Невенка Спалевић

ПРЕВОЂЕЊЕ БРОЈЕВА

Да људи који случајем уместо 10 имају 8 прстију на рукама, за запис бројева и рачунање вероватно би се користио октални бројни систем и прича о превођење у бинарни бројни систем који се користи у рачунарима била би тривијална. Али пошто није тако, развијени су многи алгоритми за превођење бројева из нашег „анатомски заснованог“ декадног бројног система у „природан рачунарски“ бинарни бројни систем и обрнуто. Како је ова трансформација једна од машински најзависнијих операција, инжењери се стално баве различитим начинима за хардверску реализацију ове функције. Управо овим машински зависним алгоритмима превођења посвећен је следећи текст. Но пре него што пређемо на ствар,

Мало историје

Интерес за уређаје који врше аритметичке операције, специјално множење, нагнао је многе истраживаче да се баве бинарним бројним системом још пре појаве дигиталних електронских рачунара. Наиме, бинарни бројни систем има низ особина које га чине најпогоднијим за ове сврхе: у њему се лако и брзо извршавају аритметичке и логичке операције, а бинарне цифре се лако представљају помоћу електричних сигнала и једноставно меморишу.

Тридесетих година нашег века у Француској је начињено више рачунских машина заснованих на бинарном бројном систему. Прва рачунска кола од електронских лампи пројектовао је Џон Атанасов (John Atanasoff) 1937. године, а прве релејне мреже исте године независно Џорџ Штибиц (George Stibitz). Обојица су у својим пројектима користили бинарни бројни систем. Штибиц је ускоро разрадио и бинарни код „вишак 3“ за декадне цифре. У исто време у Немачкој Конрад Цузе (Konrad Zuse) сачинио је механички рачунар заснован на бинарном представљању бројева са покретним зарезом, а 1941. заменио механичку логичку мрежу релејном.

У првим брзим рачунарима у Америци почетком четрдесетих година коришћен је декадни бројни систем, али после значајног рада Беркса, Голдштајна и фон Нојмана о пројекту првог рачунара са унутрашњим програмом (1946. године) у коме су подробно били изложени разлози за прекид са декадним и прелазак на бинарни бројни систем, бинарни рачунари постају широко распрострањени.

Основни алгоритми за превођење бројева

Превођење бројева записаних у једном бројном систему у бројни систем друге основе решава се у основи на два начина: коришћењем аритметике изворног или коришћењем аритметике одредишног бројног система. Други начин је посебно згодан када треба преводити број у декадни бројни систем, јер тада користимо аритметичка правила на која смо навикли.

Алгоритам PREVU

за превођење броја X записаног у позиционом бројном систему основе $N1$ у бројни систем основе $N2$ по правилима одредишне ($N2$) аритметике са задатом тачношћу e.

1) Број X записати у развијеном облику, при чему цифре, основу и степене основе треба записати онако како се пишу у одредишном бројном систему.

2) Извршити назначене операције по правилима $N2$ аритметике, при чему се негативни степени основе који се не могу изразити апсолутно тачно рачунају са задатом тачношћу

Добијени број представља запис броја X у бројном систему основе $N2$.

Пример 1. $X = 344.201$, $N1 = 5$, $N2 = 10$

$$\begin{aligned} 344.201_5 &= 3*5^2+4*5^1+4*5^0+2*5^{-1}+0*5^{-2}+1*5^{-3} \\ &= 75+20+4+0.4+0.008 \\ &= 99.408_{10} \end{aligned}$$

Пример 2. $X = 212.102$, $N1 = 3$, $N2 = 10$, $e = 10^{-4}$

$$\begin{aligned} 212.102_3 &= 2*3^2+1*3^1+2*3^0+1*3^{-1}+0*3^{-2}+2*3^{-3} \\ &= 18+3+2+0.3333+0.0740 \\ &= 23.4073_{10} \end{aligned}$$

Пример 3. $X = 767.1$, $N1 = 8$, $N2 = 6$

$$\begin{aligned} 767.1_8 &= 11*12^2+10*12^1+11*12^0+1*12^{-1} \\ &= 2024+120+11+0.043 \\ &= 2155.043_6 \end{aligned}$$

Алгоритам PREVIZ

за превођење броја X записаног у позиционом бројном систему основе $N1$ у бројни систем основе $N2$ по правилима изворне ($N1$) аритметике са задатом тачношћу e.

1) Раздвоји се цео (CX) од разломљеног (RX) дела броја.

2) Цифре превода целог дела CX у основу $N2$: $C_M \ldots C_2C_1C_0$ добијамо на следећи начин:

$$\begin{aligned} C_0 &= CX \bmod N2 \\ C_1 &= (CX \operatorname{div} N2) \bmod N2 \\ C_2 &= ((CX \operatorname{div} N2) \operatorname{div} N2) \bmod N2 \end{aligned}$$

итд. све док не добијемо:

$$(\dots((CX \text{ div } N2) \text{ div } N2)\dots \text{ div } N2) = 0$$

3) Цифре превода разломљеног дела RX у основу $N2$: $0.R_{-1}R_{-2}\dots R_{-k}$ са задатом тачношћу e добијамо на следећи начин:

$$\begin{aligned} R_{-1} &= [RX * N2] \\ R_{-2} &= [\{RX * N2\} * N2] \\ R_{-3} &= [\{\{RX * N2\} * N2\} * N2] \end{aligned}$$

где се ознака $[x]$ користи за цео део реалног броја x (што се у PASCAL-у може израчунати коришћењем уграђене функције `int(x)`), док се ознака $\{x\}$ користи за разломљени део реалног броја x (у PASCAL-у се добија коришћењем уграђене функције `frac(x)`).

Збир бројева $C_M\dots C_2C_1C_0$ и $0.R_{-1}R_{-2}\dots R_{-k}$ представља запис броја X у систему основе $N2$.

Пример 4. $X = 99.408$, $N1 = 10$, $N2 = 5 \implies CX = 99$, $RX = 0.408$

$$\begin{aligned} C_0 &= 99 \bmod 5 = 4 \qquad & R_{-1} &= [0.408 * 5] = 2 \\ C_1 &= 19 \bmod 5 = 4 & R_{-2} &= [0.04 * 5] = 0 \\ C_2 &= 3 \bmod 5 = 3 & R_{-3} &= [0.2 * 5] = 1 \end{aligned}$$

$99.408_{10} = 344.201_5$.

Пример 5. $X = 23.4073$, $N1 = 10$, $N2 = 3$, $e = 3^{-3}$, $CX = 23$, $RX = 0.4073$

$$\begin{aligned} C_0 &= 23 \bmod 3 = 2 \qquad & R_{-1} &= [0.4073 * 3] = 1 \\ C_1 &= 7 \bmod 3 = 1 & R_{-2} &= [0.2219 * 3] = 0 \\ C_2 &= 2 \bmod 3 = 2 & R_{-3} &= [0.6657 * 3] = 1 \\ & & R_{-4} &= [0.9971 * 3] = 2, \end{aligned}$$

дакле, после заокруживања $R_{-3} = 2$, $23.44073_{10} \approx 212.102_3$.

Пример 6. $X = 2155.043$, $N1 = 6$, $N2 = 8 \implies CX = 2155$, $RX = 0.043$

$$\begin{aligned} C_0 &= 2155 \bmod 12 = 11 \qquad & R_{-1} &= [0.043 * 12] = 1 \\ C_1 &= 142 \bmod 12 = 6 \\ C_2 &= 11 \bmod 12 = 11 \end{aligned}$$

$2155.043_6 = 767.1_8$.

Изложени општи алгоритми могу се применити и за реализацију веома значајног задатка за бинарне рачунаре — превођења улазних података из декадног у бинарни и излазних података из бинарног у декадни бројни систем. Њихова реализација у било ком вишем програмском језику је једноставна, али треба имати на уму да се ти програми могу извршити тек пошто буду преведени на машински језик. Како и алгоритам PREVU и алгоритам PREVIZ користе операције множења и дељења које многи процесори немају у свом репертоару наредби, произлази да се ови задаци које рачунар мора да извршава при сваком уношењу и издавању бројних података понуђеним алгоритмима не решавају на најбољи начин. Да будемо прецизнији, осмобитни микропроцесори по правилу нису имали наредбе за множење и дељење у свом репертоару, а код шеснаестобитних оне су по правилу реализоване микропрограмски и њихово извођење је релативно дуго трајало. Без обзира што су данас на располагању моћни микропроцесори по приступачним ценама који множење и дељење реализују у једном такту, за многе примене (процесну контролу нпр.) и даље се користе јефтини осмобитни процесори, јер коришћење сложенијих и скупљих уређаја у те сврхе напросто нема смисла. Значи, од интереса су алгоритми који коришћењем једино операција сабирања и померања, које може да реализује сваки процесор, врше захтевана превођења. Како су улазни и излазни подаци записани обично у ASCII коду из кога се за декадне цифре врло лако може добити NBCD код (Природни BCD код — код са тежинама "8421", у даљем тексту BCD), описаћемо неке алгоритме који преводе бинарно кодиране декадне бројеве у бинарни бројни систем и обратно.

Алгоритам BCDBIN1

који врши конверзију BCD бројева у праве бинарне бројеве *коришћењем сабирања.* Према том алгоритму се прво свакој бинарној цифри BCD броја придружи одговарајући тежински фактор. Идући са десна улево, тежински фактори су: 1, 2, 4, 8, 10, 20, 40, 80, 100 итд. Затим се изврши сабирање бинарних еквивалената свих тежинских фактора који одговарају битовима у BCD броју чија је вредност јединица. У ову сврху је згодно користити претходно формирану табелу бинарних еквивалената тежинских фактора из које се види да је нпр. $80_{10} = 101000_2$, $100_{10} = 11001000_2$, $200_{10} = 11001000_2$ итд.

Алгоритам BCDBIN2

који врши конверзију BCD бројева у праве бинарне бројеве *коришћењем померања са одузимањем.* Код овог алгоритма BCD број се подели у групе од по 4 бита почевши од најмлађе позиције. Изврши се померање за једно место удесно задржавајући границе декада. Цифра BCD броја која излази ван границе постаје најмлађа цифра бинарног превода. Затим се испитује да ли су нови бројеви у границама декада $\geqslant 8$, и ако јесу од њих се одузима 3. Померање и испитивање се понавља све док се удесно не помере сви битови BCD броја. Цифре које излазе на десну страну су цифре бинарног еквивалента BCD броја. За илустрацију описаног алгоритма извршимо конверзију декадног броја 343 у бинарни еквивалент.

Пример 7.

343_{10}	0011|0100|0011|	
	0001|1010|0001|1	померај
	0001|0111|0001|1	одузимање 3
	|1011|1000|11	померај
	|1000|0101|11	одузимање 3 (из две декаде)

$$\begin{array}{rl}
|0100|0010|111 & \text{померај} \\
|0010|0001|0111 & \text{померај} \\
|0001|0000|1\ 0111 & \text{померај} \\
|1000|01\ 0111 & \text{померај} \\
|0101|01\ 0111 & \text{одузимање } 3 \\
|0010|101\ 0111 & \text{померај} \\
|0001|0101\ 0111 & \text{померај} \\
|1\ 0101\ 0111 & \text{померај } 256+64+16+4+2+1=343_{10}
\end{array}$$

Алгоритам BINBCD1

који врши конверзију бинарног броја у BCD број *коришћењем одузимања.* Овај алгоритам подразумева да се користи табела бинарних еквивалената тежинским факторима 1, 2, 4, 8, 10, 20, 40, 80, 100, 200 итд. Из табеле се нађе највећи број који је мањи или једнак датом бинарном броју. Тај се број одузме од бинарног броја. Поступак се понавља са остатком све док остатак не постане 0. На свакој позицији где је вршено одузимање упише се јединица, док се остале позиције попуне нулама. Добијени запис представља BCD еквивалент декадног броја. Као пример посматрајмо конверзију бинарног броја 11010101_2 у BCD.

Пример 8.

$$\begin{array}{rl}
11010101 & \text{бинарни број} \\
-11001000 & -200 \quad (p9) \\
\hline
00001101 & \text{остатак} \\
-00001010 & -10 \quad (p3) \\
\hline
00000011 & \text{остатак} \\
-00000010 & -2 \quad (p1) \\
\hline
00000001 & \text{остатак} \\
-00000001 & -1 \quad (p0) \\
\hline
00000000 & \text{крај}
\end{array}$$

Конверзијом је добијена вредност $0010\,0001\,0011_{BCD}$.

Алгоритам BINBCD2

врши конверзију бинарног у BCD запис *коришћењем померања и сабирања.* Прво се изврши подела у декаде од по 4 бита у које ће бити смештене цифре BCD броја. Бинарни број се испише десно од декаде најмање вредности. Затим се бинарни број помери улево за један бит. Потом се испитује да ли декаде представљају бројеве $\geqslant 5$. Ако је то случај, додаје се 3 садржају такве декаде.

Поступак померања и испитивања се наставља све док се сви битови бинарног броја не помере у област BCD броја. После сваког померања добијени BCD број представља еквивалент бинарног броја који је померен до тог тренутка. Да би се илустровао алгоритам померања и сабирања биће извршена конверзија бинарног броја 101010111_2 у BCD формат. Конверзија се може приказати следећом табелом:

Пример 9.

<table>
<tr><td align="right">|101010111</td><td>бинарни број</td></tr>
<tr><td align="right">1 |01010111</td><td>померај</td></tr>
<tr><td align="right">10 |1010111</td><td>померај</td></tr>
<tr><td align="right">101 |010111</td><td>померај</td></tr>
<tr><td align="right">|1000|010111</td><td>сабирање са 3</td></tr>
<tr><td align="right">1|0000|10111</td><td>померај</td></tr>
<tr><td align="right">10|0001|0111</td><td>померај</td></tr>
<tr><td align="right">100|0010|111</td><td>померај</td></tr>
<tr><td align="right">1000|0101|11</td><td>померај</td></tr>
<tr><td align="right">1011|1000|11</td><td>сабирање са 3</td></tr>
<tr><td align="right">1|0111|0001|1</td><td>померај</td></tr>
<tr><td align="right">1|1010|0001|1</td><td>сабирање са 3</td></tr>
<tr><td align="right">11|0100|0011|</td><td>крај конверзије $0011\,0100\,0011_{BCD} = 343_{10}$</td></tr>
</table>

Дакле, од прихватања улазних података у декадном облику до издавања декадних резултата обавља се низ конверзија: 1) ASCII–BCD конверзија улазних података; 2) BCD–BIN конверзија; 3) обрада BIN улазних података и добијање BIN резултата; 4) BIN–BCD конверзија; 5) BCD–ASCII конверзија резултата. Све је ово, разуме се, скривено од корисника који ради са апликативним програмима или програмира у вишим програмским језицима. Ако сте од оних које интересује више од оног што се види на површини, ево како би на симболичком језику Интелових микропроцесора x86 могла да изгледа реализација алгоритма BCDBIN2 (без оптимизација, јер би оне отежале праћење програма).

```
;procedura za konvertovanje pakovanog BCD broja u binarni zapis
;           ulaz ax=pakovan bcd broj
;           izlaz ax=binarni prevod ulaznog podatka
;procedura cuva sadrzaje svih preostalih opstih registara
bcdbin2     proc
            push bx    ;pamcenje izvornog sadrzaja radnih registara
            push cx
            push dx
            mov bx,0   ;registar koji prihvata "istisnut" bit
            mov dh,16  ;brojac preostalih bitova za konverziju
```

```
zavrsi:     cmp ax,0    ;konvertovan ceo sadrzaj?
            je gotovo
            shr ax,1
            rcr bx,1    ;0.  bit iz ax na 15.  bit u bx
            dec dh
            mov cx,4    ;postupak testiranja obavice za 4 dekade
                        ;testiranje da li je u dekadi broj >=8
proveri:    mov dl,0fh  ;maska za izdvajanje mladjeg polubajta
            and dl,al
            cmp dl,8
            jl dalje    ;ako je dekada >=8, oduzimanje 3
            sub al,3
dalje:      ror ax,4    ;testirana dekada na najstariju poziciju
            loop proveri;ponavljanje postupka za sledecu dekadu
            jmp zavrsi
gotovo:     mov cl,dh
            shr bx,cl   ;pomeranje broja iz bx na najnizu poziciju
            mov ax,bx
            pop dx      ;obnavljanje izvornih sadrzaja radnih reg.
            pop cx
            pop bx
            ret
bcdbin2     endp
```

Колико детаљно ће и на који начин у оквиру часова рачунарства и информатике наставници упознати ученике са начинима регистровања података у рачунару, бинарним бројним системом и проблемима превођења бројева зависи, разуме се, од наставног програма и заинтересованости ученика. Наше искуство говори да ученици много боље усвајају оне поступке и чињенице које сами освајају, па зато о превођењу бројева говоримо више пута у току школовања. У првом пролазу очекујемо да ученици умеју да преведу број из декадног у бинарни систем и обрнуто, директно и преко окталног или хексадекадног бројног система, и да знају да се разломљени бројеви у општем случају не могу тачно превести. У другом пролазу тражимо од ученика да напишу програме за превођење бројева из декадног и у декадни бројни систем, да направе целобројни калкулатор за рачунање у бројном систему задате основе и слично. Касније (радимо по програмима у којима се рачунарство и информатика изучава све четири године) поново се враћамо на проблеме превођења приликом упознавања симболичког језика програмирања. Тада детаљније говоримо о начинима регистровања података и очекујемо да ученици разумеју не само како се реализују бинарна рачунања, већ и како се симулира декадна аритметика (са BCD бројевима) и како се врше потребне конверзије података од улаза до презентирања излазних резултата. Разуме се, при раду са мањим фондом часова не може се оволико времена посветити проблемима превођења бројева, али се одговарајућа тема може понудити за семинарски или матурски рад.