הטכניון - מכון טכנולוגי לישראל

מבנה מחשבים ספרתיים (234267)

מבחן מסכם מועד א'

31 ינואר 2016

<u>מרצים</u>: **ליהוא רפופורט, עדי יועז.**

<u>מתרגלים</u>: **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, <u>אך יש לנמק כל תשובה</u>.
- בדקו שבטופס שבידכם יש 14 עמודים כולל עמוד זה.
- המבחן כולל ארבע שאלות, יש לענות על כולן.

שאלה 1	/ 22
שאלה 2	/ 20
שאלה 3	/ 28
שאלה 4	/ 20
שאלה 5	/ 10
ציון סופי	/100

בהצלחה !

שאלה 1 – זיכרון וירטואלי (22 נק')

נתון מעבד דמוי x86 העובד במוד של 64 ביט ומבנה הכתובת הבא:

63 36	35 28	27 20	19 12	11 0
Sign Ext	PML3	PML2	PTE	offset

- במעבד קיים TLB בעל 4 כניסות, direct map.
  - במידה ויש TLB hit, התרגום מתקבל תוך מחזור שעון אחד.
  - TLB miss מתגלה תוך מחזור שעון אחד, ובמקרה זה פונים ל-PMH.
- ב-PMH של המעבד ישנם translation caches עבור כל אחת מרמות התרגום PML2 ו-PML3.
  - הגישה ל-PML2 ול-PML3 מתבצעת במקביל.
  - ב-PML2 יש 4 כניסות fully associative וב-PML3 יש כניסה אחת.
  - במידה ויש hit, הכניסה המתאימה (מ-PML2 או PML3) מתקבלת תוך 3 מחזורי שעון.
  - הזמן לקביעת miss גם הוא 3 מחזורי שעון.
  - במידה וה-PMH נאלץ להביא כניסה מהזיכרון, הוא מזריק load על-מנת להביא את הכניסה הנדרשת מה-data cache.
- ה-data cache הוא בגודל 32KB, 8 way set associative, בעל גודל שורה של 64 בתים.
  - במידה ויש cache hit, הנתון מתקבל תוך 6 מחזורי שעון.
  - הזמן לקביעת miss הוא 4 מחזורי שעון, ובמקרה זה יש לפנות לזיכרון .
  - הניחו שלאחר ששורה מובאת ל-cache, היא לא נזרקת ממנו במהלך סידרת הפניות.
- גישה לזיכרון אורכת 100 מחזורי שעון.

א. (2 נק') מהן הסיביות בכתובת באמצעותן יש לפנות אל כל אחד מה-translation caches ?

TLB_____

PML2 cache_____

PML3 cache_____

ב. (1 נק') בהנחה שגודל כל טבלה בעץ טבלאות הדפים שווה לגודל דף וירטואלי, מהו גודל כניסה בטבלת הדפים ? הסבירו.

_____

_____

_____

ג. (10 נק') נתונה סידרת פניות לכתובות וירטואליות (בבסיס 16). עבור כל אחת מהפניות יש לפרט:

- 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.
- זמן הגישה הכולל לקבלת התרגום.

הניחו כי בתחילת הסידרה כל ה-caches ריקים.

כתובת	TLB hit/ miss	PML2 hit/miss/ n.a.	PML3 hit/miss/ n.a.	D$ Num hits	D$ num misses	זמן גישה לקבלת התרגום
FFFF FFF9 8765 4321						
FFFF FFF9 0765 4321						
FFFF FFF9 8765 7321						
FFFF FFF9 8755 7321						
FFFF FFF9 0765 4321						

ד. (4 נק') עבור סידרת הפניות מהסעיף הקודם, מהו הגודל המינימלי (בבתים) שתופסות טבלאות הדפים הנדרשות למיפוי סידרת הפניות ? הסבירו

ה. (5 נק') יש לשנות את מבנה הכתובת הוירטואלית, כך שגודל דף יהיה 2^{13}, המרחב הפיזי הנתמך יהיה 2^{50} והמרחב הוירטאולי הנתמך יהיה לפחות 2^{60}. עוד נדרש, שגודל כל טבלה בכל הרמות יהיה בגודל דף, ושבכל כניסה בטבלה בכל הרמות 16 סיביות משמשות לניהול. על גודל הכתובת הוירטואלית להישאר 64 סיביות. ציירו את מבנה הכתובת הוירטואלית החדשה והסבירו.

שאלה 2 – זיכרון Cache (20 נק')

נתון מעבד עם היראכרית זיכרון בעלת שתי רמות מטמון Cache: L1 ו – L2.

L1: 2-Way set associative, 32 בתים בשורה, גודל 32KBytes, מדיניות החלפה LRU,
מדיניות כתיבה: Write Allocate, WB.

L2: 4-Way set associative, 64 בתים בשורה, גודל 256KBytes, מדיניות החלפה LRU,
מדיניות כתיבה: Write Allocate, WB.

נתונים זמני הגישה הבאים:

L1 lookup latency 1 clk cycle; L1 fill latency 1 clk cycle
L2 lookup latency 12 clk cycle; L2 fill latency 12 clk cycle
Main memory lookup latency 100 clk cycle

נתונים שלושה מערכים:

```
uint32 HW_grade [Student_num];      // HW_grade located at address 0x00000
uint32 Exam_grade [Student_num];    // Exam_grade located at address 0x48000
uint32 Final_grade [Student_num];   // Final_grade  located at address 0x96000
```

כל איבר בכל אחד מהמערכים הוא מסוג unit32 שגודלו 4 bytes.

נתונה התכנית הבאה, לצורך חישוב הציונים הסופיים בקורס מבנה מחשבים:

```
for (int i=0; i< Student_num; i++)
        Final_grade [i] = 0.2* HW_grade [i] + 0.8* Exam_grade [i];
```

נתון כי מספר הסטודנטים בקורס, Student_num, מוחזק ברגיסטר וערכו 4K=4096, ונתון כי המשתנה i גם הוא מוחזק ברגיסטר, וכי בתחילת הביצוע כל זיכרונות המטמון ריקים.

א. (5 נק') מהו ה-hit rate ב- L1, ומהו ה-hit rate ב- L2 ? יש להסביר

ה-hit rate ב- L1:_____

_____

_____

ה-hit rate ב- L2:_____

_____

_____

ב. (5 נק') מהו הזמן (clk cycles) הדרוש לביצוע התכנית הנתונה (במקרה של Miss זכרו להוסיף את הזמן הדרוש לביצוע Fill לפני העברת ה – Data מה – Cache).
לחישוב הזמן ב clk cycles התחשבו רק בזמני הגישות לזיכרון.

ג. (5 נק') פתרון בחומרה: נוסיף Victim Cache במקביל ל L1 בעל שורה אחת באורך 32 בתים כך שיתקבל Hit Rate טוב יותר מזה שהתקבל בסעיף א, מהו ה-Hit Rate המשותף שיתקבל בקומבינציה הזו של L1+Victem Cache ? יש להסביר

ד. (5 נק') פתרון בתוכנה (ללא שימוש ב – Victim Cache): יש לכתוב את התכנית מחדש תוך שימוש ב-4 רגיסטרים לשמירת תוצאות ביניים כך שיתקבל L1 Hit Rate טוב יותר מזה שהתקבל בסעיף א, מהו הזמן (clk cycles) הדרוש לביצוע התכנית החדשה? יש להסביר

שאלה 3 – Out-Of-Order Execution (28 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 לפקודה: **עד 3 פקודות בכל מחזור**, החל מ- t =1.
  - ב-ROB יש 8 כניסות, ב-Load Buffer יש 4 כניסות, ב-Store Buffer יש 2 כניסות, וב-RS יש 6 כניסות. **ניתן לבצע אלוקציה לפקודה רק כאשר יש מקום במבנים הנדרשים לה.**
- src1, src2: מספרי הרגיסטרים המשמשים כ-sources לפקודה:
  Pi עבור רגיסטר פיזי, ו-Ri במידה וקוראים ישירות את הרגיסטר הארכיטקטוני.
  עבור store: src1 – הרגיסטר המשמש לחישוב הכתובת. src2 – הרגיסטר המכיל את הנתון.
- T src1 ready , T src2 ready: הזמן בו מוכן כל אחד ערכי ה-sources לפקודה.
  אם ה-src 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. הניחו כי ישנן אינסוף יחידות ביצוע.
  - פקודה יכולה להיכנס לביצוע לכל המוקדם במחזור שלאחר האלוקציה.
  - פקודה נכנסת לביצוע במחזור השעון שלאחר המחזור בו כל ה-src-ים (עבור store ו-load: ה-src-ים לחישוב הכתובת) מוכנים: Texe = max(T alloc, T src1 rdy, T src2 rdy) + 1
- Load block code: עבור load שנשלח לביצוע בזמן t=Texe, או שהוסר עבור תנאי חסימה קודם בזמן t, תנאי החסימה נבדקים בזמן t+1 לפי הסדר:
  - 1 – חסימה כתוצאה מ- unresolved store address
  - 2 – חסימה כתוצאה מ- waiting for store data

  יש לרשום את כל קודי החסימה עליהם ה-load נחסם לפי הסדר.
- T data ready:
  - עבור load שנשלח לחישוב כתובת בזמן t = Texe ולא נחסם, או שהוסר עבורו תנאי חסימה בזמן t ולא נחסם פעם נוספת:
    - במידה וה-load פוגע ב-cache, או שיש store to load forwarding: בזמן **t+3**.
    - אחרת: במידה ובוצע load לאותה שורה בזמן t'<t (לאחר שכבר הוסרו כל החסימות שלו) בזמן **max(t+3, t'+12)**. אחרת, בזמן **t+12**.
  - עבור store: מחזור השעון בו הן ה-data לכתיבה לזיכרון והן הכתובת מוכנים.
    - כלומר T data ready = max(Texe, T src2 ready).
    - הכתובת של ה-store ידועה בזמן Texe. בפרט store המבוצע באותו זמן של Load, אינו גורם לחסימת ה-Load על unresolved store address.
  - עבור פקודות ALU: **Texe+1**.

- עבור פקודת Jump עם חיזוי שגוי, מבוצע flush בזמן Texe+1, והפקודות מהמסלול הנכון מבצעות אלוקציה בזמן **Texe+8**.

- T commit: הזמן בו הפקודה מבצעת commit
  - פקודה יכולה לבצע commit החל מזמן T data ready+1, ובתנאי שהפקודה שלפניה ביצעה/מבצעת commit.
  - אין מגבלה על כמות הפקודות שמבצעות commit בכל מחזור.
  - פקודת store מבצעת את הכתיבה אל ה-cache בזמן post-commit.

- הנחות:
  - הכתובות בתוכנית הן פיזיות (אין צורך בתרגום).
  - כל הערכים המספריים (כתובות, קבועים וכו) בשאלה הם בבסיס 16.
  - L1 data cache הוא ריק בתחילת הביצוע.
  - גודל שורה ב-L1 cache היא **32_{10}B** **(**$32_{10} = 20_{16}$**).**
  - ה-cache עובד במדיניות write no allocate.
  - 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
0	load R3 ← m[R1+10]	10	20	30	20	30									
1	store m[R3+20] ← R1	10	20	30	50	10									
2	load R2 ← m[R2+30]	10	10	30	50	10									
3	add R1 ← R1 + 10	20	10	30											
4	store m[R1+40] ← R2	20	10	30	60	10									
5	load R3 ← m[R1+40]	20	10	10	60	10									
6	add R1 ← R1 + 10	30	10	10											
7	load R3 ← m[R1]	30	10	20	30	20									
8	if (R1>10) jmp wrongly predicted	30	10	20											
9	add R1 ← R2 + R3	30	10	20											

שאלה 4 – Power/Performance & SMT impact (20 נק')

דרוש לתכנן מערכת Thin and light Notebook בעלת שני Core's במסגרת מעטפת הספק נתונה של 6Watt 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.

כל Core הוא מעבד 4wide. שטח כל Core הינו $5mm^2$.

ההספק הסטטי (Leakage Power) הוא 0.1Watt לכל מילימטר רבוע
(ההספק הסטטי קבוע ולא משתנה עם המתח).

הקיבול הדינאמי של כל Core נתון כפונקציה של ה – IPC של האפליקציה אותה הוא מריץ

וערכו הוא: Cdyn = IPC × 500pF

נתון ה – IPC של אפליקציות מסוגים שונים: Virus=4, TDP=3, Warm=2, Cold=1

להלן נתונה טבלה המראה את נקודות מתח ותדר אפשריות לעבודת המעבדים.

מתח ב Volt's	תדר ב Ghz
0.60	1
0.65	1.3
0.70	1.6
0.75	1.75
0.80	2.25
0.85	2.5
0.90	3
1	3.5
1.1	4

א. (5 נק') מיצאו את תדר העבודה בנקודת תכנון P1 (guaranteed frequency)? הסבירו.

ב. (5 נק') חשבו לאיזה תדר תוכל להגיע המערכת כאשר מורצת אפליקציה Warm אחת על Core אחד כאשר ה – Core השני הוא Power Gated ? הסבירו.

ג. (5 נק') חשבו את מספר הפקודות המבוצעות בשנייה בכל אחד משני הסעיפים לעיל.

ד. (5 נק') עתה מוסיפים תמיכה ב – Multi-Threading כך שכל אחד מה – Core's יכול להריץ 2 Threads בו זמנית 2 way SMT.
תוספות התמיכה ב – SMT מגדילות את הקיבול הסטאטי ב 10% והקיבול הדינאמי עבור אפליקציות TDP גדל ב – 20% בצורה קורלטיבית לגידול הכולל של ה- IPC של שני ה – Threads הרצים בו זמנית על כל Core.
חשבו את מספר הפקודות המבוצעות בשנייה כאשר המערכת מריצה 4 Threads TDP בו זמנית.

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

נתונה סידרת הפניות הבאה: … 011010 011010 011010

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: 0 XOR X = X , 1 XOR X = not X).

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

היסטוריה	011	110	101	010	100	001	011	110	101	010	100	001
סיביות כתובת	111	000	111	000	111	000	111	000	111	000	111	000
מצביע לחזאי מס'												
חיזוי												
קפיצה בפועל	0	1	0	0	1	1	0	1	0	0	1	1
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 011.

מהו אחוז החיזוי הנכון במצב היציב עבור סידרה זו ? יש להסביר

_____

_____

_____

_____

_____

_____

_____