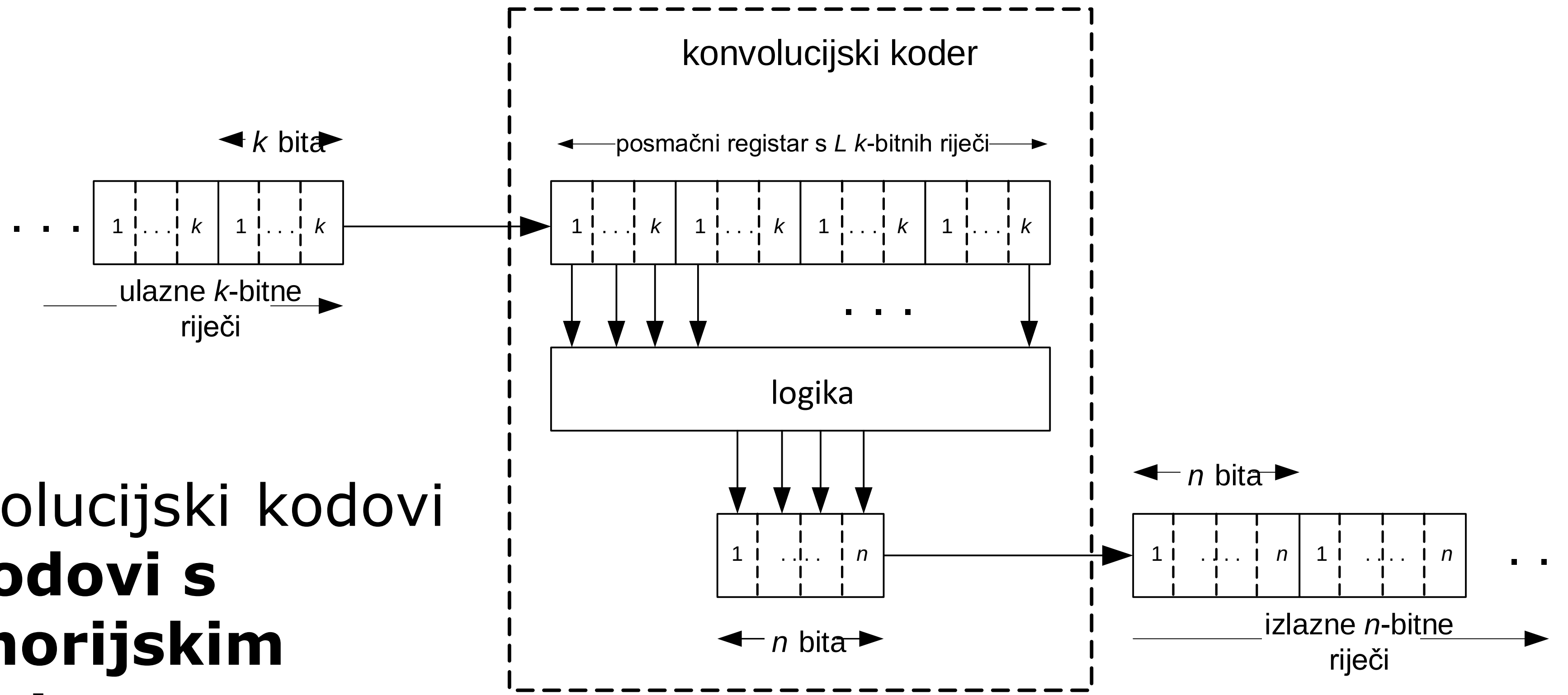
Zaštitno kodiranje III
– konvolucijsko kodiranje

Teorija informacije

Općeniti model konvolucijskog kodera

- konvolucijski koder izgrađen je od
  - posmačnih registara i logike (zbrajači u aritmetici modulo 2)



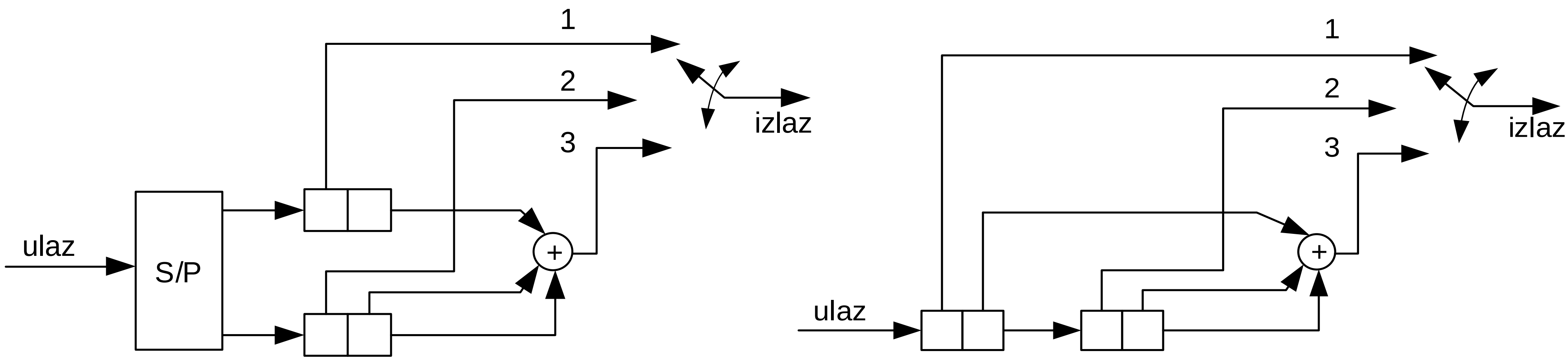
koder za konvolucijski kôd (n, k, L)

- konvolucijski kodovi su **kodovi s memorijskim svojstvom**
  - blok kodovi su bezmemorijski

Općeniti model konvolucijskog kodera (II)

- na ulazu podaci organizirani u okvire duljine k bita
- na svaki brid takta koji upravlja radom kodera k bita se pomiče u desno
- time je svaki put generirano n bita na izlazu, $n > k$
- takav odnos osigurava zalihost u kodu
- koder ima i memoriju, svaki slijed on n bita na izlazu ovisi o sadašnjem kao i o $L - 1$ prethodnih ulaznih okvira, $L > 1$
- L se naziva *constraint length* – granična duljina kodera
  - kod komercijalnih kodera $L \leq 10$
- za kodnu brzinu R vrijedi izraz: $R = k/n$
- primjer: $k = 1$, $L = 3$ i $n = 2$
- svaki bit s ulaza ($k = 1$) generira 2 bita na izlazu ($n = 2$)
- dakle, $R = 1/2$
- u praksi: k i n su mali cijeli brojevi, a L se mijenja u svrhu upravljanja redundancijom koda

Koder sa serijskim i paralelnim ulazom



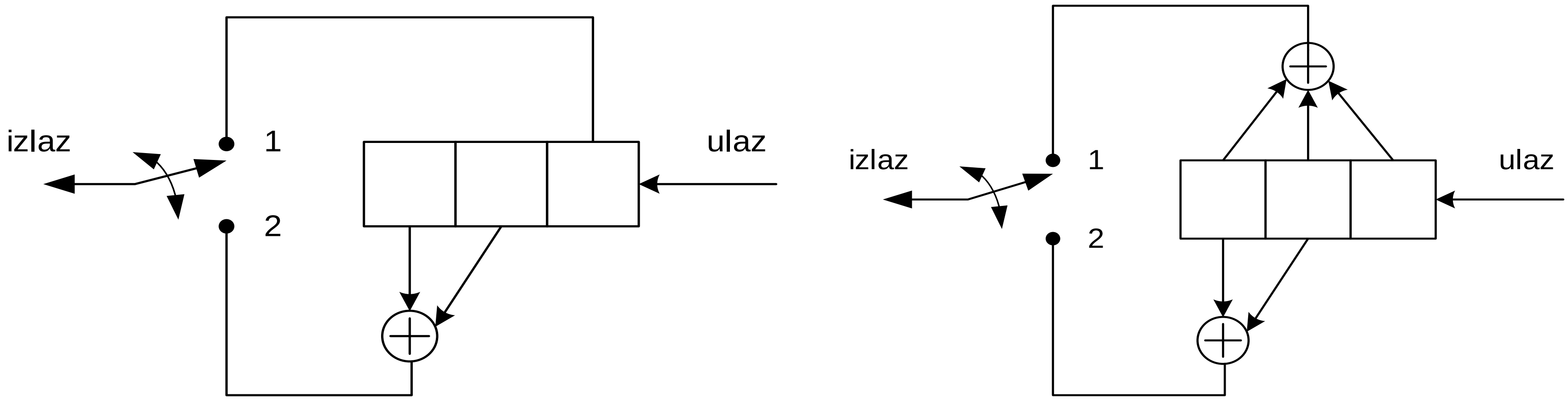
a) b)

- konvolucijski koderi, $k = 2$, $n = 3$
- a) paralelni ulaz
  - S/P – serijsko-paralelna pretvorba slijeda bita
- b) serijski ulaz

Veze u koderu

- veze povezuju zbrajače i izlaze bistabila registra
  - te veze se ne odabiru proizvoljno
  - one utječu na udaljenost koda
  - dobri kodovi su pronađeni za sve $L < 20$
- za razliku od blok kodova, konvolucijski kodovi nemaju veličinu bloka
  - problem je što na kraju ulaznog slijeda treba dodati bitove u stanju nula kako bi se posmačni registar u koderu „očistio” od podatkovnih bita
  - dodane nule ne predstavljaju korisnu informaciju pa je $R < k/n$

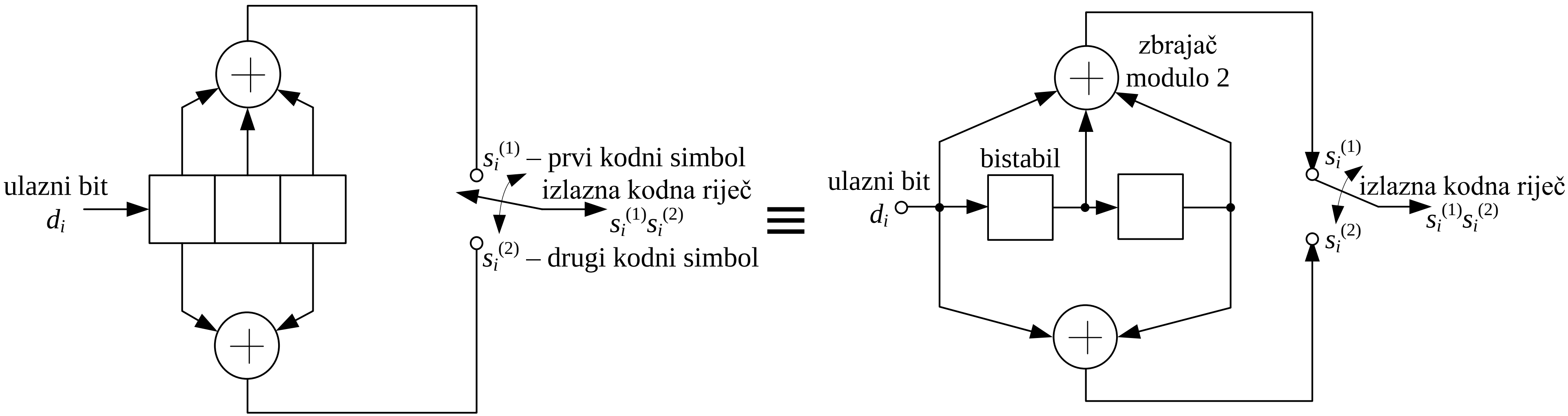
Sistematski i nesistematski koder



a) b)

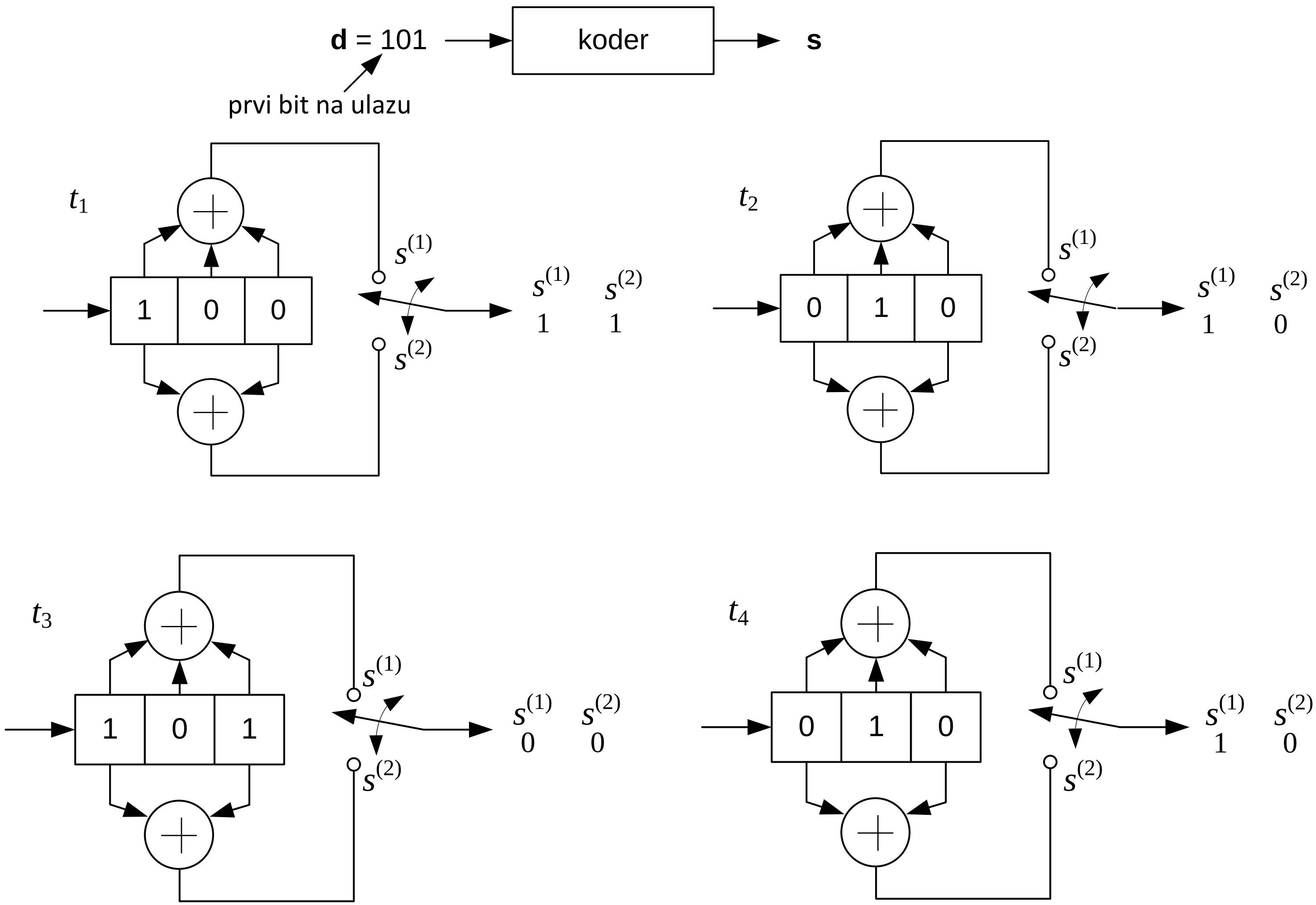
- a) sistematski koder (k = 1, n =2)
  - ulazna k-torka pojavljuje se na izlazu kodera kao dio n-torke pridružene toj k-torci
  - smanjuje se Hammingova udaljenost između kodnih riječi
- b) nesistematski koder (k = 1, n = 2)

Primjer konvolucijskog kodera za $k = 1$ i $n = 2$

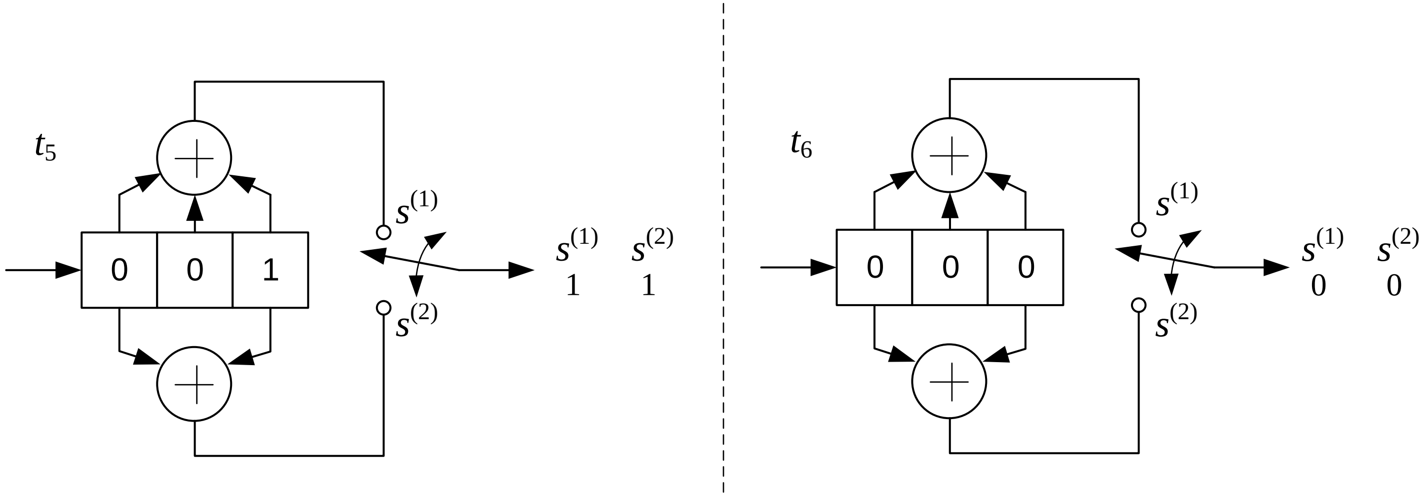


- primjer: konvolucijski koder, $R = 1/2$, $L = 3$
- na izlazu kodera je komutator (*switch*), tj. sklopka koja prvo uzme uzorak izlaza gornjeg zbrajača (bit $s_i^{(1)}$), a onda donjeg (bit $s_i^{(2)}$)
  - uzorkovanje se ponavlja za svaki ulazni bit d_i
  - od svakog bita d_i nastaju dva bita na izlazu: $s_i^{(1)}s_i^{(2)}$
  - izlaz ovisi o d_i kao i o d_{i-1} i d_{i-2} (zbog $L = 3$)

Primjer rada kodera R = 1/2, L = 3



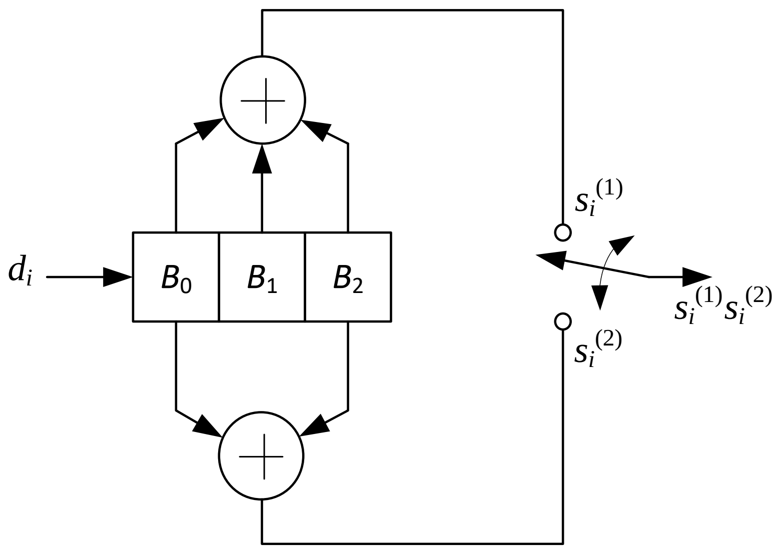
Primjer rada kodera $R = 1/2$, $L = 3$ (II)



- izlazni slijed **s** = 1110001011
- t_1 do t_6 su vremenski trenuci
- u t_4 i t_5 su na ulaz dovedeni simboli 0
  - dodane su $L - 1 = 2$ nule
  - u t_5 je koder „očišćen”
  - bit nove poruke može doći na ulaz u t_6

Impulsni odziv kodera

- odziv kodera na bit 1 na njegovom ulazu
- ulazni slijed **d** = 100
- izlazni slijed **s** = 111011 – impulsni odziv



d_i	B_0	B_1	B_2	$s_i^{(1)}$	$s_i^{(2)}$
1	1	0	0	1	1
0	0	1	0	1	0
0	0	0	1	1	1

- impulsni odziv izlaza $s^{(1)}$ je 111, a izlaza $s^{(2)}$ 101

Impulsni odziv kodera (II)

- izlaz kodera na bilo koji ulazni slijed možemo promatrati kao superpoziciju vremenski pomaknutih impulsnih odziva
  - to pokazuje da su konvolucijski kodovi linearni
- primjer:

$$
101 * 111 = \begin{array}{lccccc}
1 \cdot 111 = & 1 & 1 & 1 & & \\
0 \cdot 111 = & & 0 & 0 & 0 & \\
1 \cdot 111 = & & & 1 & 1 & 1 \\
\sum & 1 & 1 & 0 & 1 & 1
\end{array}
$$

Ulaz **d**	Izlaz **s**				
1	11	10	11		
0		00	00	00	
1			11	10	11
⊕	11	10	00	10	11

  - pri tome je na izlazu $s^{(1)}$ slijed 11011 – to je konvolucija slijeda **d** = 101 i slijeda 111, tj. impulsnog odziva od $s^{(1)}$
  - isto vrijedi i za izlaz $s^{(2)}$: konvolucija slijeda **d** = 101 i slijeda 101, tj. impulsnog odziva od $s^{(2)}$ daje 10001

Impulsni odziv kodera (III)

- izlaz kodera je isti kao i na slajdu 9
- pokazuje da je konvolucijski kod linearan
  - izlaz kodera je konvolucija ulaznog slijeda i impulsnog odziva kodera – otuda i naziv
- u primjeru kodera sa slajda 7 vrijedi:
  - efektivna kodna brzina $R = 3/10 < 1/2$ $(= k/n)$
  - na primjer, za poruku duljine 300 bita, izlazna kodna riječ imala bi duljinu 604 bita
  - $R = 300/604$ – puno bliže 1/2 od 3/10

Prikaz kodera pomoću vektora veza

- za koder se definira n vektora veza
- **vektor veze = funkcijski generator** (FG) h_i
  - i-ti izlaz ($1 \leq i \leq n$), po jedan FG za svaki zbrajač
- uz serijski ulaz, svaki vektor ima dimenziju $1 \times L \cdot k$
  - za $k = 1$ potrebno n vektora, svaki dimenzije $1 \times L$
- h_j opisuje veze između izlaza bistabila posmačnog registra i j-tog zbrajača
  - simbol 1 na i-toj poziciji unutar vektora veza j-tog zbrajača označava da je izlaz i-tog bistabila posmačnog registra vezan na j-ti zbrajač
  - simbol 0 označava da takva veza ne postoji
- primjer (slajd 7): $\mathbf{h}^{(1)} = [1\ 1\ 1]$ i $\mathbf{h}^{(2)} = [1\ 0\ 1]$
- $\mathbf{h}^{(1)}$ i $\mathbf{h}^{(2)}$ – <u>impulsni odzivi</u> izlaza $s^{(1)}$, odnosno $s^{(2)}$

Prikaz kodera pomoću vektora veza (II)

- (za primjer sa slajda 7) ako je na ulazu kodera slijed **d**, izlazi su: $\mathbf{s}^{(1)} = \mathbf{d} * \mathbf{h}^{(1)}, \quad \mathbf{s}^{(2)} = \mathbf{d} * \mathbf{h}^{(2)}$
  - pri čemu * označava konvoluciju u diskretnom vremenu
- cijeli izlazni slijed nastaje prepletanjem $\mathbf{s}^{(1)}$ i $\mathbf{s}^{(2)}$

$$\mathbf{s} = \left(s_1^{(1)} s_1^{(2)} s_2^{(1)} s_2^{(2)} s_3^{(1)} s_3^{(2)} \ldots \right)$$

- operacija konvolucije je ekvivalentna množenju u domeni transformacije
- (<u>definicija</u>) *D*-transformacija: $f(D) = \sum_{i=0}^{\infty} f_i D^i$
  - ovo je i dalje prikaz u vremenu

Prikaz kodera pomoću vektora veza (III)

- *D* označava jedinično kašnjenje *T* [s] koje uzrokuje bistabil u posmačnom registru
  - množenje s *D* je pomak za *T* [s], množenje s D^2 je pomak za 2*T* [s] itd.
- prijenosne funkcije impulsnih odziva su:
  - (slajd 7) $h^{(1)}(D)=1+D+D^2, \quad h^{(2)}(D)=1+D^2$
- transformacije izlaza $\mathbf{s}^{(1)}$ i $\mathbf{s}^{(2)}$ su:

$$s^{(1)}(D)=d(D)h^{(1)}(D), \quad s^{(2)}(D)=d(D)h^{(2)}(D)$$

- transformacija ukupnog izlaza **s** je:

$$s(D)=s^{(1)}(D^2)+Ds^{(2)}(D^2)$$

impulsni odziv kodera

$$\text{za } d(D)=100 \quad s(D)=1+D^2+D^4+D\left(1+D^4\right)=1+D+D^2+D^4+D^5$$

Prikaz kodera pomoću vektora veza (IV)

$$s(D) = s^{(1)}\left(D^2\right) + Ds^{(2)}\left(D^2\right)$$

- objašnjenje izraza:
  - $s^{(1)}$ se množi s 1, a $s^{(2)}$ s D zato jer se radi o prepletanju tih izlaznih simbola, pa svaki simbol $s_i^{(2)}$ kasni za T vremena (množenje s D) u odnosu na relevantni simbol $s_i^{(1)}$
  - u izrazu se nalaze $s^{(1)}(D^2)$ i $s^{(2)}(D^2)$, a ne $s^{(1)}(D)$ i $s^{(2)}(D)$: s obzirom da se u ukupnom izlaznom slijedu **s** simboli $s_i^{(1)}$ pojavljuju naizmjenično sa simbolima $s_i^{(2)}$, to znači da su $s_i^{(1)}$ i $s_{i+1}^{(1)}$ međusobno razmaknuti za $2T$ vremena, isto vrijedi i za $s_i^{(2)}$
    - zato treba promatrati sljedove $f(D^2)$, tj. $s^{(1)}(D^2)$ i $s^{(2)}(D^2)$

Primjer prikaza pomoću vektora veza

- za primjer odziva kodera sa slajdova 10 i 11
- **d** = [1 0 1], $d(D) = 1 + D^2$

$$h^{(1)}(D)=1+D+D^2, \quad h^{(2)}(D)=1+D^2$$

$$s^{(1)}(D)=d(D)h^{(1)}(D)=\left(1+D^2\right)\left(1+D+D^2\right)=1+D+D^3+D^4$$

$$s^{(2)}(D)=d(D)h^{(2)}(D)=\left(1+D^2\right)\left(1+D^2\right)=1+D^4$$

$$s(D)=s^{(1)}\left(D^2\right)+Ds^{(2)}\left(D^2\right)=$$

$$=1+D^2+D^6+D^8+D\left(1+D^8\right)=$$

$$=1+D+D^2+D^6+D^8+D^9$$

- što odgovara slijedu 11 10 00 10 11

Generirajuća matrica koda

- za generirajuću matricu koda **G**, ulazni slijed **d** i izlazni slijed **s** vrijedi: **s = d·G**
- konvolucijski koder je linearan
  - svaki bit s ulaza možemo promatrati kao samostalan ulazni simbol
  - odziv na više uzastopnih bitova je zbroj pomaknutih impulsnih odziva kodera
- za primjer kodera sa slajda 7:
  - neka su $\mathbf{h}^{(1)} = \left[h_1^{(1)} h_2^{(1)} h_3^{(1)} \right]$, $\mathbf{h}^{(2)} = \left[h_1^{(2)} h_2^{(2)} h_3^{(2)} \right]$
    - $h_j^{(m)}$ je j-ti bit u impulsnom odzivu izlaza $s^{(m)}$, $1 \le j \le 3$, $1 \le m \le 2$
  - na ulaz kodera slijed bita … d_{i-2} d_{i-1} d_i d_{i+1} d_{i+2} …

Generirajuća matrica koda (II)

- kad bit d_i uđe u krajnji lijevi bistabil kodera (B_0) u preostala dva bistabila su $d_{i\text{-}1}$ (u B_1) i $d_{i\text{-}2}$ (u B_2)
- bit d_i proizvede sljedeće vrijednosti izlaza kodera:

$$s_i^{(1)} = d_i \cdot h_1^{(1)}, \quad s_{i+1}^{(1)} = d_i \cdot h_2^{(1)}, \quad s_{i+2}^{(1)} = d_i \cdot h_3^{(1)}$$

$$s_i^{(2)} = d_i \cdot h_1^{(2)}, \quad s_{i+1}^{(2)} = d_i \cdot h_2^{(2)}, \quad s_{i+1}^{(2)} = d_i \cdot h_3^{(2)}$$

  - kako se bit d_i pomiče kroz koder svaki put pomnoži sljedeći koeficijent impulsnog odziva $\mathbf{h}^{(m)}$
    - pojašnjenje: bit d_i = 1 svojim prolaskom kroz koder proizvede na izlazu $s^{(m)}$ slijed od tri bita $h_1^{(m)}$ $h_2^{(m)}$ $h_3^{(m)}$ (= 1·$\mathbf{h}^{(m)}$), a bit d_i = 0 proizvede slijed 0 0 0 (= 0·$\mathbf{h}^{(m)}$)
  - ukupni slijed **s** kojeg na izlazu proizvede prolazak bita d_i kroz koder je:

$$\mathbf{s} = \left[s_i^{(1)}\ s_i^{(2)}\ s_{i+1}^{(1)}\ s_{i+1}^{(2)}\ s_{i+2}^{(1)}\ s_{i+2}^{(2)} \right] = \left[d_i h_1^{(1)}\ d_i h_1^{(2)}\ d_i h_2^{(1)}\ d_i h_2^{(2)}\ d_i h_3^{(1)}\ d_i h_3^{(2)} \right]$$

  - dakle, za poruku duljine 1 bit $\mathbf{G} = \left[h_1^{(1)}\ h_1^{(2)}\ h_2^{(1)}\ h_2^{(2)}\ h_3^{(1)}\ h_3^{(2)} \right]$

Generirajuća matrica koda (III)

- na sličan način bit d_{i+1} proizvede odziv

$$s_{i+1}^{(1)} = d_{i+1} \cdot h_1^{(1)}, \quad s_{i+2}^{(1)} = d_{i+1} \cdot h_2^{(1)}, \quad s_{i+3}^{(1)} = d_{i+1} \cdot h_3^{(1)}$$

$$s_{i+1}^{(2)} = d_{i+1} \cdot h_1^{(2)}, \quad s_{i+2}^{(2)} = d_{i+1} \cdot h_2^{(2)}, \quad s_{i+3}^{(2)} = d_{i+1} \cdot h_3^{(2)}$$

- sada je ukupni odziv kojeg proizvedu bitovi d_i i d_{i+1}

$$\mathbf{s} = \left[s_i^{(1)}\ s_i^{(2)}\ s_{i+1}^{(1)}\ s_{i+1}^{(2)}\ s_{i+2}^{(1)}\ s_{i+2}^{(2)}\ s_{i+3}^{(1)}\ s_{i+3}^{(2)} \right] = \left[d_i \cdot h_1^{(1)}, d_i \cdot h_1^{(2)}, d_i \cdot h_2^{(1)} + d_{i+1} \cdot h_1^{(1)}, d_i \cdot h_2^{(2)} + d_{i+1} \cdot h_1^{(2)}, \right.$$

$$\left. d_i \cdot h_3^{(1)} + d_{i+1} \cdot h_2^{(1)}, d_i \cdot h_3^{(2)} + d_{i+1} \cdot h_2^{(2)}, d_{i+1} \cdot h_3^{(1)}, d_{i+1} \cdot h_3^{(2)} \right]$$

- ako se prikaz pretvori u **s** = **d·G**, tada proizlazi

$$\mathbf{G} = \begin{bmatrix} h_1^{(1)} h_1^{(2)} & h_2^{(1)} h_2^{(2)} & h_3^{(1)} h_3^{(2)} & 0\ \ 0 \\ 0\ \ 0 & h_1^{(1)} h_1^{(2)} & h_2^{(1)} h_2^{(2)} & h_3^{(1)} h_3^{(2)} \end{bmatrix}$$

- evidentno je donji redak pomak gornjeg udesno
  - razlog tome je kašnjenje bita d_{i+1} za bitom d_i za T [s], tj. za trajanja kašnjenja zbog prolaska bita kroz bistabil

Generirajuća matrica koda (IV)

- općeniti oblik generirajuće matrice konvolucijskog koda $(n, 1, L)$:
  - impulsni odziv izlaza $s^{(m)}$, $1 \le m \le n$
  - je $\mathbf{h}^{(m)} = [h_1^{(m)}\ h_2^{(m)}\ \dots\ h_L^{(m)}]$
  - ulazna poruka $\mathbf{d} = [\mathbf{d}_1\ \mathbf{d}_2\ \dots\ \mathbf{d}_r]$

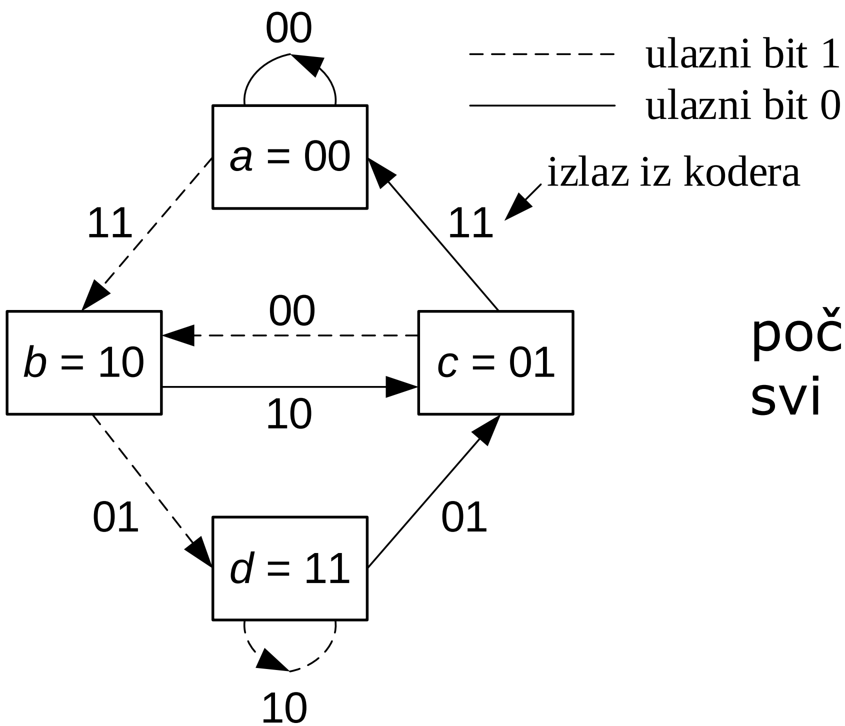
$$\mathbf{G} = \begin{bmatrix} h_1^{(1)}h_1^{(2)}\dots h_1^{(n)} & h_2^{(1)}h_2^{(2)}\dots h_2^{(n)} & h_3^{(1)}h_3^{(2)}\dots h_3^{(n)} & \dots & h_L^{(1)}h_L^{(2)}\dots h_L^{(n)} & & & \\ & h_1^{(1)}h_1^{(2)}\dots h_1^{(n)} & h_2^{(1)}h_2^{(2)}\dots h_2^{(n)} & \dots & h_{L-1}^{(1)}h_{L-1}^{(2)}\dots h_{L-1}^{(n)} & h_L^{(1)}h_L^{(2)}\dots h_L^{(n)} & & \\ & & h_1^{(1)}h_1^{(2)}\dots h_1^{(n)} & \dots & h_{L-2}^{(1)}h_{L-2}^{(2)}\dots h_{L-2}^{(n)} & h_{L-1}^{(1)}h_{L-1}^{(2)}\dots h_{L-1}^{(n)} & h_L^{(1)}h_L^{(2)}\dots h_L^{(n)} & \\ & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

  - Toeplitzova matrica – svaki redak je pomak prethodnog za jednu poziciju udesno
    - elementi matrice koji nisu prikazani jednaki su nulama
    - dimenzija matrice je $r \times (L + r - 1)$

Dijagram stanja konvolucijskog kodera

- konvolucijski koder – automat s konačnim brojem stanja (engl. *finite-state machine*)
- stanje – najmanja količina informacije koja, zajedno s trenutnim ulazom u automat, određuje izlaz iz automata
  - koder ima 2^{L-1} mogućih stanja
- za koder s $R = 1/n$, stanje predstavlja sadržaj $L - 1$ <u>krajnje desno pozicioniranih bistabila</u> u posmačnom registru kodera (<u>ne nužno</u>)
  - ako je u t_i stanje kodera $X_i = m_{i-1}m_{i-2}\dots m_{i-L+1}$
  - tada je kodna riječ U_i potpuno određena s X_i i m_i
  - pri određivanju izlaza stanje X_i predstavlja prošlost kodera
  - koder je Markovljev ako vrijedi $P\left(X_{i+1} \middle| X_i, X_{i-1}, \dots, X_0\right) = P\left(X_{i+1} \middle| X_i\right)$

Dijagram stanja konvolucijskog kodera (II)



početno, tj. nulto stanje – svi bistabili u stanju nula

- primjer: koder sa slajda 7, R = 1/2, L = 3
- četiri stanja: *a*, *b*, *c* i *d*
- iz svakog stanja moguća su samo dva prelaza, ovisno o ulaznom bitu

Dijagram stanja konvolucijskog kodera (III)

- neka je **d** = 11011 iza kojih slijedi 00
  - pretpostavka: na početku svi bistabili u stanju 0

ulazni bit m_i	sadržaj registra	stanje u t_i	stanje u t_{i+1}	izlaz u t_i	
				u_1	u_2
	000	00	00		
1	100	00	10	1	1
1	110	10	11	0	1
0	011	11	01	0	1
1	101	01	10	0	0
1	110	10	11	0	1
0	011	11	01	0	1
0	001	01	00	1	1

- **U** = 11 01 01 00 01 01 11

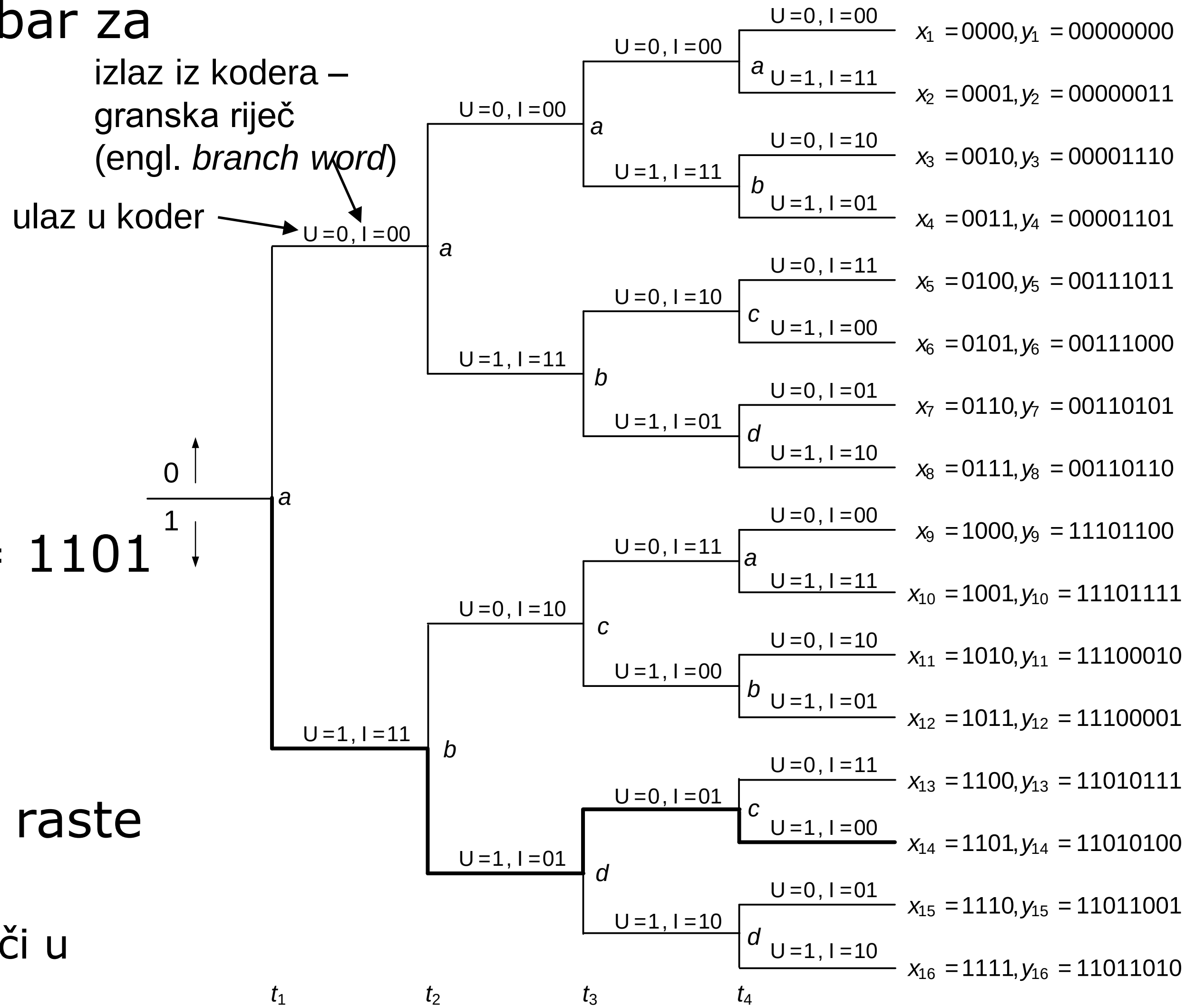
Ovisnost izlaza o početnom stanju registra

ulazni bit m_i	sadržaj registra	stanje u t_i	stanje u t_{i+1}	izlaz u t_i	
				u_1	u_2
	11x	1x	11		
1	111	11	11	1	0
1	111	11	11	1	0
0	011	11	01	0	1
1	101	01	10	0	0
1	110	10	11	0	1
0	011	11	01	0	1
0	001	01	00	1	1

- **U** = 10 10 01 00 01 01 11
  - **U** nije samo funkcija ulaznog bita i L – 1 bita koji mu prethode

Stablasti dijagram konvolucijskog kodera

- dijagram stanja nije dobar za praćenje rada kodera u vremenu
- rješenje: **stablasti dijagram**
  - vertikalna linija – čvor
  - horizontalna - grana
- primjer:
  - za ulazni slijed x_{14} = 1101
  - izlazni slijed y_{14} = 11010100
    - masno otisnuti put
  - problem: broj grana raste s 2^r
    - r je broj granskih riječi u uzastopnom slijedu

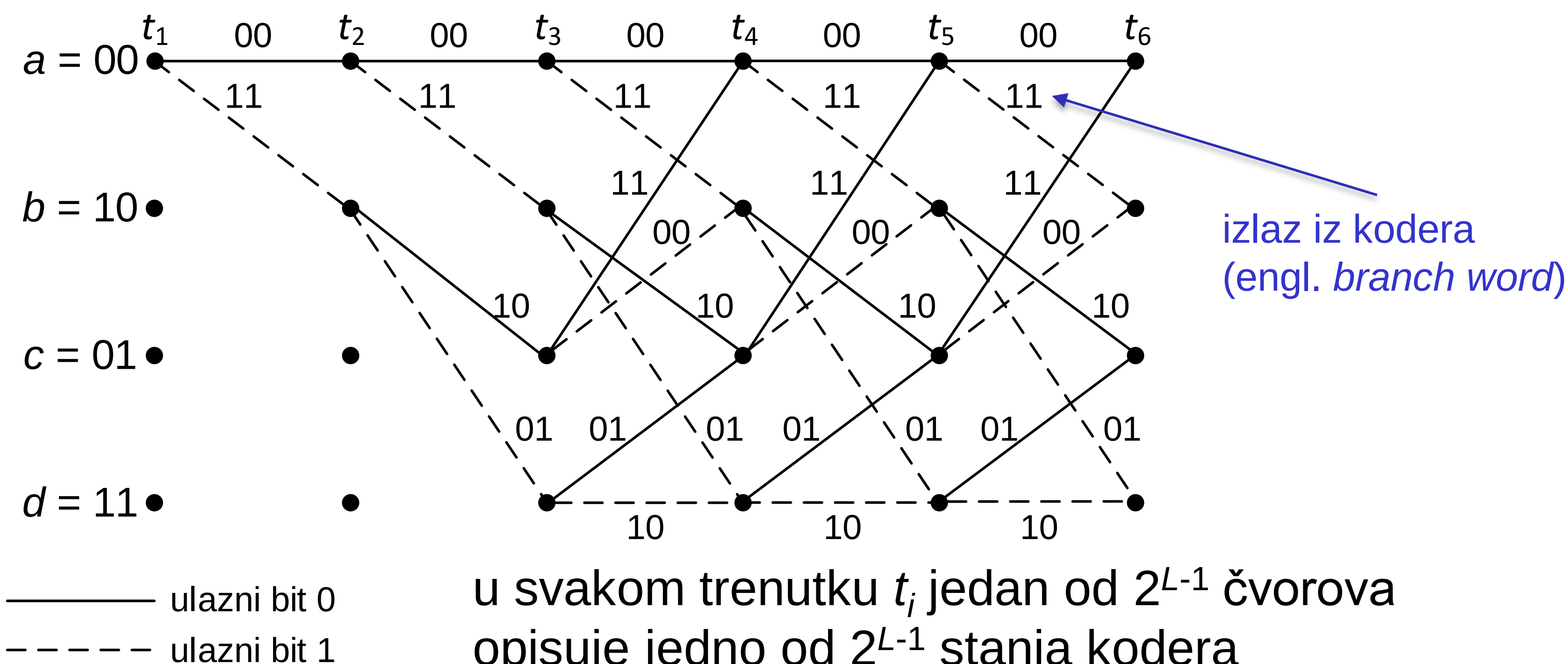


Rešetkasti dijagram konvolucijskog kodera

- stablasta struktura se ponavlja nakon L grananja (u stablastom dijagramu od t_4)
  - izlaz kodera ovisi samo o trenutnom i prethodnih $L - 1$ ulaza
  - npr. ako je ulazni slijed 1 0 0 x y ili 0 0 0 x y
    - promatrano slijeva (prvi bit slijeda) udesno
  - za $L = 3$, kad četvrti bit (bit x) uđe u koder, krajnji lijevi bit je „izbačen” iz kodera i više ne utječe na izlaz
  - dakle, bilo koja dva čvora koji imaju istu oznaku (a, b, c ili d u stablu) u trenutku t_i mogu biti stopljeni u jedan čvor
    - svi putovi koji iz njih proizlaze bit će međusobno identični

Rešetkasti dijagram konvolucijskog kodera (II)

- u svakom trenutku rešetka koristi 2^{L-1} čvorova za prikaz 2^{L-1} stanja kodera
  - u trenutku t_4 dosegnuta je dubina $L = 3$
    - od tog trenutka struktura se ponavlja



u svakom trenutku t_i jedan od 2^{L-1} čvorova opisuje jedno od 2^{L-1} stanja kodera

Dekodiranje konvolucijskih kodova

- u kanalima bez memorije uveden pojam **udaljenost koda**
  - to je udaljenost između primljene riječi i 2^k mogućih poslanih kodnih riječi (k je broj bita poruke)
    - **riječ** je ono što primimo kanalom (nije nužno dio korištenog koda K)
    - **kodna riječ** je dio koda K
    - obje imaju isti broj simbola (bita)
  - u prijemniku se odabire ona kodna riječ koja je najbliža primljenoj riječi
  - takvo pravilo zahtijeva računanje 2^k udaljenosti odnosno **metrika**
  - optimalno je ono odlučivanje koje daje rezultat s minimalnom vjerojatnosti pogreške za binarni simetrični kanala ($p_g < 0,5$) i za kanal s aditivnim bijelim Gaussovim šumom (AWGN)

Dekodiranje prema načelu najveće vjerojatnosti (engl. *Maximum Likelihood*, ML)

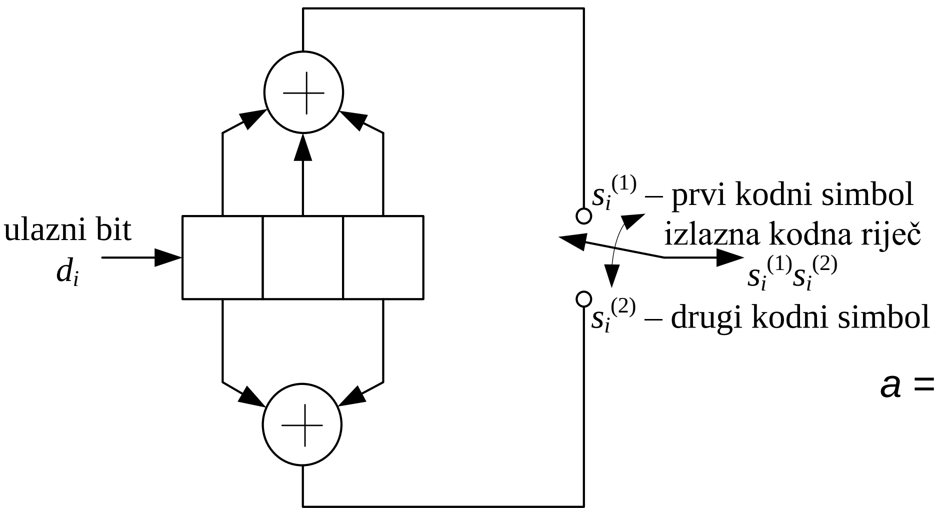
- problem dekodiranja ML-om je u tome što je u rešetkastom dijagramu nužno odrediti sve putove kako bi se dekodiranje provelo
  - broj putova raste eksponencijalno s brojem primljenih simbola
- zato se za dekodiranje konvolucijskih kodova najčešće koristi Viterbijev algoritam
  - algoritam je optimalan jer radi prema načelu ML
  - koristi strukturu kodne rešetke i tako smanjuje složenost proračuna
  - složenost dekodera nije funkcija broja simbola u kodiranom slijedu

Viterbijev algoritam (2)

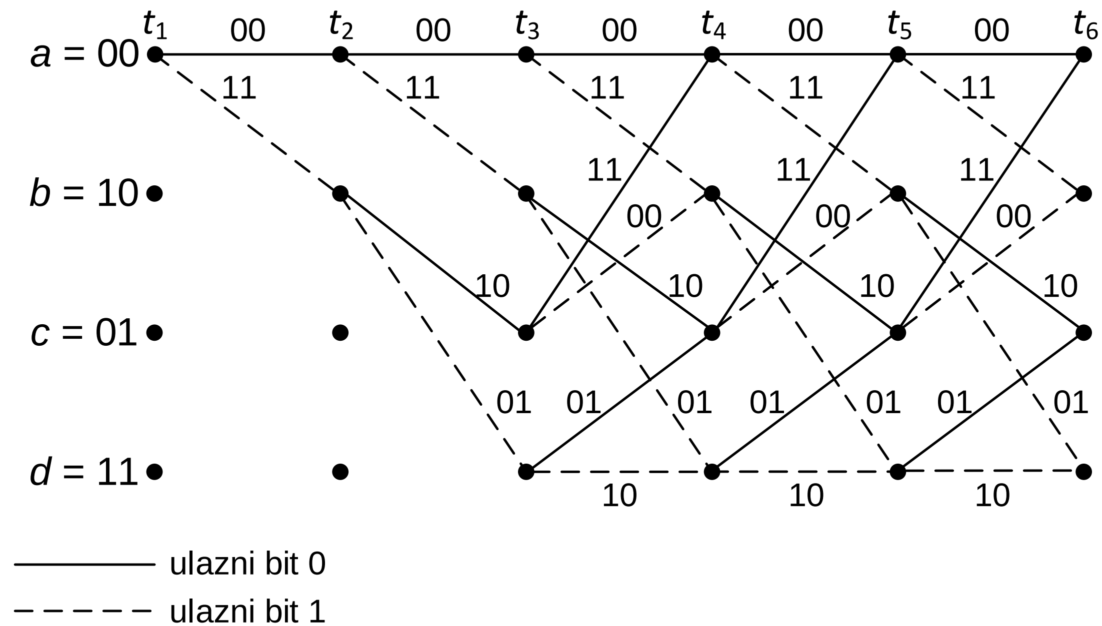
- Viterbijev algoritam poboljšava i ubrzava proračun tako što uspoređuje metrike putova koji se spajaju u nekom stanju i odbacuje putove s lošijom metrikom
  - navedeni postupak se ponavlja za sva stanja
  - na ovaj način na svakoj razini rešetke imamo 2^m "preživjelih" putova (engl. *surviving paths*)
  - $L = m + 1$, m je broj stanja
  - broj proračuna eksponencijalno raste s k i m
  - Viterbijev je algoritam ograničen na kodove s malim k i m

Prikaz načina rada Viterbijevog algoritma

- konvolucijski koder, $R = 1/2$, $L = 3$



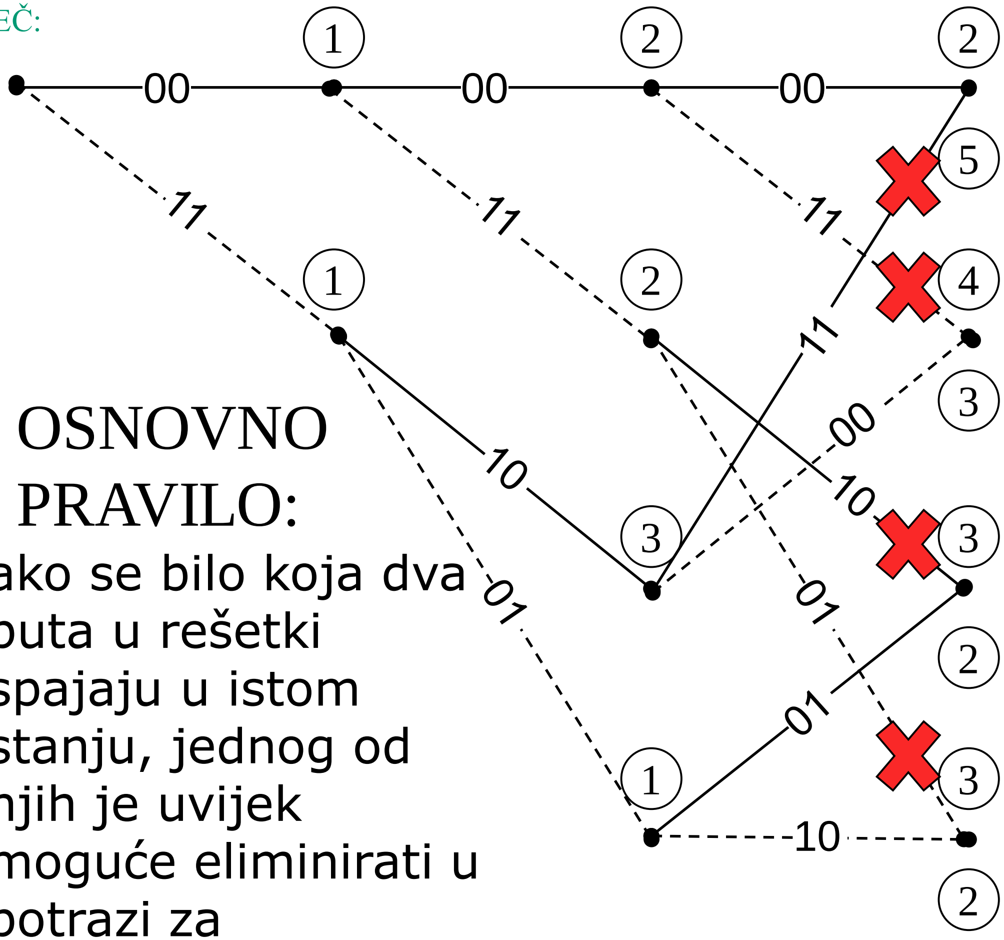
- optimalno dekodiranje konvolucijskih kodova svodi se na pronalaženje puta u rešetkastom dijagramu koji od primljene kodne riječi **c**’ ima minimalnu Hammingovu udaljenost



Viterbijev algoritam: primjer (1/5)

KORAK: 1 2 3 4 5 6

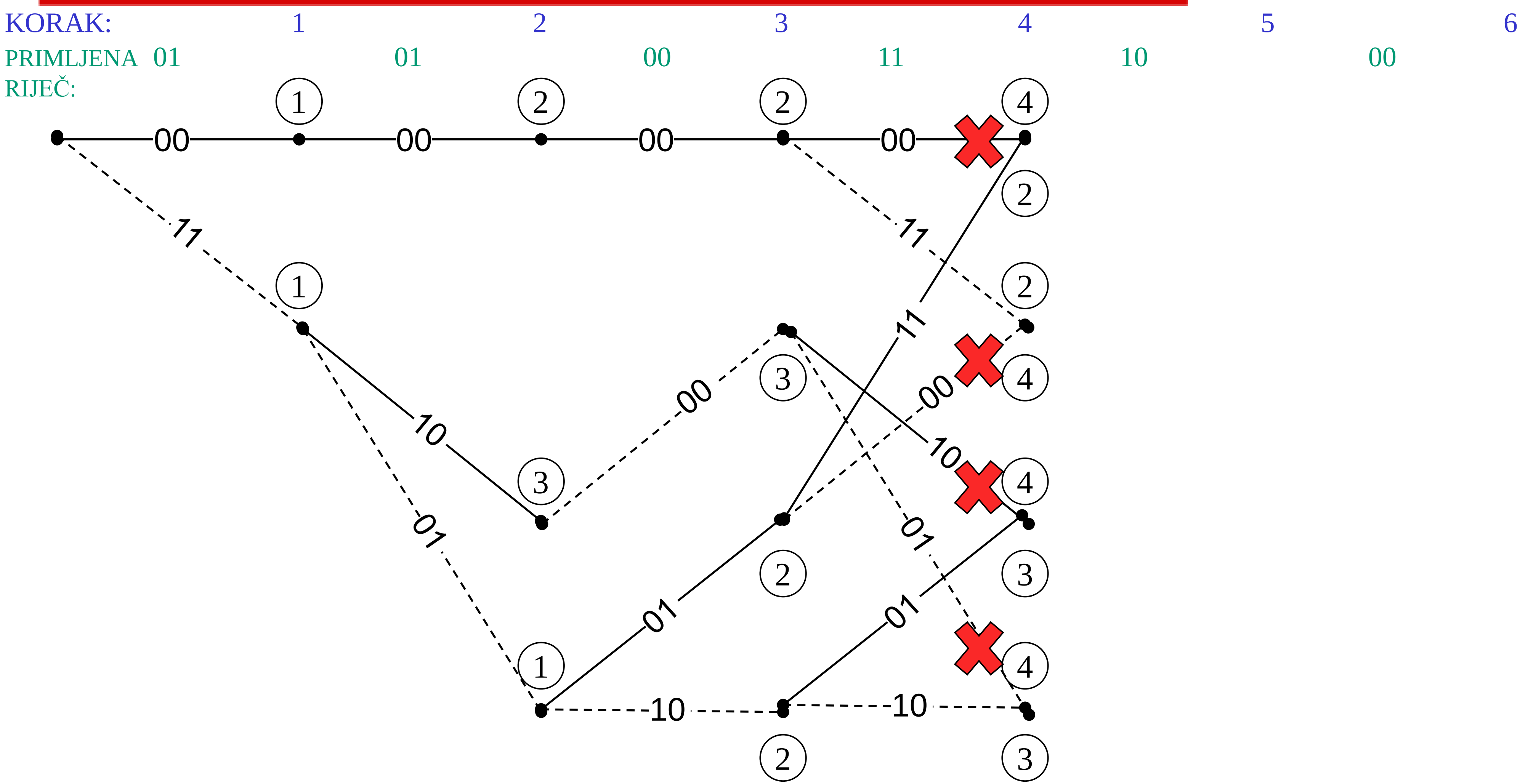
PRIMLJENA RIJEČ: 01 01 00 11 10 00



OSNOVNO PRAVILO:

ako se bilo koja dva puta u rešetki spajaju u istom stanju, jednog od njih je uvijek moguće eliminirati u potrazi za optimalnim putom: elimiramo put s većom kumulativnom Hammingovom metrikom

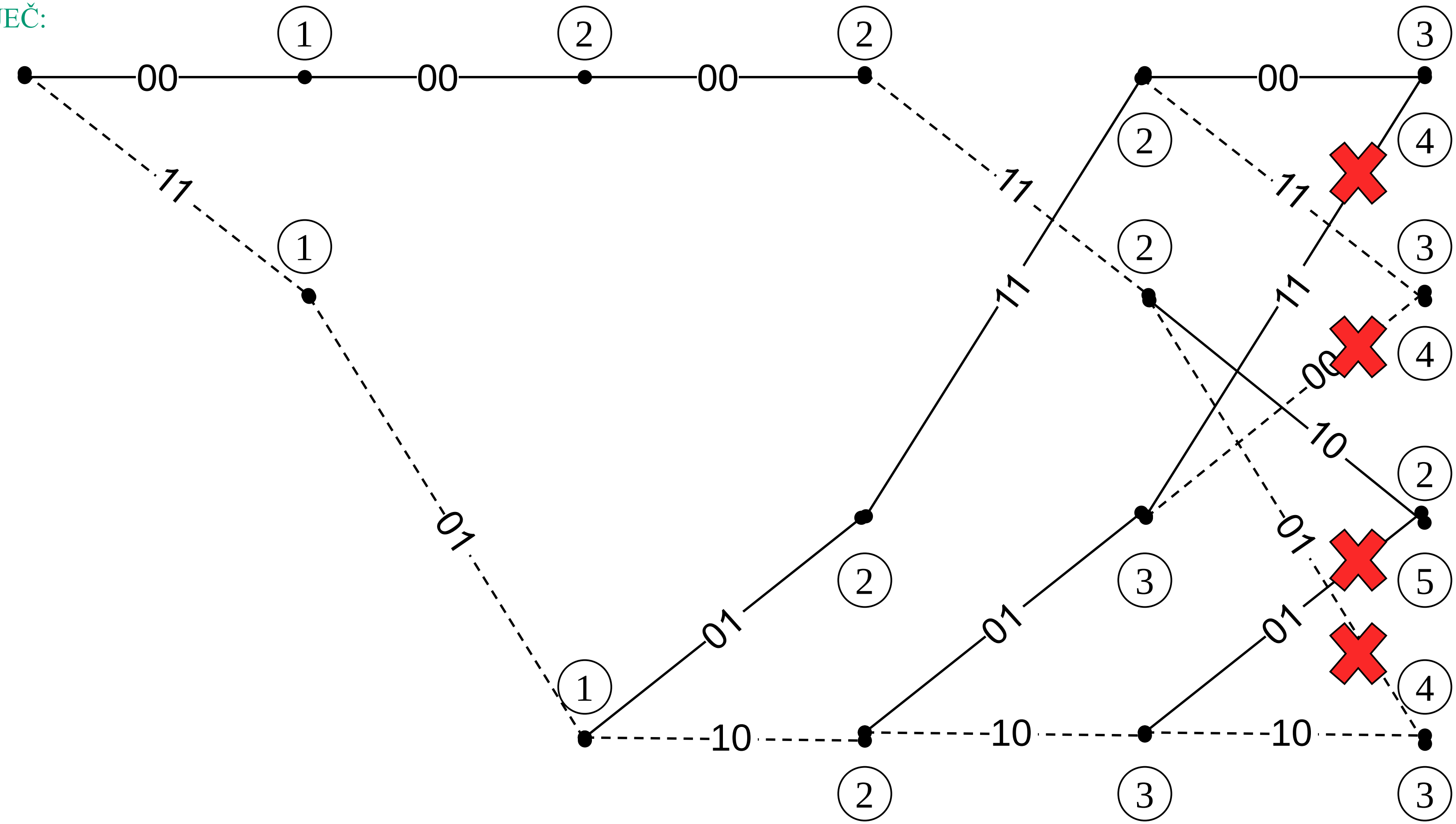
Viterbijev algoritam: primjer (2/5)



Viterbijev algoritam: primjer (3/5)

KORAK: 1 2 3 4 5 6

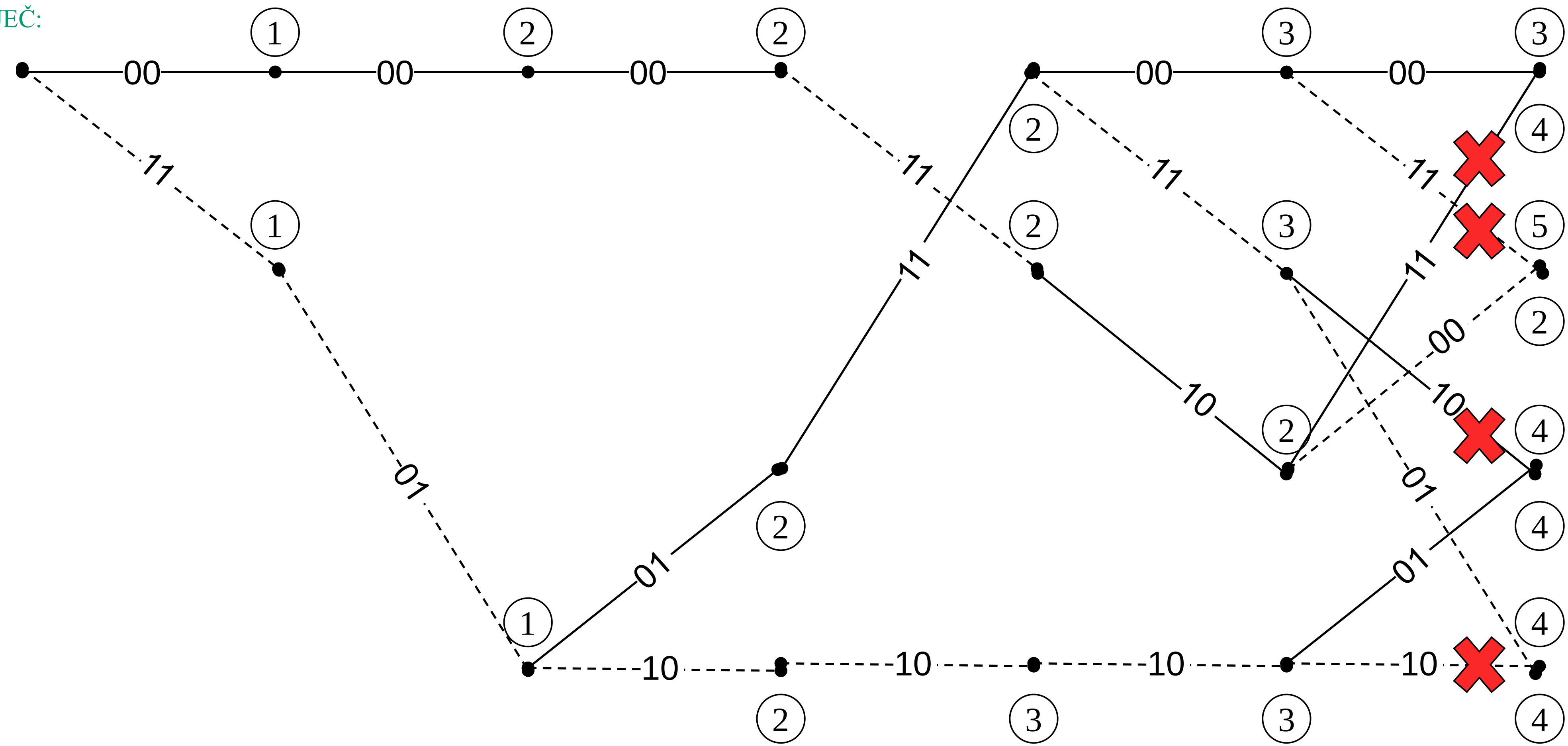
PRIMLJENA RIJEČ: 01 01 00 11 10 00



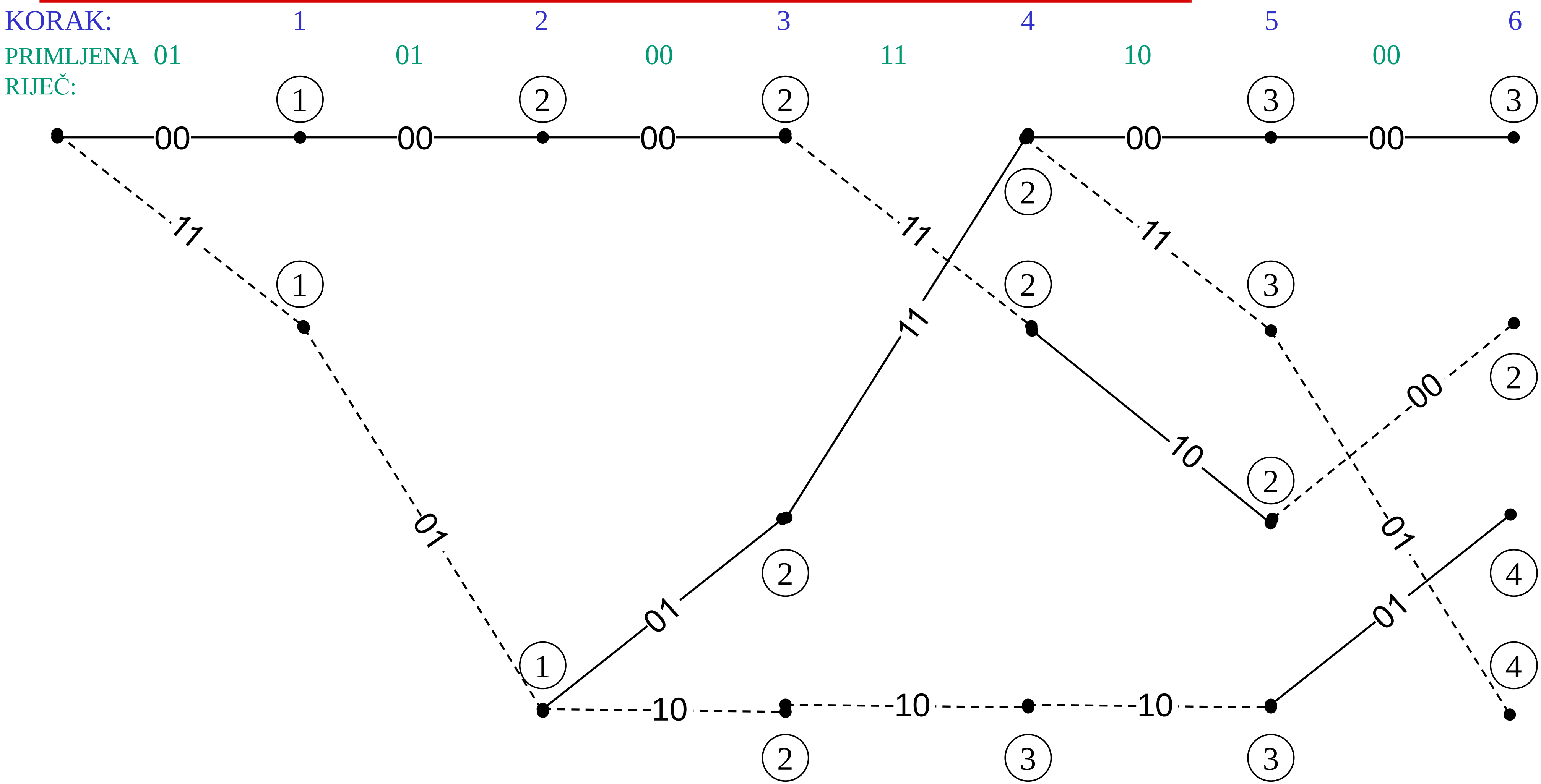
Viterbijev algoritam: primjer (4/5)

KORAK: 1 2 3 4 5 6

PRIMLJENA RIJEČ: 01 01 00 11 10 00



Viterbijev algoritam: primjer (5/5)



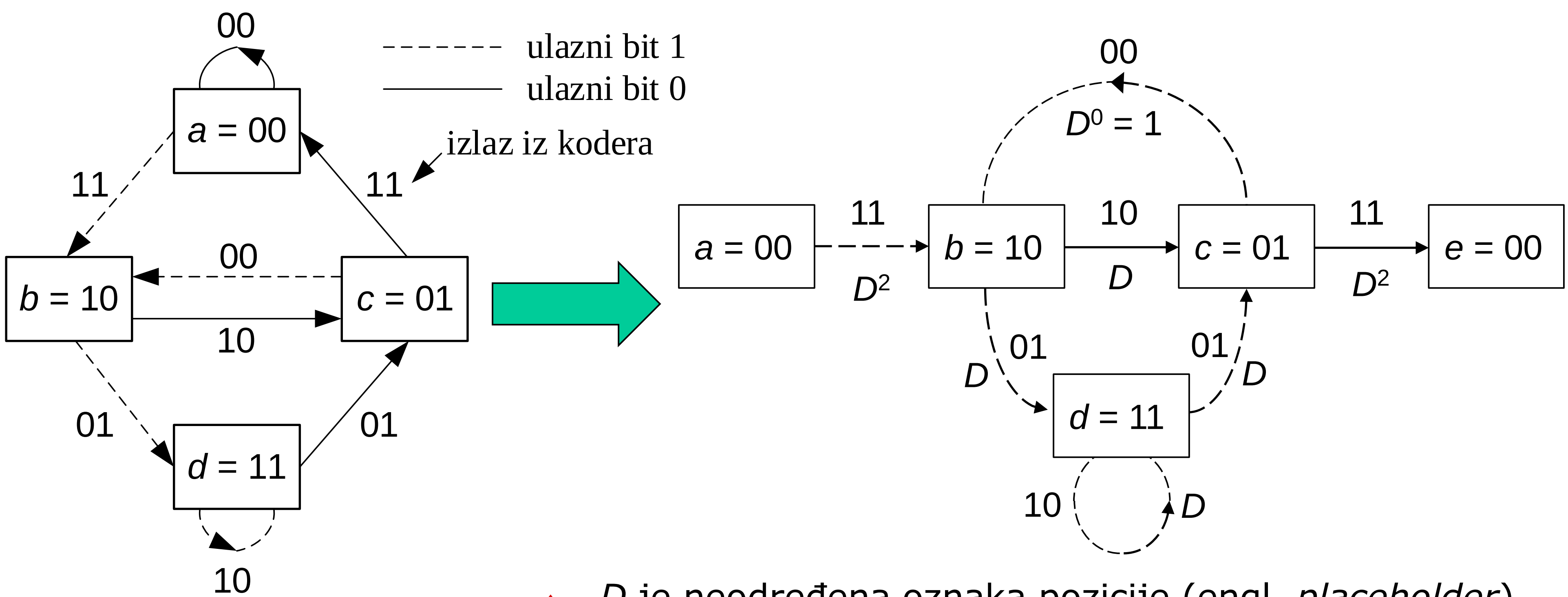
DEKODIRANI NIZ SIMBOLA: 0 0 0 1 0 1

Viterbijev algoritam: neka praktična pitanja

- iz primjera je vidljivo da Viterbijev dekoder može u potpunosti početi s radom (kad su sva stanja uključena) nakon trećeg koraka grananja
- pitanje: Koliko dugo (do kojeg koraka grananja) algoritam treba ponavljati, tj. kada treba donijeti odluku o primljenom slijedu bitova?
  - na ovaj se način određuje dio bitova koji pripadaju izvornoj poruci
  - odgovor na dano pitanje je jako bitan jer cijena dekodera ovisi o veličini memorije u koju se spremaju "preživjeli" putovi
  - pokazuje se da veličina memorije koja je 4 do 5 puta dulja od L daje performanse koda bliske optimumu
    - kao izlaz dekodera, tj. bitovima poruke proglašavaju se bitovi koji pripada najvjerojatnijem putu od svih "preživjelih"
    - kad se donese odluka o izlazu dekodera, svi "preživjeli" putovi u memoriji brišu se i u istu spremaju novi
- pitanje: Što dekoder radi ako u istom stanju ima dva puta koji imaju jednaku metriku? U takvim prilikama dekoder odabire slučajno jedan od ta dva puta

Prijenosna funkcija konvolucijskog kodera

- primjer kodera, R = 1/2, L = 3



- D je neodređena oznaka pozicije (engl. *placeholder*)
- eksponent od D označava Hammingovu udaljenost izlazne riječi prema riječi sastavljenoj isključivo od simbola 0

Proračun prijenosne funkcije kodera

- put *a b c e* (počinje i završava u stanju 00) ima prijenosnu funkciju D^5
  - udaljenost od puta sastavljenog samo od „0" je 5
- putovi *a b d c e* i *a b c b c e* imaju prijenosnu funkciju D^6
- skup jednadžbi stanja

$$X_b = D^2 X_a + X_c$$

$$X_c = D X_b + D X_d$$

$$X_d = D X_b + D X_d$$

$$X_e = D^2 X_c$$

$X_a, \ldots, X_e$ su prazne varijable za označavanje djelomičnih putova do čvorova grafa

Prijenosna funkcija i udaljenost koda

- Prijenosna funkcija, *T*(*D*)
  - (engl. *transfer function* ili *generating function*)

$$T(D)=\frac{X_e}{X_a}=\frac{D^5}{1-2D}=D^5+2D^6+4D^7+\ldots+2^l D^{l+5}+\ldots,\ l=0,1,2,\ldots$$

  - *T*(*D*) pokazuje da postoji:

pomoćni izraz: $\frac{1}{1-x}=\sum_{n=0}^{\infty}x^n, |x|<1$

    - jedan put s udaljenošću 5 od puta sastavljenog isključivo od „0”, dva puta s udaljenošću 6, četiri s udaljenošću 7
    - općenito postoji 2^l putova s udaljenošću l + 5 od puta sastavljenog isključivo od „0”, l = 0, 1, 2, ...
  - **slobodna udaljenost** koda, d_f, (engl. *free distance*) jednaka je Hammingovoj težini najmanjeg člana u izrazu za *T*(*D*), u ovom primjeru d_f = 5 (zbog D^5)
    - *T*(*D*) nije iskoristiv za veliki *L*
      - složenost mu raste eksponencijalno raste s *L*

Kodno pojačanje

- engl. *coding gain*
  - *CG* definiran kao smanjenje odnosa energije bita prema spektralnoj gustoći snage šuma, E_b/N_0, potrebnog za postizanje tražene vjerojatnosti pogreške bita, P_e, u sustavu s kodiranjem u odnosu na sustav bez kodiranja, a s istim modulacijskim postupkom i kanalom
  - izražava se jedinicom decibel (dB)
  - CG se mijenja u ovisnosti o potrebnom P_e
  - za dekoder s tvrdim odlučivanjem: $CG \leq 10\log_{10}\left[R(t+1)\right]$
    - R je kodna brzina, t je broj pogrešaka koje kôd može ispraviti
  - za dekoder s mekim odlučivanjem: $CG \leq 10\log_{10}\left(Rd_f\right)$
    - d_f je slobodna udaljenost koda

$$t = \left\lfloor \frac{d_f - 1}{2} \right\rfloor$$

Granične vrijednosti kodnih pojačanja

- gornja granica na kodno pojačanje u odnosu na BPSK sustav bez kodiranja
  - Gaussov kanal i dekodiranje s tvrdim odlučivanjem
  - kodna brzina $R = 1/2$

L	d_f	gornja granica (dB)	L	d_f	gornja granica (dB)
3	5	3,97	3	8	4,26
4	6	4,76	4	10	5,23
5	7	5,43	5	12	6,02
6	8	6,00	6	13	6,37
7	10	6,99	7	15	6,99
8	10	6,99	8	16	7,27
9	12	7,78	9	18	7,78

Optimalni kodovi male granične duljine

- kodna brzina $R = 1/2$

granična duljina L	**slobodna udaljenost d_f**	**kodni vektori**
3	5	111 101
4	6	1111 1011
5	7	10111 11001
6	8	101111 110101
7	10	1001111 1101101
8	10	10011111 11100101
9	12	110101111 100011101

- ovi kodovi ne propagiraju katastrofičnu pogrešku
- imaju maksimalnu slobodnu udaljenost za zadani R i L

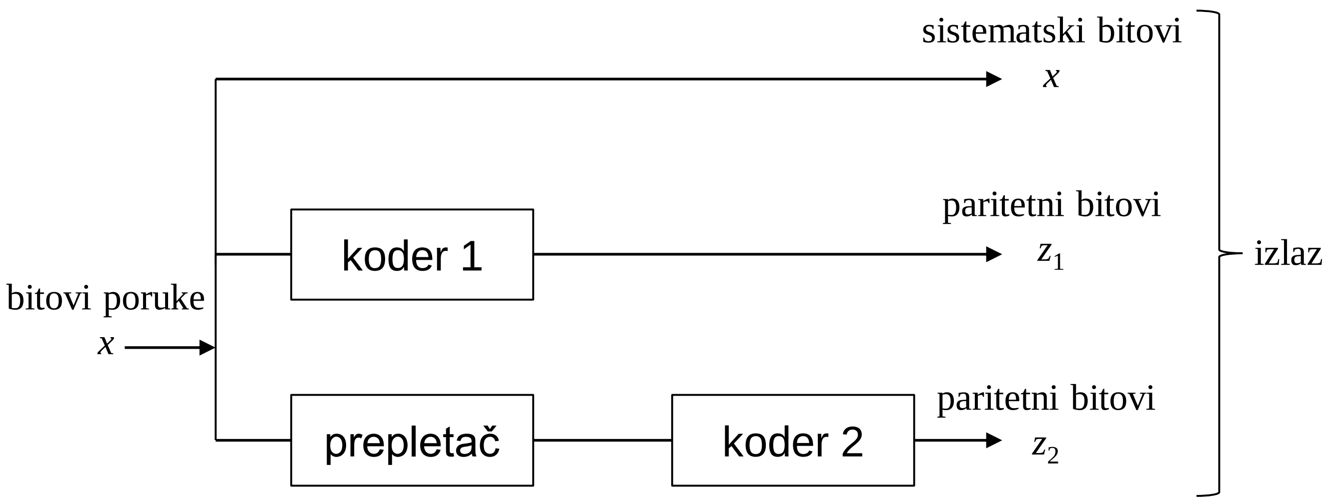
Optimalni kodovi male granične duljine (II)

- kodna brzina $R = 1/3$

granična duljina L	slobodna udaljenost d_f	kodni vektori
3	8	111 111 101
4	10	1111 1011 1101
5	12	11111 11011 10101
6	13	101111 110101 111001
7	15	1001111 1010111 1101101
8	16	11101111 10011011 10101001

Turbo kodovi

- engl *Parallel Concatenated Convolutional Code* (skr. PCCC) *with interleaving*



- prepletač (engl. *interleaver*)
  - mijenja poredak ulaznog slijeda bita/okteta na deterministički način
    - ulaz i izlaz sadrže iste simbole, ali drugačije poredane

RSC koder

- vrste prepletanja
  - blokovsko, pseudoslučajno (u turbo kodovima), ...
  - prepletač popravlja performanse koda
- obično se koriste dva ista konstitutivna koda
  - rekurzivni sistematski konvolucijski (RSC) kodovi
    - obično imaju malu graničnu duljinu *L*
    - RSC kôd se dobiva koristeći nesistematski konvolucijski koder s povratnom vezom

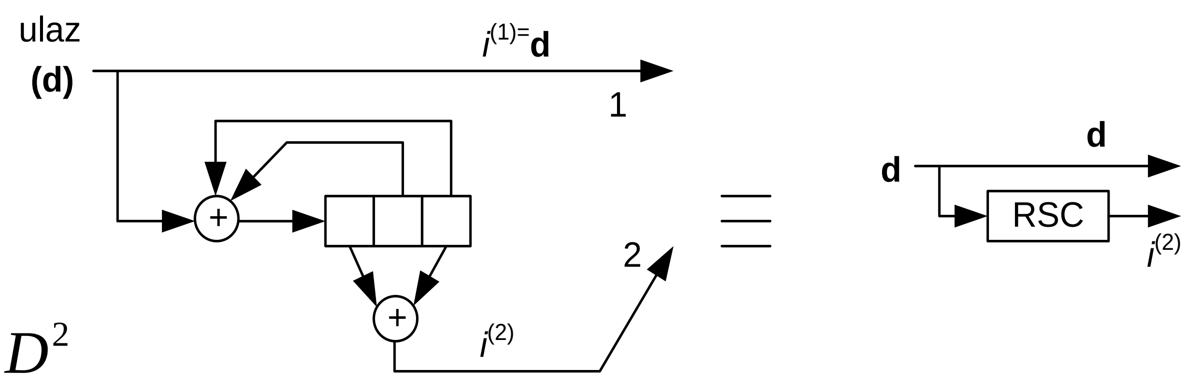
$$g_2(D)=1+D^2$$

feedforward

$$\mathbf{G}(D)=\left[1 \quad \frac{g_2(D)}{g_1(D)}\right]$$

feedback

$$g_1(D)=1+D+D^2$$



Turbo koder

- turbo koder: paralelna veza najčešće dva jednaka RSC kodera odvojena sklopom za prepletanje bitova (engl. *interleaver*)