

BAR-ILAN UNIVERSITY (RA)

Faculty of Engineering

Ramat-Gan 52900, Israel



אוניברסיטת בר-אילן (ע"ר)

הפקולטה להנדסה

רמת-גן 52900

Tel: 03-5317722

engbi@mail.biu.ac.il

מבנה מחשבים ספרתיים

תשע"ט סמסטר ב' מועד א'

83-301

מרצה: פרופ' שמואל וימר

מתרגל: מר בנימין פרנקל

- יש לקרוא היטב את ההוראות.
- חובה לענות על כל השאלות.
- ציון מקסימלי בבחינה: 100 נקודות.
- יש לנמק את כל תשובותיכם.
- יש להקפיד על כתב יד קריא!
- יש לרשום תשובות בתוך הטבלאות המצורפות במקום שנדרש.
- חומר עזר מותר בשימוש: מחשבון, ספר הקורס ושקפי ההרצאות בלבד! כל חומר עזר אחר אסור בהחלט!
- משך הבחינה: שלוש שעות.
- יש לצרף את שאלוני הבחינה למחברת!

בהצלחה!

שאלה מספר 1 (40 נקודות):

נתונה התכנית הבאה:

```

Loop:  LD      R2,0(R1)      ;R2=array element
       DADDIU  R2,R2,#1     ;increment R2
       SD      R2,0(R1)     ;store result
       DADDIU  R1,R1,#8     ;increment pointer
       BNE     R2,R3,LOOP   ;branch if not last element
    
```

התכנית עוברת על איבריו של מערך המכיל מספרי integer ומגדילה את ערכו של כל איבר ב- 1.

מריצים את התכנית על מעבד המסוגל להוציא לדרך 2 פקודות במקביל (two-issue processor) וכן תומך ב- dynamic scheduling. הנח כי המעבד מסוגל גם לסיים שתי פקודות מכל סוג שהוא במחזור שעון אחד. במעבד קיימות יחידות ביצוע נפרדות עבור חישוב הכתובת, ביצוע פעולות ALU ובדיקת תנאי הקפיצה המותנית. נתון כי ביצוע של פעולת ALU אורך מחזור שעון אחד.

- מריצים את התכנית על המעבד כאשר המעבד לא תומך ב- speculation (זאת-אמרת שעובדים על-פי אלגוריתם Tomasulo, אך ללא ROB).
- הטבלה שלפניך מתארת את התקדמות הפקודות לפי מספר מחזור השעון במהלך ריצת התכנית:

No speculation

Loop iteration	Instruction	Issue	Execute	Mem access	Write CDB	Comment
1	LD R2, 0(R1)	1	2	3	4	First issue
1	DADDIU R2, R2, #1	1	5		6	Wait for LW
1	SD R2, 0(R1)	2	3	7		Wait for DADDIU
1	DADDIU R1, R1, #8	2	3		4	Execute directly
1	BNE R2, R3, Loop	3	7			Wait for DADDIU
2	LD R2, 0(R1)	4	8	9	10	Wait for BNE
2	DADDIU R2, R2, #1	4	11		12	Wait for LW
2	SD R2, 0(R1)	5	9	13		Wait for DADDIU
2	DADDIU R1, R1, #8	5	8		9	Wait for BNE
2	BNE R2, R3, Loop	6	13			Wait for DADDIU
3	LD R2, 0(R1)	7	14	15	16	Wait for BNE
3	DADDIU R2, R2, #1	7	17		18	Wait for LW
3	SD R2, 0(R1)	8	15	19		Wait for DADDIU
3	DADDIU R1, R1, #8	8	14		15	Wait for BNE
3	BNE R2, R3, Loop	9	19			Wait for DADDIU
4	LD R2, 0(R1)	10	20	21	22	Wait for BNE
4	DADDIU R2, R2, #1	10	23		24	Wait for LW
4	SD R2, 0(R1)	11	21	25		Wait for DADDIU
4	DADDIU R1, R1, #8	11	20		21	Wait for BNE
4	BNE R2, R3, Loop	12	25			Wait for DADDIU
5	LD R2, 0(R1)	13	26	27	28	Wait for BNE
5	DADDIU R2, R2, #1	13	29		30	Wait for LW
5	SD R2, 0(R1)	14	27	31		Wait for DADDIU
5	DADDIU R1, R1, #8	14	26		27	Wait for BNE
5	BNE R2, R3, Loop	15	31			Wait for DADDIU

א. מהו ה- CPI עבור שלוש האיטרציות הראשונות המתוארות בטבלה?

$$CPI = \frac{19}{15} = 1.2667$$

ב. השלם את הטבלה עבור שתי איטרציות נוספות.

ג. מהו ה- CPI עבור חמש האיטרציות הראשונות?

$$CPI = \frac{31}{25} = 1.24$$

ד. לאיזה ערך שואף ה- CPI עבור מספר גדול של איטרציות?

בין כל איטרציה לאיטרציה הבאה חולפים 6 מחזורי שעון בהם מתבצעות 5 פקודות, ולכן,

$$\text{עבור מספר גדול של איטרציות ה- CPI שואף ל- } 1.2 = \frac{6}{5}$$

- כעת, מוסיפים למעבד תמיכה ב- speculation (עם ROB) ומריצים את התכנית. הטבלה שלפניך מתארת את התקדמות הפקודות במהלך ריצת התכנית:

With speculation

Loop iteration	Instruction	Issue	Execute	Read access	Write CDB	Commit	Comment
1	LD R2, 0(R1)	1	2	3	4	5	First issue
1	DADDIU R2, R2, #1	1	5		6	7	Wait for LW
1	SD R2, 0(R1)	2	3			7	Wait for DADDIU
1	DADDIU R1, R1, #8	2	3		4	8	Commit in order
1	BNE R2, R3, Loop	3	7			8	Wait for DADDIU
2	LD R2, 0(R1)	4	5	6	7	9	No execute delay
2	DADDIU R2, R2, #1	4	8		9	10	Wait for LW
2	SD R2, 0(R1)	5	6			10	Wait for DADDIU
2	DADDIU R1, R1, #8	5	6		7	11	Commit in order
2	BNE R2, R3, Loop	6	10			11	Wait for DADDIU
3	LD R2, 0(R1)	7	8	9	10	12	Earliest possible
3	DADDIU R2, R2, #1	7	11		12	13	Wait for LW
3	SD R2, 0(R1)	8	9			13	Wait for DADDIU
3	DADDIU R1, R1, #8	8	9		10	14	Executes earlier
3	BNE R2, R3, Loop	9	13			14	Wait for DADDIU
4	LD R2, 0(R1)	10	11	12	13	15	Earliest possible
4	DADDIU R2, R2, #1	10	14		15	16	Wait for LW
4	SD R2, 0(R1)	11	12			16	Wait for DADDIU
4	DADDIU R1, R1, #8	11	12		13	17	Commit in order
4	BNE R2, R3, Loop	12	16			17	Wait for DADDIU
5	LD R2, 0(R1)	13	14	15	16	18	Earliest possible
5	DADDIU R2, R2, #1	13	17		18	19	Wait for LW
5	SD R2, 0(R1)	14	15			19	Wait for DADDIU
5	DADDIU R1, R1, #8	14	15		16	20	Executes earlier
5	BNE R2, R3, Loop	15	19			20	Wait for DADDIU

ה. השלם את הטבלה עבור שתי איטרציות נוספות.

ו. מהו ה- CPI עבור חמש האיטרציות הראשונות?

אם מסתכלים על מחזור השעון שבו פקודת ה- branch החמישית מתבצעת (כדי להשוות

$$\text{לסעיף ג') הרי שהיא מתבצעת במחזור השעון ה- 19, ולכן: } CPI = \frac{19}{25} = 0.76$$

אם מסתכלים על מחזור השעון שבו פקודת ה-branch החמישית עשתה commit הרי שזה

$$CPI = \frac{20}{25} = 0.8 \text{ ולכן: } CPI = 0.8$$

שתי התשובות התקבלו.

ז. לאיזה ערך שואף ה-CPI עבור מספר גדול של איטרציות?

בין כל איטרציה לאיטרציה הבאה חולפים 3 מחזורי שעון בהם מתבצעות 5 פקודות, ולכן,

$$CPI = 0.6 \text{ שואף ל- } \frac{3}{5}$$

- מרצה מהפקולטה להנדסה באוניברסיטת בר אילן רצה להשתמש בתכנית הנ"ל על-מנת להמחיש לסטודנטים את שיפור הביצועים הנובע משימוש ב-speculation. כאשר התבונן המרצה בתנאי העצירה של התכנית החליט לשנות את תנאי הקפיצה ולהשוות את הערך האגור ב-R3 לערך האגור ב-R1 במקום לערך האגור ב-R2.
- ח. מלא את הטבלאות בהתאם לתנאי העצירה החדש כאשר בטבלה הראשונה המעבד לא תומך ב-speculation ובטבלה השנייה המעבד תומך ב-speculation (די בשלוש איטרציות ראשונות):

מכיוון שלא הוגדר בשאלה האם ה-memory הוא dual-ported או single-ported, הוחלט לקבל את שתי האופציות.
מי שהניח שהזיכרון הינו dual-ported הרי שטבלה הראשונה (no speculation) צריכה להיראות כך:

No speculation

Loop iteration	Instruction	Issue	Execute	Mem access	Write CDB	Comment
1	LD R2, 0(R1)	1	2	3	4	First issue
1	DADDIU R2, R2, #1	1	5		6	Wait for LW
1	SD R2, 0(R1)	2	3	7		Wait for DADDIU
1	DADDIU R1, R1, #8	2	3		4	Execute directly
1	BNE R1, R3, Loop	3	5			Wait for DADDIU
2	LD R2, 0(R1)	4	6	7	8	Wait for BNE
2	DADDIU R2, R2, #1	4	9		10	Wait for LW
2	SD R2, 0(R1)	5	7	11		Wait for DADDIU
2	DADDIU R1, R1, #8	5	6		7	Execute directly
2	BNE R1, R3, Loop	6	8			Wait for DADDIU
3	LD R2, 0(R1)	7	9	10	11	Wait for BNE
3	DADDIU R2, R2, #1	7	12		13	Wait for LW
3	SD R2, 0(R1)	8	10	14		Wait for DADDIU
3	DADDIU R1, R1, #8	8	10		11	Wait for ALU
3	BNE R1, R3, Loop	9	12			Wait for DADDIU

מי שהניח שהזיכרון הינו single-ported הרי שטבלה הראשונה (no speculation) צריכה להיראות כך:

No speculation

Loop iteration	Instruction	Issue	Execute	Mem access	Write CDB	Comment
1	LD R2, 0(R1)	1	2	3	4	First issue
1	DADDIU R2, R2, #1	1	5		6	Wait for LW
1	SD R2, 0(R1)	2	3	7		Wait for DADDIU
1	DADDIU R1, R1, #8	2	3		4	Execute directly
1	BNE R1, R3, Loop	3	5			Wait for DADDIU
2	LD R2, 0(R1)	4	6	8	9	Wait for BNE + Mem
2	DADDIU R2, R2, #1	4	10		11	Wait for LW
2	SD R2, 0(R1)	5	7	12		Wait for DADDIU
2	DADDIU R1, R1, #8	5	6		7	Execute directly
2	BNE R1, R3, Loop	6	8			Wait for DADDIU
3	LD R2, 0(R1)	7	9	10	11	Wait for BNE
3	DADDIU R2, R2, #1	7	12		13	Wait for LW
3	SD R2, 0(R1)	8	10	14		Wait for DADDIU
3	DADDIU R1, R1, #8	8	9		10	Execute directly
3	BNE R1, R3, Loop	9	11			Wait for DADDIU

הטבלה השנייה (with speculation) לא מושפעת מהשאלה האם הזיכרון הוא dual-ported או single-ported, והיא בכל מקרה צריכה להיראות כך:

With speculation

Loop iteration	Instruction	Issue	Execute	Read access	Write CDB	Commit	Comment
1	LD R2, 0(R1)	1	2	3	4	5	First issue
1	DADDIU R2, R2, #1	1	5		6	7	Wait for LW
1	SD R2, 0(R1)	2	3			7	Wait for DADDIU
1	DADDIU R1, R1, #8	2	3		4	8	Execute directly
1	BNE R1, R3, Loop	3	5			8	Wait for DADDIU
2	LD R2, 0(R1)	4	5	6	7	9	No Execution delay
2	DADDIU R2, R2, #1	4	8		9	10	Wait for LW
2	SD R2, 0(R1)	5	6			10	Wait for DADDIU
2	DADDIU R1, R1, #8	5	6		7	11	Commit in order
2	BNE R1, R3, Loop	6	8			11	Wait for DADDIU
3	LD R2, 0(R1)	7	8	9	10	12	Earliest possible
3	DADDIU R2, R2, #1	7	11		12	13	Wait for LW
3	SD R2, 0(R1)	8	9			13	Wait for DADDIU
3	DADDIU R1, R1, #8	8	9		10	14	Executes earlier
3	BNE R1, R3, Loop	9	11			14	Wait for DADDIU

ט. לאיזה ערך שואף ה-CPI עבור מספר גדול של איטרציות בכל אחת מהאפשרויות שבסעיף הקודם? האם השינוי שעשה המרצה בתנאי הקפיצה ישפיע על המטרה הפדגוגית של התכנית (להמחיש לסטודנטים את שיפור הביצועים הנובע משימוש ב-speculation)? הסבר.

עבור האפשרות של no speculation:

מי שהניח כי הזיכרון הוא dual-ported הרי שעבור מספר גדול של איטרציות ה-CPI שואף ל-0.7, שכן המבנה של איטרציות 2+3 חוזר על עצמו, ולכן בין כל צמד איטרציות לצמד הבא

חולפים 7 מחזורי שעון בהם מתבצעות 10 פקודות, ולכן, עבור מספר גדול של איטרציות ה-CPI שואף ל- $\frac{7}{10} = 0.7$.

מי שהניח כי הזיכרון הוא single-ported הרי שעבור מספר גדול של איטרציות ה-CPI שואף ל- $\frac{2}{3}$, שכן המחזוריות היא של 3 איטרציות, בהן מתבצעות 15 במשך 10 מחזורי שעון,

ולכן, עבור מספר גדול של איטרציות ה-CPI שואף ל- $\frac{10}{15} = \frac{2}{3} = 0.667$.

לשם נוחות, מצורף נספח עם טבלאות של 7 איטרציות ראשונות (טבלה עבור כל אפשרות), שם ניתן לזהות בקלות את המחזוריות של ריצת התכנית.

עבור האפשרות של with speculation:

בין כל איטרציה לאיטרציה הבאה חולפים 3 מחזורי שעון, בהם מתבצעות 5 פקודות, ולכן,

עבור מספר גדול של איטרציות ה-CPI שואף ל- $\frac{3}{5} = 0.6$.

לכן, השינוי שעשה המרצה בתנאי הקפיצה פוגע במידת מה במטרה הפדגוגית (אם כי לא מחסל אותה לגמרי), שכן היתרון מבחינת ביצועים של השימוש ב-speculation נעשה קטן באופן משמעותי עבור תנאי הקפיצה החדש. אילו היה המרצה משתמש בתנאי הקפיצה המקורי, היתרון של השימוש ב-speculation היה מובהק יותר.

שאלה מספר 2 (50 נקודות):

בשאלה זו נעסוק במעבד בעל חזאי קפיצות מבוסס 2-bit-saturated counters (כל counter הינו בעל מצבים: SNT, WNT, WT, ST). החזאי משתמש ב- $k=3$ סיביות היסטוריה. ההיסטוריה הינה פרטית (לוקאלית), ומכונות המצבים פרטיות (לכל פקודת branch יש את החזאים שלה).

נתונה התכנית הבאה:

```

MOV R5, 0          # i=0
Loop: LW  R1, A(R5)  # A[i]
      BL  R1, 5, Cond # if A[i] < 5 jump to Cond --br1
      ADD R2, R2, R1  # R2 = R2 + A[i]
      BL  R1, 10, Cond # if A[i] < 10 jump to Cond --br2
      ADD R3, R3, R1  # R3 = R3 + A[i]
Cond: ADD R5, R5, 1   # i = i + 1
      BL  R5, N, Loop # if i < N jump to Loop --br3
    
```

נתון כי איברי המערך הם: $\{0, 6, 12, 0, 6, 12, \dots\}$ (משמאל לימין), וגודלו (וכך גם הערך N) הוא כפולה שלמה של 3. בכל סעיפי השאלה התכנית מורצת אינסוף פעמים, ואנו נתעניין במצב מתמיד בלבד (ולא בשלבי הלימוד של החזאי).

עבור כל אירוע בהיסטוריה נסמן: $NT = 0$ ו- $T = 1$. נתון כי אוגרי ההיסטוריה מאותחלים באפסים וכי ערכי היסטוריה חדשים מוכנסים ל-LSB.

שים לב: המצב ההתחלתי של מכונות המצבים לא נתון. בכל סעיף בשאלה בו קיימת תלות במצב זה, יש לציין זאת, ולענות על השאלה כתלות במצב ההתחלתי.

א. עבור כל קפיצה בנפרד ($br1, br2, br3$), מהי סדרת ה- T/NT ? הכוונה היא למהלך התכנית בפועל, ולא לחיזוי. אם קיימת מחזוריות, ציין אותה. התייחס למצב מתמיד בלבד! עבור $T, NT, NT - br1$ מחזורי.

עבור $T, NT - br2$ מחזורי. יש לשים לב שאם $br1$ נלקח אז $br2$ לא ירוץ בכלל.

עבור $br3 - N-1$ פעמים T ולבסוף NT (מחזורי, כיוון שהתכנית רצה אינסוף פעמים).

ב. נריץ את התכנית הנתונה על מעבד בעל החזאי הנתון.

- מהם ערכי ההיסטוריה השונים של כל קפיצה בנפרד, ומהם מצבי המכונה והחיזויים בכל ערך היסטוריה? התייחס גם להשפעתו של הערך N על התוצאה.
- מהו ה- misprediction rate לכל קפיצה בנפרד? מהו ה- total misprediction rate?

br1		br2		br3	
History	SM	History	SM	History	SM
100	ST - T	010	ST - T	110	ST - T
001	SNT - NT	101	SNT - NT	101	ST - T
010	SNT - NT			011	if N=3: SNT - NT if N>3: ST - T
				if N>3: 111	ST (mostly) - T WT (once every iteration) - T

פקודות br1 ו-br2 לא טועות לעולם, ולכן: $MR_{br1} = MR_{br2} = 0$
 לגבי br3: אם $N=3$ אזי br3 לא טועה לעולם, ולכן $MR_{br3} = 0$. אם $N>3$ אזי br3 טועה באחד מתוך N חיזויים, ולכן: $MR_{br3} = \frac{1}{N}$.

סה"כ: פקודות br1 ו-br3 רצות פעם אחת בכל איטרציה, ואילו br2 רצה ב- $\frac{2}{3}$ מהאיטרציות. ולכן, אם $N=3$ אז: $total\ misprediction\ rate = 0$.

$$total\ misprediction\ rate = \frac{mispredictions}{branches} = \frac{1}{N + \frac{2N}{3} + N} = \frac{3}{8N} \quad \text{אם } N>3 \text{ אז:}$$

- מכאן ואילך נניח כי $N=6$.

ג. מהו הערך k (מספר סיביות ההיסטוריה) הנמוך ביותר שיאפשר $total\ misprediction\ rate = 0\%$?

בתכנית זו רק br3 תושפע מ- k . על מנת לאפשר $MR_{br3} = 0\%$ עלינו להיות במצב בו עבור ערך היסטוריה החיזוי יהיה נכון. אם $k = N - 1 = 5$ אז יהיה ערך היסטוריה ייחודי לכל איטרציה של הלולאה (מה שאומר שיש מכונה ייעודית לכל שלב בלולאה), ולכן החיזוי יהיה מושלם. כל k קטן מזה גורם לכך שקיים שיתוף של אותה מכונת מצבים בין מצב שבו צריך לקפוץ ובין מצב שבו לא צריך לקפוץ (מכונת המצבים שמתאימה לערך היסטוריה '1111' תהיה משותפת, ולכן לא ייתכן חיזוי מושלם).

- התכנית רצה על מעבד מצונר בעל 5 שלבים. נתון כי החיזוי מתבצע בשלב הראשון, וכי ה-branch resolution הינו בשלב השלישי. אין עיכובים של מערכת הזיכרון ($CPI_{ideal} = 1$).

ד. מהו החלק היחסי של פקודות הקפיצה המותנית בפילוג הדינאמי של פקודות התכנית? עבור $k=3$, מהו ה- CPI של המעבד עבור התכנית הנתונה? כתבו ביטוי כללי תחילה, ולאחר מכן הציבו מספרים.

באיטרציה בה $A[i]=0$ יש 4 פקודות, מתוכן שתי פקודות branch.
 באיטרציה בה $A[i]=6$ יש 6 פקודות, מתוכן 3 פקודות branch.
 באיטרציה בה $A[i]=12$ יש 7 פקודות, מתוך 3 פקודות branch.
 בנוסף, יש את פקודת האתחול של הרגיסטר R5, המתבצעת פעם אחת ב-6 איטרציות (ז"א 0.5 פקודה כל 3 איטרציות).
 סה"כ: 8 פקודות branch מתוך 17.5 פקודות (או 16 פקודות branch מתוך 35 פקודות), ולכן: $\%br = 45.714\%$.
 ה- MR כבר חושב בסעיף ב'. ה- BR בשלב שלישי, ולכן שגיאת חיזוי גוררת אובדן ביצועים של שני מחזורי שעון.

$$CPI = CPI_{ideal} + \%br \times MispredictionRate \times MispredictionPenalty = 1 + \frac{16}{35} \times \frac{1}{16} \times 2 = 1.057$$

ה. על-מנת לחסוך במשאבי מערכת, הוחלט להשאיר את ההיסטוריות פרטיות, אך להשתמש במכונות מצבים משותפות לכל פקודות ה-branch. עבור $k=3$, מהו עתה ה- $misprediction\ rate$ לכל קפיצה בנפרד? מהו ה- $total\ misprediction\ rate$? נמק היטב את צעריך.

נתבונן במצבי ההיסטוריה ונזהה מצבים משותפים ומצבים ייחודיים:
 מצבים '100' ו-'001' הם מצבים ייחודיים ל-br1 בלבד, ולכן מכונות אלו יתפקדו כפרטיות ל-br1.
 מצבים '111', '011' ו-'110' הם מצבים ייחודיים ל-br3 בלבד, ולכן מכונות אלו יתפקדו כפרטיות ל-br3.

מצב '010' הינו משותף ל- br1 ול- br2, כאשר br2 תמיד תקפוץ במצב הזה, ואילו br1 תמיד לא תקפוץ במצב הזה. מכיוון שנתון שסיביות ההיסטוריה מאותחלות לאפס, יוצא שהפעם הראשונה ש"נבקר" במצב של '010' זה עבור פקודה br1, ולאחר מכן נבקר לסירוגין, פעם עבור br2 ופעם עבור br1. לכן, ה- misprediction rate של כל פקודה תלוי באתחול של מכונות המצבים:

עבור אתחול של SNT או של WNT נקבל שהמכונה המתאימה להיסטוריה '010' תחזה תמיד NT, ולכן החיזוי שלה עבור br1 תמיד יהיה נכון, ואילו החיזוי שלה עבור br2 תמיד יהיה שגוי. עבור אתחול של ST נקבל שהמכונה המתאימה להיסטוריה '010' תחזה תמיד T, ולכן החיזוי שלה עבור br1 תמיד יהיה שגוי, ואילו החיזוי שלה עבור br2 תמיד יהיה נכון. עבור אתחול של WT נקבל שהמכונה המתאימה להיסטוריה '010' תחזה לסירוגין NT, T, באופן שהיא תמיד טועה, ולכן החיזוי שלה גם עבור br1 וגם עבור br2 יהיה שגוי. שימו-לב שהסיטואציה האחרונה לא הייתה מתרחשת עם מכונות המצבים שהוצגה בהרצאה.

מצב '101' הינו משותף ל- br2 ול- br3, כאשר br3 תמיד תקפוץ במצב הזה, ואילו br2 תמיד לא תקפוץ במצב הזה. בכל איטרציה של ריצת התכנית "נבקר" במצב זה עבור פקודות: br3, br2, br2 (משמאל לימין) וחוזר חלילה. לכן, המכונה הזאת תמיד תחזה NT (במצב היציב), ולכן, התרומה שלה ל- misprediction rate לא תלויה באתחול של מכונות המצבים. יוצא, שהחיזוי של מכונה זו יהיה תמיד נכון עבור br2, ויהיה תמיד שגוי עבור br3 (וזאת בנוסף לטעות שיש למכונות המצבים של ההיסטוריה '111' באיטרציה השישית).

לסיכום:

$$MispredictionRate_1 = \begin{cases} 0, & \text{if init SNT, WNT} \\ 1/3, & \text{if init ST, WT} \end{cases}$$

$$MispredictionRate_2 = \begin{cases} 0, & \text{if init ST} \\ 1/2, & \text{if init SNT, WNT, WT} \end{cases}$$

$$MispredictionRate_3 = 1/3$$

$$Total \ MispredictionRate = \frac{\text{mispredictions}}{\text{branches}} = \begin{cases} \frac{2+2}{6+4+6} = 25\%, & \text{if init ST, WNT, SNT} \\ \frac{2+2+2}{6+4+6} = 37.5\%, & \text{if init WT} \end{cases}$$

שאלה מספר 3 (20 נקודות):

בשאלה זו נעסוק בתכנון מערכת של multi-level cache.

נדרש לתכנן זיכרון מטמון היררכי. על זיכרון המטמון להיות בעל **Inclusive policy**. נפח הזיכרון המוקצה לצורך בניית זיכרון המטמון ההיררכי הינו $10a$ Blocks, כאשר נתון כי זמן הגישה ל- $L1$ יכול לעמוד במחזור שעון בודד אם הנפח של $L1$ לא עולה על a Blocks. עוד נתון כי זמני הגישה להיררכיות השונות של זיכרון המטמון הם לינאריים בגודלה של כל היררכיה, כך שזמן הגישה להיררכיה מסוימת בגודל של a בלוקים הינו n מחזורי שעון. בנוסף, זמן הגישה ל- main memory הינו 100 מחזורי שעון.

שני מהנדסים צעירים הציעו שתי הצעות שונות לתכנון מערכת זיכרון המטמון ההיררכי:

מהנדס א': זיכרון המטמון יהיה בנוי משתי היררכיות בלבד, כאשר:

$L1 = a$ Blocks, $L2 = 9a$ Blocks. המהנדס בדק ומצא כי עבור תכנית הבדיקה מתקיים:

$$MR_{L1} = 10\%, \text{ וכן: } MR_{L2} = 10\%.$$

מהנדס ב': זיכרון המטמון יהיה בנוי מארבע היררכיות, כאשר: $L1 = a$ Blocks, וכל היררכיה גדולה מקודמתה ב- a Blocks. המהנדס בדק ומצא כי עבור תכנית הבדיקה מתקיים: $MR_{L1} = 10\%$,

$$\text{וכן: } MR_{L2} = MR_{L3} = MR_{L4} = 50\%.$$

א. חשב את ה- $AMAT$ עבור כל אחת מהארכיטקטורות שלעיל. איזו הצעה מבין השתיים עדיפה?

עבור ההצעה של מהנדס א':

$$AMAT = L1 \text{ access time} + L1 \text{ miss rate} \cdot (L2 \text{ access time} + L2 \text{ miss rate} \cdot \text{memory access time})$$

$$AMAT = 1 + 0.1 \cdot (9 + 0.1 \cdot 100) = 2.9 [\text{cycles}]$$

עבור ההצעה של מהנדס ב':

כל היררכיה גדולה מקודמתה ב- a בלוקים, ולכן, מדובר ב- 4 היררכיות של זיכרון מטמון, בעלות גודל של: $a, 2a, 3a, 4a$ בתיים, ובעלות זמני גישה של: 1, 2, 3, 4 מחזורי שעון בהתאמה. לכן:

$$AMAT = 1 + 0.1 \cdot (2 + 0.5 \cdot (3 + 0.5 \cdot (4 + 0.5 \cdot 100))) = 2.7 [\text{cycles}]$$

ולכן, ההצעה של מהנדס ב' עדיפה.

- מהנדס ג', אשר שמע את הדיון בין השניים, הציע הצעה שלישית, לפיה זיכרון המטמון יהיה בנוי מ- 10 רמות היררכיות של a Blocks כל אחת. הוא אמנם אינו ביצע כל בדיקות לגבי ה- Miss Rate , אבל בהתבסס על הבדיקות שעשו חבריו הוא ידע כי מתקיים: $MR_{L1} = 10\%$. לטענתו, מכיוון שהוא מחלק את הזיכרון להיררכיות בעלות a Blocks כל אחת, לכן זמן הגישה לכל היררכיה יהיה מינימאלי (מחזור שעון יחיד), ובכך ביצועי הזיכרון ישתפרו.

ב. האם יתכן שמהנדס ג' צודק? אם כן, באילו תנאים? אם לא – נמק.

מהנדס ג' אמנם צודק שזמן הגישה לכל היררכיה יהיה מינימאלי, אך הוא טועה לגבי ביצועי המערכת. מכיוון שנתון שמדובר במטמון בעל **Inclusive policy**, הרי שכל היררכיה חייבת להכיל את המידע הקיים בהיררכיות שמתחתיה (שקרובות יותר ל- CPU). אם כן, הצעתו של מהנדס ג' גוררת מצב אבסורדי של 10 היררכיות של cache אשר מכילות כולן את אותו המידע. מובן שאם היה miss ב- $L1$, אין שום סיכוי למצוא את המידע המבוקש בשאר ההיררכיות, אשר לא מכילות שום מידע נוסף.

ג. חשב את ה- $AMAT$ עבור הארכיטקטורה שהציע מהנדס ג'.

$$\text{עבור ההצעה של מהנדס ג' נקבל: } AMAT = 1 + 0.1 \cdot (9 + 100) = 11.9 [\text{cycles}]$$

נספח (עבור שאלה מספר 1):

7 איטרציות ראשונות של ריצת התכנית:

- בהנחה שהזיכרון הוא dual-ported:

No speculation – Dual-Ported

Loop iteration	Instruction	Issue	Execute	Mem access	Write CDB	Comment
1	LD R2, 0(R1)	1	2	3	4	First issue
1	DADDIU R2, R2, #1	1	5		6	Wait for LW
1	SD R2, 0(R1)	2	3	7		Wait for DADDIU
1	DADDIU R1, R1, #8	2	3		4	Execute directly
1	BNE R1, R3, Loop	3	5			Wait for DADDIU
2	LD R2, 0(R1)	4	6	7	8	Wait for BNE
2	DADDIU R2, R2, #1	4	9		10	Wait for LW
2	SD R2, 0(R1)	5	7	11		Wait for DADDIU
2	DADDIU R1, R1, #8	5	6		7	Execute directly
2	BNE R1, R3, Loop	6	8			Wait for DADDIU
3	LD R2, 0(R1)	7	9	10	11	Wait for BNE
3	DADDIU R2, R2, #1	7	12		13	Wait for LW
3	SD R2, 0(R1)	8	10	14		Wait for DADDIU
3	DADDIU R1, R1, #8	8	10		11	Wait for ALU
3	BNE R1, R3, Loop	9	12			Wait for DADDIU
4	LD R2, 0(R1)	10	13	14	15	
4	DADDIU R2, R2, #1	10	16		17	
4	SD R2, 0(R1)	11	14	18		
4	DADDIU R1, R1, #8	11	13		14	
4	BNE R1, R3, Loop	12	15			
5	LD R2, 0(R1)	13	16	17	18	
5	DADDIU R2, R2, #1	13	19		20	
5	SD R2, 0(R1)	14	17	21		
5	DADDIU R1, R1, #8	14	17		18	Wait for ALU
5	BNE R1, R3, Loop	15	19			
6	LD R2, 0(R1)	16	20	21	22	
6	DADDIU R2, R2, #1	16	23		24	
6	SD R2, 0(R1)	17	21	25		
6	DADDIU R1, R1, #8	17	20		21	
6	BNE R1, R3, Loop	18	22			
7	LD R2, 0(R1)	19	23	24	25	
7	DADDIU R2, R2, #1	19	26		27	
7	SD R2, 0(R1)	20	24	28		
7	DADDIU R1, R1, #8	20	24		25	Wait for ALU
7	BNE R1, R3, Loop	21	26			

- בהנחה שהזיכרון הוא single-ported:

Loop iteration	Instruction	Issue	Execute	Mem access	Write CDB	Comment
1	LD R2, 0(R1)	1	2	3	4	First issue
1	DADDIU R2, R2, #1	1	5		6	Wait for LW
1	SD R2, 0(R1)	2	3	7		Wait for DADDIU
1	DADDIU R1, R1, #8	2	3		4	Execute directly
1	BNE R1, R3, Loop	3	5			Wait for DADDIU
2	LD R2, 0(R1)	4	6	8	9	Wait for BNE + Mem
2	DADDIU R2, R2, #1	4	10		11	Wait for LW
2	SD R2, 0(R1)	5	7	12		Wait for DADDIU
2	DADDIU R1, R1, #8	5	6		7	Execute directly
2	BNE R1, R3, Loop	6	8			Wait for DADDIU
3	LD R2, 0(R1)	7	9	10	11	Wait for BNE
3	DADDIU R2, R2, #1	7	12		13	Wait for LW
3	SD R2, 0(R1)	8	10	14		Wait for DADDIU
3	DADDIU R1, R1, #8	8	9		10	Execute directly
3	BNE R1, R3, Loop	9	11			Wait for DADDIU
4	LD R2, 0(R1)	10	12	13	14	Wait for BNE
4	DADDIU R2, R2, #1	10	15		16	
4	SD R2, 0(R1)	11	13	17		
4	DADDIU R1, R1, #8	11	13		14	Wait for ALU
4	BNE R1, R3, Loop	12	15			
5	LD R2, 0(R1)	13	16	18	19	Wait for BNE + Mem
5	DADDIU R2, R2, #1	13	20		21	
5	SD R2, 0(R1)	14	17	22		
5	DADDIU R1, R1, #8	14	16		17	
5	BNE R1, R3, Loop	15	18			
6	LD R2, 0(R1)	16	19	20	21	
6	DADDIU R2, R2, #1	16	22		23	
6	SD R2, 0(R1)	17	20	24		
6	DADDIU R1, R1, #8	17	19		20	
6	BNE R1, R3, Loop	18	21			
7	LD R2, 0(R1)	19	22	23	24	
7	DADDIU R2, R2, #1	19	25		26	
7	SD R2, 0(R1)	20	23	27		
7	DADDIU R1, R1, #8	20	23		24	Wait for ALU
7	BNE R1, R3, Loop	21	25			

No speculation – Single-Ported