

מבנה מחשבים
234267

תרגול מס' 1:

פרמטרים של הערכת ביצועים
חוק אמדל

מבנה מחשבים

- Grade
 - 20% : 4 exercises (all mandatory) תקן
 - Submission in pairs
 - 80% Final exam
- Come to the lectures and to the tutorials !
 - The material for the exam includes all that is taught during the lectures and the tutorials – the foils do not contain everything
 - Students who don't show up, and/or don't listen carefully, and/or don't do the HW on their own get low grades in the exam
- Course web site
 - <http://webcourse.cs.technion.ac.il/234267/>
 - Foils will be on the web several days before each lecture

זמן CPU ומדידת ביצועים

מושגים והגדרות

CPU – יחידת העיבוד המרכזית של המחשב. זהו רכיב סינכרוני (עובד בהתאם לאות של שעון פנימי). ה- **ALU** הינו החלק ב- CPU שמבצע את החישובים האריתמטיים.

מחזור השעון (Clock Cycle – CC): משך הזמן בין שתי פעימות עוקבות שמבצע השעון הפנימי של המחשב. נמדד בשניות ($1_{\text{nanosec}} = 10^{-9}_{\text{sec}}$).

קצב השעון (Clock Rate – CR): מספר הפעימות שמבצע השעון הפנימי של המחשב בשנייה אחת. נמדד בהרצים (**Hz**) (מגה הרץ = מליון פעימות לשנייה, $1\text{Mhz} = 10^6 \text{hz}$).

$$ClockRate_{(Hz)} = \frac{1}{ClockCycle_{(sec)}}$$

ארכיטקטורה מול מיקרו ארכיטקטורה

- שני דברים שמשלימים זה את זה
- **ארכיטקטורת** המעבד מגדירה את הממשק בין המשתמש (קומפילר או מתכנת) לבין המעבד
 - הממשק: סט הפקודות (סוג פקודות, מספר אופרנדים, שיטות מיעון וכדומה).
למעבדים שונים תתכן ארכיטקטורה שונה
- **מיקרו ארכיטקטורה** - המבנה הפנימי של המעבד.

• (IC) Instruction Count

– מספר פקודות המכונה (לא שורות!) בתכנית

• מחזורים לפקודה (CPI = Cycles Per Instruction)

– מספר מחזורי השעון הדרושים בממוצע לביצוע כל פקודה בתכנית. שימו לב – זהו פרמטר שתלוי בתוכנית.

$$CPI = \frac{\text{Total cycles in program runtime}}{\text{Total instructions in program}}$$

• מחזורים לפקודה i (CPI_i = Cycles Per Instruction i)

– מספר מחזורי השעון הדרושים לביצוע פקודה אחת מסוג i

• זמן ריצה או זמן CPU (execution time)

– הזמן שלוקח למעבד לבצע תכנית. נמדד בשניות. ניתן לחישוב ע"י

$$CPU\ time\left(\frac{seconds}{program}\right) = CC\left(\frac{seconds}{cycle}\right) * \text{number of cycles}\left(\frac{cycles}{program}\right)$$

$$CPU\ time\left(\frac{seconds}{program}\right) = CC\left(\frac{seconds}{cycle}\right) * CPI\left(\frac{cycles}{instruction}\right) * IC\left(\frac{instruction}{program}\right)$$

חישוב CPI לתכנית מבחן (benchmark)

I_i – מספר הפעמים שהפקודה i בוצעה בתכנית.

IC – מספר ההוראות הכולל שבוצע בתכנית. $IC = \sum_i I_i$

P_i – השכיחות של פקודה i בתכנית

$$CPI = \frac{Total\ Cycles}{IC} = \frac{\sum (CPI_i * I_i)}{\sum I_i} = \sum (CPI_i * P_i)$$

שאלה: האם **CPI** נמוך יותר גורר בהכרח מחשב מהיר יותר?

$$CPU\ time\left(\frac{seconds}{program}\right) = CC\left(\frac{seconds}{cycle}\right) * CPI\left(\frac{cycles}{instruction}\right) * IC\left(\frac{instruction}{program}\right)$$

דוגמא א'

נתון מחשב בו כל הפקודות מתבצעות בין רגיסטרים, והגישה לזיכרון היא רק דרך load\store.
שכיחות הפקודות בתכנית המבחן:

<u>CPI</u>	<u>שכיחות</u>	<u>הפקודה</u>
1	40%	ALU
2	20%	LOAD
2	10%	STORE
2	30%	BRANCH

בבדיקה נמצא **שרבע מפקודות ה-ALU** משתמשות באחד האופרנדים בנתון שהובא בפקודת
LOAD קודמת ומשמש רק לפקודת ALU זאת

שאלה: מוצע להוסיף לסט הפקודות פקודת ALU שמבצעת ישירות פעולה בין רגיסטר לבין תא
בזיכרון (במקום LOAD ואחריו פקודת ALU בין שני רגיסטרים). פקודה זאת תבוצע עם $CPI=2$
אבל תגדיל את ה CPI של פקודת BRANCH ל-3. האם השיפור כדאי?

CPI	שכיחות חדשה	שכיחות	הפקודה
1	0.3/0.9=4/9	40%	ALU
2	0.1/0.9=2/9	20%	LOAD
2	0.1/0.9=1/9	10%	STORE
2/3	0.3/0.9=1/3	30%	BRANCH
-/2	0.1/0.9=1/9	-	ALU-LOAD

פתרון

לפני השיפור:

$$CPI_{(old)} = 0.4*1 + 0.2*2 + 0.1*2 + 0.3*2 = 1.6$$

$$CPU_Time_{(old)} = IC_{(old)} * CPI_{(old)} * CC_{(old)} = 1.6 * IC_{(old)} * CC_{(old)}$$

אחרי השיפור:

$$CPI_{new} = \frac{Cycles_{new}}{IC_{new}} = \frac{\sum_i [(IC_{old} * percentage_{i(new)}) * CPI_{i(new)}]}{IC_{new}} =$$

$$= \frac{IC_{old}}{IC_{old} * (1 - 0.4/4)} * [(0.4 - 0.4/4) * 1 + (0.2 - 0.4/4) * 2 +$$

$$+ 0.4/4 * 2 + 0.1 * 2 + 0.3 * 3] = \frac{1.8}{1 - 0.4/4} = 2$$

$$CPU\ Time_{new} = IC_{new} * CPI_{new} * CC_{new} = (IC_{old} * (1 - 0.4/4)) * 2 * CC = 1.8 * IC_{old} * CC$$

$$\frac{CPU\ Time_{new}}{CPU\ Time_{old}} = \frac{1.8 * IC_{old} * CC}{1.6 * IC_{old} * CC} = 1.125$$

דרך פתרון שונה, נוחה יותר, לחישוב זמן הריצה לאחר השינוי:

נאמר שבתכנית המקורית יש 100 פקודות, ואז ע"פ האחוזים הנתונים, אפשר לדעת מהו מספר הפקודות מכל סוג:

הפקודה	מספר פקודות לפני השינוי	CPI _i לפני השינוי	מספר פקודות אחרי השינוי	CPI _i אחרי השינוי
ALU	40	1	30	1
LOAD	20	2	10	2
STORE	10	2	10	2
BRANCH	30	2	30	3
MEM-ALU			10	2

מתוך 40 פקודות ALU שהיו – עכשיו רק $\frac{3}{4}$ נשארו פקודות ALU, כלומר $100 * 0.4 * \frac{3}{4} = 30$

שאר $\frac{1}{4}$ פקודות ALU הפכו לפקודות MEM-ALU: $100 * 0.4 * \frac{1}{4} = 10$

כל פקודת MEM-ALU מחליפה גם פקודת LOAD, ולכן נותרו רק $10 = 20 - 10$ פקודות LOAD

הפקודה	מספר פקודות לפני השינוי	CPI _i לפני השינוי	מספר פקודות אחרי השינוי	CPI _i אחרי השינוי
ALU	40	1	30	1
LOAD	20	2	10	2
STORE	10	2	10	2
BRANCH	30	2	30	3
MEM-ALU			10	2

שימו לב! מספר הפקודות הכולל ירד עקב השינוי:

$$IC = 30 + 10 + 10 + 30 + 10 = 90$$

כזכור:

$$CPU\ time = CC * CPI * IC = CC * \text{number of cycles}$$

נשווה את מספר מחזורי השעון לפני ואחרי השינוי:

$$Cycles_new = 30*1 + 10*2 + 10*2 + 30*3 + 10*2 = 180$$

$$Cycles_old = 40*1 + 20*2 + 10*2 + 30*2 = 160$$

מכיוון שקצב השעון (CC) לא השתנה, מספיק להשוות את מספר מחזורי השעון לפני ואחרי השינוי (אחרת היינו צריכים להכפיל ב-CC החדש כדי לחשב את ה-CPUtime).

מדדים "אובייקטיביים"

MIPS = Millions of Instructions Per Second

כמה מיליוני פעולות מבצע המעבד בשנייה:

$$MIPS = \frac{IC}{CPU\ time_{(sec)} * 10^6} = \frac{IC}{IC * CPI * CC * 10^6} = \frac{clock\ rate_{(cycle/sec)}}{CPI_{(cycle/instruction)} * 10^6}$$

בעיה: בניגוד ל-CPU_time, מדד MIPS תלוי בסט הפקודות של המכונה ובהרכב תכנית הבדיקה. אם במעבד מסוים יש פקודות אסמבלר מורכבות יותר, שלוקח הרבה זמן לבצע אותן, אבל עקב כך צריך להשתמש בפחות פקודות מאשר במעבדים אחרים כדי לבצע את אותם דברים, אז מדד ה-MIPS שיתקבל עבור אותו מחשב יהיה נמוך, ולא בהכרח בצדק.

לדוגמא: עבור תכנית בשפה עילית המורצת ע"י שני מעבדים שונים, עבור מעבד-1 מתקיים IC=1000 וזמן הריצה הוא 2 שניות, ועבור מעבד-2 מתקיים IC=100 וזמן הריצה הוא שניה אחת. אזי $MIPS_1 = 500/10^6$ גדול מ- $MIPS_2 = 100/10^6$, בעוד מעבד-2 מהיר יותר!

דוגמא ב'

נתונים שני מחשבים בעלי תדר שעון של 100MHz, המריצים אותה תכנית מבחן (benchmark) בעלת פילוג הפעולות (בשפה עילית) הבא:

שכיחות	פעולה
10%	כפל FP
15%	חיבור FP
5%	חילוק FP
70%	פעולות INT

המחשב הראשון ("MFP"), בעל חומרה ייעודית לחישוב פעולות FP, מבצע כל פעולת FP בשפה עילית ע"י פעולת-מכונה FP אחת, וכל סוג פעולה מתבצעת במספר מחזורי שעון שונה. המחשב השני ("NMFP") יכול לבצע רק פעולות בשלמים, והקומפיילר מתרגם כל פעולת FP בשפה עילית למספר פעולות INT שאורכות 2 מחזורי שעון כל-אחת.

פעולה	מספר פעולות INT שמבוצעות לכל פקודת FP במחשב NMFP
כפל FP	30
חיבור FP	20
חילוק FP	50

פעולה	מספר מחזורי שעון במחשב MFP
כפל FP	6
חיבור FP	4
חילוק FP	20
פעולות INT	2

מספר פעולות INT שמבוצעות לכל פקודת FP במחשב NMFP	פעולה
30	כפל FP
20	חיבור FP
50	חילוק FP

מספר מחזורי שעון במחשב MFP	פעולה
6	כפל FP
4	חיבור FP
20	חילוק FP
2	פעולות INT

שכיחות	פעולה
10%	כפל FP
15%	חיבור FP
5%	חילוק FP
70%	פעולות INT

א. חשב את מדד ה- MIPS של שתי המכונות.
להזכירכם:

$$MIPS = \frac{\text{Instruction count}_{(instructions)}}{\text{CPU time}_{(sec)} * 10^6} = \frac{\text{clock rate}_{(cycle/sec)}}{CPI_{(cycle/instruction)} * 10^6}$$

$$CPI_{MFP} = (0.10 * 6) + (0.15 * 4) + (0.05 * 20) + (0.70 * 2) = 3.6$$

$$CPI_{NMFP} = 2$$

הסבר: מחשב ה- NMFP מבצע רק פעולות INT שאורכות 2 מחזורי שעון.

$$MIPS_{MFP} = \frac{\text{clock rate}_{(cycle/sec)}}{CPI_{(cycle/instruction)} * 10^6} = \frac{100 * 10^6}{3.6 * 10^6} = 27.8$$

$$MIPS_{NMFP} = \frac{100 * 10^6}{2 * 10^6} = 50$$

שימו לב שמדד ה MIPS של מחשב NMFP גבוה מזה של MFP !

ב. מחשב MFP מבצע את התכנית ב- $300 \cdot 10^6$ פקודות. כמה פקודות Integer נדרשות למחשב NMFP כדי לבצע את התכנית?

מספר פעולות NMFP	פקודות INT לכל פקודת FP	מספר פעולות MFP	שכיחות	פעולה
$900 \cdot 10^6$	30	$30 \cdot 10^6$	10%	כפל FP
$900 \cdot 10^6$	20	$45 \cdot 10^6$	15%	חיבור FP
$750 \cdot 10^6$	50	$15 \cdot 10^6$	5%	חילוק FP
$210 \cdot 10^6$	1	$210 \cdot 10^6$	70%	פעולות בשלמים
$2760 \cdot 10^6$		$300 \cdot 10^6$		סה"כ

ג. מהו זמן הריצה של התכנית בשני המחשבים?

$$\begin{aligned}
 \text{CPU Time}_{\text{MFP}} &= \sum_i (IC_i * CPI_i) * CC = \\
 &= [(30 \cdot 10^6 * 6) + (45 \cdot 10^6 * 4) + (15 \cdot 10^6 * 20) + (210 \cdot 10^6 * 2)] * 10^{-8} = \\
 &= 10.8_{(\text{sec})} = \frac{IC}{\text{MIPS}} = \frac{300 \cdot 10^6_{(\text{instr})}}{27.8 \cdot 10^6_{(\text{IPS})}}
 \end{aligned}$$

$$\begin{aligned}
 \text{CPU Time}_{\text{NMFP}} &= IC * CPI * CC = \\
 &= (2760 \cdot 10^6) * 2 * 10^{-8} = 55.2_{(\text{sec})} = \frac{IC}{\text{MIPS}} = \frac{2760 \cdot 10^6_{(\text{instr})}}{50 \cdot 10^6_{(\text{IPS})}}
 \end{aligned}$$

שימו לב להבדלים בזמן הריצה: מעבד MFP מהיר יותר (למרות שמדד ה-MIPS שלו נמוך יותר)

MFLOPS – Millions of Floating Point Operations Per Second

הכוונה היא לכמה פעולות floating point (פעולות אריתמטיות על משתנים מסוג float או double) **בשפה עילית** (כמו C) ולא בשפת אסמבלר.

$$MFLOPS = \frac{\text{FP Operations per Program}}{\text{CPU time}_{(\text{sec})}} * 10^6$$

בעיה 1: MFLOPS מחושב ביחס לזמן הריצה של תכנית בשפה עילית ולכן תלוי בקומפילר. קומפילר טוב יבצע אופטימיזציה ולכן זמן הריצה יהיה קצר יותר.

בעיה 2: פעולות שונות אורכות משכי זמן שונים (למשל: חיבור שני מספרי FP הוא מהיר לעומת חילוק). MFLOPS תלוי בסוג הפעולות המופיע בתוכנית.

פתרון לבעיה 2: נגדיר מדד חדש: **MFLOPS משוקלל** - נוסיף שקלול של פעולות FP שונות עפ"י זמן הביצוע שלהן. כלומר, סופרים את מספר ההופעות בתכנית של סוגי הפעולות השונות, ונותנים להם משקל שונה בציון הסופי.
לדוגמא:

Add, sub cmp, mult	1
Div, sqrt	4
Exp, sin	8

ד. מהו מדד MFLOPS של שני המחשבים?

$$MFLOPS_{MFP} = \frac{\# \text{ of } FP \text{ instructions}}{\text{Execution Time} * 10^6} = \frac{30 * 10^6 + 45 * 10^6 + 15 * 10^6}{10.8 * 10^6} = \frac{90}{10.8} = 8.33$$

$$MFLOPS_{NMF} = \frac{30 * 10^6 + 45 * 10^6 + 15 * 10^6}{55.2 * 10^6} = \frac{90}{55.2} = 1.63$$

הסבר: מספר פעולות ה-FP בשני המחשבים שווה, כי סופרים FP בשפה עילית ולכן לא חשוב איך הן ממומשות בקוד מכונה. הדבר השונה בחישוב הוא זמן הביצוע (שחישבנו בסעיף ג').

נשים לב שמדד MFLOPS של מחשב FP גבוה יותר ממדד ממחשב NFP, כלומר מכונה FP מבצעת מהר יותר פעולות FP (בשפה עילית), וזאת בניגוד למדד MIPS.

בעיה משותפת לשני המדדים הקודמים: שניהם תלויים בתוכנית שבחרנו כ-benchmark. צריך תמיד להשקיע מחשבה בשאלה איזו תכנית נבחר.

• **מדדים אמיתיים** – תכניות אמיתיות (מעבד תמלילים, גיליון אלקטרוני, תכנת נווט)

• **מדדים סינטטיים** – אוסף פקודות שאמור לייצג תכנית ממוצעת (יש דבר כזה?).

• **מדדים משולבים** – אוסף תוכיות אמיתיות וסינטטיות, משוקללות עפ"י מפתח כלשהו.

שני המדדים שראינו הם מלאכותיים ובעייתיים, ולכן נשתדל להשוות בין מעבדים בתנאים כמה שיותר דומים: אותה תכנית בדיקה, אותו קומפיילר וכדומה.

להגברת הדיוק, רצוי להריץ כמה תכניות ולשקלל את התוצאות.

המדד הכי טוב יהיה השוואת זמני הביצוע של אותן תכניות ע"ג כל אחד מהמעבדים.

חוק אמדל

99%

- ניקח מכונה כלשהי שזמן הריצה של תכנית P עליה הוא $ExTime_{old}$.
- נניח שאנו משפרים את המעבד כך שחלק $Fraction$ מזמן ריצת התכנית ירוץ פי $Speedup$ יותר מהר בזכות השיפור
- השאלה היא מה יהא זמן הריצה כעת?

$$ExTime_{new} = ExTime_{old} \times \left[(1 - Fraction) + \frac{Fraction}{Speedup} \right]$$

- שימו לב: $Fraction$ מתייחס לחלק הזמן, לא לחלק הפקודות או לחלק התכנית P.

חוק אמדל

- פועל יוצא הוא שחישוב ההאצה הכללית שהשגנו ניתן לחישוב באופן הבא:

$$t'_{\text{exeNew}} = t_{\text{exeOld}} \times \left[(1 - \text{Fraction}) + \frac{\text{Fraction}}{S} \right]$$

$$\text{Speedup}_{\text{overall}} = \frac{t_{\text{exeOld}} - t'_{\text{exeNew}}}{t_{\text{exeOld}}} = 1 - \frac{t'_{\text{exeNew}}}{t_{\text{exeOld}}} = 1 - \left[(1 - F) + \frac{F}{S} \right] = F - \frac{F}{S}$$

דוגמא

- שינינו מעבד כלשהו באופן הבא:
 - ההוראות שמטפלות בנקודה צפה (floating point) ירוצו פי 2.5 יותר **מהר**
 - פעולות גישה לזיכרון פי 3 יותר **מהר**
 - פעולות חיבור/חיסור בשלמים פי 1.5 יותר **לאט**
- מבדיקה עבור תוכנית מבחן עולה ש:
 - פעולות נקודה צפה תופסות 15% זמן מכלל תכנית המבחן
 - פעולות זיכרון 20%
 - פעולות חיבור/חיסור בשלמים – 40%
- כמה שיפרנו בסך הכל?

Operation	Speed up	Frequency
Floating point	2.5	15%
Memory	3	20%
Integer	1/ 1.5	40%

$$Time_{new} =$$

$$= Time_{old} \times \left[(1 - (0.15 + 0.20 + 0.40)) + \left(0.15 * \frac{1}{2.5} \right) + \left(0.20 * \frac{1}{3} \right) + (0.40 * 1.5) \right]$$

$$= Time_{old} \times \left[0.25 + \frac{0.15}{2.5} + \frac{0.20}{3} + \frac{0.40}{(1/1.5)} \right] =$$

$$= Time_{old} \times 0.98$$

$$Speedup = 1 - \frac{ExeTime\ New}{ExeTime\ Old} = 1 - 0.98 = 2\%$$

ניתן להכליל גם לחישוב CPI

$$CPI_{new} = CPI_{old} \times \left[(1 - Fraction) + \frac{Fraction}{Speedup(in\ cycles)} \right]$$

ניתן לשימוש רק כאשר מספר הפקודות לא השתנה
(Fraction צריך להיות תקף בשני המקרים)

לדוגמא

- נתונה מכונה CISC-ית, רוצים להוציא דגם משופר מבוסס RISC כך שהוראות מסובכות יפורקו (בפנים) למספר הוראות פשוטות
- כמו כן הצליחו במעבד החדש להשיג ביצוע הוראות **סיעוף מהיר פי 2** (במ"ש) מאשר במעבד המקורי. מהרצת תוכנית מבחן התברר ש:
 - עבור 20% מזמן הביצוע של הקוד קיבלנו האטה של 70% במ"ש (אין שינוי בשאר ה-80%).
 - הוראות סיעוף תופסות 10% מזמן הביצוע (שאינן נמנות על ה-20% שגרמו לגידול).
 - פי כמה השתנה ה-CPI?

	20%	70% slower
	70%	No change
branch	10%	*2

$$\frac{CPI_{new}}{CPI_{old}} = (1 - 0.2 - 0.1) + 0.2 \times 1.7 + 0.1 \times 0.5 = 1.09$$

דוגמא (2)

מהרצת תוכנית אחרת:

- מתברר כי מניין ההוראות בקוד הדינאמי גדל ב-50%
- תדר השעון לעומת זאת גדל פי 2
- היחס בין ה- CPI_{old} ל- CPI_{new} נשאר כמו בתוכנית המבחן
- מהו שיפור הביצועים, אם בכלל, עבור תוכנית המבחן הנתונה?

$$\begin{aligned} Speedup_{overall} &= 1 - \frac{CPU\ Time_{new}}{CPU\ Time_{old}} = 1 - \frac{CC_{new}}{CC_{old}} \times \frac{CPI_{new}}{CPI_{old}} \times \frac{IC_{new}}{IC_{old}} \\ &= 1 - \frac{0.5 * CC_{old}}{CC_{old}} \times \frac{1.09 * CPI_{old}}{CPI_{old}} \times \frac{1.5 * IC_{old}}{IC_{old}} = 1 - \frac{1.09 * 1.5}{2} = 18.25\% \end{aligned}$$

שאלות

א. נתונה תכנית מחשב **בשפה עלית** אשר מכילה 20 פעולות FP. התכנית רצה מהר יותר על מחשב A מאשר על מחשב B. אילו מהקביעות הבאות נכונה **בהכרח** לגבי המחשבים הנ"ל והתכנית הנתונה?

1. למחשב A מדד MIPS גבוה או שווה לזה של מחשב B
2. למחשב A מדד MFLOPS גבוה או שווה לזה של מחשב B
3. ה-CPI של מחשב A גבוה או שווה ל-CPI של מחשב B
4. ה-CPI של מחשב A נמוך או שווה ל-CPI של מחשב B
5. קצב השעון של מחשב A גבוה או שווה לקצב השעון של מחשב B
6. זמן המחזור של מחשב B ארוך או שווה לזמן המחזור של מחשב A
7. אף תשובה אינה מתקיימת בוודאות.

שאלות - המשך

מה תהיה תשובתך לסעיף א' כאשר מדובר על תכנית [אסמבלר](#) זהה הרצה על שני המחשבים?

1. למחשב A מדד MIPS גבוה או שווה לזה של מחשב B
2. למחשב A מדד MFLOPS גבוה או שווה לזה של מחשב B
3. ה-CPI של מחשב A גבוה או שווה ל-CPI של מחשב B
4. ה-CPI של מחשב A נמוך או שווה ל-CPI של מחשב B
5. קצב השעון של מחשב A גבוה או שווה לקצב השעון של מחשב B
6. זמן המחזור של מחשב B ארוך או שווה לזמן המחזור של מחשב A
7. אף תשובה אינה מתקיימת בוודאות.