הטכניון - מכון טכנולוגי לישראל

מבנה מחשבים ספרתיים (234267)

מבחן מסכם מועד ב'

28 פברואר 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	/ 21
שאלה 2	/ 24
שאלה 3	/ 29
שאלה 4	/ 20
שאלה 5	/ 6
ציון סופי	/100

בהצלחה !

שאלה 1 – זיכרון וירטואלי (21 נק')

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

63 48	47 36	35 24	23 12	11 0
Sign Ext	PML3	PML2	PTE	offset

- במעבד קיים TLB גדול. במידה ויש TLB hit, התרגום מתקבל תוך מחזור שעון אחד.
  - TLB miss מתגלה תוך מחזור שעון אחד, ובמקרה זה פונים ל-PMH.
- ב-PMH ישנם translation caches גדולים עבור כל אחת מרמות התרגום PML2-3.
  - הגישה אליהם מתבצעת במקביל.
  - במידה ויש hit, הכניסה המתאימה מתקבלת תוך 6 מחזורי שעון.
  - הזמן לקביעת miss גם הוא 6 מחזורי שעון.
- במעבד קיים data cache בגודל 64KB, 2 way set associative, וגודל שורה של 64 בתים.
  - עבור hit הנתון מתקבל תוך 4 מחזורי שעון. הזמן לקביעת miss הוא 3 מחזורי שעון.
  - הניחו שלאחר ששורה מובאת ל-cache, היא לא נזרקת ממנו במהלך סידרת הפניות.
  - גישה לזיכרון אורכת 200 מחזורי שעון.

א. (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 גודל כניסה בטבלאות הדפים היא 8 בתים וכן 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 CBA9 8765 4321						
FFFF CBA9 8765 0321						
FFFF CBA0 8765 4321						
FFFF CBA9 8765 4021						
FFFF CBB9 8765 4321						

ב. (6 נק'). עבור המערכת המתוארת בתחילת התרגיל, הציעו שתי דרכים שונות על מנת לאפשר פנייה ל-L1 set במקביל לתרגום הכתובות מווירטואליות לפיזיות.
ניתן לשנות את מבנה ה-cache (אך יש להשאירו בגודל 64KB). אין לשנות את מבנה הכתובת הווירטואלית. יש להסביר.

אפשרות א': _____

_____

_____

אפשרות ב': _____

_____

_____

ג. (5 נק') סעיף זה אינו מתבסס על הנתונים שבתחילת השאלה ולא על הסעיפים הקודמים.
יש לתכנן מבנה לכתובת וירטואלית, כך שמתקיימים התנאים הבאים:

- גודל דף 2^{15}B, גודל כל טבלה בכל אחת מהרמות הוא בגודל דף
- המרחב הפיזי הנתמך 2^{56} Byte, והמרחב הוירטאולי הנתמך 2^{63} Byte.
- בכל כניסה בטבלאות בכל הרמות 20 סיביות משמשות לניהול (בנוסף למיפוי לכתובת פיזית).
- גודל כתובת וירטואלית: 64 סיביות.

ציירו את מבנה הכתובת הוירטואלית והסבירו.

_____

_____

_____

_____

שאלה 2 – זיכרון Cache (24 נק')

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

ונתון כי הרמה העליונה L1 Cache יכולה להיות קונפיגורבילית פעם אחת המבנה של Unified Cache בין Instructions ו – Data ופעם שניה במבנה של Split Cache המורכב מ: I-Cache ו – D-Cache נפרדים.

כאשר ה – L1 Cache הוא במבנה של Split Cache אזי:

I-Cache: Direct Mapped, 8 בתים בשורה, גודל 1KBytes, מדיניות כתיבה: Write Allocate, WB.

D-Cache: Direct Mapped, 64 בתים בשורה, גודל 16KBytes, מדיניות כתיבה: Write Allocate, WB.

כאשר ה – L1 Cache הוא במבנה של Unified Cache אזי:

Unified L1 Cache: 2-Way set associative, 8 בתים בשורה, גודל 16KBytes, מדיניות החלפה LRU, מדיניות כתיבה: Write Allocate, WB.

L2: 4-Way set associative, 16 בתים בשורה, גודל 128KBytes, 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 80 clk cycle

נתונה הלולאה המקוננת הבאה:

```
for (int i=0; i< X; i++)
        for (int j=0; j< Y; j++)
                Z=A[ j ]+W;
```

נתונן כי המערך A[ ] הוא מערך של בתים היושב בכתובת בסיס 0x40000000 (כך שלדוגמה הכתובת בזhכרון של איבר A[j] היא : 0x40000000 + j.

W הוא קבוע השמור ברגיסטר, וכן – X,Y,Z הם משתנים המאוחסנים ברגיסטרים. כמו כן גם אנדקסי הלולאות i ו - j מאוחסנים כל אחד ברגיסטר.

הקומפיילר מייצר 16 פקודות מכונה למימוש הלולאה המקוננת שלעיל, אחת מהפקודות היא פקודת Load Byte (המשמשת לקריאת הנתון היושב בכתובת A[j]) כל פקודה היא באורך קבוע של 4 בתים והקוד יושב בטווח הזיכרון המצוי בכתובות 0x80000000 עד 0x8000003F.

נתון כי בתחילת ביצוע התכנית כל זיכרונות המטמון ריקים.

נתון כי כל זיכרונות המטמון הינם "Blocking/non pipelined" היינו לא ניתן לבצע גישה חדשה לפני שהגישה הקודמת סופקה.

א. (6 נק') בקונפיגורציה בה ה - L1 Cache הוא Split Cache ובהנחה ש Y=1 מהו ה-hit rate ב- I-Cache, וב- D-Cache כפונקציה של X ? יש להסביר

ה-hit rate ב- L1 I-Cache :_____

_____

_____

_____

ה-hit rate ב-L1 D-Cache:_____

_____

_____

_____

ב. (6 נק') באותה הקונפיגורציה בה ה-L1 Cache הוא Split Cache, מהו הזמן (clk cycles) הדרוש לביצוע התכנית הנתונה כאשר X=1 ו–Y=1 ? (במקרה של Miss זכרו להוסיף את הזמן הדרוש לביצוע Fill). לחישוב הזמן ב clk cycles התחשבו רק בזמני הגישות לזיכרון.

_____

_____

_____

_____

_____

_____

_____

_____

ג. (6 נק') נמשיך באותה בקונפיגורציה בה ה- L1 Cache הוא Split Cache ובתנאי התחלה בהם כל זיכרונות המטמון ריקים. בהנחה ש–X הוא מספר גדול מאוד מה יהיה ה-hit rate ב-D-cache אם:

i) Y=8192 כאשר ה – D-Cache הוא Direct Mapped? יש להסביר

ii) Y=32768 כאשר ה – D-Cache הוא Direct Mapped? יש להסביר

iii) Y=32768 כאשר ה –D-Cache הוא Fully Associative עם מדיניות החלפה LRU ? יש להסביר

ד. (6 נק') עתה הקונפיגורציה של ה-L1 Cache היא Unified Cache. האם יהיו קונפליקטים בין Code ל-Data כך שאחד יפנה את השני מה- Unified L1 Cache?
אם התשובה היא "כן יהיו קונפליקטים" אזי יש להסביר מי יפנה את מי: ה-Code את ה-Data או ה-Data את ה-Code ? יש להסביר.

i) כאשר Y=8192

ii) כאשר Y=16384

שאלה 3 – Out-Of-Order Execution (29 נק')

א. 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, וב-RS יש **5** כניסות.
    ניתן לבצע אלוקציה לפקודה רק כאשר יש עבורה מקום ב-ROB וב-RS.
  - כל פקודה (כולל פקודות store) תופסת מקום אחד ב-ROB ומקום אחד ב-RS.

- 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.
  - פקודת מוצאת מה-RS במחזור שלאחר הביצוע שלה (Texe+1), וכבר במחזור זה פקודה חדשה יכולה לבצע alloc ולהשתמש במקום שהתפנה.

- 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 – חסימה כתוצאה מ- unknown store address
  - 2 – חסימה כתוצאה מ- waiting for store data

  Load פונה ל-L1 cache בזמן Texe+1 (גם אם נחסם), כך שמידה ויש L1 cache miss, נשלחת כבר פניה ל-L2 cache, ובזמן Texe+10 הנתונים כבר זמינים עבור ה-load.

- T data ready:
  - עבור load שהוסרו עבורו תנאי החסימה בזמן t ולא נחסם פעם נוספת:
    - 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'exe <Texe: בזמן **max(T'exe+10, t+4)**.
    - אחרת, בזמן **max(Texe+10, t+4)**.

- עבור store: מחזור השעון בו הן ה-data לכתיבה לזיכרון והן הכתובת מוכנים.
  - כלומר T data ready = max(**Texe+1**, T src2 ready).
  - הכתובת של ה-store ידועה בזמן **Texe+1** (ובזמן זה מוסר תנאי חסימה של load שנחסם ע"י ה-store על unknown store address).
  - בפרט, load שמבוצע בזמן t (תנאי החסימה שלו משוערכים בזמן t+1), לא נחסם על unknown store address ע"י store קודם לו, שמבוצע גם הוא בזמן t (כתובת ה-store ידועה בזמן t+1).
- עבור פקודות ALU: **Texe+1**.

- עבור פקודת Jump עם חיזוי שגוי, מבוצע flush בזמן Texe+1, והפקודות מהמסלול הנכון מבצעות אלוקציה בזמן **Texe+6**.

- T commit: הזמן בו הפקודה מבצעת commit
  - פקודה יכולה לבצע commit החל מזמן T data ready+1, ובתנאי שהפקודה שלפניה ביצעה/מבצעת commit.
  - ניתן לבצע commit לעד **4 פקודות** בכל מחזור.
  - פקודת store מבצעת את הכתיבה אל ה-cache בזמן post-commit.
  - פקודת מוצאת מה-ROB במחזור שלאחר commit (Tcommit+1), וכבר במחזור זה פקודה חדשה יכולה לבצע alloc ולהשתמש במקום שהתפנה.
- 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. ה-cache עובד במדיניות write no allocate.
  - גודל שורה ב-L1 cache היא **32_{10}B** **($32_{10} = 20_{16}$).**
  - בטבלה רשומות אך ורק הפקודות מהמסלול הנכון.

ב. (4 נק') מוסיפים למעבד חזאי Memory Disambiguation. בהנחה שהחזאי חוזה נכון תמיד, אילו מפקודות ה-load שבתוכנית (מתוך הפקודות 0, 3, 5, 7) יחסמו על unknown store address ? הסבירו.

_____

_____

_____

_____

_____

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 R1 ← m[R2+50]	10	20	30	70	10									
1	store m[R1+60] ← R3	10	20	30	70	30									
2	add R3 ← R3 + R2	10	20	50											
3	load R1 ← m[R2+R3]	30	20	50	70	30									
4	store m[R2+20] ← R1	30	20	50	40	30									
5	load R2 ← m[R3-10]	30	30	50	40	30									
6	if (R3>20) jmp wrongly predicted	30	30	50											
7	load R2 ← m[R3]	30	10	50	50	10									
8	add R1 ← R1 + R2	40	10	50											

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

נתון Core שתוכנן בטכנולוגית ייצור 14nm (Process Technology) עם המאפיינים הבאים:

ה- Core הוא מעבד 4wide. שטח ה- Core הינו $4mm^2$.

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

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

וערכו הוא: Cdyn = IPC × 525pF.

נתון כי ה- Core אינו תומך ב -Multi-Threading היינו מסוגל להריץ רק Thread אחד.

נתון ה – IPC של אפליקציות מסוגים שונים: Virus=3.5, TDP=3, Warm=2, Cold=1.5.

להלן נתונה טבלה המראה את נקודות מתח ותדר אפשריות לעבודת ה- Core:

מתח ב Volt's	תדר ב Ghz
0.60	1
0.65	1.33
0.70	1.66
0.75	1.75
0.80	2.25
0.85	2.5
0.90	3.33
1	3.5
1.1	4

בשנה הבאה ניתן יהיה לייצר את ה-Core המדובר בטכנולוגית ייצור חדשה של 10nm בגרסת "speed" או בגרסת "power" עם המאפיינים הבאים:

גרסת ה- "speed" תאפשר הרצה בתדר הגבוה ב- 20% יחסית לטכנולוגית 14nm (תדר הגבוה ב – 20% לעומת התדר המצוין בטבלה שלעיל עבור כל נקודת מתח אפשרית).

גרסת ה- "power" לעומת זאת תספק הפחתת הקיבול הדינמי Cdyn ב- 30% יחסית לטכנולוגית 14nm.

דרוש לתכנן מערכות שונות העושות שימוש ב – Core בטכנולוגית ייצור 10nm במסגרת מעטפות הספק נתונות ועבור כל מערכת להחליט אם לבחור בגרסת ה - "speed" או בגרסת ה - "power" של טכנולוגיית הייצור החדשה על מנת למקסם את סך כל הביצועים של המערכת.

א. (5 נק') באיזו גרסה של טכנולוגית ייצור 10nm תבחרו להשתמש כדי לקבל ביצועים מקסימליים ממערכת Tablet המריצה בו זמנית שני Threads (שני Core's) בתנאי TDP במעטפת הספק של 4Watt המוקצים עבור ה – Core's? הסבירו.

ב. (5 נק') באיזו גרסה של טכנולוגית ייצור 10nm תבחרו להשתמש כדי לקבל ביצועים מקסימליים ממערכת Tablet כאשר מורצת עליה אפליקציית Warm אחת על Core אחד כאשר ה – Core השני הוא Power Gated במעטפת הספק של 4Watts המוקצים עבור ה – Core? הסבירו.

ג. (5 נק') השוו את הביצועים של גרסת טכנולוגית הייצור 10nm שבחרתם בסעיף א לביצועים של מערכת דומה המיוצרת בטכנולוגיה 14nm. לצורך ההשוואה חשבו את מספר הפקודות המבוצעות ביחידת זמן, 1 sec בכל אחת מהמערכות (ערכו את ההשוואה באותם תנאים כמו בסעיף א: מערכת המריצה בו זמנית שני Threads (שני Core's) בתנאי TDP במעטפת הספק של 4Watt המוקצים עבור שני ה – Core's).
כמו כן מצאו את השיפור באחוזים "Throughput Speedup" בין שני דורות הטכנולוגיה שהשוויתם.

ד. (5 נק') עתה נדרש לתכנן מערכת Desk-Top המורכבת מ – 6 מעבדים (6 Core's) באיזו גרסה של טכנולוגית ייצור 10nm תבחרו להשתמש כדי לקבל ביצועים מקסימליים ממערכת זו המריצה בו זמנית 6 Threads בתנאי TDP במעטפת הספק של 60Watts המוקצים עבור ה – Core's? הסבירו.
כמו כן סכמו את המסקנות העולות מהחישובים שערכתם בסעיפים א ו - ד של שאלה זו ותארו את השיקולים בבחירת מאפייני טכנולוגית הייצור Process Technology עבור מעטפות הספק שונות.

שאלה 5 – חיזוי קפיצות (6 נק')

במעבד קיים branch predictor מסוג lshare, עם 4 כניסות היסטוריה באורך 5 סיביות כל אחת.
בנוסף ישנו מערך של 32 מונים (bimodal counters), האינדקס למערך זה מורכב מהיסטורית ה branch כפי ששמורה במערך ההיסטוריות XORed עם 5 הסיביות התחתונות של כתובת ה branch.
מערך ההיסטוריה מאותחל ל-0 והמונים מותחלים ל-2 (weakly taken) .

המעבד מריץ את התוכנית הבאה:

```
100  mov  R1,0x0B      ; R1=0x0B (binary 01011)
104  mov  R2,5         ; R2=5
108  and  R9,R1,0x10   ; R9=R1&0x10 (0x10 is binary 10000)
10c  beq  R9,0x10,114  ; if (R9==0x10) PC=114
110  nop               ; no operation
114  sll  R1,R1,1      ; R1=R1<<1  (Shift Left Logical)
118  sub  R2,R2,1      ; R2=R2-1
11c  bne  R2,0,108     ; if (R2!=0) PC=108
```

א. (6 נק') השלימו את הטבלה הבאה עבור סידרת הקפיצות בסדר אותו המעבד מריץ את התוכנית לעיל.
שתי השורות הראשונות בטבלה כבר מולאו לצורך הדגמה.
שימו-לב כי כתובת הקפיצה נתונה בבסיס 16, ואילו ההיסטוריה בבסיס 2.

כתובת ה-branch	Taken/ not-taken	ערך ההיסטוריה לפני הקפיצה	מספר המונה	ערך המונה לפני הקפיצה	החיזוי (0/1)	נכונות החיזוי
10c	0	00000	01100	2	1	×
11c	1	00000	11100	2	1	✓
10c						
11c						
10c						
11c						
10c						
11c						
10c						
11c						