ארגון המחשב ומערכות הפעלה

אביב תשפ"ד

תרגול 5 – Floating point

זוכרים?

בתרגול הראשון הראינו איך להציג את המיקומים בספרות של מספר בלשהו

$$2^i \quad 2^{i-1} \quad \dots\dots\dots \quad 2^2 \quad 2^1 \quad 2^0 \quad 2^{-1} \quad 2^{-2} \dots\dots \quad 2^{-j}$$
$$b_i \; b_{i-1} \dots\dots\dots b_2 \; b_1 \; b_0 \; . \; b_{-1} \; b_{-2} \dots\dots b_{-j}$$

כדי להעביר מבינארי לדצימלי ← לוקחים את המיקום של הספרה b_i ומכפילים ב- 2^i ← בכה סובמים על בל הספרות

$$\sum_{k=-j}^{i} b_k * 2^k$$

הצגת שברים בצורה בינארית

דצימלי	שבר	ייצוג
0_{10}	0	0.0_2
0.25_{10}		0.010_2
	3/16	0.0011_2
0.203125_{10}		0.001101_2
	13/128	0.00011010_2
0.19921875_{10}		0.00110011_2

לשיטה הזו קוראים fixed point ← אנחנו יודעים מראש איפה הנקודה

ככל שהמספר מהצורה $0.1111\ldots\ldots_2$ יותר ארוך ← ככה הוא יותר קרוב ל-1

הצגת שברים בצורה בינארית

ייצוג	שבר	דצימלי
0.0_2	0	0_{10}
0.010_2	1/4	0.25_{10}
0.0011_2	3/16	0.1875_{10}
0.001101_2	13/64	0.203125_{10}
0.00011010_2	13/128	0.1015625_{10}
0.00110011_2	51/256	0.19921875_{10}

לשיטה הזו קוראים fixed point ← אנחנו יודעים מראש איפה הנקודה

ככל שהמספר מהצורה $0.1111\ldots\ldots_2$ יותר ארוך ← ככה הוא יותר קרוב ל-1

ראינו בהרצאה שיש מגבלות על מספרים מסוימים

נרצה להעביר את המספר 0.625 שבבסיס 10 לבסיס 2

$0.625 * 2 = 1.25$ מוציאים 1

$0.25 * 2 = 0.5$ מוציאים 0

$0.5 * 2 = 1.0$ מוציאים 1

0 עוצרים כשמגיעים ל-0

$\Rightarrow (0.625)_{10} = (0.101)_2$

עכשיו נרצה להעביר את המספר 0.6 בבסיס 10 לבסיס 2

$0.6 * 2 = 1.2$ מוציאים 1

$0.2 * 2 = 0.4$ מוציאים 0

$0.4 * 2 = 0.8$ מוציאים 0

$0.8 * 2 = 1.6$ מוציאים 1

$0.6 * 2 = \cdots$ ...

נשים לב שאף פעם לא נסיים, יש כאן תבנית חוזרת של 1001
עד אינסוף ← $(0.6)_{10} = (0.1001\ 1001\ 1001\ \ldots)_2$

המחשב לא יכול להכיל מספר עם ייצוג אינסופי
ייתן סבומים לא מדויקים של חזקות של 2 עבור מספר נתון של ביטים

ניקח שוב את 0.6 ונציג אותו בבסיס 2 עם 8 סיביות

$$\Rightarrow (0.6)_{10} = (0.1001\ 1001)_2$$

אנחנו יודעים ש- $0.6 + 0.6 = 1.2$

נבדוק אם מתקיים גם אצלנו

$$\begin{array}{r} 0.1001\ 1001 \\ 0.1001\ 1001 \ + \\ \hline 1.0011\ 0010 \end{array}$$

נעביר את התוצאה לבסיס 10: 1.1953125

בעצם יש לנו שגיאה של $1.2 - 1.1953125 = 0.0016875$

זו לא שגיאה גדולה ← אבל במערכות מסוימות הדבר יכול לגרום לתקלות ואסונות

תרגיל 1

מלאו את הטבלה הבאה:

דצימלי	בינארי	שבר
0.375		
	1.0111	23/16
2.8125		45/16
		11/8
5.625		
	11.0001	49/16

תרגיל 1 - פתרון

דצימלי	בינארי	שבר
0.375	0.011	3/8
1.4375	1.0111	23/16
2.8125	10.1101	45/16
1.375	1.011	11/8
5.625	101.101	45/8
3.0625	11.0001	49/16

IEEE

ראינו בהרצאה את סטנדרט IEEE
← ראינו שאם אנחנו מדפיסים ביטים של float או double הביטים מחולקים ל-3 חלקים:

1. S- sign bit - האם המספר חיובי או שלילי- מייצג את ה-MSB
2. M- mantisa - ערך בטווח של [1.0,2.0)- מייצג את השבר (frac)
3. E- exponent - מייצג את חזקת ה-2 שהמספר מוכפל בו (exp)

בעצם- הערך מיוצג ע"י $V = (-1)^S * M * 2^E$

S	exp	frac
Sign bit	E	M

הגדלים של E ו-M שולטים על רמת הדיוק:
למשל: ב-32 סיביות- sign bit=1, exp=8, frac=23
ב-64 סיביות- sign bit=1, exp=11, frac=52

Bias

ה-exp צריך לייצג גם חזקות חיוביות וגם שליליות

←אפשר לייצג עם 2's Complement – מקשה יותר על החישובים

לכן נבצע המרה של הערך לערך unsigned ע"י הטיה (biasing)

בגודל של $2^{|\exp|-1}-1$

למשל: כאשר אנחנו ב-32 סיביות עם |exp|=8 ← $bias = 2^{8-1}-1 = 127$

דוגמה

נניח שאנחנו עם 8 סיביות, חלוקה אפשרית תהיה:

באשר e=0000 ← הערך יהיה שווה ל-0 או מספרים Denormalized

באשר e=1111 ← הערך יהיה שווה לאינסוף או NaN

כמה ערכי exp נשארו לנו? 7 חיוביים, 6 שליליים ו 0.

נחשב את ה-bias: $bias = 2^{|\exp|-1} - 1 = 2^{4-1} - 1 = 8 - 1 = 7$

נוסחת המעבר מערכי E (שיכולים להיות 7 → 6-) לערכי e (שיכולים להיות 1110→0001)

היא $E + bias = e$

חשוב : E מייצג את החזקה המקורית ו e זאת החזקה כפי שהיא מיוצגת ב IEEE (אחרי הוספת bias)

תרגיל 2

ייצגו את המספר 18.25- ב-floating point.

הניחו שנמצאים ב-32 סיביות

תרגיל 2 - פתרון

$18_{10} = 10010_2$

$0.25_{10} = 0.01_2 \qquad \Rightarrow 18.25_{10} = 10010.01_2$

נרצה להציג בחזקות של 2

$10010.01 = 1.001001 * 2^4$

$frac = 001001000 \ldots, E = 4$

עבשיו רוצים לדעת את exp (המספר הוא normal לכן מחשבים הטייה רגילה)

$E + bias = e \Rightarrow bias = 2^{|\exp|-1} - 1 = 2^{8-1} - 1 = 127$

$e = 4 + 127 = 131_{10} = 10000011_2$

לכן התוצאה היא ... $1|10000011|00100100000$

S e frac

תרגיל 3

העבירו את המספר E7 שמיוצג ב-floating point ע"י 8 ביטים לבסיס 10.

הניחו: |exp| = 3 ו |frac| = 4.

תרגיל 3 - פתרון

<u>שלב 1</u>: נמיר את E7 לייצוג בינארי:

$$(E7)_{16}=(11100111)_2$$

<u>שלב 2</u>: נחלק את המספר למקומות המתאימים וניעזר בנתון על הגדלים:

תרגיל 3 - פתרון

שלב 3: כעת נחשב את E

Mantisa=1.0111 (normalized number)

$e = (110)_2 = 6_{10}$

$Bias = 2^{3-1} - 1 = 3$

E = e-bias = 6-3 = 3

$=> 1.0111_2 * 2^3 = 1011.1$

נעביר לבסיס 10

$1011.1_2 = 11.5_{10}$

נשים לב ש-s=1 לכן המספר הוא במינוס

לכן התשובה היא: $E7_{16} = -11.5_{10}$

denormalized numbers

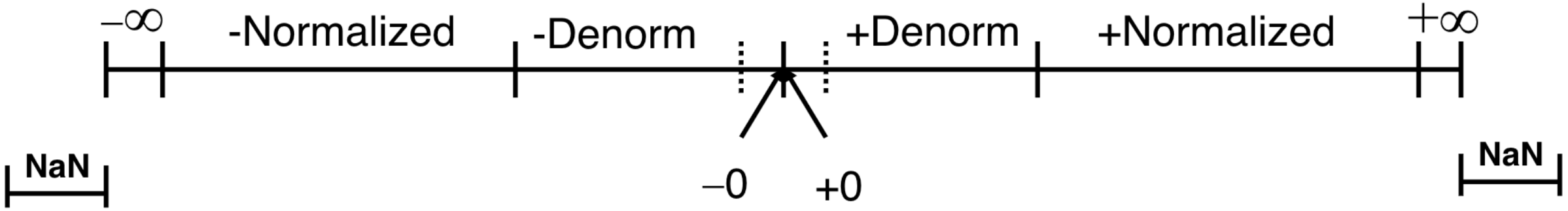
מספרים שמאפשרים דיוק באשר נמצאים מסביב ל-0 (מספרים מאוד קטנים)

←באשר ... $\exp = 000$ ו- $frac > 0$ ← M = 0.frac

מחשבים את הערך E באופן שונה: $E = 1 - bias = 1 - (2^{|\exp|-1} - 1)$

באשר ...frac=000 וגם ...exp = 000 ← מייצגים את 0 או את 0- (תלוי ב Sign bit)

אם נשים את כל המספרים על ציר



תרגיל 4

ייצגו את המספר ה-denormalized החיובי הקטן ביותר שניתן לייצג ב-32 ביטים

תרגיל 4 - פתרון

ראשית, אנו רוצים מספר חיובי לכן s=0.

בנוסף, המספר המבוקש הוא denormalized לכן exp = 00…0.

נחשב את E:

$$\text{Bias} = 2^{|\text{exp}|-1} - 1 = 2^7 - 1 = 127$$

E = 1-bias = 1-127 = -126

כי זה denormalized

נחשב את ה frac:

ב denormalized , frac>0 ואנו רוצים את המספר הקטן ביותר לכן: frac = 00…01

-1,-2,…,-22,-23

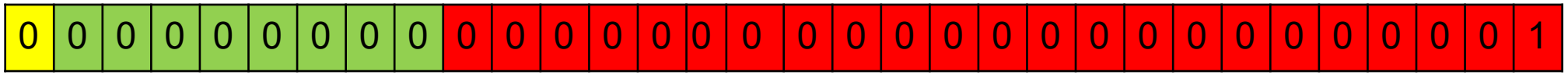
תרגיל 4 - פתרון

אזי: 2^{-23} = 0.00....01=M, -126 = E, S=0.

נחשב את הערך בעזרת הנוסחה:

$$V = (-1)^S \cdot M \cdot 2^E = 1 \cdot 2^{-23} \cdot 2^{-126} = 2^{-149}$$

והייצוג של המספר הוא:



bits	e	E	frac	M	value
0 00 00					
0 00 01					
0 00 10					
0 01 00					
0 01 01					
0 01 10					
0 01 11					
0 10 00					
0 10 01					
0 10 10					
0 10 11					
0 11 00					
0 11 01					
0 11 10					
0 11 11					

bits	e	E	frac	M	value
0 00 00	0	0	00	0.00	0
0 00 01	0	0	01	0.01	0.25
0 00 10	0	0	10	0.10	0.5
0 01 00	1	0	00	1.00	1
0 01 01	1	0	01	1.01	1.25
0 01 10	1	0	10	1.10	1.5
0 01 11	1	0	11	1.11	1.75
0 10 00	2	1	00	1.00	2
0 10 01	2	1	01	1.01	2.5
0 10 10	2	1	10	1.10	3
0 10 11	2	1	11	1.11	3.5
0 11 00	3	2	00	1.00	∞
0 11 01	3	2	01	1.01	Nan
0 11 10	3	2	10	1.10	Nan
0 11 11	3	2	11	1.11	Nan