ייצוג מספרים בנקודה קבועה

אלון עילם

ייצוג מספרים טבעיים:

במחשב ספרתי הרמה הבסיסית של המידע מורכבת מסיביות בעלות ערכים לוגיים של '0' ו-'1'. על מנת לייצג מספרים בתחום רחב, מיקום הסיבית באוגר (Register) מגדיר את החזקה של 2 בה יש לכפול אותה. הדבר דומה לפעולה שאנו שאנו עושים ביום יום עבור ייצוג עשרוני, כאשר המספר 265 משמעותו $2x10^2 + 6x10^1 + 5x10^0$.

ייצוג זה, שנראה לנו טבעי, לא היה קיים בעת העתיקה. בגימטריה, לדוגמא, לכל אות יש ערך קבוע שאינו נגזר ממיקומה. תשע"ב היא בגימטריה 772, ואין זה משנה אם נאמר שאנו בשנת בעת"ש. גם בשיטת ייצוג המספרים הרומית לכל תו היה ערך מספרי קבוע, אך שם ניתנה משמעות מסויימת למיקומו. לדוגמא IV פירושו 4 ו-VI פירושו 6.

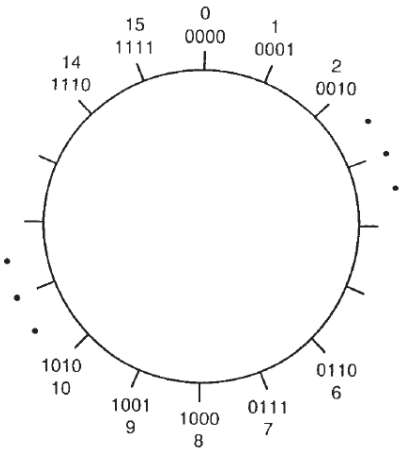
במחשב ספרתי, מיקום הסיבית באוגר בגודל N מייצג את חזקת 2 בה יש לכפול אותה. אינדקס מיקום הסיבית הוא בין 0 ל-N-1, ולפיכך הערך המיוצג ע"י הסיבית במיקום הגבוה ביותר באוגר הוא 2^{N-1}. את המספר הטבעי הגדול ביותר הניתן לייצוג ע"י N סיביות נקבל כאשר כל הסיביות הן '1', וערכו הוא 2^N-1. מכאן נובע כי ככל שהאוגר גדול יותר, ניתן לייצג מספרים בתחום רחב יותר וכל סיבית נוספת מגדילה פי 2 את תחום הייצוג.

לאוגר בעל N סיביות יהיה על כן תחום דינמי:

$$DynamicRange[dB] = 20\log_{10}(2^N) = 20N\log_{10}(2) \approx 6N$$

פעולת החיבור

בפעולת חיבור בינארית מחברים כל סיבית באוגר אחד עם הסיבית המקבילה באוגר השני. במידה ושתי הסיביות הן '1', נוצר carry, שהוא סיבית בערך '1' הנוספת לסכום הסיביות שבמיקום הבא. ה-carry הנוצר מחיבור הסיביות הגבוהות ביותר גולש "החוצה" ולכן פעולת חיבור בין שני אוגרים באורך N נותנת תוצאה שהיא $modulo(2^N)$. ניתן לתאר זאת אם נסתכל על ייצוג מספרים טבעיים באופן מעגלי כלהלן:



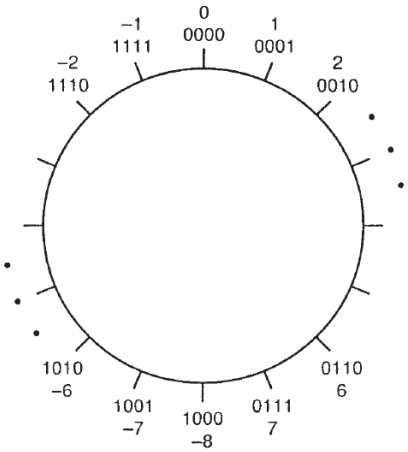
איור 9. ייצוג מספרים טבעיים באוגר בעל 4 סיביות

יצוג מספרים טבעיים באוגר בעל 4 סיביות מודגם באיור 9, ותחום המספרים אותו ניתן לייצג הוא 0-15. כדי לחבר את המספרים 9 ו-5, נעמוד על המספר 9 ונתקדם 5 צעדים. נגיע למספר 14 הניתן לייצוג ב-4 סיביות. לעומת זאת אם נחבר למספר 9 את המספר 9, הרי שכאשר נתקדם 9 צעדים נגיע ל- 18modulo(16)=2.

<u>ייצוג מספרים שלמים:</u>

בעולם העתיק, אימפריות קמו ונפלו בזמן שנעשה שימוש בייצוג מספרים טבעיים בלבד. לצערנו לא ניתן להסתפק בכך כיום, והמינוס בבנק, לדוגמא, הינו תנאי קיומי אצל רבים. בשיטה העשרונית סימן המינוס (-) מייצג ערך שלילי, והנקודה העשרונית משמשת לייצוג שבר.במחשב ספרתי, על מנת לייצג מספרים שלמים בעלי ערך שלילי מקובלת שיטת ה"משלים ל-2". להדגמת שיטה זו נחלק את המעגל שראינו קודם לשני תחומים. מחציתו תייצג ערכים חיוביים ומחציתו ערכים שליליים. היתרון בשיטת ה"משלים ל-2" הוא שפעולות החיבור והכפל מבוצעות באותו אופן עבור מספרים חיוביים ועבור מספרים שליליים.

האיור הבא מדגים ייצוג בשיטת ה"משלים ל-2" עם אוגר בעל 4 סיביות:



איור 10. ייצוג מספרים שלמים בשיטת ה"משלים ל-2" עם אוגר בעל 4 סיביות

נשים לב שתחום המספרים מחולק בהתאם לערכה של הסיבית הגבוהה ביותר, שנקראת "סיבית הסימן". אם ערכה של סיבית הסימן הוא '0' המספר המיוצג הוא חיובי או אפס, ואם ערכה של סיבית הסימן הוא '1' המספר המיוצג הוא שלילי. כדי לחשב את ערכו הדצימלי של המספר המיוצג נשתמש בטור החזקות הבא: $I(B) = -b_{n-1} \times 2^{n-1} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$

כאשר b_i מייצג את ערך הסיבית במיקום i, ו-I(B) הוא הערך הדצימלי של המספר.

<u>דוגמא</u>: באוגר באורך 4, מה מייצג הצירוף 1110?

$$1110 = -1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = -8 + 4 + 2 + 0 = -2$$

<u>דוגמא</u>: נדגים את ישימות הייצוג בשיטת ה"משלים ל-2" לביצוע פעולת חיבור של מספר חיובי ומספר שלילי:

עבור הפעולה 6-2=6+(-2)=4 נבצע חיבור modulo(16) של ייצוג המספר 6 עם הייצוג של 2-, כלומר <u>נתקדם</u> 1110 צעדים (14 צעדים) מהמספר 6 על פני המעגל ונקבל: (6+14)mod(16)=20mod(16)=4.

חיבור בינארי ייתן את אותה תוצאה:

$$\begin{array}{r} 0110 \\ 1110 \\ \hline 10100 \end{array}$$

מאחר והאוגר בדוגמא שלנו הוא בעל 4 סיביות, הסיבית החמישית "נזרקת" ואנו נשארים עם הערך 0100 המיצג את המספר 4.

מהדוגמא ניתן להבין מדוע משתמשים בשיטת ה"משלים ל-2" לעומת שימוש בסיבית סימן לציון מספר שלילי, עם הערך המוחלט של המספר ביתר הסיביות. פעולת החיבור ממומשת עבור ייצוג ה"משלים ל-2" בעזרת מעגל לוגי פשוט, ועבור שיטות ייצוג אחרות פשטות זו אינה קיימת.

<u>ייצוג של שברים בשיטת "נקודה קבועה" (Fixed Point):</u>

כדי לייצג שברים, נרצה לתת משמעות אחרת למיקום הסיביות באוגר. הפולינום הבא לחישוב ערכו הדצימלי של תוכן אוגר, מאפשר לייצג שברים:

$$F(B) = -b_{N-1} x 2^0 + b_{N-2} x 2^{-1} + \ldots + b_1 x 2^{-(N-2)} + b_0 x 2^{-(N-1)}$$

<u>שאלה:</u> מה הוא הערך המקסימלי ומה הערך המינימלי אותם ניתן לייצג בעזרת הפולינום לעיל באוגר בעל 4 סיביות?

<u>פתרון:</u> הערך המקסימלי יתקבל כאשר סיבית הסימן היא '0', וכל יתר הסיביות '1':

$-0\ x2^{-0} +1x2^{-1} +1x2^{-2} +1x2^{-3} =0.875$

הערך המינימלי יתקבל כאשר סיבית הסימן היא '1' ויתר הסיביות '0':

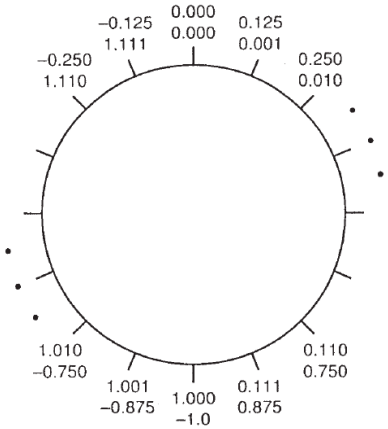
$-1\ x2^{-0} +0x2^{-1} +0x2^{-2} +0x2^{-3} =-1$

<u>שאלה:</u> מה הוא השבר החיובי הקטן ביותר אותו ניתן לייצג בעזרת הפולינום לעיל באוגר בעל 4 סיביות?

<u>פתרון:</u> השבר החיובי הקטן ביותר יתקבל כאשר כל הסיביות הן '0' למעט הראשונה:

$-0\ x2^{-0} +0x2^{-1} +0x2^{-2} +1x2^{-3} =0.125$

על פני המעגל ייצוג השברים נראה כלהלן:



איור 11. ייצוג שברים באוגר בעל 4 סיביות

הייצוג בשיטה זו נקרא ייצוג בנקודה קבועה (Fixed Point), כדי להבחין בינו ובין ייצוג בנקודה צפה (Floating pint) שיידון בהמשך.

<u>הגדרה:</u> בייצוג בנקודה קבועה, נסמן Q.k, כאשר הערך k הוא מספר הסיביות באוגר המייצגות את השבר.

<u>דוגמא:</u> עבור האוגר בן 4 הסיביות בדוגמא שלעיל, הייצוג היה Q.3.

לסיום הדיון בייצוג שברים בנקודה קבועה, מספר הערות:

א. אף כי תחום הערכים השתנה, במעבר מייצוג של מספרים שלמים לייצוג שברים התחום הדינמי לא השתנה עבור אוגר באורך N והוא נשאר בקירוב 6N .

ב. ניתן להגדיר את מיקום הנקודה הבינארית כך שתחום הערכים המיוצג (בערך מוחלט) יהיה גדול מ-1. נעשה זאת ע״י שינוי הגדרת הפולינום, כך שניתן משקל חדש לכל סיבית באוגר – כולל סיבית הסימן!

<u>דוגמא</u>: עבור אוגר באורך 4 נגדיר את הפולינום הבא:

$F(B)= -b_{N-1}x2^{1}+b_{N-2}x2^{0}+\ldots+b_{1}x2^{-(N-3)}+b_{0}x2^{-(N-2)}$

<u>שאלה:</u> מה הוא הערך המקסימלי ומה הערך המינימלי אותם ניתן לייצג, ומה הוא k בסימון Q.k עבור מקרה זה?

<u>פתרון:</u> הערך המקסימלי הוא כאשר סיבית הסימן היא '0', וכל יתר הסיביות '1':

$-0\ x2^{1} +1x2^{0} +1x2^{-1} +1x2^{-2} =1.75$

הערך המינימלי הוא כאשר סיבית הסימן היא '1' ויתר הסיביות '0':

$-1\ x2^{1} +0x2^{0} +0x2^{-1} +0x2^{-2} =-2.00$

מאחר ויש לנו 2 סיביות לייצוג השבר, זה הוא ייצוג Q.2

שאלה: מה היתרון ומה החיסרון במעבר מ-Q.3 ל-Q.2?
תשובה: הייתרון הוא היכולת לייצג ערכים בתחום גדול יותר, והחיסרון הוא פגיעה ברזולוציה בה נייצג שברים: 0.25 ב-Q.2 לעומת 0.125 ב-Q.3

פעולת הכפל:

כאשר נבצע כפל בין 2 אוגרים באורך N סיביות כל אחד, תוצאת הכפל תהיה באורך 2N. במידה ונרצה לשמור אותה באוגר או בזכרון בעל N סיביות, נאלץ לשמור אותה בשני תאי זכרון או בשני אוגרים. מאחר ובמימוש אלגוריתמים מבצעים סדרה של פעולות חישוב, כאשר התוצאות מכל פעולת חישוב הן הנתון עבור פעולת החישוב הבאה (לדוגמא - מסנן IIR), אין זה מעשי לשמור 2N סיביות, כי תוצאת הכפל הבאה תהיה באורך 4N וכן הלאה! נאלץ לכן לשמור ולהשתמש רק ב- N סיביות מתוך תוצאת הכפל.

הפתרון הפשוט הוא לקחת את N הסיביות העליונות מתוך 2N הסיביות של תוצאת הכפל, וכך נבטיח שלא נאבד סיביות בעלות משמעות. החיסרון של גישה זו הוא אבדן מידע בעל ערך, כאשר אחד הכופלים או שניהם קטנים.

דוגמא: מה תהיה תוצאת הכפל של 25 ו-12 המיוצגים באוגרים בני 16 סיביות (N=16), מידה ונשמור רק את N הסיביות העליונות של תוצאת הכפל?

פתרון: המספר 25 הוא 0x0019, (0x מסמל רישום הקסדצימלי) והמספר 12 הוא 0x000C. תוצאת הכפל תהיה 0x0000012C, ובמידה ונקח את 16 הסיביות הגבוהות נקבל תוצאה אפס. מובן כי במקרה זה היה נכון לקחת את 16 הסיביות הראשונות של התוצאה, כי כל יתר הסיביות הן שכפול של סיבית הסימן.

כופלים בעלי ערך נמוך אינם נדירים, ונפגוש אותם כמעט בכל מימוש של מסנן ספרתי.

כדי לשמר את N הסיביות בעלות המשמעות בעת ביצוע פעולות כפל ב-Fixed Point נפעל באופן הבא:

א. כאשר מבוצע רצף של פעולות כפל וחיבור (כגון במסנן FIR) נשמור את תוצאות הביניים באוגר באורך 2N. לשם כך כל מעבדי האותות בנויים באופן שתוצאת הכפל מתווספת לסכום תוצאות הכפל הקודמות, אלא אם איפסנו את אוגר תוצאת המכפלה לפני כן.
ב. עם סיום רצף פעולות הכפל והחיבור, נבצע הזזה (פעולת <<) לאוגר תוצאת הכפל לשמירת מירב הסיביות בעלות המשמעות ב- N הסיביות הנמוכות שלו ונשמור בזכרון או באוגר N סיביות להמשך העבודה. מספר הסיביות שנזיז יהיה בהתאם לידע מוקדם שיש לנו באשר לתחום הדינמי של הכופלים ומכאן של התוצאה.

רוויה כתוצאה מביצוע רצף של פעולות כפל וחיבור:

בעת ביצוע רצף של פעולות כפל וחיבור במעבד אותות, במידה והתוצאה חורגת מגודל אוגר התוצאה הוא ייכנס לרוויה – כלומר יכיל את הערך החיובי או השלילי הגדול ביותר האפשרי. רוויה היא פתרון טוב בהרבה מזריקת סיביות בעלות משמעות, ועם זאת נשאף להמנע ממנה ככל האפשר. לכן נוסיף עוד כלל עליו נקפיד לפני ביצוע רצף פעולות כפל וחיבור:

ג. בהסתמך על ידע מוקדם נבצע Scaling של הכופלים לפני ביצוע פעולות הכפל והחיבור, כדי שרצף של פעולות כפל וחיבור לא יכניס את אוגר התוצאה לרוויה.

פעולת ה- Scaling מבוצעת על ידי בחירה באיזה Q עובדים (כאשר מדובר בייצוג מקדמים של מסנן, למשל) ועל ידי ביצוע הזזה ימינה או שמאלה של הכופלים.

Guard Bits באוגר תוצאת המכפלה

כדי לאפשר ביצוע סדרות ארוכות של כפל וחיבור, במעבדי אותות נראה בדרך כלל יותר מ- 2N סיביות באוגר תוצאת המכפלה. סיביות אלה נקראות Guard Bits. הן מאפשרות ביצוע יותר פעולות סיכום לפני כניסה לרוויה.

מכפלה של מספרים בייצוג נקודה קבועה:

כאשר נכפול מספרים בייצוג נקודה קבועה ו"משלים ל-2", הסיבית N-1 בכל אוגר היא סיבית סימן. תוצאת המכפלה תכיל, לכן, 2 סיביות סימן זהות וניתן לבצע הזזה אחת שמאלה של התוצאה ללא חשש מאבדן מידע.

דוגמא: עבור אוגרים באורך 4, מה תהיה התוצאה של הכפלת 0.5- ב- 0.75 המיוצגים ב- Q.3?

פתרון:

$$
\begin{array}{rl}
-0.50 = & 1.100 \\
 & \text{S FFF} \\
\times 0.75 = & 0.110 \\
 & \text{S FFF} \\
\hline
 & 0000000 \\
 & 111100 \\
 & 11100 \\
 & 0000 \\
\hline
 & 11.101000 \\
 & \text{SS FFFFFF}
\end{array}
$$

$$= -2^1 + 2^0 + 2^{-1} + 2^{-3} = -0.375$$

הערות:

א. סיביות הסימן והשבר של שני הכופלים והתוצאה מסומנים ב-S ו-F בהתאמה.

ב. ביצענו כפל ארוך. הסיביות עם קו מתחתן בתוצאות הביניים הן ה- sign extension של תוצאת הביניים.

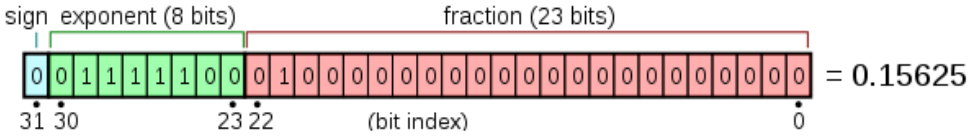
ג. מכפלה של 2 אוגרים בייצוג Q.3 תיתן תוצאה בייצוג Q.6

ד. בשל שכפול סיבית הסימן, בחישוב הפולינום הנותן את הערך הדצימלי לוקחים רק סיבית סימן אחת.

ייצוג מספרים בנקודה צפה (Floating Point)

במידה ואין מגבלה של סיבוכיות החומרה וצריכת הספק, נפוץ השימוש בייצוג המאפשר תחום דינמי גדול ככל האפשר. הוא נקרא "נקודה צפה", וייצוג המספרים הוא לוגריתמי: חלק מהאוגר מכיל את המנטיסה, וחלק מהאוגר מכיל את האקספוננט שהוא חזקה של 2. בנוסף, סיבית סימן משמשת לייצוג מספרים שליליים.

האיור הבא מתאר ייצוג בנקודה צפה של המספר 0.15625 באוגר באורך N=32, בהתאם לתקן IEEE 754-2008:



הפולינום באמצעותו נחשב את הערך הדצימלי של תוכן האוגר הוא:

$$value = (-1)^{\text{sign}}(1 + \sum_{i=1}^{23} b_{-i}2^{-i}) \times 2^{(e-127)}$$

כאשר הערך של sign הוא סיבית ה-MSB, הערך של e הוא 8 סיביות האקספוננט (exponent) וערכי b_i הם סיביות ה-fraction משמאל לימין (i=1 הוא סיבית 22). למעשה, ה-fraction הוא ייצוג בנקודה קבועה של Q.22 באוגר בן 23 סיביות.

ובדוגמא שלנו:

$$sign = 0$$
$$1 + \sum_{i=1}^{23} b_{-i}2^{-i} = 1 + 2^{-2} = 1.25$$
$$2^{(e-127)} = 2^{124-127} = 2^{-3}$$

ולכן:

$$value = 1.25 \times 2^{-3} = 0.15625$$

ייצוג עם 32 סיביות הוא זה שאנו מכירים כ-float בשפת C. קיים ייצוג עם 64 סיביות, הידוע כ-double.

ייצוג בנקודה צפה מגדיל באופן משמעותי את התחום הדינמי אותו ניתן לייצג, אך יש לו גם גם מחיר – המורכבות הגדולה של הפעולות החשבוניות, יחסית לייצוג בנקודה קבועה. על מנת לבצע פעולת חיבור, יש להביא תחילה את שני המחוברים לאותו אקספוננט, ולאחר מכן לחבר את המנטיסות. לביצוע פעולת כפל יש לחבר את האקספוננטים, לכפול את המנטיסות, ולאחר מכן להחזיר את תוצאת מכפלת המנטיסות לגודל המתאים תוך עדכון ערך האקספוננט. בנוסף יש להתייחס לסיבית הסימן. על מנת ליישם את הללו בחומרה, המעבדים המודרניים כוללים בתוכם יחידת FPU (Floating Point Unit) אשר אחראית ליישום בחומרה של פעולות כגון חיבור, חיסור, כפל וחילוק בין משתנים בייצוג נקודה צפה. ברור כי מבחינת שטח הסיליקון הנדרש ל-FPU הינו גדול משמעותית מה-ALU (Arithemtical Logical Unit כפי שזוכרה לכם מקורס תכן לוגי [046262]) ולכן תצרוך הספק רב יותר בנוסף לזמן חישוב ארוך יותר.

מחיר זה הוא זניח כאשר מדובר ביישומים בהם אין חשיבות לצריכת ההספק או עלות הרכיבים, אולם הוא הופך לקריטי כאשר מדובר ביישומים אשר צריכים לפעול עם צריכת הספק נמוכה ו/או בעלות תחרותית. לכן, במחשב אישי עבודה בנקודה צפה היא הסטנדרט זה עשרות שנים, ואילו במכשירים ניידים מרבית היישומים הם בנקודה קבועה. עקב תפוצתם

במכשירים ניידים, מעבדי DSP מסוג נקודה קבועה פופולריים הרבה יותר ממעבדי DSP מסוג נקודה צפה.

<u>מעבר מייצוג בנקודה צפה לייצוג בנקודה קבועה</u>

בעת פיתוח אלגוריתמים באמצעות כלים כגון Matlab או סימוציה בשפת C, לא ניתן את הדעת בדרך כלל לתחום הערכים שהמשתנים מקבלים, והפיתוח ייעשה עם משתנים המוגדרים כ-float או double.

לאחר השלמת פיתוח האלגוריתם, במידה ומדובר במימושו על גבי מעבד העובד בנקודה קבועה, נעביר את כל המשתנים ל-short. פעולה זו דורשת תשומת לב רבה, כדי לבצע את העיבוד בתחום הדינמי הנתון.