



הפקולטה להנדסה
ארכיטקטורה של מחשבים - פתרון מקוצר
תשע"ד סמס' ב' מועד א'
83-248

- מרצה : פרופ' שמואל וימר
- מתרגל : יוסף לונדון
- חומר עזר מותר : מחברות/ דפים מהרצאות, תרגולים ותרגילי בית, ספרים.
- משך המבחן שלוש שעות
- סך כל הנקודות הוא 110.
- 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 !

בהצלחה

שאלה מס' 1 (40 נק')

נתון מעבד בעל 2 רמות cache. גישה לזיכרון מרכזי לוקחת 100 מחזורי שעון.

א. מלאו ע"פ הטבלה הבאה את מספר הסיביות בשורות הריקות:

	L1	L2
Cache DATA Size	32KB	512KB
Block Size	8 Byte	32 Byte
Associativity	4 way	Direct
Hit Time	2 cycles	35 cycles
Miss Rate	10%	2.5%
Writing scheme	Write through	Write through
Replacement scheme	LRU	-----
Tag	19	13
Index	10	14
Offset	1	3

Size - 32KB will be divided into 8 Byte words therefore $2^{15}/2^3=2^{12}$.

If it were a direct cache there would be 2^{12} lines. 4 way associative so $2^{12}/2^2= 2^{10}$

we now know we have 2^{10} rows therefore **10** bits for the **INDEX**.

since there are 8 BYTES, equivalent to 2 32 bit words, we need **1** bits to define the **OFFSET**.

the remainder are the **TAG** therefore $32-2-10-1=19$

Size - 512KB will be divided into 32 Byte words therefore $2^{19}/2^5=2^{14}$.

If it were a direct cache there would be 2^{14} lines.

we now know we have 2^{14} rows therefore **14** bits for the **INDEX**.

since there are 32 BYTES, equivalent to 8 32 bit words, we need **3** bits to define the **OFFSET**.

the remainder are the **TAG** therefore $32-2-14-3=13$

ב. מהו ה-AMAT של כל CACHE

Average Memory Access Time = HitTime+MissRate*MissPenalty

$$AMAT_{L1} = 2 + 0.1 \cdot 35 = 5.5$$

$$AMAT_{L2} = 35 + 0.025 \cdot 100 = 37.5$$

ג. מהו ה-AMAT הכולל.

$$AMAT = HitTime + MR \cdot MP$$

$$AMAT = 2 + 0.1(35 + 0.025 \cdot 100) = 5.75$$

ד. ללא L2 מה היה ה-AMAT הכולל?

$$AMAT = 2 + 0.1 \cdot 100 = 12$$

ה. כמה סיביות נצרכות עבור L1 למעקב כתיבה לצורך שימוש בשיטת LRU כשיטת הכתיבה ?
 רשום קידוד לדוגמא שיבטא את השיטה הנ"ל עבור CACHE הנ"ל. בכל כתיבה, כמה ביטים
 סה"כ נצטרך לעדכן?

4 way associative therefore 2 bits needed. For example 'last used'=00,
 'penultimate(second to last)'=01 etc..

all the bits in their write procedure field must be redone when writing a new word to
 the cache or when accessing one. We must keep proper track of the order.

מפאת בעיות בביצועים החליטו להוסיף עוד 2 רמות ל-CACHE הקיים:

	L3	L4
Cache DATA Size	512KB	512KB
Block Size	4 Byte	4 Byte
Associativity	Direct	Direct
Hit Time	???	???
Miss Rate	5%	10%
Writing scheme	Write through	Write through
Replacement scheme	Always replace	Always replace
Tag	13	13
Index	17	17
Offset	----	----

ו. השלימו השדות החסרים (כמו בסעיף א').

Size - 512KB will be divided into 4 Byte words therefore $2^{19}/2^2=2^{17}$.

If it were a direct cache there would be 2^{17} lines.

we now know we have 2^{17} rows therefore **17** bits for the **INDEX**.

since there are 4 BYTES, equivalent to 1 32 bit words, we need **0** bits to define the
OFFSET.

the remainder are the **TAG** therefore $32-2-17=13$

ז. נתון כי לאחד זמן גישה של 100 מחזורים ולשני של 35. למי איזה? הסבירו למה.

The faster cache will be the one closer to the lower levels, therefore L3 access time will
 be 35 cycles (like L2) and L4 will have an access time of 100 cycles.

ח. הניחו כי בסעיף הקודם חלה טעות וכי אחד מזמני הגישה צריך תיקון פי 80%. איזה מהם ולמה
 לדעתכם?

There is no reason to add an entire cache system that is just as slow as the access time
 to the main memory, there is nothing to be gained yet much to lose (real estate on the
 cycles.80chip). Therefore, L4 access time is reduced to

ט. מתוך מה שלמדתם בקורס, איזה תכונה של זכרון מנצלת L3 ואיזה תכונה של זכרון מנצלת L2?
 Temporal locality (L3) vs spatial locality (L2)

י. מהו ה-AMAT הכולל החדש? האם יש שיפור ביצועים למחשב? הציעו שיפור אפשרי אחר במקום
 4 רמות CAHCE. מותר לכם לשנות נתונים ב-CACHE קיים אך תצטרכו להסביר כיצד.

$$AMAT = HitTime + MR \cdot MP$$

$$AMAT = 2 + 0.1(35 + 0.025 \cdot (35 + 0.05 \cdot (80 + 0.1 \cdot 100))) = 5.59$$

As we can see, the improvement was small. It can be seen as well that the 4th cache made almost no impact whatsoever. By letting go of the 4th cache and enlarging the third, we may be able to diminish the miss rate of the third and thus improve the average miss access time.

שאלה מס' 2 (30 נק')

נתונה מכונת ה MIPS שלהלן. מתבצע קטע הקוד הבא:

```
lw $s0, 24($t3)
sub $t4, $s0, $t5
add $t6, $t7, $t1
sw $t4, 20($t7)
add $t8, $t2, $t6
```

בהמשכו הוכנסה סדרה של חמש פקודות NOP.

א. אילו אוגרים נקראים ואילו אוגרים נכתבים במחזור שעון 3, 4, 5, 6, 7 ?
יש לציין את שמות האוגרים הנכתבים והנקראים לחוד (ציינו ההבדל בין קריאה לכתיבה מבחינת דגימת מידע ובאיזה מחזור), ולכל מחזור לחוד. כמו כן – במידה ויש FORWARDING, ציינו מתי קורה כל חלק ב-FORWARDING וציינו מי מועבר לאן.

מחזור שעון שלישי: הפקודה השנייה (sub \$t4, \$s0, \$t5) נמצאת בשלב ID, על כן נקראים רגיסטרים \$t5 ו-\$s0 במחזור שעון זה אין פקודה בשלב WB ולא נכתב מידע באף רגיסטר.

מחזור שעון רביעי: הפקודה השנייה (sub \$t4, \$s0, \$t5) תלויה בפקודה שלפניה ב-LOAD HAZARD ולכן ייכנס NOP. על כן נקראים רגיסטרים \$t5 ו-\$s0 שוב. במחזור שעון זה אין פקודה בשלב WB ולא נכתב מידע באף רגיסטר.

מחזור שעון חמישי: הפקודה (add \$t6, \$t7, \$t1) נמצאת בשלב ID, על כן נקראים רגיסטרים \$t1 ו-\$t7. במחזור שעון זה גם מועבר מידע מיחידת ה-FORWARDING לגבי רגיסטר \$s0 ליחידת החישוב (ALU) לצורך הפקודה השניה. במחזור שעון זה הפקודה הראשונה נמצאת בשלב ה-WB ועל כן נכתב רגיסטר \$s0 (בעליית שעון הבא המידע יידגם).

מחזור שעון שישי: הפקודה (sw \$t4, 20(\$t7)) נמצאת בשלב ID, על כן נקרא רגיסטר \$t7.

מחזור שעון שביעי: הפקודה (add \$t8, \$t2, \$t6) בשלב ה ID, על כן נקראים רגיסטרים \$t2 ו-\$t6. הרגיסטר הנכתב הוא \$t4. נשים לב שיש כאן FORWARDING משלב ה WB של פקודה 2 לשלב ה EX של פקודה 4.

ב. מה מבצעת יחידת ה FORWARDING במחזור 8? אלו רגיסטרים נקראים ואלו נכתבים? במידה ומתבצעות השוואות כלשהן, ציין מהן.

מחזור שעון שמיני: הפקודה NOP בשלב ה ID, על כן נקרא רגיסטר האפס. הרגיסטר הנכתב הוא \$t6. נשים לב שיש כאן FORWARDING משלב ה WB של פקודה 3 לשלב ה EX של פקודה 5 כיוון שרגיסטר \$t6 צריך להיות מעודכן.

יחידת ה FORWARDING בודקת אם הרגיסטרים הנקראים בפקודה add \$t8, \$t2, \$t6 מעודכנים. מכיוון שמדובר ב- pipelined MIPS יתכן שאחת או שתי הפקודות הקודמות אמורות לשנות את ערך הרגיסטרים הנקראים, אך עוד לא הספיקה לכתוב את הערך החדש ב- register file. לכן יש להשוות את 2 כתובות הרגיסטרים הנקראים לכתובת הרגיסטרים הנכתבים ב-2 הפקודות הקודמות. יחידת ה FORWARDING שולטת על הכניסות ל-ALU ומוודאת שהערך העדכני של הרגיסטרים הוא הערך בו נשתמש בחישוב.

ההשוואות הינן:

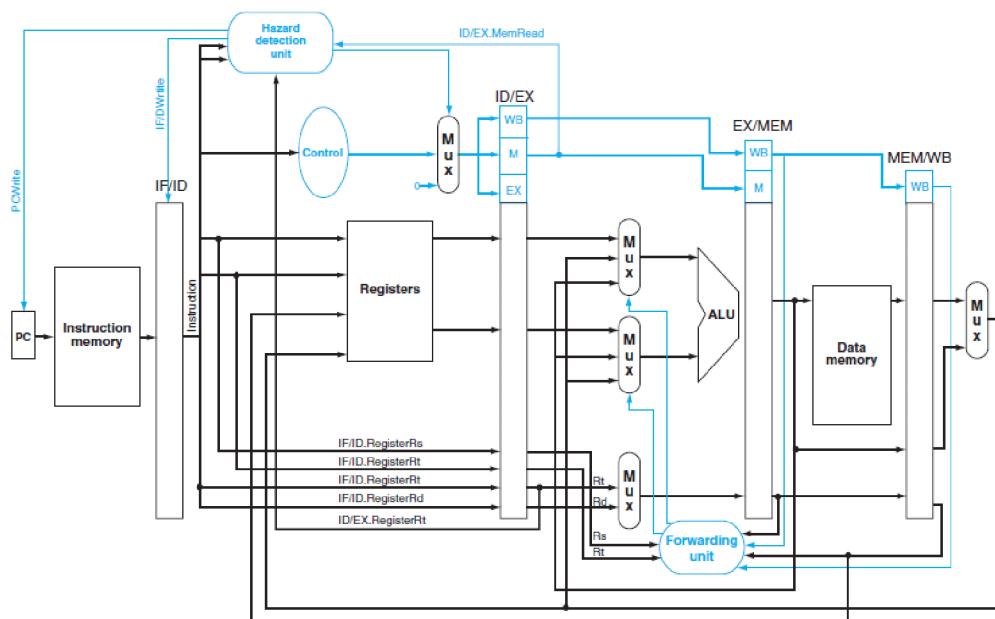
none=2 ?

none=6 ?

6=2 ?

6=6 ?

	stages	IF	ID	EX	MEM	WB	
cycles	1	lw(1)	-----	-----	-----	-----	
	2	sub(2)	lw(1)	-----	-----	-----	
	3	add(3)	sub(2)	lw(1)	-----	-----	
	4	add(3)	sub(2)	NOP	lw(1)	-----	
	5	sw(4)	add(3)	sub(2)	NOP	lw(1)	
	6	add(5)	sw(4)	add(3)	sub(2)	NOP	lw(1)
	7	NOP	add(5)	sw(4)	add(3)	sub(2)	NOP
	8	NOP	NOP	add(5)	sw(4)	add(3)	sub(2)



שאלה מס' 3 (40 נק')

למעבד MIPS single cycle פקודה חדשה `branch sub equal` כאשר ה syntax שלה הוא:

`bsq $rs, $rt, $rd, target`

פקודה זו קופצת ל `target` מסוים אם ורק אם תוכן הרגיסטרים מקיים:

`$rd=$rs-$rt`

א. איזה סוג פקודה יכולה להכיל פקודה כזו עם הכי מעט שינויים למבנה (לא ערכים) שדות הפקודה? `RTYPE, JTYPE, ITYPE`? הסבירו.

סוג הפקודה היא `RTYPE` מכיוון שיש כאן קבלה של 3 רגיסטרים כאופרנדים והפקודות האחרות אינן בנויות לזה. נתבקשתם לא לשנות את מבנה השדות ולכן זו התשובה הנכונה.

ב. אילו שדות צריכים שינוי בכדי לקיים פקודה שכזו? לאיזה מרחק בזיכרון ממיקומו הנוכחי של ה PC ניתן לקפוץ? הסבירו בפרוטרוט.

השדה היחיד שפנוי לשינוי (אחזקת מידע) הוא שדה ה `shift` שבמקום לבצע את הפעולה הנ"ל ננצלה לתת אינדיקציה ליעד קפיצה. מספר הסיביות הוא 5 ולכן זה יגדיר לנו את גודל הקפיצה האפשרי. כלומר $(pc+4+4*(shift\ field))$ כאשר השדה הנ"ל יכול להכיל כתובות $(2^{n-1} - 1)$ (1 כלומר מספרים הנעים מ 15:16 – שורות זכרון ואם נכפיל ב 4 כי יש לנו הכפלה כזאת תמיד כי מדברים בבתים כשקופצים בזכרון הפקודות) אזי 64:60 – בתים.

כמובן שנצטרך לשנות גם את שדה ה `opcode` (ולכן יתקבלו כאן גם תשובות אחרות מאשר `jump`) או להשתמש באינדיקטור נוסף מתוך שדה ה `function` שלא בשימוש בכדי שהבקר יוציא אינדיקציה להדליק את אפשרות הקפיצה כלומר את ה `jump`. ולכן התשובות כאן מגוונות.

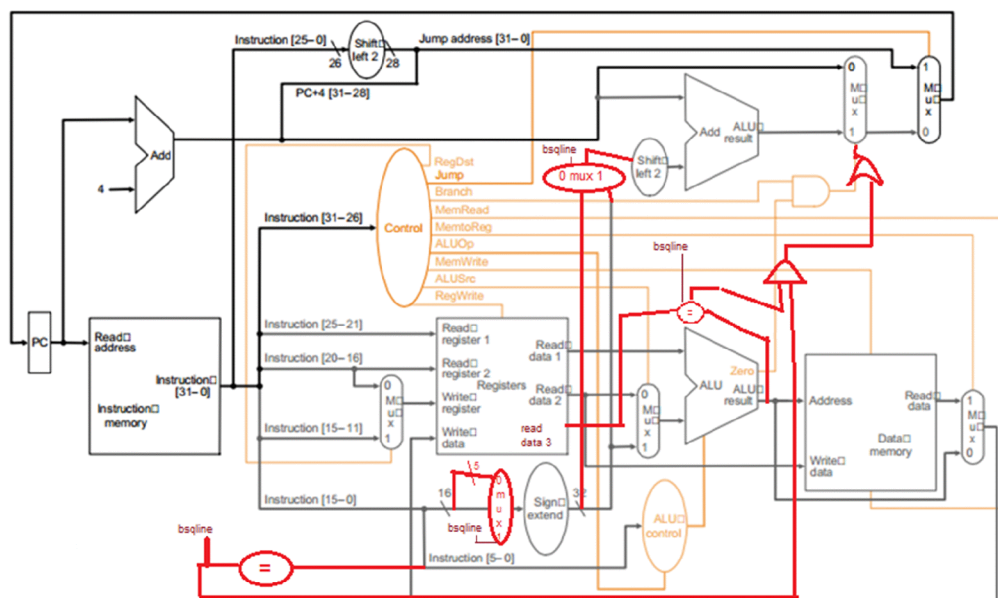
ג. אלו שינויים ליחידות אריתמטיות נצרכות (גם תוספות נחשב שינוי) ו/או ליחידות אחרות הקיימות (כגון REGISTER FILE, הזכרון וכדו') בכדי לממש את הפקודה. דייקו והסבירו לפרוטרוט מה הוספתם למערכת הקיימת (כולל בוררים שערים וכדו').

כאן נדגים את האפשרות של func שלא בשימוש בALU (במידה והכל בשימוש נצטרך פתרון אחר כמו עם opcode חדש) שמשנה את הפעולה של הALU לעשות חיסור כמו גם מוציא קו בקרה ממשוון במקרה שזה שדה הfunc שקיבלנו בכדי להדליק את שדה הBRANCH או לחילופין (כמו שנראה) אלטרנטיבה לBRANCH. השיטה שייצגתי היא בניית מעגל בקרה חיצוני בעצם. הנקודות החשובות ביותר להבין כאן היא הצורך בפורט נוסף וחישוב אחר חישוב (עוד מערכת חישובית). הוספנו פורט שלישי לקריאה כי אין כזה וכאן **קוראים** 3 רגיסטרים וכפי שאמרנו בכיתה לא ניתן לעשות זאת כרגע בsingle cycle. כמובן הוספנו גם בוררים (לshift ולsign extension) כדי שלא ייקח את הסיביות שהיו מהווים לפקודת immediate.

ד. ציירו על התרשים של single cycle mips את השינויים לביצוע כדי לקיים את הפקודה כפי שעניתם בסעיפים הקודמים. הסבירו.

נשים לב שנצרכו המון שינויים! הוספנו משוון בין תוצאת הALU (החיסור) לבין הרגיסטר השלישי. הוספנו קו בקרה בכדי שלא יפריע לפעילות תקינה של פקודות אחרות, שיוצאת ממעגל הבקרה שבנינו. תשובות אחרות שיתקבלו יכולים להיות מסגנון של הוספת מצב לOPCODE.

Single Cycle MIPS:



ה. כיצד ישפיעו השינויים שבניתם מבחינת CLOCKCYCLE של המערכת?

מכיוון שיש לנו מסלול נתונים ארוך יותר – בגלל שיש לנו שלב חישוב נוסף אחרי שלב החישוב שהיה לפקודות RTYPE רגילות – זמן המעבר במערכת גדל ולכן זמן המחזור גדל וקצב השעון ירד. יש לבחון האם הוא ארוך יותר מגישה לזכרון. במידה ולא – הרי מקצב השעון לא ישתנה.