

ארכיטקטורת מחשבים

תשע"ד סמס' ב' מועד ג'

83-248

- מרצה : פרופ' שמואל וימר
- מתרגל : מר יוסף לונדון
- **חומר עזר מותר:** מחברות/ דפים מהרצאות, תרגולים ותרגילי בית, ספר computer organization and design, מילון עברי צרפתי.
- משך המבחן שלוש שעות
- סך כל הנקודות הוא 110. הציון המרבי הינו 100.
- משקל השאלות השונות נתון בגוף השאלות.
- **יש לנמק את כל תשובותיכם.** אין צורך לפתח מחדש תוצאות שהוכחו בכיתה, אלא אם כן נאמר מפורשות לעשות כן.
- יש לשרטט דיאגרמות באופן ברור !
- הכתיבה בעט בלבד. כתיבה בעפרון לא תיבדק.

בהצלחה!

שאלה 1 (40 נקודות)

שאלת מטמון:

נתונה חלק מתוכנית בC:

```
char a[STOP];

int i, j, temp;

for (i=0 ; i<10,000; i=i+1)

    for (j=0 ; j<STOP; j=j+STEP)

        sum=sum+a[j];
```

הניחו גודל של CHAR הוא byte יחיד. הניחו שבזכרון המרכזי שמור רק ערך המערך a.

הרעיון של השאלה הוא ניתוח מבנה המטמון (CACHE) מתוך זמן ההרצה של התוכנית הנ"ל.

התוכנית רצה כתוצאה מהלולאה החיצונית 10,000 פעמים. הלולאה הפנימית קוראת מידע ממערך של bytes כתלות בגודל של STEP. כל המשתנים נשמרים ברגיסטרים כך שזה מהווה "אפס זמן" גישה וכן שחישוב סכום וכדו' מהווה "אפס זמן".

מריצים את התוכנית על מעבד עם ערכים שונים לצורך בחינה של מבנה הCACHE ומודדים את הזמן הממוצע (!) לביצוע שורת הקוד: $sum=sum+a[j];$

		STEP							
STOP		1	2	4	8	16	32	64	128
	8	10ns	10ns	10ns	10ns	10ns	10ns	10ns	10ns
	16	10ns	10ns	10ns	10ns	10ns	10ns	10ns	10ns
	32	10ns	10ns	10ns	10ns	10ns	10ns	10ns	10ns
	64	20ns	30ns	50ns	50ns	10ns	10ns	10ns	10ns
	128	20ns	30ns	50ns	50ns	50ns	10ns	10ns	10ns
	256	20ns	30ns	50ns	50ns	50ns	50ns	10ns	10ns

רמזים לפתרון השאלה:

- שימו לב לגודל של STEP ונסו להדגים לעצמכם מה יקרה עבור גדלים שונים. השוו עם תוצאות הטבלה.
- $AMAT = Hit\ time + Miss\ Rate * Misspenalty$
- זכרו, הזמנים הם זמנים ממוצעים. כלומר איטרציות יחידאיות מבחינת זמן, נהיים זניחים א. מהו גודל ה CACHE מבחינת DATA? כלומר, מהו כמות המידע שמכיל הCACHE?
- ניתן לזהות די מיידית כי הלולאה הפנימית מבצעת מספר גישות לזכרון בהתאם לחסם שבחוץ. כאשר החסם (STOP) באורך של 32 ויחידת הקפיצה (STEP) באורך של 1, ניגשים צעד אחר צעד בזכרון. כפי שניתן לראות, עד חסם של 64 אין שינוי בזמני הגישה ולכן כנראה עד לשם יש hit. מה שאומר, שגודל המטמון הוא כנראה 32 byte שכן עבור צעד של 1 byte תמיד אנחנו ב HIT וברגע שנחרג ל64 אנחנו מתחילים לפספס.
- נדגיש שוב – הלולאה רצה 10,000 פעם כך שגם אם בפעם הראשונה תמיד פספסנו, בפעם השניה (וכן הלאה עד 10000) המידע כבר שמור במטמון.

כבר בפעם הראשונה שמבצעים את הלולאה cache מתמלא במידע ובזה נגמר הסיפור. לא משנה מהו אורך הבלוק או האסוציאטיביות של ה-CACHE, עבור CACHE בגודל הנ"ל תמיד נוכל להיות עם כל המידע של המערך מההתחלה ועד ל-32 byte.

ב. מהו גודל בלוק יחיד ב-CACHE?

נסתכל על החסם של 64. אנחנו רואים כי זמני הגישה גדלים עם גודל הקפיצה.

$$AMAT = Hit\ time + Miss\ Rate * Misspenalty$$

כלומר – התחלנו לפספס, וזמן הפספוס לכל ניסיון גישה לא משתנה – לכן רק אחוזי קצב הפספוס משתנה. תחת ההנחה הסבירה שזמן גישה הוא 10ns, (כך הבנו מסעיף א) אנו רואים שניתן לכתוב:

$$20 = 10 + Miss\ Rate_1 * Misspenalty ; 30 = 10 + Miss\ Rate_2 * Misspenalty$$

$$10ns = MR_1 * T ; 20ns = MR_2 * T$$

כלומר, קצב הפספוס גדל פי 2 בין צעד יחיד לצעד כפול.

כמו כן הוא גדל פי 4 לצעד של 4 ולא משתנה לצעד של 8. מכאן ניתן להסיק גודל בלוק של 4.

עבור צעד יחיד – יש פספוס כל byte ראשון ואז hit במשך 3 byte הבאים. המידע גם מתחלף במשך כל לולאה. על כן אחוזי הפספוס הינם 25% במקרה הנ"ל. עבור צעד כפול – פספוס כל byte שני. לכן 50% פספוס.

עבור צעד של 4 – כל byte מבוקש מביא איתו את ה-3 byte הצמודים לו אך הם אינם מענייננו שכן אנו קופצים בארבעה בתים ולכן יש 100% פספוס. וכנ"ל לצעד של 8. הצעדים שלאחר מכן יתבהרו בהמשך.

ג. מהו תצורת ה-CACHE n-way associative\directly mapped (אם n)?
למה שווה n)?

לפי מה שאמרנו יש ל-CACHE 32 byte ובבלוקים של 4 ולכן סה"כ שמונה בלוקים. המעבר בין זמן גבוה עבור חסם של 64 וצעד של 8 לזמן מאוד נמוך עבור אותו חסם בצעד של 16 נותן אינדיקציה טובה להבנה.

צעד של 8 ניגש למקומות ה-0,8,16,24,32,40,48,56. גם עבור directly mapped לכל אינדקס כזה ב-cache של 8 בלוקים נקבל אותו אינדקס ב-CACHE (כולם mod 8 = 0) (וגם נדרוש את המידע בתוך כל לולאה) אבל עבור צעד של 16 זה לא אמור לרדת פתאום. אלא, בצעד של 16 כל יחידת מידע אומנם ממופה לאותו אינדקס אך יש 4 סטים ולכן הם לא דורסים האחד את השני.

ד. מהם זמני Hit\Miss?

מתוך מה שאמרנו זמני הפגיעה והפספוס ברורים והינם 10ns ו-40ns בהתאמה. מבנה ה-CACHE:

	SET 1	SET 2	SET 3	SET 4
MOD2=0	☐	☐	☐	☐
MOD2=1	☐	☐	☐	☐

שאלה 2 (40 נקודות)

עבור מעבד ה-MIPS, נתונה טבלת ההשהיות הבאה של מרכיביו:

Element	Parameter	Delay (pSec)
register clock-to-Q	t_{pcq}	30

register setup	t_{setup}	20
multiplexer	t_{mux}	25
equality comparator	t_{eq}	40
AND gate	t_{AND}	15
ALU	t_{ALU}	200
memory read	t_{MEMread}	250
memory write	t_{MEMwrite}	220
register-file read	t_{RFread}	150
register-file write	t_{RFwrite}	100
register-file setup	t_{RFsetup}	20

א. חשבו מהו תדר השעון המרבי בו ניתן להפעיל את המעבד.

- תארו בפירוט מלא את חישוביכם.
 - רשמו ביטוי מפורש לזמן מחזור השעון במונחי הפרמטרים המופיעים בטבלה.
 - לאחר מכן השתמשו בערכיהם לחישובים מספריים.
- הנחיות:

1. היעזרו בתרשימים המצורפים וסמנו לכל דרגה ודרגה את מסלול החישוב הארוך ביותר. שימו לב שזהו תרשים של MIPS שנלמד בקורס, בתוספת MUX במוצא של ה REGISTER FILE, המשמשים לניתוב נתונים מקודמים (FORWARD) במקרה של HAZARD.
2. שימו לב שלחלק מהרגיסטרים שעון שונה מלאחרים. חלקם מגיבים לעליית שעון וחלקם לירידת שעון.

The clock cycle is dictated by the longest delay path of the processor.

Consider first the critical path at every pipeline stage. Such paths always start at the clock of a register and terminate at the data input of a register.

Notice that we need to be able to complete each task in half a cycle.

Instruction fetch stage:

$$T_{\text{cyc min IF}} = 2 \times (t_{\text{pcq}} + t_{\text{MEMread}} + t_{\text{setup}}) = 2 \times (30 + 250 + 20) = 600 \text{ pSec}$$

Decode stage:

$$T_{\text{cyc min ID}} = 2 \times (t_{\text{RFread}} + t_{\text{mux}} + t_{\text{eq}} + t_{\text{AND}} + t_{\text{mux}} + t_{\text{setup}}) = 2 \times (150 + 25 + 40 + 15 + 25 + 20) = 550 \text{ pSec}$$

Execution stage:

$$T_{\text{cyc min EXE}} = 2 \times (t_{\text{pcq}} + t_{\text{mux}} + t_{\text{mux}} + t_{\text{ALU}} + t_{\text{setup}}) = 2 \times (30 + 25 + 25 + 200 + 20) = 600 \text{ pSec}$$

Memory stage:

$$T_{\text{cyc min MEM}} = 2 \times (t_{\text{pcq}} + \max\{t_{\text{MEMread}}, t_{\text{MEMwrite}}\} + t_{\text{setup}}) =$$

$$2 \times (30 + 250 + 20) = 600 \text{ pSec}$$

Writeback stage:

$$T_{cyc \min \text{ WB}} = 2 \times (t_{pcq} + t_{mux} + t_{RFwrite}) = 2 \times (30 + 25 + 100) = 310 \text{ pSec}$$

$$T_{CLK} = \max T_{cyc \min} \{ \text{IF}, \text{ID}, \text{EXE}, \text{MEM}, \text{WB} \} = 600 \text{ pSec}.$$

ב. כמה זמן תימשך ריצת אוסף תכניות המכילות 100 ביליון פקודות מכונה. הניחו שמספר המחזורים הממוצע לפקודה הוא $\text{CPI} = 1.15$?

$$T_{\text{total}} = (100 \text{ billion instructions}) \times (1.15 \text{ cycles/instruction}) \times (600 \text{ pSec}) = 10^{11} \times 1.15 \times 600 \times 10^{-12} = 69 \text{ Sec}$$

ג. מהו ה **THROUPUT** של המערכת? מהו ה **LATENCY**?

כאן ניתנו נקודות על הסברים שונים כל עוד הם הוסברו.

ה **LATENCY** הוא זמן שלוקח לפקודה לעבור – 5 מחזורים.

ה **THROUGHPUT** לפקודה יחידה היא מחזור יחיד. אבל בממוצע

ה **THOUGHPUT** היה יותר ממחזור יחיד שכן **CPI** היה גדול מאחד.

ד. נמדוד את ביצועי המעבד ע"י מספר הפקודות לשנייה בתדר שעון מרבי. בניסיון לשפר ביצועים, ראובן, מהנדס מתחיל בחברה (בוגר בר-אילון) מציע להעביר את בדיקת תנאי השוויון של קפיצה מותנית משלב ה **ID** לשלב ה **EXE**, ואילו שמעון, מהנדס וותיק בחברה (בוגר אוניברסיטה אחרת) מציע להעביר את בדיקת תנאי השוויון לשלב ה **MEM**.

- האם כל אחת מההצעות הנ"ל תשפר, תקלקל או לא תשפיע על הביצועים? הסבר בפרוטרוט את תשובתך.

- במידה והביצועים משתנים, חשב באופן מספרי פי כמה $(T_{cycle}, \text{THROUGHPUT}, \text{LATENCY}, \text{cpi})$.

נתון BENCHMARK כדלהלן:

25% פקודות קריאה (LOADS), ב 40% מהן מופיעה מיד פקודה המשתמשת בנתון שנקרא.
10% פקודות כתיבה (STORES).
11% הסתעפויות (BRANCHES), ב 25% מתוכן ניחוש הקפיצה היה שגוי.
2% פקודות קפיצה. הנח שהפקודה שבאה מיד אחרי קפיצה מרוקנת (FLUSH) מהצינור.
52% פקודות אריתמטיות (R-TYPE).

The critical delay paths of the ID stage gets shorter by $t_{eq} + t_{AND} + t_{mux} = 80 \text{ pSec}$.

$$T_{cyc \min \text{ ID}} = 2 \times (t_{RFread} + t_{mux} + t_{setup}) = 390 \text{ pSec}$$

The critical path in the EXE stage gets longer by 80pSec at most in the first case and the critical paths in the MEM gets longer by 80pSec at most in the second case.

Execution stage:

$$T_{cyc \min \text{ EXE}} = 2 \times (t_{pcq} + t_{mux} + t_{mux} + t_{ALU} + t_{setup}) + 160 =$$

$$2 \times (30 + 25 + 25 + 200 + 20) + 160 = 760 \text{ pSec}$$

Memory stage:

$$T_{cyc \min \text{ MEM}} = 2 \times (t_{pcq} + \max\{t_{\text{MEMread}}, t_{\text{MEMwrite}}\} + t_{\text{setup}}) + 160 =$$

$$2 \times (30 + 250 + 20) + 160 = 760 \text{ pSec}$$

So it makes no difference which of the two – they will both dictate a slower clock rate!

$$\text{Perfbranch_ID original} = (1/(600 \times 10^{-12}))/1.1475 = 1.4524 \times 10^9 \text{ instructions/Sec}$$

Ignoring the branch prediction differences:

$$\text{Perfbranch_ID} = (1/(760 \times 10^{-12}))/1.1475 = 1.1467 \times 10^9 \text{ instructions/Sec}$$

Unfortunately we can't therefor:

If equality check takes place at EXE stage two instructions are flushed in miss:

$$\text{CPI_BRANCH} = 0.75 \times 1 + 0.25 \times 3 = 1.5.$$

$$\text{CPI} = 0.25 \times 1.4 + 0.1 \times 1 + 0.11 \times 1.5 + 0.02 \times 2 + 0.52 \times 1 = 1.175.$$

$$\text{Perfbranch_EXE} = (1/(760 \times 10^{-12}))/1.175 = 1.1198 \times 10^9 \text{ instructions/Sec}$$

If equality check takes place at MEM stage three instructions are flushed in miss:

$$\text{CPI_BRANCH} = 0.75 \times 1 + 0.25 \times 4 = 1.75.$$

$$\text{CPI} = 0.25 \times 1.4 + 0.1 \times 1 + 0.11 \times 1.75 + 0.02 \times 2 + 0.52 \times 1 = 1.2025.$$

$$\text{Perfbranch_MEM} = (1/(760 \times 10^{-12}))/1.2025 = 1.0942 \times 10^9 \text{ instructions/Sec}$$

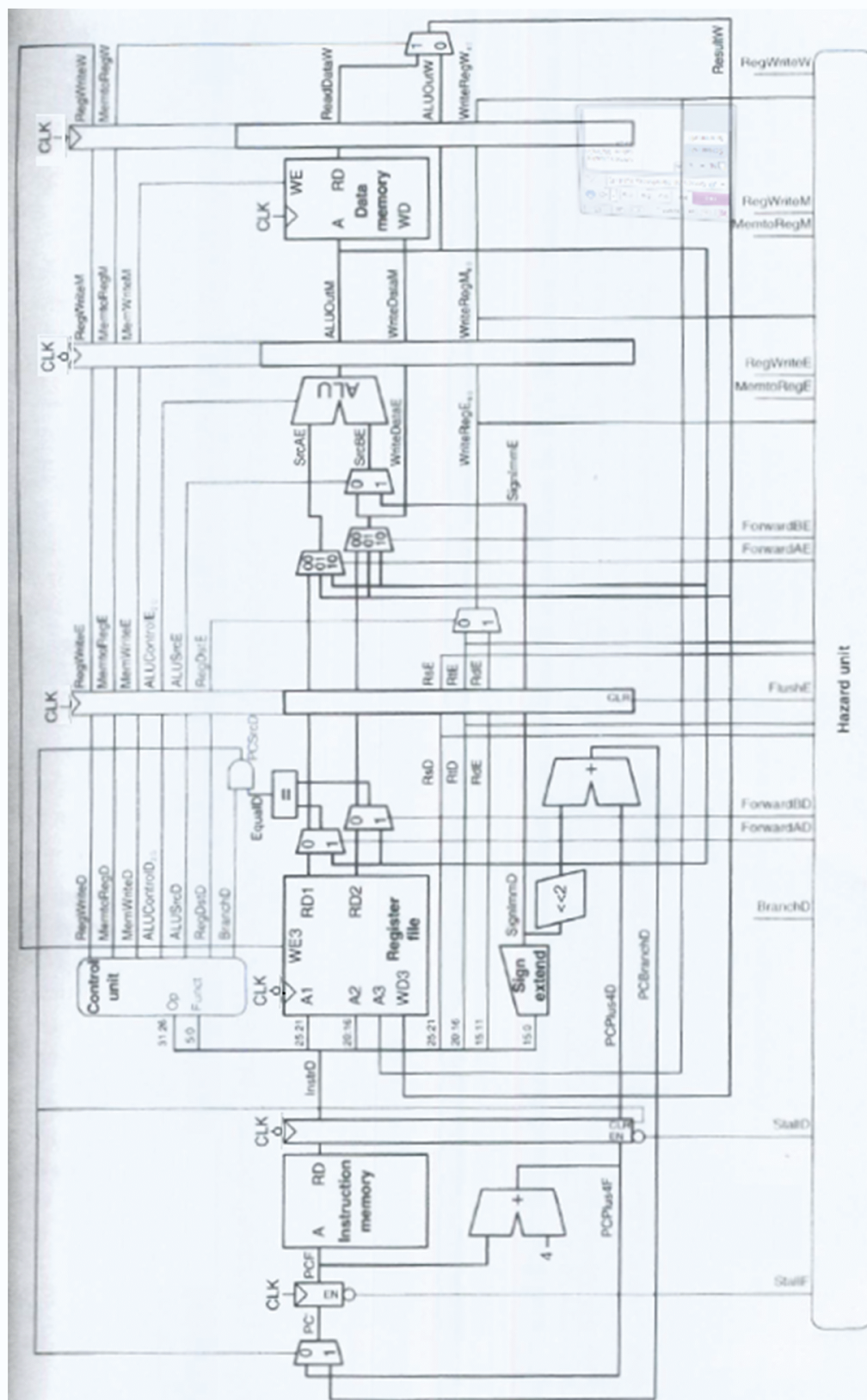


Figure 7.58 Pipelined processor with full hazard handling

שאלה 30 נק':

לרשותכם מערכת MIPS. רכיב הALU במערכת שלכם יכולה לבצע פקודות ADD, SUB, OR, AND.

נתון כי במקום מסוים בזכרון המכונה LABEL1 שמור ערך בזכרון X . כמו כן במקום אחר המכונה LABEL2 שמור הערך Y .

הכתובות בזכרון של X, Y שמורים ברגיסטרים $\$S0, \$S1$ בהתאמה.

א. כתבו תוכנית בשפת ASSEMBLY המבצעת את הפעולה $X(XOR)Y$.

דגשים:

- שימו לב לSYNTAX נכון ושימוש נכון ברגיסטרים הנכונים.
- זכרו ששיטת הקידוד בMIPS הוא במשלים ל-2.

ב. בכמה מחזורים יתבצע התוכנית בMIPS מסוג SINGLE CYCLE\MULTI CYCLE\PIPELINE?

מבחינת פעולות לביצוע NOT על מספר המיוצג בשיטת המשלים ל-2:

$$X + X' = -1 \implies X' = -X - 1$$

מכיוון שהם לא למדו את הפעולה la שעושה load של label לתוך רגיסטר אני אשנה את השאלה כך שרגיסטרים $\$S0, \$S1$ יכילו אותם בהתאמה.

```
la $S0, LABEL1
la $S1, LABEL2

lw $S0, 0($S0)           // s0 contains X
lw $S1, 0($S1)           // s1 contains Y

sub $t0, $ZERO, $S0
sub $t1, $ZERO, $S1

addi $t0, $t0, -1         // t0 contains X'
addi $t1, $t1, -1         // t1 contains Y'

and $t0, $t0, $S1         // to contains X'Y
and $t1, $t1, $S0

or $t0, $t0, $t1
```

את החישובים של מספר מחזורים לכל פקודה וכדו – לפי ההתפלגות הידועה. (בצינור זה מחזור לפקודה וכן single cycle וב multi לפי הנלמד בהרצאה).

במבחן אמרתי שלא חובה להתייחס כן HAZARDS. אבל, מי שיתחשב צריך להתחשב לו יותר בציון.