ארגון המחשב ומערכות הפעלה

אביב תשפ"ד

תרגול 4 – מספרים שלמים

בפרקים הקודמים

- למדנו איך להעביר לבסיסים שונים
- למדנו זיזות
- למדנו איך מייצגים מספרים שלמים

היום, נחזור על איך מיוצגים מספרים שלמים בעזרת וקטור ביטים ואילו פעולות אפשר לעשות עליו.

קידוד שלמים- Unsigned

אם המספר הוא unsigned (0 או חיוביים בלבד)- מקודדים את המספר בבסיס 2.

לשיטה הזו קוראים B2U- Binary to Unsigned

$$B2U(x)_2 = \sum_{i=0}^{w-1} x_i * 2^i$$

דוגמה: $B2U(1010)_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 10$

מהו המספר הכי נמוך שאפשר לייצג?

מהו המספר הכי גבוה שאפשר לייצג?

קידוד שלמים- Unsigned

אם המספר הוא unsigned (0 או חיוביים בלבד)- מקודדים את המספר בבסיס 2.

לשיטה הזו קוראים B2U- Binary to Unsigned

$$B2U(x)_2 = \sum_{i=0}^{w-1} x_i * 2^i$$

דוגמה: $B2U(1010)_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 10$

מהו המספר הכי נמוך שאפשר לייצג? Umin 0 = 00........00

מהו המספר הכי גבוה שאפשר לייצג? Umax $2^w - 1$ = 11.......11

w הוא מספר הסיביות

קידוד שלמים- Signed

X	B2U(X)	B2T(X)
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	$-2^3=-8$
1001	9	$-2^3+1=-7$
1010	10	$-2^3+2=-6$
1011	11	$-2^3+3=-5$
1100	12	$-2^3+4=-4$
1101	13	$-2^3+5=-3$
1110	14	$-2^3+6=-2$
1111	15	$-2^3+7=-1$

אמרנו בהרצאה שאפשר לייצג כל מספר –

- אם הוא חיובי- הביט השמאלי יהיה 0
- אם הוא שלילי- הביט השמאלי יהיה 1

מה הבעיה בייצוג הזה?

יש ייצוג נפרד ל-0 ול -0 ← מבזבזים מקום!

מה האלטרנטיבה? 2’s Complement

קידוד שלמים- Signed- 2's Complement

- זוהי הצורה שבה המחשבים עובדים היום
- כמות המספרים שאפשר לייצג קטנה פי 2 (למה?)

sign bit
(MSB)

$$B2T(x)_2 = -x_{w-1} * 2^{w-1} + \sum_{i=0}^{w-2} x_i * 2^i$$

דוגמה: $B2T(1011)_2 = -1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = -5$

מהו המספר הכי נמוך שאפשר לייצג?

מהו המספר הכי גבוה שאפשר לייצג?

קידוד שלמים- Signed- 2’s Complement

- זוהי הצורה שבה המחשבים עובדים היום
- כמות המספרים שאפשר לייצג קטנה פי 2 (למה?)

sign bit
(MSB)

$$B2T(x)_2 = -x_{w-1} * 2^{w-1} + \sum_{i=0}^{w-2} x_i * 2^i$$

דוגמה: $B2T(1011)_2 = -1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = -5$

מהו המספר הכי נמוך שאפשר לייצג? Tmin -2^{w-1} =10........00

מהו המספר הכי גבוה שאפשר לייצג? $2^{w-1} - 1$ = 01.......11 Tmax

w הוא מספר הסיביות

קשרים חשובים

מה אפשר להגיד על הקשר בין Tmin ו-Tmax? (3 דברים)

1. Tmin = ~Tmax

2. Tmin + Tmax= -1

3. |Tmin|= Tmax+1

מה אפשר להגיד על הקשר בין Tmax ו-Umax?

2Tmax + 1 = Umax

טווחי מספרים – שליליים וחיוביים

מה מבין הבאים נכון?

1. טווחי המספרים שווים
2. טווח המספרים השליליים גדול ב-1 מטווח המספרים החיוביים
3. טווח המספרים השליליים קטן ב-1 מטווח המספרים החיוביים

איך המספר 1- מיוצג?

טווחי מספרים – שליליים וחיוביים

מה מבין הבאים נכון?

1. טווחי המספרים שווים
2. טווח המספרים השליליים גדול ב-1 מטווח המספרים החיוביים
3. טווח המספרים השליליים קטן ב-1 מטווח המספרים החיוביים

משום ש 0 הוא לא חיובי ולא שלילי

איך המספר 1- מיוצג? 1111111...111 (וקטור שכולו אחד)

חיבור

- כללי החיבור נשארים אותו הדבר

נגדיר את x,y להיות מיוצגים לפי 4 סיביות

$$x = 4, y = 1$$

האם $x + y = 5$?

```
x = 0100
            +
y = 0001
---------------
    0101

    == 5
```

חיבור

עכשיו נגדיר את x,y להיות מיוצגים לפי 5 סיביות

$$x = -5, y = -6$$

האם $x + y = -11$?

מעבירים לביטים-

$$\begin{array}{r} x = 11011 \\ y = 11010 \\ \hline \cancel{(1)}\ 10101 \end{array} +$$

האם צדקנו? כן! 10101 = 11-

חיבור

לא תמיד אנחנו צודקים! יש מקרים שבהם הגלישה הזו יוצרת תשובה לא נכונה!

<u>דוגמה 1</u>: נרצה לתאר את 3- ו-4- בעזרת 3 סיביות

-4=100, -3=101

האם -4+(-3)= -7?

$$
\begin{array}{r l}
-3 = 101 & + \\
-4 = 100 & \\
\hline
1 = 001 &
\end{array}
$$

למה? כי 4- הוא הערך המינימלי (Tmin) ב-3 סיביות- כלומר 7- לא בטווח!

חיבור

<u>דוגמה 2</u>: נרצה לתאר את 2 ו-3 בעזרת 3 סיביות

2=010 , 3=011

האם 2+3=5?

$$\begin{array}{r} 2 = 010 \\ + \quad 3 = 011 \\ \hline -3 = 101 \end{array}$$

למה? כי 3 הוא הערך המקסימלי (Tmax) ב-3 סיביות- כלומר 5 לא בטווח!

אז מתי מזהים גלישה?

1. NegOver- אם שלילי + שלילי = חיובי

התוצאה יוצאת יותר קטנה מ- Tmin אבל זורקים את ה-1 ← יוצא חיובי.

2. PosOver- אם חיובי + חיובי = שלילי

התוצאה יוצאת יותר גדולה מ-Tmax אבל זורקים את ה-0 ← יוצא שלילי.

מה אפשר להגיד?

$c_3 \neq c_2 \Rightarrow$ גלישה

$$\begin{array}{rr} & c_3c_2c_1c_0 \\ + & 100 \\ & 101 \\ \hline \end{array}$$

$$\begin{array}{rr} & c_3c_2c_1c_0 \\ + & 010 \\ & 011 \\ \hline \end{array}$$

אז מתי מזהים גלישה?

1. NegOver- אם שלילי + שלילי = חיובי

התוצאה יוצאת יותר קטנה מ-Tmin- אבל זורקים את ה-1 ← יוצא חיובי

2. PosOver- אם חיובי + חיובי = שלילי

התוצאה יוצאת יותר גדולה מ-Tmax- אבל זורקים את ה-0 ← יוצא שלילי

$$\begin{array}{r l} & c_5\ c_4c_3c_2c_1c_0 \\ + & 11011 \\ & 11010 \\ \hline & [1]\ 10101 \end{array}$$

לעומת זאת $c_4 = c_5 \Rightarrow correct$

האם חיבור של חיובי + שלילי יכול לגלוש?

לא כי הכל נשאר בטווח

אמרנו קודם איך מייצגים את המספר 1-

$$-1 = 111\ldots\ldots\ldots111$$

בואו נבדוק שזה באמת נכון

אם 1- באמת מיוצג ע"י וקטור שכולו אחדים ← אם נוסיף לו 1 נקבל 0

$$
\begin{array}{r}
+\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 \\
0\ 0\ 0\ 0\ 0\ 0\ 0\ 1 \\
\hline
[1]\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0
\end{array}
$$

האם נכון? כן

האם יש בעיה עם הגלישה? לא כי זה חיבור של + עם –

-x

נתון לנו x, אנחנו רוצים לדעת מהו x–

האם מספיק לנו לעשות רק NOT?

למה? כי החיבור ביניהם ייתן לנו 1- (ולא 0 כמו שהיינו מצפים)

למשל: x = 4

$$\begin{array}{r} x = 0100 \\ \sim x = 1011 \\ \hline 1111 \end{array} +$$

-x

קיבלנו $x + \sim x = -1$

אבל אנחנו רוצים לקבל 0- מה עושים?

לכן

$$-x = \sim x + 1$$

פעולה לא חוקית

למדנו על זיזות (shifting)- האם מותר לנו להזיז לאן שבא לנו?

מהי פעולה לא חוקית?
פעולה שהתוצאה שלה לא מוגדרת ויכולה להשתנות באופן לא צפוי

למשל: אנחנו מיוצגים לפי 32 סיביות והמשתנה שלנו x הוא מסוג int או unsigned

האם x>>31 זאת פעולה חוקית? כן

האם x>>32 זאת פעולה חוקית? לא

פעולה לא חוקית

x=001
x<<1 => x=010
x<<2 => x=100
x<<3 => x=000

x=010
x>>1 => x=001
x>>2 => x=000

x=100
x>>1 => x=110
x>>2 => x=111
x>>3 => x=111

מה הלו"ז עם unsigned?

דיברנו קצת על unsigned- מייצג רק 0 או חיוביים

אבל נשתמש בו רק כשלא נצטרך שליליים?

למה?

- יש קומפיילרים שיתנו קוד פחות יעיל כשמשתמשים עם unsigned
- קל מאוד לעשות טעויות (ראינו בהרצאה)

מה הלו"ז עם unsigned?

אז מתי כן נשתמש?

- כשרוצים טווח יותר גדול למספרים החיוביים
- לפעמים משתמשים ב-unsigned בקבוצת דגלים
- כשרוצים להשוות תוצאות של פונקציות שמחזירות unsigned

תרגיל

נניח מחשב שמבוסס על 32 סיביות עם שיטת 2’s Complement

נגדיר:

$$2^{31} = 2147483648$$
$$2^{31} - 1 = 2147483647$$

נזכיר:

- אם אחד הצדדים של הביטוי הוא unsigned <= ההשוואה היא לפי unsigned
- אם שני הצדדים הם signed <= ההשוואה היא לפי signed

ביטוי	סוג השוואה	True / False
-2147483647-1 == 2147483648U		
-2147483647 - 1 < -2147483647		
U(-2147483647-1) < -2147483647		
U(-2147483647-1) < 2147483647		

פתרון

נניח מחשב שמבוסס על 32 סיביות עם שיטת 2’s Complement

נגדיר:

$$2^{31} = 2147483648$$
$$2^{31} - 1 = 2147483647$$

נזכיר:

- אם אחד הצדדים של הביטוי הוא unsigned <= ההשוואה היא לפי unsigned
- אם שני הצדדים הם signed <= ההשוואה היא לפי signed

ביטוי	סוג השוואה	True / False
-2147483647-1 == 2147483648U	Unsigned	True
-2147483647 - 1 < -2147483647	Signed	True
U(-2147483647-1) < -2147483647	Unsigned	True
U(-2147483647-1) < 2147483647	Unsigned	False