OSNOVI RAČUNARSTVA

Matematičke osnove rada računara

GOTTFRIED WILHELM VON LEIBNITZ (1646-1716)

- promovisao BINARNI brojni sistem
- **bit** (od eng. **bi**nary digi**t**)

$$a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \ldots + a_0 \times 2^0$$

se predstavlja kao

$$a_{n-1}\, a_{n-2}\, \ldots\, a_0$$

MATEMATIČKE OSNOVE RADA RAČUNARA

- Brojni sistemi
  - Pozicioni i nepozicioni brojni sistemi
  - računarske operacije
  - predstavljanje negativnih brojeva
- Predstavljanje razlomljenih brojeva u računaru
- Kodiranje informacija u računaru
- Kodovi za detekciju i korekciju grešaka

BROJNI SISTEMI

- Pozicioni i nepozicioni brojni sistemi
  - Nepozicioni brojni sistem – Rimski Brojni Sistem:
    - I – jedan
    - V – pet
    - X – deset
    - L – pedeset
    - C – sto
    - D – pet stotina
    - M – hiljadu
    - M – milion (broj crtica iznad slova M označava koliko puta množimo sa hiljadu).

Pozicioni brojni sistemi

- Pozicioni brojni sistemi su sistemi zapisivanja brojeva gde vrednost broja zavisi od:

  - vrednosti svake cifre u broju,

  - pozicije cifre u okviru broju.

- Svaki pozitivan prirodni broj u pozicionom brojnom sistemu se može zapisati u obliku:

$$X = a_n q^n + a_{n-1} q^{n-1} + \dots + a_1 q^1 + a_0 q^0 \qquad (1)$$

gde su:

q prirodni broj koji zovemo osnova brojnog sistema

a_i cifre brojnog sistema

$$X = a_n q^n + a_{n-1} q^{n-1} + ... + a_1 q^1 + a_0 q^0$$

- Binarni brojni sistem:

$q = 2$, $a_i \in \{0,1\}$

- Oktalni brojni sistem:

$q = 8$, $a_i \in \{0,1,2,3,4,5,6,7\}$

- Decimalni brojni sistem:

$q = 10$, $a_i \in \{0,1,2,3,4,5,6,7,8,9\}$

- Heksadecimalni brojni sistem:

$q = 16$, $a_i \in \{0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F\}$

1.1.1 BINARNI BROJNI SISTEM

- Binarni brojni sistem je najčešće korišćeni brojni sistem u digitalnim i računarskim uređajima.
- Predstavljanje informacija sa samo dva znaka najviše odgovara mogućnostima trenutne elektronske tehnologije.
- Smenom ***q=2*** jednačina 1. dobija oblik:

$$X = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_1 2^1 + a_0 2^0$$

PRIMER 1. KONVERZIJA IZ BINARNOG U DECIMALNI BROJNI SISTEM

- Konvertovati **10010110(2)** u decimalni broj.

$$x = a_n 2^n + a_{n-1} 2^{n-1} + ... + a_1 2^1 + a_0 2^0$$

PRIMER 2. KONVERZIJA IZ DECIMALNOG U BINARNI BROJNI SISTEM

- Konvertovati **169(10)** u binarni broj.

169	**: 2**	**=**	**84**	**(1)**	**LSB**
84	**: 2**	**=**	**42**	**(0)**	
42	**: 2**	**=**	**21**	**(0)**	
21	**: 2**	**=**	**10**	**(1)**	
10	**: 2**	**=**	**5**	**(0)**	
5	**: 2**	**=**	**2**	**(1)**	
2	**: 2**	**=**	**1**	**(0)**	
1	**: 2**	**=**	**0**	**(1)**	**MSB**

$$\mathbf{169_{(10)} = 10101001_{(2)}}$$

NEDOSTATCI BINARNOG PREDSTAVLJANJA BROJEVA

- Osnovni nedostatak kod binarnog predstavljanja brojeva je predugačak zapis broja.
- Zbog toga se u računarskim sistemima najčešće koristi heksadecimalni sistem predstavljanja brojeva. Pri tome, iako se brojevi korisniku predstavljaju heksadecimalno, računar i dalje radi sa binarnim brojevima.
- Za predstavljanje brojeva izabran je heksadecimalni brojni sistem zbog jednostavne konverzije brojeva između njega i binarnog brojnog sistema.

1.1.2 Heksadecimalni brojni sistem

- Cifre heksadecimalnog brojnog sistema su:

$$0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ A=10,$$
$$B=11, C=12, D=13, E=14, F=15$$

- Jednačina 1. smenom ***q=16*** dobija oblik:

$$\boldsymbol{X = a_n 16^n + a_{n-1} 16^{n-1} + \ldots + a_1 16^1 + a_0 16^0}$$

PRIMER 3. KONVERZIJA IZ HEKSADECIMALNOG U DECIMALNI BROJNI SISTEM

- Konvertovati **5E3(16)** u decimalni broj.

$$X = a_n 16^n + a_{n-1} 16^{n-1} + \ldots. + a_1 16^1 + a_0 16^0$$

$$\mathbf{5E3}_{(16)} =$$

$$\mathbf{5} \cdot 16^2 + \mathbf{14} \cdot 16^1 + \mathbf{3} \cdot 16^0 =$$

$$5 \cdot 256 + 14 \cdot 16 + 3 \cdot 1 =$$

$$1280 + 224 + 3 = \mathbf{1507}_{(10)}$$

PRIMER 4. KONVERZIJA IZ DECIMALNOG U HEKSADECIMALNI BROJNI SISTEM

- Konvertovati **4328(10)** u heksadecimalni broj.

$$
\begin{array}{llll}
\mathbf{4328} & \mathbf{:16=270} & \mathbf{(8)} \\
\mathbf{270} & \mathbf{:16=16} & \mathbf{(14\ =\ E)} \\
\mathbf{16} & \mathbf{:16=1} & \mathbf{(0)} \\
\mathbf{1} & \mathbf{:16=0} & \mathbf{(1)}
\end{array}
$$

$$\mathbf{4328_{(10)} = 10E8_{(16)}}$$

Primer 5. Konverzija iz binarnog u heksadecimalni brojni sistem

- Konvertovati **$110111110_{(2)}$** u heksadecimalni broj.

Konverzija se vrši grupisanjem po 4 cifre binarnog broja, počevši sa **desne** strane:

$$1110_{(2)} = 14_{(10)} = E_{(16)}$$

$$1011_{(2)} = 11_{(10)} = B_{(16)}$$

$$1_{(2)} = 1_{(10)} = 1_{(16)}$$

$$1\ 1011\ 1110_{(2)} = 1BE_{(16)}$$

PRIMER 6. KONVERZIJA IZ HEKSADECIMALNOG U BINARNI BROJNI SISTEM

- Konvertovati **$3A9_{(16)}$** u binarni broj.

Konverzija se vrši tako što se svaka cifra heksadecimalnog broja konvertuje u 4 cifre binarnog broja:

$$9_{(16)} = 9_{(10)} = 1001_{(2)}$$
$$A_{(16)} = 10_{(10)} = 1010_{(2)}$$
$$3_{(16)} = 3_{(10)} = 0011_{(2)}$$

$$3A9_{(16)} = 0011\ 1010\ 1001_{(2)} = 11\ 1010\ 1001_{(2)}$$

Brojevni sistemi

Osnovi programiranja

Pozicioni brojni sistemi

naziv	**vrednost osnove**	**cifre**
dekadni	10	0,1,2,3,4,5,6,7,8,9
binarni	2	0,1
oktalni	8	0,1,2,3,4,5,6,7
heksadekadni	16	0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F

Pozicioni brojni sistem - primeri

- $1984_{10} = 1 \cdot 10^3 + 9 \cdot 10^2 + 8 \cdot 10^1 + 4 \cdot 10^0 =$
  $1 \cdot 1000 + 9 \cdot 100 + 8 \cdot 10 + 4 \cdot 1 =$
  $1000 + 900 + 80 + 4 = 1984$
- $10011_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 =$
  $1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 16 + 2 + 1 =$
  19
- $12{,}3_{10} = 1 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} =$
  $1 \cdot 10 + 2 \cdot 1 + 3 \cdot 0{,}1 =$
  $10+2+0{,}3 =$
  $12{,}3$

PREDSTAVLJANJE BROJEVA U RAZLIČITIM BROJNIM SISTEMIMA

- Cifre brojnog sistema su: 0 do (baza – 1)
- Primer:
  - binarni (b=2): {0, 1}
  - oktalni (b=8): {0, 1, 2, 3, 4, 5, 6, 7}
  - decimalni (b=10): {0, 1, 2, 3,4 ,5 ,6 ,7 , 8, 9}
  - heksadecimalni (b=16): {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Dekadni u binarni

$(26)_{10} = (?)_2$

	količnik	ostatak	rezultat
26 / 2	= 13	0	
13 / 2	= 6	1	
6 / 2	= 3	0	
3 / 2	= 1	1	
1 / 2	= 0 **kraj postupka konverzije**	1	
			$(11010)_2$

Dekadni u oktalni

$(181)_{10} = (?)_8$

	količnik	**ostatak**	**rezultat**
181 / 8	= 22	5	
22 / 8	= 2	6	
2 / 8	= 0 **kraj postupka konverzije**	2	
			$(265)_8$

Dekadni u heksadekadni

$(181)_{10} = (?)_{16}$

	količnik	**ostatak**	**rezultat**
181 / 16	= 11	5	
11 / 16	= 0 **kraj postupka konverzije**	11 (heksadekadna cifra B)	
			$(B5)_{16}$

PRIMER

Binarni	Oktalni	Decimalni	Heksadecimalni
00000	00	00	00
00001	01	01	01
00010	02	02	02
00011	03	03	03
00100	04	04	04
00101	05	05	05
00110	06	06	06
00111	07	07	07
01000	10	08	08
01001	11	09	09
01010	12	10	0A
01011	13	11	0B
01100	14	12	0C
01101	15	13	0D
01110	16	14	0E
01111	17	15	0F
10000	20	16	10
10001	21	17	11
10010	22	18	12

1.2 Sabiranje binarnih brojeva

- Kod sabiranja binarnih brojeva važe ista pravila kao kod sabiranja decimalnih brojeva.
- Tablica sabiranja:

c_{ul}	a	b	c_{iz}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

PRIMER 7. SABIRANJE BINARNIH BROJEVA

- Sabrati brojeve **$10110111_{(2)}$** i **$10011010_{(2)}$**

$$
\begin{array}{lccccccccc}
C_{ul} & & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\
A & & & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
B & + & & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\hline
A+B & & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}
$$

1.3 ODUZIMANJE BINARNIH BROJEVA

- Tablica oduzimanja:

p_{ul}	a	b	p_{iz}	r
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Primer 8. Oduzimanje binarnih brojeva

- Oduzeti broj **10011010(2)** od broja **10110111(2)**

$$
\begin{array}{lccccccccc}
 & & & & 0 & 0 & & & & \\
A & & 1 & 0 & \cancel{1} & \cancel{1} & 0 & 1 & 1 & 1 \\
B & - & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
\hline
A-B & & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1
\end{array}
$$

1.4 Množenje i delenje binarnih brojeva

- Množenje se obavlja tako što se množenik množi svakom cifrom množioca, a potom se parcijalni proizvodi, pomereni za po jedno mesto u levo, sabiraju.

PRIMER 9. MNOŽENJE BINARNIH BROJEVA

- Pomnožiti brojeve **1100(2)** i **1101(2)**

$$
\begin{array}{rr}
1100\cdot 1101 \quad = & 1100 \\
 & 0000 \\
 & 1100 \\
 & +1100 \\
\hline
 & 10011100
\end{array}
$$

Primer 10. Deljenje binarnih brojeva

- 100010001(2) 1101(2)

Računske operacije – binarni brojni sistem

- Sabiranje:

X	Y	X+Y	Prenos
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Oduzimanje:

X	Y	X-Y	Pozajmica
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

RAČUNSKE OPERACIJE – BINARNI BROJNI SISTEM

- Množenje:

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

- Deljenje:
  - nulom nije dozvoljeno
  - jedinicom - trivijalno

Računske operacije – binarni brojni sistem

```
 11
+11
---
110
```

```
 110
-101
 ---
 001
```

```
   110 x 11
   --------
   110
+ 110
-----------
 10010
```

```
 1001 : 11 = 11
 ----
 100
-011
-----
 0011
-0011
------
 0000
```

RAČUNSKE OPERACIJE – OKTALNI BROJNI SISTEM

```
 447
+652
----
1321
```

```
 54,3
-45,4
 ----
  6,7
```

```
   123 x 21
   --------
   123
+ 246
-----------
  2603
```

```
2603 : 21 = 123
----
 26
-21
----
  50
 -42
 ----
   63
  -63
 -----
    0
```

Računske operacije – heksadecimalni brojni sistem

```
 127
+1AA
----
 2D1
```

```
 2C
-25
----
  7
```

```
   53 x 11
   ---------
   53
+ 53
-----------
  583
```

```
583 : 11 = 53
 ----
 58
-55
----
  33
 -33
 ----
   0
```

```
   1A0 x 13
   --------
   4E0
+ 1A0
-----------
  1EE0
```

KONVERZIJE BROJNIH SISTEMA

- Opšta formula
  - celobrojni deo:

    celobrojni deo (**a**) u novu bazu **b**:

    a : b = r1 i ostatak o1

    r1 : b = r2 i ostatak o2

    r2 : b = r3 i ostatak o3

    ...

    rn : b = 0 i ostatak on

    ----

    rezultat: on ... o3 o2 o1

Konverzije brojnih sistema

- Opšta formula
  - razlomljeni deo:

    razlomljeni deo (**a**) u novu bazu **b**:

    $a \cdot b = c1,r1$ tj. celobrojni deo c1 i razlomljeni deo r1

    $r1 \cdot b = c2,r2$ tj. celobrojni deo c2 i razlomljeni deo r2

    $r2 \cdot b = c3,r3$ tj. celobrojni deo c3 i razlomljeni deo r3

    ...

    $rn \cdot b = cn,0$ tj. celobrojni deo cn i razlomljeni deo 0

    Rezultat: c1 c2 … cn
  - Problem: ako razlomljeni deo ne bude 0

PRIMER

- Broj $37{,}625_{10}$ konvertovati u binarni brojni sistem.

  37 : 2 = 18 i ostatak 1
  18 : 2 = 9 i ostatak 0
  9 : 2 = 4 i ostatak 1
  4 : 2 = 2 i ostatak 0
  2 : 2 = 1 i ostatak 0
  1 : 2 = 0 i ostatak 1
  ----
  rezultat: 100101

Primer

- Razlomljeni deo: 0,625

  $0{,}625 \cdot 2 = 1{,}250$ tj. celobrojni deo 1 i razlomljeni deo 0,250

  $0{,}250 \cdot 2 = 0{,}5$ tj. celobrojni deo 0 i razlomljeni deo 0,5

  $0{,}5 \cdot 2 = 1{,}0$ tj. celobrojni deo 1 i razlomljeni deo 0

- Konačan rezultat: $100101{,}101_2$

KONVERZIJA IZ BINARNOG U OKTALNI I HEKSADECIMALNI BROJNI SISTEM

- Drugačijim grupisanjem bitova:
  - po tri bita za oktalni brojni sistem
  - po četiri bita za heksadecimalni brojni sistem
- Primer:

$01\ 100\ 111_2 = 147_8$

$0110\ 0111_2 = 67_{16}$

HEKSADECIMALNI ←→ OKTALNI

- Preko binarnog brojnog sistema.
- Primer:

$A3_{16} = 1010\ 0011_2$

$10\ 100\ 011_2 = 243_8$

PREDSTAVLJANJE NEGATIVNIH BROJEVA

- Preko znaka i apsolutne vrednosti
  - komplikovan algoritam za sabiranje i oduzimanje
- Preko komplementa
  - jednostavan algoritam za sabiranje i oduzimanje

PREDSTAVLJANJE NEGATIVNIH BROJEVA KOMPLEMENTOM

- Potpuni komplement (u binarnom brojnom sistemu se još zove i komplement dvojke).
- Nepotpuni komplement (u binarnom brojnom sistemu se još zove i komplement jedinice).
- U oba sistema se poslednja cifra koristi za znak broja (pozitivan ili negativan).

Potpuni komplement

- broj x
- n cifara
- baza b
- PotpuniKomplement (x) = $b^{n+1} - x$

Potpuni komplement

- Primer:

znak!

$x = 0010_2 = 2_{10}$

$n = 3$

$b = 2$

$\text{PotpuniKomplement}(2) = 2^{3+1} - 2 =$

$10000_2 - 0010_2 = 1110_2$

znak!

Potpuni komplement

- Primer:

znak!

$x = 1110_2 = -2_{10}$

$n = 3$

$b = 2$

$PotpuniKomplement\ (-2) = 2^{3+1} - (-2) =$

$10000_2 - 1110_2 = 0010_2$

znak!

Sabiranje sa potpunim komplementom

- Pravilo:

  A – B = A + PotpuniKomplement(B) =

  Rezultat + Prenos
- Ako je Prenos = 1 onda je Rezultat korektan.
- Ako je Prenos = 0 onda je rezultat negativan (stvarni rezultat je potpuni komplement od rezultata sa negativnim predznakom).

Primer

- $0101_2 - 0010_2 = 0101_2 + 1110_2 = 10011_2$

prenos

- $0001_2 - 0010_2 = 0001_2 + 1110_2 = 01111_2$

prenos

Stvarni rezultat: - PotpuniKomplement(1111_2) =

$10000_2 - 1111_2 = - 00001_2$

NEPOTPUNI KOMPLEMENT

- broj x
- n cifara
- baza b
- NepotpuniKomplement (x) = $(b^{n+1} - 1) - x$

Nepotpuni komplement

- Primer:

znak!

$x = 0\underbrace{010}_{2} = 2_{10}$

$n = 3$

$b = 2$

$NepotpuniKomplement\ (2) = (2^{3+1} - 1) - 2 =$

$(10000_2 - 0001_2) - 0010_2 = 1101_2$

znak!

NEPOTPUNI KOMPLEMENT

- Primer:

znak!

$x = 1101_2 = -2_{10}$

$n = 3$

$b = 2$

NepotpuniKomplement $(-2) = (2^{3+1} - 1) - (-2) =$

$(10000_2 - 0001_2) - 1101_2 = 0010_2$

znak!

SABIRANJE SA NEPOTPUNIM KOMPLEMENTOM

- Pravilo:

  A – B = A + NepotpuniKomplement(B) =

  Rezultat + Prenos
- Ako je Prenos = 1 onda je

  konačan rezultat = rezultat bez prenosa + 1.
- Ako je Prenos = 0 onda je rezultat negativan (stvarni rezultat je nepotpuni komplement od rezultata sa negativnim predznakom).

PRIMER

- $0101_2 - 0010_2 = 0101_2 + 1101_2 = 1\,0010_2$

prenos

Pravi rezultat: $0010_2 + 1 = 0011_2$

- $0001_2 - 0010_2 = 0001_2 + 1101_2 = 0\,1110_2$

prenos

Stvarni rezultat: -NepotpuniKomplement(1110_2) =

$10000_2 - 0001_2 - 1110_2 = -\,00001_2$

Sračunavanje komplementa bez oduzimanja!

- Potpuni komplement: invertovati sve bitove i dodati 1.
  - Primer: 0010_2 => $1101_2 + 1 = 1110_2$
- Nepotpuni komplement: invertovati sve bitove.
  - Primer: 0010_2 => 1101_2

PREDSTAVLJANJE RAZLOMLJENIH BROJEVA U RAČUNARU

- U nepokretnom zarezu
  - fiksna pozicija decimalnog zareza.
- U pokretnom zarezu (floating point)
  - brojevi se predstavljaju u obliku: $m \cdot b^e$

m – mantisa

b – baza

e – eksponent

POKRETNI ZAREZ

- Sabiranje odn. oduzimanje - pre sabiranja (oduzimanja) brojevi se svedu na isti eksponent:

  $m1 \cdot b^{e} + m2 \cdot b^{e} = (m1 + m2) \cdot b^{e}$

- Množenje, odn. deljenje:

  $(m1 \cdot b^{e1}) \cdot (m2 \cdot b^{e2}) = (m1 \cdot m2) \cdot b^{(e1+e2)}$

POKRETNI ZAREZ

- Normalizovana mantisa: kada je

  $b^{-1} \leq |m| \leq 1$
- Tada je preciznost najveća.
- Pokretni zarez u nekim situacijama nije dovoljno precizan!

KODIRANJE INFORMACIJA U RAČUNARSKOM SISTEMU

- Binarno kodiranje numeričkih informacija.
- Binarno kodiranje alfanumeričkih informacija.

BINARNO KODIRANJE NUMERIČKIH INFORMACIJA

- Sa x bitova moguće je predstaviti 2^x različitih podataka.
- Znači da je potreban broj bitova za neki broj različitih podataka:

$\log_2$ br_podataka

- Primer:

za 10 različitih cifara decimalnog brojnog sistema potrebno je $\log_2 10 = 3{,}32 \rightarrow 4$

Binarno kodiranje numeričkih informacija

- Jedan oblik kodiranja decimalnih cifara upotrebom 4 bita: NBCD kod (Natural Binary Coded Decimal) ili 8421 kod

Decimalna cifra	NBCD kod
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

BINARNO KODIRANJE NUMERIČKIH INFORMACIJA

- Drugi oblik kodiranja decimalnih cifara upotrebom 4 bita:

  XS-3 kod

Decimalna cifra	XS-3 kod
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

Binarno kodiranje alfanumeričkih informacija

- Alfanumerički simboli:
  - numerički simboli (0, 1, ..., 9)
  - slovni simboli (A, B, ..., Z)
  - inteprunkcijski znakovi (, . ; : “ ...)
  - specijalni simboli (#, $, %, ...)
- Standardi:
  - ASCII (American Standard Code for Information Interchange)
  - ISO 8859-1
  - Windows CP 1250
  - Unicode

Kodovi za detekciju i korekciju grešaka

- Koncentrisaćemo se na binarni brojni sistem. Sve informacije će biti kodirane binarno!
- Uzrok pojave grešaka.
- Kodovi za detekciju grešaka
  - u stanju su da detektuju grešku, ali ne i da je koriguju
- Kodovi za korekciju grešaka
  - detekcija i korekcija grešaka

Kodovi za detekciju grešaka

- Najjednostavnije je da se doda još jedan bit tako da ukupan broj jedinica u poruci bude paran ili neparan.
- Primer:
  - originalna poruka: 001101
  - sa dodatnim bitom (uk. br. jedinica paran):
    0011011
  - sa greškom: 0001011
  - vidimo da je došlo do greške pošto je ukupan broj jedinica neparan!
- Greške od više od jednog bita mogu da prođu nedetektovane!

  1111011

Karakter za proveru bloka

b1 b2 b3 b4 p1

b5 b6 b7 b8 p2

p3 p4 p5 p6 p7

- U slučaju greške od jednog bita bilo gde, moguće je detektovati i korigovati grešku:

b1 b2 b3 b4 p1

b5 b6 b7 b8 p2 ←

p3 p4 p5 p6 p7

↑

CRC KOD

- Cyclic Redundancy Character
- Poruka se kao niz bitova deli sa nekim unapred dogovorenim brojem, rezultat se odbacuje a ostatak pri deljenju se doda uz poruku.
- Na prijemnoj strani se primljena poruka deli istim brojem i ostatak se poredi sa primljenim ostatkom.