



הטכניון - מכון טכנולוגי לישראל

מבנה מחשבים ספרתיים (234267)

מבחן מסכם מועד ב'

3 מרץ 2017

מרצים: ליהוא רפופורט, עדי יועז.

מתרגלים: פרנק סלה, איתי רביד.

שם :	_____
מס. ת.ז. :	_____

- משך הבחינה: שלוש שעות.
- מותר כל חומר עזר.
- יש לכתוב את התשובות בטופס הבחינה בלבד ובמקום המיועד לתשובה.
- יש לכתוב בקיצור ככל האפשר, אך יש לנמק כל תשובה.
- בדקו שבטופס שבידכם יש 14 עמודים כולל עמוד זה.
- המבחן כולל 5 שאלות, יש לענות על כולן.

שאלה 1	/ 22
שאלה 2	/ 28
שאלה 3	/ 25
שאלה 4	/ 15
שאלה 5	/ 10
ציון סופי	/100

בהצלחה !

שאלה 1 – זיכרון וירטואלי (22 נק')

נתון מעבד בעל מבנה הכתובת הבא:

63	12+3n	12+2n	12+n	12	0
Sign Ext	PML3	PML2	PTE	offset	

- גודל כל אחת מטבלאות הדפים, בכל רמות התרגום, היא כגודל דף.
- בכל כניסה בטבלאות הדפים, כל הסיביות משמשות עבור PFN בלבד.
- גודל המרחב הפיזי הנתמך הוא 2^{20} בתים.

א. (4 נק') מהו m ? פרטו והסבירו.

- ב. (10 נק') במעבד יש TLB גדול, PMH עם translation caches גדולים עבור כל-אחת מרמות התרגום (שהגישה אליהם מתבצעת במקביל), ו-data cache גדול בעל גודל שורה של 64 בתים. בטבלה שלהלן נתונה סידרת פניות לכתובות וירטואליות בבסיס 16. עבור כל אחת מהפניות ציינו:
- עבור כל אחד מה-translation caches וה-TLB האם הניב hit או miss, או שלא ניגשו אליו.
 - מספר הגישות ל-Data cache שהסתיימו ב-hit, ומספר הגישות שהסתיימו ב-miss (עבור התרגום בלבד, לא כולל הפניות להבאת הנתון עצמו).
- בשורות בהן לא רשומה כתובת, רישמו כתובת שעבורה יתקיימו הנתונים הכתובים כבר בטבלה, והשלימו את שאר הנתונים. **בכל שורה מותר לשנות רק סיפורה הקסדיצמאלית אחת ביחס לכתובת בשורה שקודמת לה.**

הניחו כי בתחילת הסידרה כל ה-caches ריקים.

D\$ # misses	D\$ #hits	PML3 hit/miss/n.a.	PML2 hit/miss/n.a.	TLB hit/miss	כתובת
					FFFF CBA9 8765 4321
0	0				
0	1				
2	1				
1	1				

ג. (4 נק') מערכת ההפעלה מקצה 1Mbyte מתוך הזיכרון הפיסי עבור טבלאות הדפים. במערכת רצים 128 תהליכים, שכל אחד מהם ניגש לקטע זיכרון וירטואלי רציף בגודל 2^{25} בתים. האם הזיכרון הפיסי המוקצה עבור טבלאות הדפים מספיק עבור כל התהליכים? פרטו והסבירו

ד. (4 נק') מה הם השינויים הנדרשים על מנת להגדיל את הכתובת הפיזית הנתמכת ל-50 סיביות ?
במידה ונדרש גם שינוי במבנה הכתובת, הוסיפו ציור של מבנה הכתובת החדש. אין לשנות את גודל
טבלאות הדפים (בבתים), אין להקטין את אורך הכתובת הוירטואלית, ואין לשנות את גודל הדפים.

שאלה 2 – זיכרון Cache (28 נק')

נתון מעבד בעל הירארכית הזיכרון הבאה:

- L1 I-Cache (I\$): Direct Mapped, 32 בתים בשורה, גודל 4KBytes. מוכל ב-L2.
- L1 D-Cache (D\$): 2-way set associative, 32 בתים בשורה, גודל 8Kbytes, מדיניות החלפה LRU. מוכל ב-L2.
- L2 cache: 2-Way set associative, 32 בתים בשורה, גודל 128KBytes, מדיניות החלפה LRU. פעולת ה-Lookup ב-L2 מתבצעת במקביל ל-Lookup ב-L1 כדי למקבל את פעולת החיפוש בשתי רמות ה-Cache על כן ה-LRU ב-L2 Cache מעודכן בכל פעם שיש Lookup ב-L2 cache ללא קשר לתוצאה באחד מה-L1 caches.

נתון קטע הפסאודו קוד הבא בשפת מכונה לתוספת פאקטור קבוע בערך 5 נקודות לציוני הסטודנטים בקורס ממ"ס. כל פקודה היא באורך קבוע של 4 בתים והקוד מתחיל בכתובות 0x10000000 הציונים מאורגנים בזיכרון בצורת מערך דו-מימדי רציף בזיכרון המתחיל בכתובת 0x00001000 (הקידומת 0x מציינת מספר בבסיס 16).

1) Mov	R1, 5	// R1=5 (הקבוע 5 הוא בבסיס 10)
2) Mov	R7, 1600	// (הקבוע 1600 הוא בבסיס 10)
3) Xor	R2, R2	// R2=0
4) Xor	R3, R3	
5) Sll	R5, R3, 2	// 2 bit Shift left logical
6) Mul	R6, R2, R7	// R6=R2*R7
7) Add	R4, R5, R6	// R4=R5+R6
8) LD	R0, [R4]	// Load 4 Bytes
9) Add	R0, R0, R1	
10) ST	[R4], R0	// Store 4 Bytes
11) Add	R3, R3, 1	
12) Bne	R3, 400, 0x10000010	// Branch if not equal (הקבוע 400 הוא בבסיס 10)
13) Add	R2, R2, 1	
14) Bne	R2, 100, 0x1000000c	// (הקבוע 100 הוא בבסיס 10)

נתון כי בתחילת ביצוע התכנית כל זיכרונות המטמון ריקים.

א. (5 נק') חשבו את מספר השורות היכולות להיות מאוכסנות בכל אחת מרמות ה- L_2 Cache, L_1 ו- $D\$$. וכן חשבו את מספר השורות הדרושות לאחסן את המערך הדו-מימדי בזיכרון (גודל כל שורה היא 32 בתים).

ב. (5 נק') עבור ביצוע קטע הקוד כולו, מהו ה- hit rate ב- $D\$$ (יש להתחשב בכל הגישות ל- $D\$$ גם קריאה וגם כתיבה)? הסבירו

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

ד. (5 נק') במידה ומשנים את אסוציאטיביות ה- L2 cache להיות Direct Mapped מבלי לשנות את גודלו ומבלי לשנות את גודל השורה שלו מה יהיה מספר ה- Misses המדויק ב-\$I עבור ביצוע קטע הקוד כולו? הסבירו

לשם פשטות נניח שהמעבד אינו pipelined. כל פקודת Load ו Store מייצרת פניה ל-D\$ וכל קריאת פקודה בודדת מייצרת פניה לשורה ב-\$I, שבה מביאים את הפקודה המבוקשת באיטרציה הנוכחית. במידה ונוצרים קונפליקטים בין הקוד ל-Data ב-L2 קחו זאת בחשבון.

ה. (6 נק') כעת נחזור לארגון ה- L2 cache כמו בתחילת השאלה כשהוא 2-Way set associative, מהו ה- hit rate ב- L2 cache עבור ביצוע קטע הקוד כולו?
את ה- hit rate ב- L2 יש לחשב כאחוז מסך החיפושים (Lookups) שהסתיימו כ- miss באחד ה- L1 Caches.

שימו לב כי לא ניתן להזניח את ההשפעה של פניות להבאת קוד מה- L2 cache.

שאלה 3 – Out-Of-Order Execution (25 נק')

יש למלא את הטבלה שבהמשך. לכל פקודה יש לרשום:

- R3, R2, R1 – ערכי הרגיסטרים הארכיטקטוניים לאחר commit של הפקודה.
- addr – כתובת הגישה לזיכרון – עבור פקודות load ו-store בלבד.
- data – ערך זיכרון שנקרא או נכתב – עבור פקודות load ו-store בלבד.

כבר מולאו {

- T alloc: הזמן בו מבוצעת אלוקציה: עד 3 פקודות בכל מחזור, החל מ- $t=1$. ניתן לבצע אלוקציה רק כאשר לכל הפקודות שסיפק ה-frontend במחזור יש מקום הן ב-ROB והן ב-RS.
- ב-ROB יש 10 כניסות, וב-RS יש 6 כניסות. Store תופסת מקום אחד ב-ROB וב-RS.
- src1, src2: מספרי הרגיסטרים המשמשים כ-sources לפקודה: P_i עבור רגיסטר פיזי, ו- R_i במידה וקוראים ישירות את הרגיסטר הארכיטקטוני.
 - עבור store: src1 – הרגיסטר המשמש לחישוב הכתובת. src2 – הרגיסטר המכיל את הנתון.
 - עבור פקודת JZ (jump if zero): פקודת קפיצה מותנית: בצע קפיצה אם Zero flag=1.
 - פקודת ALU (Add, Sub) שתוצאת החישוב שלה היא 0 מעדכנת את הדגל ל-1 ואחרת ל-0.
 - הדגל מהווה dst נוסף של פקודות ALU, ומשמש כ-src עבור פקודת JZ.
 - הדגל עובר renaming, בדומה לרגיסטרים: PZi אם נכתב ע"י פקודה i, ו-Z אם ארכיטקטוני.
- T src1 ready, T src2 ready: הזמן בו מוכן כל אחד ערכי ה-sources לפקודה.
 - אם ה-src כבר מוכן בזמן האלוקציה, אז זמן זה יהיה שווה לזמן האלוקציה.
 - אחרת, זמן זה שווה ל-T data ready של הפקודה שמחשבת את הערך של ה-src.
- T exe: הזמן בו הפקודה נשלחת לביצוע:
 - פקודה יכולה להיכנס לביצוע לכל המוקדם במחזור שלאחר האלוקציה, לאחר המחזור בו כל ה-srcים מוכנים (עבור store : לאחר המחזור בו src1, המשמש לחישוב הכתובת, מוכן), וכאשר קיימת יחידת ביצוע מתאימה פנויה
 - במעבד קיימות יחידות הביצוע הבאות:
 - יחידת add/sub עבור ביצוע פעולות add/sub. משך הביצוע: מחזור 1.
 - יחידת AGU עבור חישוב כתובת של פעולות load/store. משך הביצוע: מחזור 1.
 - יחידת jump עבור ביצוע פעולות jump. משך הביצוע: מחזור 1.
 - יחידת DIV עבור ביצוע פעולות DIV. משך הביצוע של פקודת DIV נתונה הוא 5 מחזורים. היחידה היא pipelined, כך שבכל שני מחזורים ניתן להתחיל פקודה חדשה.
 - פקודת מוצאת מה-RS במחזור שלאחר סיום הביצוע שלה, וכבר במחזור זה פקודה חדשה יכולה לבצע alloc ולהשתמש במקום שהתפנה.
- Load block code: עבור load שנשלח לביצוע בזמן $t=T_{exe}$, או שהוסר עבורו תנאי חסימה קודם בזמן t , תנאי החסימה נבדקים בזמן $t+1$ לפי הסדר (רישמו את כל תנאי החסימה לפי הסדר):
 - 1 – חסימה כתוצאה מ- unknown store address
 - 2 – חסימה כתוצאה מ- waiting for store data

- T data ready:
 - עבור פקודות add/sub: $T_{exe}+1$. עבור פקודות div: $T_{exe}+5$.
 - עבור load שהוסרו עבורו כל תנאי החסימה בזמן t או שבוצע בזמן t ולא נחסם:
 - במידה וה-load פוגע ב-cache, או שיש store to load forwarding: בזמן $t+4$.
 - אחרת, במידה ובוצע load אחר לאותה שורה ב-cache בזמן $t' < t$: בזמן $\max(t'+10, t+4)$.
 - אחרת, בזמן $t+10$.
 - עבור store: מחזור השעון בו הן ה-data לכתיבה לזיכרון והן הכתובת מוכנים.
 - כלומר $T \text{ data ready} = \max(T_{exe}+1, T \text{ src2 ready})$.
 - הכתובת של ה-store ידועה בזמן $t = T_{exe}+1$. בזמן זה מוסר תנאי החסימה של load שנחסם ע"י ה-store על unknown store address. בפרט, load שמבוצע בזמן t, לא נחסם על unknown store address ע"י ה-store, אך load שמבוצע בזמן $t < t$ כן נחסם.
- עבור פקודת Jump עם חיזוי שגוי, מבוצע flush בזמן $T_{exe}+1$, והפקודות מהמסלול הנכון מבצעות אלוקציה החל מזמן $T_{exe}+6$ (במידה ואין סיבה אחרת שמעכבת את האלוקציה).
- T commit: הזמן בו הפקודה מבצעת commit. ניתן לבצע commit לעד 3 פקודות בכל מחזור.
 - פקודה יכולה לבצע commit החל מזמן $T \text{ data ready}+1$.
 - פקודת מוצאת מה-ROB במחזור שלאחר commit ($T_{commit}+1$), וכבר במחזור זה פקודה חדשה יכולה לבצע alloc ולהשתמש במקום שהתפנה.
- RS entries – מספר הכניסות התפוסות ב-RS לאחר האלוקציה של הפקודה הנתונה (בהתחשב גם בפקודות שהוצאו מה-RS במחזור זה).
- ROB entries – מספר הכניסות התפוסות ב-ROB לאחר האלוקציה של הפקודה הנתונה (בהתחשב גם בפקודות שהוצאו מה-ROB במחזור זה).
- הנחות:
 - הכתובות הן פיזיות (אין צורך בתרגום). כל הערכים המספריים הם בבסיס 16.
 - L1 data cache: גודל שורה $20_{16}B = 32_{10}B$. write no allocate. ריק בתחילת הביצוע.
 - הניחו כי ה-frontend יכול לספק 3 פקודות בכל מחזור.
 - בטבלה רשומות אך ורק הפקודות מהמסלול הנכון.

Pdst	instruction	R1	R2	R3	addr	data	src1	src2	T alloc	T src1 ready	T src2 ready	T exe	block code	T data ready	T commit	#RS entries	#ROB entries
0	Div R1 ← R2 / 2	40	80	160													
1	store m[R1+10] ← R3	40	80	160	50	160											
2	load R2 ← m[R2+10]	40	60	160	90	60											
3	Div R1 ← R3 / 2	80	60	160													
4	store m[R1-10] ← R2	80	60	160	70	60											
5	load R1 ← m[R3-90]	60	60	160	70	60											
6	sub R3 ← R3 – 20	60	60	140													
7	JZ 1000 wrongly predicted	60	60	140													
8	add R3 ← R3 – 20	60	60	120													

שאלה 4 – Power/Performance (15 נק')

נתונה מערכת Thin and light Notebook בעלת שני Core's במסגרת מעטפת הספק של 9Watt כאשר שני שליש מתקציב ההספק הינו עבור ה-Core's (והשאר עבור ה-Uncore). נתון כי המעבדים אינם תומכים ב-Multi-Threading, היינו מסוגלים להריץ כל אחד Thread אחד. המערכת תוכננה להריץ בו זמנית שני Threads בתנאי TDP.

נתון כי כל מעבד הוא בעל הנתונים הבאים:

- הקיבול הדינאמי של כל Core נתון כפונקציה של ה-IPC של האפליקציה המורצת: $C_{dyn} = IPC \times 700pF$
- שטח כל Core הינו: $4mm^2$
- ההספק סטטי (Leakage Power) ליחידת שטח: $0.165W/mm^2$
- אופיין מתח/תדר היינו נקודות מתח ותדר אפשריות, עם ההספק הדינאמי המתאים לכל נקודה נתונות בטבלה שלהלן.

מתח ב Volt's	תדר ב Ghz	Pdyn
0.60	1.0	0.50
0.65	1.1	0.65
0.70	1.3	0.89
0.75	1.5	1.18
0.80	1.8	1.61
0.85	2.3	2.33
0.90	2.8	3.18
1	3.4	4.76
1.1	4.0	6.78

א. (5 נק') מהו "תדר המדבקה" Sticker Frequency (guaranteed frequency) של כל מעבד במערכת הנתונה? יש לפרט את החישוב ולהסביר.

ב. (3 נק') מהו ה-IPC של כל מעבד במערכת הנתונה עבור אפליקציית TDP ? יש לפרט את החישוב ולהסביר

ג. (7 נק') כעת משפרים את מנגנון ה- Prefetch של המעבדים כך שה- IPC עבור אפליקציית TDP גדל ב-15%, השינוי המתואר יוצר אופיין חדש של מתח/תדר new v/f curve מה יהיה ה-guaranteed frequency החדש ומה יהיה השיפור בביצועים Speedup של המעבד המשופר אם ידוע כי עתה מתח העבודה בנקודת ה-TDP יהיה 0.8v? יש להסביר

שאלה 5 – חיזוי קפיצות (10 נק')

א. (5 נק') נתון חזאי קפיצות מסוג gshare בעל היסטוריה באורך 3. המצביע למערך החיזוי מחושב ע"י ביצוע XOR בין ההיסטוריה לבין 3 הסיביות התחתונות של הכתובת בה נמצא ה-jump. כל איבר במערך החיזוי הוא בן סיבית אחת, ומאותחל ל-0.

ABABAB

011010 011010 011010 ...

נתונה סידרת הפניות הבאה:

האיברים האי-זוגיים בסדרה שייכים ל-jump A הנמצא בכתובת שהסיביות התחתונות שלה הן 000. האיברים הזוגיים בסדרה שייכים ל-jump B הנמצא בכתובת שהסיביות התחתונות שלה הן 111. (תזכורת: $1 \text{ xor } X = \text{not } X$, $0 \text{ xor } X = X$).

יש למלא את הטבלה הבאה, כאשר העמודה הראשונה (כבר מולאה) מתייחסת לחיזוי האיבר הרביעי בסדרה (0 ששייך ל-jump שבכתובת 111), העמודה השנייה לחיזוי האיבר החמישי, וכו'.

001	100	010	101	110	011	001	100	010	101	110	011	היסטוריה
000	111	000	111	000	111	000	111	000	111	000	111	סיביות כתובת
											100	מצביע לחזאי מס'
											0	חיזוי
											0	קפיצה בפועל
											נכון	חיזוי נכון/שגוי

ב. (5 נק') עבור אותו חזאי, נתונה סידרת הפניות הבאה: 00001 00001 00001 ...
הסידרה שייכת כולה לאותו jump יחיד, שנמצא בכתובת שהסיביות התחתונות שלה הן 010.
מהו אחוז החיזוי הנכון במצב היציב עבור סידרה זו? יש להסביר
