

Rekursives Schaltungsdesign

Zahlendarstellungen

- Was sind Zahlen?
- Wie stellt man Zahlen dar?
- Was macht man mit Zahlen?

Mmmh, das sind eigentlich mathematische und philosophische Fragen, denen wir hier nicht bis ins Detail auf den Grund gehen wollen. Aus praktischen Gründen - wir wollen nur wenig Zeit verwenden, um Geld zu verdienen und damit viel Zeit haben, um es wieder auszugeben - und da wir schon einiges in der Schule gelernt haben, vereinbaren wir folgendes:

- Zahlen sind etwas, was wir hinschreiben können, d.h. sie sind noch da, auch wenn wir schon weg gegangen sind.
- Wir können Zahlen mithilfe von vorher vereinbarten Regeln manipulieren, d.h. recht mechanische Operationen auf Mengen von Zahlen ausführen, und die Ergebnisse dann wieder als Zahlen interpretieren.

Wie wäre es, jeder Zahl ein Symbol zuzuordnen und die Manipulationsregeln in großen Tabellen zu vereinbaren?

Das käme uns wie Chinesisch vor und das Umgehen mit Zahlen wäre eine reine Sisyphus-Arbeit, also nicht genau das, was wir wollen.

Natürliche Zahlen

1. Eins ist eine natürliche Zahl.
2. Eins ist nicht Nachfolger einer natürlichen Zahl.
3. Verschiedene natürliche Zahlen haben verschiedene Nachfolger.
4. Jede natürliche Zahl hat einen Nachfolger.
5. Enthält eine Menge von natürlichen Zahlen die Eins und zu jeder Zahl in der Menge auch deren Nachfolger, so enthält die Menge alle natürlichen Zahlen.

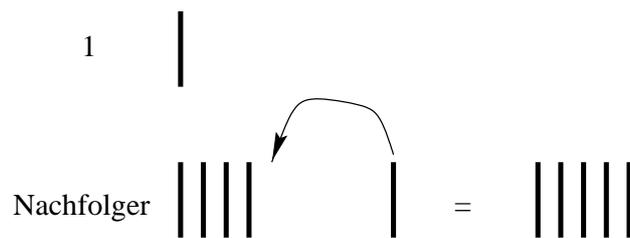
Das sind die berühmten Peano-Axiome. Wenn wir die glauben, dann glauben wir an die natürlichen Zahlen.

Darstellungsmöglichkeiten für die natürlichen Zahlen (mit der Null)

- Unär-Darstellung
- Binär-Darstellung
- Modulo-Darstellung

Unär-Darstellung

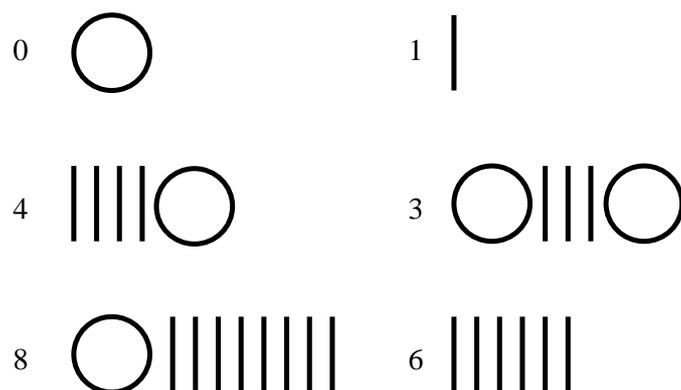
Vereinbaren wir als Symbol für die Eins einen Strich, z.B. auf einem Blatt Papier. Den Nachfolger einer Zahl, d.h. einer Reihe von Strichen, erhalten wir dadurch, dass wir noch einen Strich hintendran malen.



Wir haben nun aber keine Null, d.h. kein Symbol, welches wir hinschreiben können, wenn wir eigentlich nichts hinschreiben wollen.

Vereinbaren wir für die Null einen Kringel.

Wir können dann Zahlen in Unär-Darstellung wie folgt schreiben:



Unär dargestellte Zahlen haben die recht unangenehme Eigenschaft, dass sie sehr lang werden können, was uns ungemein beim Geldverdienen und Geldausgeben stört, da wir die meiste Zeit damit beschäftigt wären, allein die Steuern richtig zu notieren.

Zudem führt das Hinschreiben der Symbole für die Null zu mehrdeutigen Darstellungen für die Zahlen.

Binär-Darstellung

Vereinbaren wir folgendes:

- Wir schreiben die Zahlen von rechts nach links (eben so wie die Araber, aus der Gegend kommt vieles über Zahlen...).
- Wir interpretieren einen Strich nicht als den Nachfolger, sondern als Abkürzung für eine Zahl, die sich ergibt, wenn wir die Stelle, an der der Strich steht mit berücksichtigen. Am Ende summieren wir alles auf.

Um Probleme zu vermeiden, schreiben wir unsere Zahlen in spitze Klammern und fügen als Index die Basis hinzu (falls notwendig). Genauer (jetzt greifen wir gehörig auf das zurück, was wir schon in der Schule gelernt haben):

$$\langle a_{n-1} \dots a_0 \rangle_2 = \sum_{i=0}^{n-1} a_i \cdot 2^i$$

Mit den indizierten a 's bezeichnen wir die Ziffern, die von rechts nach links beginnend mit Null durchnummeriert sind.

Insbesondere bedeutet dies, dass wir die Zahlen (z.B. als Geld) nicht mehr einfach wie eine Menge Steine in einem Sack mit uns herum tragen können, da es jetzt auf die Anordnung der Ziffern ankommt.

***p*-när-Darstellung**

Das Ganze klappt auch mit mehr Symbolen (also Fast-Chinesisch ...). Üblicherweise benutzen wir neben dem Dualsystem auch das Zehnersystem, das Hexadezimalsystem, und das Oktalsystem mit den folgenden Symbolen für die Ziffern:

Name	Basis	Alphabet (Ziffern)
Binärzahlen	2	0,1
Oktalzahlen	8	0,1,2,3,4,5,6,7
Dezimalzahlen	10	0,1,2,3,4,5,6,7,8,9
Hexadezimalzahlen	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Damit das ganze funktioniert, muss eine Zahl eindeutig in Ziffern zerlegbar sein. Ferner sei bemerkt, dass diese *p*-näre Zahlendarstellung eindeutig ist, d.h. zu gegebener Basis (und gewählten Ziffern) hat jede natürliche Zahl genau eine Zahlendarstellung zu dieser Basis. (Bemerkung: Dies müsste man beweisen.)

Die Zahl berechnet sich dann analog wie bei der Binär-Darstellung mit folgender Formel, wobei *b* die Basis angibt:

$$\langle a_{n-1} \dots a_0 \rangle_b = \sum_{i=0}^{n-1} a_i \cdot b^i$$

Beachte: wir müssen dazu schon wissen, was Addieren und Multiplizieren ist, und was die abkürzende Schreibweise mit Exponenten bedeutet. Das wissen wir aber bereits aus der Schule (oder wir machen es wie die Ägypter mit Steinen). Weiterhin stehen die a 's links vom Gleichheitszeichen für Ziffern, rechts dagegen für Zahlen! Diesen feinen kleinen Unterschied wollen wir allerdings nicht vertiefen.

Natürlich schreiben wir in Zukunft Zahlen im Zehnersystem wie gewohnt ohne spitze Klammern und Index und unterscheiden nicht zwischen der Darstellung der Zahl und der dargestellten Zahl. Das mit der 10 hat wohl nur den besonderen Grund, dass wir 10 Finger haben.

Was haben wir gewonnen?

Betrachten wir die Längen der Darstellungen:

Für eine natürliche Zahl n ist die Länge der Darstellung von n in Unär-Darstellung gerade gleich n , in Darstellung zur Basis b , also $\langle a \rangle_b$, hingegen lediglich $\lceil \log_b(n+1) \rceil$, wobei wir mit $\lceil r \rceil$ die nächstgrößere ganze Zahl zu r meinen, bzw. gerade r selbst, wenn r bereits eine ganze Zahl ist. Um eine Zahl bis zur Größe $n - 1$ darstellen zu können, sind $\lceil \log n \rceil$ viele Stellen erforderlich.

Also:

- die Zahldarstellung ist wesentlich kürzer geworden,
- allerdings muss die Ordnung der Bits bekannt sein.

Bemerkung: in vielen Programmiersprachen ist die Zahldarstellung zu anderen Basen als 10 ebenfalls vorhanden! (in C schreibt man beispielsweise Hexadezimalzahlen als `0x39A7B` und Oktalzahlen sind durch eine führende Null gekennzeichnet; dies kann leicht zu Verwirrung führen!)

Negative Zahlen

Insbesondere wenn man Schulden hat oder macht, sollte man auch mit negativen Zahlen rechnen können. Außerdem erleichtern sie es, algebraische Strukturen zu definieren, und mit denen zu rechnen (wir wollten ja Algorithmen entwickeln!). Wir erweitern also unsere Vorstellung von natürlichen Zahlen zu den sogenannten ganzen Zahlen, die sowohl die natürlichen als auch die negativen Zahlen umfassen (die Null ist natürlich auch dabei, da wir diese bereits zu den natürlichen Zahlen zählen).

Darstellungsmöglichkeiten für die ganzen Zahlen

- mit Betrag und Vorzeichen
- im Zweier-Komplement (im Komplement zur Basis)
- mit Vorspannung (bias)

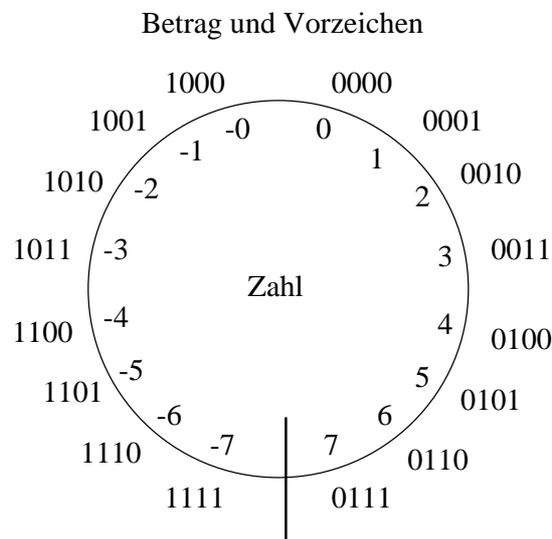
Negative Zahlen mit Betrag und Vorzeichen

Wir könnten natürlich für das Vorzeichen ein neues Symbol einführen (so wie wir es aus der Schule gewohnt sind, z.B. ein Minus-Zeichen zu verwenden), allerdings können wir auch die Darstellung mit Strichen und Kringeln neu interpretieren.

Das linke Symbol in einer Zeichenreihe gibt an,

- wenn es ein Strich ist, dass die Zahl negativ ist,
- und wenn es ein Kringel ist, dass die Zahl positiv ist.

Veranschaulichung der Darstellung durch einen Zahlenkreis, innen die dargestellte Zahl (in der gewohnten Dezimalschreibweise), außen die Darstellung mit Zeichenfolgen (mit 0en und 1en):



Wir sehen:

- die Null hat jetzt zwei Darstellungen.

Wir schreiben Zahlen mit Betrag und Vorzeichen mit runden Klammern.

Bemerkung: Diese Darstellung funktioniert auch in Unär-Darstellung. Wenn ein Kringel vorhanden ist (egal wo), sagen wir, die Zahl ist negativ, ansonsten positiv. Dann gibt es dort nur eine Null, nämlich wenn nur ein Kringel vorhanden ist.

Negative Zahlen im Zweier-Komplement

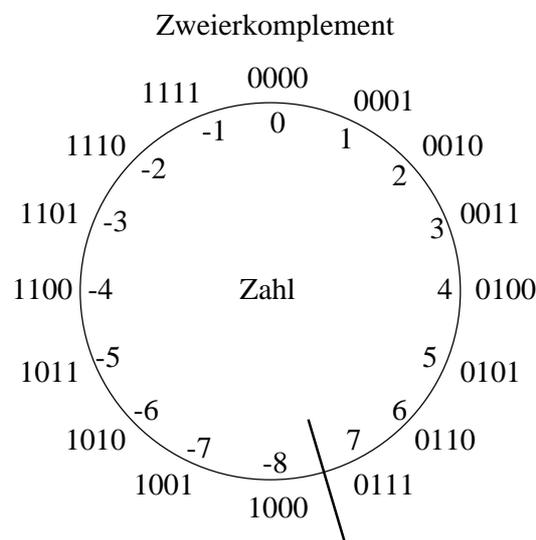
Wir schreiben ebenfalls Zeichenfolgen von 0en und 1en und interpretieren wie folgt:

$$[a_n \dots a_0] = -a_n \cdot 2^n + \langle a_{n-1} \dots a_0 \rangle$$

Beispiel für eine 7-bit Zahl im Zweier-Komplement:

$$\begin{aligned} [1010111] &= \underbrace{-1 \cdot 2^6}_{=-64} + \underbrace{\langle 010111 \rangle}_{=23} \\ &= -41 \end{aligned}$$

Veranschaulichung der Darstellung durch einen Zahlenkreis, innen die dargestellte Zahl (in der gewohnten Dezimalschreibweise), außen die Darstellung mit Zeichenfolgen (mit 0en und 1en):



Wir sehen:

- die Null hat nur eine Darstellung,
- die Darstellung ist nicht symmetrisch zur Null.

Wir schreiben Zahlen im Zweier-Komplement mit eckigen Klammern.

Wie erhalten wir die Darstellung einer Zahl, die wir mit Betrag und Vorzeichen kennen, im Zweier-Komplement?

Trick: wir nehmen das Einer-Komplement des Betrags und addieren noch eins drauf.

Genauer:

Schreiben wir im Folgenden für ein $c_i \in \{0, 1\}$: $\bar{c}_i = 1 - c_i$ und entsprechend $[\bar{c}] = [\bar{c}_n \dots \bar{c}_0]$. Damit hat die Zweierkomplement-Darstellung nun folgende sehr nützliche Eigenschaft:

$$-[c] = [\bar{c}] + 1$$

Das rechnen wir mal nach:

Wir zeigen $[\bar{c}] = -[c] - 1$.

$$\begin{aligned}
 [\bar{c}] &= [\bar{c}_n \dots \bar{c}_0] \\
 &= -\bar{c}_n \cdot 2^n + \langle \bar{c}_{n-1} \dots \bar{c}_0 \rangle \\
 &= -\bar{c}_n \cdot 2^n + \sum_{i=0}^{n-1} \bar{c}_i \cdot 2^i \\
 &= -(1 - c_n) \cdot 2^n + \sum_{i=0}^{n-1} (1 - c_i) \cdot 2^i \\
 &= -2^n + c_n \cdot 2^n + \sum_{i=0}^{n-1} 2^i - \sum_{i=0}^{n-1} c_i \cdot 2^i \\
 &= -2^n + c_n \cdot 2^n + 2^n - 1 - \langle c_{n-1} \dots c_0 \rangle \\
 &= -[c_n \dots c_0] - 1
 \end{aligned}$$

Wir haben hier schon eine Menge von Regeln zum Manipulieren der Zahlen und Formeln benutzt, zum Glück kennen einige davon schon aus der Schule und andere können wir zur Not in einer Formelsammlung nachlesen.

Bemerkung: Zahlen im Zweier-Komplement sind nicht mehr so einfach miteinander zu vergleichen.

Die Methode der Komplementbildung kann man auch auf p -när dargestellte Zahlen erweitern. Man spricht dann z.B. vom Zehner-Komplement und vom Neuner-Komplement anstelle des Zweier- und Einer-Komplements.

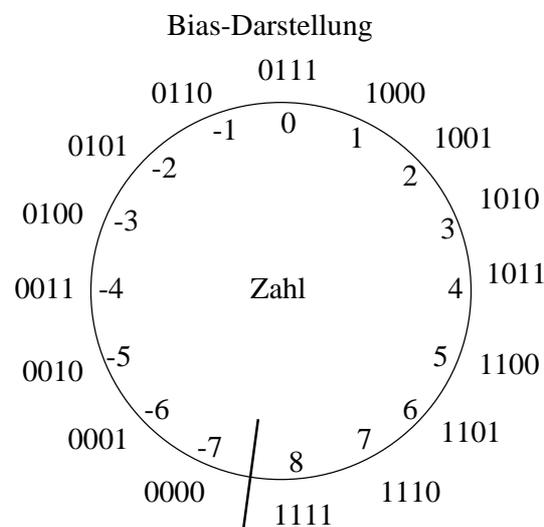
Negative Zahlen in Bias-Darstellung

Wir spannen die Darstellung durch die Addition eines Wertes vor, d.h. wir erhalten die Zahl, indem wir diesen Wert wieder subtrahieren.

$$|a_n \dots a_0| = \langle a_n \dots a_0 \rangle - (2^n - 1)$$

Wir notieren Zahlen in Bias-Darstellung durch Klammerung mit senkrechten Strichen.

Veranschaulichung der Darstellung durch einen Zahlenkreis, innen die dargestellte Zahl (in der gewohnten Dezimalschreibweise), außen die Darstellung mit Zeichenfolgen (mit 0en und 1en):



Zahlen in Bias-Darstellung sind wieder einfach miteinander zu vergleichen.

Modulo-Darstellung

Wir kennen die Primzahlen, das waren die Zahlen, die wir nur durch Eins und durch sich selbst ohne Rest teilen konnten.

Können wir die Primzahlen für eine eigenartige Darstellung der Zahlen verwenden?

Arbeiten wir mit einem Beispiel. Nehmen wir uns eine Menge von Primzahlen, z.B. 2,3,5 und 7.

Wir notieren eine Zahl im Intervall $[0:2*3*5*7-1]=[0:209]$, d.h. von 0 bis zum Produkt aller Primzahlen minus 1, durch die Angabe der Reste, wenn wir die Zahl durch die Primzahlen teilen. Dabei verwenden wir geklammerte Zeichenfolgen mit Kommas zum Abtrennen der Blöcke.

Z.B: 68 ist (5,3,2,0)

Der umgekehrte Schritt ist ein wenig komplizierter. Wir benötigen den euklidischen Algorithmus zum Auffinden des größten gemeinsamen Teilers und einiges Wissen über diophantische Gleichungen. Dies wollen wir hier nicht weiter ausführen.

Machen wir hier nur ein Beispiel zum Rechnen in dieser Zahlendarstellung.

68	(5, 3, 2, 0)
+37	+(2, 2, 1, 1)
= 105	= (0, 0, 0, 1)

Dabei haben wir stellenweise addiert und jeweils wieder den Rest zur bestimmenden Primzahl notiert.

Und das Ganze in Binär-Darstellung für die Komponenten:

68	(101, 011, 10, 0)
+37	+(010, 010, 01, 1)
= 105	= (000, 000, 00, 1)

Wir wollen diese Umrechnung hier nicht weiter vertiefen, da wir sie im Folgenden nicht brauchen. Wir wollen nur festhalten: es ist nicht ganz einfach.

Allerdings sind drei Dinge zu bemerken:

- Die Null ist leicht umzurechnen.
- Wir rechnen immer modulo der Primzahlen, verlassen also nie den Bereich des Intervalls.
- Die Überträge laufen nicht von einem Block in den nächsten, was schnelle Additionen ermöglicht.

Da mussten wir eine Menge tun, damit wir ordentlich mit Zahlen handtieren können und keine riesigen Zahldarstellungen mit uns herum tragen müssen. Fassen wir nochmal zusammen:

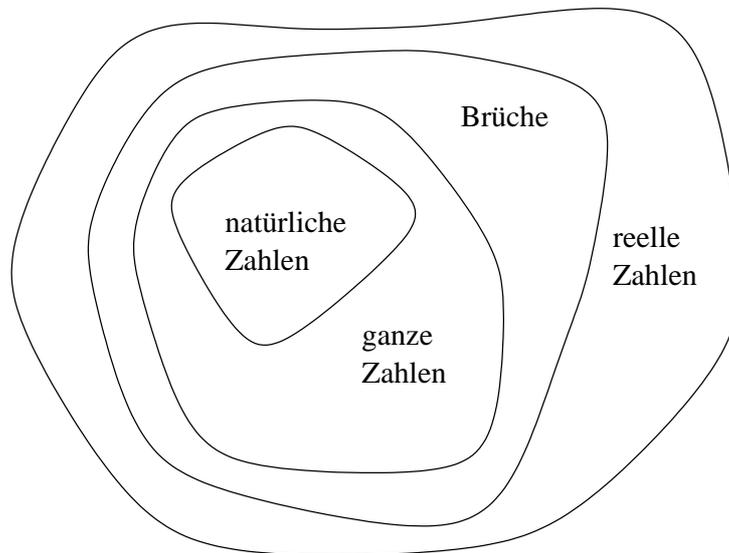
- An die Existenz der Zahlen mussten wir glauben.
- Ein nützlicher - weil kurzer - Glaubenssatz (oder auch Axiom) für natürliche Zahlen waren die Peano-Axiome.
- Das Hinschreiben von Nichts, also der Null, ist höchst geschickt.
- Die Einführung der negativen Zahlen erleichtert wieder das Leben, da wir schöne algebraische Strukturen, d.h. wohldefinierte Regeln zum Manipulieren der Zahlen, recht einfach handhaben können.
- natürliche Zahlen können wir darstellen in:
 - Unär-Darstellung (Sack von Steinen)
 - Binär-Darstellung (Position spielt eine Rolle)
 - (p-när-Darstellung, z.B. im Zehner-System)
 - Modulo-Darstellung (wozu kann man die gebrauchen?)
- ganze Zahlen können wir darstellen:
 - mit Betrag und Vorzeichen (Vorsicht: 2 Nullen)
 - im Zweier-Komplement (Vorsicht: nicht symmetrisch)
 - in Bias-Darstellung

Folgende Tabelle zeigt einige Beispiele der verschiedenen Darstellungen:

7	1111111	Unär-Darstellung
71	$\langle 1000111 \rangle$	Binär-Darstellung
71	(01000111)	Betrag und Vorzeichen
71	[01000111]	Zweier-Komplement
71	11000110	Bias-Darstellung
-71	(11000111)	Betrag und Vorzeichen
-71	[10111001]	Zweier-Komplement
-71	00111000	Bias-Darstellung
71	(001, 001, 10, 1)	Modulo-Darstellung

Gleitkommazahlen

Neben den natürlichen Zahlen und den ganzen Zahlen, gibt es noch eine Menge anderer Zahlen:



Rationale Zahlen, d.h. die Brüche, kann man noch genau darstellen, sofern man sich genügend Bits spendiert; dies ist i.A. bei reellen Zahlen jedoch nicht mehr möglich.

$1/2$	0.5
$1/7$	$0.\overline{142857}$
$1/1000$	0.001
$\sqrt{2}$	1.41421356...
π	3.14159...

Dabei bedeutet der Strich, dass dieser Teil der Ziffern unendlich oft wiederholt werden soll (dabei ist $0.\overline{9} = 1$, d.h. einige (eigentlich unendlich viele) Zahlen haben keine eindeutige Darstellung).

Um nun diese Zahlen zumindest näherungsweise darstellen zu können (und nicht wieder ellenlange Folgen von Nullen schreiben zu müssen) schreiben wir die Zahlen mit

- Vorzeichen,
- Mantisse (oder fraction), und
- Exponent.

d.h. wir fixieren die Ziffern um das Komma, um die Zahl mit genügender Genauigkeit schreiben zu können und verschieben die Stelligkeit durch Multiplizieren mit einer Potenz der Basis (dies nennt man gelegentlich auch halblogarithmische Darstellung).

Beispiel mit der Basis 10:

$1/2$	$5 \cdot 10^{-1}$
$1/7$	$1.42857 \cdot 10^{-1}$
$1/1000$	$1.0 \cdot 10^{-3}$
$\sqrt{2}$	$1.41421356 \dots \cdot 10^1$
π	$3.14159 \dots \cdot 10^1$

Als Formel:

$$s e_m \dots e_0 m_n \dots m_0 = (-1)^{\langle s \rangle} \cdot 1.m_n \dots m_0 \cdot 2^{|e_m \dots e_0|}$$

d.h. der Exponent ist in Bias-Darstellung und es wird eine sogenannte versteckte Eins (hidden one) eingefügt.

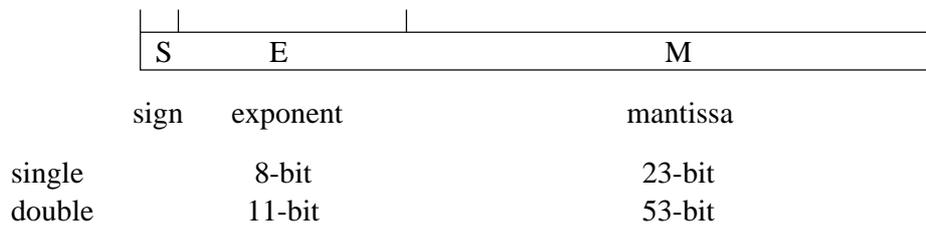
Beachte: Wir haben hier die Bits von links nach rechts nummeriert! Zudem benutzt der IEEE-Standard die Bias-Darstellung für den Exponenten, um so auch Verschiebungen des Kommas nach links bewirken zu können, und trotzdem den Vergleich zweier Exponenten in Funktionseinheiten leicht zu gestalten.

Wir schreiben also mindestens eine von Null verschiedene Stelle vor das Komma! Da wir diese immer schreiben, können wir sie bei der Schreibweise zur Basis 2 auch weglassen und immer bei der Interpretation der Zahl automatisch einfügen.

Problem: damit haben wir keine Null mehr.

Eine Lösung, die fast alle Computer der Welt nutzen (macht halt das Austauschen von Daten einfacher), ist das IEEE-Format.

IEEE-Format



E	M	Bedeutung
nur 0en	nur 0en	Null
nur 0en	nicht nur 0en	denormalisierte Darstellung
nicht nur 1	beliebig	Zahl
nur 1en	nur 0en	unendlich
nur 1en	nicht nur 0en	keine Zahl

Denormalisiert bedeutet, dass man die führende 1 nicht benutzt! Keine Zahl bedeutet, dass es keine gültige Darstellung einer Zahl ist. Natürlich wird das Vorzeichenbit noch entsprechend benutzt.

Übliche Breiten einer sogenannten floating point Zahl sind single precision (8 Bit E, 23 Bit M) und double precision (11 Bit E, 53 Bit M).

Vorteile der Darstellung:

- Es ist ein Standard.
- Man kann integer ALUs für gewisse Gleitkomma-Operationen verwenden (Vergleiche).

Vorsicht: Operationen mit Gleitkommazahlen sind wegen der durchgeführten Rundungsoperationen nicht mehr assoziativ.

$$\begin{aligned}x + (z + y) &= -1.5 \cdot 10^{38} + (1.5 \cdot 10^{38} + 1.0 \cdot 10^1) \\ &= -1.5 \cdot 10^{38} + 1.5 \cdot 10^{38} \\ &= 0.0\end{aligned}$$

$$\begin{aligned}(x + z) + y &= (-1.5 \cdot 10^{38} + 1.5 \cdot 10^{38}) + 1.0 \cdot 10^1 \\ &= 0.0 + 1.0 \\ &= 1.0\end{aligned}$$

Wir halten nochmal fest:

Entscheidend ist also nicht die Zeichenfolge an sich, sondern deren Interpretation durch die angewendeten Operationen!

CVS: \$Id\$

© Arno Formella, November 1998,

formella@cs.uni-sb.de

<http://www-wjp.cs.uni-sb.de/~formella/izfp.html>
