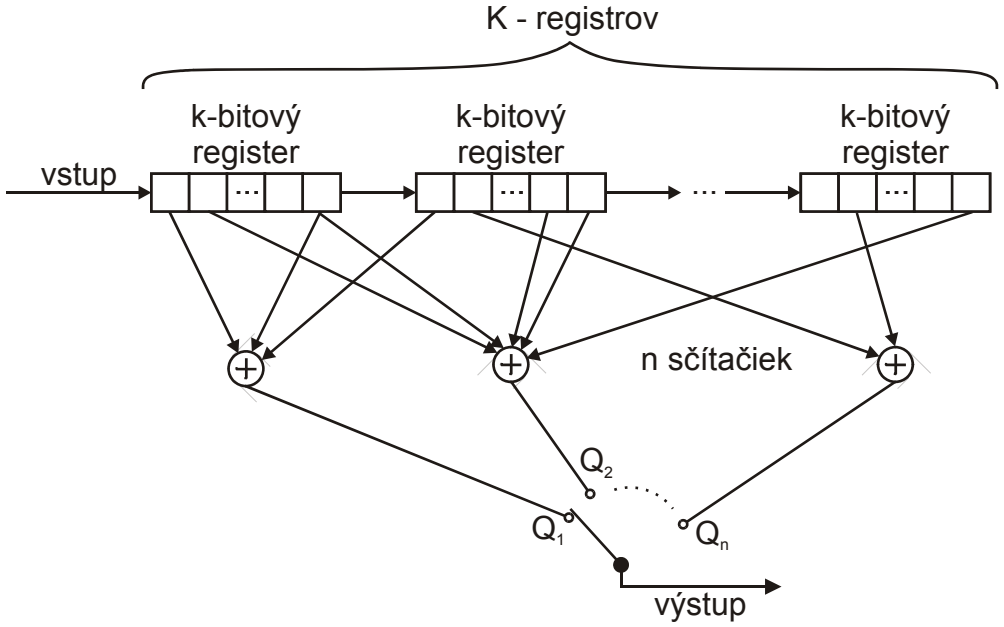
Konvolučné kódy

Sú to zabezpečovacie kódy a patria do skupiny kanálových kódov. Konvolučné kódy využívajú pri kódovaní pamäťové prvky (registre), ktoré sú zapojené kaskádne. Vstupná postupnosť sa posúva v kaskáde **K** posuvných registrov o dĺžke **k** bitov. Zvolené výstupy z registrov sú pripojené na **n** sčítačiek modulo 2. Na každý posun vstupnej postupnosti (o **k** bitov) vstup zosníma z výstupov sčítačiek **n** bitový kód ($\mathbf{Q_1}$ až $\mathbf{Q_n}$).



Obrázok 1 Všeobecná bloková schéma konvolučného kódera

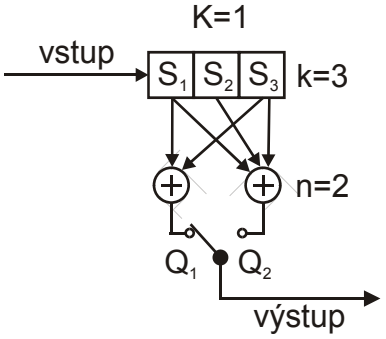
Príklad:

a) Zakódujte 5 bitovú zdrojovú postupnosť v tvare 01101 konvolučným kóderom s parametrami K=1, k=3, n=2, pričom zapojenie kódera je na nasledujúcom obrázku.

b) Zostrojte stromový diagram pre daný kóder.

c) Zostrojte mriežkový diagram pre daný kóder.

d) Dekódujte a opravte prijatú postupnosť w=00 10 10 10 01 01 11



Obrázok 2 Schéma konvolučného kódera

Riešenie a): Vstupná postupnosť $b_1b_2b_3b_4b_5$=01101. Počiatočný obsah registra je $S_1S_2S_3$=000. Výstupné kombinácie sa vypočítajú:

$$Q_1 = S_1 \oplus S_3$$

$$Q_1 = S_1 \oplus S_2 \oplus S_3$$

Pri kódovaní sa vstupná postupnosť musí doplniť o **K.(k-1)** núl, ktoré sa doplnia na koniec postupnosti a teda postupnosť sa rozšíri na $b_1b_2b_3b_4b_5b_6b_7$=0110100.

Symbol	S_1 S_2 S_3	Výpočet	Q_1 Q_2
b_1=0			
b_2=1			
b_3=1			
b_4=0			
b_5=1			
b_6=0			
b_7=0			
Vyslaná postupnosť, teda kód v =			

Stromový diagram

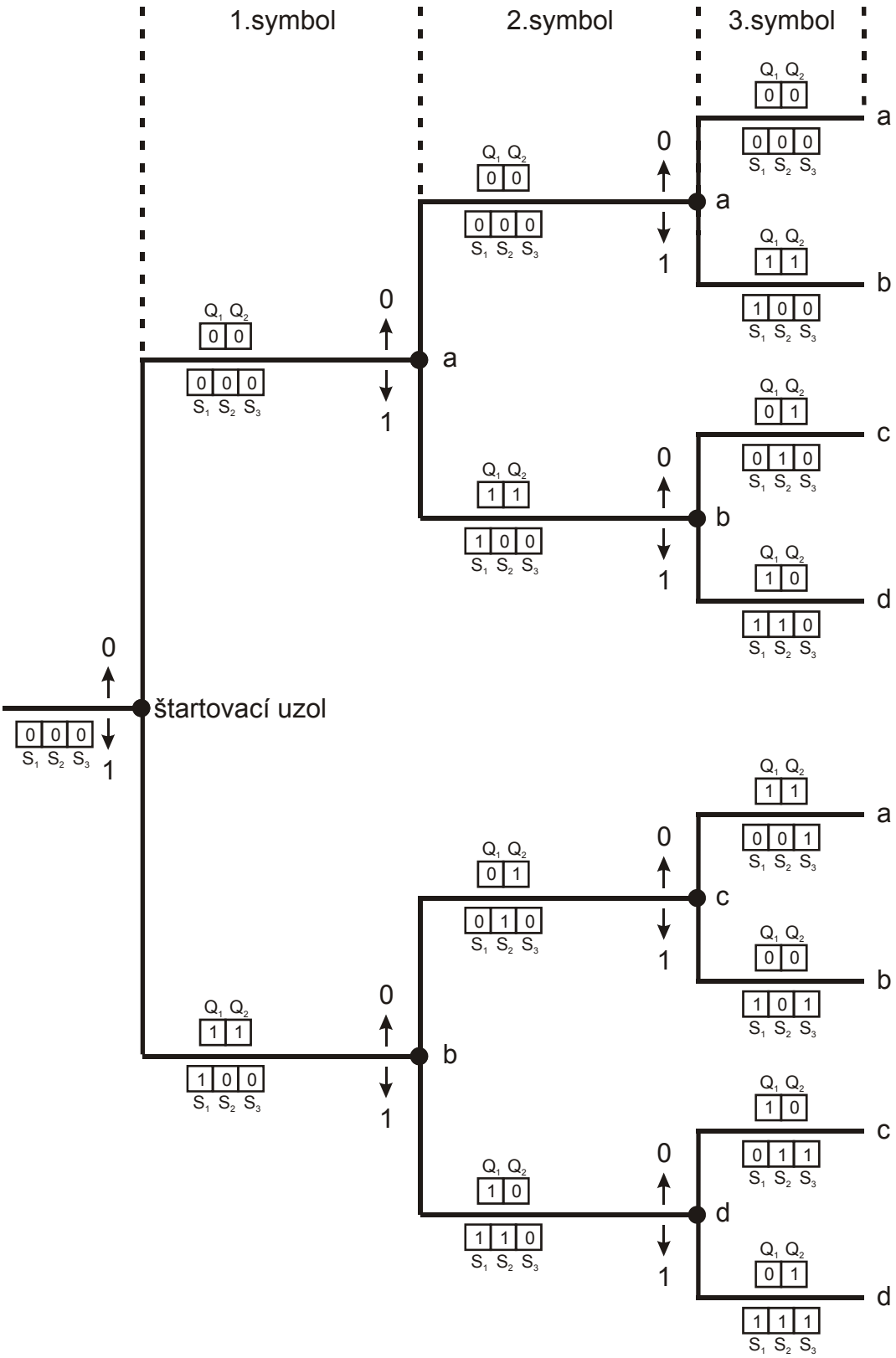
Činnosť kódera možno vyjadriť stromovým diagramom, kde sú graficky znázornené všetky možné činnosti kódera, resp. všetky možné výstupné sekvencie, ktoré zodpovedajú rôznym cestám v strome. Ak sa vo vstupnej postupnosti vyskytne **0,** zodpovedá to ceste hore v príslušnom uzle, a ak **1** zodpovedá to ceste dole v príslušnom uzle. Z každého uzla stromu vychádzajú dve cesty, ktoré zodpovedajú výskytu 0, resp. 1 vo vstupnej postupnosti.

Mriežkový diagram (trelisov diagram)

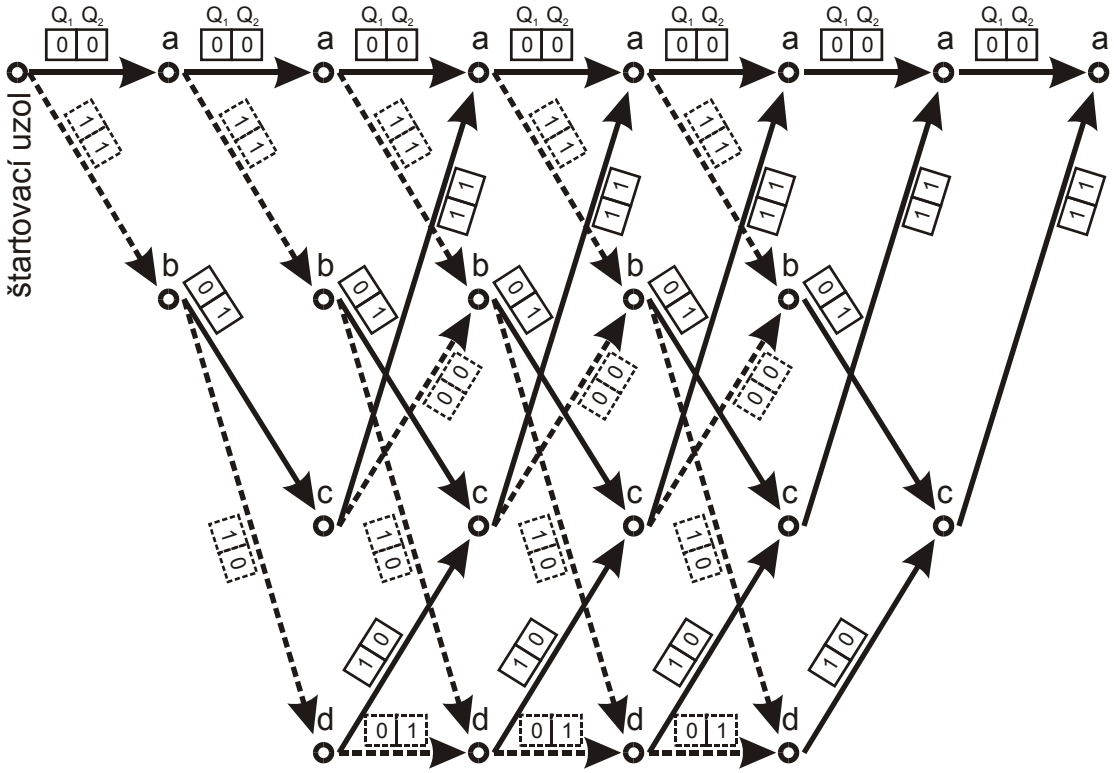
Nesie rovnakú informáciu o činnosti kódera ako stromový diagram. Z každého uzla mriežkového diagramu vychádzajú opäť dve vetvy, ktoré zodpovedajú výskytu **0**, resp. **1** vo vstupnej postupnosti. Plnou čiarou vyznačená vetva reprezentuje výskyt **0** vo vstupnej postupnosti, čiarkovaná vetva zodpovedá výskytu **1** vo vstupnej postupnosti. Po zakódovaní prvých dvoch vstupných symbolov sa kóder môže nachádzať v jednom zo štyroch stavov **a,b,c,d.** Proces kódovania možno graficky znázorniť cestou v mriežkovom diagrame, pričom každá cesta vychádza zo štartovacieho uzla a končí v niektorom stave.

Dekódovanie – Viterbiho algoritmus

Najznámejší algoritmus dekódovania konvolučných kódov je Viterbiho algoritmus. Tento algoritmus vyberá v mriežkovom diagrame cestu s najmenšou Hammingovou vzdialenosťou d_{min} medzi prijatou postupnosťou **w** a postupnosťou symbolov, ktoré sa generujú pri ceste. Dekóder pre každý takt určuje **metriku** t.j. Hammingovú vzdialenosť všetkých možných ciest. Ak do jedného uzla vchádza viac ako jedna cesta, vyberá sa len cesta, ktorá má najmenšiu Hammingovú vzdialenosť. Cesty s horšou metrikou nikdy v ďalšom kroku nenadobudnú lepšiu metriku ako cesty s najlepšou metrikou. Dekódovanie sa končí výberom cesty s najlepšou metrikou spomedzi tých ciest, ktoré prežijú.



Obrázok 3 Stromový diagram



Obrázok 4 Mriežkový diagram

Riešenie d): V prvom kroku načítame prvé tri dvojice znakov Q_1Q_2, ktoré reprezentujú 3 vstupné symboly a vypočítame metriku všetkých ciest po troch krokoch. Stavy a,b,c,d možno dosiahnuť vždy dvoma rôznymi cestami, takže je potrebné preveriť 8 ciest, vzhľadom k prijatej prvej šestici symbolov w=001010.

Z pôvodného mriežkového diagramu zistíme kódové kombinácie pri prechádzaní týchto ciest a realizujeme operáciu $\oplus$ s kombináciou w=001010.

w=001010	
1.cesta do a	2.cesta do a
1.cesta do b	2.cesta do b
1.cesta do c	2.cesta do c
1.cesta do d	2.cesta do d

Z 8 ciest nám zostali 4, ktoré majú lepšie metriky. V ďalšom kroku porovnávame 8 znakov v tvare w=00101010 a preveríme ďalších 8 ciest.

w=00101010	
1.cesta do a	2.cesta do a
1.cesta do b	2.cesta do b
1.cesta do c	2.cesta do c
1.cesta do d	2.cesta do d

Z 8 ciest nám zostali 4, ktoré majú lepšie metriky. V ďalšom kroku porovnávame 10 znakov v tvare w=0010101001 a preveríme ďalších 8 ciest.

w=0010101001	
1.cesta do a	2.cesta do a
1.cesta do b	2.cesta do b
1.cesta do c	2.cesta do c
1.cesta do d	2.cesta do d

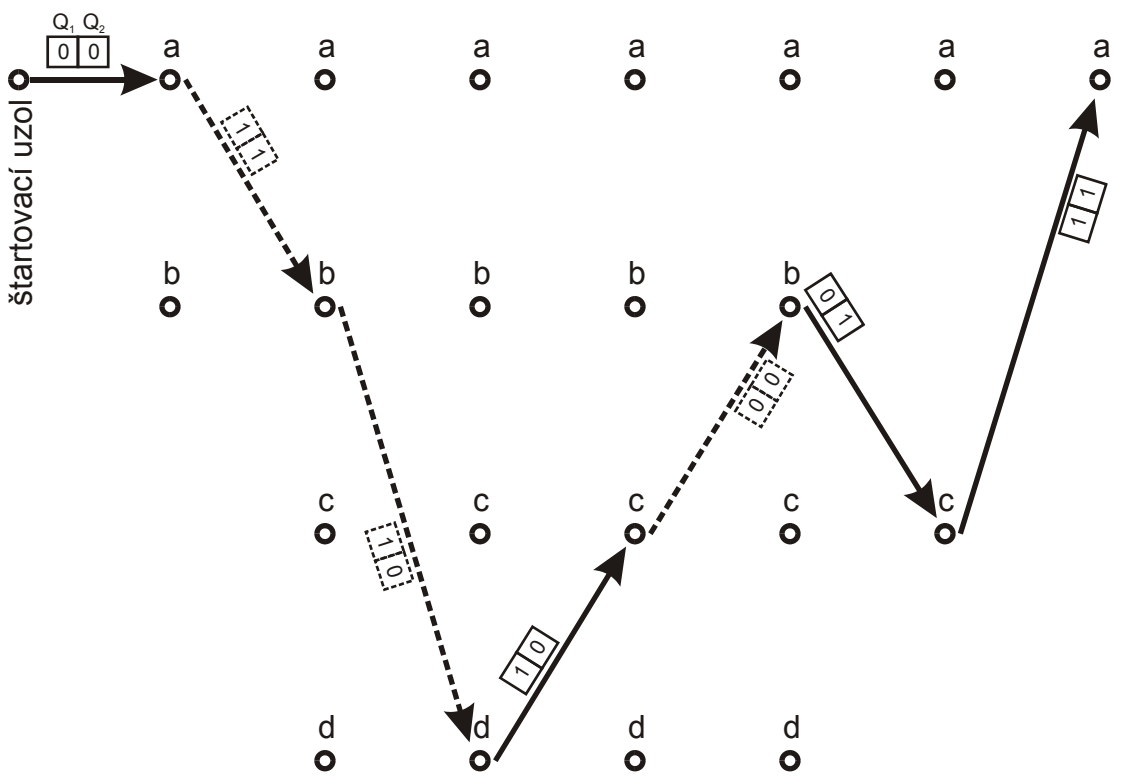
Z 8 ciest nám zostali 4, ktoré majú lepšie metriky. V ďalšom kroku porovnávame 12 znakov v tvare w=001010100101 a preveríme ďalšie 4 cesty, už nie 8, lebo už vedú len 2 cesty do **a** a 2 cesty do **c**.

w=001010100101	
1.cesta do a	2.cesta do a
1.cesta do c	2.cesta do c

Zo 4 ciest nám zostali 2, ktoré majú lepšie metriky. V ďalšom kroku porovnávame 14 znakov v tvare w=00101010010111 a preveríme posledné 2 cesty, lebo už vedú len 2 cesty do **a.**

w=00101010010111	
1.cesta do a	2.cesta do a

Získaná výsledná cesta s najlepšou metrikou ..... je opravená kombinácia. Proces kódovania, chyby a dekódovania je znázornený v nasledujúcej tabuľke. Dekódujeme tak, že prechádzame v trelise výslednou cestou a ak vetva cesty je plná čiara dekódujeme **0**, a ak vetva cesty je prerušovaná čiara dekóduje sa symbol **1**.

	informačné bity					synchronizácia	
Zdrojová (vstupná postupnosť)	0	1	1	0	0	0	0
Vyslaný kód							
Chyba	00	01	00	00	01	00	00
Prijatý kód	00	10	10	10	01	01	11
Cesta s najlepšou metrikou							
Opravený kód							
Dekódovaná postupnosť							