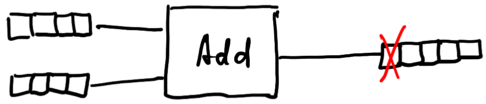
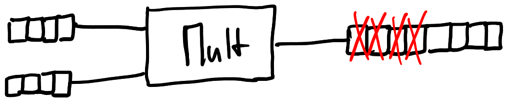
Clock arithmetic und negative Zahlen

schriftliche Addition ⇝ 4-Bit-Addierer

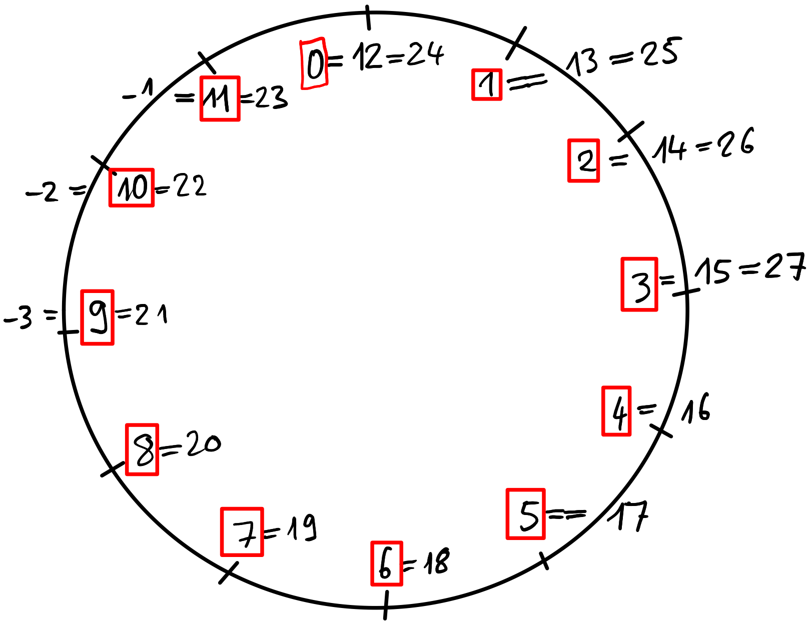


analog: schriftliche Multiplikation ⇝ 4-Bit-Multiplizierer



Annahme jetzt: Computer hat 4 Bit für jede Zahl.
Ergebnis von $\boxed{\text{Add}}$ und $\boxed{\text{Mult}}$ stimmt nur noch bis auf Vielfache von $16 = 2^4$.

Exkurs: Clock arithmetic = Rechnen mit Uhrzeiten = Rechnen mit Resten
= modulare Arithmetik



„=" bedeutet hier: unterscheidet sich um Vielfaches von 12

Rechnen mit roten Uhrzeiten: Addition: addiere wie üblich, nimm Rest bei Division durch 12 (= Rest modulo 12)
Multiplikation: multipliziere —"—
gleichbedeutend: – diejenige rote Uhrzeit, die sich vom Ergebnis um ein Vielfaches von 12 unterscheidet.

Aufgaben:

$3 + 7 = 10$

$9 + 7 = (16 =)\ 4$

$3 \cdot 2 = 6$

$3 \cdot 7 = (21 =)\ 9$

$9 \cdot 7 = (63 =)\ 3$

Die üblichen Rechenregeln (Kommutativ-, Assoziativgesetze, Distributivgesetz) gelten.

Aufgaben: Löse

$5 + x = 7$ — $x = 2$

$7 + x = 5$ — $x = 10$

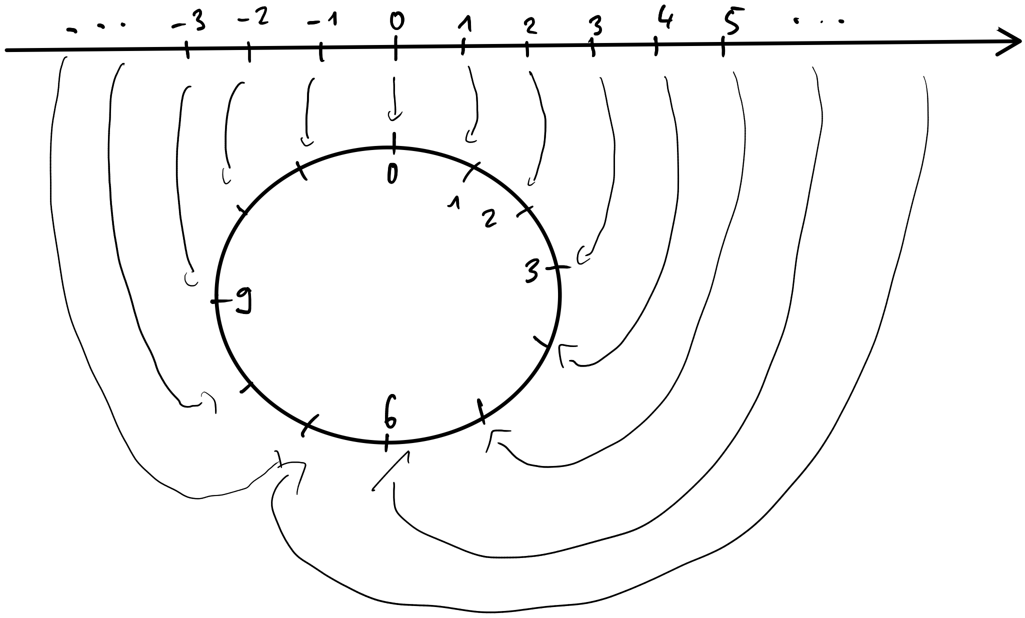
$7 + x = 0$ — $x = 5\ (= -7)$ — $\overline{5} = \overline{-7}$

$7 \cdot x = 11$ — $x = 5$ (da $7 \cdot 5 = 35 = 2 \cdot 12 + 11 = 11$)

$7 \cdot x = 1$	$x = 7$
$2 \cdot x = 11$	keine Lösung
$3 \cdot x = 1$	— " —
$2x + 1 = 3$	$x = 1$ oder $x = 7$
$2x + 2 = 3$	keine Lösung
$3 \cdot x = 9$	$x = 3$ Nachtrag: $x = 7$, $x = 11$

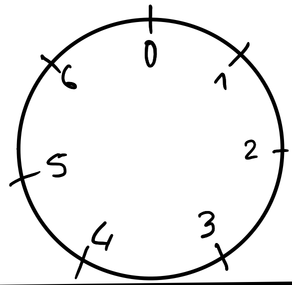
Die Menge dieser Uhrzeiten wird als $\mathbb{Z}/12\mathbb{Z}$ notiert. „$\mathbb{Z}$ modulo $12\mathbb{Z}$“.

Vorstellung: Zahlenstrahl $\mathbb{Z}$ wird zum „Zahlenring“ $\mathbb{Z}/12\mathbb{Z}$ aufgerollt

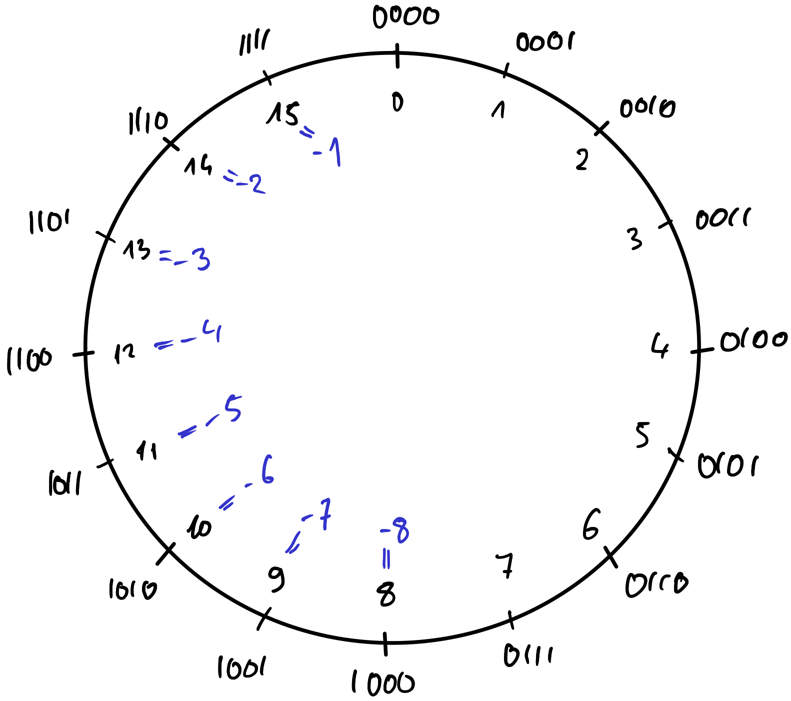


Statt 12 kann jede andere natürliche Zahl nehmen, etwa

$\mathbb{Z}/7\mathbb{Z}$ = Uhr mit 7 Stunden



Fazit: Unsere Bauteile $\boxed{\text{Add}}$ und $\boxed{\text{Mult}}$ rechnen auf der 16-Stunden-Uhr.



Da wir jetzt Uhrzeiten, die sich um 16 oder Vielfache unterscheiden, als gleich ansehen, können wir alle Ziffern $8, 9, 10, \ldots, 15$ um 16 verringern

$$\begin{array}{cccc} 8, & 9, & 10, & \ldots, 15 \\ \| & \| & \| & \quad \| \\ -8 & -7 & -6, & \ldots\ -1 \end{array}$$

Mit anderen Worten: alle 4-stelligen Binärzahlen, deren erstes Bit 1 ist, werden nun als negative Zahl / Uhrzeit interpretiert.

Fazit:

- Bauteile $\boxed{\text{Add}}$ und $\boxed{\text{Mult}}$ können mit negativen Zahlen rechnen (Ergebnisse korrekt bis auf Vielfache von 16).
- Wir können negative Zahlen binär darstellen und unsere Bauteile $\boxed{\text{Add}}$ und $\boxed{\text{Mult}}$ weiterverwenden.
  Alternativ könnte Zusatzbit verwenden, das das Vorzeichen kodiert $\left(\begin{smallmatrix} + & 0 \\ - & 1 \end{smallmatrix}\right)$,
  aber Nachteil: bräuchte neues Bauteil $\boxed{\text{Subtraktion}}$
- In Praxis rechnet Computer mit 32-stelligen Binärzahlen,
  d.h. auf Uhr mit $2^{32} \approx 4$ Milliarden
  mit Zahlen von ≈ -2 Milliarden bis ≈ 2 Milliarden
  Die Ergebnisse stimmen bis auf Vielfache von 2^{32},
  d.h. Ergebnisse korrekt, falls man mit „kleinen" Zahlen rechnet.