Zaštitno kodiranje signala

Laboratorijska vježba 16

SP

Viterbijevo dekodiranje tvrdom odlukom

Student:

prezime	Ime	mat. broj
	Laboratorij	Datum

Sadržaj:

Sazdano od:

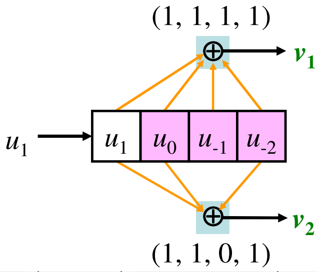
Napomena: Cjelovit opis vježbe nalazi se na: MOODLE

120 Convolutional coding and ModulationH.doc (drugi dio)

Zadaci:

1. Konvolucijski koder prikazan na donjoj slici, nacrtajte u simulacijskome alatu LogiSim 2.7.1. Ispratite promjene u stanjima registra i popunite stanja registara u tablici za ulazni niz bitova , **u** = [**1011000**].
2. Također popunite dobivena izlazna stanja u tablici.

Polinom-generatori za (2, 1, 4) kôd su:



$$g_1 = 1\cdot x^3 + 1\cdot x^2 + 1\cdot x^1 + 1\cdot x^0$$

i

$$g_2 = 1\cdot x^3 + 1\cdot x^2 + 0\cdot x^1 + 1\cdot x^0$$

polinom-generatori	ulaz	stanje registra	izlaz v_1v_2	novo stanje	polinom-generatori	ulaz	stanje registra	izlaz v_1v_2	novo stanje
	0	000	⟶	**000**		**0**	100	⟶	
$x^3\oplus x^2\oplus x^1\oplus x^0$	0	000=0	**00**		$x^3\oplus x^2\oplus x^1\oplus x^0$	0	100=1		
$x^3\oplus x^2\oplus 0\oplus x^0$	0	0 0=0			$x^3\oplus x^2\oplus 0\oplus x^0$	0	1 0=1		
	1	000	⟶	**100**		**1**	100	⟶	
$x^3\oplus x^2\oplus x^1\oplus x^0$	1	000=1	**11**		$x^3\oplus x^2\oplus x^1\oplus x^0$	1	100=0		
$x^3\oplus x^2\oplus 0\oplus x^0$	1	0 0=1			$x^3\oplus x^2\oplus 0\oplus x^0$	1	1 0=0		
	0		⟶			**0**	101	⟶	
$x^3\oplus x^2\oplus x^1\oplus x^0$	0				$x^3\oplus x^2\oplus x^1\oplus x^0$	0	101=0		
$x^3\oplus x^2\oplus 0\oplus x^0$	0				$x^3\oplus x^2\oplus 0\oplus x^0$	0	1 1=0		
	1		⟶			**1**	101	⟶	
$x^3\oplus x^2\oplus x^1\oplus x^0$	1				$x^3\oplus x^2\oplus x^1\oplus x^0$	1	101=1		
$x^3\oplus x^2\oplus 0\oplus x^0$	1				$x^3\oplus x^2\oplus 0\oplus x^0$	1	1 1=1		
	0		⟶			**0**	110	⟶	
$x^3\oplus x^2\oplus x^1\oplus x^0$	0				$x^3\oplus x^2\oplus x^1\oplus x^0$	0	110=0		
$x^3\oplus x^2\oplus 0\oplus x^0$	0				$x^3\oplus x^2\oplus 0\oplus x^0$	0	1 0=1		
	1		⟶			**1**	110	⟶	
$x^3\oplus x^2\oplus x^1\oplus x^0$	1				$x^3\oplus x^2\oplus x^1\oplus x^0$	1	110=1		
$x^3\oplus x^2\oplus 0\oplus x^0$	1				$x^3\oplus x^2\oplus 0\oplus x^0$	1	1 0=0		
	0		⟶			**0**	111	⟶	
$x^3\oplus x^2\oplus x^1\oplus x^0$	0				$x^3\oplus x^2\oplus x^1\oplus x^0$	0	111=1		
$x^3\oplus x^2\oplus 0\oplus x^0$	0				$x^3\oplus x^2\oplus 0\oplus x^0$	0	1 1=0		
	1		⟶			**1**	111	⟶	
$x^3\oplus x^2\oplus x^1\oplus x^0$	1				$x^3\oplus x^2\oplus x^1\oplus x^0$	1	111=0		
$x^3\oplus x^2\oplus 0\oplus x^0$	1				$x^3\oplus x^2\oplus 0\oplus x^0$	1	1 1=1		

Zadaci:

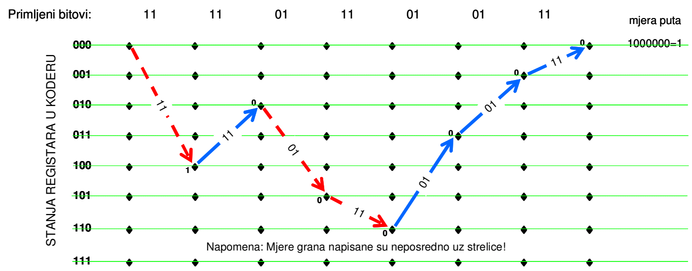
3. Popunite sljedeću tablicu.

Primjer: Koder počinje iz stanja 000, **u** = [**1011000**], **v** = ? → Viterbijevo dekodiranje tvrdom odlukom!

stanja ulaz	bit	stanje prije	prijelaz izlaz	stanje nakon	bit	stanje prije	prijelaz izlaz	stanje nakon	bit	stanje prije	prijelaz izlaz	stanje nakon	bit	stanje prije	prijelaz izlaz	stanje nakon
u	**1**	000	→	**100**	**0**	100	→	**010**	**1**	010	→	**101**	**1**	101	→	**110**
g_1	1	000=1	**11**		0	100=1	**11**		1	010=0	**01**		1	101=1	**11**	
g_2	1	0 0=1			0	1 0=1			1	0 0=1			1	1 1=1		

stanja ulaz	bit	stanje prije	prijelaz izlaz	stanje nakon	bit	stanje prije	prijelaz izlaz	stanje nakon	bit	stanje prije	prijelaz izlaz	stanje nakon
u	**0**	110	→	**011**	**0**	011	→	**001**	**0**	001	→	**000**
g_1	0	110=0	**01**		0	011=0	**01**		0	001=1	**11**	
g_2	0	1 0=1			0	0 1=1			0	0 1=1		

(De)kodiran izlaz je: **v** = [**11 11 01 11 01 01 11**], a dijagram rešetke (*trellis diagram*) je:



Slika 1: Potpuno popunjena rešetka Viterbijevim dekodiranjem

16. 16 VITERBIJEV ALGORITAM DEKODIRANJA TVRDOM ODNOSNO MEKOM ODLUKOM

16.1. 6.1. Viterbijev algoritam

- Želimo dobiti niz bitova najvjerojatnije poruke
- Dobili smo (vjerojatno oštećen) niz bitova
- Viterbijev algoritam za zadani K i r:
  - Radimo postupno za izračunati niz najvjerojatnije poruke.
  - Koristimo dvije metrike:
- Metriku grane BM (*branch metric*) BM(*xmit*, *rcvd*) koja je proporcionalna negativnome logaritmu funkcije vjerojatnosti (*log likelihood*)[1], tj. negativnome logaritmu vjerojatnosti da smo primili *rcvd*, ako je poslan *xmit*.
  - Dekodiranje "tvrdom odlukom": koristimo digitalizirane bitove, računamo Hammingovu udaljenost između *xmit* i *rcvd*. Manja udaljenost je vjerojatnija ako je BER < 1/2.
  - Dekodiranje "mekom odlukom": izravno se koristiti funkcija primljenih napona.
- Metrika puta PM (*path metric*) PM[s, i] za svako stanje s od mogućih 2^{K-1} odaslanih stanja u vremena prijenosa bita i, gdje je $0 \le i < L$ = duljina(poruke).

[1] Funkcija se definira kao prirodan logaritam funkcije vjerojatnosti, $l(\theta; x) = \log L(\theta; x)$

- PM[s, i] = najmanji zbroj BM($xmit$, $rcvd$) nad svim nizovima poruka m što smještaju odašiljač u stanje s u vremenu i.
- PM[s, i+1] računa se iz PM[s, i] i $p_0[i]$, …, $p_{r-1}[i]$

16.1.1. 16.1.1. DEKODIRANJE TVRDOM ODLUKOM

- Nakon prijema svakoga bita, on se odmah digitalizira kao 0 ili 1 usporedbom naponskoga praga.
- Gubimo informacije o tome koliko je bit "dobar", jer se "1" tretira jednako za naponsku razinu od 0,9999 [V] kao i za naponsku razinu od 0,5001 [V]
- Metrika grane što se koristi u Viterbijevome dekoderu za dekodiranje tvrdom odlukom je Hammingova udaljenost između digitaliziranih primljenih naponskih razina i očekivanih paritetnih bitova.
- Odbacivanje informacija nije (skoro) nikada dobra ideja prilikom donošenja odluka.

16.1.2. 16.1.2. VITERBIJEV ALGORITAM ŠTO KORISTI DEKODIRANJE TVRDOM ODLUKOM

- Mjere grana, mjere pojedinačan doprinos negativnoj logaritamskoj vjerojatnost uspoređujući primljene paritetne bitove u odnosu na moguće prenesene paritetne bitove izračunate iz mogućih poruka.
- Mjera puta PM[s, i] proporcionalna je negativnoj logaritmiranoj vjerojatnosti da odašiljač bude u stanju s u vremenu i, uz pretpostavku da je najvjerojatnija poruka dužine i što napušta odašiljač u stanju s.
- Najvjerojatnije poruku je ona poruka što proizvodi najmanji PM[s, N].
- U bilo kojemu trenutku postoji 2^{K-1} najizglednijih poruka što ih pratimo. Vremenska složenost algoritma raste eksponencijski ograničenjem duljine K, a linearno dužinom poruke (za razliku od eksponencijskoga porasta dužine poruke za jednostavna pobrojenja).

16.1.3. 16.1.3. MJERA GRANE ZA DEKODIRANJE TVRDOM ODLUKOM

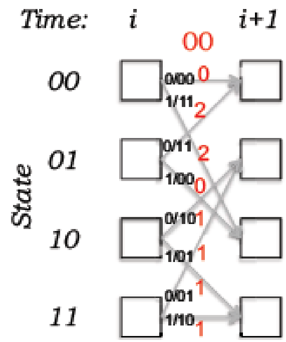
- Mjera grane BM (*branch metric*) = Hammingova udaljenost između očekivanih paritetnih bitova (na predajnoj strani kanala - koder) i dobivenih paritetnih bitova (na prijemnoj strani kanala - dekoder).
- Računa se BM za svaki prijelazni luk (usmjerenu crtu) u dijagramu rešetke.

Primjer: primljena je kombinacija = 00

Mjera grane		Stanja se razlikuju u:
BM(00, 00)	= 0	0 mjesta
BM(01, 00)	= 1	1 mjesto
BM(10, 00)	= 1	1 mjesto
BM(11, 00)	= 2	2 mjesta

16.1.4. 16.1.4. RAČUNANJE PM[*]

- Koristit ćemo izračun mjere puta PM[s, i+1] iz PM[s, i].



Slika 2.a: Mjere grana (crvene brojke uz svaku granu) za odabir grana što će „preživjeti“

Koder i dekoder uvijek započinju rad iz stanja "sve 0".

Iz slike se vide:

- svi izlazi iz kodera za kôd $R = 1/2$ (za 1 ulazni bit dobiju se 2 izlazna bita) i
- ispravni prijelazi kodera i dekodera za sva stanja registara.

Ako se na strani dekodera primila 00:

Zadaci:

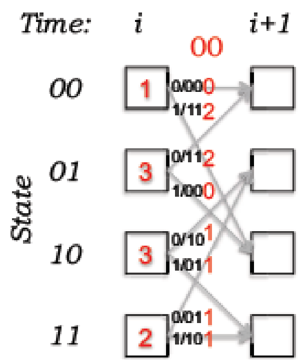
4. Popunite mjere grana u tablici.

Stanje: kodera/dekodera	ulaz u koder	izlaz kodera	BM za poslanu 0	BM za poslanu 1
00	0	00		
	1	11		
01	0	11		
	1	00		
10	0	10		
	1	01		
11	0	01		
	1	10		

Znači, poslala se 00 pa nema pogreške (ili je pogreška dvostruka), a mjera je 0.

Za prethodno prikazanu rešetku, ako bi se primila npr. 10 (umjesto 00), znamo da je nastupila pogreška ali ne znamo je li pogrešan 1. ili 2. bit pa bi mjere bile: MB(0) = 1, a za MB(1) = 1, jer se u oba slučaja raspoloživa stanja razlikuju za 1 znamenku.

Polazna točka: Izračunata PM[s, i], grafički se prikazuje kao natpis u okviru rešetke i za svako stanje u vremenu i.



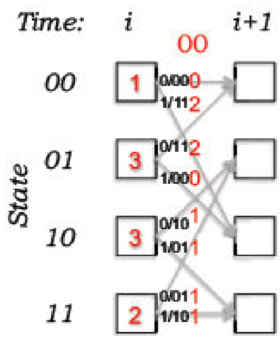
Slika 2.b: Mjere puta upisuju se u kvadrat čvorišta (crvene brojke) u trenutku i

Primjer: PM[00, i] = 1 znači da je otkrivena pogreška od 1 bita kada bi se usporedili primljeni paritetni bitovi s onima što bi se prenosili prilikom slanja najvjerojatnije poruke, uzevši u obzir sve poruke koje se šalju u stanju 00 i u vremenu i.

5. P: Koje je najvjerojatnije stanje s odašiljača u vremenu i?

6. P: Ako je odašiljač u stanju s u vremenu i+1, u kojim bi stanj(u)ima mogao biti u vremenu i?

Primjer: za stanje 01, α = 10 i β = 11.

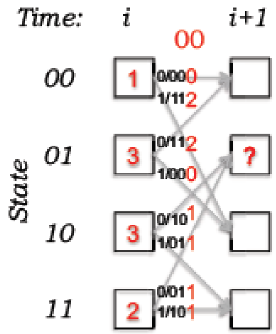


Slika 2.c: Prethodna stanja α i β za stanje 01

Svaki niz poruka koja ostavlja odašiljač u stanju *s* u vremenu *i*+1 mora napustiti odašiljač u stanju α ili β u vremenu *i*.

Primjer nastavak (*cont'd*[2]): za doći u stanju 01 u vremenu *i*+1:

1. Odašiljač je u stanju 10 u vremenu *i*, a *i*-ti bit poruke je 0. Ako je to slučaj, odašiljač je poslao 10 kao paritetne bitove pa postoji jedan pogrešan bit, jer smo primili 00. Ukupna pogreška bita = PM[10, *i*] + 1 = 4, ili
2. Odašiljač je u stanju 11 u vremenu *i*, a *i*-ti bit poruke je 0. Ako je to slučaj, odašiljač je poslao 01 kao paritetne bitove pa postoji jedan pogrešan bit, jer smo primili 00. Za mjeru puta, ukupno pogreška bita = PM[11, *i*] + 1 = 3 što je vjerojatnije?



Slika 2.d: U vremenu i, odašiljač je u stanju 11, a i-ti bit poruke je 0

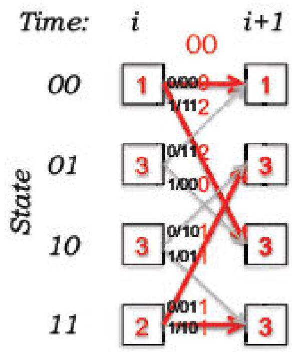
16.1.5. 16.1.5. FORMALAN IZRAČUN BM:

$$PM[s, i+1] = \min(PM[\alpha, i] + BM[\alpha - s], PM[\beta, i] + BM[\beta - s])$$

Primjer:

$$PM[01, i+1] = \min(PM[10, i] + 1, PM[11, i] + 1) = \min(3+1, 2+1) = 3$$

[2] continued

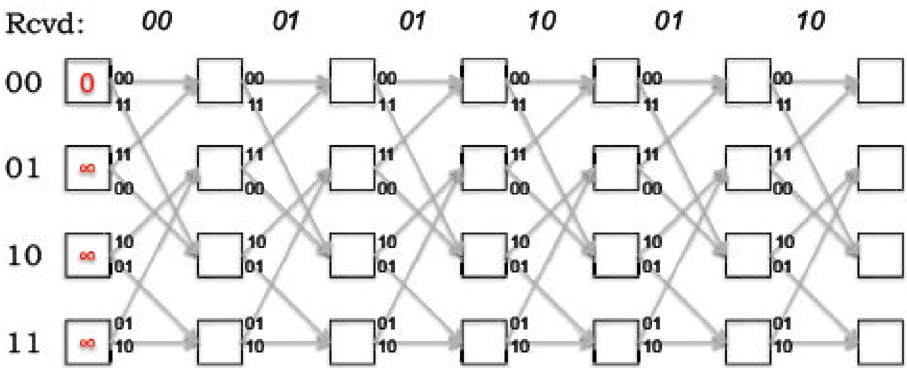


Slika 2.e: Izbor puta najmanje mjere

Bilješke:

1. Zapamtite koji je pravac minimalan; spremljeni lukovi će formirati put kroz rešetke.
2. Ako oba pravac imaju isti ukupan zbroj, dobije se podjednako neriješen rezultat (npr., prilikom računanja PM[11, *i* +1]).

16.1.6. 16.1.6. ALGORITAM DEKODIRANJA TVRDOM ODLUKOM – PROLAZ KROZ REŠETKU

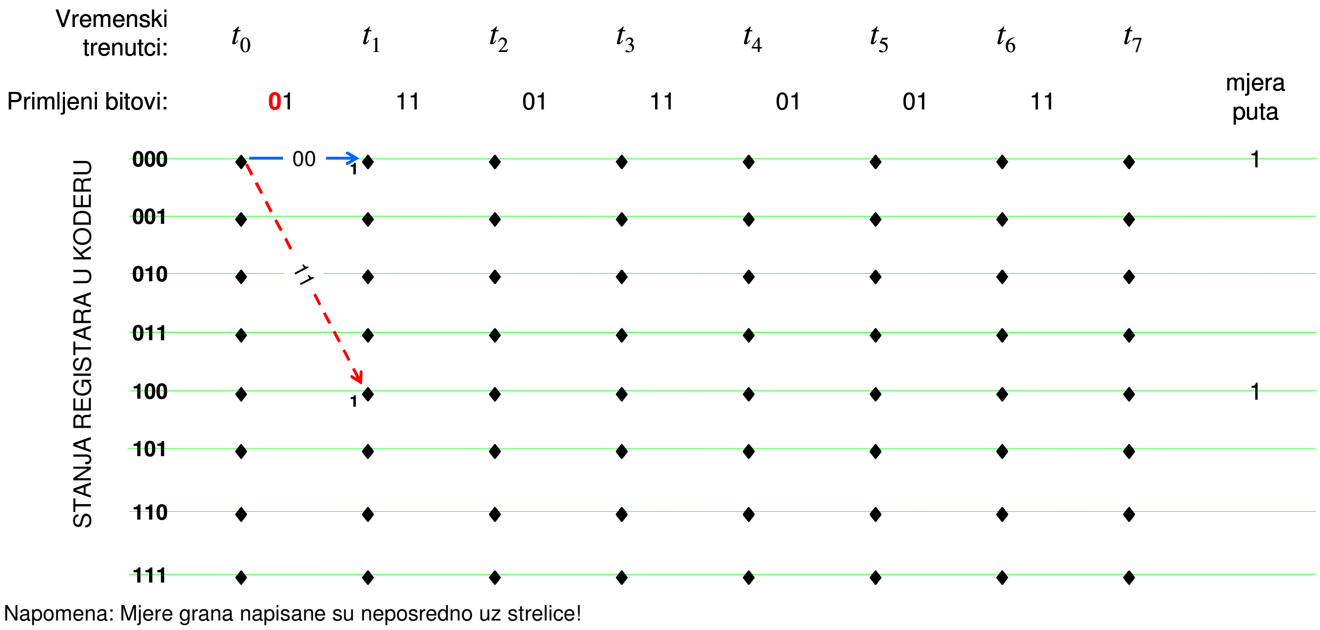


Slika 2.f: Popunjavanje prazne rešetke dekodera ovisi o parametrima kodera

- Metrika puta: broj pogrešaka na putu s najvećom vjerojatnošću do određenoga stanja (min svih putova koji vode do stanja)
- Metrika grane: za svaku strelicu, Hammingova udaljenost između primljenih paritetnih bitova i očekivanih paritetnih bitova.

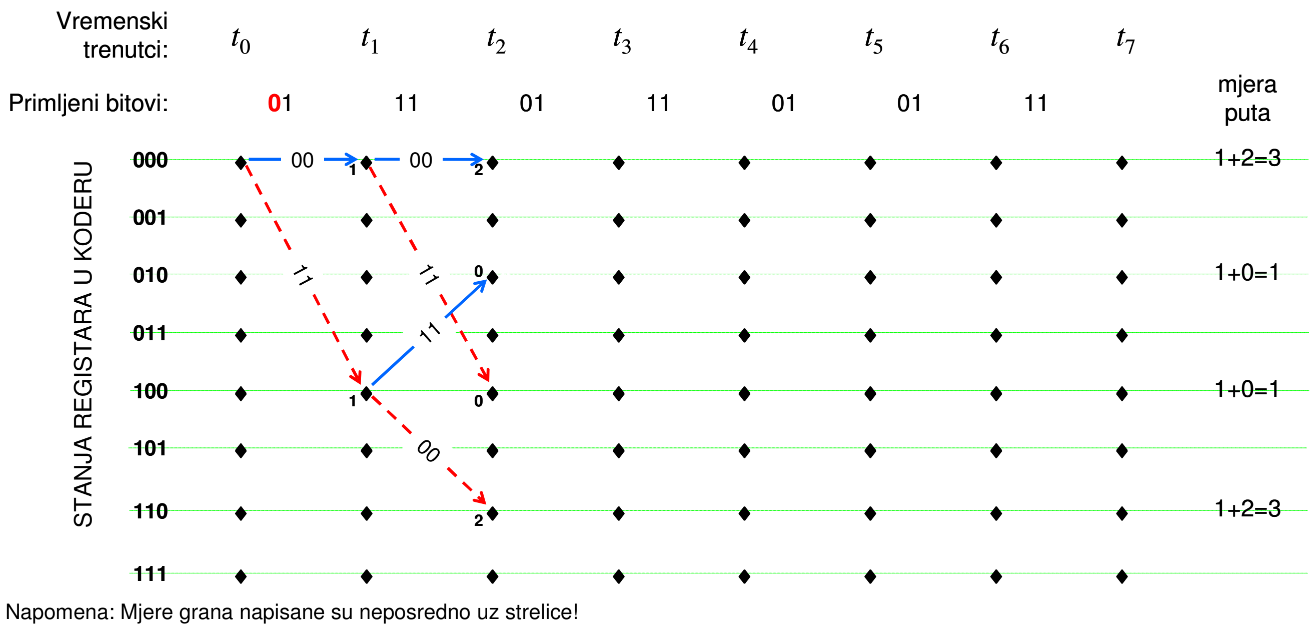
16.2. 16.2. Dekoder s 3 stanja za (2, 1, 4) kôd (isti kao za primjer serijskoga dekodiranja)

Pomoću Viterbijeva dekodiranja dekodirajmo primljen niz [01 11 01 11 01 01 11] (slika 3.a).



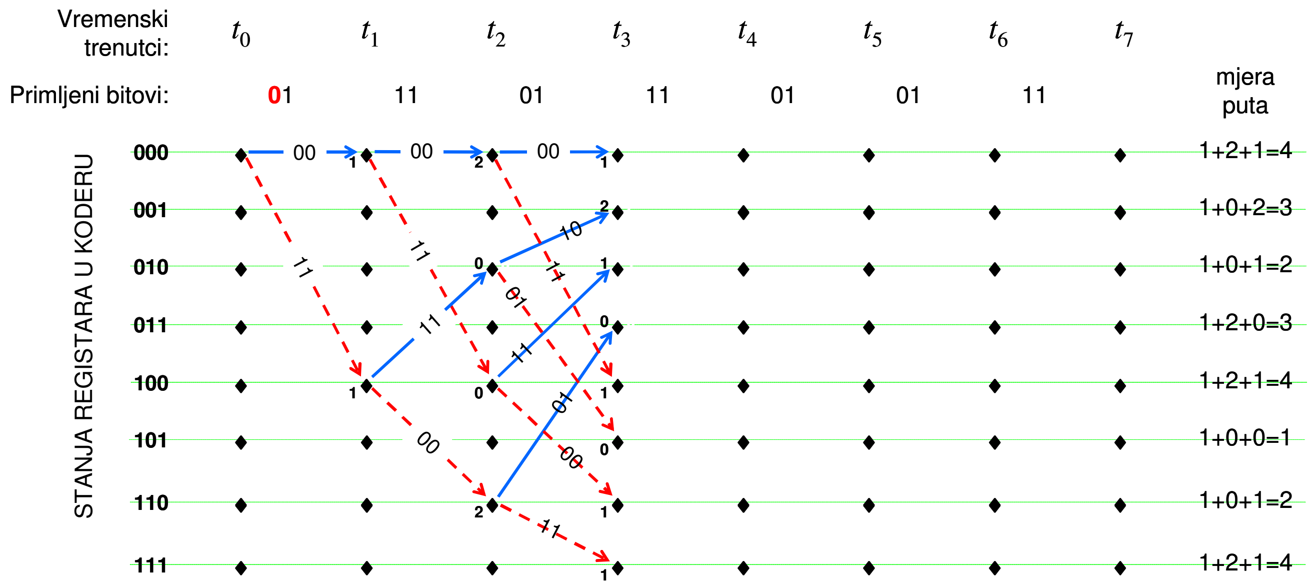
Slika 3.a: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 1

1. U trenutku t_0, primili smo bitove 01. Dekoder uvijek počinje iz stanja 000. Od ovoga čvorišta postoje dvije raspoložive staze, ali one se ne podudaraju s dolaznim bitovima. Dekoder računa mjeru svake grane i nastavlja istodobno (paralelno) duž obje ove grane. Ovo je u suprotnosti sa serijskim dekodiranjem gdje se izbor pravi u svakome čvorištu odluke. *Mjera* za svaku od ove dvije *grane* jednaka je 1, što znači da jedan od dva bita (prvi ili drugi) nije u skladu s dolaznim bitovima (slika 3.b).



Slika 3.b: Viterbijevo dekodiranje za (2, 1, 4) kôd, korak 2

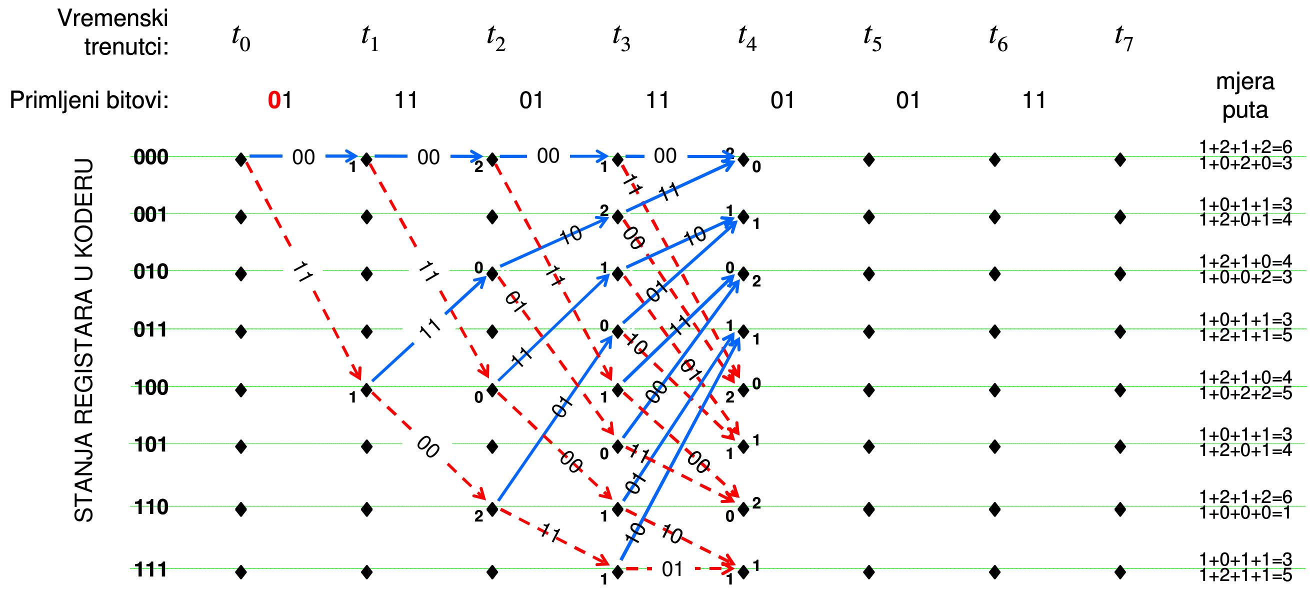
2. U trenutku t_1, dekoder izlazi iz ovih dvaju mogućih stanja i grana se prema četiri stanja. Računaju se mjere za ove grane promatranjem podudarnosti s dolaznim bitovima koji su jednaki 11. Nova mjera grane prikazana je na desnoj strani rešetke (slika 3.c).



Napomena: Mjere grana napisane su neposredno uz strelice!

Slika 3.c: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 3

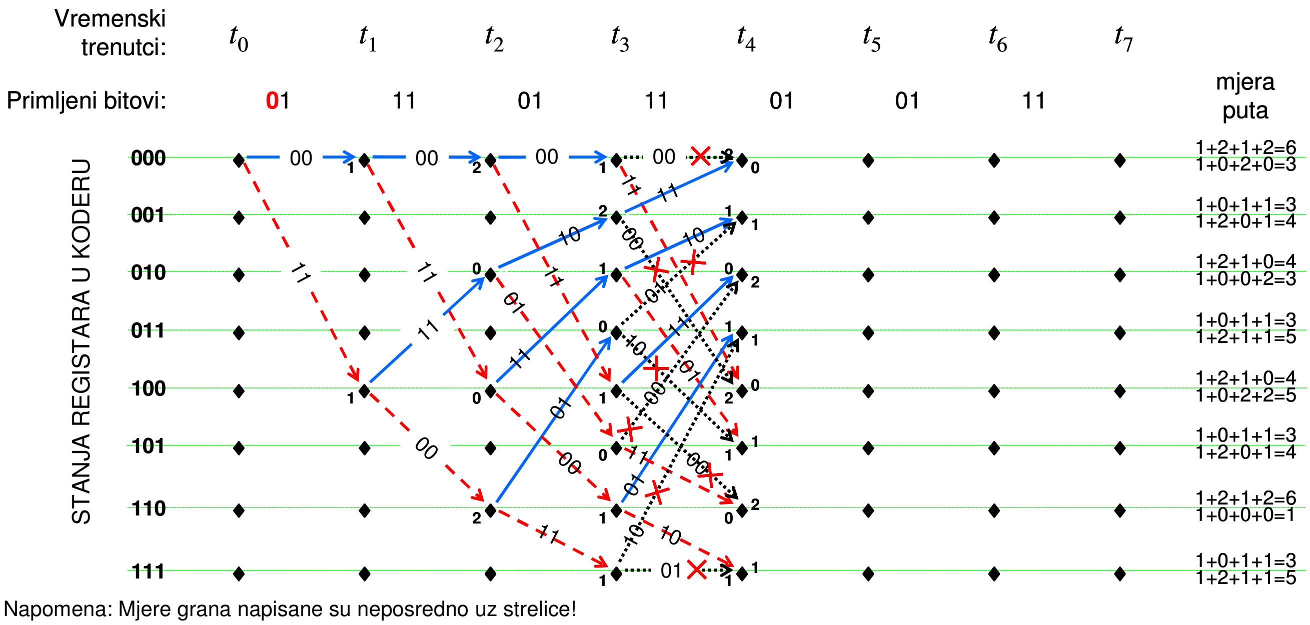
3. U trenutku t_2, rešetka se grana iz četiri u osam stanja za prikazati sve moguće staze. Računaju se mjere putova za primljene bitove 01 (u t_2) i dodaju se prije akumuliranim mjerama puta sve do trenutka t_1 (slika 3.d)



Napomena: Mjere grana napisane su neposredno uz strelice!

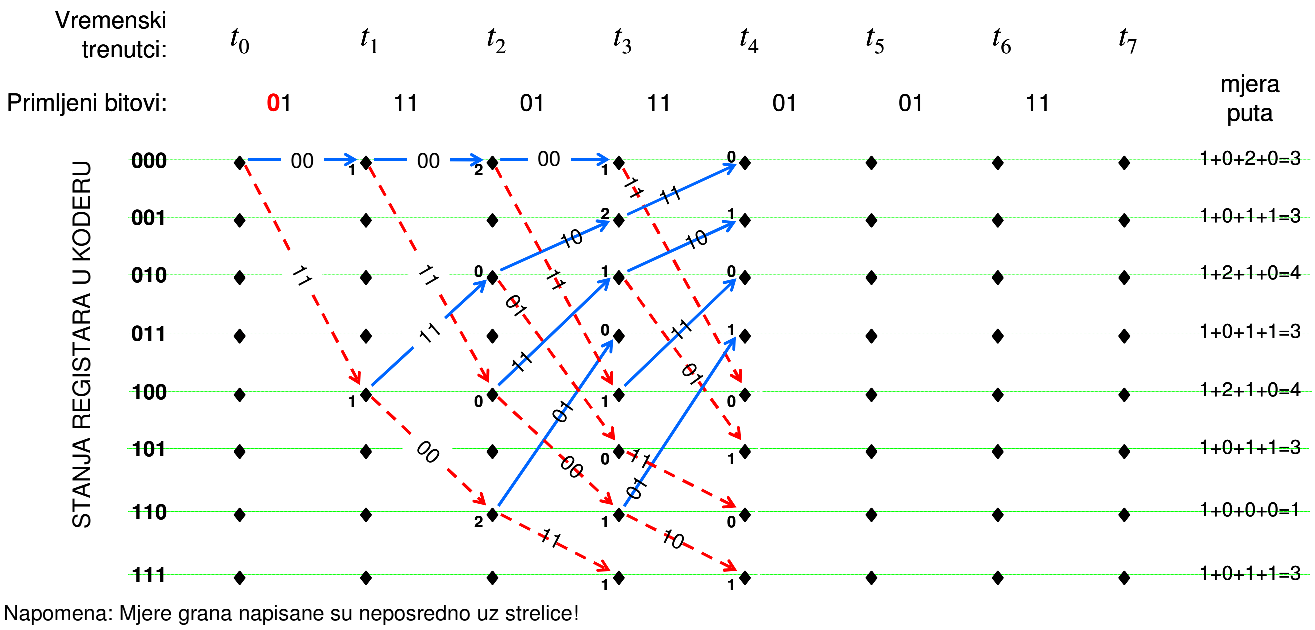
Slika 3.d: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 4

4. U trenutku t_3 iz osam stanja dijagram se grana također prema osam stanja ali za sve kombinacije 0 i 1 što znači da se radi o 16 smjerova prema 8 mogućih stanja (i 16 mjera grana, a treba nam ih 8). Međutim, prikazane grane prema istome čvoru u t_4 imaju različite mjere pa o mjeri grane ovisi "preživljavanje" pojedine grane od čvora u t_3 do čvora u t_4, odnosno staze sve do čvora u t_4. Staze s većom mjerom grane, odbacuju se (slika 3.e).



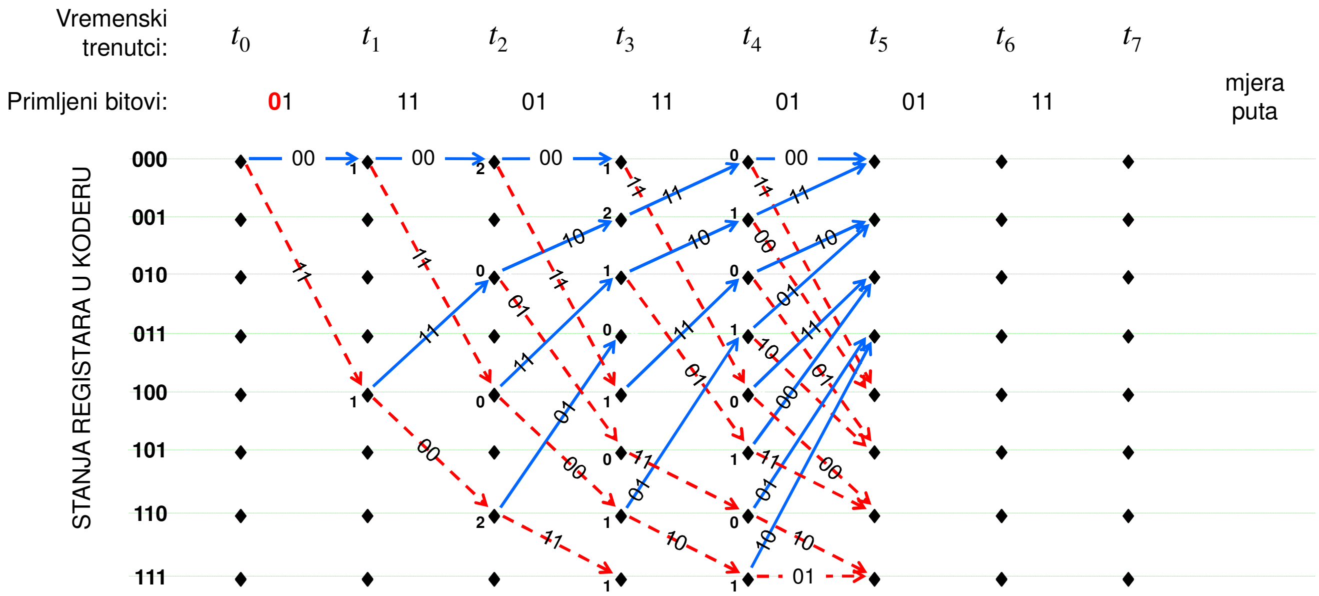
Slika 3.e: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 4 - prije odbacivanja

5. U trenutku t_4, rešetka je u potpunosti ispunjena. Svaki čvor ima barem jedan put što ulazi u njega. Mjere grane i puta prikazane su na slici. Grane za brisanje označene su isprekidanom crnom crtom i znakom ×. U slučaju jednakih mjera grana, briše se samo donja grana (ovo pravilo određujete sami i držite ga se dosljedno do kraja dijagrama, dakle, umjesto donjih, možete odbacivati gornje grane), a ako ima više od dvije grane, ostaje ona s najmanjom mjerom grane. Ako u čvorište ulazi samo jedna grana ona preživljava (slika 3.f).



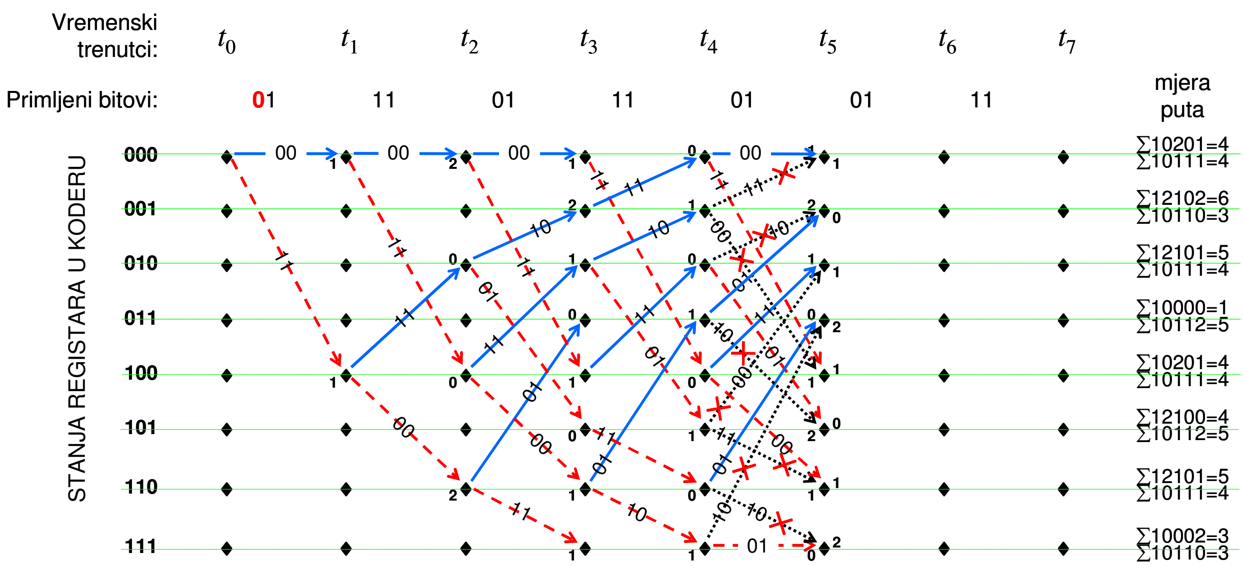
Slika 3.f: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 4 - nakon odbacivanja

6. Na slici su prikazane samo preživjele grane do trenutka t_4. Od ovoga trenutka, slika dijela rešetke između trenutaka t_4 i t_5, ista je kao i ona između trenutaka t_3 i t_4 (slika 3.g).



Napomena: Mjere grana napisane su neposredno uz strelice!

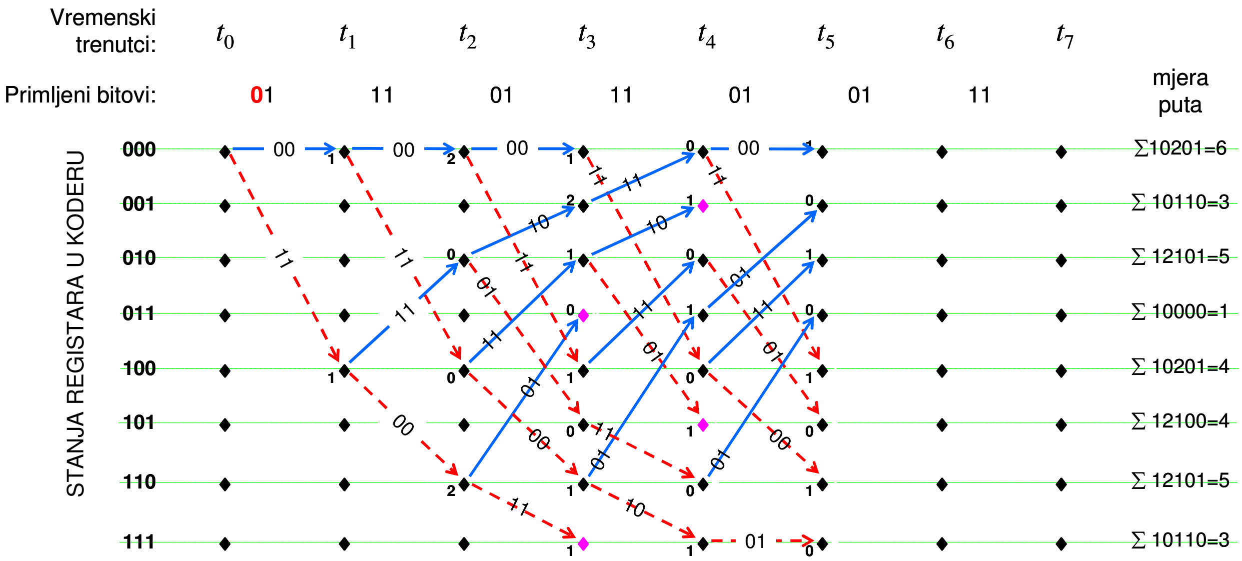
Slika 3.g: Viterbijevo dekodiranja za (2, 1, 4) *kôd*, *korak 5*Na isti način kao i do sada, odrede se mjere grana te se odbace one grane s većom mjerom, a prežive one grane s manjom mjerom. U trenutku t_4 ponavlja se crtanje grana kao i u trenutku t_3. Zbog drugačije primljene kombinacije od dva bita, odabir i odbacivanje grana je drugačije. Za svaku od staza što ulaze u čvor, dane su mjere grane i mjere puta (slika 3.h).



Napomena: Mjere grana napisane su neposredno uz strelice!

Slika 3.h: Viterbijevo dekodiranja za (2, 1, 4) *kôd*, *korak 6*

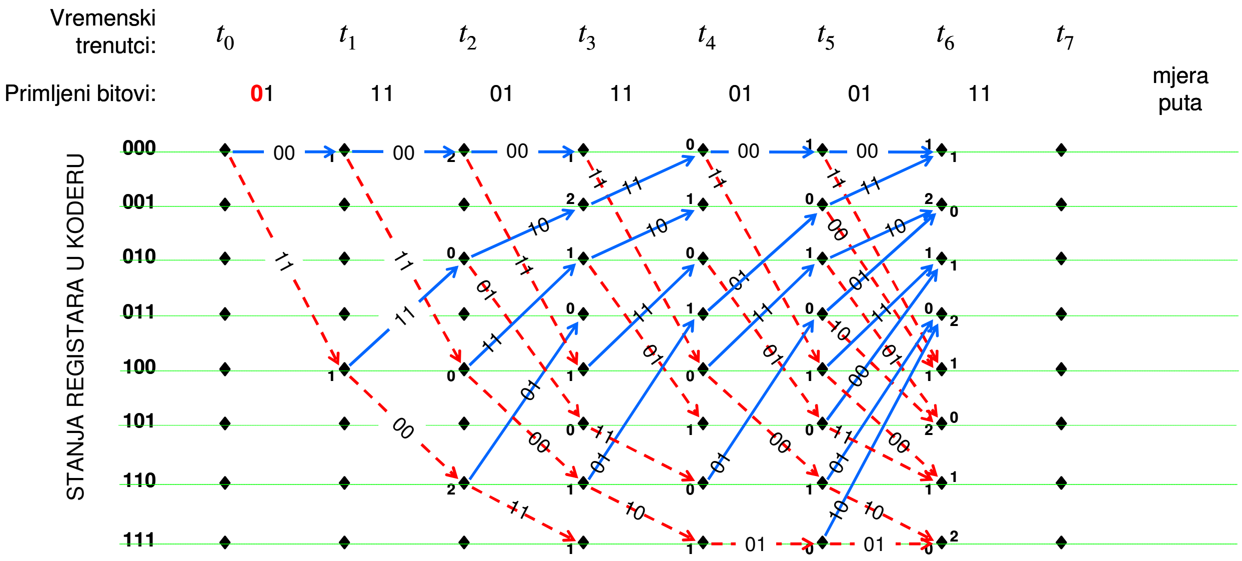
8. Do trenutka t_5, dijagram je prikazan na slici se stazama prema naprijed i sada se počinju približavati čvorovima. Prema načelu najveće vjerojatnosti, u svakome čvoru odbacujemo put s višom mjerom grane (a time se odbacuju i putovi s višim mjerama), jer su oni najmanje vjerojatni. Ovo odbacivanje puta, u svakome čvoru pomaže smanjenju broja staza koje treba ispitati i potpomaže učinkovitost Viterbijeve metode (slika 3.i).



Napomena: Mjere grana napisane su neposredno uz strelice!

Slika 3.i: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 7

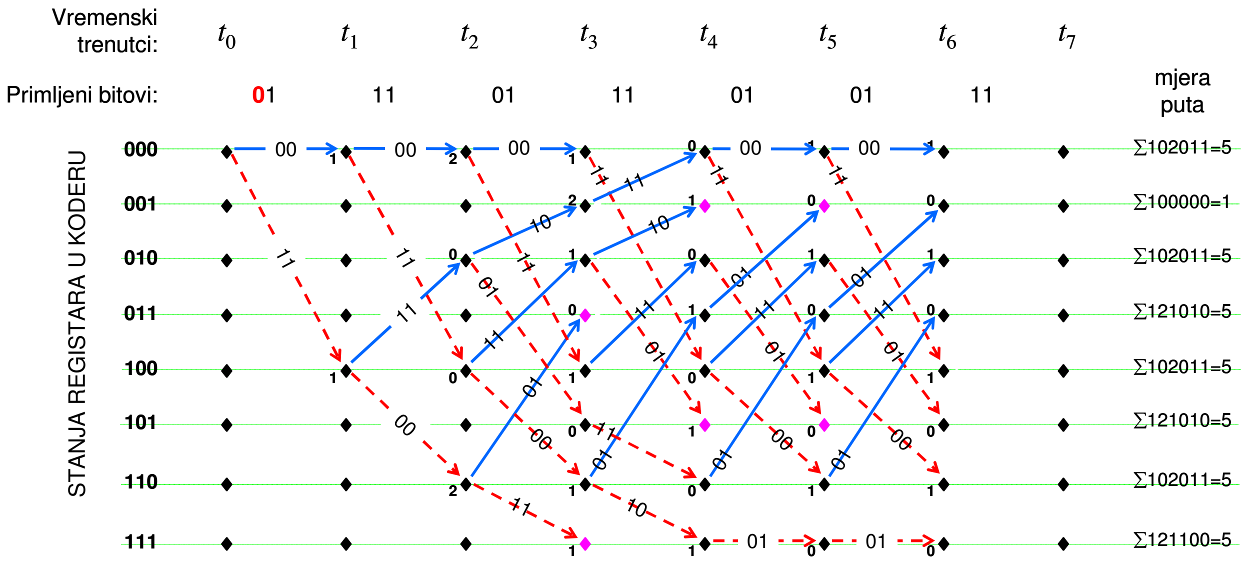
9. Do trenutka t_5, dijagram je prikazan na slici se stazama prema naprijed i sada se počinju približavati čvorovima. Prema načelu najveće vjerojatnosti, u svakome čvoru odbacujemo put s višom mjerom grane (a time se odbacuju i putovi s višim mjerama), jer su oni najmanje vjerojatni. Ovo odbacivanje puta, u svakome čvoru pomaže smanjenju broja staza koje treba ispitati i potpomaže učinkovitost Viterbijeve metode (slika 3.j).



Napomena: Mjere grana napisane su neposredno uz strelice!

Slika 3.j: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 8

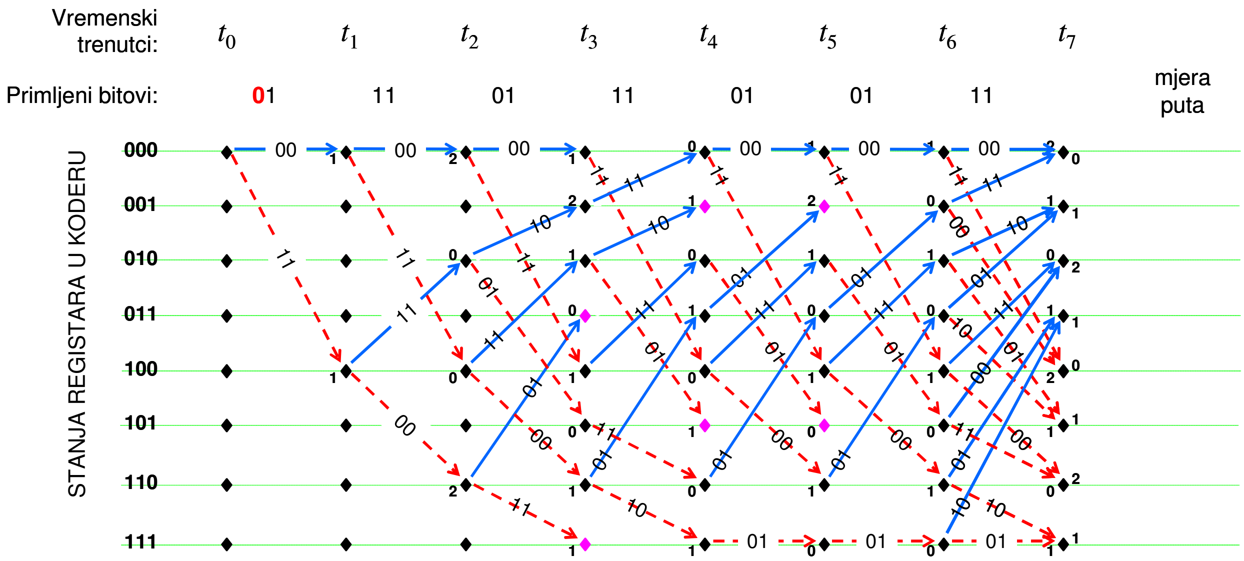
10. Postupak se ponavlja u sljedećem trenutku t_5 s istim dijagramom kao i u prethodna dva koraka. Nakon određivanja mjera grana, opet se odbacuju grane prema prethodno opisanim pravilima (slika 3.k).



Napomena: Mjere grana napisane su neposredno uz strelice!

Slika 3.k: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 9

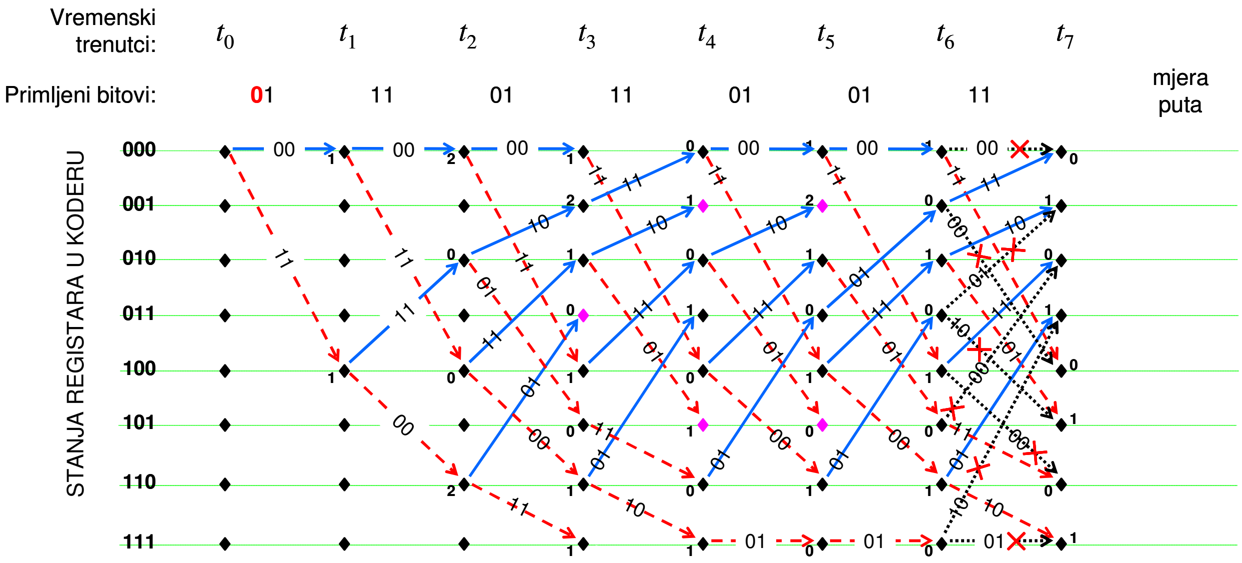
11. Sada, u svakome čvoru, imamo jedan ili više putova što se razdvajaju. Mjere za sve staze nalaze se na desnoj strani. U svakome čvoru, „držimo“ se samo puta s najnižom mjerom, a odbacujemo sve druge putove, što prikazuje crveno obojana crta. Nakon odbacivanja staza s najvećom mjerom puta, imamo sljedeće preostale staze. Prikazana mjera je „pobjednik“ puta (slika 3.l).



Napomena: Mjere grana napisane su neposredno uz strelice!

Slika 3.l: Viterbijevo dekodiranja za (2, 1, 4) kôd, korak 10

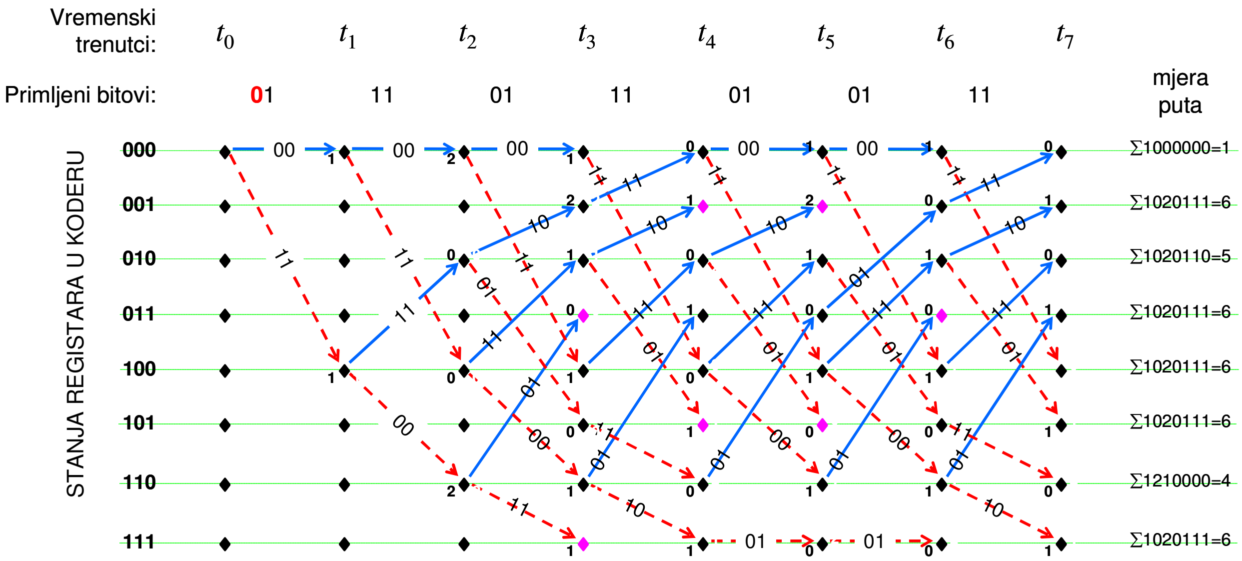
12. U trenutku t_6, ponavljamo prethodan postupak, a nakon odbacivanja grana, kako je prikazano, završavamo dekodiranje u t_7 računanjem novih mjera grana. U ovome čvoru, ponovno se staze približavaju i opet odbacujemo one s većim mjerama (slika 3.m).



Napomena: Mjere grana napisane su neposredno uz strelice!

Slika 3.m: Viterbijevo dekodiranje za (2, 1, 4) kôd, korak 11

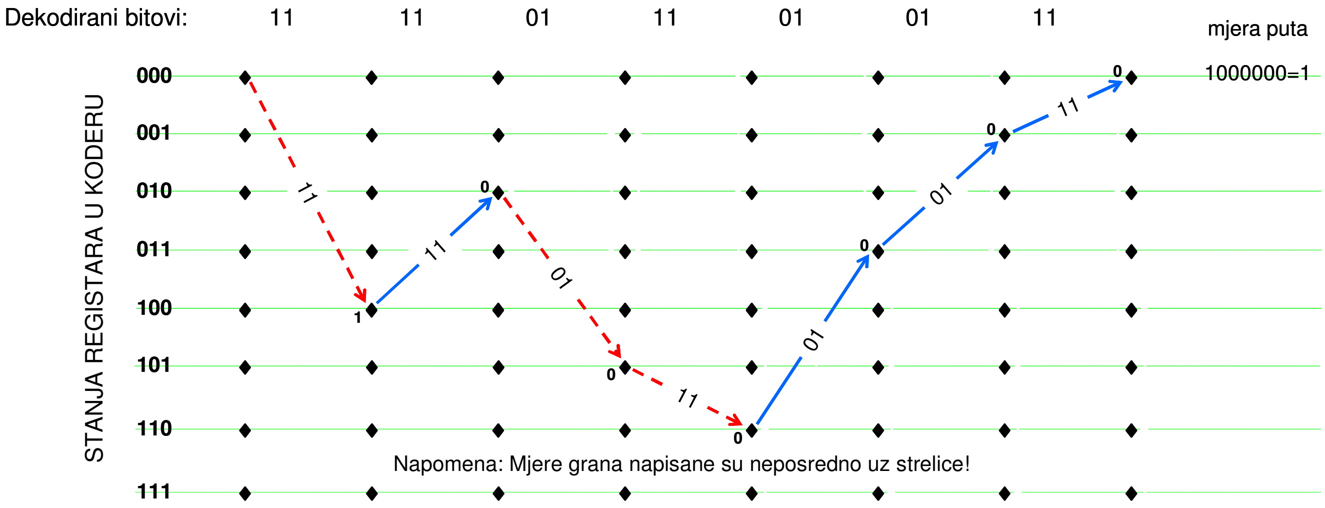
13. U trenutku $t = 6$, primamo bitove 11. Opet se računaju mjere za sve staze. Odbacujemo sve veće mjere grana, a zadržavamo one grane s manjom mjerom u tome čvorištu (slika 3.n).



Napomena: Mjere grana napisane su neposredno uz strelice!

Slika 3.n: Viterbijevo dekodiranje za (2, 1, 4) kôd, korak 12

14. U sedmome koraku rešetka završava. Sada tražim put s najmanjom mjerom puta i otkrivamo pobjednika. U ovome trenutku (t_7) do svakoga stanja u dijagramu vodi samo jedan put kojega karakteriziram *ukupna mjera puta*. Onaj put s najmanjom *mjerom puta* je pobjednik i kombinacija bitova koju on predstavlja, najvjerojatnija je poslana kombinacija od kodera preko kanala do dekodera (slučaj *dekodiranja tvrdom odlukom*). Dodatna potvrda ispravnosti postupka je to što "pobjednička" putanja počinje i završava u stanju 000 (jer se i kodna kombinacija iz kodera dobila "ispiranjem" kodera nulama (stanje sve nule) (slika 3.o).



Slika 3.o: Viterbijevo dekodiranje za (2, 1, 4) *kôd, korak 12*

15. Put što ga pratimo prema stanjima 000, 100, 010, 101, 110, 011, 001, 000 i pošto on odgovara bitovima 1011000, predstavlja dekodiran niz. Ovaj dijagram isti je onome dijagramu dobivenom serijskim dekodiranjem. Put započinje i završava u stanju sve nule.

Duljina ove rešetke bila je 4 bita + *m* bitova. Idealno, ovo bi trebalo biti jednako duljini poruke, ali zbog postupka skraćivanja i zahtjeva skladištenje, može se smanjiti pa dekodiranje ne treba kasniti do kraja prijenosa niza. Tipična duljina sakaćenja konvolucijskoga kodera je 128 bita ili 5 - 7 puta veća od duljine ograničenja.

Viterbijevo dekodiranje je vrlo važno, jer ono također vrijedi i za dekodiranje blok-kodova. Ovaj oblik dekodiranja rešetkom također se koristi za *modulaciju kodirane rešetke* TCM (*Trellis-Coded Modulation*).