Ganzzahlige Division per Multiplikation mit dem doppelt-breiten Kehrwert

M. Anton Ertl
TU Wien

Division und Multiplikation

- Divisionsbefehle sind teuer
  Ganzzahlige Division auf Skylake:
  Latenz: 35–42 Zyklen
  Durchsatz: 1 Division pro 21–24 Zyklen

- Multiplikation ist billiger
  Ganzzahlige Multiplikation auf Skylake:
  Latenz: 4 Zyklen
  Durchsatz: 1 Multiplikation pro Zyklus

- $n/d \to n(1/d)$
  für konstante d
  für schleifeninvariante d

Vorzeichenlose Computerarithmetik: Randbedingungen

$$0 \leq n, d < 2^w$$

$$q = \lfloor \frac{n}{d} \rfloor$$

$$d = 0?$$

bestimmt von der Sprache
oder von der unoptimierten Implementierung

$$p = ab$$

$$0 \leq a, b < 2^w$$

$$0 \leq p < 2^{2w}$$

Multiplikation mit dem Kehrwert

$$q = \lfloor \frac{n}{d} \rfloor = \lfloor \frac{nC}{2^k} \rfloor$$

Welches C? Welches k?
Allgemein: $C < 2^w$ reicht nicht

Frühere Arbeiten: $C < 2^w$, Abweichungen korrigieren, z.B. Robison/Fish:

$$\lfloor \frac{n}{d} \rfloor = \lfloor \frac{nC + b}{2^k} \rfloor$$

Diese Arbeit: $C = \lceil \frac{2^{2w}}{d} \rceil = \lfloor \frac{2^{2w} + d - 1}{d} \rfloor$ $q = \lfloor \frac{nC_l + 2^w nC_h}{2^{2w}} \rfloor = \lfloor \frac{nC_l}{2^{2w}} + \frac{nC_h}{2^w} \rfloor$

Spezialfall: $d = 1$

Code

gcc $n/10$
Latenz: 6 Zyklen

```
movabs $C,%rdx
mov    %rdi,%rax
mul    %rdx
mov    %rdx,%rax
shr    $0x3,%rax
```

gcc $n/7$
Latenz: 8 Zyklen

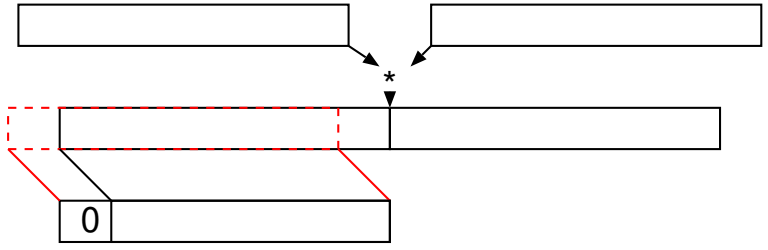
```
movabs $C,%rdx
mov    %rdi,%rax
mul    %rdx
sub    %rdx,%rdi
shr    %rdi
lea    (%rdx,%rdi),%rax
shr    $0x2,%rax
```

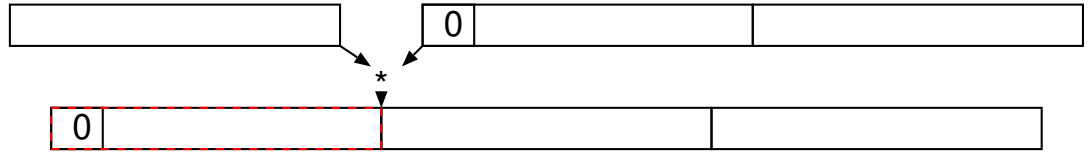
Fish $n/7$
Latenz: 6,25 Zyklen

```
movabs $C,%rax
mov    %rax,%rcx
mul    %rdi
add    %rcx,%rax
adc    $0x0,%rdx
shr    $0x2,%rdx
mov    %rdx,%rax
```

diese Arbeit $n/7$
Latenz: 6 Zyklen

```
movabs $Cl,%rax
mul    %rdi
mov    %rdx,%rcx
movabs $Ch,%rax
mul    %rdi
add    %rcx,%rax
adc    $0x0,%rdx
mov    %rdx,%rax
```





Berechnung von C

$$C = \lceil \frac{2^{2w}}{d} \rceil = \lfloor \frac{2^{2w} + d - 1}{d} \rfloor$$

$$C_h = \lfloor \frac{2^w}{d} \rfloor \qquad r_h = 2^w \bmod d$$

$$C_l = \lfloor \frac{2^w r_h + d - 1}{d} \rfloor$$

```
#in:  d = %rdi
mov    $1,%rdx
mov    $0,%rax
div    %rdi
mov    %rax,%rsi
lea    -1(%rdi),%rax
div    %rdi
mov    %rsi,%rdx
#out: ch=%rdx cl=%rax
```

Division in $\mathbb{Z}$

symmetrisch:	$r = 0$	$\vee$	$\mathrm{sgn}(r) = \mathrm{sgn}(n)$
floored:	$r = 0$	$\vee$	$\mathrm{sgn}(r) = \mathrm{sgn}(d)$
Euklidisch:	$r \geq 0$		

	$-7/2$		$7/-2$		$-7/-2$	
	q	r	q	r	q	r
symmetrisch	-3	-1	-3	1	3	-1
floored	-4	1	-4	-1	3	-1
Euklidisch	-4	1	-3	1	4	1

Negative Zahlen und Computerarithmetik

Zweierkomplementdarstellung: $-2^{w-1} \leq x < 0$ repräsentiert durch $x' = x + 2^w$

$$ab = (a' - 2^w)b = a'b - 2^w b \quad \text{wenn } a < 0$$

Mit vorzeichenloser Multiplikation: abhängig von sgn(d) korrigieren

Oder Multiplikation mit Vorzeichen:
braucht trotzdem Korrekturen
Komplikation durch nötige sign extension

Zusammenfassung

- Multiplikation viel billiger als Division
- Division durch Konstante oder Schleifeninvarianten:
  Multiplikation mit dem Kehrwert
- Einfachbreiter Kehrwert braucht Shift und evtl. Korrekturen
- Doppeltbreiter Kehrwert braucht breite Multiplikation
- vorzeichenlos: doppeltbreit konkurrenzfähig