אוניברסיטת בר-אילן

הפקולטה להנדסה

BAR-ILAN UNIVERSITY

Engineering Faculty

מבנה מחשבים ספרתיים 83-301

תשע"ו סמסטר ב' מועד ב'

שם הקורס	מבנה מחשבים ספרתיים		
מספר הקורס	83-301		
שם המרצה	פרופ' שמואל וימר + מר עמית קזימירסקי		
	תשע"ו	סמסטר ב'	מועד ב'
משך הבחינה	שעתיים וחצי		
חומר עזר	כל חומר עזר אסור בשימוש. שאלון סגור! **יש לענות על טופס הבחינה מחברות הבחינה הן טיוטה בלבד**.		
	יש לענות על כל השאלות. כל תשובה יש לנמק ולהסביר היטב. כל תשובה מספרית מחייבת את הצגת דרך החישוב. סה"כ הנקודות האפשריות 110. ציון הבחינה לא יעלה על 100. **יש לכתוב בעט בלבד. כתיבה בעפרון לא תיבדק**.		

בהצלחה!

<u>שאלה א'</u> (60 נקודות)

נתונה מערכת מחשב בעלת זיכרון פיזי של 4GB, המערכת תומכת במרחב כתובות וירטואלי של 36 סיביות. גודל דף הינו 4KB. המערכת מאפשרת גישה לזיכרון על פי כתובות של בתים. גודל כניסה בטבלת הדפים הינו 4 בתים המשמשים ככתובת פיזית. ענו על השאלות הבאות באופן מפורט תוך הסבר מלא של כל חישוב. **<u>תשובות מספריות ללא הסבר לא תתקבלנה</u>**.

1. (3 נק') כמה דפים ישנם במרחב הכתובות הווירטואלי?

<u>תשובה</u>: 36 סיביות מאפשרות למען 2^{36} בתים. מאחר וגדלו של דף הינו $4K = 2^{12}$ בתים, מספר הדפים במערכת הינו $2^{36}/2^{12} = 2^{24}$.

2. (5 נק') בהנחה שתהליך יכול לגשת לכל דף בזיכרון, כמה מקום בזיכרון הראשי תופסת טבלת הדפים (page table) לכל אחד מהתהליכים במערכת?

<u>תשובה</u>: מאחר ובמרחב הכתובות הווירטואלי שלנו ישנם 2^{24} דפים, וכל כניסה בטבלת הדפים הינה בת 4 בתים, גודל הטבלה בזיכרון הינו $2^{24} \times 4 = 2^{26} = 64MB$.

3. (6 נק') נתון שגדלו הממוצע של תהליך הינו 0.5GB. כמה דפי זכרון בממוצע נדרשים לתהליך? מה המספר הממוצע של תהליכים שהמערכת תומכת בו זמנית? ניתן להזניח את גודל טבלת המיפוי

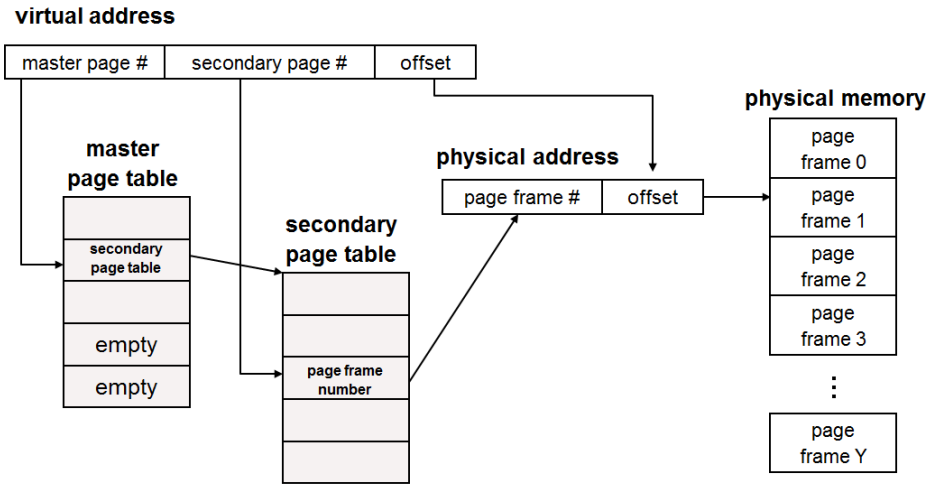
<u>תשובה</u>: מספר הדפים הממוצע בתהליך הינו $0.5GB/4KB = 2^{29}/2^{12} = 2^{17}$. מספר הדפים במערכת הינו 2^{24}. על כן המספר הממוצע של תהליכים שהמערכת תומכת בו זמנית הינו $2^{24}/2^{17} = 2^7 = 128$.

אפשר גם להקיש זאת מכך שגודל הזיכרון הווירטואלי הינו $2^{36} = 64GB$ ולכן מספר התהליכים הממוצע $64GB/0.5GB = 128$.

4. (6 נק') בכתה נלמד שטבלת הדפים מצויה בזיכרון הפיזי. מה ההשלכות של תמיכה בריבוי תהליכים על ביצועי תכניות משתמש? האם מסקנתכם תקפה גם לביצועי מערכת ההפעלה? הסבירו מדוע.

<u>תשובה</u>: על פי סעיף 2 גודל טבלת הדפים 64MB, שזהו $64MB/4GB = 2^{26}/2^{32} = 1/64$ כפול 128 תהליכים בממוצע שזהו כרבע מגודל הזיכרון. תמיכה בריבוי תהליכים יצמצם באופן משמעותי את מרחב הזכרון הפיזי שנותר לתכניות, הן של המשתמש והן של מערכת ההפעלה. כתוצאה מכך מערכת ההפעלה תבצע כל הזמן swapping, דבר שיאט מאד את זמן הריצה. המסקנה נכונה כמובן גם למערכת ההפעלה שבעצם מריצה את עצמה ואיננה שונה במובן זה משום תכנית משתמש.

5. מהנדסת צעירה בוגרת בר-אילן החליטה להקטין את גודל הזכרון הפיזי הנדרש לטבלאות דפים. הצעתה הינה לבנות את טבלת הדפים של תהליך בשתי רמות כמתואר בשרטוט: טבלה א' master page table הממוענת ע"י מחצית ה MSBs של הכתובת הווירטואלית. כל כניסה בטבלה זו מצביעה על כתובת שבה מתחילה טבלה ב' secondary page table הממוענת ע"י מחצית ה LSBs של הכתובת הווירטואלית.



(12 נק') באיזה זכרון מצויה טבלה א' ובאיזה טבלה ב'? מהו מספרן וגדלן של הטבלאות עבור תהליך בודד?

תשובה: טבלה א' חייבת להיות בזיכרון הפיזי וטבלה ב' הן בזיכרון הפנימי והן בדיסק. לו ב' הייתה גם בזיכרון הפיזי הרי שלא עשינו דבר לחסכון בזיכרון פיזי. גודל טבלה א' $2^{12} \times 36 = 144\text{Kbit}$ וישנה כמובן אחת כזאת. גדלה של טבלה ב' $2^{12} \times 32 = 128\text{Kbit}$. מספר טבלאות ב' עשוי להגיע עד 2^{12}. התקבלה גם התשובה שגודל כל שורה בטבלה הוא 4 בתים.

(3 נק') האם האמרה "אין ארוחות חינם" תופסת לגבי הצעתה של המהנדסת? במדה וישנו מחיר נוסף בעקבות ההצעה, הסבר מהו.

תשובה: גישה לכתובת פיזית נעשית כעת באופן דו שלבי, כאשר הטבלה המשנית מצויה בזיכרון הווירטואלי, כלומר יתכן ועצם המיפוי של כתובת וירטואלית לפיזית תגרום ל page fault, דבר שלא קרה המצב של טבלת דפים אחת בזיכרון הפיזי.

6. (18 נק') המכונה מריצה את הקוד שלהלן המשמש להכפלת מטריצות:

```
int a[1024][1024], b[1024][1024], c[1024][1024];
multiply()
{
   unsigned i, j, k;
   for(i = 0; i < 1024; i++)
       for(j = 0; j < 1024; j++)
           for(k = 0; k < 1024; k++)
               c[i][j] += a[i,k] * b[k,j];
}
```

מניחים שהקוד הבינארי של התכנית הנ"ל תופס דף זכרון אחד, ואילו המשתנים המקומיים של multiply (הנמצאים ב stack) תופסים דף זכרון אחד אחר. משתנה integer תופס זכרון בן 4 בתים. במערכת הזכרון של המכונה הנ"ל קיים TLB בן 8 כניסות ואלגוריתם החלפה LRU. הנח שהמטריצות מאורגנות בזיכרון ברצף של שורות אלמנטים אחת אחרי השנייה.

חשבו מהו מספר ה TLB misses. הניחו שבתחילת הריצה של multiply הדפים של הקוד הבינארי וה stack מצויים כבר ב TLB.

תשובה:

מאחר והקוד הבינארי והמשתנים המקומיים תופסים דף אחד כל אחד, לכל אחד מהם ישנה כניסה אחת ב TLB. מאחר ובזמן ריצת התכנית ניגשים כל הזמן לקוד הבינארי ולמשתנים המקומיים המצויים ב stack, שתי הכניסות הללו לא ישתנו ב TLB במהלך כל הריצה ולא יגרמו ל miss, משום שאלגוריתם LRU לא יגע בהם. זה מותיר 6 כניסות ב TLB לדפי נתונים.

מאחר ו integer תופס 4 בתים וגדלי המטריצות a, b, c הינן $1K \times 1K$ הרי שגודל כל שורה במטריצות הינו 4KB, כלומר דף שלם.

בלולאה הפנימית שבה משתנה האינדקס k, כל פניה ל $b[k, j]$ נדרשת לדף חדש וגורמת ל miss ב TLB, סה"כ 1024 פעמים. הדבר חוזר על עצמו עם כל שינוי של j בלולאה האמצעית שקורית גם כן 1024 פעמים, וכנ"ל בלולאה החיצונית שמשנה את i. סה"כ misses בשל המטריצה b $1024 \times 1024 \times 1024 = 2^{30}$.

שינויי דפים כתוצאה מהמטריצה a מתרחשים רק בעקבות שינוי i, וזאת משום ש k הינו אינדקס עמודה סה"כ 1024 פעמים.

שינויי דפים כתוצאה מהמטריצה c מתרחשים רק בעקבות שינוי i, וזאת משום ש j הינו אינדקס עמודה, שאיננו מחליף דף נתונים. סה"כ 1024 פעמים.

סה"כ מספר ה misses $2^{30} + 2 \times 2^{10} = 1073743872$.

7. (7 נק') בהנחה שהקומפיילר חכם מספיק, האם ניתן ע"י שנוי ארגון הנתונים בזיכרון להקטין את מספר ה TLB misses ? במדה וכן, כיצד ובכמה, ואם לא אז מדוע?

תשובה: הדבר ניתן לבצוע ע"י ארגון המטריצה b בזיכרון על פי עמודות במקום על פי שורות. שינויי האינדקס k בלולאה הפנימית לא יגרמו להחלפת דפים ב TLB. הדבר יוביל ל $1024 \times 1024 = 2^{20}$ misses במקום 2^{30} קודם.

שאלה ב' (50 נק')

מהם מצבי טבלאות ה RS, ROB, וה- REGISTER FILE המצורפות עבור מעבד ספקולטיבי המממש את אלגוריתם TOMASULO, אשר מריץ את התכניות שלהלן, תחת ההנחות הבאות:

- רק פקודה אחת יכולה לצאת (ISSUE) בכל מחזור שעון.
- ל ROB ישנן 7 כניסות (SLOTS) ופקודות נמחקות מה-ROB לפי סדר **הטיפול** בהם ורק כאשר יש פקודה אחרת המבקשת להיכנס.
- כל היחידות הפונקציונליות (FU) הינן FULLY PIPELINED.
- ישנן 2 יחידות RS לכפל (MULTIPLY), 3 יחידות לחבור (ADD) ו 3 יחידות המשמשות ל LOAD ו STORE. פקודות נמחקות מה-RS רק לאחר שידורן על ה- CDB
- חריגות (EXCEPTIONS) אינן קורות.
- השיהוי בבצוע (EXECUTION) פעולות הינו מחזור שעון אחד עבור פעולת חיבור ו3 מחזורי שעון עבור פעולת כפל.
- להוציא STRUCTURAL HAZARD, לפעולת ה- LOAD דרושים 5 מחזורי שעון על מנת שמידע יהיה זמין לפקודה התלויה במידע. ופקודת ה-STORE מתבצעת במחזור שעון אחד.
- במקרה של קונפליקט בכתיבה ל Common Data Bus (CDB), הקדימות (PRIORITY) לפקודה שיצאה לדרך מוקדם יותר.
- בצוע (EXECUTION) של פקודות תלויות יכול להתחיל במחזור השעון לאחר שנתוניהן משודרים ב CDB.
- מניחים שטבלאות ה RS, ROB ו FP RF הינן ריקות ופנויות בעת תחילת בצוע התכניות שלהלן.
- העמודה "VALUE" מתעדכנת בעת שידור הערך ב CDB.
- הפרדיקציה עבור פקודת ה-BRANCH היא taken.

נתונה התוכנית הבאה:

```
        LD    F0, 0 (R1)
        LD    F1, R2(R1)
LOOP1:  ADD   F0, F0, F1
        ADD   F1, F0, F1
        ADD   R2, 4, R2
        SD    F0, R2(R1)
        ADD   R2, 4, R2
        SD    F1, R2(R1)
        BNE   R2, R3, LOOP1
```

נתון כי R1 מצביע אל תחילת המערך ושני הערכים הראשונים במערך הם 1. כמו כן, R2 מאותחל ל 4 ו R3 מאותחל ל-N>>2.

1. מה מבצעת התוכנית? איזו סדרת מספרים מתקבלת במערך R1 (5 נק')?

תשובה: התוכנית מייצרת את סדרת פיבונצ'י

2. מה מצב הטבלה לאחר 5 מחזורי שעון? מלאו את הטבלאות המצורפות(10 נק'):

Reorder Buffer				
Status (issued, Ex,commit)	Destination	Value	Instruction	Entry
				ROB7
				ROB6
issued	R2		ADD R2, 4, R2	ROB5
issued	F1		ADD F1, F0, F1	ROB4
EX	F0		ADD F0, F0, F1	ROB3
commit	F1	1	LD F1, R2(R1)	ROB2
commit	F0	1	LD F0, 0 (R1)	ROB1

Reservation Station		
Operation	Registers	Destination
MULT (2)		
MULT (1)		
ADDI (3)	4, R2	ROB 5
ADDI (2)	ROB3, ROB2	ROB 4
ADDI (1)	ROB2, ROB1	ROB3
Load/store (3)		
Load/store (2)		
Load/store (1)		

Registers Reorder										
Reg	F0	F1	F2	F3	F4	F5	F6	R1	R2	R3
Reorder#	ROB3	ROB4							ROB5	

3. מה מצב הטבלה לאחר סיום שני סבבים של הלולאה, כלומר לאחר issue לפקודת ה- branch בפעם השניה? (10נק') מלאו את הטבלאות המצורפות:

Reorder Buffer				
Status (issued, Ex,commit)	Destination	Value	Instruction	Entry
commit	R2	R2+16	ADD R2, 4, R2	ROB7
commit	-	-	SD F0, R2(R1)	ROB6
commit	R2	R2+12	ADD R2, 4, R2	ROB5
commit	F1	8	ADD F1, F0, F1	ROB4
commit	F0	5	ADD F0, F0, F1	ROB3
issued	-	-	BNE R2, R3, LOOP1	ROB2
EX	-	-	SD F1, R2(R1)	ROB1

Reservation Station		
Operation	Registers	Destination
MULT	empty	
MULT	empty	
ADDI	empty	
ADDI	empty	
ADDI	empty	
Load/store	empty	
Load/store	empty	
Load/store	ROB 3	R2(R1)

Registers Reorder										
Reg	F0	F1	F2	F3	F4	F5	F6	R1	R2	R3
Reorder#	ROB6	ROB1							ROB7	

כעת, רגיסטר R1 מצביע על האיבר השני במערך שנוצר בסעיף הקודם (אין חשש לחריגה מגודל המערך), ומריצים את הקוד הבא:

```
LD    F0, 0 (R1)
LD    F1, 4(R1)
```

```
LD      F2, 8 (R1)
LD      F3, 12(R1)
MULT F4, F0, F3
SD      F4, 0(R4)
MULT F4, F2, F1
MULT F4, F4, 2
SD      F4, 4(R4)
MULT F1, F1, F1
MULT F2, F2, F2
ADD F4, F1, F2
SD      F4, 8(R4)
```

נתון כי R4 מצביע על מערך חדש.

4. מה מצב הטבלאות לאחר 5 מחזורי שעון? מלאו את הטבלאות המצורפות (ניתן להניח כי הטבלה נקייה משאריות מהסעיפים הקודמים וכי היא ריקה בתחילת הקוד הנ"ל)(10 נק'):

Reorder Buffer				
Status (issued, Ex,commit)	Destination	Value	Instruction	Entry
				ROB7
				ROB6
				ROB5
issued			LD F3, 12(R1)	ROB4
EX			LD F2, 8 (R1)	ROB3
EX			LD F1, 4(R1)	ROB2
Commit		1	LD F0, 0 (R1)	ROB1

*פקודת ה MULT לא נכנסה כיוון שפקודת ה LOAD לא יכולה להיכנס עד אשר הפקודה הראשונה של הLOAD עושה commit כיוון שיש מקום רק ל 3 פקודות LOAD בו זמנית

Reservation Station		
Operation	Registers	Destination
MULT		

MULT		
ADDI		
ADDI		
ADDI		
Load/store	ROB 3	12(R1)
Load/store	ROB 2	8 (R1)
Load/store	ROB1	4(R1)

Registers Reorder										
Reg	F0	F1	F2	F3	F4	F5	F6	R1	R2	R3
Reorder#	ROB1	ROB2	ROB3	ROB4						

5. מה מצב הטבלאות לאחר כניסת השורה האחרונה של הקוד לתוך reorder buffer? (10 נק') מלא את הטבלה הבאה:

Reorder Buffer				
Status (issued, Ex,commit)	Destination	Value	Instruction	Entry
issued	F4		ADD F4, F1, F2	ROB7
commit			SD F4, 0(R4)	ROB6
issued			SD F4, 8(R4)	ROB5
EX	F2		MULT F2, F2, F2	ROB4
commit	F1	4	MULT F1, F1, F1	ROB3
commit			SD F4, 4(R4)	ROB2
commit	F4	12	MULT F4, F4, 2	ROB1

Reservation Station		
Operation	Registers	Destination
MULT	ROB 4, ROB 4	ROB 4
MULT		

ADD		
ADD		
ADD	ROB 3, ROB 4	ROB7
Load/store		
Load/store		
Load/store	ROB 5	8(R4)

Registers Reorder										
Reg	F0	F1	F2	F3	F4	F5	F6	R1	R2	R3
Reorder#		ROB3	ROB4		ROB7					

6. כמה מחזורי שעון נדרשים על מנת לבצע commit לכל הפקודות נ"ל? פרטו את החישוב! (5 נק')

23 מחזורי שעון, ישנן 13 פקודות ואליהן נוספים 2 השהיות לפני כניסת פקודת הLOAD הרביעית. 2 השהיות לפני כניסת פקודת ה-MULT השלישית אשר מתעכבת בביצוע 2 מחזורי שעון. ולכן 4 השהיות לפני כניסת פקודת ה- MULTהחמישית. בנוסף פקודת ה- MULT החמישית מסתיימת במחזור שעון 22 פקודת ה-ADD מסתיימת במחזור 23 SD 24 וב-25 הופכת ל- commit.

7. **בונוס**: מה עושה הקוד בחלק השני (2 נק')

מייצר שלשות פיתגוריות ע"י סדרת פיבונצ'י