



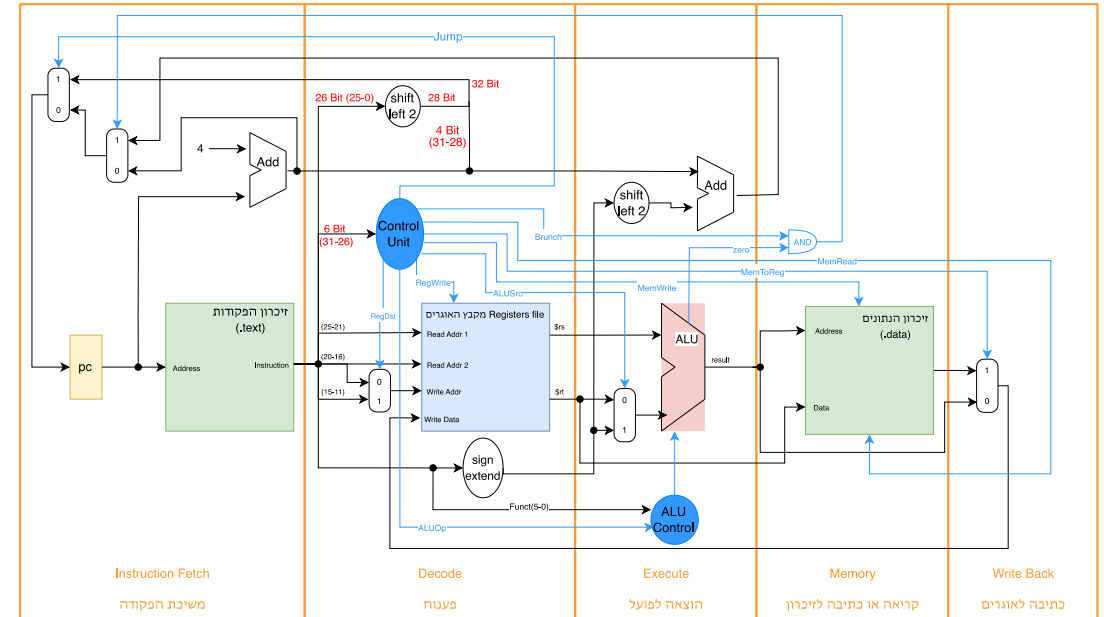
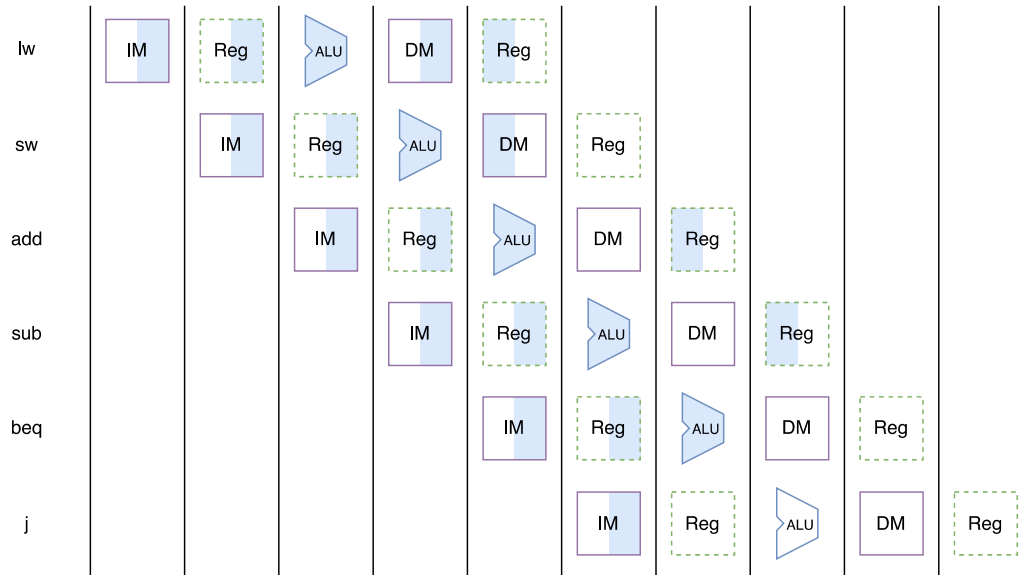
מעבד צנרת - Pipeline

- הקדמה ומתן מוטיבציה
- חלוקה לשלבים ושימוש באוגרי הצנרת
- סיכוני נתונים (Data Hazards)
- סיכוני בקרה (Control Hazards)
- פסיקות וחריגות (סוג של סיכוני בקרה)

מעבד צנרת - סיכום



שלבי הפקודה במעבד והצגתם באופן ויזואלי באמצעות תרשימים



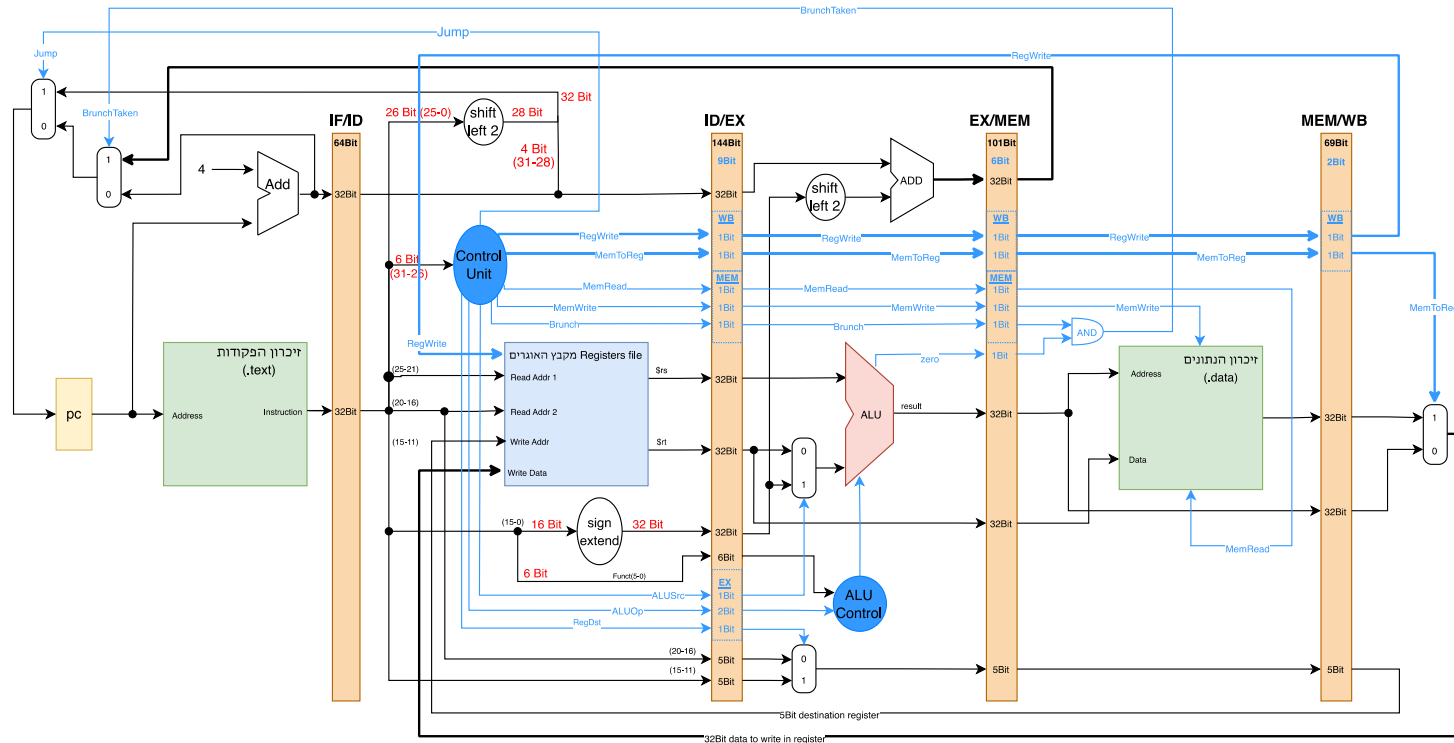
שלב	שם	ראשי תיבות	הרכיב המרכזי	ובקצרה
1	Instruction Fetch	IF	Instruction Memory	IM
2	Instruction Decode	ID	Registers	Rge
3	Execute	EX	ALU	ALU
4	Memory	MEM	Data Memory	DM
5	Write Back	WB	Registers	Reg

מעבד צנרת - סיכום



שימוש באוגרי צנרת וחלוקת אותות הבקרה לפי שלבי הביצוע

	ID	EX				MEM			WB	
פקודה	Jump	RegDst	ALUOp1	ALUOp0	ALUSrc	Branch	MemRead	MemWrite	RegWrite	MemToReg
טיפוס R	0	1	1	0	0	0	0	0	1	0
sw	0	X	0	0	1	0	0	1	0	X
lw	0	0	0	0	1	0	1	0	1	1
beq	0	X	0	1	0	1	0	0	0	X
j	1	X	X	X	X	X	0	0	0	X

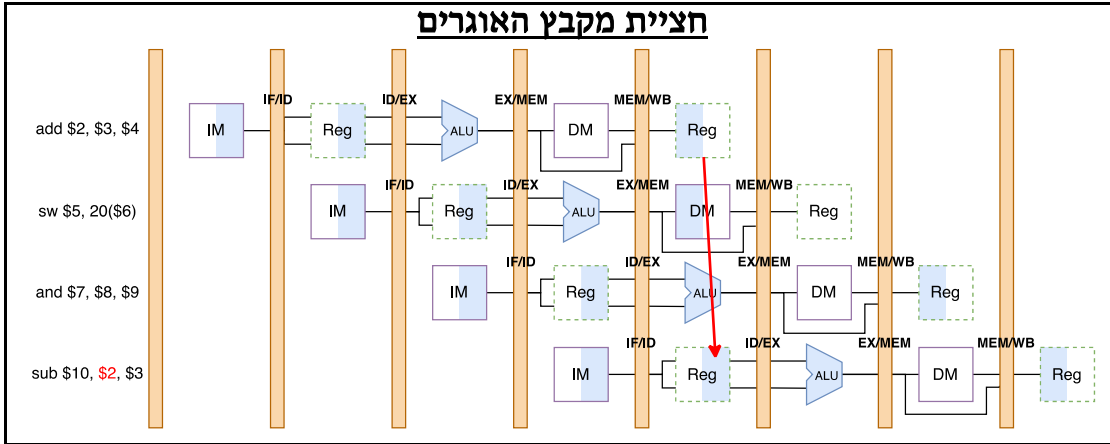


מעבד צנרת - סיכום

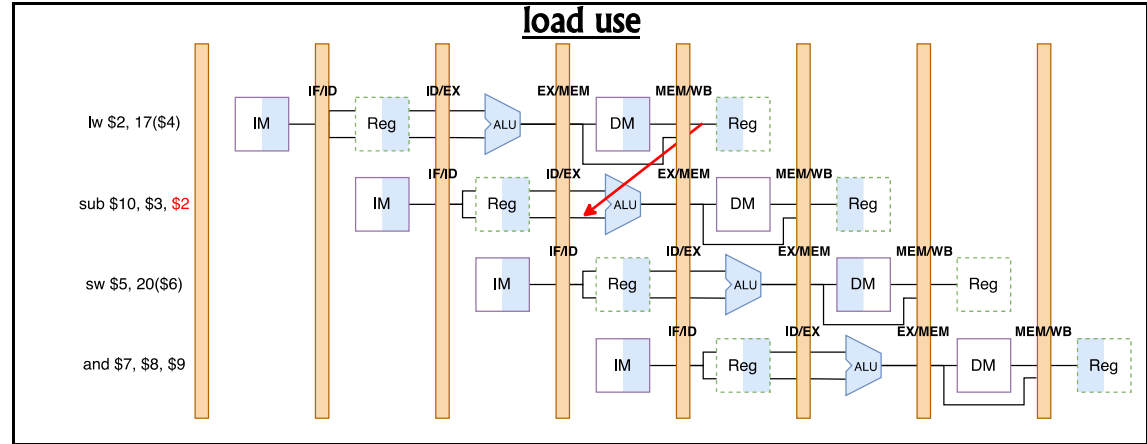


סיכוי נתונים מסוגים שונים

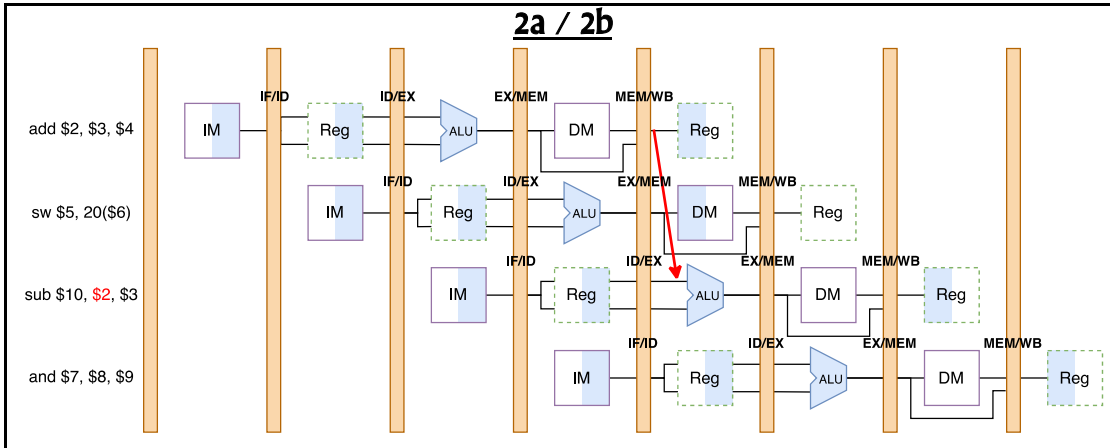
חציית מקבץ האוגרים



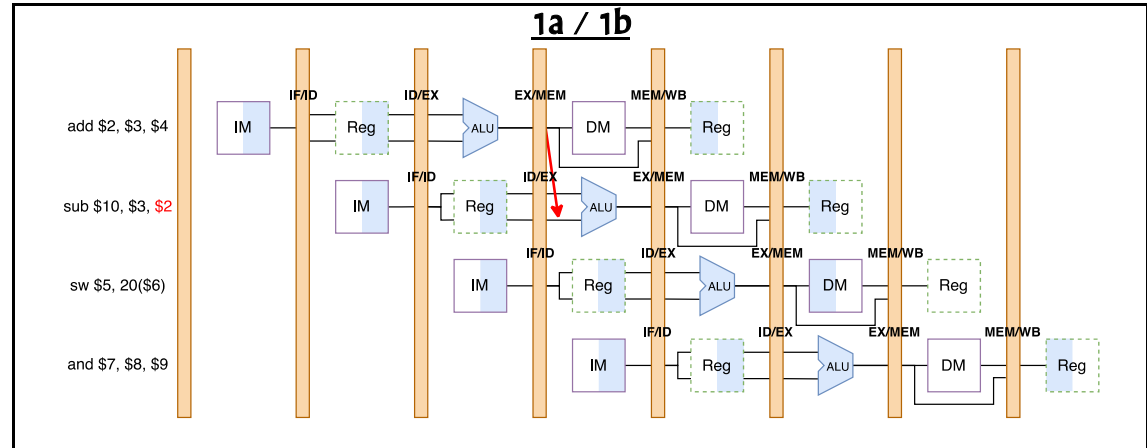
load use



2a / 2b



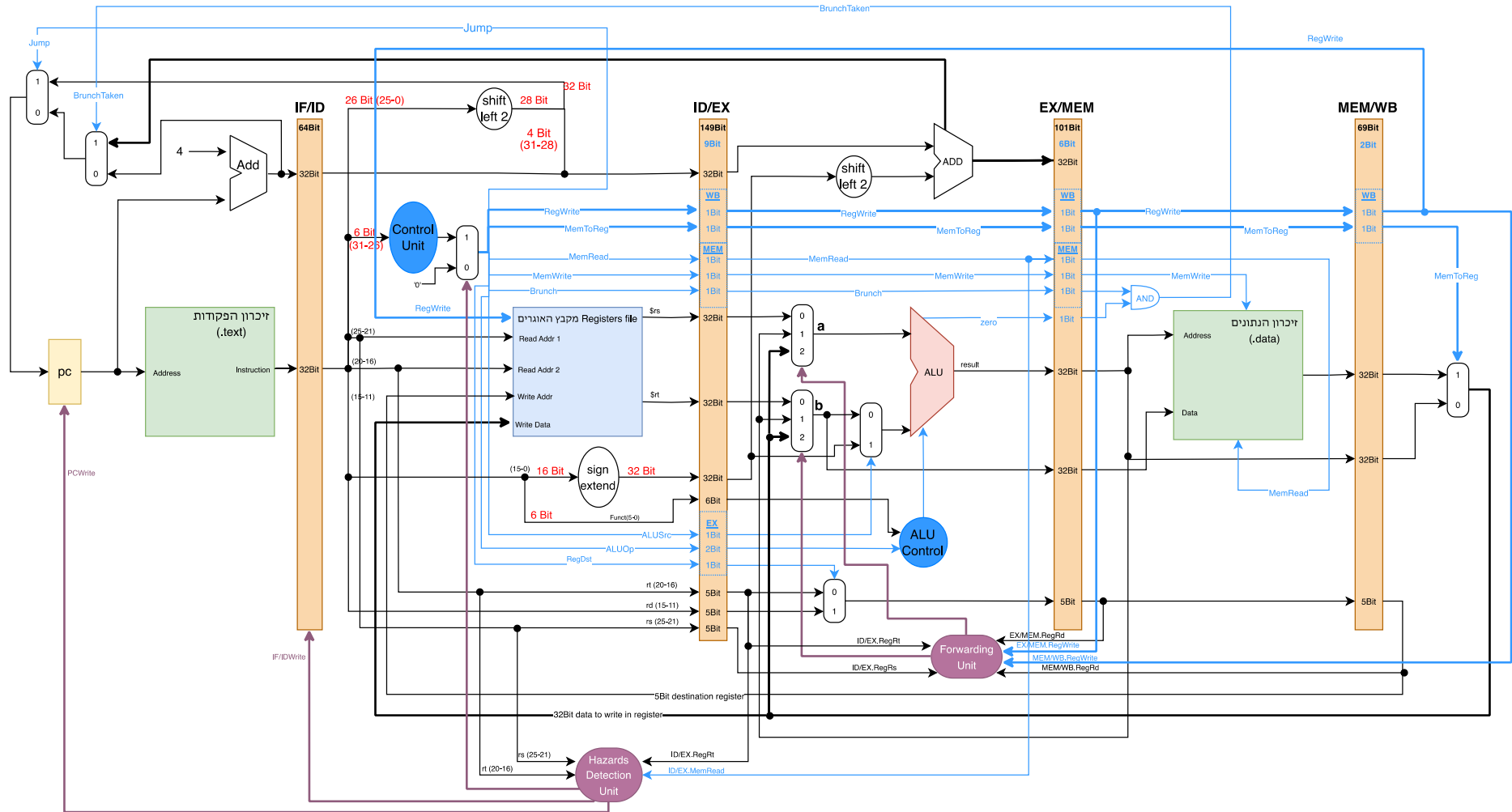
1a / 1b



מעבד צנרת - סיכום



יחידת העברה קדימה לאיתור וטיפול בסיכוני נתונים ויחידה לזיהוי סיכונים לאיתור וטיפול בסיכון load use



מעבד צנרת - סיכום



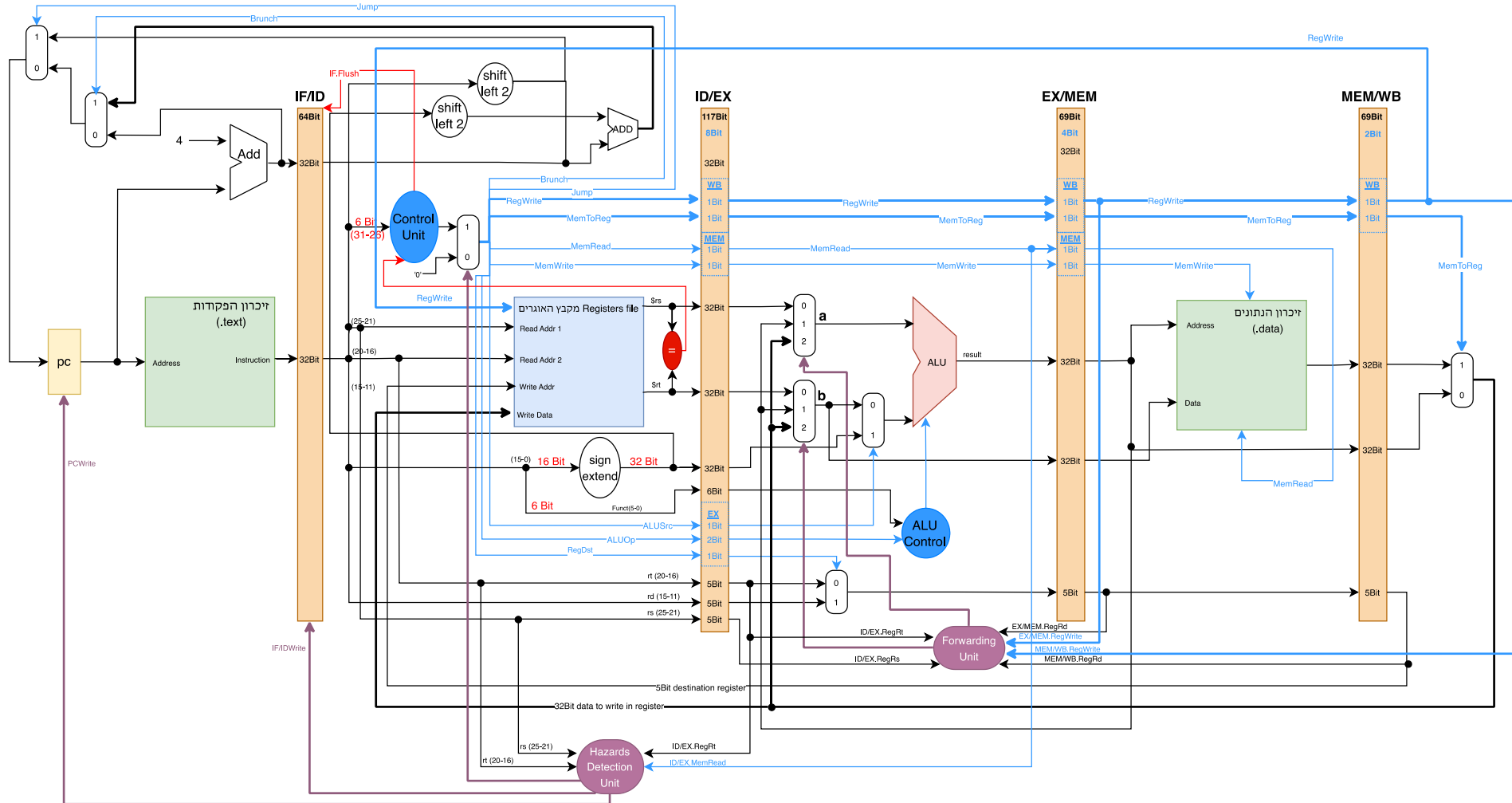
בדיקות שהיחידה להעברה קדימה - Forwarding Unit - מבצעת					
הבדיקות שיחידת העברה קדימה עושה		אוגר המקור של הפקודה הבאה		מרחק בין פקודות	
1a	EX/MEM.RegRd=ID/EX.RegRs & EX/MEM.RegWrite=1 & EX/MEM.RegRd!=0	rs אוגר מקור = rd	a	פקודה אחרי פקודה	1
1b	EX/MEM.RegRd=ID/EX.RegRt & EX/MEM.RegWrite=1 & EX/MEM.RegRd!=0	rt אוגר מקור = rd	b		
2a	MEM/WB.RegRd=ID/EX.RegRs & MEM/WB.RegWrite=1 & MEM/WB.RegRd!=0	rs אוגר מקור = rd	a	פקודה עם רווח של פקודה אחת באמצע	2
2b	MEM/WB.RegRd=ID/EX.RegRt & MEM/WB.RegWrite=1 & MEM/WB.RegRd!=0	rt אוגר מקור = rd	b		

בדיקות שהיחידה לזיהוי סיכונים - Hazards Detection Unit - מבצעת	
הבדיקות שיחידת העברה קדימה עושה	אוגר המקור של הפקודה הבאה
ID/EX.RegRd=IF/ID.RegRs & ID/EX.MemRead=1	rs אוגר מקור = rd
ID/EX.RegRd=IF/ID.RegRt & ID/EX.MemRead=1	rt אוגר מקור = rd

מעבד צנרת - סיכום



סיכוי בקרה - הקדמת Branch לשלב 2



מעבד צנרת - סיכום



Scheduling					
הנחות שיש להניח	דוגמה לקוד אחרי	דוגמה לקוד לפני	הנחת האופטימיזר	מקרה	
אין תלות בתוצאת ה-Brunch ולכן אין צורך להניח שום דבר	beq \$4, \$5, Target sub \$1, \$2, \$3 ... add \$6, \$7, \$8 Target: ... and \$9, \$10, \$11 or \$12, \$13, \$14	sub \$1, \$2, \$3 beq \$4, \$5, Target ... add \$6, \$7, \$8 Target: ... and \$9, \$10, \$11 or \$12, \$13, \$14		ייבוא פקודה הנמצאת לפני ה-Brunch	a
1. מניחים שהמהדר מסוגל לקבוע בצורה דיי מובהקת האם ה-Brunch ילקח או לא, ולכן הוא יכול למלא בפקודה שרוב הסיכויים שהיא באמת צריכה להתבצע. 2. מניחים שהמהדר יודע לבחור פקודות למילוי שגם אם הן תתבצעה הן לא יגרמו לטעות לוגית בתוכנית אלא רק יבזבו מחזור שעון אחד.	sub \$1, \$2, \$3 beq \$4, \$5, Target and \$9, \$10, \$11 ... add \$6, \$7, \$8 Target: ... or \$12, \$13, \$14	sub \$1, \$2, \$3 beq \$4, \$5, Target ... add \$6, \$7, \$8 Target: ... and \$9, \$10, \$11 or \$12, \$13, \$14	Brunch Taken	ייבוא פקודה הנמצאת בכתובת היעד של פקודת ה-Brunch	b
	sub \$1, \$2, \$3 beq \$4, \$5, Target add \$6, \$7, \$8 ... Target: ... and \$9, \$10, \$11 or \$12, \$13, \$14	sub \$1, \$2, \$3 beq \$4, \$5, Target ... add \$6, \$7, \$8 Target: ... and \$9, \$10, \$11 or \$12, \$13, \$14	sub \$1, \$2, \$3 beq \$4, \$5, Target ... add \$6, \$7, \$8 Target: ... and \$9, \$10, \$11 or \$12, \$13, \$14	Brunch NOT Taken	ייבוא פקודה הנמצאת אחרי פקודת ה-Brunch

מעבד צנרת - סיכום



חיזוי דינאמי במעבדים בעלי צנרת עמוקה

BHT (Branch History Table)						1-bit Prediction Accuracy
תוצאת ה-Branch האחרונה	כתובת הפקודה					
BTB (Branch Target Buffer)						
כתובת הפקודה	כתובת היעד לקפיצה	תוצאת ה-Branch אחת לפני האחרונה	תוצאת ה-Branch האחרונה	מיקום	חיזוי	
		0	0	0	0	
		0	1	0	0	
		1	1	0	1	
		1	0	0	1	
		0	1	1	1	
		1	0	1	1	
		0	0	1	0	
		1	1	1	1	


```

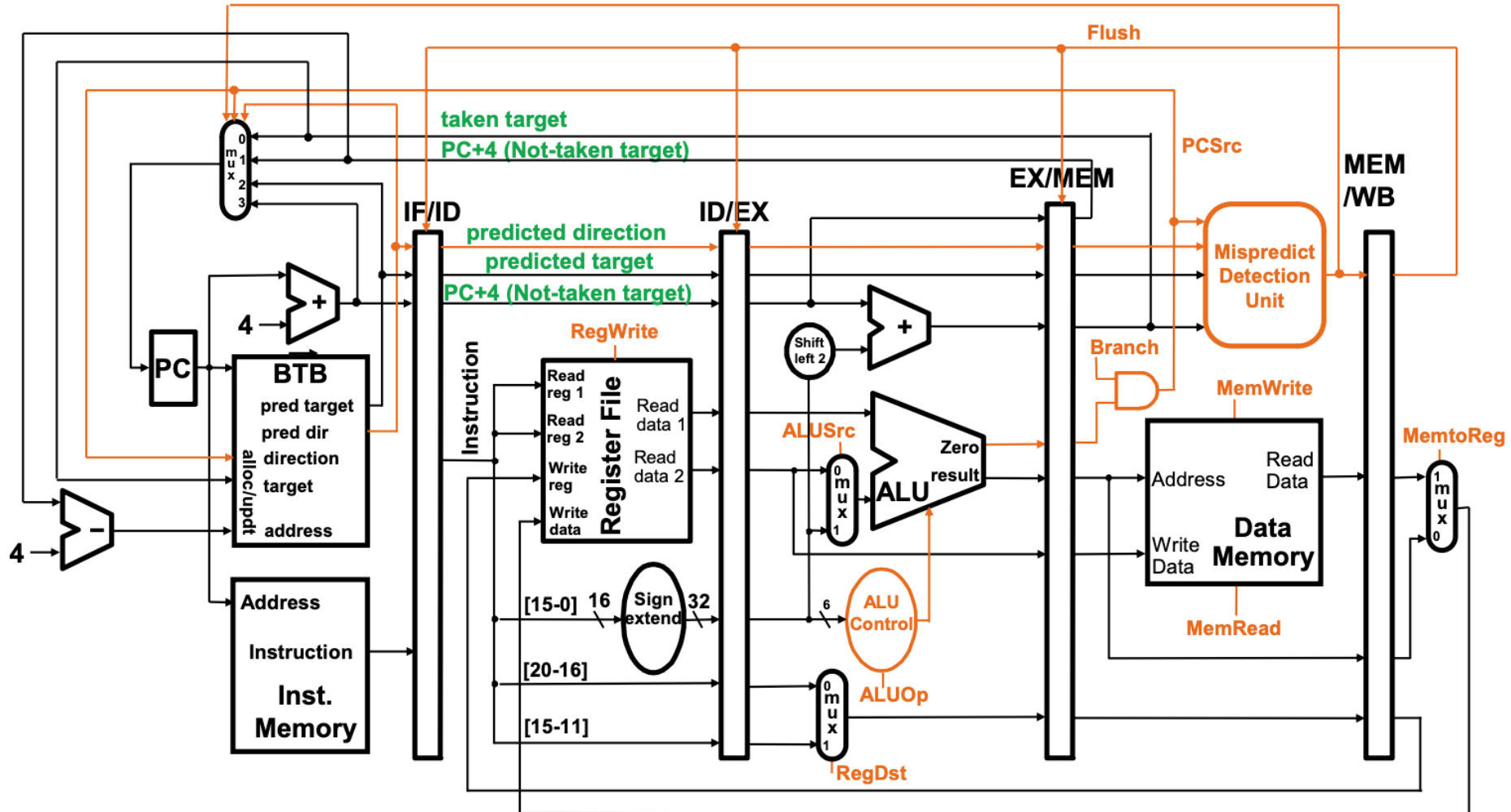
    graph TD
      11((חיזוי: נלקח 11)) -- 1 תוצאה אחרונה --> 11
      11 -- 0 תוצאה אחרונה --> 10((חיזוי: נלקח 10))
      10 -- 0 תוצאה אחרונה --> 10
      10 -- 1 תוצאה אחרונה --> 11
      01((חיזוי: לא נלקח 01)) -- 1 תוצאה אחרונה --> 01
      01 -- 0 תוצאה אחרונה --> 00((חיזוי: לא נלקח 00))
      00 -- 0 תוצאה אחרונה --> 00
      00 -- 1 תוצאה אחרונה --> 01
    
```

2-bit Prediction Accuracy

מעבד צנרת - סיכום



חיזוי דינאמי באמצעות BTB במעבד MIPS



מעבד צנרת - סיכום



פסיקות וחריגות

הטרמינולוגיה של ההפרעות במעבד ה-MIPS:

הטרמינולוגיה של MIPS	מהיכן הגיעה ההפרעה	סוג ההפרעה
פסיקות (interrupts)	חיצוני	I/O - פעולות קלט / פלט של רכיבי חומרה שונים
חריגות (exception)	פנימי	שימוש במערכת ההפעלה מתוך התוכנית
חריגות (exception)	פנימי	גלישה אריתמטית
חריגות (exception)	פנימי	שימוש בפקודה לא חוקית
פסיקות (interrupts) או חריגות (exception)	חיצוני / פנימי	תקלת חומרה

במעבד ה-MIPS קיימים מספר אוגרים המסייעים בטיפול בחריגות. אנחנו נזכיר שניים מהם:

1. אוגר הנקרא cause - נועד לשמירת הסיבה שגרמה לפסיקה / חריגה.
2. אוגר הנקרא EPC (Exception PC) - נועד לשמירת הכתובת של הפקודה המגיעה מיד אחרי הפקודה שגרמה לחריגה, למקרה שמערכת ההפעלה תרצה לחזור לתוכנית ולהמשיך להריץ אותה.

הטיפול בפסיקות וחריגות מתחלק למספר משימות שהמעבד מבצע:

1. עצירה התוכנית: שטיפת הפקודה שגרמה לתקלה וכן של כל הפקודות שבאות אחריה, סיום ביצוע הפקודות שהיו לפני הפקודה הבעייתית.
 2. שמירת המידע הנדרש בנוגע לתקלה: סיבת התקלה נשמרת באוגר cause וכתובת הפקודה שגרמה לתקלה נשמרת באוגר EPC.
 3. קפיצה לכתובת 0x80000180 שבה נמצא ה-exception handler של מערכת ההפעלה שהוא בודק את סיבת הפסיקה ובהתאם מחליט מה לעשות וכיצד לנהוג.
- במידה ושתי פקודות שונות הנמצאות בשני שלבים שונים של הצנרת גורמות לחריגה באותו הזמן, מעבד ה-MIPS יעדיף לטפל בחריגה שנוצרה מהפקודה הראשונה שנכנסה לצנרת (זאת אומרת: הפקודה הנמצאת בשלב מתקדם יותר בצנרת), אך יש מעבדים אשר בוחרים באיזו חריגה לטפל בשיטות שונות.