Fehler in der Ganzzahl-Arithmetik von Computern

Zitat aus: Ziegenbalg J.: Der Computer rechnet mit Bytes
mathematik lehren, Heft 7, Dezember 1984
Erhard Friedrich Verlag, Seelze 1984

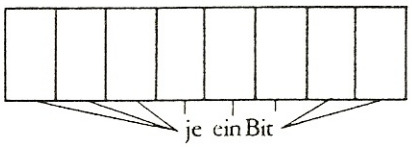
Im Jahre 1996 startete die ESA (European Space Agency) ihre neue Ariane-5 Rakete zum ersten Mal. Man war sich seiner Sache so sicher, dass die Rakete gleich beim ersten Start mit einer Nutzlast von vier Satelliten im Wert von 370 Millionen Dollar ausgestattet wurde. Bereits 37 Sekunden nach dem Start musste die Rakete gesprengt werden. Die Ursache des Fehlers war extrem simpel. Der Messwert für die Beschleunigung wurde als (vorzeichenbehafteter) 16-Bit Integer gespeichert. Als der Wert den maximalen Wert von 32768 übersprang und daraufhin den Wert -32767 annahm, löste das eine Fehlermeldung aus. In der Konsequenz blieben die Steuerdüsen in ihrer Position stecken; die Rakete drehte sich quer zur Flugrichtung und musste gesprengt werden.

Das Szenario der fehlerhaften Ganzzahl-Rechnung wird in schülergemäßer Darstellung auf den folgenden Seiten nachgespielt. Allerdings der übersichtlicheren Darstellbarkeit halber am Beispiel des 8-Bit-Wort-Formats an Stelle von Zahlen im 16-Bit-Wort-Format.

Dass der Fehler 12 Jahre nach Erscheinen des Artikels auftrat, zeigt nur, wie hartnäckig sich dieser (allgemein gut bekannte) Fehler in der Software gehalten hat.

Die Bezeichnungen Bit und Byte sind Maßeinheiten, mit denen man die Größe des Computer-Speichers mißt. Der Elementarbaustein des Speichers von Computern ist das **Bit** (binary digit). Jedes Bit kann genau einen der beiden Zustände 0 oder 1 einnehmen. Mehrere solcher Bits werden zu größeren Feldern zusammengepackt. Ein Paket aus 8 Bits nennt man ein **Byte**. Es ist bei den heutigen Mikrocomputern die am häufigsten benutzte Speichereinheit.

Ein Byte kann man sich also folgendermaßen vorstellen:



Hier sind einige „aufgefüllte" Bytes:

0100 0001
1010 0101
0000 1000
1111 1111
0000 0000

Aufgabe 1: Schreibe weitere Bytes auf! Wie viele verschiedene Zustände kann ein Byte annehmen?

Das Rechenwerk eines Computers kann zu jedem Zeitpunkt nur Felder einer bestimmten festen Länge verarbeiten; zum Beispiel 8 Bit (ein Byte), 16 Bit (zwei Bytes), 32 Bit (vier Bytes), . . . Diese Länge wird als die **Wortlänge** des Computers bezeichnet. Sie ist gemeint, wenn von 8-Bit-Prozessoren, 16-Bit-Prozessoren usw. die Rede ist. Die meisten der heute benutzten Mikrocomputer haben eine Wortlänge von einem Byte.

Aufgabe 2: Wie viele Zustände kann ein 16-Bit-Wort, ein 32-Bit-Wort und ein 64-Bit-Wort annehmen?

Wie wir die Inhalte eines Bytes deuten, liegt bei uns. So können wir das Bitmuster 0100 0001 zum Beispiel als den Buchstaben A aber auch als die Zahl 65 deuten. Im folgenden wollen wir die Inhalte der Bytes als Zahlen deuten. Es liegt nahe, jedes Byte als die Binärdarstellung einer natürlichen Zahl anzusehen:

0000 0000 : 0
0000 0001 : 1
0000 0010 : 2
0000 0011 : 3
0000 0100 : 4
0000 0101 : 5
0000 0110 : 6
0000 0111 : 7
0000 1000 : 8
. . .
0001 0000 : 16
. . .
0010 0000 : 32
. . .
0100 0000 : 64
. . .
1000 0000 : 128
. . .
1111 1111 : 255

Aufgabe 3:

(a) Berechne 1+2, 1+2+4, 1+2+4+8, 1+2+4+8+16, 1+2+4+8+16+32, 1+2+4+8+16+32+64, 1+2+4+8+16+32+64+128

(b) Versuche, eine Gesetzmäßigkeit zu erkennen.

(c) Schreibe die obigen Summen im Zweiersystem auf.

(d) Versuche jetzt noch einmal, den Aufgabenteil (b) zu lösen.

Man kann mit diesen Binärzahlen (Dualzahlen) ganz entsprechend wie mit Dezimalzahlen rechnen. Zum Beispiel: 5+13 = 18

```
  0000 0101
+ 0000 1101
-----------
    1 1 1    (Überträge)
  0001 0010  (Ergebnis)
```

Wenn die Summe 255 überschreitet, entsteht ein Übertrag, der in dem Ergebnis-Wort nicht dargestellt werden kann. Im Computer wird dann ein Übertrag-Schalter auf EIN gesetzt. Man kann diesen Übertrag-Schalter dazu benutzen, um die Addition (durch ein eigenes Programm) auf mehrere Worte auszudehnen.

Läßt man den Überlauf-Schalter außer Acht, so werden die Zahlen 256 und 0 identifiziert. Ebenso die Zahlen 257 und 1, 258 und 2 usw.

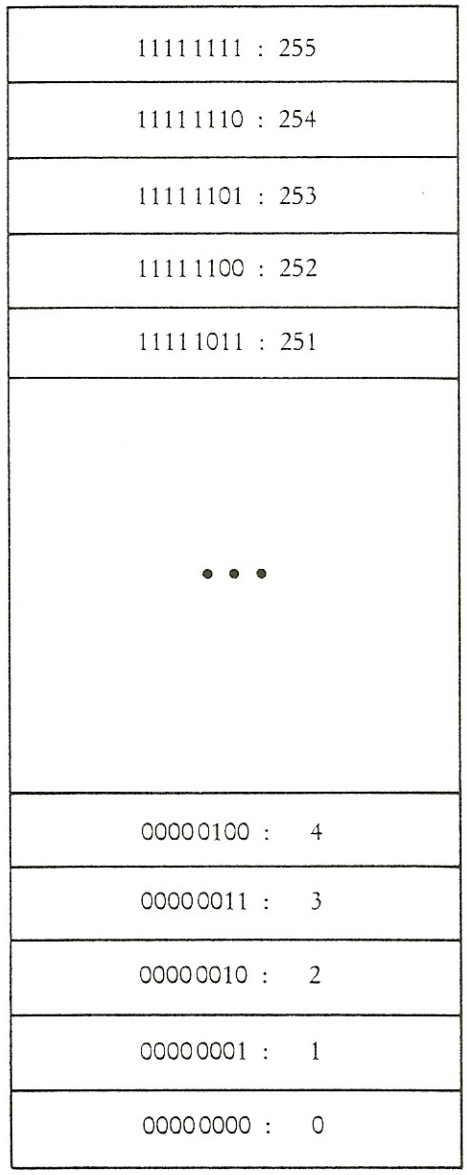
Insgesamt werden alle Zahlen identifiziert, die sich um genau 256 oder ein Vielfaches von 256 unterscheiden.

Stellt man sich die Zahlen von 0 bis 255 wie ein Band mit den Enden 0 und 255 vor, so kann man die Ignorierung des Überlauf-Schalters so deuten, daß das Band an den beiden Enden zusammengeklebt wurde. Man kann die Zahlen auch wie auf einer „Zahlenuhr" anordnen (siehe Abbildung 1).

Addition von 1 bedeutet „ein Feld vorrükken". Es ist also stimmig, daß 255+1 das Ergebnis Null ergibt.

Gegen die bisherige Deutung der Wort-Zahlen ist im Prinzip nichts einzuwenden. Diese Interpretation wird als „unsigned binary" (vorzeichenfreie Binärzahl) bezeichnet.

Die Interpretation der Worte 0000 0000 . . . 1111 1111 als Zahlen von 0 bis 255 ist zwar möglich, ja sogar plausibel, aber nicht zwingend. Wir könnten sie genauso gut als die Zahlen von −128 bis +127 deuten. Dabei sollten natürlich die Worte 0000 0000 bis 0111 1111 (wie vorher) mit den nichtnegativen Zahlen 0 bis 127 übereinstimmen. Aber welchen negativen Zahlen sollten die Worte 1000 0000 bis 1111 1111 entsprechen? Wenn wir uns an die Regel „Addition von 1 bedeutet einen Schritt vorrücken" halten, so muß 1111 1111 als −1 gedeutet werden, denn 1111 1111 + 0000 0001 ergibt 0000 0000 in der Wortarithmetik. Die neue Interpretation der Wort-Zahlen sieht auf der Zahlenuhr also jetzt wie in Abbildung 2 aus.



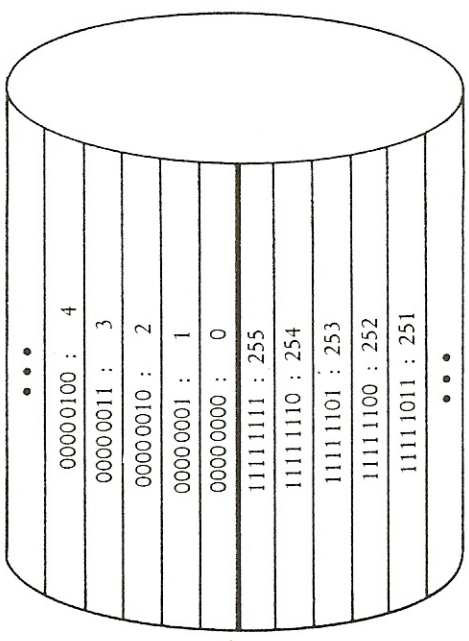
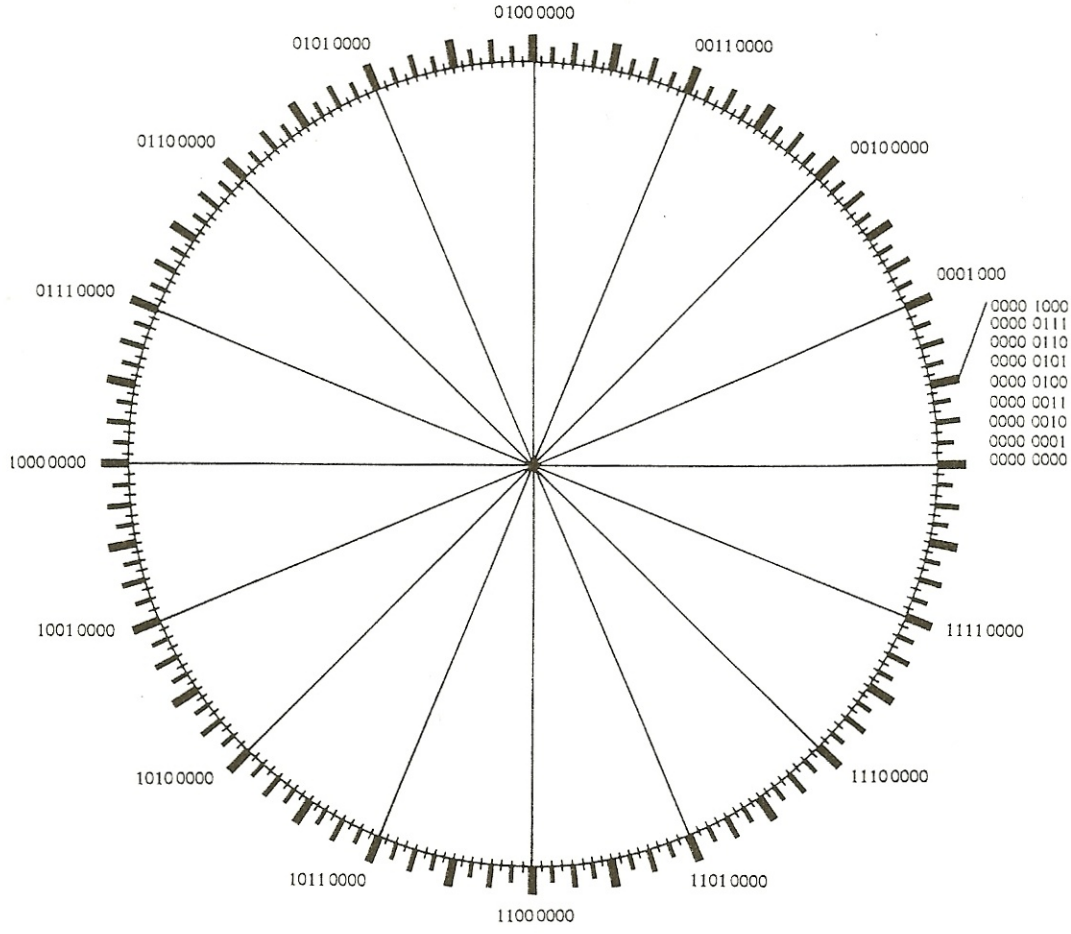
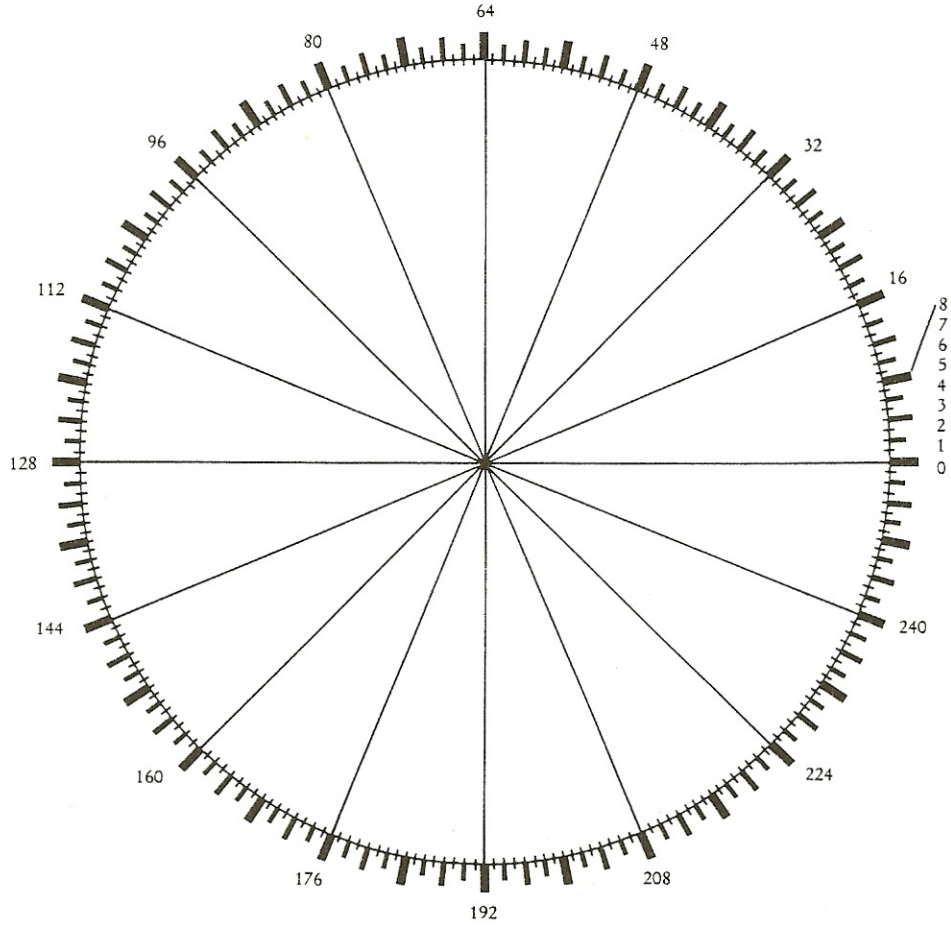


Abbildung 1: Die Computer-Worte als vorzeichenfreie Zahlen auf dem offenen Zahlenband (a), auf dem zusammengeklebten Zahlenband (b) und auf der Zahlenuhr (c).


0100 0000
0011 0000
0010 0000
0001 000
0000 1000
0000 0111
0000 0110
0000 0101
0000 0100
0000 0011
0000 0010
0000 0001
0000 0000
1111 0000
1110 0000
1101 0000
1100 0000
1011 0000
1010 0000
1001 0000
1000 0000
0111 0000
0110 0000
0101 0000


64
48
32
16
8
7
6
5
4
3
2
1
0
240
224
208
192
176
160
144
128
112
96
80

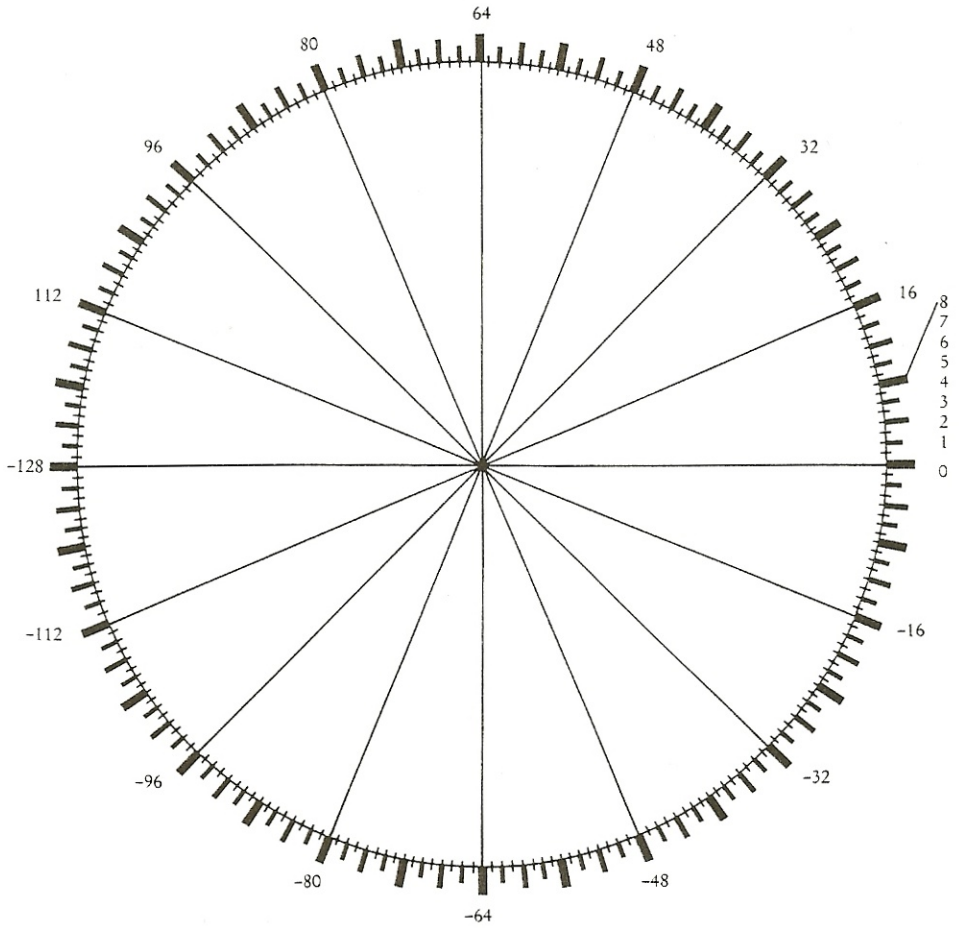


Abbildung 2: Die Computer-Worte als vorzeichenbehaftete Zahlen auf der Zahlenuhr.

Überprüfen wir noch, ob der Übergang von +127 (0111 1111) nach −128 (1000 0000) sinnvoll ist. 127+1 ergibt eigentlich 128. Dieses Ergebnis stimmt aber in unserer vorzeichenbehafteten Wortarithmetik mit −128 überein, da sich diese beiden Zahlen um 256 unterscheiden.

Aufgabe 4: Überprüfe die Übergänge von 1111 1111 zu 0000 0000 und 0111 1111 zu 1000 0000 an weiteren Beispielen einerseits in der Wortarithmetik, andererseits in der Dezimalarithmetik.

Beispiele:	0101 1101 →	93 dezimal:	93
	+ 0011 0110 →	54	54
	1001 0011 →	−109	147

Das Ergebnis ist „in Ordnung", denn der Unterschied zwischen 147 und −109 ist 256.

1110 1111 →	−17	
+ 0010 1010 →	42	(*)
0001 1001 →	25	

(Hier wurde der Übertrag ignoriert.)

Die Deutung der Computer-Worte als vorzeichenbehaftete Dualzahlen ermöglicht nun die Subtraktion von Dualzahlen, denn die Subtraktion der Zahl b von der Zahl a ist nichts anderes als die Addition von −b zu a.

Wenn wir die Summanden im Beispiel (*) vertauschen, können wir die Rechnung entweder als 42 + (−17) oder als 42 − 17 deuten.

Aufgabe 5: Schreibe die Binärdarstellungen der Zahlen 1, −1, 2, −2, 3, −3 usw. in einer Tabelle auf.
Versuche, Gesetzmäßigkeiten zu erkennen.

Schauen wir uns das Paar 43 und −43 etwas genauer an.

43 : 0010 1011
−43 : 1101 0101

Die Binärdarstellung von −43 unterscheidet sich nur geringfügig von dem Wort 1101 0100, das man erhält, wenn man die Nullen und Einsen in der Darstellung von 43 gerade „umdreht". Man bezeichnet die Darstellung 1101 0100 als das **Einerkomplement** von 0010 1011.

0010 1011 (Ausgangs-Byte)
1101 0100 (Einerkomplement)

Dieses Einerkomplement müssen wir noch um 1 erhöhen, um zur richtigen Binärdarstellung von −43 zu kommen.

Aufgabe 6: Überprüfe an weiteren Beispielen, ob man für jede Binärzahl durch den Algorithmus

(1.) Einerkomplement bilden
(2.) 0000 0001 dazuaddieren

stets das Negative der Ausgangszahl erhält.

Den durch (1.) und (2.) beschriebenen Prozeß nennt man auch den Übergang zum **Zweierkomplement** der Ausgangszahl.

Aufgabe 7: Begründe
(a) Die Summe jeder Binärzahl mit ihrem Einerkomplement ergibt stets 1111 1111.
(b) Die Summe jeder Binärzahl mit ihrem Zweierkomplement ergibt stets 0000 0000.
(c) Die Bildung des Zweierkomplements ist identisch mit dem Übergang zum Negativen einer Binärzahl.

Der Übergang zum Einerkomplement ist für den Computer eine leichte Angelegenheit, denn das „Umdrehen" von Bits ist eine seiner elementarsten Grundoperationen. Auch die Addition von 0000 0001 zu einem Computerwort ist für ihn sehr leicht. Der Übergang zum Zweierkomplement ist also eine höchst computergerechte Methode, um das Negative einer Zahl zu finden.

Aufgabe 8: Prüfe, ob die Zweierkomplementmethode zur Subtraktion von Computer-Worten auf die Arithmetik im Dezimalsystem übertragbar ist.

Abschließend noch ein Wort der Warnung:
Man kann die negativen Computer-Worte daran erkennen, daß das höchstwertige Bit (most significant bit MSB) auf 1 gesetzt ist. Deswegen wird dieses Bit auch manchmal als das Vorzeichen-Bit bezeichnet. Dies ist eine etwas unglückliche Bezeichnungsweise, denn sie erweckt die (falsche) Vorstellung, daß man beim Übergang zum Negativen einer Zahl nur das Vorzeichenbit umzudrehen habe.

Impressum:
© Friedrich Verlag, Postfach 10 01 50, 3016 Seelze

Schülerarbeitsheft der Zeitschrift „mathematik lehren", Heft 7/1984 mit Texten und Aufgaben von Peter Bardy, Wolfgang Kroll und Jochen Ziegenbalg.
Redaktion: Jürgen Ricke.
Druck: W. Schröer, Seelze.

Das Schülerarbeitsheft kann auch einzeln bezogen werden. Preis 2,–DM (zzgl. Versandkosten). **Best.-Nr. 20 00 07.** Bei Bestellung im Klassensatz (mindestens 5 Exemplare) wird ein Freiexemplar mitgeliefert.