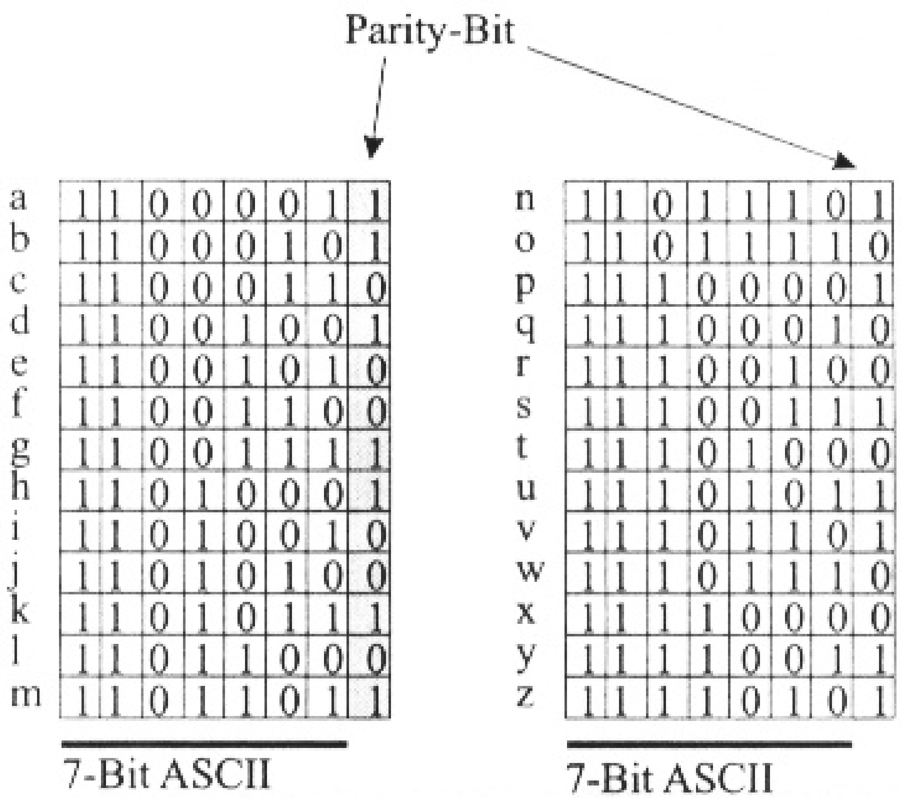
Codes (8a)

(eindimensionaler) Paritätscode:

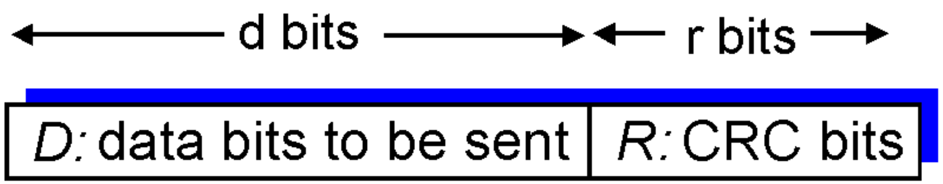


a	1	1	0	0	0	0	1	1
b	1	1	0	0	0	1	0	1
c	1	1	0	0	0	1	1	0
d	1	1	0	0	1	0	0	1
e	1	1	0	0	1	0	1	0
f	1	1	0	0	1	1	0	0
g	1	1	0	0	1	1	1	1
h	1	1	0	1	0	0	0	1
i	1	1	0	1	0	0	1	0
j	1	1	0	1	0	1	0	0
k	1	1	0	1	0	1	1	1
l	1	1	0	1	1	0	0	0
m	1	1	0	1	1	0	1	1

n	1	1	0	1	1	1	0	1
o	1	1	0	1	1	1	1	0
p	1	1	1	0	0	0	0	1
q	1	1	1	0	0	0	1	0
r	1	1	1	0	0	1	0	0
s	1	1	1	0	0	1	1	1
t	1	1	1	0	1	0	0	0
u	1	1	1	0	1	0	1	1
v	1	1	1	0	1	1	0	1
w	1	1	1	0	1	1	1	0
x	1	1	1	1	0	0	0	0
y	1	1	1	1	0	0	1	1
z	1	1	1	1	0	1	0	1

Cyclic Redundancy Check (CRC)

- view data bits, D, as a binary number
- choose r+1 bit pattern (generator), G
- goal: choose r CRC bits, R, such that
  - <D,R> exactly divisible by G (modulo 2)
  - receiver knows G, divides <D,R> by G. If non-zero remainder: error detected!



bit pattern

$D * 2^r \ \text{XOR}\ R$ *mathematical formula*

CRC Example

Want:

$D{\cdot}2^r$ XOR R = nG

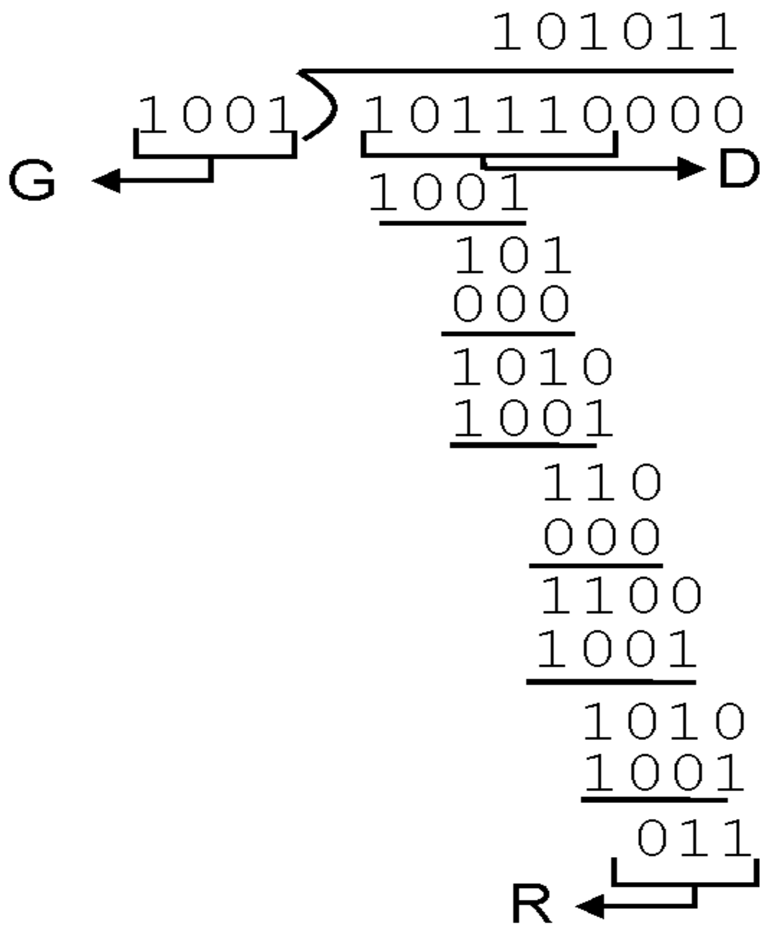
equivalently:

$D{\cdot}2^r$ = nG XOR R

equivalently:

if we divide $D{\cdot}2^r$ by G, the remainder will be R

$$R = \text{remainder}\left[\frac{D\cdot 2^r}{G}\right]$$



Beispiele und Eigenschaften

Examples for (generator) polynomials that have become international standards:

CRC-12	$= x^{12}+x^{11}+x^3+x^2+x^1+1$	(1100000001111)
CRC-16	$= x^{16}+x^{15}+x^2+1$	(11000000000000101)
CRC-CCITT	$= x^{16}+x^{12}+x^5+1$	(10001000000100001)
CRC-32		(100000100110000010001110110110111)

Properties:

They detect

- all single and double errors
- all errors with an odd number of bits
- all burst errors of length r or less
- all burst errors of length greater than r with probability $1 - 0.5^r$
  - ➔ the higher r, the better the protection
- e.g. for CRC-16:
  - 99.997% of 17-bit error bursts
  - 99.998% of 18-bit and longer bursts
- widely used in practice (Ethernet, WLAN, IP)

Computerarithmetik (1)

Fragen:

- Wie werden natürliche Zahlen repräsentiert und konvertiert?
- Wie werden negative Zahlen und Brüche (rationale Zahlen) repräsentiert?
- Wie werden die Grundrechenarten ausgeführt?
- Was ist, wenn das Ergebnis einer Operation größer ist als die größte darzustellende Zahl?

Hauptunterschied zwischen Computer- und menschlicher Arithmetik:

- Computer arbeiten mit einer anderen Zahlendarstellung
- Genauigkeit der sowie Platzbedarf für die Darstellung von Zahlen sind beim Computer endlich und begrenzt.

Rechner speichern die Information (Zahlen) in Einheiten festgesetzter Bitlänge, genannt **Worte.** So dargestellte Zahlen heißen **„Zahlen mit begrenzter Genauigkeit.“**

Computerarithmetik (1a)

Prozessortyp	**Wortlänge (in Bits)**
8085, Z80, 6809	8
8086, 68000	16
80386, 68020	32
Pentium, PowerPC (Sun SPARC, IBM AIX)	32
typischer Mikrocontroller	4
Cray-1 Supercomputer	64

Computerarithmetik (2)

Beispiel für Zahlendarstellung mit unterschiedlichen Basen:

binär

$$1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1$$

$$1 \bullet 2^{10} + 1 \bullet 2^{9} + 1 \bullet 2^{8} + 1 \bullet 2^{7} + 1 \bullet 2^{6} + 1 \bullet 2^{5} + 1 \bullet 2^{4} + 1 \bullet 2^{3} + 1 \bullet 2^{2} + 1 \bullet 2^{1} + 1 \bullet 2^{0}$$

$$1024 + 512 + 256 + 128 + 64 + 0 + 16 + 0 + 0 + 0 + 1$$

oktal

$$3 \quad 7 \quad 2 \quad 1$$

$$3 \bullet 8^{3} + 7 \bullet 8^{2} + 2 \bullet 8^{1} + 1 \bullet 8^{0}$$

$$1536 + 448 + 16 + 1$$

dezimal

$$2 \quad 0 \quad 0 \quad 1$$

$$2 \bullet 10^{3} + 0 \bullet 10^{2} + 0 \bullet 10^{1} + 1 \bullet 10^{0}$$

$$2000 + 0 + 0 + 1$$

hexa-
dezimal

$$7 \quad D \quad 1$$

$$7 \bullet 16^{2} + 13 \bullet 16^{1} + 1 \bullet 16^{0}$$

$$1792 + 208 + 1$$

Computerarithmetik (3)

Kollektion von Zahlendarstellungen mit den 4 verschiedenen Basen:

dezimal	binär	oktal	hexadezimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
7	111	7	7
8	1000	10	8
10	1010	12	A
11	1011	13	B
12	1100	14	C
15	1111	17	F
16	10000	20	10
20	10100	24	14
50	110010	62	32
60	111100	74	3C
80	1010000	120	50
100	11001000	144	64
1000	1111101000	1750	3E8
2989	101110101101	5655	BAD

Computerarithmetik (4)

Tabelle für Umwandlung binär - hexadezimal:

Hexa-dezimal	binär	Hexa-dezimal	binär
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

Beispiel für Konversion einer Dezimalzahl in eine Binärzahl:

Quotient	Rest
1492	
746	0
373	0
186	1
93	0
46	1
23	0
11	1
5	1
2	1
1	0
0	1

1 0 1 1 1 0 1 0 1 0 0

Computerarithmetik (5)

BCD (Binary Coded Decimal):

weitere Möglichkeit der Zahlendarstellung mit Hilfe von nur 2 Ziffern, aber im Dezimalsystem verbleibend.

Prinzip:

Jede Dezimalziffer wird für sich in die entsprechende Binärzahl konvertiert.
(Analogie zum Binärblock mit den Basen 2, 8, 16)

Vorteil:

sehr einfache Konvertierung von dezimaler zu binärer Darstellung

Nachteile:

- komplexere Arithmetik
- verschwenderische (ineffiziente) Ausnutzung der zur Verfügung stehenden Wortbreite und damit des gesamten Speichers

Konsequenz:

Einsatz nur in Applikationen mit sehr geringem Speicherbedarf

Beispiele:

Taschenrechner, Digitaluhr

Computerarithmetik (6)

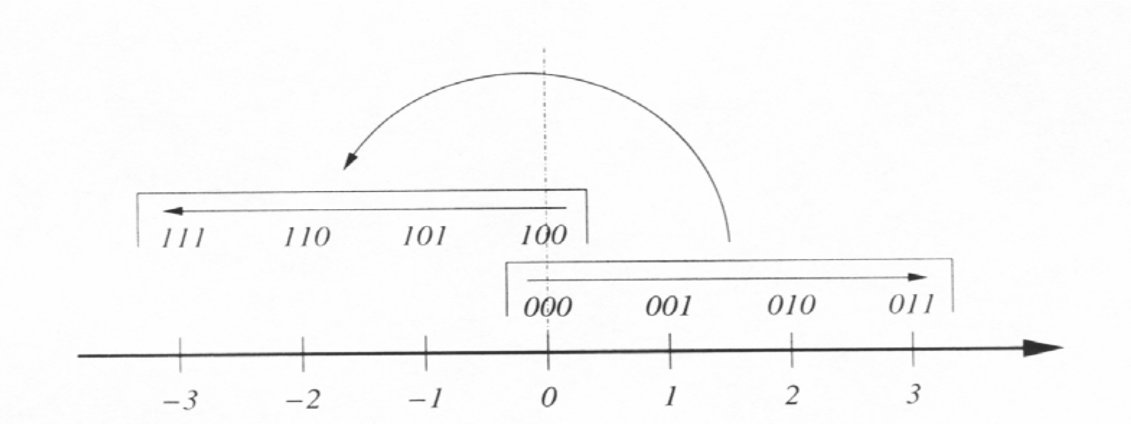
Darstellung ganzer Zahlen (signed numbers)

1. Die Vorzeichen/Betrags - Darstellung (sign and magnitude):

Das höchstgewichtete Bit wird exklusiv für die Angabe des Vorzeichens genutzt.

Sei S ($=d_n$) das Vorzeichenbit und M der Betrag (Größe) einer ganzen Zahl Z, dann ist ihr Wert gegeben durch: $Z =: (-1)^{d_n} M$

Beispiel für n=3:



Der Wertebereich bei einem gegebenen n-bit-Wort liegt im Intervall $[-(2^{n-1}-1), 2^{n-1}-1]$

---> Der Zahlenbereich ist symmetrisch bzgl. des Nullpunkts

---> keine eindeutige Darstellung der Null (-0, +0)