Sekvenční dělení

INP 2019

FIT VUT v Brně

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Dělení čísel s pevnou řádovou čárkou

Budeme zabývat dělením čísel s pevnou řádovou čárkou **bez znaménka**.
Pro jednotlivé činitele operace dělení zavedeme symboly

D	dělenec
d	dělitel
Q	podíl
q_i	i-tý bit podílu
R_i	i-tý (průběžný) zbytek

Máme vypočítat Q, R tak, aby byla splněna rovnice

$$D = Q.d + R, \qquad 0 \le |R| < d.$$

Pro d, Q, R máme k dispozici n bitů, pro D vyhradíme $2n$ bitů.

Nejdříve si ukážeme dělení čísel bez znamének, resp. dělení jejich absolutních hodnot.

Pozor, d je nutné posunout o n bitů doleva.

Příklad 2293 : 51 (dekadicky vs binárně, n = 6)

Posunutý dělitel

```
 2293 : 51 = 0           100011110101  :  110011000000 = 0        (i = 0)
-00                     -000000

 2293 : 51 = 04          100011110101  :   11001100000 = 01
-204                     -110011

 0253 : 51 = 044          01010010101  :    1100110000 = 010
- 204                     -000000

   49 (zbytek)            01010010101  :     110011000 = 0101
                           -110011

                             11111101  :      11001100 = 01011
                            -110011

                             00110001  :       1100110 = 010110
                            -000000

                             00110001  :        110011 = 0101100 (44)

                               110001 (49, zbytek)
```

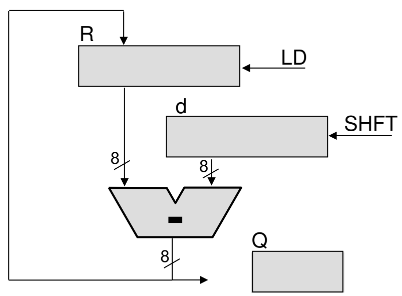
V kroku i se pokoušíme odečíst od průběžného zbytku R_i posunutý dělitel $2^{n-i}\,d$

Postup dělení a HW realizace

Při rozhodování o hodnotě bitu podílu q_{n-i} jsme postupovali podle vztahů:
je-li $2^{n-i}\,d$ **menší než nebo rovno** R_i, pak $q_{n-i} = 1$,
je-li $2^{n-i}\,d$ **větší** než R_i, pak $q_{n-i} = 0$.

$Q = q_n \dots q_0$
Nový zbytek se vypočte:

$$R_{i+1} := R_i - q_{n-i}\,2^{n-i}\,d$$



$n = 4$

Modifikovaný postup – d je v pevné poloze

HW realizace z předchozího obrázku vyžaduje uchovávat hodnotu dělitele na $2n$ bitech a používat $2n$ bitovou odčítačku / sčítačku => zbytečně drahé.

V praxi se posouvá dílčí/průběžný zbytek vlevo a *d* je v pevné poloze (posunut o 2^n bitů), takže

$$R_{i+1} := 2R_i - q_{n-i}\, d\, 2^n$$

Dále označme $d' = d \cdot 2^n$.

Dělení – modifikovaný postup (38 : 5)

(Praxe: posuv R_i vlevo, d v pevné poloze)

D=100110 d=101 ($d' = d.2^n = 101000$)

$Q= q_3 q_2 q_1 q_0$ = **0111** $R_{i+1}= 2R_i - q_{n-i}d'$

i=0	100110	$2R_0$=D	$d' > 2R_0$	=>	Q=**0**
	-000	q_3d'			
	100110	R_1			
i=1	100110x	$2R_1$	$d' < 2R_1$	=>	Q=0**1**
	-101000	q_2d'			
	10010x	R_2			
i=2	10010xx	$2R_2$	$d' < 2R_2$	=>	Q=01**1**
	-101000	q_1d'			
	1000xx	R_3			
i=3	1000xxx	$2R_3$	$d' < 2R_3$	=>	Q=011**1**
	-101000	q_0d'			
	011xxx	R = 011			

Pravidlo pro určení q_{n-i}

Z příkladu plyne pravidlo pro určení q_{n-i} :

když *d'* je **větší** než $2R_i$, pak $q_{n-i} = 0$,

jinak $q_{n-i} = 1$.

V praxi se porovnávání čísel založené na použití komparátorů nepoužívá. Odečtení se provede vždy, tedy

když $2R_i - d'$ je **menší** než 0, pak $q_{n-i} = 0$,

jinak $q_{n-i} = 1$.

Dva postupy dělení

a) Je-li $q_{n-i} = 0$, pak je výsledek "zkušebního" odečtení $R_{i+1}' = 2R_i - d'$. Správný zbytek má ale být $R_{i+1} = 2R_i$. Správný zbytek dostaneme opravou, přičtením $+d'$, což nazýváme ***restaurací nezáporného zbytku (návrat k nezápornému zbytku),*** tedy $R_{i+1} := R_{i+1}' + d'$. Je-li pravděpodobnost výskytu jedničky v podílu *Q* rovna 1/2, potřebujeme pro *n* odečtení v průměru ještě *n/2* krát přičítat.

b) Postup ***bez restaurace nezáporného zbytku (bez návratu k nezápornému zbytku):***

když $q_{n-i} = 1$ (R_i větší než 0), použijeme vztah $R_{i+1} := 2R_i - d'$,

když $q_{n-i} = 0$, použijeme vztah $R_{i+1} := 2R_i + d'$.

Postup je úspornější, protože v každém kroku jen přičítáme *d'*, nebo odčítáme *d'*, nikdy neprovádíme obě operace. Při dělení bez restaurace tedy provedeme *n* aritmetických operací (+ nebo -), při dělení s restaurací *3/2 n* operací.

```
30=00011110, 7=0111, -7=1001
 00011110
+1001          -d
 10101110      <0 => c4 = 0
 0101110x      posuv <-
+0111          +d
 1100110x      <0 => c3 = 0
 100110xx      posuv
+0111          +d
 000010xx      >0 => c2 = 1
 00010xxx      posuv
+1001          -d
 10100xxx      <0 => c1 = 0
 0100xxxx      posuv
+0111          +d
 1011xxxx      <0 => c0 = 0
+0111          +d (korekce na kladný zbytek)
 0010xxxx      zbytek 2
```

Příklad: 30:7=4, zb 2

bez návratu k nezápornému zbytku

Dělení SRT

Dělení čísel se znaménkem se po dlouhou dobu převádělo na dělení absolutních hodnot a dodatečné určení znaménka výsledku. Tuto nedokonalost odstranil ***algoritmus SRT*** **autorů Sweeneyho, Robertsona a Tochera** (1958). Prováděné operace se pro každý krok určují "zejména" podle nejvyšších bitů průběžného zbytku R_i.

Pro demonstraci principu uvedeme základní metodu, kdy se operace provádí podle 3 nejvyšších bitů R_i.

Ri	d>0 bit podílu	d>0 operace	d<0 bit podílu	d<0 operace
000 111	0	posuv vlevo	0	posuv vlevo
001 010 011	1	-d, posuv vlevo	-1	+d, posuv vlevo
101 110 100	-1	+d, posuv vlevo	1	-d, posuv vlevo


```
49=00110001, 7=0111, -7=1001
-49=11001111 (d>0)

-1   11001111
     0111      +d
     00111111  <-
+1   0111111x
     1001      -d
     0000111x
0    000111xx  <-
+1   00111xxx  <-
     1001      -d
     11001xxx  <-
-1   1001xxxx
     0111      +d
     0000xxxx
     zbytek 0

     Uplatníme váhy 2^n..2^0
     Q = -1 1 0 1 -1, tj.
         -16+8+0+2-1 = -7
```

Příklad:

-49:7=-7, zb. 0

SRT

Ri	d>0 bit podílu	d>0 operace	d<0 bit podílu	d<0 operace
000 111	0	posuv vlevo	0	posuv vlevo
001 010 011	1	-d, posuv vlevo	-1	+d, posuv vlevo
101 110 100	-1	+d, posuv vlevo	1	-d, posuv vlevo

Pozn. V případě záporného zbytku je zapotřebí provést korekci na kladný zbytek a **zvýšit** (pro d<0), popř. **snížit** (pro d>0) Q o 1.

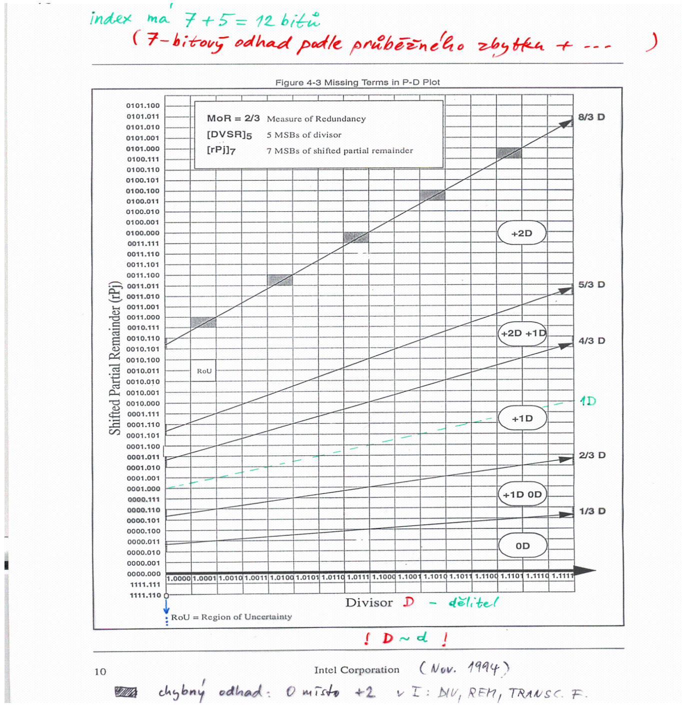
Reálný algoritmus dělení SRT

Praktické realizace postupu SRT určují hodnotu číslice podílu podle více bitů průběžného (okamžitého) zbytku a podle více bitů dělitele.

Dále pracují s kódováním „několik bitů najednou".

Například v Pentiu se určují číslice podílu **podle 7 bitů průběžného zbytku a podle 5 bitů hodnoty dělitele**. Používá se radix 4.

Chyba dělení u Pentia
(listopad 1994,
ztráta $475M)



Obvodové realizace dělení: Shrnutí

- Sekvenční děličky
  - viz předchozí slidy
- Kombinační dělička
  - založena na úplné odčítačce
  - obvodová struktura je podobná kombinační násobičce
- Iterační dělička
  - viz další přednáška