(1) U proizvoljnoj bazi B, B-1 komplement znamenke Z je znamenka $\overline{Z}^{B-1} = B-1-Z$. $Z+\overline{Z}^{B-1}$ daje najveću znamenku baze.

(B-1)-komplement višeznamenkastog broja je broj čije sve znamenke B-1 komplementi znamenaka originalnog broja. Suma ova dva broja je broj čije su sve znamenke B-1.

B-komplement višeznamenkastog broja je broj koji je za 1 veći od (B-1)-komplementa tog broja:

$$\overline{N_k N_{k-1} \cdots N_0}^{B} = \overline{N_k N_{k-1} \cdots N_0}^{B-1} + 1$$

Radimo li s fiksnim brojem znamenaka, X-Y može se izračunati kao $X+\overline{Y}^{B}$, gdje su X i Y n-znamenkasti brojevi.

npr. $5728_{(10)} - 2354_{(10)} = 5728_{(10)} + 7645_{(10)} + 1_{(10)} = (1)3374_{(10)}$

npr. $0101101_{(2)} - 0001100_{(2)} = 0101101_{(2)} + 1110011_{(2)} + 1_{(2)} = (1)0100001_{(2)}$

Koristimo li B-komplement za zapis binarnih brojeva, bit najveće težine definira predznak. Npr. radimo s 8 bitova.

$\cancel{-10_{(10)}}$ $-1010_{(2)} = \overline{00001010}^{(2)} = 11110101 + 1 = 11110110$

negativan broj

Uvjerimo se: dodamo li taj broj broju $15_{(10)}$, dobit ćemo $5_{(10)}$:

$$00001111_{(2)} + 11110110_{(2)} = (1)00000101_{(2)} = 5_{(10)}$$

② Digitalni sklop direktno ne može raditi sa znamenkama baze $\neq 2$. Želimo li raditi u $B > 2$, znamenke treba kodirati. Npr. u bazi 3 možemo raditi ako definiramo sljedeći kod:

z	kôd
0	00
1	11
2	01

Broj $2120_{(3)}$ tada prikazujemo kao 01¦11¦01¦00.

Koliko je tada $\overset{0}{00}\,\overset{2}{01}\,\overset{1}{11}\,\overset{0}{00}\,\overset{2}{01} - \overset{0}{00}\,\overset{1}{11}\,\overset{2}{01}\,\overset{2}{01}\,\overset{1}{11} =$

$$= \overset{0}{00}\,\overset{2}{01}\,\overset{1}{11}\,\overset{0}{00}\,\overset{2}{01} + \overset{2}{01}\,\overset{1}{11}\,\overset{0}{00}\,\overset{0}{00}\,\overset{1}{11} + 1$$

$$= (1)\,\overset{0}{00}\,\overset{0}{00}\,\overset{1}{11}\,\overset{1}{11}\,\overset{1}{11}$$

Zaštitni kodovi

{"upali hlađenje", "ugasi hlađenje"}

{0, 1} ... nema mogućnosti detekcije niti ispravljanja greške

- kôd n-strukog ponavljanja: bit ponovi n puta
- ispravljanje: uvijek prema najbližoj kodnoj riječi
- npr. kôd 5-strukog ponavljanja

  {00000, 11111}: može otkriti do 4 pogreške, ispraviti do 2
- distanca dvije kodne riječi: broj bitova u kojima se razlikuju; broj jedinica u njihovom ex-ili
- minimalna distanca kôda: minimalna distanca između svih parova kodnih riječi; kôd ne sadrži niti jedan par kodnih riječi koje su bliže od te distance d
- ~~takav~~ kôd može otkriti $d-1$ pogrešku a ispraviti $\left\lfloor \frac{d-1}{2} \right\rfloor$

③ Kolika je minimalna distanca kôda n-strukog ponavljanja?

$d=n \Rightarrow$ otkriva $n-1$, ispravlja $\lfloor \frac{n-1}{2} \rfloor$ pogrešaka.

Kolika je minimalna distanca kôda {000000, 010101, 111111}?

$d=3 \Rightarrow$ otkriva 2, ispravlja 1 pogrešku.

⇒ Što je poslano ako je primljeno 011101?

$$d(011101, 000000) = 4$$

$$d(011101, 010101) = 1 \Rightarrow 010101$$

$$d(011101, 111111) = 2$$

<u>Zaštita paritetnim bitom:</u> dodaje jedan bit koji osigurava da je ukupan broj bitova u kodnoj riječi paran (ili neparan). Želimo li parni paritet, računamo ga kao ex-ili svih podatkovnih bitova. Želimo li neparni paritet, računamo ga kao 1 ex-ili svih podatkovnih bitova.

npr. zaštititi 1011 paritetnim bitom uz parni paritet:

$$P = 1 \oplus 0 \oplus 1 \oplus 1 = 1 \Rightarrow \underline{1}|1011$$

npr. uz neparni paritet:

$$P = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0 \Rightarrow \underline{0}|1011$$

distanca ovako dobivenog koda je 2: otkriva jednu pogrešku, ispravlja ∅.

Redundancija kôda: omjer broja zaštitnih bitova r i sume broja zaštitnih bitova (r) i podatkovnih bitova (k): $\frac{r}{r+k} = \frac{r}{n}$ gdje je n ukupni broj bitova

- npr. zaštita paritetnim bitom 9 podatkovnih bitova ima redundanciju: $\frac{1}{1+9} = \frac{1}{10}$ tj. 10%

(4) ~~Kod~~ Hammingov kôd: kôd konstruiran tako da ima minimalnu distancu 3. Koristi više paritetnih bitova koji ispituju različite kombinacije podatkovnih bitova.

Kodnu riječ konstruira tako da na pozicije koje su potencije broja 2 stavlja zaštitne bitove a na preostale pozicije stavlja podatkovne bitove. Zaštitni bit na poziciji 2^i ispituje paritet svih podatkovnih bitova čija pozicija u binarnom zapisu sadrži potenciju 2^i.

npr. Konstruirati Hammingovu kodnu riječ za podatak

01101 0

pozicija

c_1	c_2	d_3	c_4	d_5	d_6	d_7	c_8	d_9	d_{10}
1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	1	1	1
0	0	0	1	1	1	1	0	0	0
0	1	1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0	1	0
$\boxed{0}$	$\boxed{1}$	0	$\boxed{0}$	1	1	0	$\boxed{1}$	1	0

binarni zapis pozicije

$c_1 = d_3 \oplus d_5 \oplus d_7 \oplus d_9$

$c_2 = d_3 \oplus d_6 \oplus d_7 \oplus d_{10}$

$c_4 = d_5 \oplus d_6 \oplus d_7$

$c_8 = d_9 \oplus d_{10}$

neka je to poslano, a neka je primljeno 0110110110.

1) očitati primljene zaštitne bitove: $c_8 c_4 c_2 c_1 = 1010$

2) ponovno zaštititi primljene pod. bitove: $c_8 c_4 c_2 c_1 = 1001$

3) sindrom je ex-ili $\quad : s_8 s_4 s_2 s_1 = 0011_{(3)}$

→ greška je na poziciji 3 → taj bit treba komplementirati.

Koliko zaštitnih bitova trebamo za zaštitu k podatkovnih bitova? Kako sindrom ima isti broj bitova (r), trebamo imati dovoljno različitih sindroma (2^r) za svaku situaciju koja je mogla nastupiti: (1) nema greške, (2) greška je na poziciji 1, (3) greška je na poziciji 2, ..., (n+1) greška je na poziciji n:

$2^r \geq n+1$ tj. $2^r \geq r+k+1$, biramo najmanji r koji to zadovoljava