12.

אלון שקלר - אוניברסיטת תל אביב

דוגמא נוספת

נתונות הפקודות הבאות

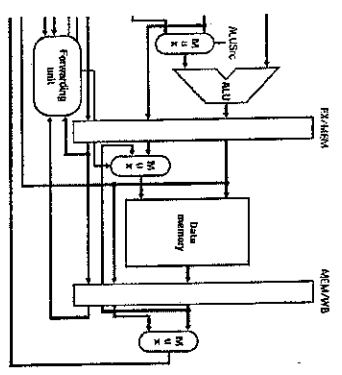
```
lw $2, 100($5)
sw $2, 200($6)
```

אילו שניונים בחומרה datapath יש להכניס על-מנת ששתי פקודות אלו תעבודנה ללא stall ?

שני פתרונות:

- פתרון א - בודק את השילוב lw-sw כאשר פקודת ה-lw נמצאת בשלב ה-MEM ופקודת ה-sw נמצאת בשלב ה-EX.
 - פתרון ב - בודק את השילוב lw-sw כאשר פקודת ה-lw נמצאת בשלב ה-WB ופקודת ה-sw נמצאת בשלב ה-MEM.
- אלון שקלר - אוניברסיטת תל אביב

פתרון ב



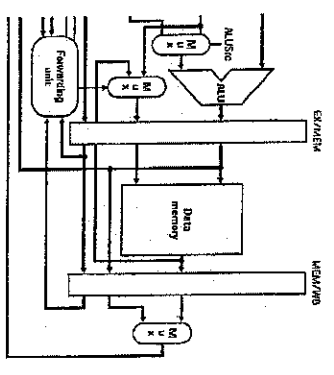
Solution is courtesy of Alex Acipolis - Textbook Sales Representative - ELSEVIER SCIENCE

LW is identified differently because the loss of the MEM control signals

```
if ($2 < 0) sw; else lw;
// same register?
// but not if
// forward by value
// do not forward
```

Actual instruction and not a FLUSH

פתרון א



Solution is courtesy of Alex Acipolis - Textbook Sales Representative - ELSEVIER SCIENCE

```
if ($2 < 0) sw; else lw;
// sw in EX stage?
// lw in MEM stage?
// same register?
// but not if
// forward by value
// do not forward
```

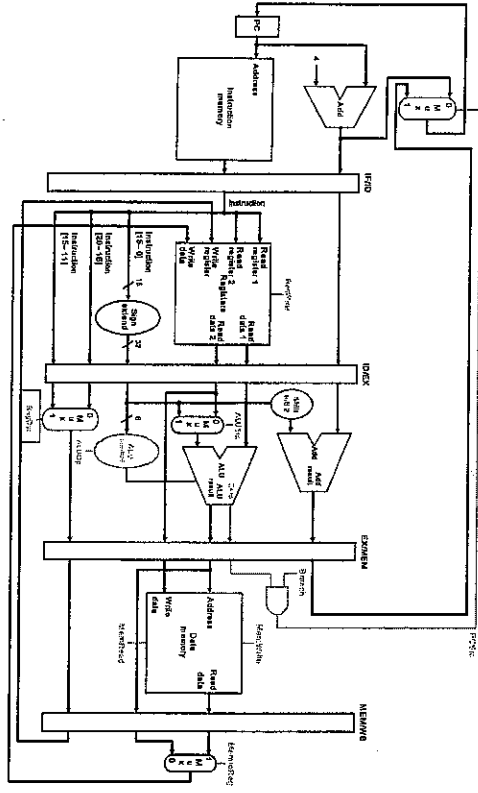
Actual instruction and not a FLUSH

פתרון ב - הערות

- כדי שפתרון ב' יעבוד עלינו לשנות מעט את החומרה: יש לבדוק אם אוגר ה-rd של הפקודה sw זהה לאוגר היעד של הפקודה (כמו בפתרון הקודם).
 - אבל, אוגר ה-rd אינו זמין בשלב ה-MEM (don't care) MUX (קרנר RegDst בוחר בין rd, כאשר היעד של פקודה) היות וערך RegDst אינו בשימוש sw ניתן לבחור לו שימוש ולכן נבחר שערנו יהיה כך שייבחר ב-MUX האוגר ז'.
 - זה מבטיח שאוגר המקור של sw יהיה זמין עבור ההתניות שבשקף הקודם (rd את ז' של פקודת sw ב-EX/MEM)
- 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

Control



אלון שקלר - אוניברסיטת תל אביב

הערות

נמנע מ-stall רק אם אוגר ההיסט (rs) של SW אינו אוגר היעד של Iw.

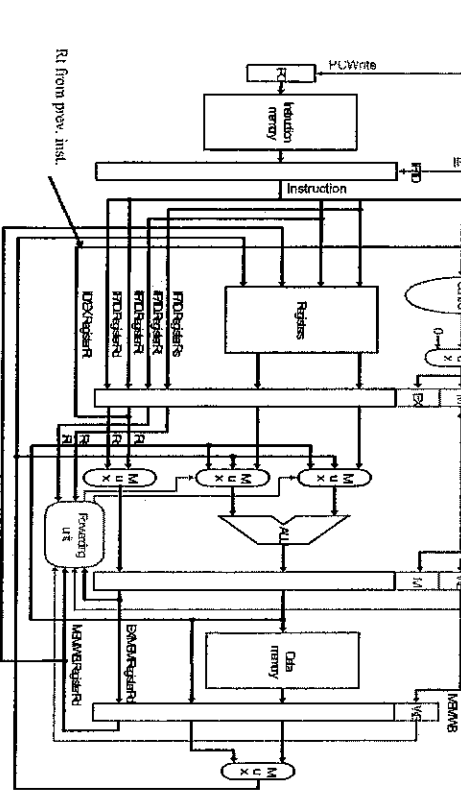
if ID/EX.MemRead and (ID/EX.RegisterRt = IF/ID.RegisterRs) or (ID/EX.RegisterRt = IF/ID.RegisterRt) and not (IF/ID.MemWrite and (ID/EX.RegisterRt = IF/ID.RegisterRs)) and (ID/EX.RegisterRt <> 0) then

Stall the pipeline

- IF/ID.MemWrite אינו סוגל חדש שמתמם פקודת opcode-מה-יפוענא יפוענא מה-opcode.

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



אלון שקלר - אוניברסיטת תל אביב

דוגמא

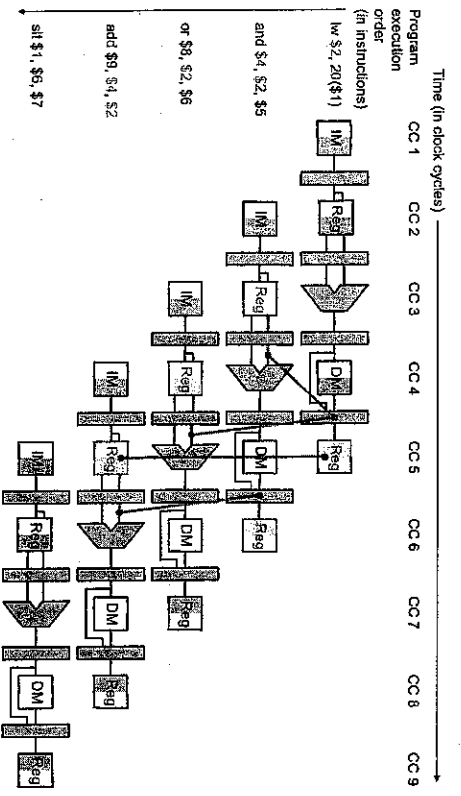
• סדרו את פקודות התכנית הבאה על-מנת שמספר מחזורי השעון שתצטרך יהיה המקסימלי (הכי גרוע מבחינת ביצועים) תוך מתן אותה תוצאה

לולאה \$10, \$11, Loop
 beq \$10, \$11, Loop
 add \$10, \$7, \$8
 add \$8, \$8, \$4
 add \$7, \$7, \$3
 add \$4, \$4, \$5
 lw \$5, 0(\$5)
 lw \$5, 4(\$5)
 lw \$3, 0(\$3)
 add \$3, \$3, \$5
 add \$4, \$4, \$5
 add \$8, \$8, \$4
 add \$7, \$7, \$8
 add \$8, \$8, \$4
 add \$10, \$7, \$8
 beq \$10, \$11, Loop

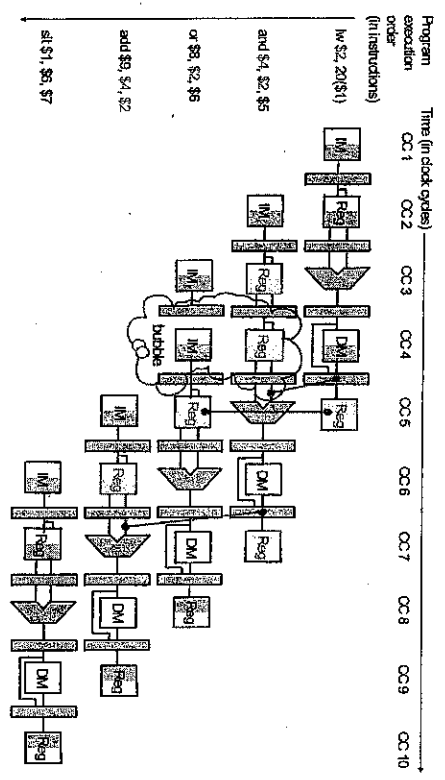
• עליון להשתדל שיהיו כמה שיותר stalls

lw	\$3, 0(\$5)	
lw	\$4, 4(\$5)	
add	\$7, \$7, \$3	
add	\$8, \$8, \$4	
add	\$10, \$7, \$8	
sw	\$6, 0(\$5)	
beq	\$10, \$11, Loop	
lw	\$3, 0(\$5)	# requires stall on \$3
add	\$7, \$7, \$3	
sw	\$6, 0(\$5)	
lw	\$4, 4(\$5)	
add	\$8, \$8, \$4	# requires stall on \$4
add	\$10, \$7, \$8	
add	\$8, \$8, \$4	
beq	\$10, \$11, Loop	

אלון שקלר - אוניברסיטת תל אביב



אלון שקלר - אוניברסיטת תל אביב



אלון שקלר - אוניברסיטת תל אביב