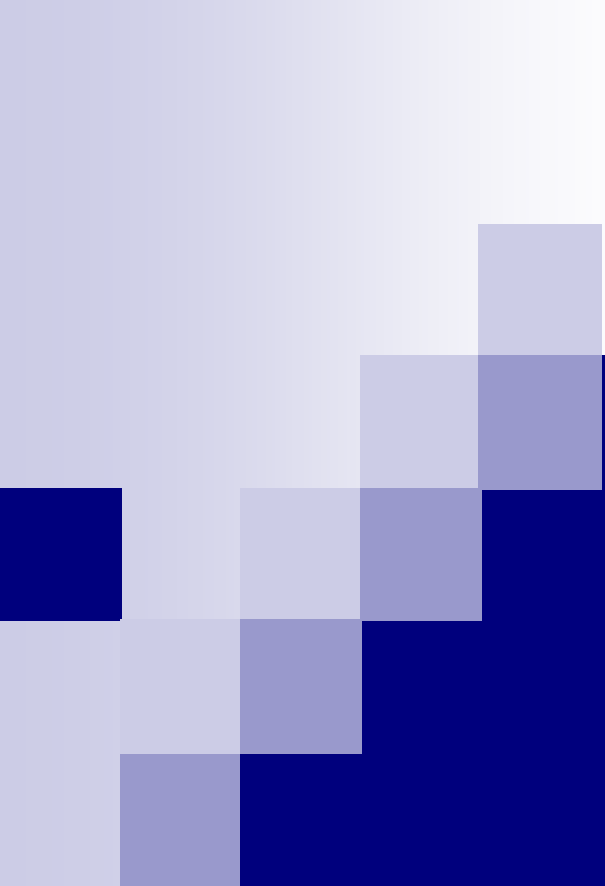




Vhodno-izhodne naprave (VIN)

Predavanja

9. Kodiranja signalov

Robert Rozman

rozman@fri.uni-lj.si

Vsebina

1. Uvod
2. Kodiranje RZ (Return to Zero)
3. Kodiranje NRZ (No Return to Zero)
4. Kodiranje PE (Phase Encoded)
5. Kodiranje RLL (Run Length Limited)
 - Kodiranje 2/3 (1,7) RLL
 - Kodiranje 8b/10b

❑ Vir: 'Communication – Digital encode'

http://www.sharetechnote.com/html/Communication_Digital_Encoding.html

Uvod

- Kodiranje digitalnih podatkov - pretvorba podatkov ali določenega zaporedja znakov, simbolov, abeced itd. v signal za varen prenos podatkov.



Uvod

Zakaj kodiranje?

Spektralna „učinkovitost“

- Digitalni signal **moramo koncentrirati** v določeno, čim ožje frekvenčno območje.

Manjšanje enosmernega odmika („DC offset“)

- Če imamo za bit 1 napetosti $+V$ in za bit 0 napetost $+0V$ (ali $-V$), lahko določeni vzorci (vsi biti so 0 ali vsi biti so 1), **ustvarijo odmik signala** vzdolž komunikacijskih medijev, kar povzroči določene težave.

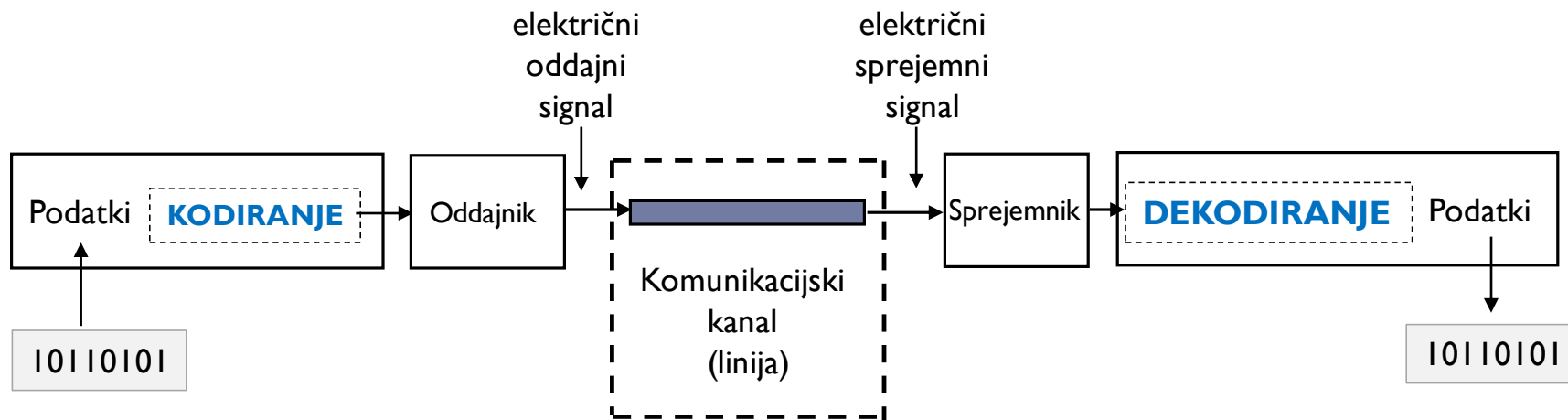
Boljša sinhronizacija sprejemnika (zadostno število prehodov)

- **Določen vzorec signala**, ki nima veliko sprememb napetosti (dolgo zaporedje $+V$ ali $-V$) lahko **oteži sinhronizacijo** na sprejemniku.

Manjšanje občutljivosti na napake (BER vs. SNR)

- **Za zagotavljanje boljše učinkovitosti** v smislu števila napačno prenesenih bitov v časovni enoti (BER - Bit error rate) pri danem razmerju signal-šum (SNR – Signal-to noise ratio).

❑ Model prenosa podatkov med napravami



❑ Enostavna predstavitev enic z visokim in ničel z nizkim nivojem za

- prenos po komunikacijskih kanalih in
- shranjevanje na polprevodniške, magnetne in optične medije.

❑ Kodiranje (ang. encoding) in dekodiranje (ang. decoding)

❑ Sinhronizacija - restavriranje urinega signala (ang. clock recovery)

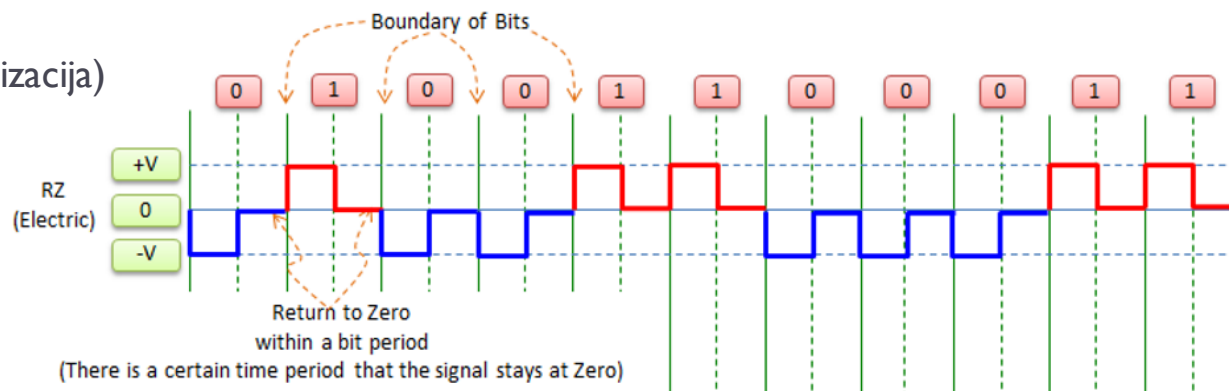
1 Kodiranje RZ (Return to Zero)

❑ Slabosti:

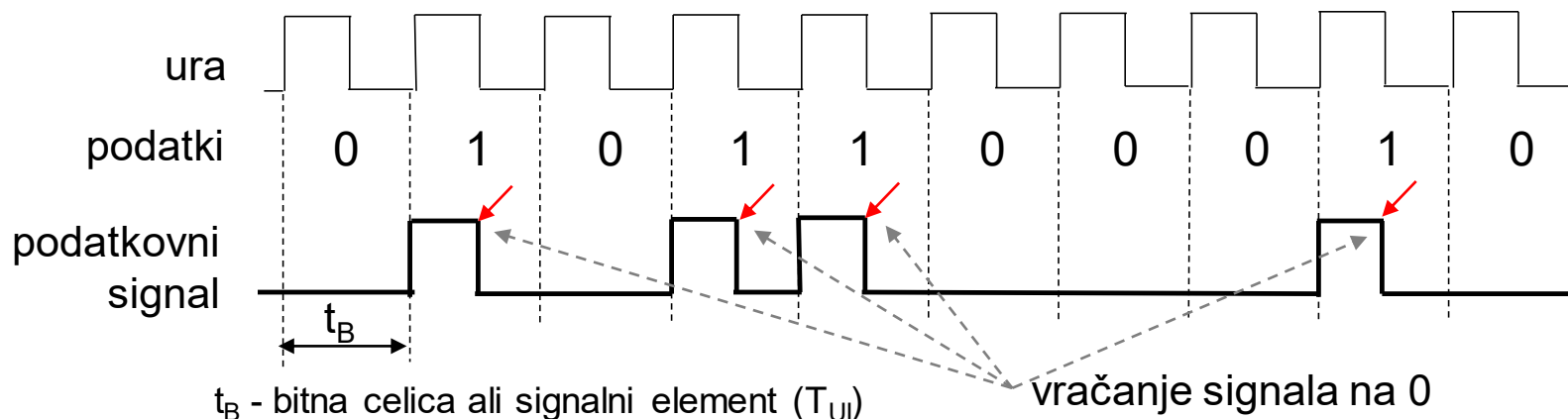
- ❑ DC pomik (daljši nizi 0 ali 1)

❑ Prednosti:

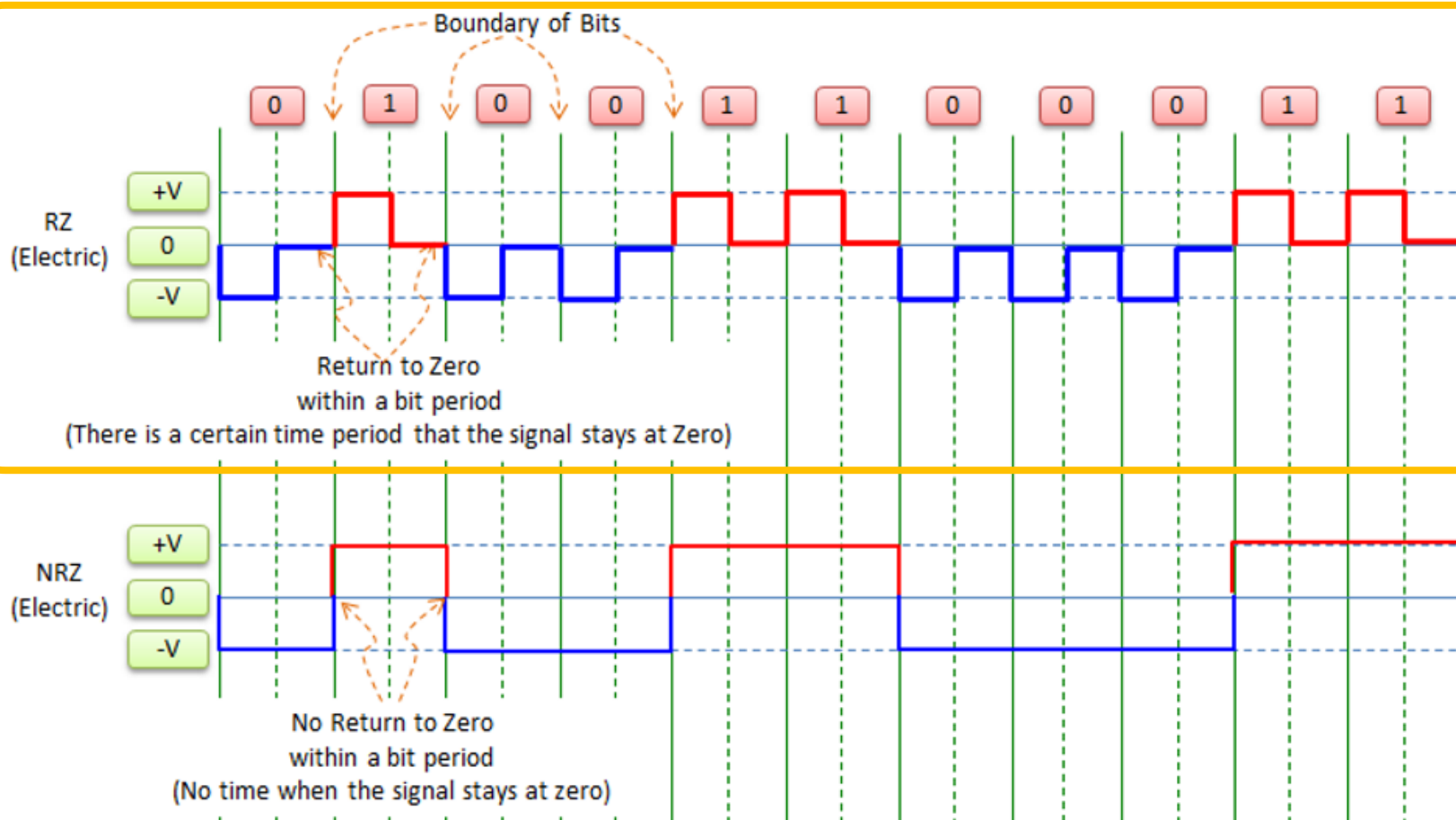
- ❑ Več prehodov (sinhronizacija)



❑ Primer kodiranja za 10-bitni niz (optika), ki ima 4 logične enice → 8 sprememb nivoja



1 Kodiranje RZ (Return to Zero)



❑ Slabosti:

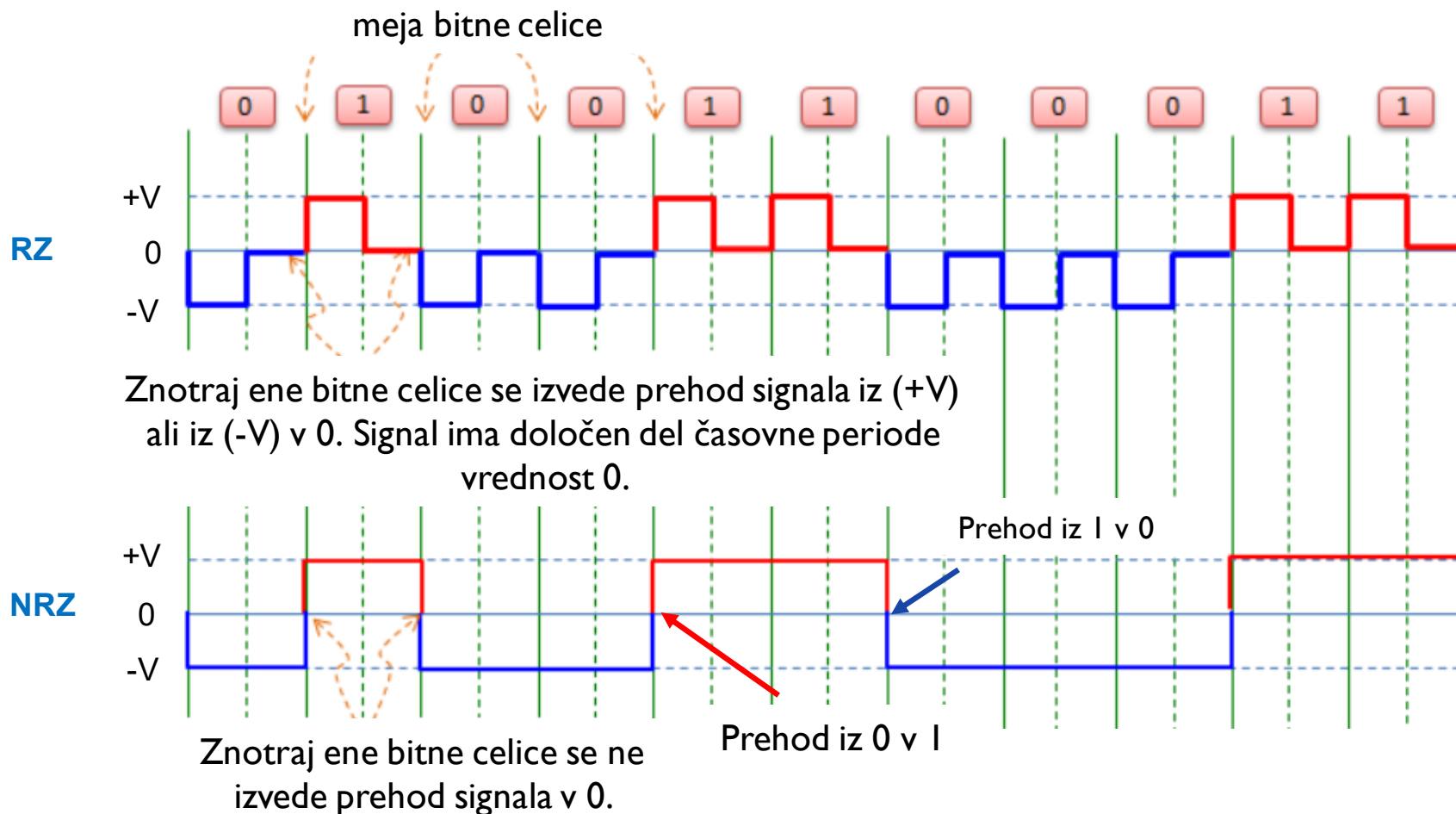
- ❑ DC pomik (daljši nizi 0 ali 1)

❑ Prednosti:

- ❑ Več prehodov (sinhronizacija)

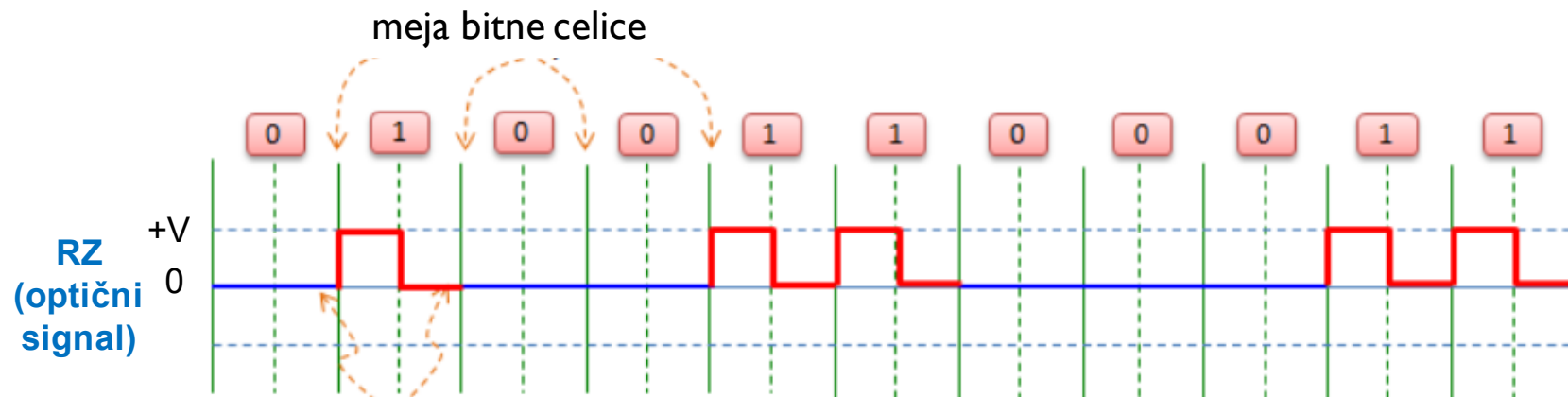
Primer kodiranja RZ, NRZ

- ❑ **Električni signal** - signal plus (+V) označuje bit 1 in minus (-V) označuje bit 0.

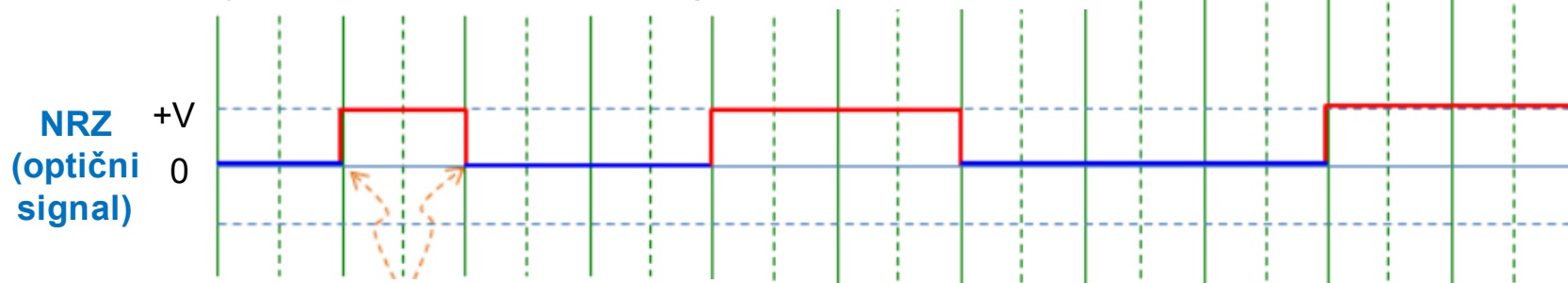


Primer kodiranja – optika: RZ, NRZ

- ❑ **Optični signal** – negativni signal ($-V$) ni možen, zato je eden od bitov preslikan v 0.



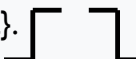
Znotraj ene bitne celice se izvede prehod signala iz (+V) v 0.
Signal ima določen čas znotraj periode vrednost 0.



Znotraj ene bitne celice se ne
izvede prehod signala v 0.

2 Kodiranje NRZ (Non Return to Zero)

□ več vrst kodiranj NRZ (Non Return to Zero)

Code name	Alternate name	Complete name	Description
NRZ(L)	NRZL	Non-return-to-zero level	Appears as raw binary bits without any coding. Typically binary 1 maps to logic-level high, and binary 0 maps to logic-level low. Inverse logic mapping is also a type of NRZ(L) code.
NRZ(I)	NRZI	Non-return-to-zero inverted	Refers to either an NRZ(M) or NRZ(S) code.
NRZ(M)	NRZM	Non-return-to-zero mark	Serializer mapping {0: constant, 1: toggle}.  1 - sprememba nivoja
NRZ(S)	NRZS	Non-return-to-zero space	Serializer mapping {0: toggle, 1: constant}.  0 - sprememba nivoja
NRZ(C)	NRZC	Non-return-to-zero change	

2 Kodiranje NRZ (Non Return to Zero)

- najpogosteje se uporablja **kodiranje NRZI(M)** (Non Return to Zero Invert - Mark)

- Logična 1: sprememba nivoja signala (0 v 1 ali 1 v 0)

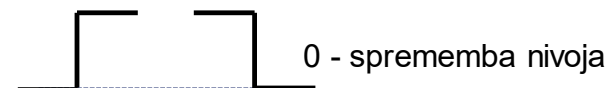
- Logična 0: brez spremembe nivoja signala



- Druga možnost kodiranja **NRZI(S)** – je definirana ravno obratno kot zgoraj

- Logična 1: brez spremembe nivoja

- Logična 0: sprememba nivoja signala (0 v 1 ali 1 v 0)



- Prednosti:

- Med vsemi načini kodiranja je z NRZI mogoče doseči **najboljšo izkoriščenost kanala**.
 - Preprosta realizacija.

- Slabosti:

- Pri zaporedju ničel (ali enic) ni spremembe nivoja signala.
 - Brez sinhronizacijske informacije pri dolgem zaporedju ničel (enic).

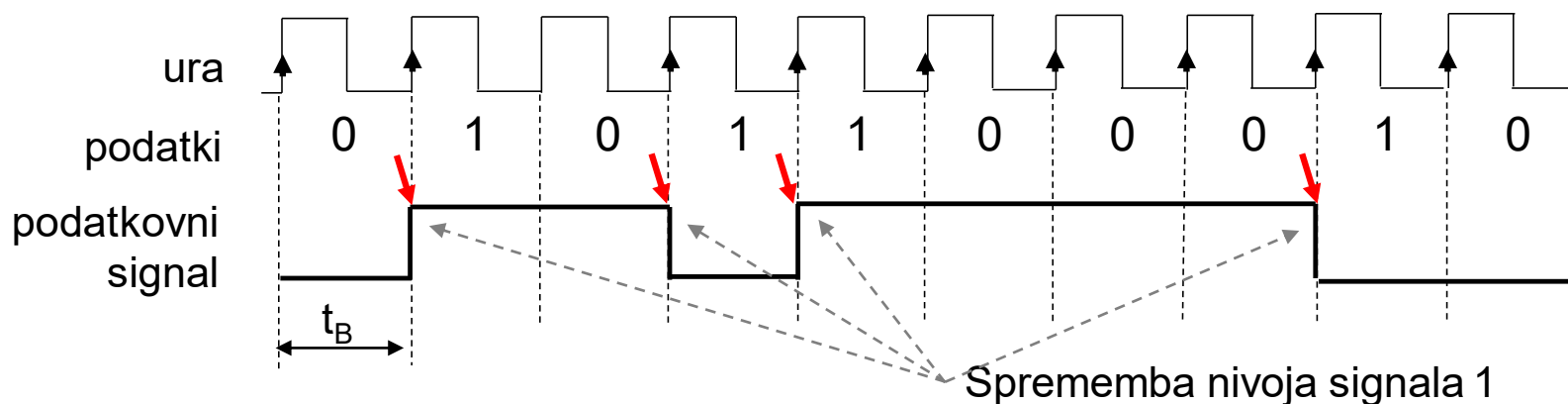
- Uporaba:

- USB 2.0 in HDLC (pri predolgem zaporedju enic se vstavlja ničla)
 - Skupaj s kodiranjem RLL.

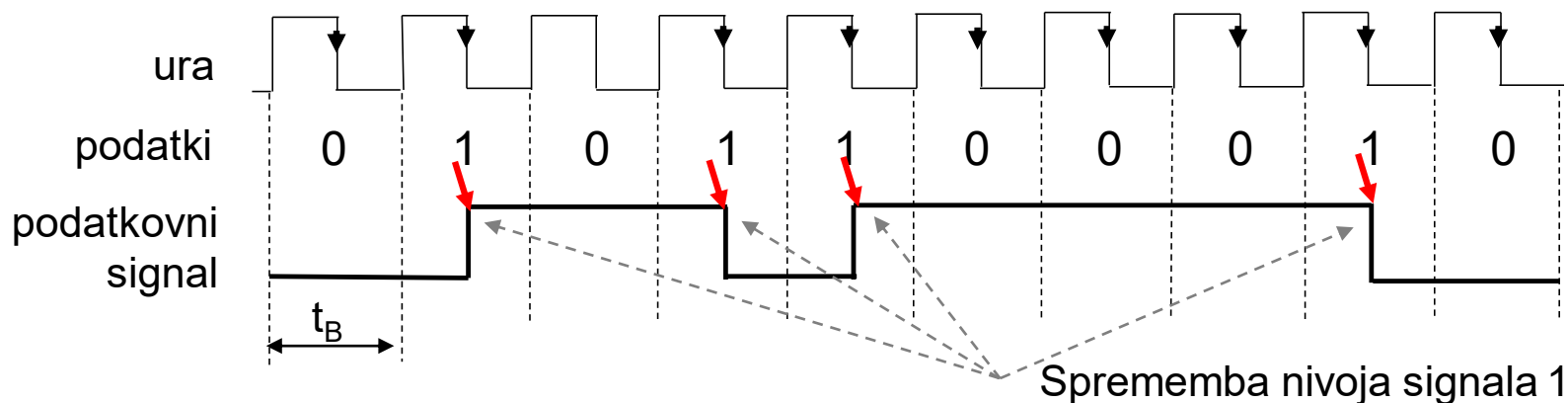
□ Primer kodiranja NRZI za 10-bitni niz, ki ima 4 logične enice → 4 spremembe nivoja

- Sprememba nivoja signala se izvede pri vsakem pojavu bita po vrednosti 1.

a) ob pozitivni fronti ure ($0 \rightarrow 1$)



b) ob negativni fronti ure ($1 \rightarrow 0$)

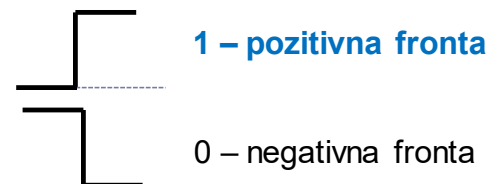


3 Kodiranje PE (Phase Encoded)

- ❑ V to skupino sodi več vrst kodiranj. Njihova skupna lastnost je, da poleg podatkovne informacije vsebujejo tudi **sinhronizacijsko informacijo** (uro).
- ❑ V vsaki bitni celici (t_B) je tako **vsaj ena sprememba** nivoja signala.

❑ **Bi-Phase L (Manchester kodiranje):**

- Logična 1: pozitivna fronta signala
- Logična 0: negativna fronta signala

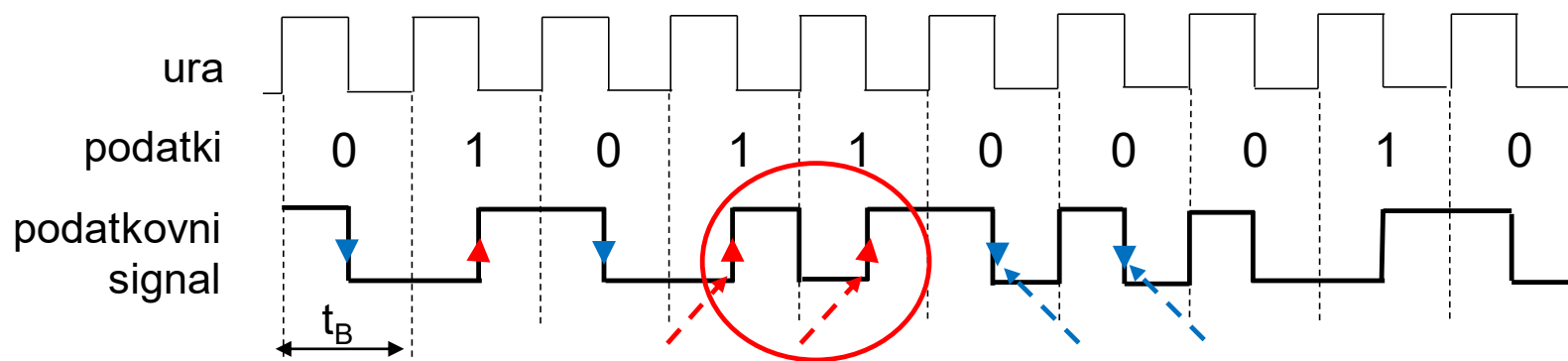


- ❑ Prednosti:
 - Odpravlja obe pomanjkljivosti NRZI kodiranja.
- ❑ Slabosti:
 - Slaba izkoriščenost kanala (dve spremembi nivoja v bitni celici).
- ❑ Uporaba:
 - Ethernet 10BASE-T (10Mb/s IEEE 802.3)

❑ Primer: kodiranje PE (Phase Encoded) – kodiranje ‘Manchester’

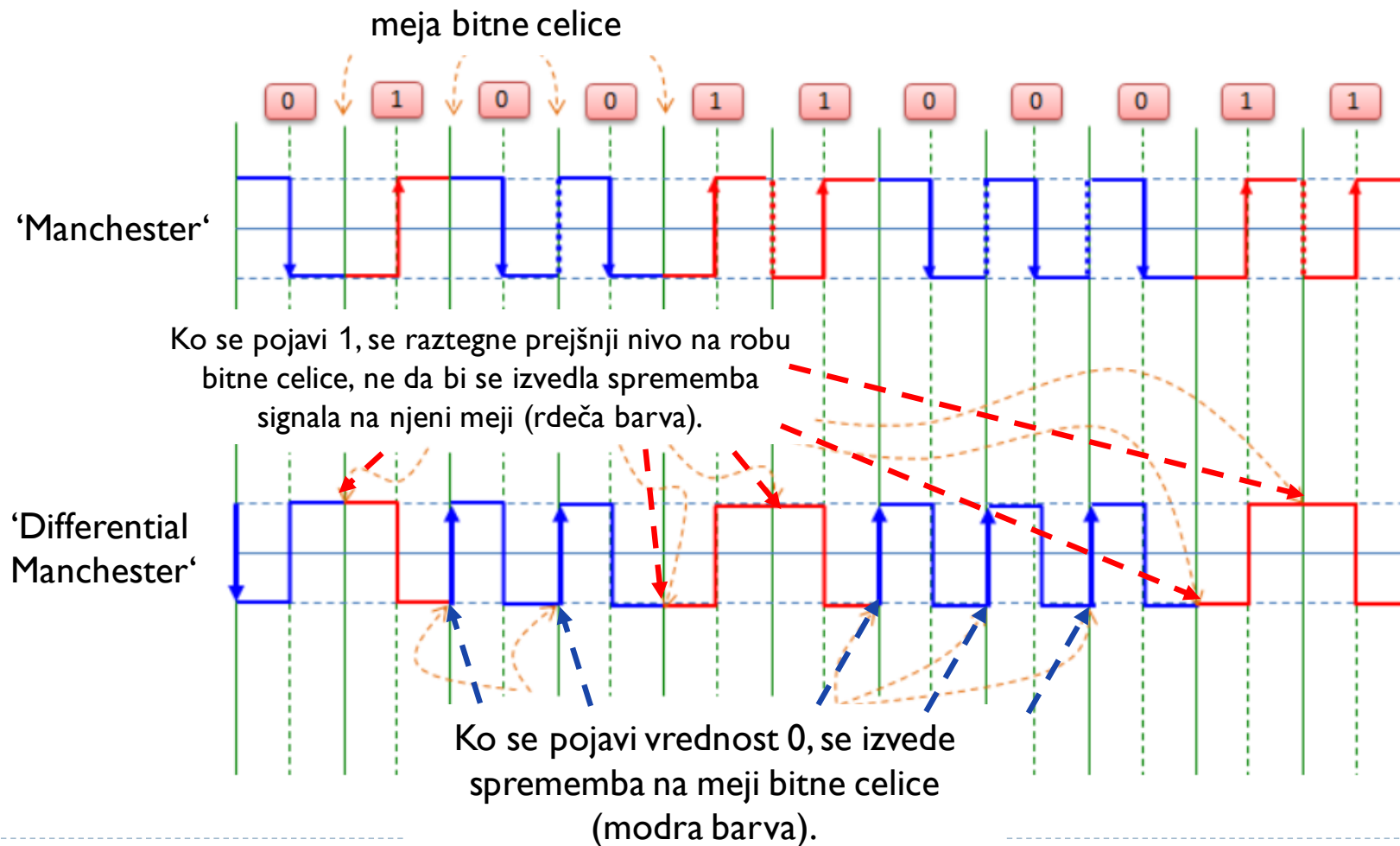
❑ Spremembe signala za posamezen bit:

- 0 → Prehod iz visokega nivoja (high) v nizki nivo (low) se izvede v sredini periode
- 1 → Prehod iz nizkega nivoja (low) v visoki nivo (high) se izvede v sredini periode
- Primer: 10-bitni niz, ki ima 4 logične enice → 13 sprememb signala



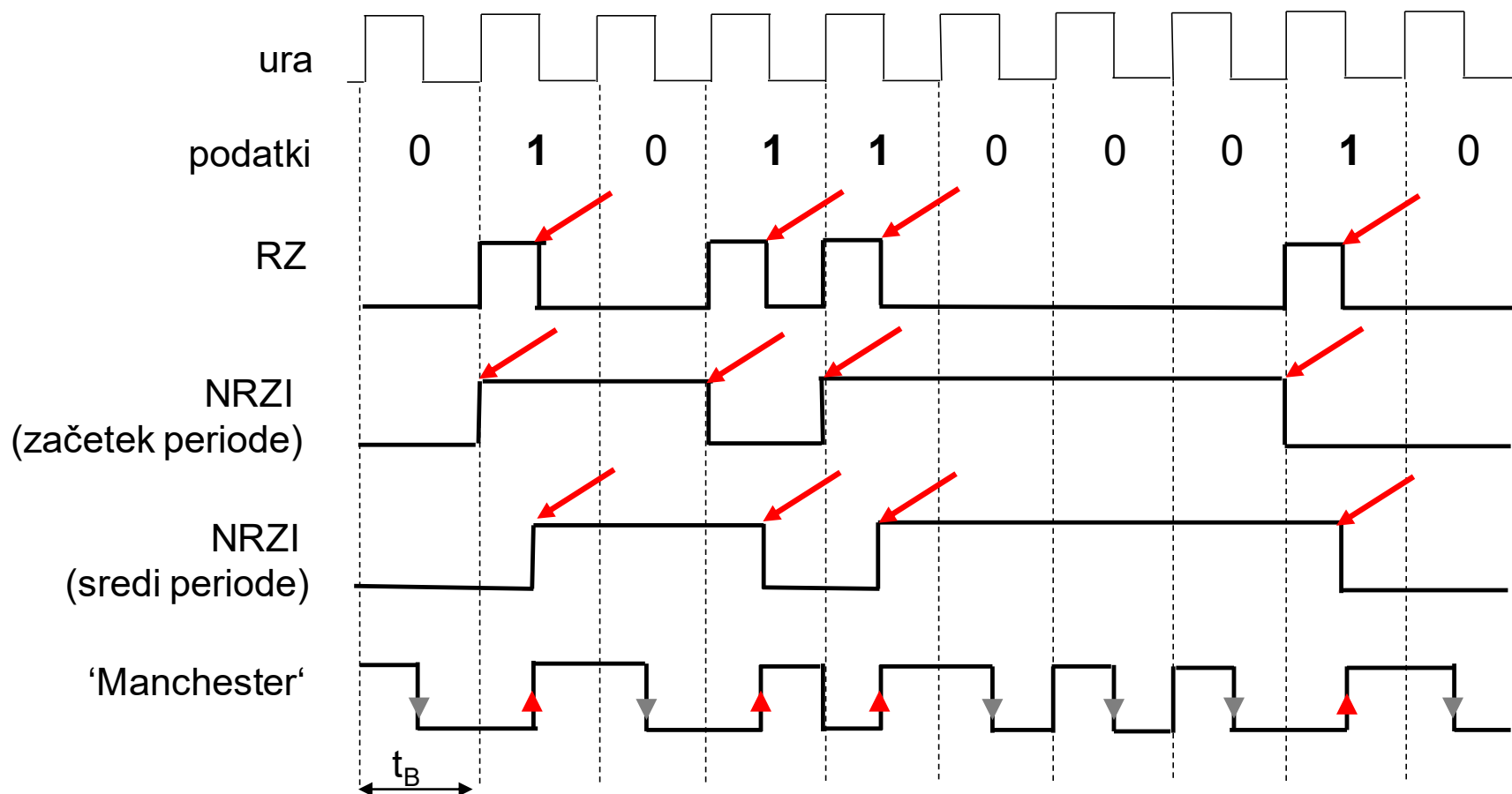
sprememba nivojev se izvede v
eni bitni celici v sredini periode

□ Primerjava kodiranja PE: 'Manchester' in 'Differential Manchester'



Primerjava kodiranja RZ, NRZI in PE (Manchester)

□ Prehodi nivojev signala



4 Kodiranje RLL (Run Length Limited)

- ❑ Podatkovni biti (uporabniški biti) se preoblikujejo tako, da **ne vsebujejo predolgega zaporedja ničel**.
 - ❑ Na tako dobljenem zaporedju bitov se nato uporabi **NRZI kodiranje**.
- ❑ Kodiranje RLL je dejansko NRZI kodiranje, kjer so vzorci bitov vnaprej določeni, tako da je določeno najdaljše in najkrajše število zaporednih ničel.
 - ❑ Različnih kod RLL kod je veliko
običajno so definirane s štirimi parametri: **m, n, d, k**
- ❑ Kodiranje RLL je določeno tako, da:
 - **se m uporabniških bitov** preslika v **n kodiranih bitov**, običajno je **$n > m$**
 - In da je v kodiranem zaporedju predpisano:
 - najmanjše dovoljeno število zaporednih ničel**, ki je določeno z **$d \geq 0$**
(gostota zapisa)
 - največje dovoljeno število zaporednih ničel**, ki je določeno s **$k \geq 0$**
(sinhronizacijska informacija)

- ❑ Uporabljajo se različne kode RLL, ki so označene z **m/n (d,k) RLL**:
 - 1/2 (2,7) RLL
 - 2/3 (1,7) RLL
 - 8/9 (0,4) RLL
 - 8/14 (2,10) RLL (CD, DVD)
 - Eno od kodiranj RLL je tudi kodiranje **8b/10b**.
- ❑ Prednosti kod RLL:
 - Dobra izkoriščenost kanala (NRZI) – večja gostota zapisa.
 - Odpravlja osnovno slabost NRZI kodiranja – pomanjkanje sinhronizacijske informacije pri predolgem zaporedju ničel (enic).
- ❑ Slabosti kod RLL:
 - Pri preslikavi se podatkovnim bitom dodajo dodatni (redundantni) biti.
- ❑ Uporaba:
 - Magnetni diski (HD), magnetni trakovi (MT), Optični diski (CD, DVD, BD)
 - Hitre serijske komunikacije
- ❑ Vir: <https://www.tomshardware.com/reviews/hard-drive-magnetic-storage-hdd,3005-6.html>

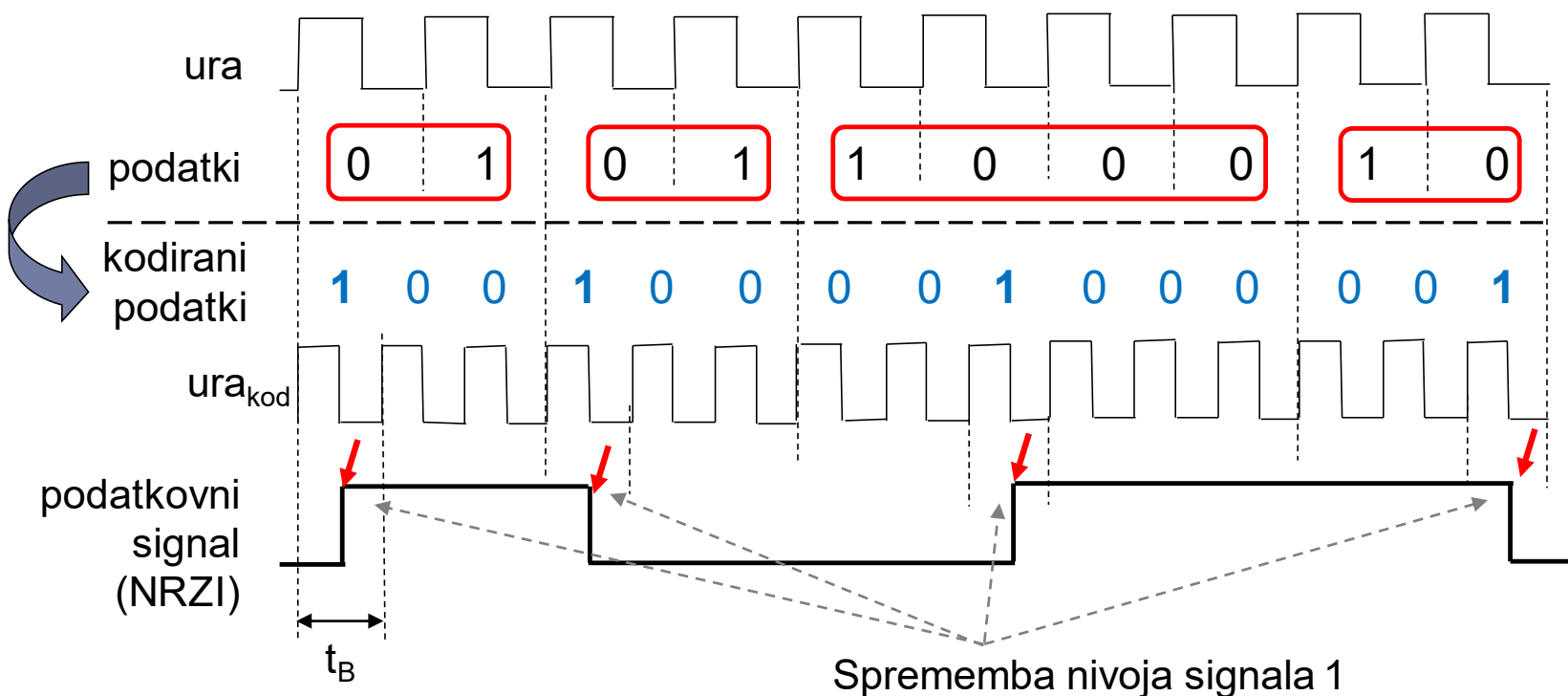
4.1 Kodiranje 2/3 (1,7) RLL

- ❑ Kodiranje 2/3 (1,7) RLL preslika 2 uporabniška bita v 3 kodirane bite tako, da je v kodiranih bitih vedno
 - najmanj 1 ničla in
 - največ 7 zaporednih ničel.
- ❑ Preslikovalna tabela za kodiranje 2/3 (1,7) RLL

Uporabniški biti	Kodirani biti
00	101
01	100
10	001
11	010
0000	101000
0001	100000
1000	001000
1001	010000

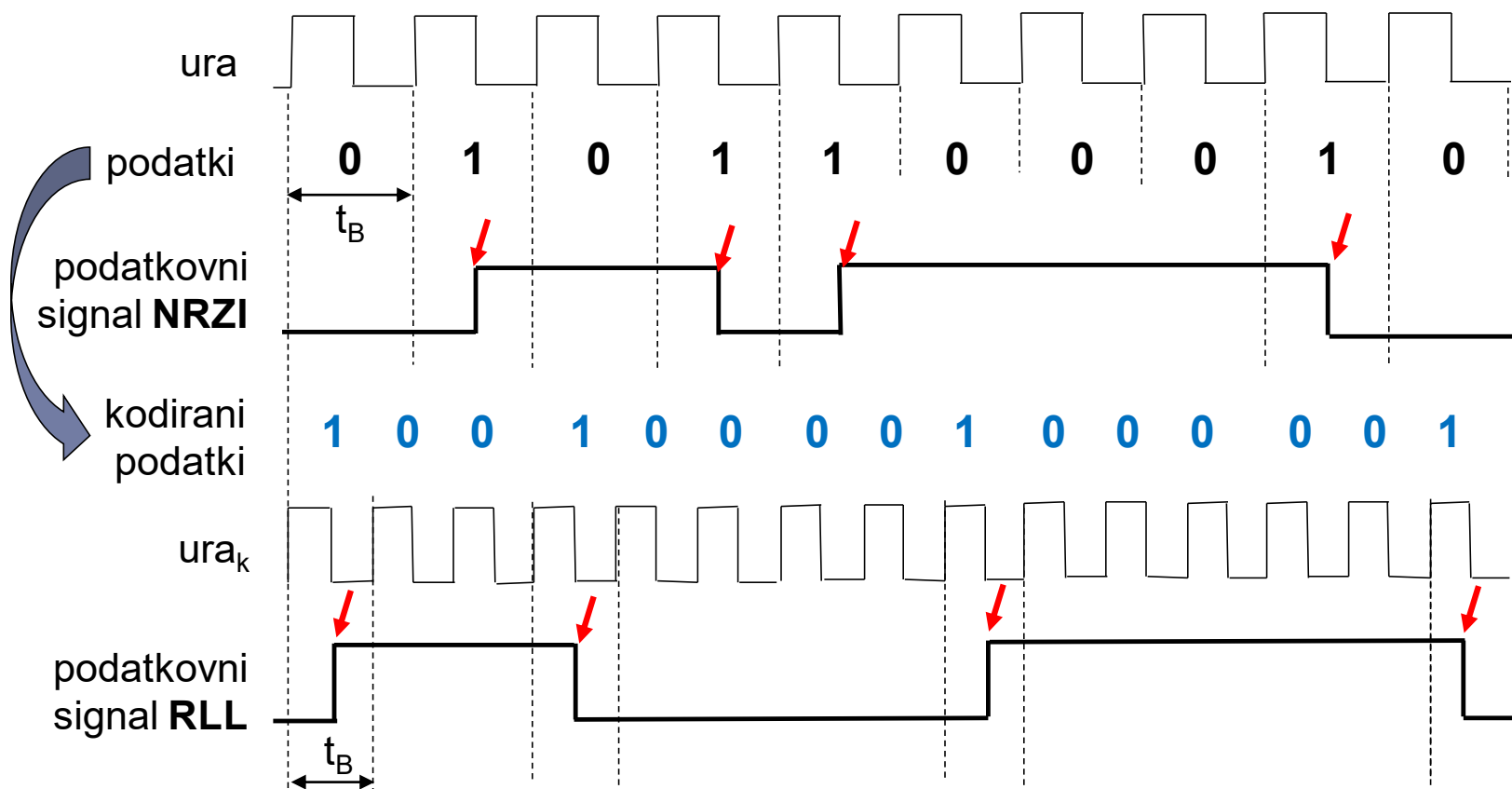
□ Primer kodiranja 2/3 (1,7) RLL

- 2 podatkovna bita → 3 kodirane bite
- 4 podatkovni biti → 6 kodiranih bitov
- 15 bitnih celic, NRZI – sprememba signala ob pojavu enic s sredini periode



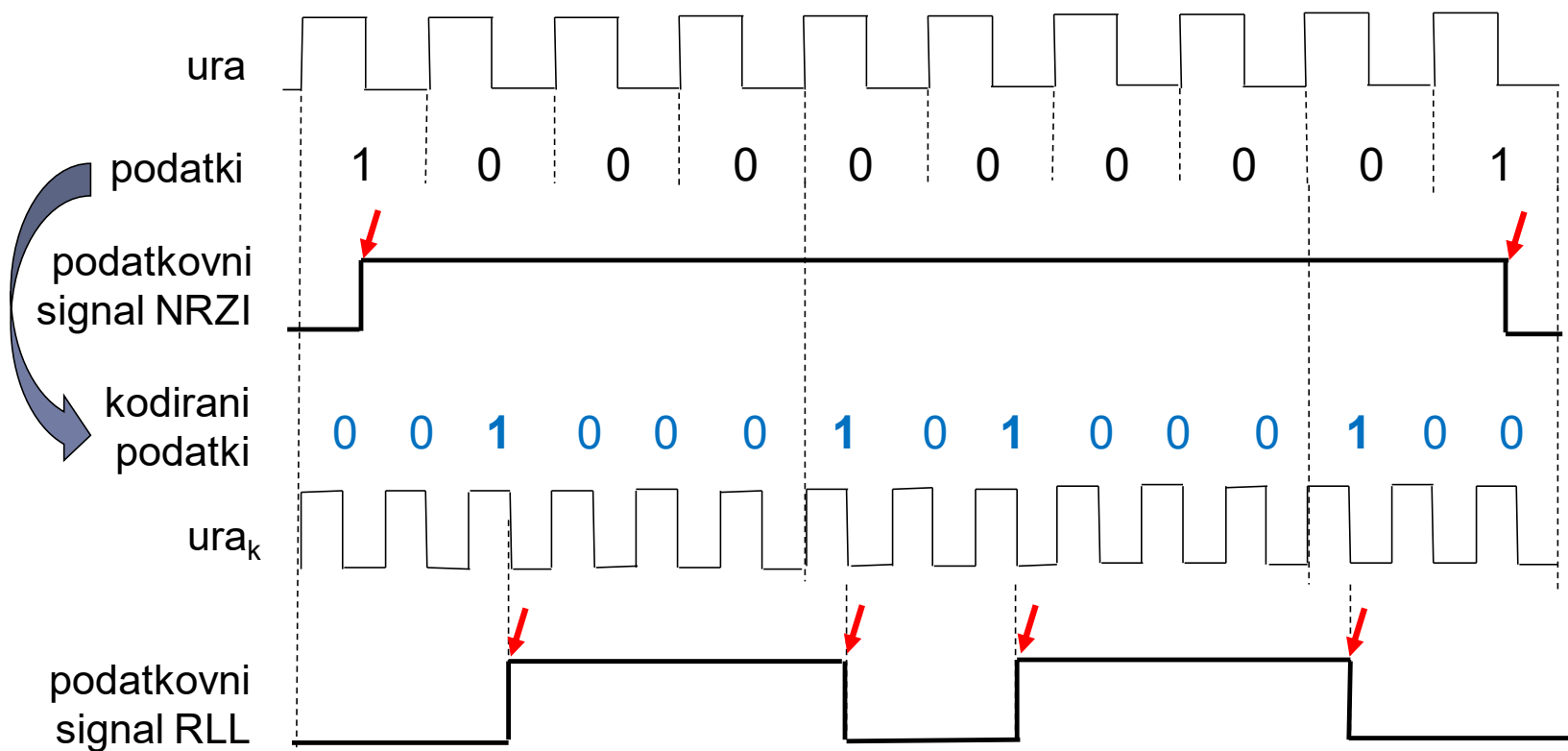
❑ Primer I: Primerjava kodiranja NRZI in kodiranja 2/3 (1,7) RLL (prejšnja stran)

- Podatkovni signal NRZI določajo podatkovni biti
- Podatkovni signal RLL določajo kodirani biti in NRZI



❑ Primer 2: Primerjava NRZI in 2/3 (1,7) RLL kodiranja

- Podatki vključujejo 8 zaporednih bitov po vrednosti 0: (100000001)



- ❑ Pri zapisu na **magnetne diske** se uporabljajo različne variante RLL kodiranja kot so:
 - 8/9 (0,4) RLL
 - 16/17 (0,6) RLL

- ❑ Tudi GCR (Group Code Recording), FM (Frequency Modulation) in MFM (Modified Frequency Modulation) kodiranja, ki so prvi poskusi izboljšanja NRZI kodiranja, lahko uvrstimo v skupino RLL kodiranj.
 - GCR = 4/5 (0,2) RLL (9-sledni magnetni trak)
 - FM kodiranje = 1/2 (0,1) RLL (5,25" diskete)
 - MFM kodiranje = 3/4 (1,3) RLL (3,5" diskete)

- ❑ Pri zapisu na **CD in DVD** se uporabljata EFM oziroma EFMP kodiranji:
 - EFM (Eight to Fourteen Modulation) je kodiranje 8/14 (2,10) RLL
 - EFMP (Eight to Fourteen Modulation Plus) je prav tako RLL kodiranje 8/16(2,10) RLL, ki je za 6% bolj učinkovito kot EFM

4.2 Kodiranje 8b/10b

- ❑ 8 podatkovnih bitov (8b) na vhodu se pretvori v 10 kodiranih bitov (10b) na izhodu.
- ❑ Za izvedbo sta uporabljeni dve enostavnejši kodiranji:
 - **3b/4b** – kodirna tabela za pretvorbo 3 bitov v 4 bite
 - **5b/6b** - kodirna tabela za pretvorbo 4 bitov v 5 bitov

- ❑ Cilj preslikave 8 bitov v 10 bitov:
 - **Zagotoviti dovolj prehodov nivojev**, ki omogočajo sinhronizacijo oziroma restavrancijo urinega signala pri sprejemu (ang. clock and data recovery).
 - V nizu kodiranih podatkov **ne sme biti več kot 5 zaporednih ničel ali enic**.
- ❑ Serijski prenos z **veliko hitrostjo in na večje razdalje** je
 - omogočen z enosmerno uravnoteženostjo signala - **razlika v številu enic in ničel v nizu najmanj 20 bitov ne sme biti večja od 2**.
- ❑ Redundančnost bitov:
 - Kodiranje 8b/10b prinese **25% dodatnih redundantnih bitov**.
 - Pri novejših povezavah se uporablja kodiranje **128b/130b**, ker je redundanca pri tem kodiranju 1,56% (128 podatkovnih bitov in 130 kodiranih bitov).

❑ Uporaba:

- PCI Express 2.0 (PCI Express 3.0 – kodiranje 128b/130b)
- USB 3.0
- USB 3.1 – kodiranje 128b/130b
- IEEE 1394 (Firewire)
- Serial ATA
- Gigabit Ethernet
- DVI in HDMI
- ...

❑ Primer:

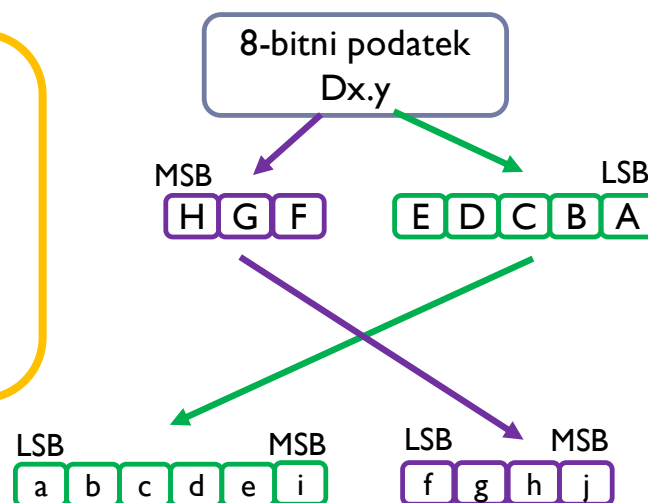
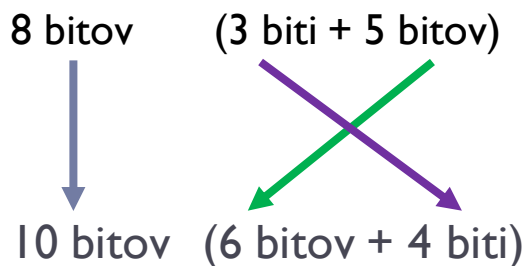
Na enojni povezavi PCIe 2.0 x1 je pri frekvenci ure 2,5GHz hitrost prenosa enaka 2,5Gb/s. Kakšna je efektivna hitrost prenosa zaradi kodiranja 8B/10B?

Izračun: $2,5 \text{ [Gb/s]} * 8 \text{ [b]} / 10 \text{ [b]} = 2 \text{ [Gb/s]}$ ali 250 [MB/s].

Postopek kodiranja 8b/10b

- Postopek zamenjave '8 bitov' (vhod) v '10 bitov' (izhod) se izvede kot kombinacija dveh metod kodiranja 3B/4B in 5B/6B.

- Vzamemo 8 bitov vhodnega podatka HGFEDCBA
- Razdelimo ga v 2 bloka: **HGF EDCBA**
- Zamenjamo poziciji rotiranih kodiranih bitov in s kodiranjem 3B/4B in 5B/6B dobimo 10 bitov: **abcdei fghj**



- V zaporedju ne smejo biti več kot 4 enaki biti (10-bitna koda mora vsebovati 5 ničel in 5 enic, ali 4 ničle in 6 enic, ali 6 ničel in 4 enice)
- Definirani morata biti dve tabeli za kodiranje: 3b/4b in 5b/6b

Primer kodiranja:

Kodirna tabela 3b/4b

3B Input (Decimal)	3B Input (Binary)	4B Output (Binary)
0	000	0100 or 1011
1	001	1001
2	010	0101
3	011	0011 or 1100
4	100	0010 or 1101
5	101	1010
6	110	0110
7	111	0001 or 1100 or 1000 or 0111

(8 bitov: MSB ... LSB):

00000100 → HGF: 000, EDCBA: 00100

abcdei: 110101, fghj: 0100

(10 bitov: LSB ... MSB): 1101010100

ali

(8 bitov: MSB ... LSB):

00000100 → HGF: 000, EDCBA: 00100

abcdei: 001010, fghj: 1011

(10 bitov: LSB ... MSB): 0010101011

Kodirna tabela 5b/6b

5B Input (Decimal)	5B Input (Binary)	6B Output (Binary)
0	00000	100111 or 011000
1	00001	011101 or 100010
2	00010	101101 or 010010
3	00011	110001
4	00100	110101 or 001010
5	00101	101001
6	00110	011001
7	00111	111000 or 000111
8	01000	111001 or 000110
9	01001	100101
10	01010	010101
11	01011	110100
12	01100	001101
13	01101	101100
14	01110	011100
15	01111	010111 or 101000
16	10000	011011 or 100100
17	10001	100011
18	10010	010011
19	10011	110010
20	10100	001011
21	10101	101010
22	10110	011010
23	10111	111010 or 000101
24	11000	110011 or 001100
25	11001	100110
26	11010	010110
27	11011	110110 or 001001
28	11100	001110
29	11101	101110 or 010001
30	11110	011110 or 100001
31	11111	101011 or 010100