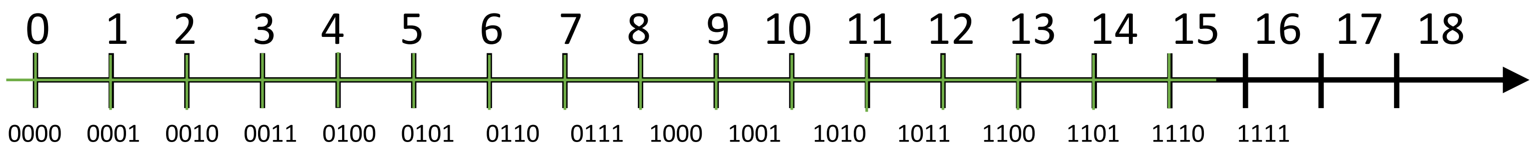
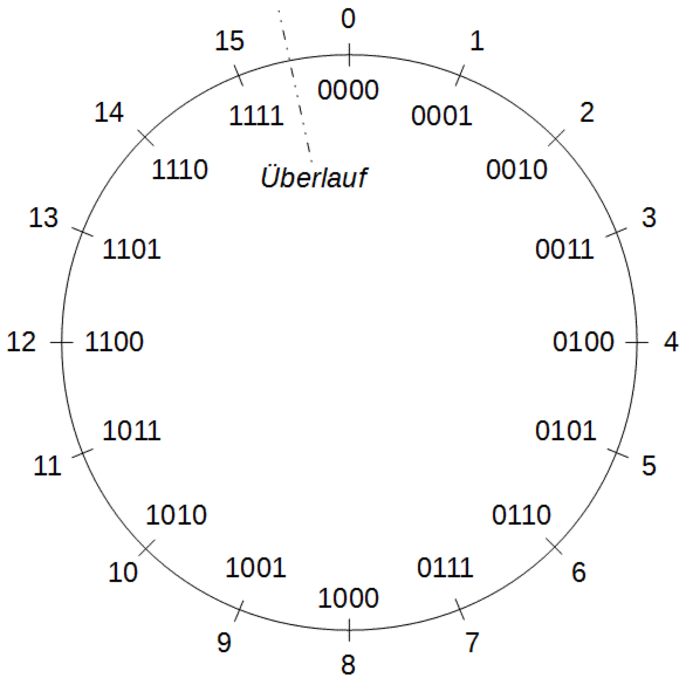
Ganze positive Zahlen



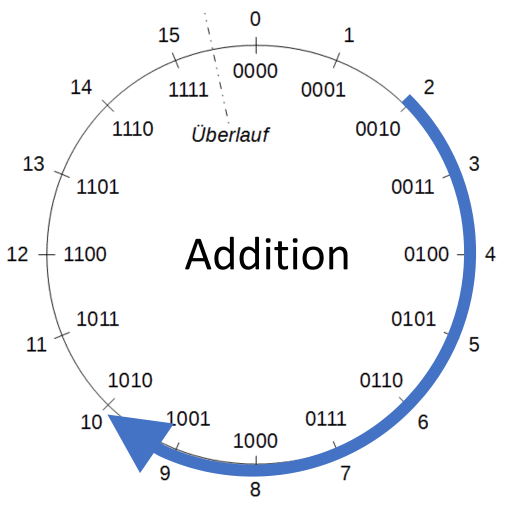
- Die Menge der ganzen positiven Zahlen ist unendlich.
- Im Computer können wir aber,
  bedingt durch die Wortbreite eines Rechners, nur endliche Zahlen bis zur Größe der Wortbreite darstellen.
- Wir haben es daher nicht mit einem Zahlenstrahl zu tun, sondern mit einem Zahlenkreis pro Maschinenwort
  (hier: 4 Bit, 0…15)



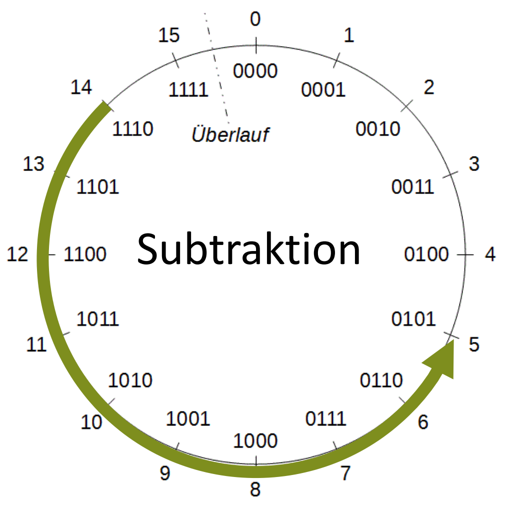
Zahlenkreis für 4-Bit-Zahlen

Addition und Subtraktion

Im Zahlenkreis lassen sich Addition und Subtraktion durch Drehungen darstellen.

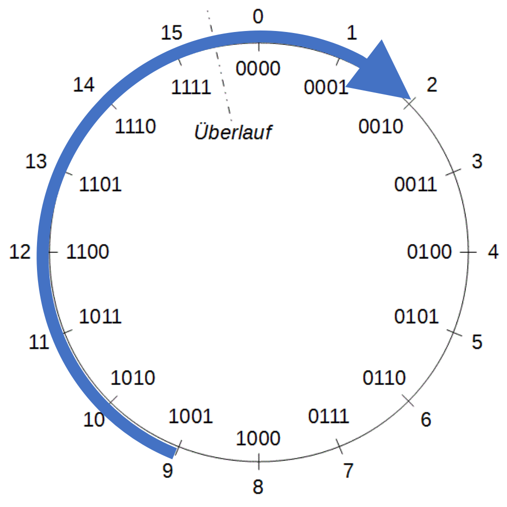


2 + 8 = 10



14 - 9 = 5

Überschreitungen des Zahlenbereichs



9 + 9 = 2?

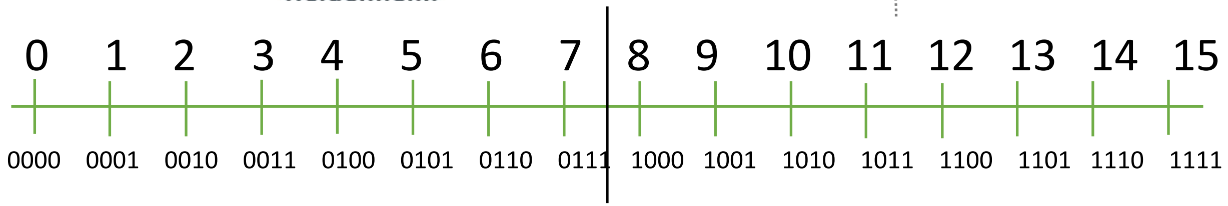
9 + 9 = 2 + Carry
= 2 + 16
= 18

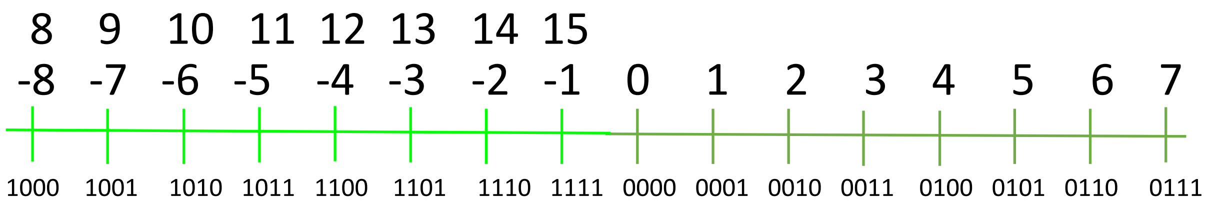
- Bsp. Uhr: 19 Uhr + 9 Stunden= 28 Uhr???
- Die Addition zweier Zahlen, deren Ergebnis den Zahlenbereich überschreitet, führt auf den ersten Blick zu keinem sinnvollen Ergebnis.
- Die Information, dass der Zahlenbereich überschritten wurde, wird im

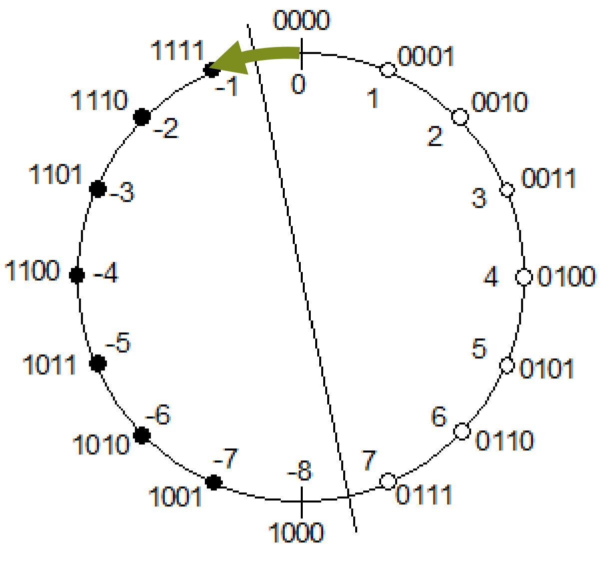
  Carry Flag (Übertragsflag)

  der CPU gespeichert.
- Das Ergebnis und das Carry-Flag kann für weitere Rechnungen herangezogen werden, so dass selbst bei beschränktem Zahlenbereich beliebig genaue Rechnungen möglich sind

Einführung von Negativen Zahlen



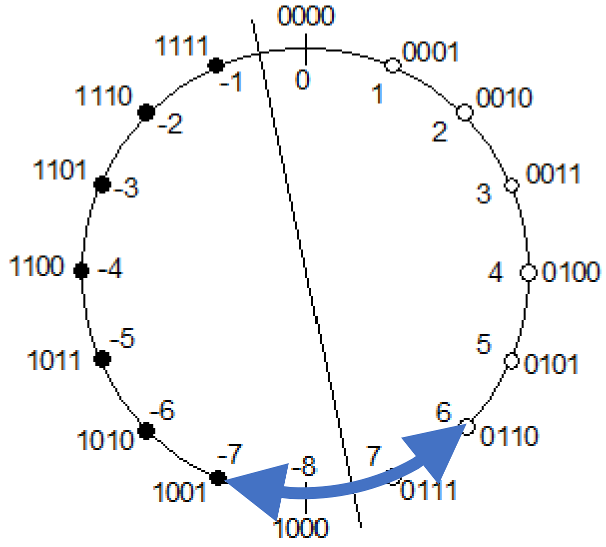




0 - 1 = 15?

- Im Bereich der positiven Zahlen inkl. Null ist die Subtraktion nur mit positivem Ergebnis definiert
- Negative Ergebnisse sind erst bei einer Erweiterung des Zahlenbereichs auf negative Zahlen sinnvoll.
- Die negativen Zahlen müssen nun auf einen Bereich des **vorhandenen (binären) Wertebereichs** abgebildet werden
- Das Verwenden des höchstwertigen Bit als Vorzeichen reduziert den positiven Wertebereich um 1 Bit, erlaubt aber den gleichen reduzierten Wertebereich im *2er-Komplement* als negativ anzusehen
- Addition und Subtraktion bleiben dann gleich

Die Einbettung der negativen Zahlen



- Das Problem dieser Einbettung ist, dass der Bereich der positiven Zahlen nur noch halb so groß ist
- Eine Addition zweier positiver Zahlen im Maschinenwort kann eine negative Zahl als Ergebnis haben
- Wieder wird der Übertrag im Statusregister gespeichert, dieses mal als

  Overflow Flag (Überlauf Flag)

- Das Overflow Flag wird immer dann gesetzt, wenn die Summe 2er positiver Zahlen eine negative Zahl oder die Summe 2er negativer Zahlen eine positive Zahl ergibt

Die Negation

dez.	binär	dez. negiert	binär negiert
0	0000	0	0000
1	0001	-1	1111
2	0010	-2	1110
3	0011	-3	1101
4	0100	-4	1100
5	0101	-5	1011
6	0110	-6	1010
7	0111	-7	1001
8	1000	-8	1000

- Eine wichtige Umrechnung ist es, zu einer Zahl das negative Äquivalent zu bestimmen.
- Es ist leicht zu sehen, dass die Zahl -2 überall dort eine Eins hat, wo die Zahl +1 eine Null besitzt. Das gleiche gilt für -4 und +3 und für alle weiteren Zahlen.
- Wir können daher eine Zahl negieren, indem wir das
  1er-Komplement
  der Zahl bilden (alle 0 durch 1 und alle 1 durch 0 ersetzen) und zu der erhaltenen Zahl 1 addieren:
  1er-Komplement + 1
- Man nennt diese Operation auch die Bildung des
  2er-Komplements.

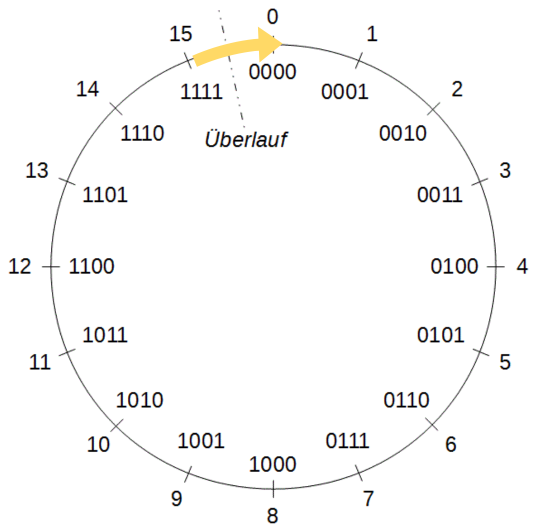
Zusammenfassung binäre Zahlen

Rechnen mit Zahlen ohne Vorzeichen (unsigned, voller Wertebereich):
Übertrag (Carry, C):

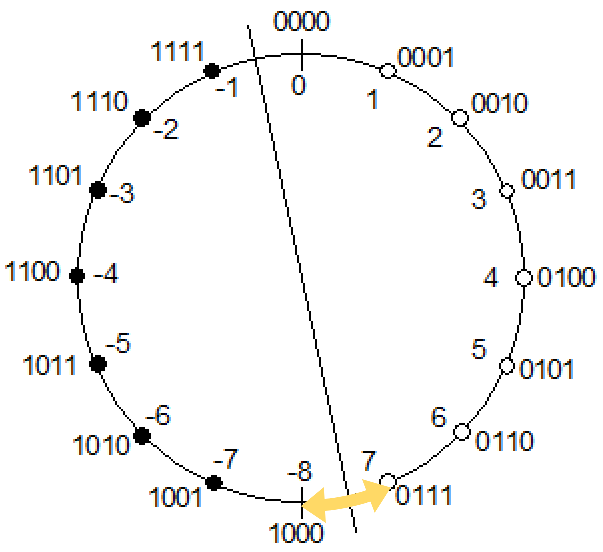
falls bei Operationen mit als vorzeichenlos angesehenen Zahlen der Wertebereich überschritten wird.

4 – 2 = 4 + (-2)
= 4 + 14
= 2 und C

4 – 6 = 4 + (-6)
= 4 + 10
= 14



Beachten bei Zahlen mit Vorzeichen (signed, 2er-Komplement):



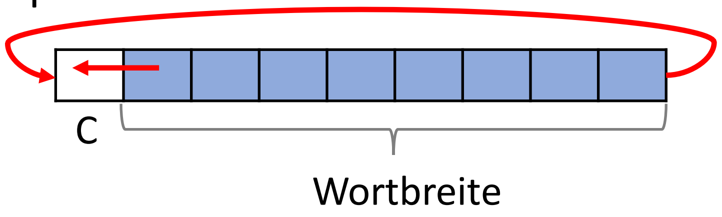
Überlauf (Overflow):

falls bei Operationen von 2er-Komplement-Zahlen der (reduzierte) Wertebereich gewechselt wird.

Binäre Rechenoperationen im Rechner

Das Carry-Flag (C)

Das Carry-Flag ist eine Information, die im Statusregister einer CPU gespeichert wird.



Das Carry-Flag wird gesetzt:

- bei einer Addition auf den Übertrag aus dem höchsten Bit
  (Substraktionen führen manche Prozessoren als Addition des negierten Operanden durch: Carry wird prozessorspezifisch unterschiedlich gesetzt)
- bei einer Schiebeoperation nach links, auf den Wert des zuletzt nach links aus dem Wort herausgeschobenen Bits
- bei einer Schiebeoperation nach rechts, auf den Wert des zuletzt nach rechts aus dem Wort geschobenen Bits
- bei einer Rotationsoperation auf das zuletzt aus dem Wort hinausrotierten Bits

Addition und Subtraktion im Rechner

- Die Addition im Rechner wird nach dem gleichen Verfahren durchgeführt wie die schriftliche Addition
- Die Subtraktion kann zurückgeführt werden auf eine Negation des Subtrahenden und eine anschließende Addition:

a – b = a + (-b)

$$3=0*2^3+0*2^2+1*2^1+1*2^0$$
$$5=0*2^3+1*2^2+0*2^1+1*2^0$$

$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & 0_1 & 1_1 & 0_1 & 1 \\ \hline & 1 & 0 & 0 & 0 \end{array}$$

Übertrag / Carry

3+ (-5):

$$\begin{array}{ccccc} & 0 & 0 & 1 & 1 \\ {}_0 & +1 & 0_1 & 1_1 & 1 \\ \hline (0) & 1 & 1 & 1 & 0 \end{array}$$

3-5:

$$\begin{array}{ccccc} & 0 & 0 & 1 & 1 \\ {}_1 & -\,0_1 & 1 & 0 & 1 \\ \hline (1) & 1 & 1 & 1 & 0 \end{array}$$

Übung

Wertebereich: 4 Bit, vorzeichenbehaftet

4 + 3 = ?

A: 7, CY=0

B: 7, CY=1

C: -7, CY=0

Übung

Wertebereich: 4 Bit, vorzeichenbehaftet

4 + 7 = ?

A: 11, CY=1

B: -5, CY=1

C: -5, CY=0

D: -4, CY=0

E: -4, CY=1

Übung

Wertebereich: 4 Bit, vorzeichenbehaftet

-8 + -8 = ?

A: 16, CY=1

B: 0, CY=1

C: -5, CY=0

D: -7, CY=0

E: 0, CY=0

Übung

Wertebereich: 4 Bit, vorzeichenbehaftet

3 – 5 = ?

A: -2, CY=0

B: 2, CY=1

C: 2, CY=0

D: -2, CY=1

E: 0, CY=0

Übung

Wertebereich: 4 Bit, vorzeichenlos

14 + 3 = ?

A: 2, CY=0

B: 2, CY=1

C: 1, CY=1

D: -2, CY=1

E: 1, CY=0

Setzen der Flags durch den ARM-Prozessor

			Flags			
4-Bit-Operation	Ergebnis (signed)	Ergebnis (unsigned)	negative N (signed)	zero Z	carry C(Y) (unsigned Addit.)	overflow V (signed)
0 - 1	-1	15	1	0	0	0
5 + 7	-4	12	1	0	0	1
7 + 10	-	1	0	0	1	0
5 + 1	6	6	0	0	0	0
5 - 1	4	4	0	0	1	0

Das Overflow-Flag (V)

Das Overflow-Flag zeigt eine Überschreitung des Zahlenbereichs bei einer signed Addition (oder einer signed Subtraktion) an.

Das Overflow-Flag wird gesetzt, wenn:

- Die Addition zweier positiver Zahlen ein negatives Ergebnis hat (4+7 im 4-Bit-Bereich)
- Die Addition zweier negativer Zahlen ein positives Ergebnis hat (-8 + -8 im 4-Bit-Bereich)
- (Subtraktion ist die Addition einer negativen Zahl)

Das Overflow-Flag (V): Berechnungstip

Das Overflow-Flag wird gesetzt, wenn Carry_{in} ungleich Carry_{out} ist.:

7 + 7 = -2   -5 + -5 = +6   3 + -5 = -2

$$\begin{array}{r} 0\;1\;1\;1 \\ +_{0}0_{1}1_{1}1_{1}1 \\ \hline 1\;1\;1\;0 \end{array} \qquad \begin{array}{r} 1\;0\;1\;1 \\ +\;_{1}1_{0}0_{1}1_{1}1 \\ \hline 0\;1\;1\;0 \end{array} \qquad \begin{array}{r} 0\;0\;1\;1 \\ +\;_{0}1_{0}0_{1}1_{1}1 \\ \hline 1\;1\;1\;0 \end{array}$$

Carry_{out} = Carry Carry_{in}

DHBW
Duale Hochschule
Baden-Württemberg
Heidenheim

Die Negation

Die Negation einer Zahl, d. h. die Wandlung von einer positiven in eine negative Zahl (*2er-Komplement*) oder umgekehrt, ist bei manchen Prozessoren als eigener Befehl vorhanden.

Andere Prozessoren wie ARM führen die Negation auf eine Subtraktion zurück:

NEG(x) = 0 + (– x)

Logische Befehle AND und OR

Neben den Additions- und Subtraktionsbefehlen besitzt jede CPU noch einen Satz von logischen Befehlen:

AND

Bitweise Berechnung eines logischen UND

Wahrheitstabelle	0	1
0	0	0
1	0	1

OR

Bitweise Berechnung eines logischen ODER: mindestens ein Operand ist 1

Wahrheitstabelle	0	1
0	0	1
1	1	1

Logische Befehle XOR und NOT

XOR

Bitweise Berechnung eines logischen XOR

Genau ein Operand ist =1

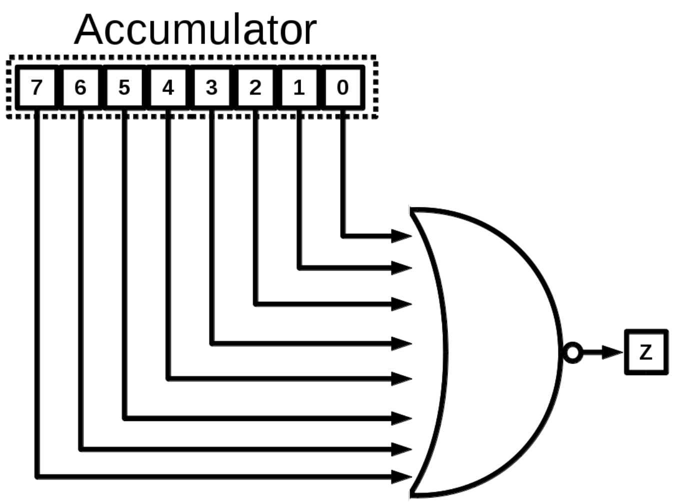
Wahrheitstabelle	0	1
0	0	1
1	1	0

NOT

- Bitweise Berechnung eines logischen NOT
- Dieser Befehl ist nicht bei allen Prozessoren vorhanden, da er durch ein XOR mit -1 erzeugt werden kann

Das Zero-Flag (Z)

- Das Null-Flag zeigt an, ob das Ergebnis einer Operation gleich 0 war.
- Dieser Sonderfall als Ergebnis einer Berechnung ist insbesondere für die Prüfung von Schleifenbedingungen und anderen bedingten Sprüngen von Bedeutung. Diese müssen aufgrund ihres häufigen Auftretens besonders effizient erfolgen können. Mittels des Null-Flags kann diese Prüfung mit minimalem Aufwand durchgeführt werden.
- Berechnung: OR aller Ergebnis-Bits, Invertiert
- Ergebnisse stehen bei primitiven Rechnern im Akkumulator, deshalb alternative Darstellung des Zero-Flags:
  ACC_Z



Das Negativ-Flag (N)

Das Negativ-Flag (auch Sign-Flag) gibt an, ob der Wert (eines Ergebnisses, einer Berechnung) negativ ist.

Wenn das oberste Bit (MSB) 1 ist, ist der Wert negativ, und das Flag wird gesetzt.

Alternative Darstellung: ACC_{15} (bei 16 Bit Wortbreite)

Multiplikation mit 2

- Die Multiplikation mit 2 entspricht im binären Zahlensystem einer Schiebeoperation nach links (wie im Dezimalsystem die Multiplikation mit 10).
- Eine Schiebeoperation füllt die niederwertigen Positionen mit 0 auf.
- Die Informationen in den hochwertigen Bits gehen verloren, bis auf den letzten Übertrag, der im Carry-Flag gespeichert werden kann.