



# Vhodno-izhodne naprave (VIN)

Predavanja

## 8. Prenosi in kodiranja signalov, očesni vzorec

Robert Rozman

[rozman@fri.uni-lj.si](mailto:rozman@fri.uni-lj.si)

# Vsebina

---

- ❑ 8.1 Povezave in hitrost prenosa podatkov
  - ❑ Dodatni pojavi, ki vplivajo na hitrost prenosa podatkov
    - Čas vzpona linije (ang. transmission line rise time)
    - Zamik (ang. skew)
    - Tresenje (ang. jitter)
    - Intersimbolna interferenca (ang. ISI .. Inter Symbol Interference)
  
- ❑ 8.2 Kodiranja signalov
  - ❑ Kodiranje RZ (Return to Zero)
  - ❑ Kodiranje NRZ (No Return to Zero)
  - ❑ Kodiranje PE (Phase Encoded)
  - ❑ Kodiranje RLL (Run Length Limited)
    - Kodiranje 2/3 (1,7) RLL
    - Kodiranje 8b/10b
  
- ❑ 8.3 Očesni vzorec (ang. eye pattern)

# 8.1 Povezave in hitrost prenosa podatkov

---

## ❑ Od prej že poznamo:

- ❑ Baudna hitrost (BR) – število signalnih elementov ( $T_{UI}$ ) v sekundi [baud].
- ❑ Bitna hitrost (C)– število bitov na sekundo [bit/s].

## ❑ Pojavi, ki omejujejo hitrost prenosa

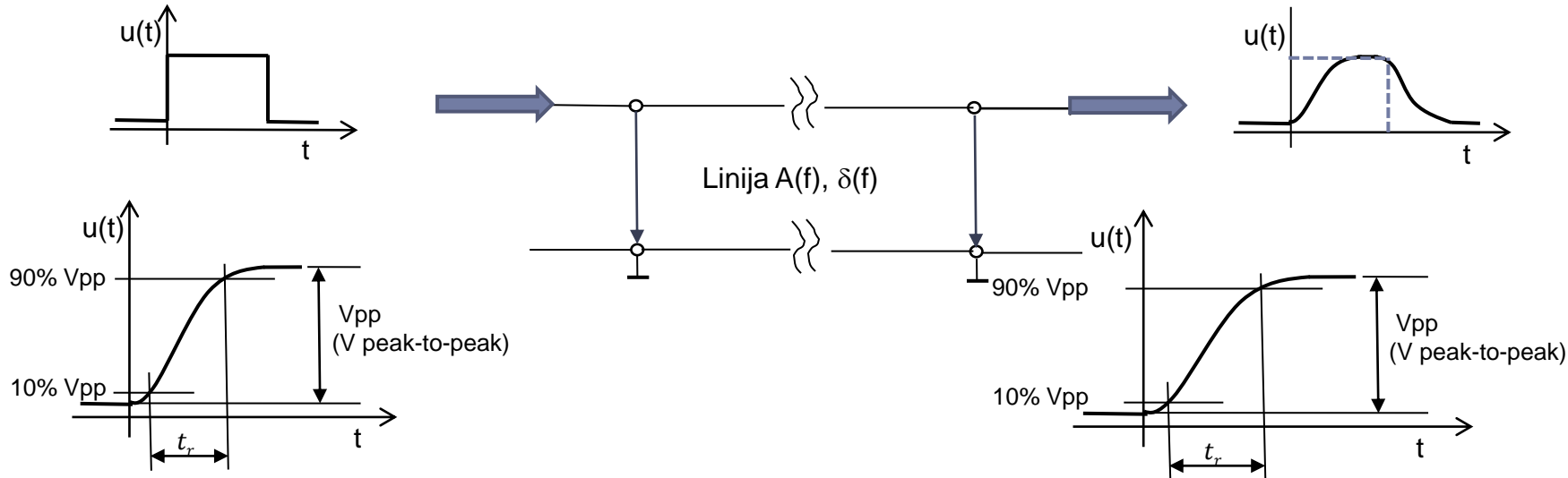
- Čas vzpona linije (ang. transmission line rise time) – na izhodu linije je čas vzpona večji kot na vходу.
- Zamik (ang. skew) - časovna razlika med dejanskim in pričakovanim časom prihoda signala.
- Tresenje (ang. jitter) - časovno odstopanje fronte signala od pravilnega položaja.
- Intersimbolna interferenca (ang. intersymbol interference) - vpliv vrednosti prejšnjega bita na sosednjega.

## ❑ Določanje (ocena) najvišje hitrosti prenosa

- Meritev očesnega vzorca (ang. eye pattern)

## 8.1.1 Čas vzpona linije

- Pravokotni signal je na izhodu linije popačen, čas vzpona signala je daljši kot pa na vhodu. Imenujemo ga **čas vzpona linije**  $t_r$

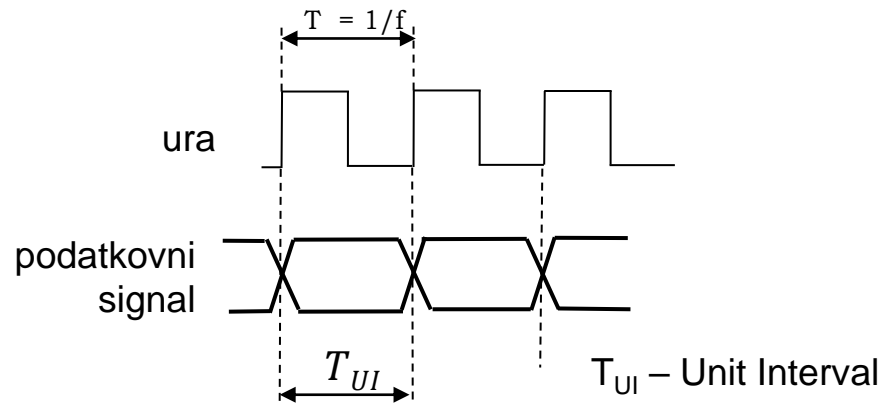
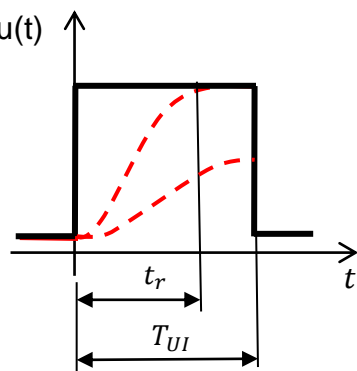


- Idealna linija - čas vzpona linije bi bil enak 0.
- Realna linija - čas vzpona linije linearno narašča z dolžino linije.
- Komponente pravokotnega signala različnih frekvenc, ki vstopajo na vhod linije istočasno, na izhod linije ne prispejo istočasno in so različno oslabljene.
  - Slabljenje linije je odvisno od frekvence signala- z višanjem frekvence slabljenje narašča.
  - Hitrost potovanja signala po liniji narašča s frekvenco – zakasnitev signala  $\delta$  [ns/m] pa pada.

- ❑ Pri prenosu pravokotnih signalov mora veljati, da je čas vzpona signala manjši od časa trajanja signalnega elementa  $t_r < T_{UI}$ .

- ❑ Za zanesljiv prenos podatkov mora veljati, da je čas vzpona  $t_r < \frac{T_{UI}}{2}$ , s čimer je določena maksimalna hitrost prenosa.

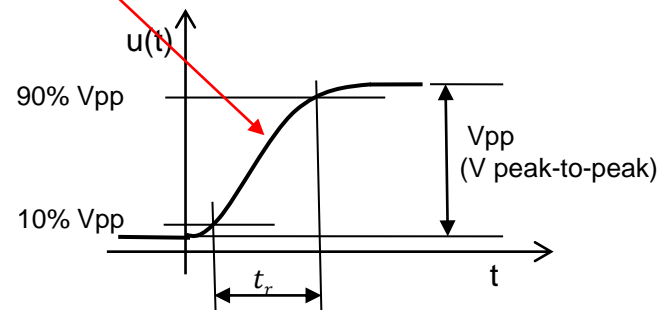
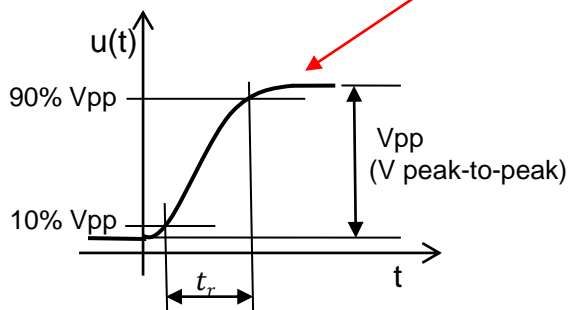
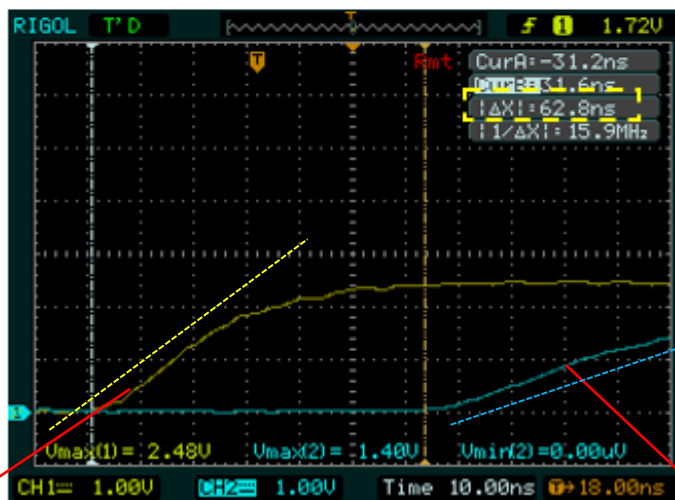
Primer:



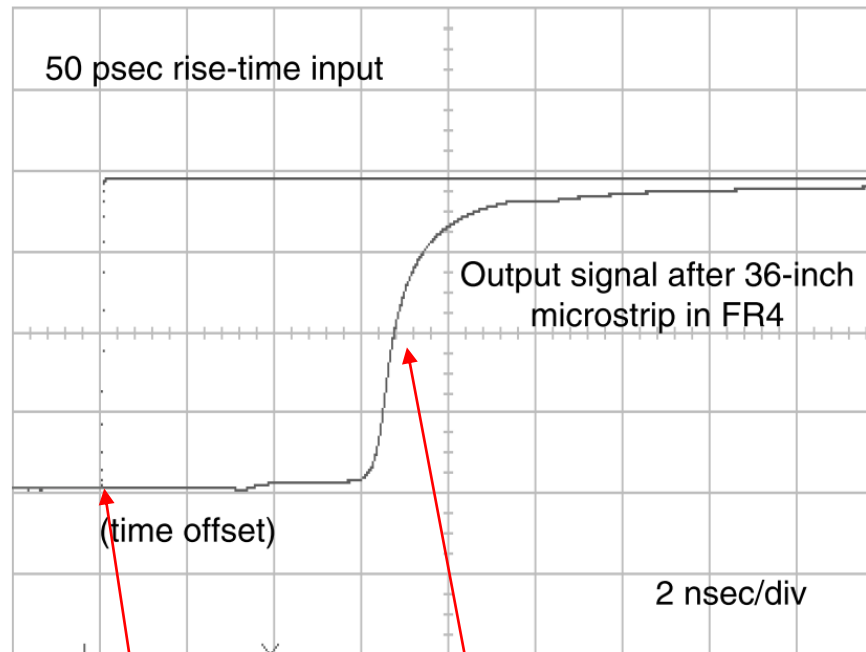
- ❑ Ker je hitrost prenosa odvisna od števila signalnih elementov na sekundo, čas vzpona  $t_r$  omejuje najvišjo hitrost prenosa.

# 8.1.1 Čas vzpona linije – primer 1: LAB ploščati kabel

Izmerite čas potovanja po ploščatem kablu

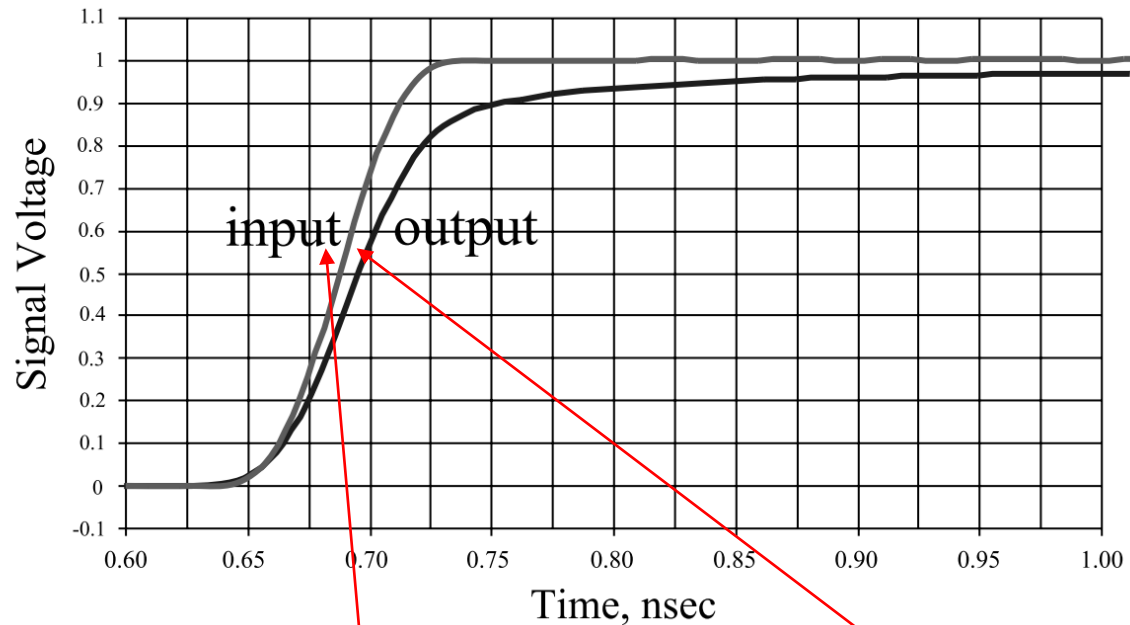


## 8.1.1 Čas vzpona linije – primer 2



**Figure 2-10** Top: The measured attenuation through a 4-inch length of 50-Ohm transmission line in FR4 showing the higher attenuation at higher frequencies. Bottom: The measured input and transmitted signal through a 36-inch 50-Ohm transmission line in FR4, showing the rise time to have degraded from 50 psec to more than 1.5 nsec.

## 8.1.1 Čas vzpona linije – primer 2



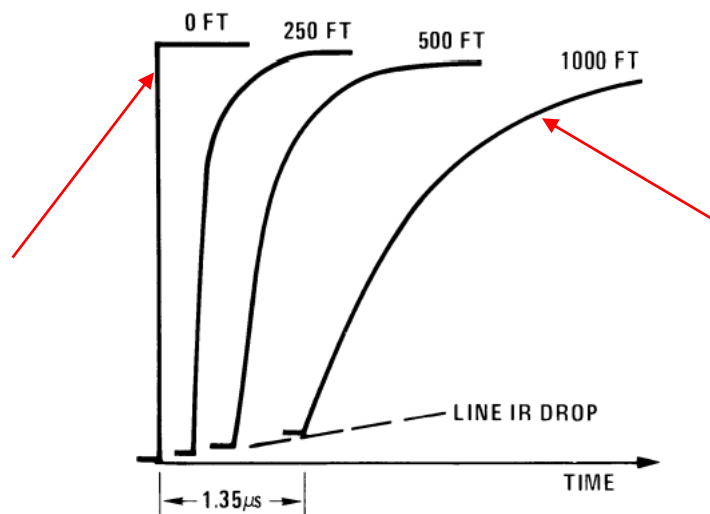
**Figure 2-20** Measured input and transmitted signal through a 4-inch long, 50-Ohm transmission line in FR4 showing the rise-time degradation. The input rise time is 50 psec. The predicted output rise time is 67 psec based on the measured bandwidth of the interconnect. Measured with a GigaTest Labs Probe Station.



## 8.1.1 Čas vzpona linije – primer 3

### Transmission Line Characteristics

National Semiconductor  
Application Note 108  
Bill Fowler  
July 1986



AN008826-5

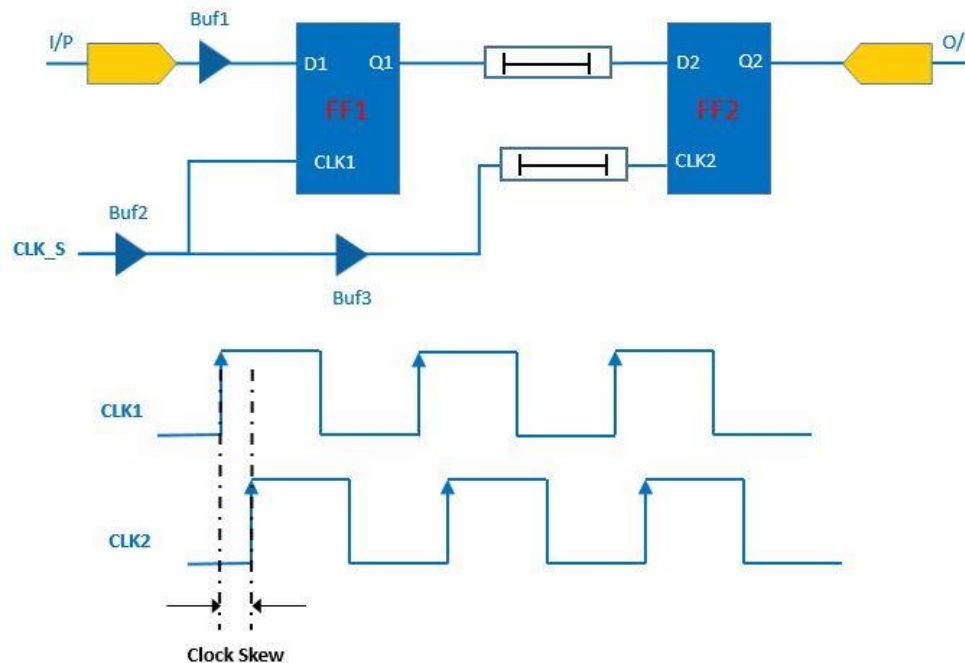
**FIGURE 5. Signal Response at Receiver**

A primary cause of distortion is the effect the transmission line has on the rise time of the transmitted data. **Figure 5** shows what happens to a voltage step from the driver as it travels down the line. The rise time of the signal increases as the signal travels down the line. This effect will tend to affect the timing of the recovered signal.

## 8.1.2 Zamik (ang. skew)

- Zamik je časovna razlika med dejanskim in pričakovanim časom prihoda signala med dvema dogodkoma, ki bi se v idealnih razmerah morala zgoditi istočasno (Definicija JEDEC Standard No. 65B).

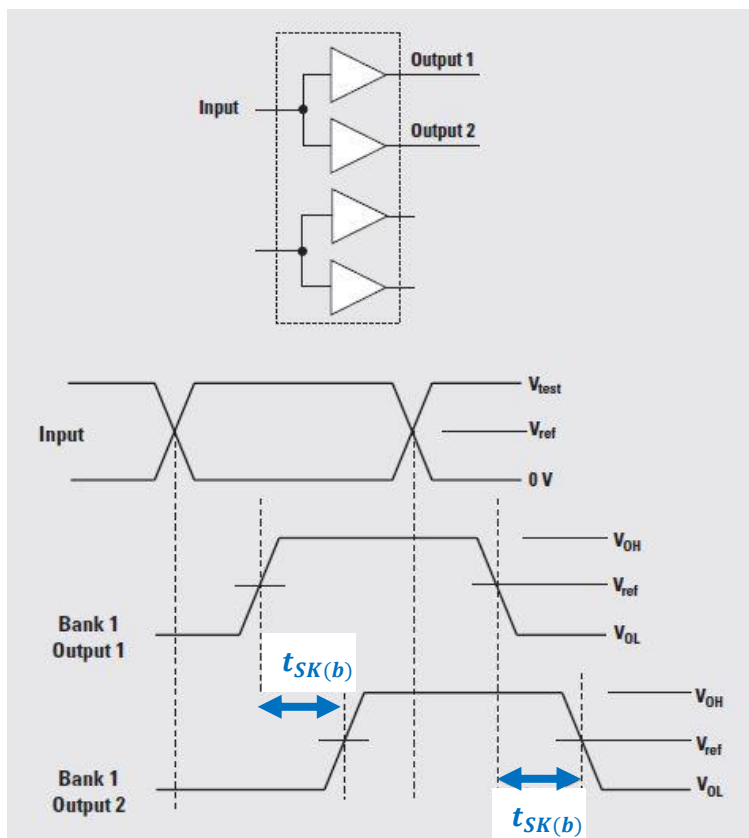
- Primer: zamik urinega signala („clock skew“)



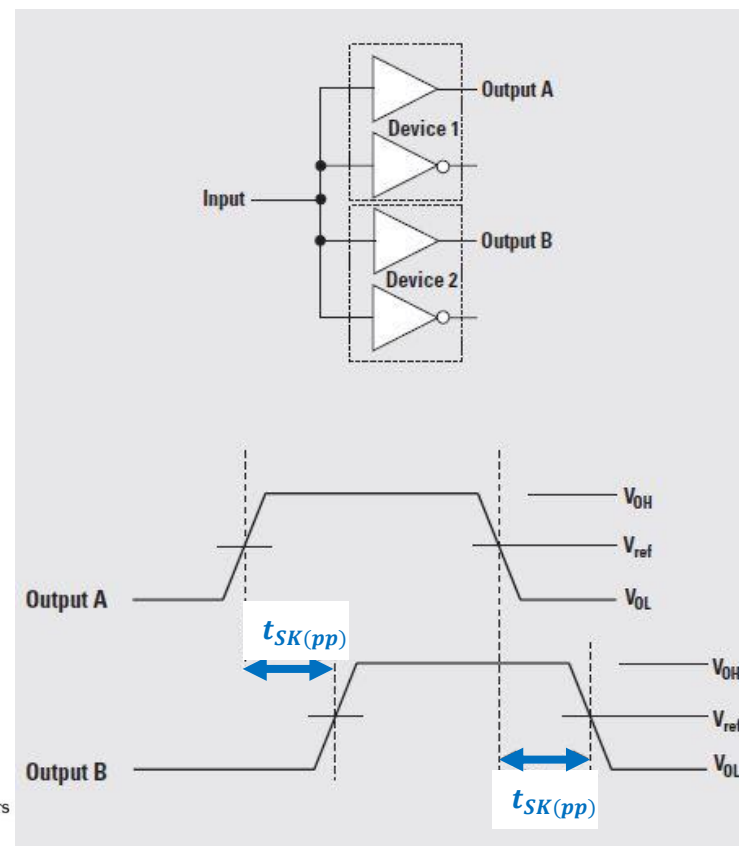
<http://www.vlsi-expert.com/2016/01/skew.html>

## Primeri različnih zamikov

- ❑ **Bank skew ( $t_{SK(b)}$ ):** Zamik med dvema izhodoma oddajnikov v istem čipu.

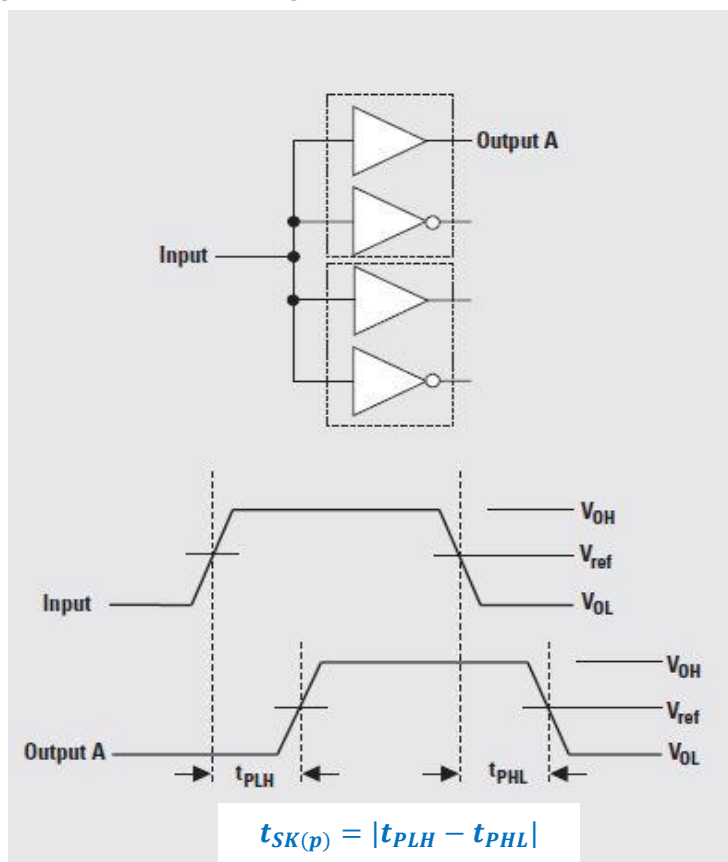


- ❑ **Part-to-part skew ( $t_{SK(pp)}$ ):** Zamik med dvema izhodoma oddajnikov v različnih čipih.

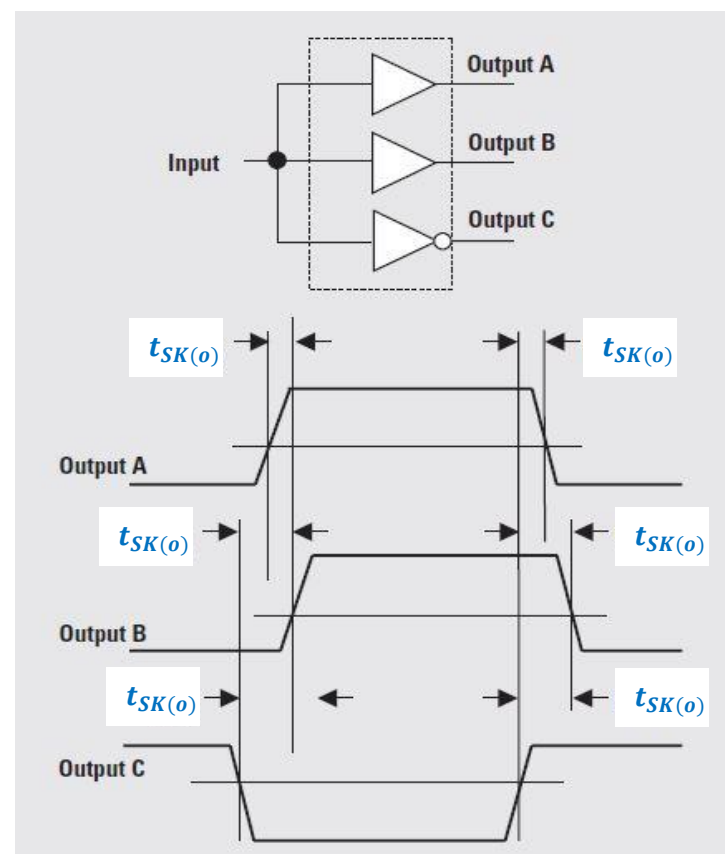


## Primeri različnih zamikov

- ❑ *Pulse skew* ( $t_{SK(p)}$ ): Impulzni zamik – časovna razlika med zakasnitvama pozitivne in negativne fronte.

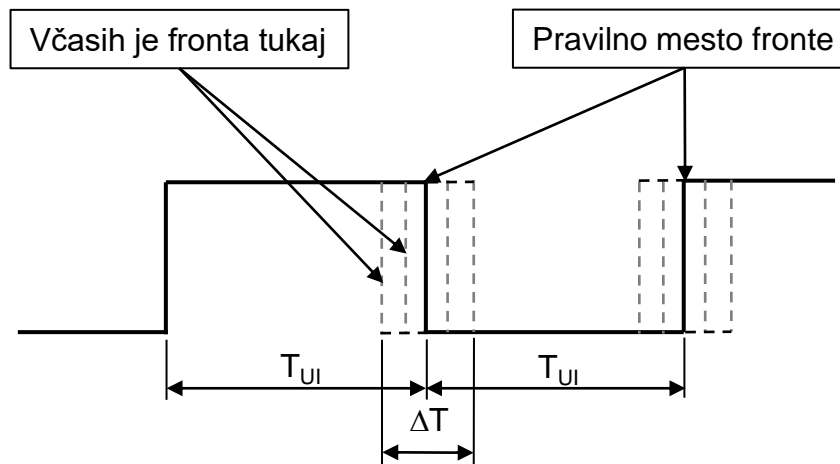


- ❑ *Output skew* ( $t_{SK(o)}$ ): Zamik med izhodi v čipu pri skupaj vezanih vhodih.

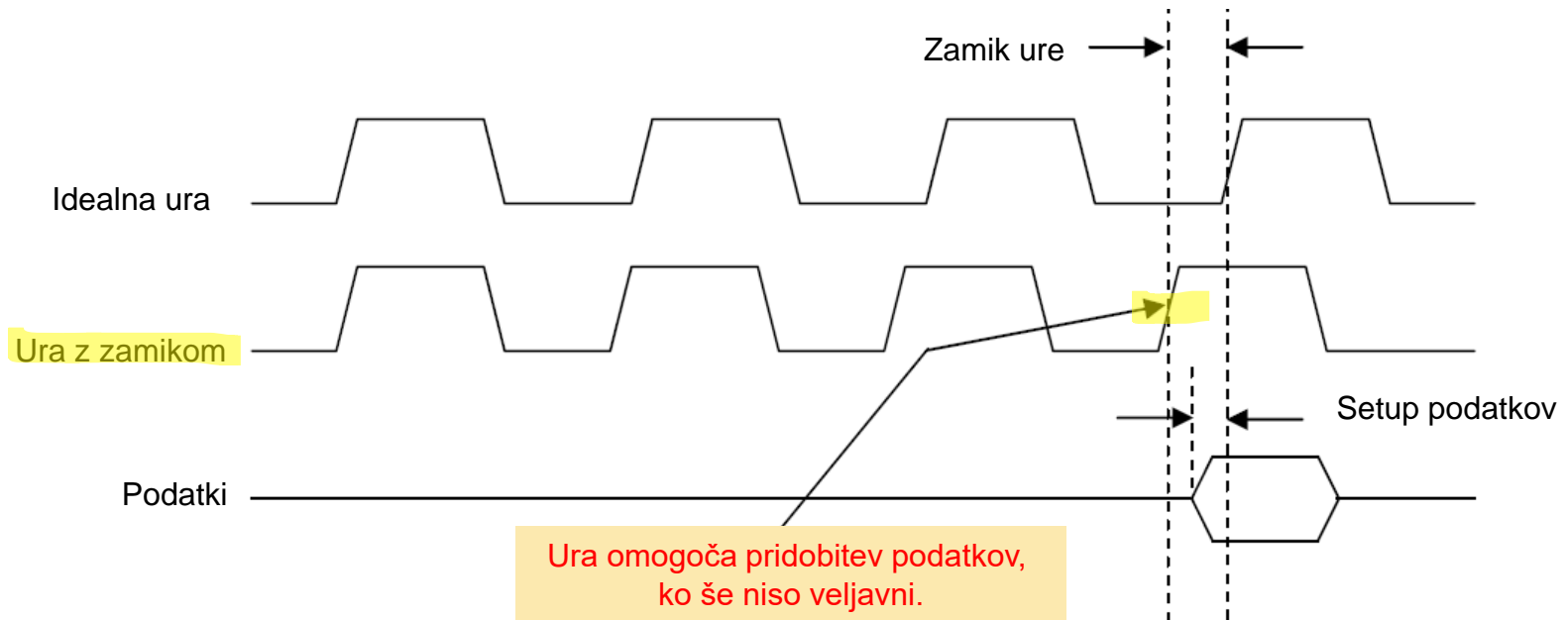


## 8.1.3 Tresenje (ang. jitter)

- **Tresenje** (ang. jitter) je časovno odstopanje fronte signala od pravilnega položaja. (Definicija JEDEC Standard No. 65B)
  - Pri prenosu podatkov je preklop iz enega stanja v drugo definiran s fronto signala, tresenje je časovno odstopanje te fronte od pravilnega mesta.
  - Pogosto se tresenje definira kot vsoto vseh zamikov signala (ang. skew), odbojev (ang. reflections), medsimbolne interference (ang. Inter Symbol Interference-ISI), zakasnitev (ang. delay) in šuma (ang. noise), ki poslabšuje kvaliteto signala.
  - *Primer: Clock Jitter – konstantna povprečna frekvenca signala, vendar pride do kratkoročnih odstopanj -> tresenje*



## ❑ Tresenje periode (ang. Period jitter)



## ❑ Primer v časovnem diagramu:

- Mikroprocesorski sistem zahteva nastavitve podatkov  $1\text{ ns}$  pred prehodom ure.
- Če je čas tresenja ure daljši, to je  $1,5\text{ ns}$ , potem lahko pride do spremembe prehoda ure preden so na voljo veljavni podatki.
- **V tem primeru bo mikroprocesor pridobil napačne podatke.**

# Ti : Jitter vs. skew (primer ohlapne definicije)

## Skew definition and jitter analysis

Figure 1). A more appropriate jitter definition would seem to be one for something called “jitter stew.”

### Jitter stew

Jitter can best be defined as the sum total of skews, reflections, pattern-dependent interference, propagation delays, and coupled noise that degrade signal quality.

Figure 2. Standard jitter measurement

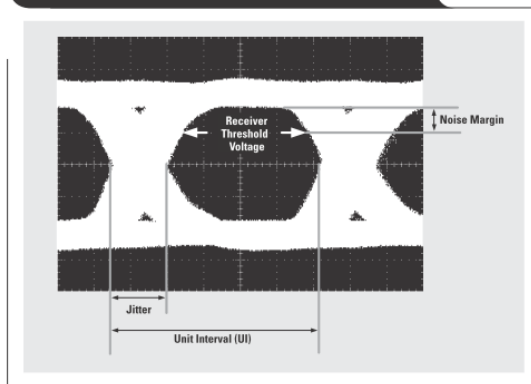
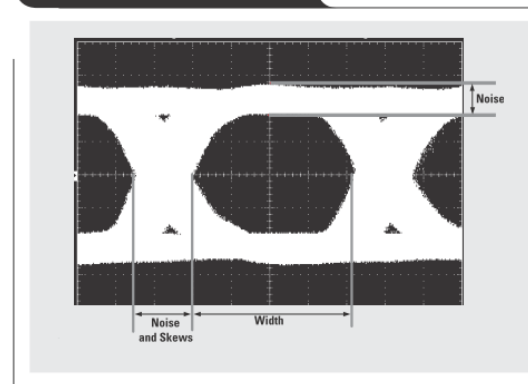
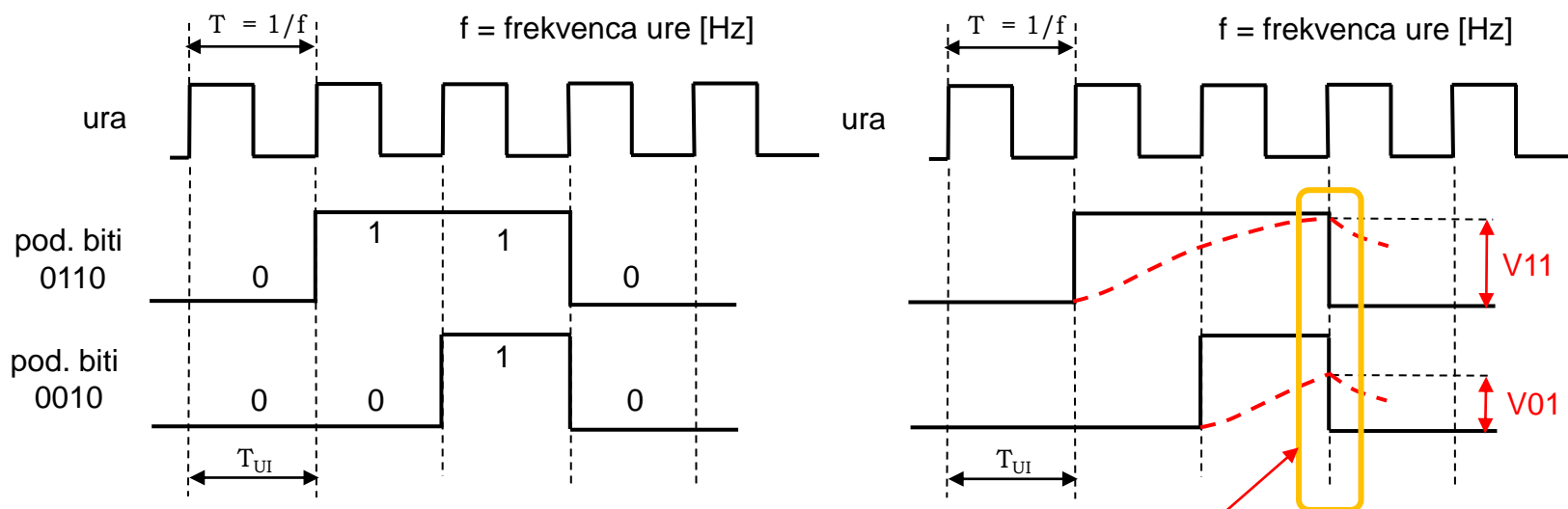


Figure 3. Noise and skews



## 8.1.4 Intersimbolna interferenca (ang. Inter Symbol Interference-ISI)

- Intersimbolna interferenca (ISI) nastopi, kadar v neko točko linije pride naslednji simbol, še preden je predhodni simbol dosegel svojo maksimalno vrednost.
- Ali enostavneje: ISI opredeljuje vpliv vrednosti prejšnjih bitov na trenutni bit.
- Pri velikih hitrostih, kjer je kratek čas trajanja simbola ( $T_{UI}$ ), pride ta vpliv do izraza.

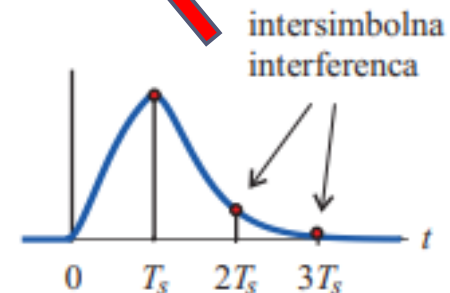
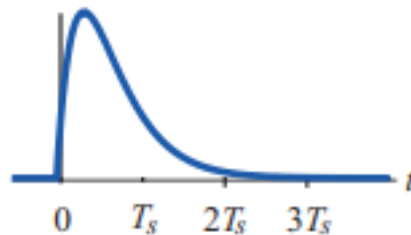
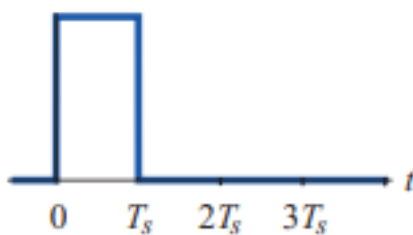
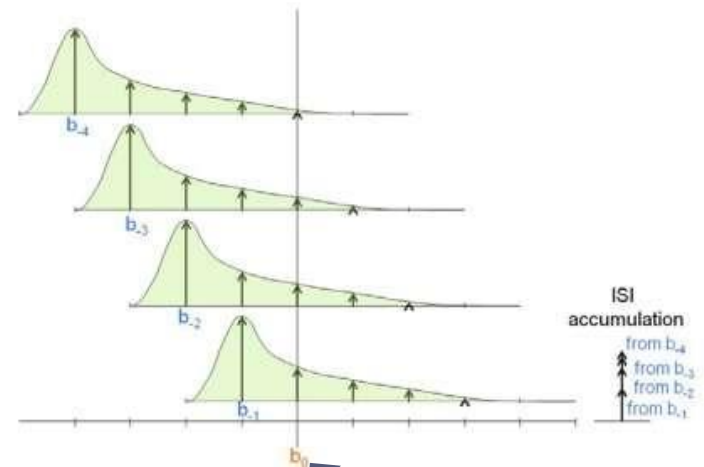
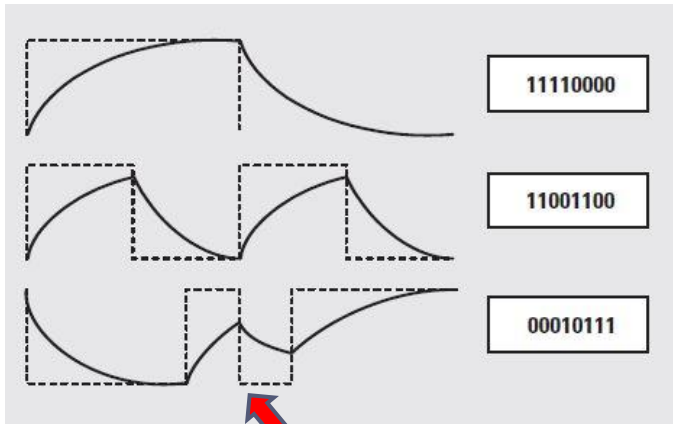


- Napetost signala v bitni celici ni enaka, če je bil predhodni bit 0 ali 1 (pri 01 < pri 11).
- Intersimbolno interferenco lahko zmanjšamo, če:
  - povečamo čas trajanja signalnega elementa (kar pomeni zmanjšamo hitrost prenosa)
  - zmanjšamo čas vzpona linije (skrajšamo linijo)
  - uporabimo izenačevalnike signala (ang. equalizer)



□ Primer intersimbolne reference:

Na prejeta stanje signala podanega podatkovnega simbola vplivajo prejšnji simboli in naslednji simboli.



## 8.2 Kodiranja signalov

---

- ❑ Uvod v kodiranja signalov
- ❑ Kodiranje RZ (Return to Zero)
- ❑ Kodiranje NRZ (No Return to Zero)
- ❑ Kodiranje PE (Phase Encoded)
- ❑ 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](http://www.sharetechnote.com/html/Communication_Digital_Encoding.html)

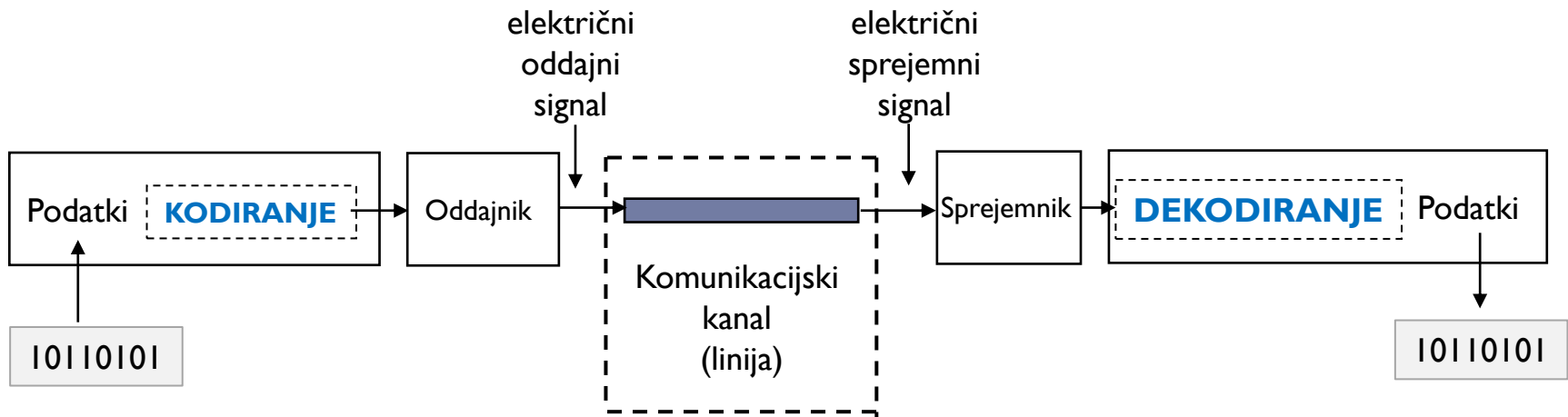
# Uvod v kodiranje signalov

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



# Uvod v kodiranje signalov

- ❑ Model prenosa podatkov med napravami



- ❑ Lahko tudi enostavno „kodiranje“:
  - ❑ predstavitev enic z visokim in ničel z nizkim nivojem za
    - prenos po komunikacijskih kanalih in
    - shranjevanje na polprevodniške, magnetne in optične medije.
- ❑ Običajno bolj kompleksno kodiranje (ang. encoding) in dekodiranje (ang. decoding)
- ❑ Sinhronizacija - restavriranje urinega signala (ang. clock recovery) v sprejemniku

# Zakaj kodiranje pomembno ?

## Spektralna „učinkovitost“

- Digitalni signal **želimo 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 odklik 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.

## Boljša učinkovitost prenosa (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).

## Lastnosti prenosnega/shranjevalnega medija

- **magnetni, optični diski, el. povezave, ...**

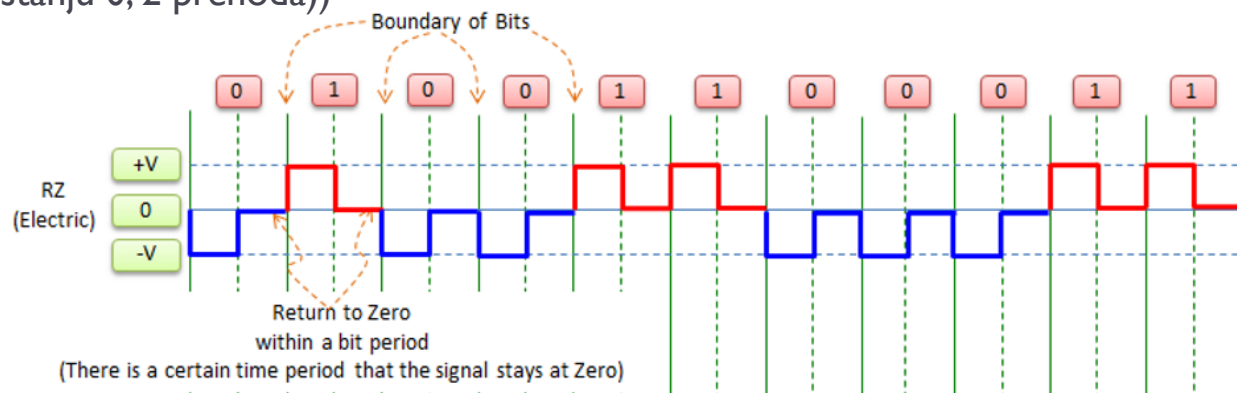
## 8.2.1 Kodiranje RZ (Return to Zero)

### ❑ Slabosti:

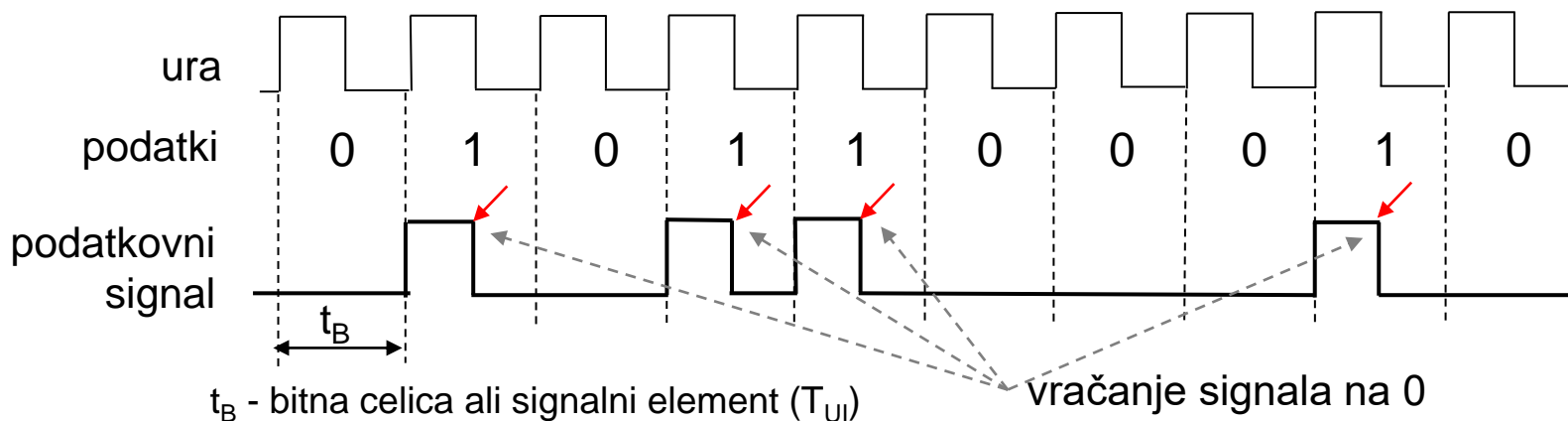
- ❑ DC pomik (daljši nizi 0 ali 1)
- ❑ Izkoristek (polovica periode v stanju 0, 2 prehoda)

### ❑ Prednosti:

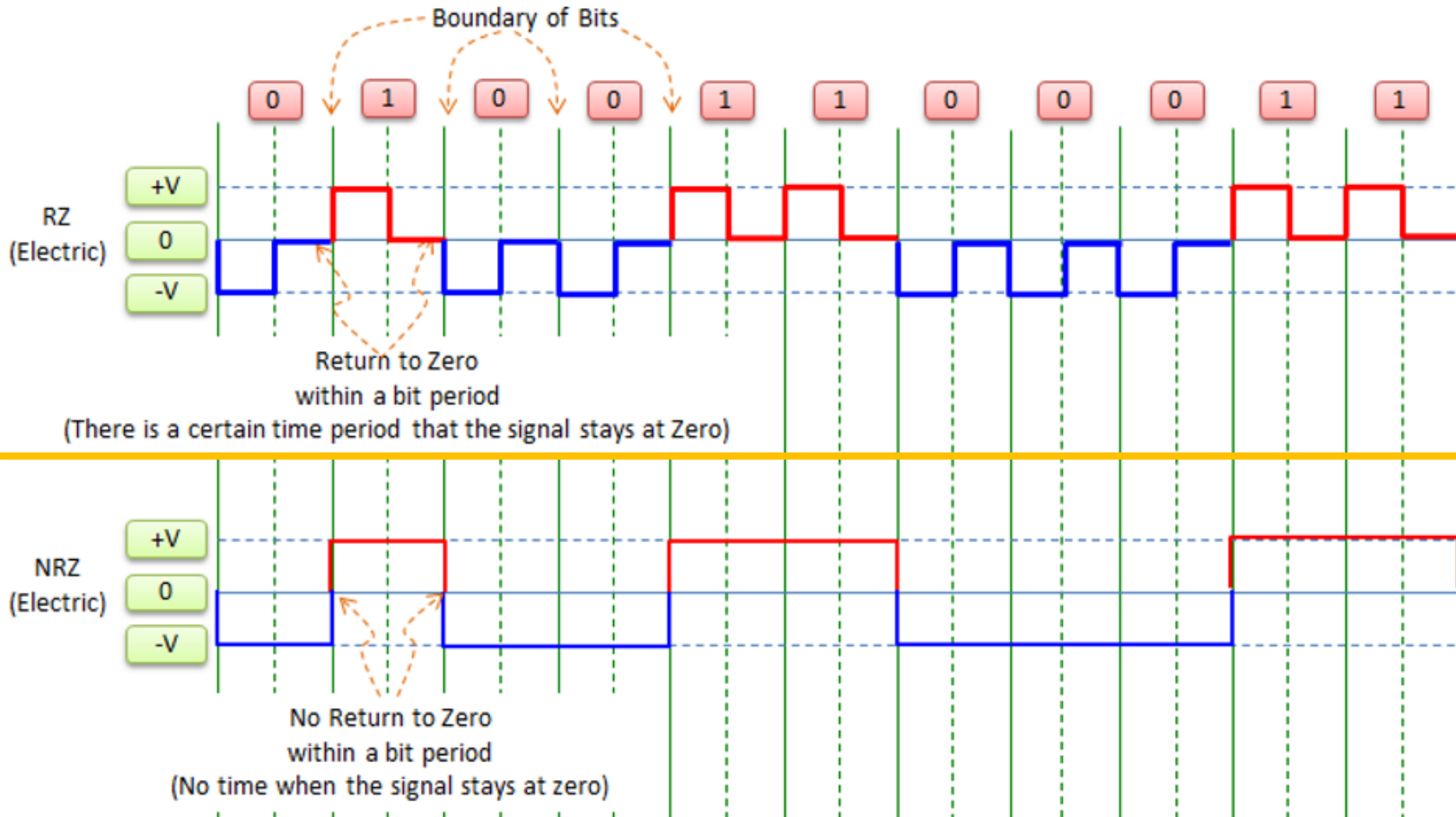
- ❑ Več prehodov (sinhronizacija)



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



# 1 Kodiranje RZ (Return to Zero)



## ❑ Slabosti:

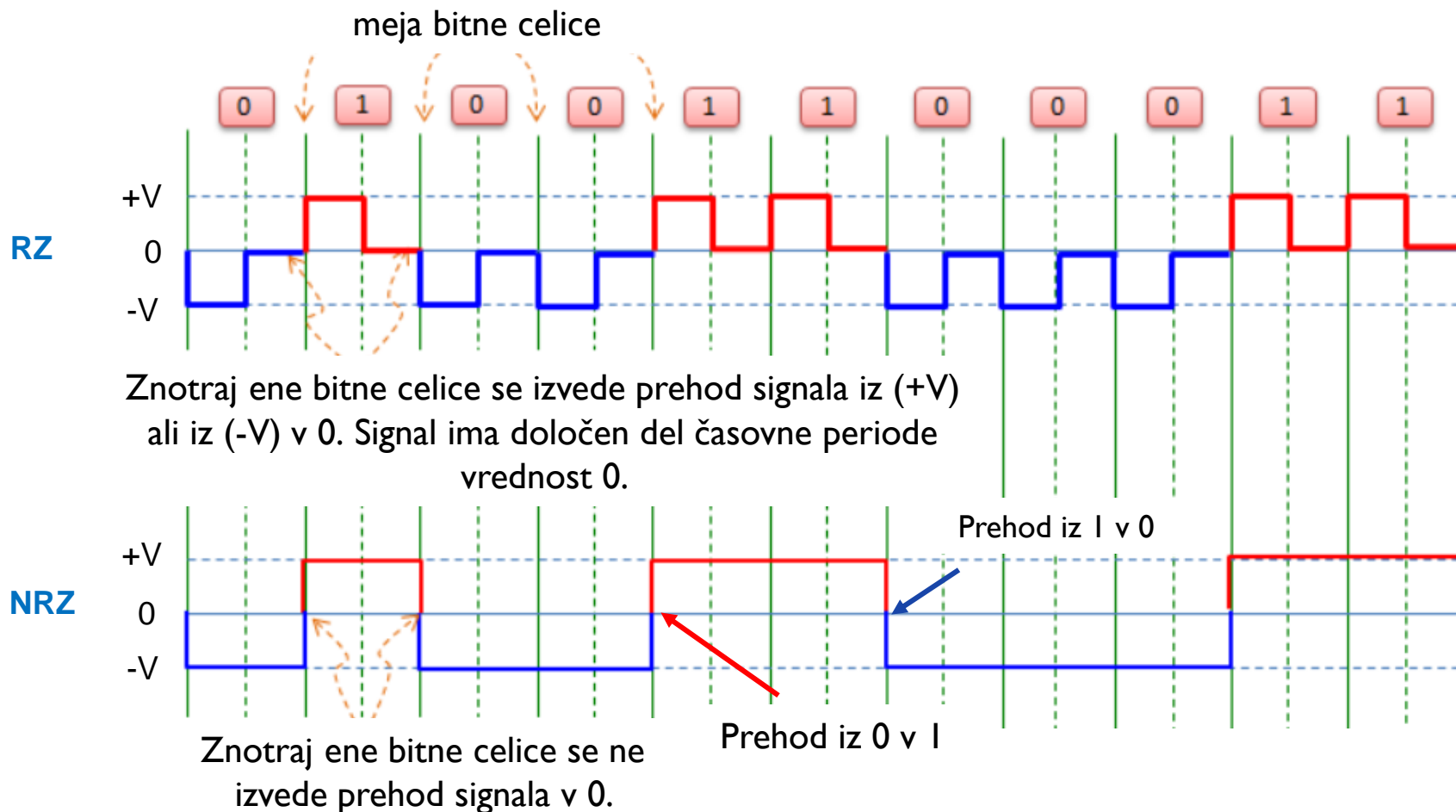
- ❑ DC pomik (daljši nizi 0 ali 1)

## ❑ Prednosti:

- ❑ Več prehodov (sinhronizacija)

## Primer kodiranja RZ, NRZ (električne povezave)

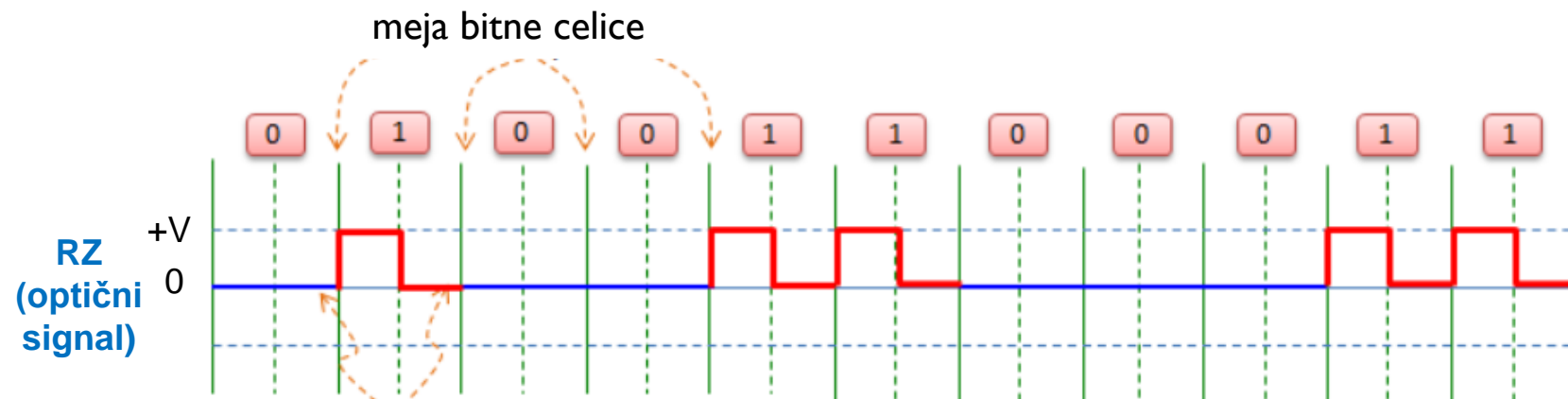
- ❑ **Električni signal** - signal plus (+V) označuje bit 1 in minus (-V) označuje bit 0.



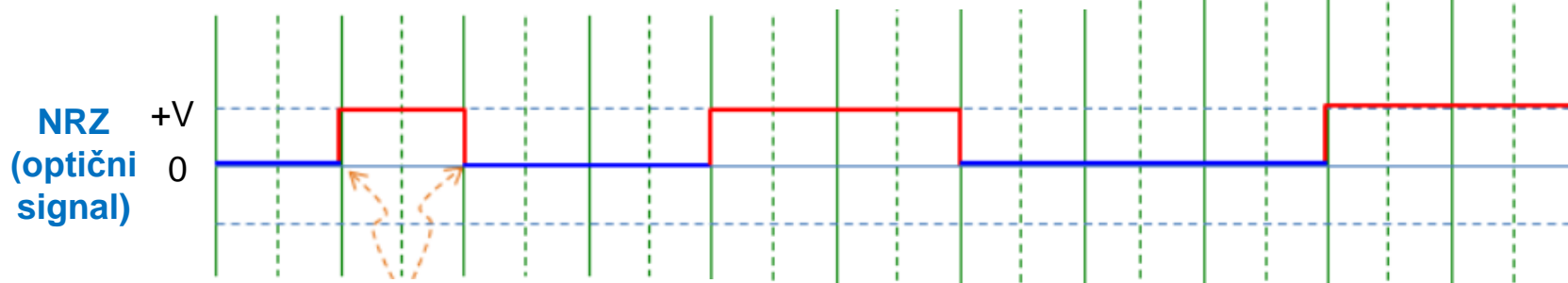


## Primer kodiranja RZ, NRZ (optika)

- 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.

## 8.2.2 Kodiranje NRZ (Non Return to Zero)

### □ več vrst kodiranj NRZ (Non Return to Zero)

Code name	Alternate name	Complete name	Description
<b>NRZ(L)</b>	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. <a href="#">Inverse logic</a> mapping is also a type of NRZ(L) code.
<b>NRZ(I)</b>	NRZI	Non-return-to-zero inverted	Refers to either an NRZ(M) or NRZ(S) code.
<b>NRZ(M)</b>	NRZM	Non-return-to-zero mark	Serializer mapping {0: constant, 1: toggle}.  1 - sprememba nivoja
<b>NRZ(S)</b>	NRZS	Non-return-to-zero space	Serializer mapping {0: toggle, 1: constant}.  0 - sprememba nivoja
<b>NRZ(C)</b>	NRZC	Non-return-to-zero change	

## 8.2.2 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:

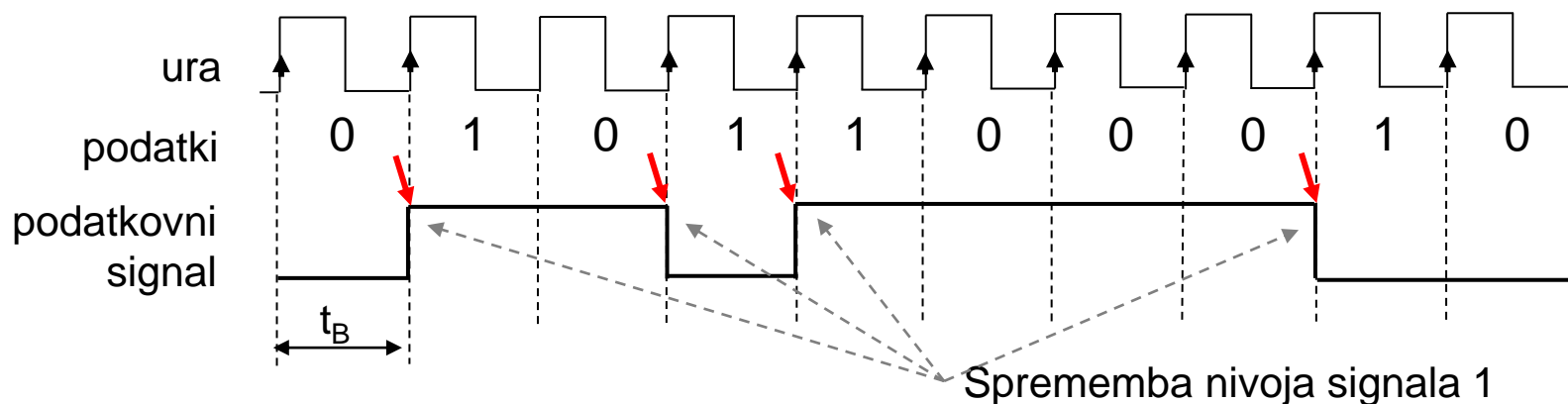
- DC odmik pri daljšem zaporedju ničel (ali enic).
  - Težja sinhronizacija pri daljšem zaporedju ničel (enic).

- Uporaba:

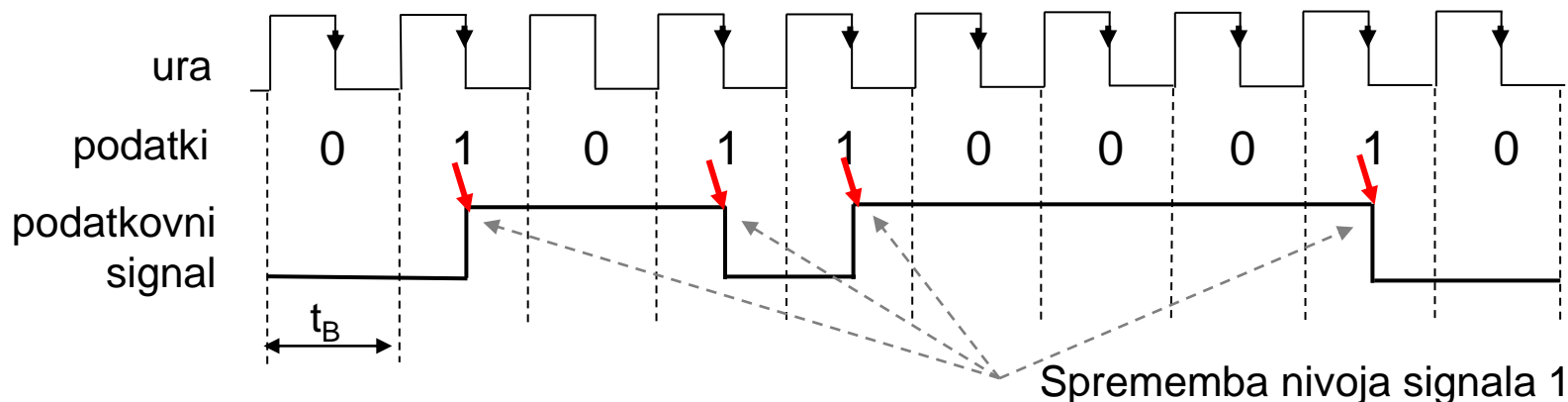
- USB 2.0 in HDLC (pri predolgem zaporedju enic se vstavlja ničla – „bit stuffing“)
  - Skupaj s kodiranjem RLL.

# Primer kodiranja NRZI

- Primer kodiranja NRZI(M) za 10-bitni niz, ki ima 4 logične enice → 4 spremembe nivoja
  - Sprememba nivoja signala se izvede pri vsakem bitu vrednosti 1.
    - a) ob pozitivni fronti ure (0 → 1)



b) ob negativni fronti ure (1 → 0)

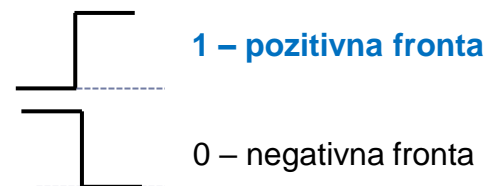


## 8.2.3 Kodiranje PE (Phase Encoded)

- ❑ 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

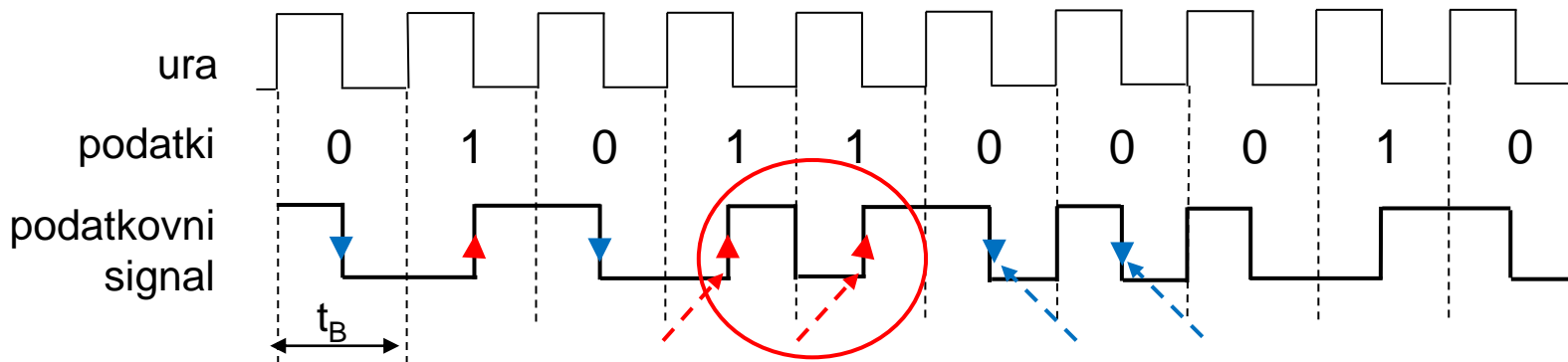


- ❑ **Prednosti:**
  - Odpravlja obe pomanjkljivosti NRZI kodiranja.
- ❑ **Slabosti:**
  - Slaba izkoriščenost kanala (dve spremembi nivoja v bitni celici).
- ❑ **Uporaba:**
  - Ethernet 10BASE-T (10Mb/s IEEE 802.3)

# Primer PE kodiranja – „Manchester“

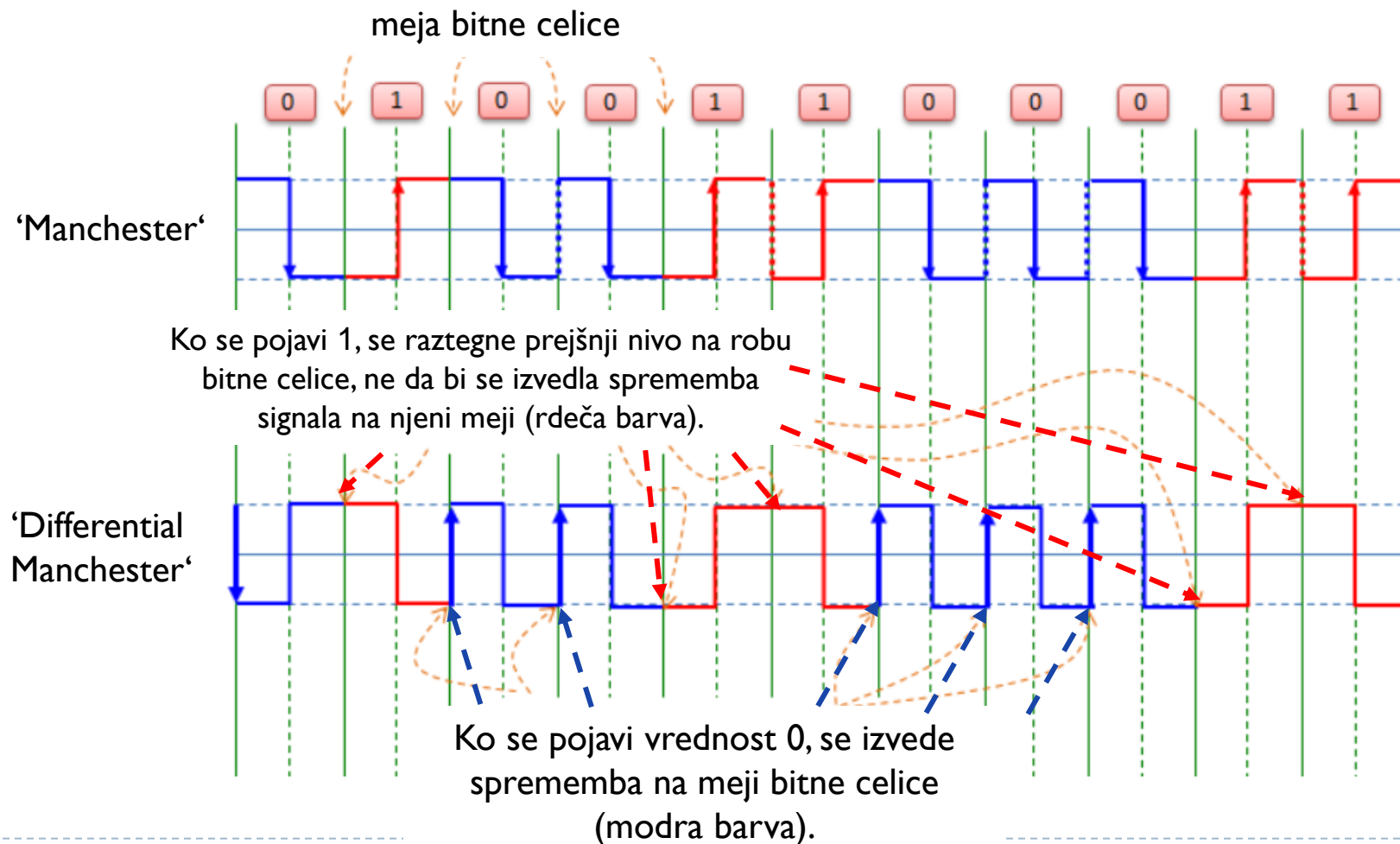
## □ Spremembe signala za posamezen bit:

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



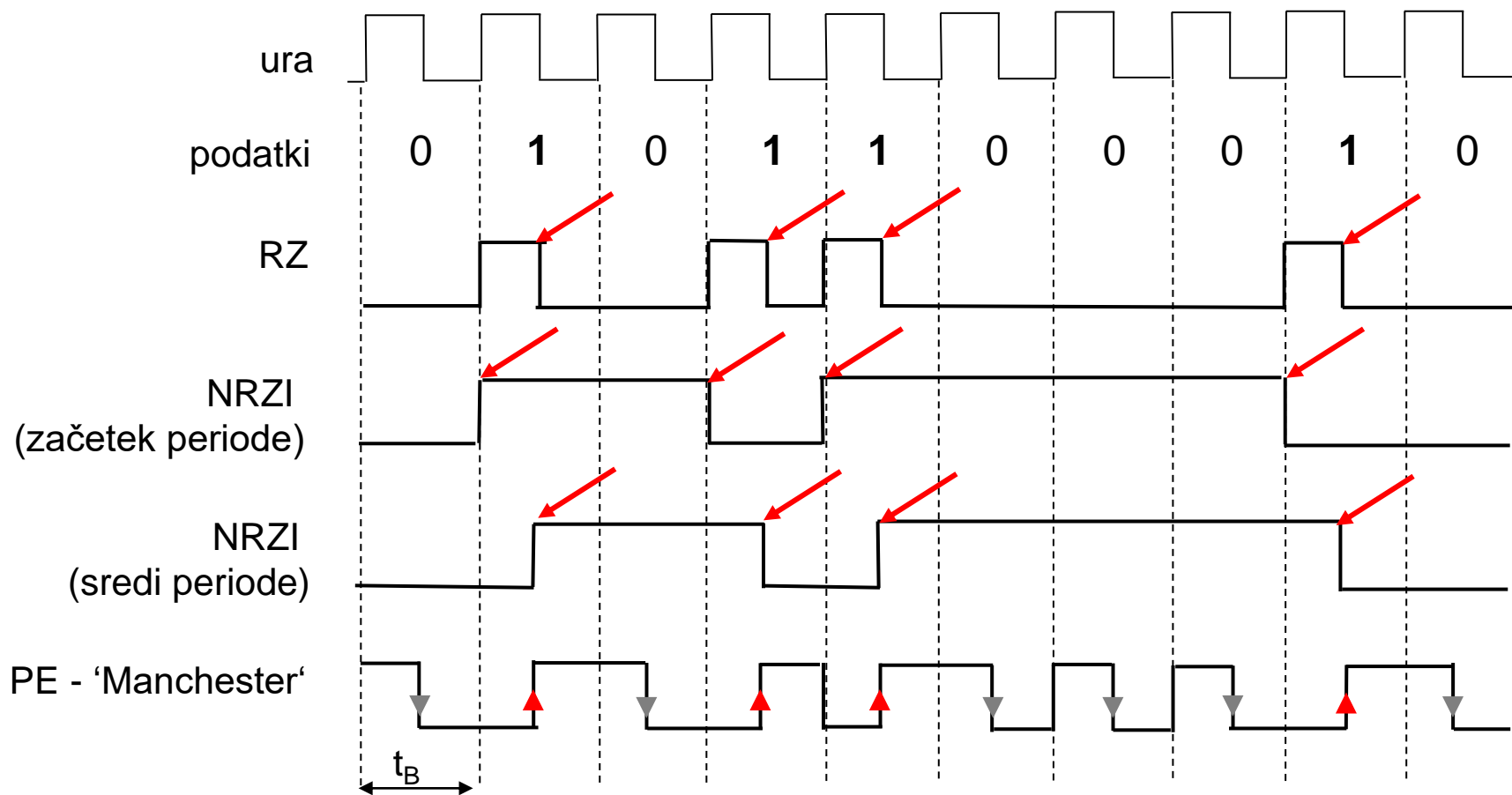
sprememba nivojev se izvede v  
eni bitni celici v sredini periode

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



# Primerjava kodiranja RZ, NRZI in PE (Manchester)

- Prehodi nivojev signala





## 8.2.4 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)

## 8.2.4 Kodiranje RLL (Run Length Limited)

---

- ❑ 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>

## 8.2.4.1 Kodiranje 2/3 (1,7) RLL

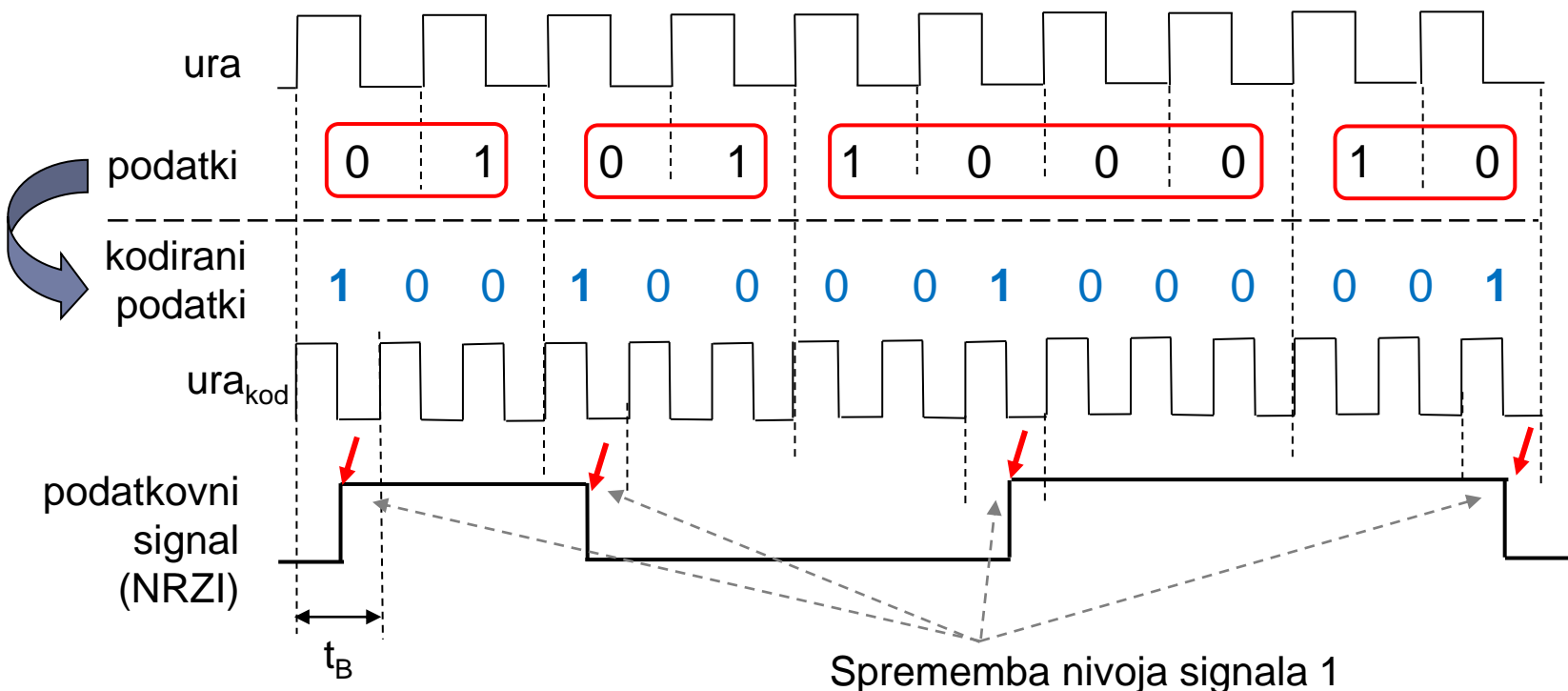
- ❑ Kodiranje 2/3 (1,7) RLL preslika 2 uporabniška bita v 3 kodirane bite tako, da je v kodiranih bitih vedno
  - najmanj 1 ničla in
  - največ 7 zaporednih ničel.
- ❑ Preslikovalna tabela za kodiranje 2/3 (1,7) RLL

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 in NRZI

## □ Primer kodiranja 2/3 (1,7) RLL

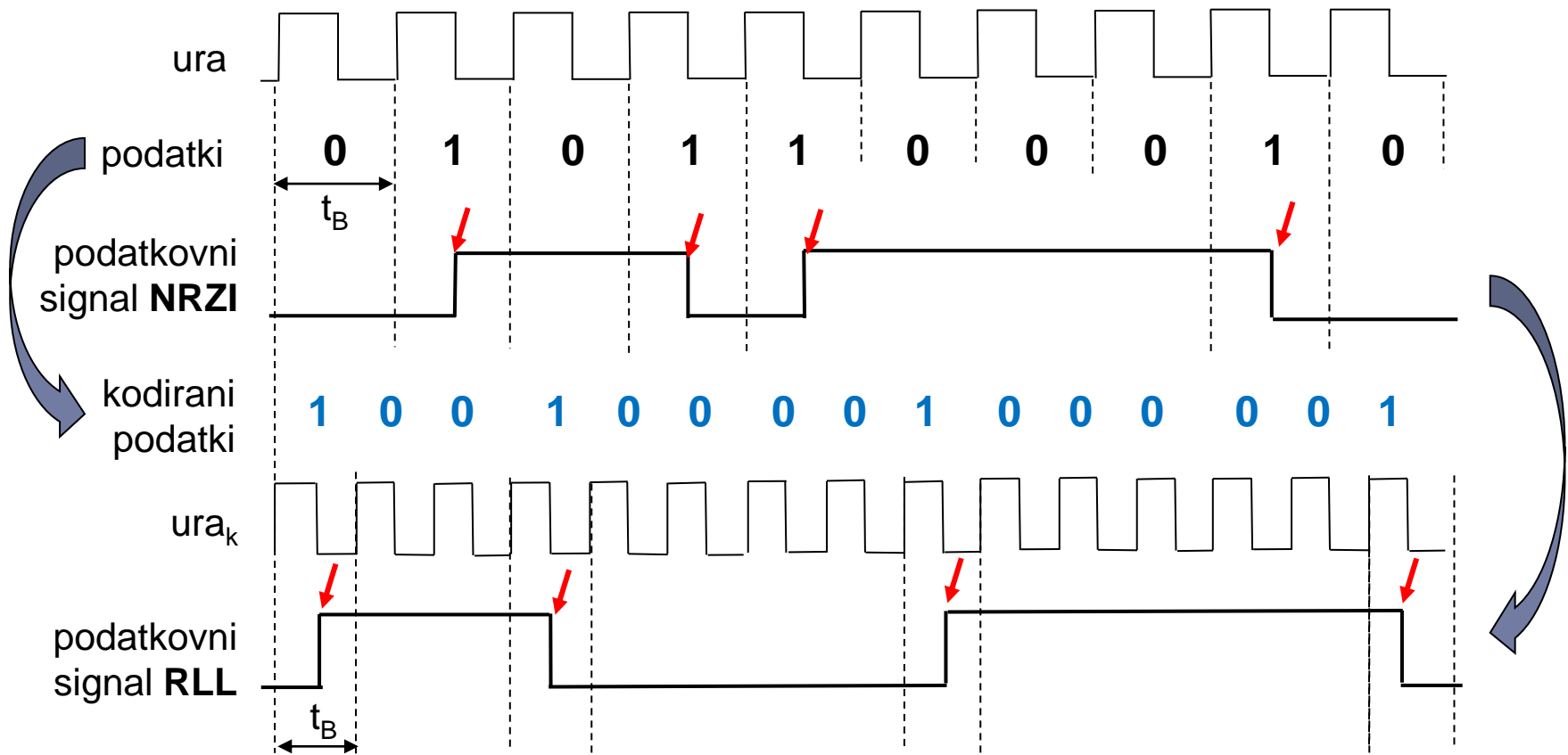
- 2 podatkovna bita → 3 kodirane bite
- 4 podatkovni biti → 6 kodiranih bitov
- 15 bitnih celic, NRZI – sprememba signala ob pojavu enic s sredini periode



# Primer kodiranja 2/3 (1,7) RLL in NRZI

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

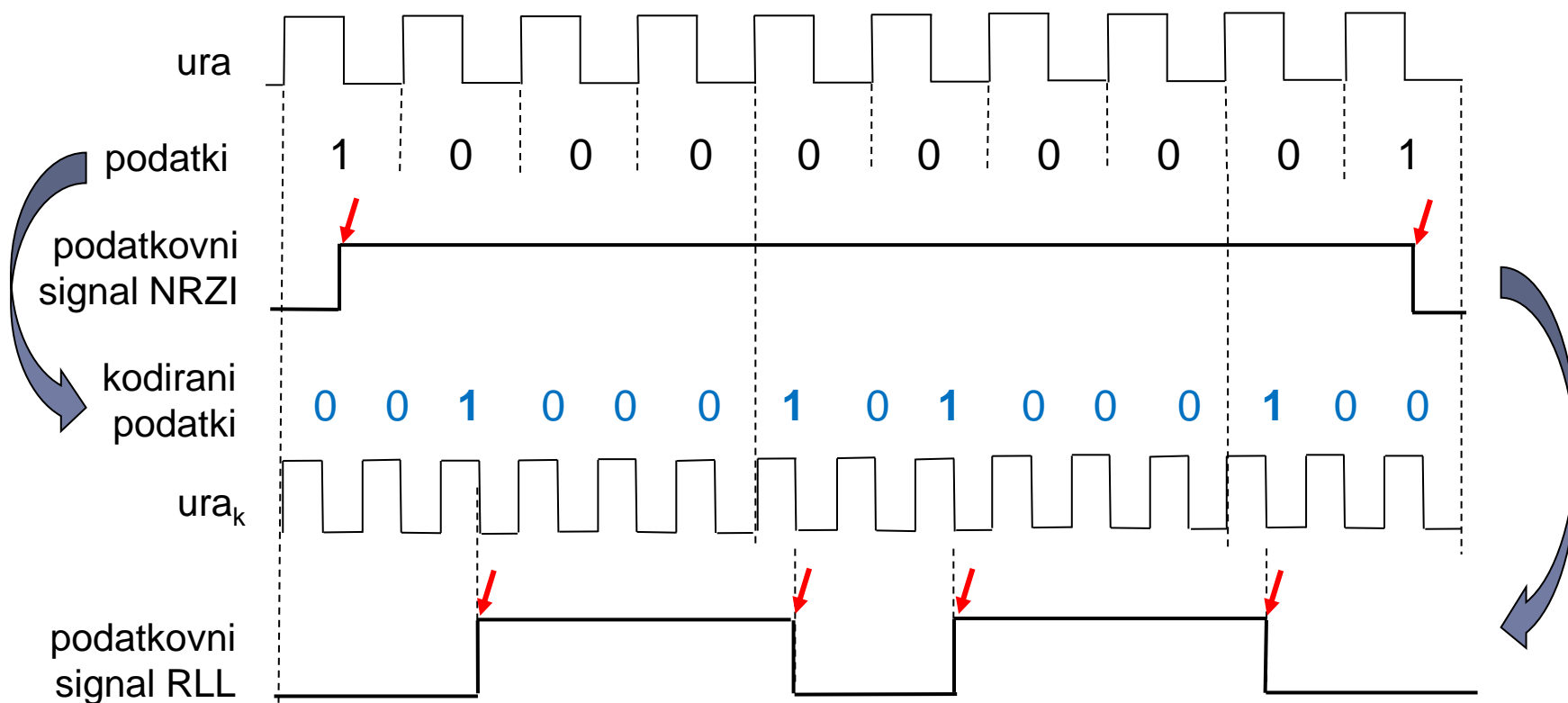
- Podatkovni signal NRZI določajo podatkovni biti
- Podatkovni signal RLL določajo kodirani biti in NRZI



# Primer kodiranja 2/3 (1,7) RLL in NRZI

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

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



# Uporaba RLL kodiranja

---

- ❑ 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

## 8.2.4.2 Kodiranje 8b/10b

- ❑ 8 podatkovnih bitov (8b) na vhodu se pretvori v 10 kodiranih bitov (10b) na izhodu.
- ❑ Za izvedbo sta uporabljeni dve enostavnejši kodiranja:
  - **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).



# Kodiranje n/m - uporaba

---

## □ 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:

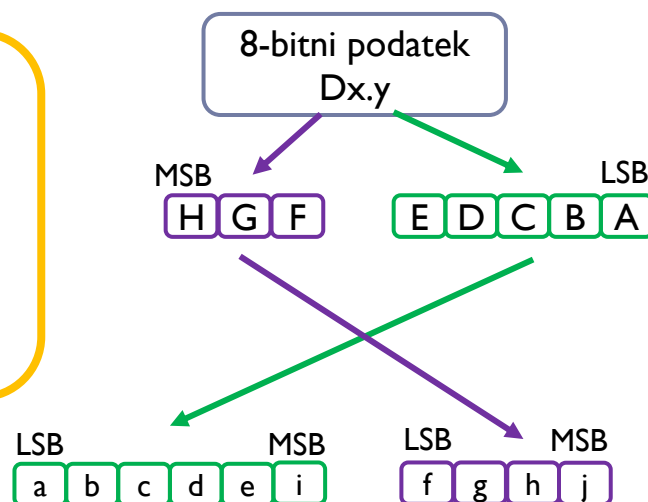
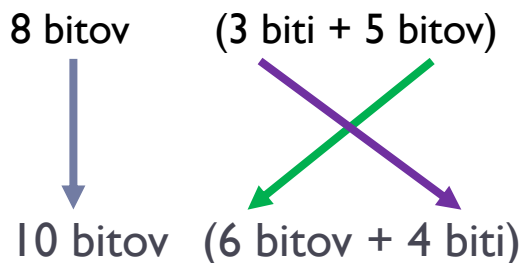
Na enojni povezavi PCIe 2.0 x1 je pri frekvenci ure 2,5GHz hitrost prenosa enaka 2,5Gb/s. Kakšna je efektivna hitrost prenosa zaradi kodiranja 8B/10B?

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

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

## 8.3 Očesni vzorec (ang. eye pattern)

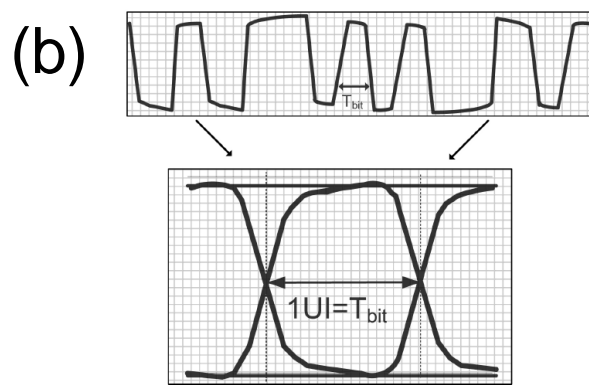
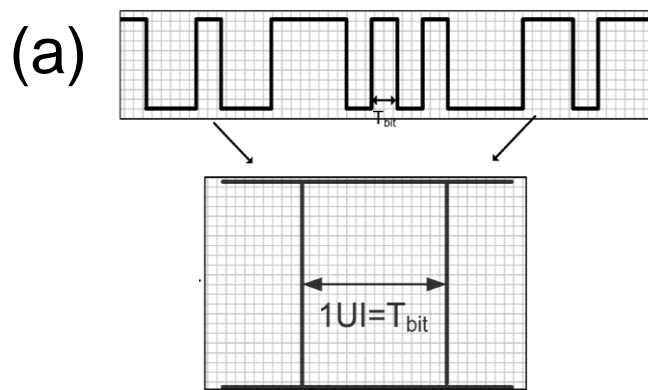
---

Običajne meritve kakovosti (integritete) signalov :

- ▶ Meritve na izhodu linije (bremenu)
  - ▶ Oblika, čas vzpona-padca, zvonjenje, druge popačitve signala
- ▶ BER (Bit Error Rate)
  - ▶ SW: Programska meritve
- ▶ **Očesni vzorec**
  - ▶ Zajema sinergijske učinke večih dejavnikov

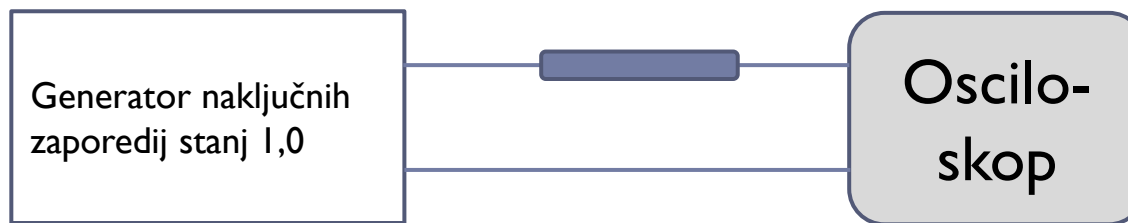
## 8.3 Očesni vzorec (ang. eye pattern)

- ❑ Metodologija, ki **analizira prenos digitalnega signala pri visokih hitrostih**.
- ❑ Omogoča, vizualizacijo in določanje ključnih parametrov električne kakovosti signala.
- ❑ Očesni vzorec ali diagram je izdelan iz digitalne valovne oblike signala, ki ustreza vsakemu posameznemu bitu in je predstavljen v grafu.
- ❑ Amplituda signala je na navpični osi, čas pa na horizontalni osi.
- ❑ Primer: Očesni vzorec za (a) idealen digitalni signal in (b) realen signal



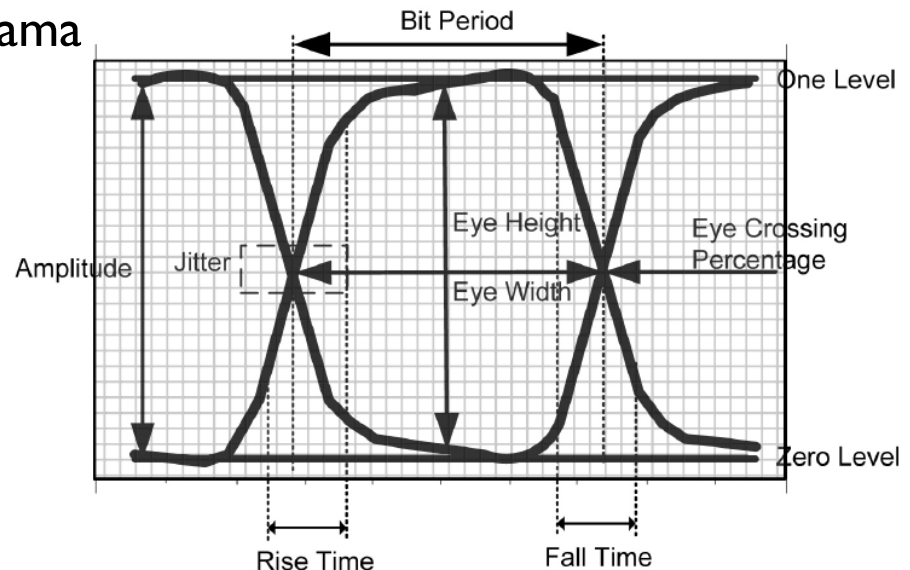
- ❑  $T_{UI}$  – čas trajanja signalnega elementa (UI – unit interval)  $\equiv T_{\text{bit}}$
- ❑ Vir: <https://www.onsemi.com/pub/Collateral/AND9075-D.PDF>

- Za izvedbo meritev očesnega vzorca potrebujemo
  - psevdonaključni generator digitalnega signala, prenosno linijo in osciloskop.



- Rezultati meritve očesnega diagrama

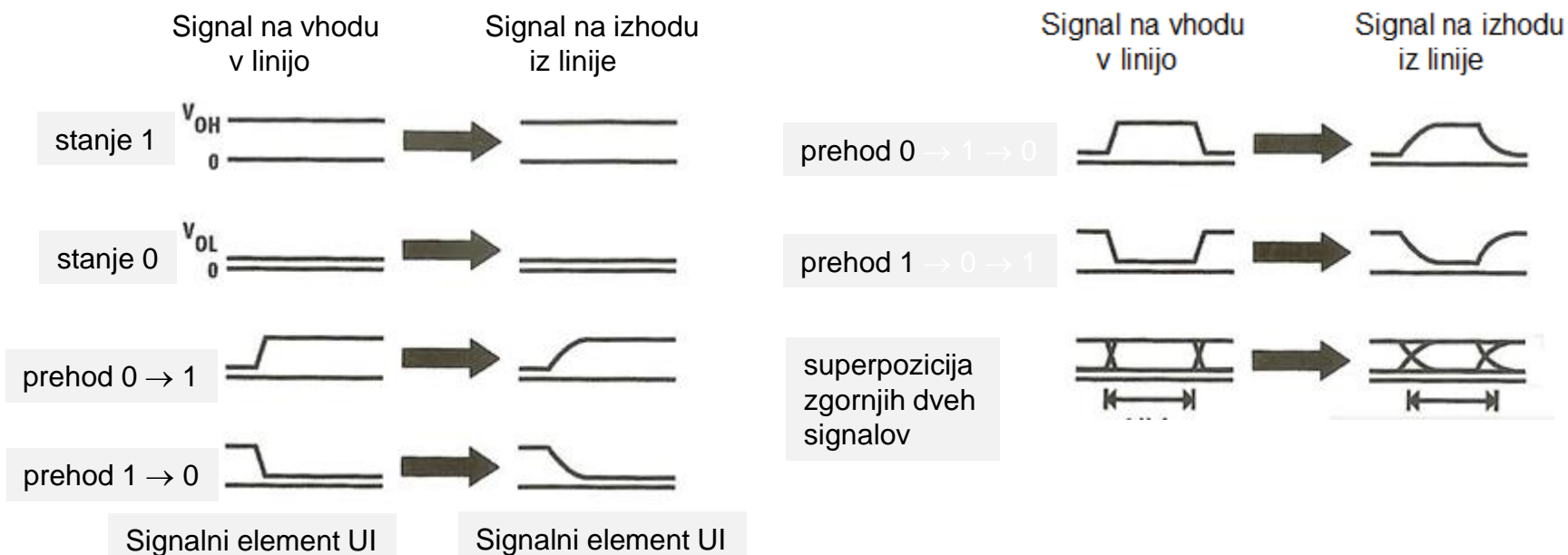
- Rise Time, Fall Time
- Zero Level, One Level
- Eye Hight, Eye Width
- Eye Crossing percentage
- Bit Period
- Amplitude
- Jitter



## □ Iz diagrama očesnega vzorca lahko:

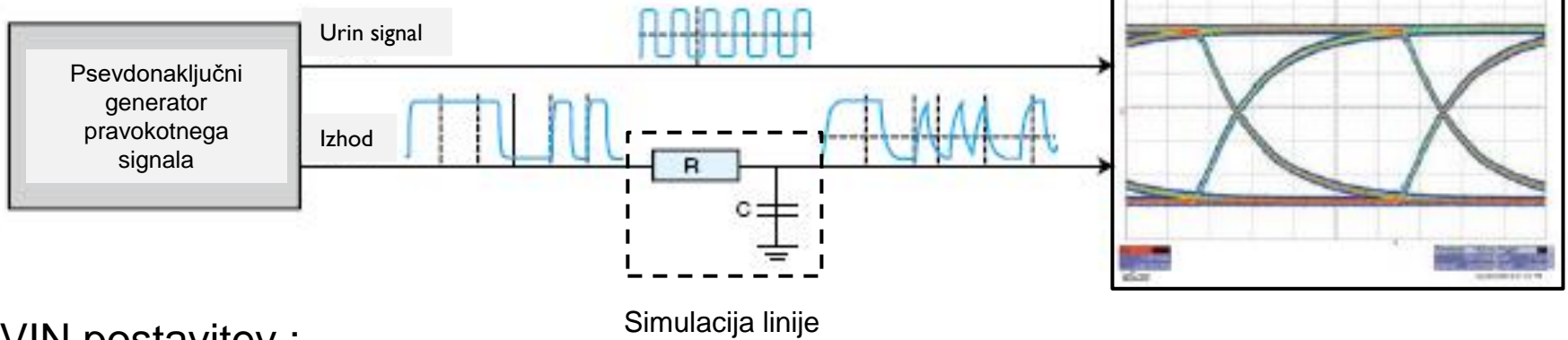
- izmerimo najvišjo možno hitrost prenosa na določeni prenosni poti ali kanalu.
- določimo najkrajši možni čas signalnega elementa ( $T_{UI}$ ), da je sprejem še možen.
- izmerimo tresenje (ang. jitter).
- določimo število možnih napetostnih nivojev, to je število bitov v signalnem elementu

## □ Očesni vzorec generiramo s superpozicijo signalov

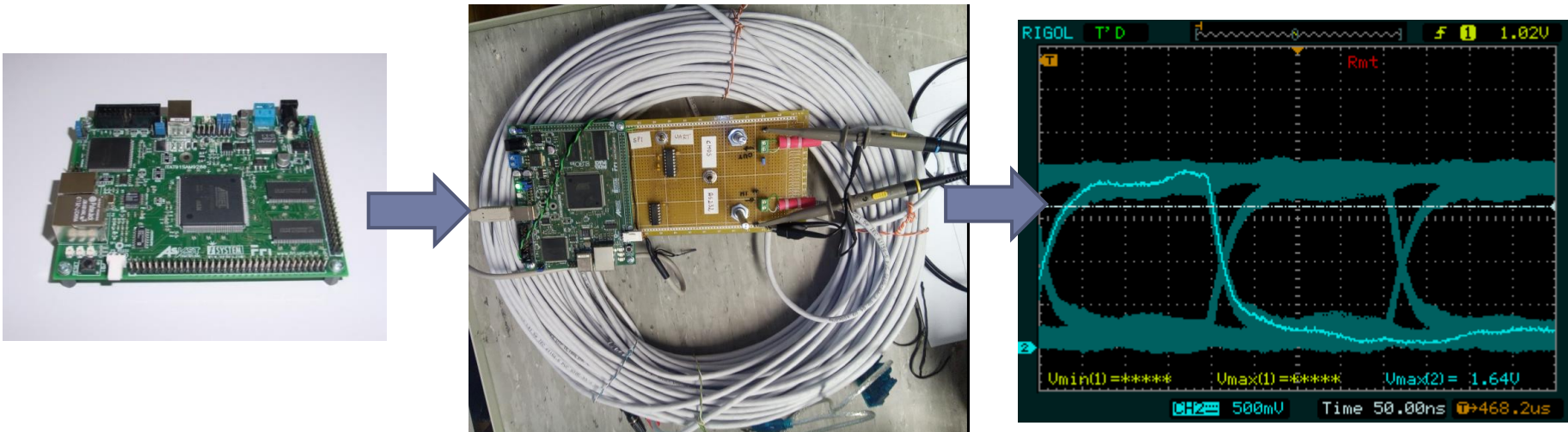


## □ Shema meritve očesnega vzorca

Splošna postavitev :



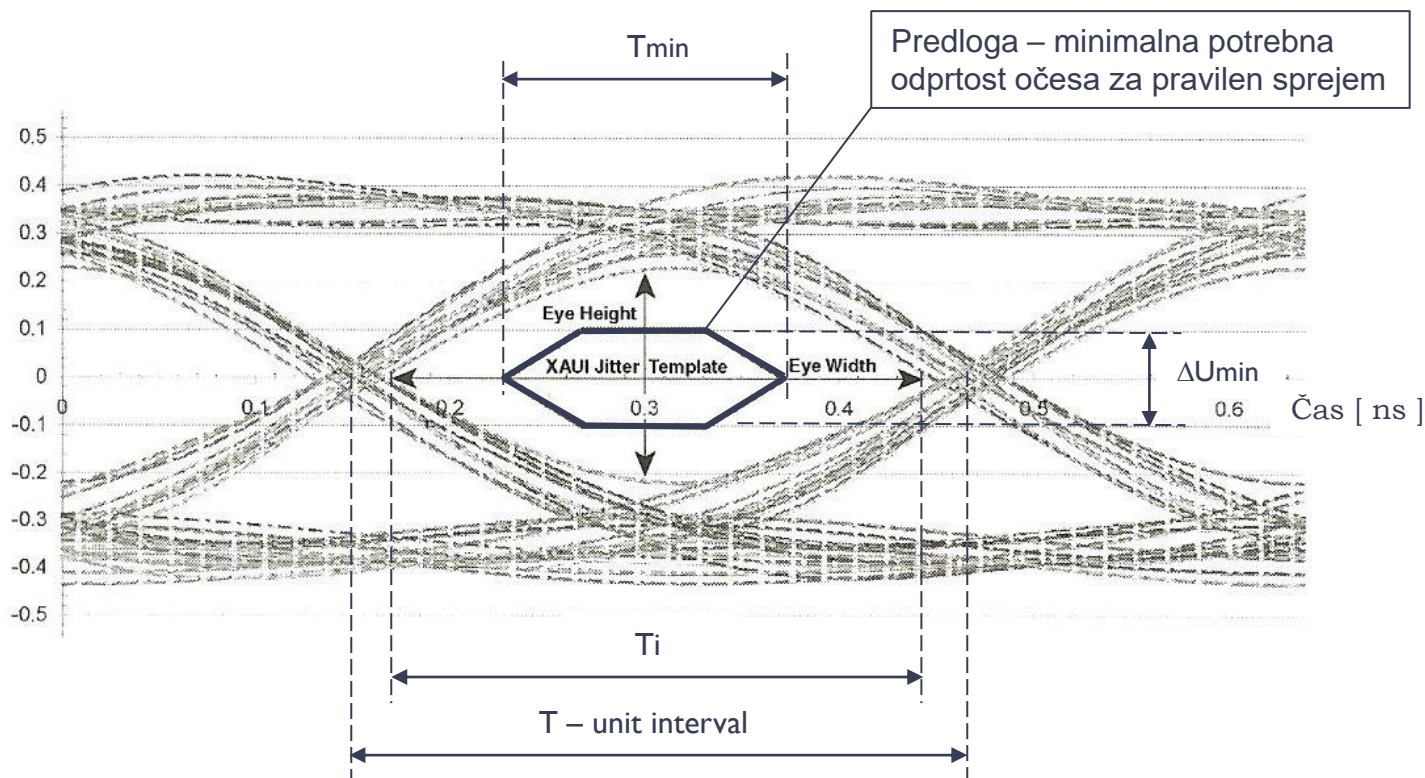
VIN postavitev :





## Primeri meritev očesnega vzorca

Očesni vzorec s predlogo za 10-Gigabit Ethernet sprejemnik, ki preklaplja med  $\pm 100$  mV



Tom Granberg, Digital Techniques for High-Speed Design

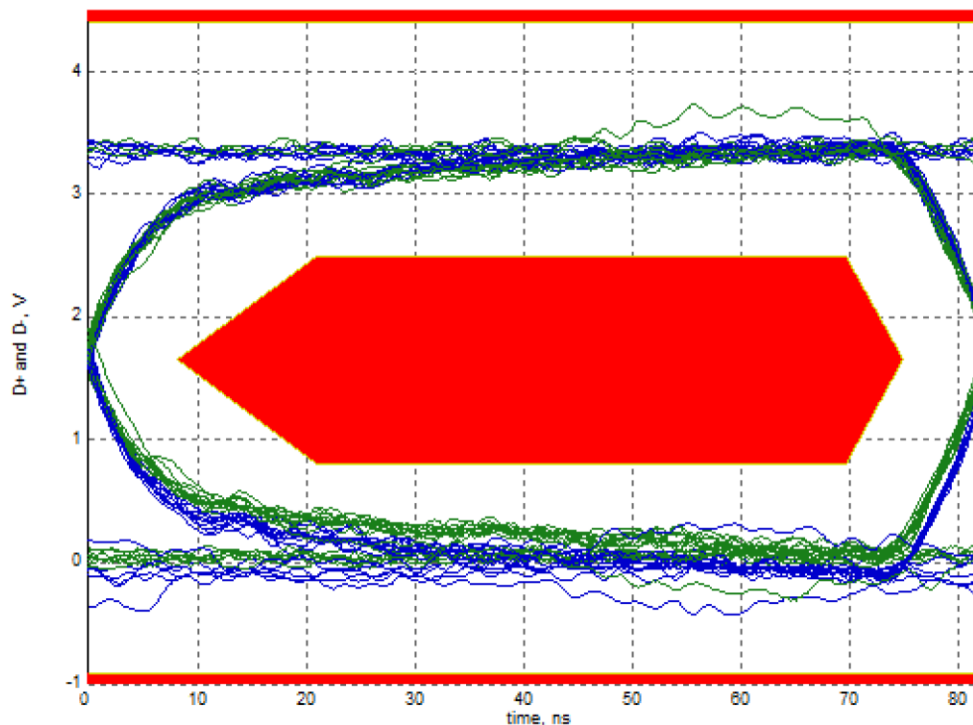
Sprejem ni več možen, ko se

- $T_i$  skrajša pod minimalno vrednost  $T_{min}$  (širina očesa se krajša).
- napetostni nivoji znižajo pod mejo ločljivosti (višina očesa se zmanjša pod  $\pm 100$  mV).

## □ Primeri meritev očesnega vzorca - USB

# Measuring signal quality ■

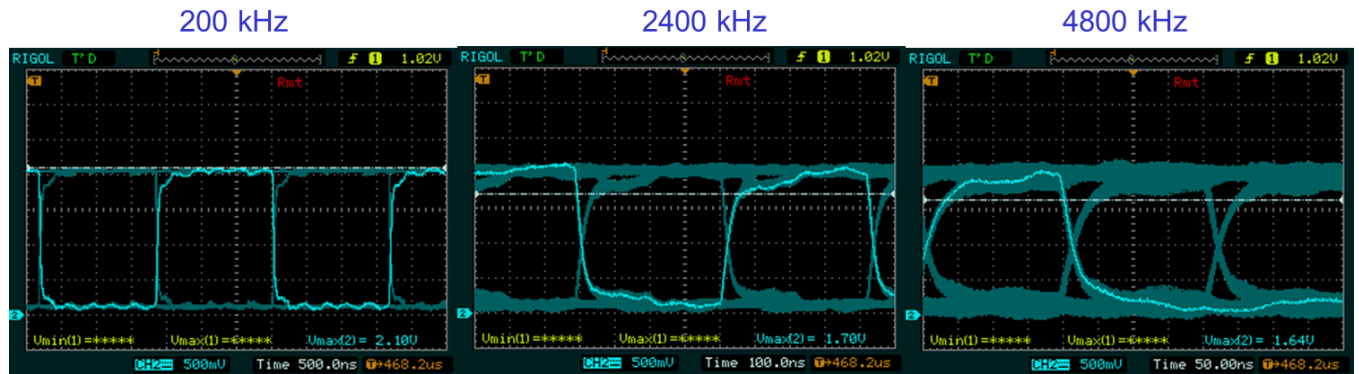
- USBET20 PC tool – can be downloaded at <http://www.usb.org/>
- Loads CSV file with signals
  - Should be measured on the other end of the USB cable
- Generates eye diagram
- Checks timing
- Checks voltage levels
- Inrush current measurement



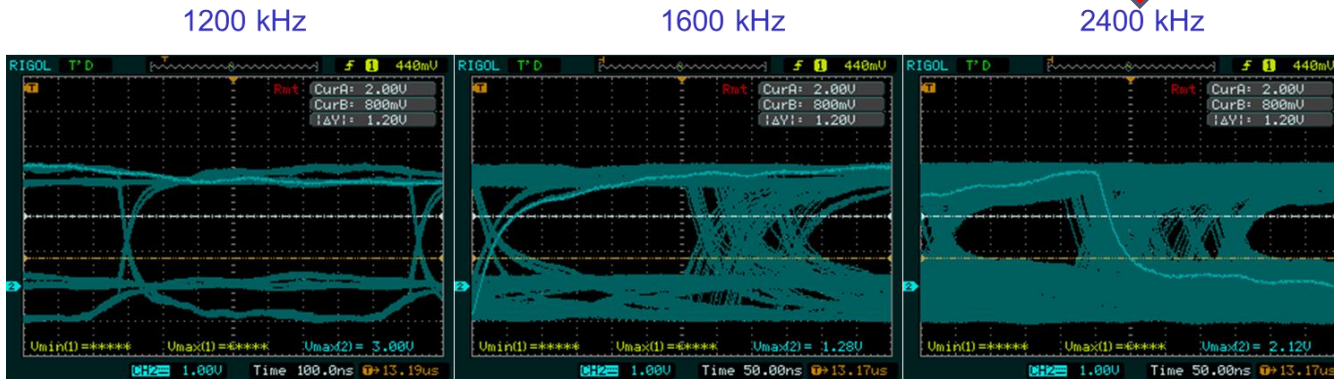
# Primeri meritev očesnega vzorca – VIN-LAB merilni kabli

Oddajnik CMOS – merimo izhod linije !

Zaključena linija



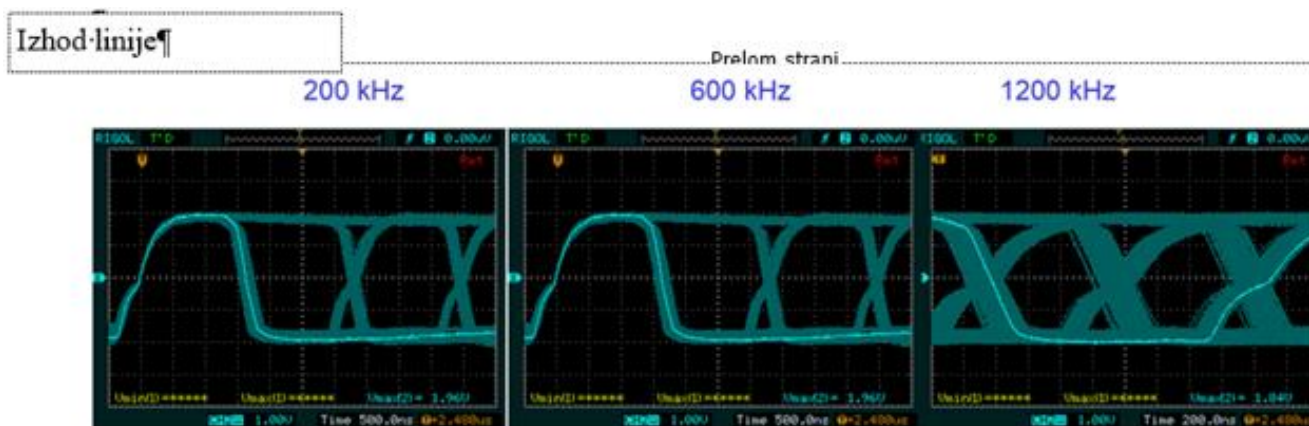
Nezaključena linija



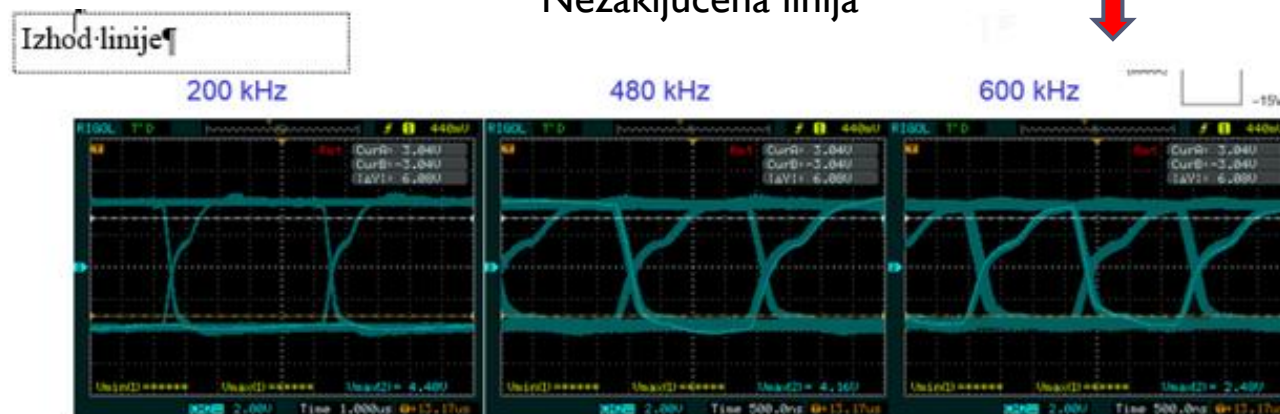
## Primeri meritev očesnega vzorca – VIN-LAB merilni kabli

Oddajnik RS232 – merimo izhod linije !

Zaključena linija

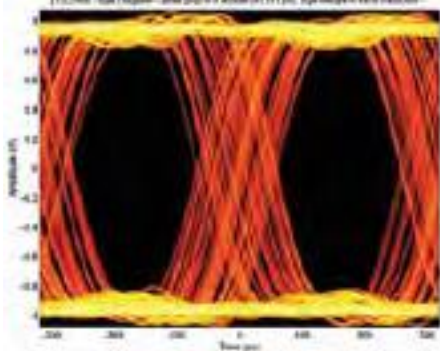
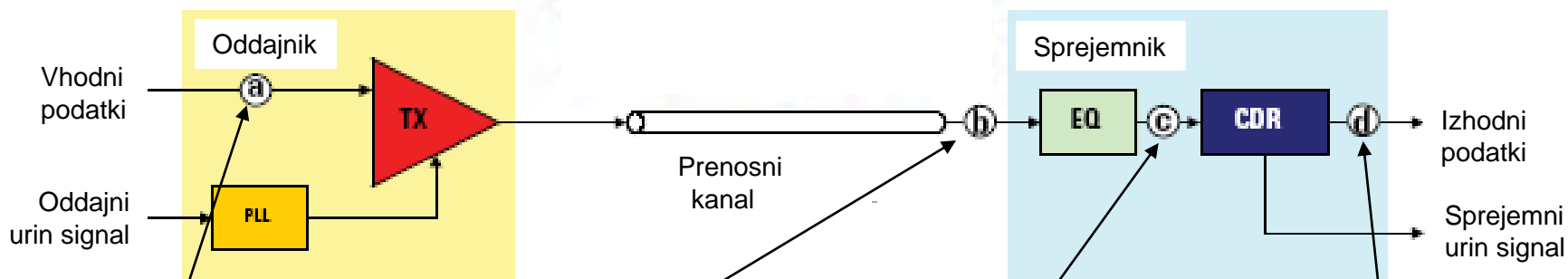


Nezaključena linija

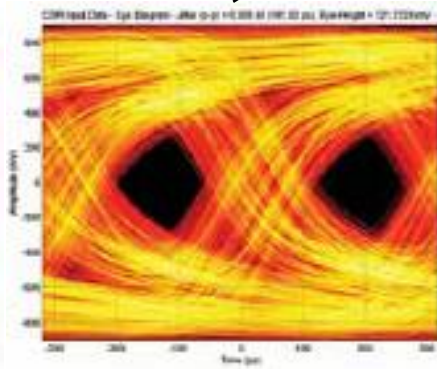




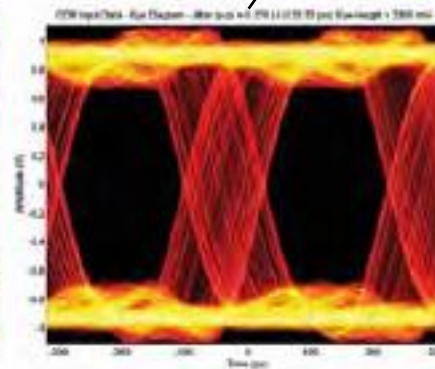
## Primeri izboljšav prenosa podatkov s pomočjo očesnega vzorca



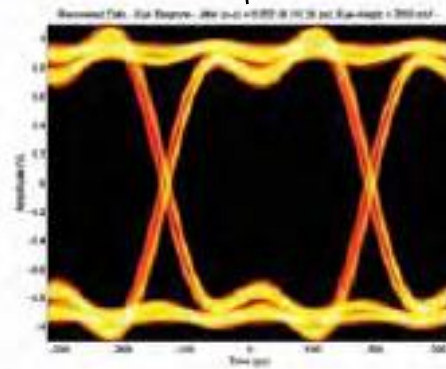
Podatkovni signal z urinim signalom  $f=20\text{MHz}$ , zamik=0.4UI



Podatkovni signal na koncu linije, na vhodu v sprejemnik

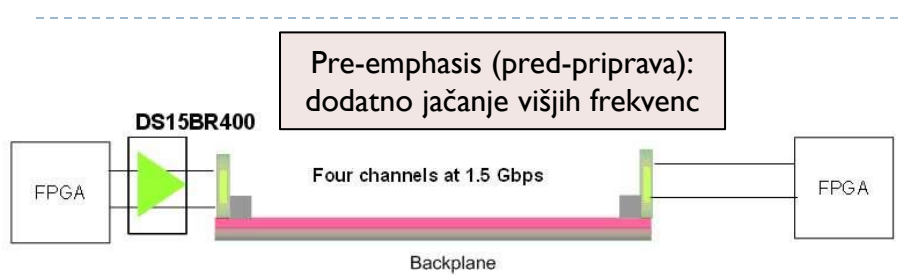


Signal po izenačevalniku (EQ- Equalizer)

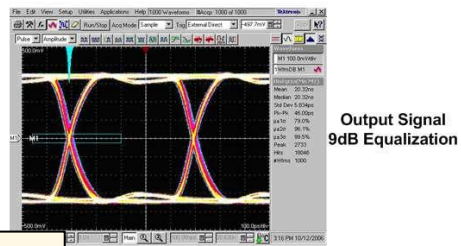
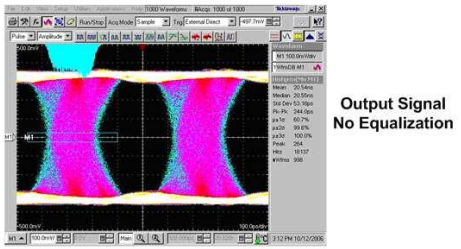
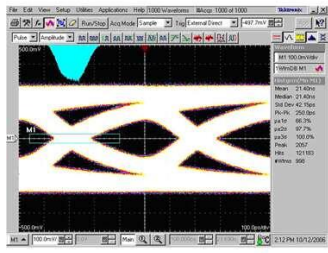
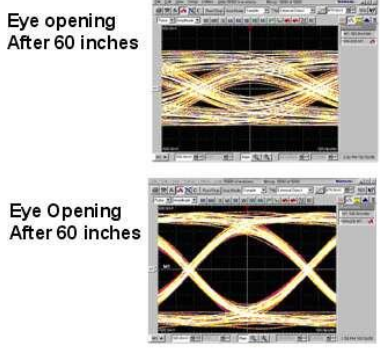
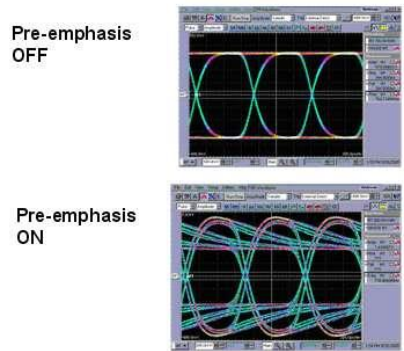


Po restavriranju urinega in podatkovnega signala (CDR - Clock and Data Recovery)

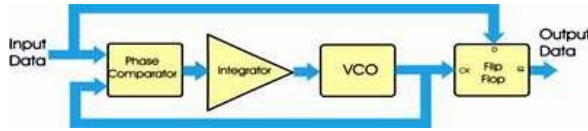
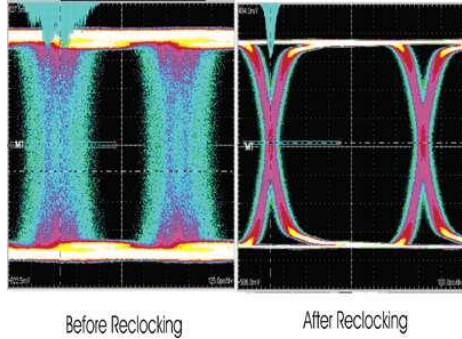
# Primeri izboljšav prenosa podatkov s pomočjo očesnega vzorca



Equalization (po-priprava): adaptivni HP filter – obraten učinek kot linija (če znana dolžina)



Reclocking (sinhronizacija sprejemne ure s frontami signala)



Vir: <https://www.edn.com/reclocking-and-equalization-for-restoring-hd-digital-video-with-low-jitter/>