מבוא לארגון המחשב

- היסטוריה ורקע כללי
- סוגי מעבדים
- הערכת ביצועי מעבד
- רקע מתמטי - המרת בסיסים
- פעולות שונות בבסיס בינארי
- אלגברה בוליאנית ושערים לוגיים

מבוא לארגון המחשב - סיכום

סימון	פירוש	יחידות	מושג
CCT	Clock Cycle Time	[sec]	זמן מחזור שעון
CR	Clock Rate	[1/sec]=[Hz]	תדירות השעון
CC	Clock Cycle	[Cc/program]	מספר מחזורי שעון בתוכנית
IC	Instruction Count	[ins/program]	מספר פקודות מכונה בתוכנית
CPI	Clock Per Instruction	[Cc/ins]	מספר ממוצע של מחזורי שעון לפקודה אחת
CPU Time	Central Processing Unit Time	[sec/program]	זמן הריצה של תוכנית
Speedup			גורם ההאצה (חלוקת הביצועים האיטיים במהירים)
MIPS	Million Instructions Per Second	[MIPS]	מיליון פקודות בשנייה

נוסחאות	יחידות	מושג
$CPI = \sum_i CPI_i \cdot \omega_i$	[Cc/ins]	מספר ממוצע של מחזורי שעון לפקודה אחת
$CPU\ Time = CC \cdot CCT = \frac{CC}{CR} = IC \cdot CPI \cdot CCT = \frac{IC \cdot CPI}{CR}$	[sec/program]	זמן הריצה של תוכנית
$MIPS = \frac{IC}{CPU\ Time \cdot 10^6} = \frac{CR}{CPI \cdot 10^6}$	[MIPS]	מיליון פקודות בשנייה
$Speedup = \frac{CPU\ Time\ SLOW}{CPU\ Time\ FAST}$		מדד האצה (להשוואה בין מעבד איטי למעבד מהיר)
$Speedup = \frac{1}{\left(1 - F_{\text{החלק המשופר}}\right) + \frac{F_{\text{החלק המשופר}}}{Speedup_{\text{אחוז השיפור}}}}$		חוק אמדל (לחישוב השיפור הכללי של תוכנית שרק חלק ממנה עבר שיפור)

מספרים גדולים או קטנים מאוד						
m	10^{-3}	מילי		K	10^{3}	קילו
μ	10^{-6}	מיקרו		M	10^{6}	מגה
n	10^{-9}	ננו		G	10^{9}	גיגה
p	10^{-12}	פיקו		T	10^{12}	טרה

המרה מבסיס עשרוני לכל בסיס אחר

מחלקים את המספר העשרוני בבסיס r. שארית החלוקה ב-r יוצרת את המספר בבסיס r אליו אנו מנסים להמיר. המספר נבנה מימין לשמאל. בכל פעם רושמים את שארית החלוקה ומחלקים שוב את התוצאה בבסיס r. ממשיכים לחלק שוב ושוב עד שתוצאת החילוק שווה 0 (שהמחולק קטן מהבסיס):

לדוגמה, בכדי להמיר את המספר 158 לבסיס בינארי:

$\frac{1}{2}=0$	$\frac{2}{2}=1$	$\frac{4}{2}=2$	$\frac{9}{2}=4$	$\frac{19}{2}=9$	$\frac{39}{2}=19$	$\frac{79}{2}=39$	$\frac{158}{2}=79$
1	0	0	1	1	1	1	0

המרה מבסיס כלשהו לבסיס עשרוני

כופלים כל מספר בבסיס r בחזקת המיקום שלו (המספר הימני ביותר במיקום 0, והמספר השמאלי ביותר במיקום n):

$$a_n \cdot r^n + a_{n-1} \cdot r^{n-1} + a_{n-2} \cdot r^{n-2} + \cdots + a_0 \cdot r^0$$

לדוגמה, בכדי להמיר את המספר 158 בבסיס בינארי לבסיס עשרוני, נשתמש בטור חזקות:

$$1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 1 \cdot 128 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 = 158$$

מבוא לארגון המחשב - סיכום

<u>המרה מהירה בין בסיסים 2, 4, 8, 16</u>

ספרה בבסיס	שווה ל-____ ספרות בבסיס בינארי	כיצד לזכור?
4	2	$2^2 = 4$
8	3	$2^3 = 8$
16	4	$2^4 = 16$

לדוגמה:

158 בבסיס 4	2		1		3		2	
158 בבסיס בינארי	1	0	0	1	1	1	1	0

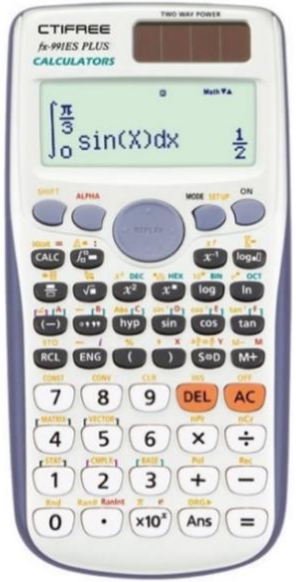
158 בבסיס אוקטלי	2			3			6		
158 בבסיס בינארי	0	1	0	0	1	1	1	1	0

158 בבסיס הקסאדצימלי	9				E			
158 בבסיס בינארי	1	0	0	1	1	1	1	0

מספרים חשובים בבסיסים נפוצים																		
בינארי	(2)	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111	10000
אוקטלי	(8)	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	20
דצימלי	**(10)**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
הקסדצימלי	(16)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10

מבוא לארגון המחשב - סיכום

<u>שימוש במחשבון להמרה בין בסיסים 10, 2, 8, 16</u> (מספרים שליליים בבסיס בינארי בשיטת משלים ל-2)



$MODE \rightarrow 4: BASE - N \rightarrow$ בחירת בסיס $(DEC\ /\ HEX\ /\ BIN\ /\ OCT) \rightarrow$ כתיבת מספר $\rightarrow = \rightarrow$ בחירת בסיס אחר

(לחזרה למצב חישוב רגיל: $MODE \rightarrow 1: COMP$)

<u>מחשבון אינטרנטי להמרה בין כל הבסיסים (לא עובד במספרים שליליים)</u>

http://www.unitconversion.org/unit_converter/numbers.html

מבוא לארגון המחשב - סיכום

הצגת מספרים עם סימן (חיובי ושלילי) בשיטת משלים ל-2

9	7	10	6	11	5	12	4	13	3	14	2	15	1
1001	0111	1010	0110	1011	0101	1100	0100	1101	0011	1110	0010	1111	0001
-7		-6		-5		-4		-3		-2		-1	
$\begin{array}{rrrrr} & 1 & 1 & 1 & \\ & 0 & 1 & 1 & 1 \\ + & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$		$\begin{array}{rrrrr} & 1 & 1 & & \\ & 0 & 1 & 1 & 0 \\ + & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$		$\begin{array}{rrrrr} & 1 & 1 & 1 & \\ & 0 & 1 & 0 & 1 \\ + & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$		$\begin{array}{rrrrr} & 1 & & & \\ & 0 & 1 & 0 & 0 \\ + & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$		$\begin{array}{rrrrr} & 1 & 1 & 1 & \\ & 0 & 0 & 1 & 1 \\ + & 1 & 1 & 0 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$		$\begin{array}{rrrrr} & 1 & 1 & & \\ & 0 & 0 & 1 & 0 \\ + & 1 & 1 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$		$\begin{array}{rrrrr} & 1 & 1 & 1 & \\ & 0 & 0 & 0 & 1 \\ + & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}$	

הפיכת מספר חיובי למספר שלילי בשיטת משלים ל-2

1. על פי ההגדרה המתמטית: $-N = (2^n - N)$, לדוגמה, המספר -5 בהצגה של 4 ביטים הוא:
2. ביצוע הפעולה "NOT" על המספר החיובי והוספת המספר 1:
   לדוגמה, המספר -5 בהצגה של 4 ביטים הוא: $NOT(0101) = 1010 + 1 = 1011$
3. מעבר על כל הסיביות של המספר החיובי מימין לשמאל עד לסיבית הראשונה שערכה 1, העתקה של סיבית זו כפי שהיא והפיכת כל הסיביות שמשמאלה באמצעות הפעולה "NOT"

$$\begin{array}{rrrrr} & -1 & -1 & -1 & \\ 1 & 0 & 0 & 0 & 0 \\ - & & 1 & 0 & 1 \\ \hline & 1 & 0 & 1 & 1 \end{array}$$

תחום המספרים להצגה ב-n ביטים

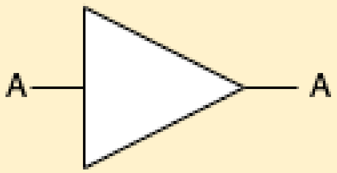
מספרים ללא סימן	מספרים עם סימן בשיטת משלים ל-2
חיוביים: 0 עד $2^n - 1$	חיוביים: עד $2^{n-1} - 1$; שליליים: עד 2^{n-1}

זיהוי גלישה בחיבור שני מספרים

מספרים ללא סימן	מספרים עם סימן בשיטת משלים ל-2
הנשא (carry) האחרון שווה 1	הנשא (carry) האחרון ואחד לפניו שונים זה מזה

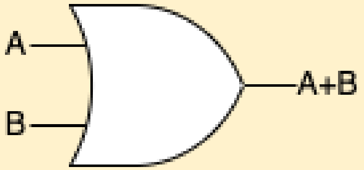
מבוא לארגון המחשב - סיכום

Buffer



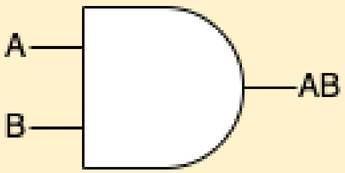
A	A
0	0
1	1

OR (+)



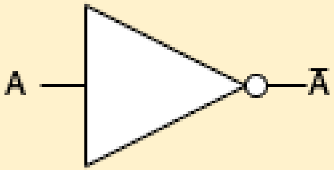
A	B	$A+B$
0	1	1
1	0	1
1	1	1
0	0	0

AND (·)



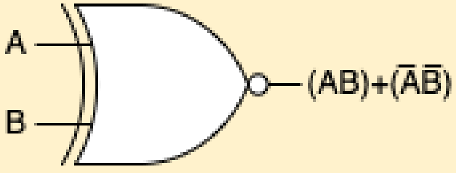
A	B	$A \cdot B$
0	1	0
1	0	0
1	1	1
0	0	0

NOT (A' או $\bar{A}$)



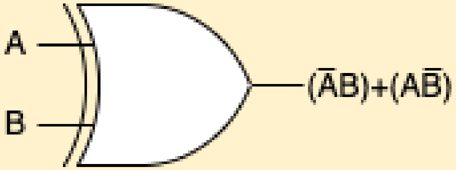
A	$\bar{A}$
0	1
1	0

XNOR ($\odot$)=$(AB)+(\bar{A}\bar{B})$



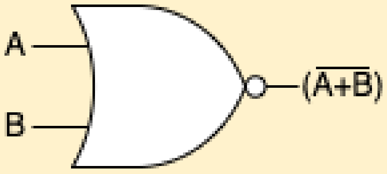
A	B	$A \odot B$
0	1	0
1	0	0
1	1	1
0	0	1

XOR ($\oplus$)=$(\bar{A}B)+(A\bar{B})$



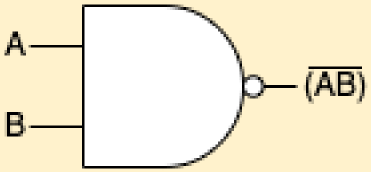
A	B	$A \oplus B$
0	1	1
1	0	1
1	1	0
0	0	0

NOR ($\downarrow$)=$\overline{(A+B)}$



A	B	$A \downarrow B$
0	1	0
1	0	0
1	1	0
0	0	1

NAND ($\uparrow$)=$\overline{(A \cdot B)}$



A	B	$A \uparrow B$
0	1	1
1	0	1
1	1	0
0	0	1

4	← 3	← 2	← 1
OR (+)	AND (·)	NOT ($\bar{A}$)	( )

מבוא לארגון המחשב - סיכום

כלל	OR (+)		AND (·)
איבר יחידה	$A + 0 = A$		$A \cdot 1 = A$
איון	$A + 1 = 1$		$A \cdot 0 = 0$
אידמפוטנט	$A + A = A$		$A \cdot A = A$
משלים	$A + \bar{A} = 1$		$A \cdot \bar{A} = 0$
חילוף	$A + B = B + A$		$A \cdot B = B \cdot A$
קיבוץ	$A + (B + C) = (A + B) + C$		$A \cdot (B \cdot C) = (A \cdot B) \cdot C$
פילוג	$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$		$A + (B \cdot C) = (A + B) \cdot (A + C)$
ספיגה	$A + (A \cdot B) = A$		$A \cdot (A + B) = A$
משפט דה מורגן	$\overline{(A + B)} = \bar{A} \cdot \bar{B}$		$\overline{(A \cdot B)} = \bar{A} + \bar{B}$
ההופכי הנעלם	$A + (\bar{A} \cdot B) = A + B$		$A \cdot (\bar{A} + B) = A \cdot B$
הופכי כפול	$\overline{(\bar{A})} = A$		

מחשבון לאלגברה בוליאנית: https://www.boolean-algebra.com