הטכניון - מכון טכנולוגי לישראל

מבנה מחשבים (236267)

מבחן מסכם מועד ב'

12 מרץ 2021

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

<u>מתרגלים</u>: **פרנק סלה, איתי רביד, אלעד שטיגמן.**

מס. ת.ז. : _____

- משך הבחינה: שלוש שעות.
- מותר חומר עזר מודפס ומחשבון בלבד,
  אסור חומר עזר במחשב ולא בכל אמצעי מקוון אחר.
- מומלץ להדפיס את הבחינה ולכתוב את התשובות בטופס הבחינה.
- יש לכתוב בקיצור ככל האפשר, <u>אך יש לנמק כל תשובה</u>.
- יש לכתוב בכתב ברור וקריא.
- המבחן כולל 4 שאלות, יש לענות על כולן. במבחן 13 עמודים.
- יש לסרוק את הבחינה בפורמט PDF בלבד, ולוודא שהבחינה הסרוקה ניתנת לקריאה בבירור.

שאלה 1	OOO	/ 32
שאלה 2	BP	/ 12
שאלה 3	power	/ 12
שאלה 4	VM/cache	/ 44
ציון סופי		/100

בהצלחה !

שאלה 1 – Out-Of-Order Execution (32 נק')

א. (26 נק') יש למלא את הטבלה שבהמשך. לכל פקודה יש לרשום:

- R3, R2, R1 – ערכי הרגיסטרים הארכיטקטוניים לאחר retire של הפקודה.
  עבור פקודה ממסלול שגוי: הערך שחושב עבור הרגיסטר הפיזי בזמן execute.
- addr – כתובת הגישה לזיכרון – עבור פקודות load ו-store בלבד.
- data – ערך זיכרון שנקרא או נכתב – עבור פקודות load ו-store בלבד.

מולאו בחלקם, יש להשלים את החסר

- T alloc: הזמן בו מבוצעת אלוקציה: **עד 3 פקודות בכל מחזור**, החל מ- t =1.
  ניתן לבצע אלוקציה רק כאשר **לכל הפקודות** שמספק ה-frontend במחזור יש מקום ב-ROB.
  ב-ROB יש **9 כניסות**. כל פקודה (כולל פקודות Store) תופסת מקום אחד ב-ROB.
- 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 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. 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 מוכנים. פקודת store נכנסת לביצוע במחזור השעון שלאחר המחזור בו src1 (המשמש לחישוב הכתובת) מוכן.
- Load block code: עבור load שנשלח לביצוע בזמן t=Texe, או שהוסר עבורו תנאי חסימה קודם בזמן t, תנאי החסימה נבדקים בזמן t+1 לפי הסדר (רישמו את כל תנאי החסימה לפי הסדר):
  1. unknown store address (התנאי מוסר בזמן T addr ready של ה-store החוסם)
  2. waiting for store data (התנאי מוסר בזמן T data ready של ה-store החוסם)
- T addr ready: ממולא עבור פקודות load ו-store בלבד: **Texe+1**

  עבור load ו-store המבוצעים באותו זמן t: תנאי החסימה של ה-load נבדקים בזמן t+1, בזמן זה הכתובת של ה-store ידועה, ולכן ה-load לא נחסם על unknown store address ע"י ה-store.
- T data ready:
  - עבור פקודות add, sub, conditional jump: **Texe+1**. עבור פקודת mul: **Texe+3**.
  - 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/store לאותה שורה ב-cache בזמן Texe >Texe':
      בזמן **max(Texe'+8, t+4)**.
    - אחרת, בזמן **max(Texe+8, t+4)**
  - עבור store: **T src2 ready + 1**

- עבור פקודת Jump עם חיזוי שגוי מבוצע flush בזמן Texe+1, וכבר במחזור זה אין אלוקציה של פקודות. הפקודות מהמסלול הנכון מבצעות אלוקציה החל מזמן **Texe+6** (במידה ואין סיבה אחרת שמעכבת את האלוקציה).

- T retire: הזמן בו הפקודה מבצעת retire. ניתן לבצע retire לעד **3 פקודות** בכל מחזור. פקודה (למעט store) יכולה לבצע retire החל מזמן T data ready+1.
  פקודת store יכולה לבצע retire החל מזמן max(T addr ready, T data ready) +1.

- #ROB entries: מספר הכניסות התפוסות ב-ROB מיד לאחר שהפקודה ביצעה אלוקציה.
  יש להתחשב גם בפקודות שהוצאו מה-ROB. פקודה שמבצעת retire בזמן t מוצאת מה-ROB במחזור t+1 וכבר במחזור זה פקודה חדשה יכולה לבצע alloc במקום שהתפנה.

- הנחות:
  - בטבלה רשומות הפקודות שמבצעות אלוקציה, כולל פקודות מהמסלול השגוי.
  - הכתובות הן פיזיות (אין צורך בתרגום). כל הערכים המספריים הם בבסיס 16.
  - L1 data cache: גודל שורה $32_{10}B = 20_{16}B$. write allocate. ריק בתחילת הביצוע.
  - בתחילת הביצוע כתובת N בזיכרון מכילה את הערך N.
  - פקודה המסומנת ב-* בטבלה אינה מסופקת ע"י ה-frontend לאלוקציה באותו מחזור של הפקודה הקודמת לה, אלא במחזור הבא. בכל מקרה אחר, ה-frontend מספק 3 פקודות למחזור.

ב. (6 נק') עבור ריצת התוכנית שמצאתם בסעיף א', מלאו לכל מחזור שעון שבטבלה שבסעיף זה את נתוני ה-Top level breakdown. (מחזורי השעון והנתונים הממולאים מתייחסים לזמן האלוקציה).

- במידה ובמחזור האלוקציה האחרון נותרו בתוכנית פחות מ-3 פקודות, יש להחשיבו כ-frontend bound.
- ייתכן שיש בטבלה עמודות מיותרות, אם כן יש להשאירן ריקות ולא להתייחס אליהם.

Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13
Backend Stall													
Alloc Slot 0													
Alloc Slot 1													
Alloc Slot 2													
Frontend Bound													
Backend Bound													
Retiring													
Bad Speculation													

אחוז Frontend Bound _____ אחוז Bad Speculation _____

אחוז Backend Bound _____ אחוז Retiring _____

Pdst	instruction	R1	R2	R3	addr	data	src1	src2	T alloc	#ROB entries	T src1 ready	T src2 ready	T exe	T addr ready	T data ready	block code	T retire
1	load R1←m[R2+30]	40	10	50	40	40											
2	Store m[R3+10] ←R1	40	10	50	60	40											
3	mul R3 ← R2 × 2 *	40	10	20													
4	Store m[R3+50] ←R2	40	10	20	80	10											
5	if (R1>0) PC←1000 **חיזוי נכון**	40	10	20													
6	load R3←m[R2+50]	40	10	40	60	40											
7	load R1←m[R2+80] *	50	10	40	50	50											
8	if (R2>0) PC←3000 **חיזוי שגוי**	50	10	40													
9	sub R3 ← R2 - 10																
10	add R2 ← R2 + 10																
11	load R1←m[R2+R3]																

שאלה 2 – חיזוי קפיצות (12 נק')

נתונה סידרת הקפיצות המחזורית הבאה: … 100110 100110

נניח חזאי מסוג local predictor, שבו ההיסטוריה מצביעה אל מערך שבו בכל כניסה מונה 2 סיביות עם רוויה. המונים מאותחלים למצב weakly taken.

א. (5 נק') מהו אורך ההיסטוריה המינימלי N המאפשר חיזוי של הסידרה הנתונה במצב היציב ללא שגיאות ? יש להסביר

ב. (7 נק') עבור החזאי שנמצא בסעיף א' (בעל היסטוריה באורך N), לאחר מחזורים רבים של הסידרה, קפיצה בודדת בדפוס משתנה, כך שהסדרה מתנהגת באופן הבא: … 110110 110110
כמה שגיאות ייווצרו עד שהחזאי ילמד את הדפוס החדש ויגיע שוב לחיזוי מושלם ? יש להסביר

שאלה 3 – Power/Performance (12 נק')

נתונה מערכת שבה שני מעבדים: מעבד אחד מסוג Big ומעבד אחד מסוג Small. נקודות מתח ותדר אפשריות נתונות בטבלה משמאל. ההספק הכולל המוקצה לשני המעבדים יחד הוא 5W.

מתח Volt	תדר Small GHz	תדר Big GHz
0.8	1.7	2.2
0.9	2.2	2.8
1.0		3.5

	C1 – Big	C2 – Small
Leakage Power	0.3W	0.1W
Cdyn	400pF×IPC	300pF×IPC
IPC עבור אפליקציה A	4	3
IPC עבור אפליקציה B	4	2

המערכת מריצה שתי אפליקציות בו-זמנית. יש לבחור על איזה מעבד להריץ את אפליקציה A ועל איזה מעבד להריץ את אפליקציה B כך שסך ביצועי המערכת יהיו מקסימליים במונחי IPS (פקודות לשניה) ? מהו ה-IPS המירבי המתקבל ? יש להסביר.

שאלה 4 – זיכרון וירטואלי ו-cache (44 נק')

נתון מעבד דמוי x86 העובד במבנה הכתובת הבא:

63 – 43	42 – 33	32 – 23	22 – 13	12 – 0
Sign Ext	PML3	PML2	PTE	offset

- כל טבלת דפים (בכל הרמות) היא בגודל דף.

המעבד כולל:

- TLB בגודל של 32 כניסות, 4 ways, מדיניות LRU.
  זמן גישה (ל-hit או miss) של מחזור אחד. במקרה של TLB miss פונים ל-PMH.
- PMH הכולל translation caches, PML2$ ו-PML3$, שהגישה אליהם מתבצעת במקביל. זמן גישה (ל-hit או miss) 6 מחזורים. בכל אחד מהם 2 כניסות Fully Associative במדיניות LRU.
- L1 cache: 64KB, 8 ways, 64B בשורה, מדיניות LRU. זמן גישה (ל-hit או miss) 4 מחזורים.
- L2 cache: 256KB, 8 ways, 64B בשורה, מדיניות LRU.
  זמן גישה (ל-hit או miss) 10 מחזורים.
- זמן גישה לזיכרון הוא 100 מחזורים.
- לאחר ביצוע תרגום, מידע התרגום המעודכן מוכנס לכל המבנים שאליהם ניגשו במהלך התרגום.

שימו לב ש-L1 ו-L2 משמשים הן עבור נתונים והן עבור כניסות בטבלאות הדפים שה-PMH מביא מהזיכרון. יש להתחשב בכך לכל אורך הפתרון ולהראות האם נוצרות התנגשויות ב-cache-ים בין נתונים לבין כניסות תרגום.

נתון מערך A[0x100,000], שבו כל איבר הוא בגודל 8 בתים.
(הקידומת 0x מציינת מספר בבסיס 16).

- המערך פרוש בקטע רציף במרחב הווירטואלי של התהליך החל מכתובת 0x10,000,000
- בזמן ההכרזה על המערך, מערכת ההפעלה מקצה עבורו דפים באזור רציף בזיכרון הפיזי, החל מכתובת 0x2,000,000.
- מערכת ההפעלה מקצה את טבלאות התרגום הנדרשות למיפוי המערך A כולו באזור רציף בזיכרון הפיזי, החל מכתובת 0x1,000,000, שבה מתחילה טבלת PML3. לאחר מכן טבלאות PML2 הדרושות למיפוי A, ולבסוף טבלאות התרגום הדרושות למיפוי A לפי הסדר.

נתונה התוכנית הבאה:

```
for (int i=0; i< 0x4,000; i++)  { T+= A[i×4] ^ 2; }
T = T / 0x4,000;
for (int i=0; i< 0x4,000; i++)  { S+= A[i×4+1] ^ T; }
```

- T, S, i מאוחסנים כל אחד ברגיסטר.
- בשאלה זו נתעלם מגישות לצורך הבאת הקוד. בתחילת הביצוע כל זיכרונות המטמון ריקים.

א. (5 נק') מהו מספר הדפים הנדרש עבור טבלאות הדפים (בכל הרמות) הנדרשות למיפוי המערך A כולו ? הסבירו.

ב. (4 נק') מהי הכתובת של ה-PTE שמשמש לתרגום הכתובת של איבר A[0x1000] ? הסבירו

ג. (3 נק') כמה set-ים יש ב-L1 cache ? הסבירו

ד. (15 נק') מהו הסכום המדויק של הזמנים הנדרשים לביצוע תרגומי כל הכתובות בלולאה הראשונה ? האם יש התנגשויות בין נתונים לתרגומים ב-cache ? הסבירו.

ה. (5 נק') מהו ה-L1 hit rate במהלך התוכנית כולה עבור פניות לנתונים ? הסבירו

ו. (5 נק') מהו ה-L2 hit rate במהלך התוכנית כולה עבור פניות לנתונים ? הסבירו

ז. (7 נק') כיצד ניתן לשנות את התוכנית, מבלי לשנות את תוצאת החישוב, על-מנת לקבל L2 hit rate טוב יותר. מהו ה-L2 hit rate שיתקבל ? הסבירו