

(19)



(10) **LT 4754 B**

(12) **PATENTO APRAŠYMAS**

- (11) Patento numeris: **4754** (51) Int. Cl.⁷: **H03M 5/14**
G11B 20/14
- (21) Paraiškos numeris: **2000 008**
- (22) Paraiškos padavimo data: **2000 01 26**
- (41) Paraiškos paskelbimo data: **2000 09 25**
- (45) Patento paskelbimo data: **2001 01 25**
- (62) Paraiškos, iš kurios dokumentas išskirtas, numeris: —
- (86) Tarptautinės paraiškos numeris: **PCT/IB99/00948**
- (86) Tarptautinės paraiškos padavimo data: **1999 05 25**
- (85) Nacionalinio PCT lygio procedūros pradžios data: **2000 01 26**
- (30) Prioritetas: **10/105280, 1998 05 29, JP**
- (72) Išradėjas:
A. Kornelis SCHOUHAMER-IMMINK, NL
A., H., M., Joseph KAHLMAN, NL
J. Gijsbert VAN-DEN-ENDEN, NL
T. NAKAGAWA, NL
Y. SHIMPUKU, NL
T. NARAHARA, NL
K. NAKAMURA, NL
- (73) Patento savininkas:
KONINKLIJKE PHILIPS ELECTRONICS N.V.
Groenewoudseweg 1, NL-5621 BA Eindhoven, NL
SONY CORPORATION, 6-7-35, Kitashina-gawa, Kitashinagawa, Tokyo 141, JP
- (74) Patentinis patikėtinis:
Rita LAURINAVIČIŪTĖ, UAB "Metida", Pilies g. 8/1-2, 2600 MTP Vilnius, LT

- (54) Pavadinimas:
Moduliavimo/demoduliavimo įrenginys ir būdas su nuoseklios minimalios sekos ilgio ribojimu

- (57) Referatas:
Išradimo tikslas - įrašyti ir atkurti duomenis dideliu linijiniu tankiu. DSV kontrolės bitų nustatymo/įterpimo įrenginys (2) įterpia DSV kontrolės bitus, skirtus vykdyti DSV kontrolę, į įvesties duomenų eilutę ir išveda duomenų

eilutę su DSV kontrolės bitais į moduliavimo įrenginį (3). Moduliavimo įrenginys (3) konvertuoja duomenų eilutę, kurios pagrindinis duomenų ilgis yra 2 bitai, į kintamo ilgio kodą, kurio pagrindinis kodo ilgis yra 3 bitai, pagal konversijos lentelę ir išveda kodą, gautąjį konversijos metu, į NRZI kodavimo įrenginį (4). Moduliavimo įrenginiu (3) naudojama konversijos lentelė turi pakaitos kodus, skirtus riboti iki iš anksto nustatytos reikšmės minimalios sekos nuoseklių atvejų skaičių, ir pakaitos kodus, skirtus išlaikyti sekos ilgio ribą. Be to, konversijos lentelė įgalina konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui.

Išradimas skirtas moduliavimo įrenginiui ir moduliavimo būdai, demoduliavimo įrenginiui ir demoduliavimo būdai, o taip pat programos laikmenai, naudojamiems įrašyti duomenis dideliu tankiu į įrašo laikmeną ir atkurti įrašytus laikmenoje dideliu tankiu duomenis.

Prieš perduodant duomenis perdavimo linija ar įrašant į įrašo laikmeną, pavyzdžiui, magnetinį diską, optinį diską ar magnetooptinį diską, jie yra moduliuojami į kodą, atitinkantį perdavimo liniją ar įrašo laikmeną. Žinomas moduliavimo būdas, aprašytas, pavyzdžiui, EP 902544, yra blokinis kodavimas. Jo metu duomenų eilutė yra sugrupuojama į blokus, kiekvienas turi $m \times i$ bitų. Kiekviena grupė, toliau vadinama duomenų žodžiu, yra tuomet konvertuojama į kodinį žodį, turintį $n \times i$ bitų, pagal tinkamą kodavimo taisyklę. Kuomet $i=1$, šis kodinis žodis yra fiksuoto ilgio kodas. Tuo atveju, jei i turi daugybę reikšmių diapazone nuo 1 iki $i_{\max}$, gautasis kodinis žodis yra kintamojo ilgio kodas. Paprastai gautasis blokiniu kodavimu kodas yra išreiškiamas kaip kintamojo ilgio kodas ($d, k; m, n; r$).

Čia i yra ribinis ilgis ir r yra $i_{\max}$, maksimalus ribinis ilgis. d yra minimalus "0" tarp dviejų nuoseklių "1" skaičius. Iš kitos pusės, k yra maksimalus "0" tarp dviejų nuoseklių "1" skaičius. k yra maksimali "0" vora.

Beje, kintamojo ilgio kodo, gautojo aukščiau aprašytuoju blokiniu kodavimu, įrašymo į laikmeną, tokią kaip optinis diskas ar magnetooptinis diskas, pavyzdžiui, kompaktinis diskas (CD) ar minidiskas (MD), metu kintamojo ilgio kodas paveikia NRZI (negrįžties prie invertuoto nulio) moduliaciją, kurioje kiekvienas kintamojo ilgio kodo "1" yra interpretuojamas kaip inversija, kai, tuo tarpu, "0" yra interpretuojamas kaip inversijos nebuvimas. Tuomet yra įrašomas kintamojo ilgio kodas, užbaigiantis NRZI moduliaciją. Kintamojo ilgio kodas, užbaigiantis NRZI moduliaciją, yra įrašymo bangų vora. Naudojant magnetooptinį diską, atitinkantį pirmąsias ISO specifikacijas ir turintį nedidelį įrašymo tankį, bitų, užbaigiančių įrašymo moduliaciją, seka įrašoma be pakitimų, t.y. nepaveikiant NRZI moduliacijos.

Tarkime, žymėjimai $T_{\min}$ ir $T_{\max}$ žymi įrašymo bangos voros atitinkamai minimalius ir maksimalius inversijos periodus. Šiuo atveju, norint įrašyti įrašymo bangos vorą dideliu tankiu linijinio greičio kryptimi, reikalingas ilgas minimalios

inversijos periodas $T_{\min}$ ar didelė minimali seka d. Be to, taktinių impulsų generavimo požiūriu, pageidaujama turėti trumpą maksimalios inversijos periodą $T_{\max}$ ar mažą maksimalią seką k. Šiuos reikalavimus patenkinti yra pasiūlyti įvairūs moduliavimo būdai.

Kalbant konkrečiai, optinių diskų, magnetinių diskų ar magnetooptinių diskų įrašymui yra pasiūlyti ar netgi naudojami moduliavimo būdai, skirti generuoti kintamojo ilgio kodą RLL (1 - 7), kuris dar yra išreikštas kaip (1, 7 ; m, n ; r), ir kintamojo ilgio kodą RLL (2 - 7), dar išreikštą kaip (2, 7 ; m, n ; r), o taip pat fiksuotojo ilgio kodą RLL (1 - 7), taip pat išreikštą kaip (1, 7 ; m, n ; 1), naudojamą ISO specifikacijoje MO. Šiuo metu tiriamuose ir tobulinamuose diskiniuose įrenginiuose, tokiuose kaip optiniai diskai ir magnetooptiniai diskai, turinčiuose didelį įrašymo tankį, paprastai naudojamas RLL kodas (riboto sekos ilgio kodas) su minimalia seka d.

Žemiau pateikta kintamojo ilgio RLL (1 - 7) kodo konversijos lentelė.

1 LENTELĖ

RLL (1, 7 ; 2, 3 ; 2)

	Duomenys	Kodas
i=1	11	00x
	10	010
	01	10x
i=2	0011	000 00x
	0010	000 010
	0001	100 00x
	0000	100 010

Konversijos lentelėje naudojamo simbolio x reikšmė yra "1" po to einančiam informacijos kanalo bitui "0" arba reikšmė "00" po to einančiam informacijos kanalo bitui "1". Maksimalus ribinis ilgis r yra 2.

Kintamojo ilgio kodo RLL (1 - 7) parametrai yra (1, 7 ; 2, 3 ; 2). Minimalus inversijos periodas $T_{\min}$, kuris gali būti išreikštas formule $(d+1)T$, yra, tuo būdu, lygus $2(=1+1)T$, kur T yra bitų tarpelis įrašymo bangos voroje. Minimalus inversijos periodas $T_{\min}$, kuris gali būti taip pat išreikštas formule $(m/n) \times 2T_{data}$, yra, tuo būdu, lygus $1,33(=2/3 \times 2)T_{data}$, kur T_{data} yra bitų tarpelis duomenų eilutėje. Maksimalus

inversijos periodas $T_{\max}$, kuris gali būti išreikštas formule $(k+1)T$, yra, tuo būdu, lygus $(7+1)T=8T=8 \times (m/n)T_{\text{data}}=8 \times 2/3 T_{\text{data}}=5,33T_{\text{data}}$. Detekcijos lango plotis T_w , kuris gali būti išreikštas formule $(m/n)T_{\text{data}}$, yra, tuo būdu, $0,67(=2/3)T_{\text{data}}$.

Beje, pastebėta, kad informacijos kanalo bitų, užbaigiančių RLL(1-7) moduliaciją, pavaizduotą 1 lentelėje, voroje po generavimo dažnio, atitinkančio $2T$ periodą, kuris yra lygus minimaliam inversijos periodui $T_{\min}$, seka generavimo dažniai, atitinkantys $3T$ ir $4T$ periodus. Tai, kad daugybė impulsinės informacijos generuojama per trumpus intervalus, tokius kaip $2T$ ir $3T$, dažnai yra naudinga sinchronizavimo signalų generacijai.

Nežiūrint į tai, kad įrašymo linijos tankis toliau didėja, minimali seka, atvirkščiai, šiuokart tampa problema. T.y., jei minimalios sekos $2T$ yra generuojamos nuosekliai, įrašymo bangos vora yra linkusi iškraipymams, generuotiems joje. Tai yra dėl to, kad $2T$ bangos išvestis yra mažesnė, nei kitų bangų išvestys, ir, tuo būdu, lengvai paveikiama, pavyzdžiui, defokusavimo ir tangentinio nukrypimo faktoriais. Be to, esant dideliame įrašymo tankiui, nuoseklių minimalių ženklų ($2T$) įrašymas yra taip pat lengvai paveikiamas tokiais trukdžiais, kaip triukšmas. Tuo būdu, duomenų atkūrimas gali būti klaidingas. Šiuo atveju, duomenų atkūrimo klaidos derinys dažniausiai stebimas kaip minimalaus ženklo priekinio ir galinio impulso postūmiai. Dėl to generuoto bito klaidos ilgis padidėja.

Kaip aprašyta aukščiau, prieš perduodant duomenis perdavimo linija arba įrašant į įrašo laikmeną, jie yra moduluojami į kodą, atitinkantį perdavimo liniją ar įrašo laikmeną. Jei moduliacijos metu gautasis kodas turi nuolatinės srovės komponentą, nesunkiai generuojamos įvairios klaidos, pavyzdžiui, sekimo įrašo takeliu klaidos, generuotos diskiniame kaupiklyje, ar virpesiai. Dėl šios priežasties yra pageidautina, kad moduluotame kode nebūtų nuolatinės srovės komponento.

Siekiant, kad moduluotame kode nebūtų nuolatinės srovės komponento, yra pasiūlyta DSV (Digital Sum Value - skaitmeninė sumos reikšmė) kontrolė. DSV yra suma, gauta sudedant bitų (informacijos simbolių) voros reikšmes, kur reikšmės $+1$ ir -1 yra priskirtos atitinkamai voros, kuri susidaro informacijos kanalo bitų NRZI moduliacijos (t.y. lygio šifravimo) metu, vienetui "1" ir nuliui "0". DSV yra nuolatinės srovės komponento, esančio kodų voroje, indikatorius. DSV absoliučios reikšmės sumažinimas DSV kontrolės metu yra lygus kodų voroje esančio nuolatinės srovės

komponento įtakos sumažinimui. DSV yra kodų voroje esančio nuolatinės srovės komponento indikatorius. Absoliučios DSV reikšmės sumažinimas DSV kontrole yra ekvivalentiškas kodų voroje esančio nuolatinės srovės komponento reikšmės slopinimui.

DSV kontrolė netaikoma moduliacijos kodui, generuotam pagal kintamo ilgio RLL(1 – 7) lentelę, aukščiau pateiktą kaip 1 lentelė. Tokiu atveju DSV kontrolė vykdoma, apskaičiuojant koduotų bitų (informacijos kanalo bitų) voros DSV, praėjus iš anksto nustatytam moduliacijos laikotarpiui ir įterpus iš anksto nustatytą DSV kontrolės bitų skaičių į užkoduotų bitų vorą (informacijos kanalo bitų vorą).

Esant bet kokiam greičiui, DSV kontrolės bitai iš esmės yra pertekliniai bitai. Jei turi būti priimamas dėmesio kodų konversijos efektyvumas, yra pageidautina sumažinti DSV kontrolės bitų skaičių iki kuo mažesnės įmanomos reikšmės.

Be to, jei yra įterpti DSV kontrolės bitai, taip pat yra pageidautina palikti minimalią seką d ir maksimalią seką k nepakeistomis. Tai yra dėl to, kad (d, k) pakeitimai paveiks įrašymo/atkūrimo charakteristikas.

Kaip jau minėta, įrašant dideliu linijiniu tankiu RLL kodą ar atkuriant įrašytą dideliu tankiu RLL kodą susiduriama su problema, kad nuoseklių minimalių sekų d derinys išsaus lengvai generuojamą ilgalaikę klaidą.

Be to, turint tokį RLL kodą, kaip RLL(1 – 7) kodas, DSV kontrolė priverčia įterpti DSV kontrolės bitus į kodinių žodžių eilutės (informacijos kanalo bitų voros) arbitražinę dalį. Nors DSV kontrolės bitai yra iš esmės pertekliniai bitai, tačiau yra pageidautina sumažinti įterptų DSV kontrolės bitų skaičių iki kuo mažesnės įmanomos reikšmės. Siekiant išlaikyti pastovias minimalios ir maksimalios sekų reikšmes, mažiausias DSV kontrolės bitų skaičius yra 2. Tačiau yra pageidautina sumažinti DSV kontrolės bitų skaičių net iki mažesnės reikšmės.

Šis išradimas skirtas išspręsti aukščiau minėtas problemas. Išradimo tikslas yra vykdyti DSV kontrolę, reikalingą sukurti labai efektyvius kontrolės bitus RLL kode $(d, k; m, n)$, kur minimali seka $d=1$, t.y. RLL kodą $(1, 7; 2, 3)$ taip, kad nuoseklių minimalių sekų skaičius būtų sumažintas, išlaikant minimalią ir maksimalią sekas.

Kitas šio išradimo tikslas yra išvengti demoduliacinio klaidos sklaidimo, naudojant kuo paprastesnės konfigūracijos konversijos lentelę.

Moduliavimo įrenginys pagal 1 apibrėžties punktą pasižymi tuo, kad turi konversijos priemones, skirtas konvertuoti įvesties duomenis į kodą pagal konversijos lentelę, įgalinančią konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$, kur d yra minimali seka, o k yra sekos trukmės riba;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą k .

Moduliavimo būdas pagal apibrėžties 23 punktą pasižymi tuo, kad turi konversijos etapą, konvertuojantį įvesties duomenis į kodą pagal konversijos lentelę, įgalinančią konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$, kur d yra minimali seka, o k yra sekos trukmės riba;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą k .

Programos laikmena pagal apibrėžties 24 punktą, skirta programai, vykdančiai dorojimą, apimančią konversijos etapą, konvertuojantį įvesties duomenis į kodą pagal konversijos lentelės duomenis moduliavimo įrenginyje, skirtą konvertuoti m bitų ilgio pagrindinius duomenis į kintamo ilgio kodą ($d, k ; m, n ; r$), turintį n bitų ilgio pagrindinį kodą, pasižymi tuo, kad konversijos lentelė įgalina konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$, kur d yra minimali seka, o k yra sekos trukmės riba;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą k .

Demoduliacinio įrenginio pagal apibrėžties 25 punktą pasižymi tuo, kad turi konversijos priemones, skirtas konvertuoti įvesties kodą į kodą pagal konversijos lentelę, kur ši konversijos lentelė įgalina konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$, kur d yra minimali seka, o k yra sekos trukmės riba;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą k .

Demoduliacinio būdas pagal apibrėžties 28 punktą pasižymi tuo, kad turi konversijos etapą, skirtą konvertuoti įvesties duomenis į duomenis pagal konversijos lentelę, kur ši konversijos lentelė įgalina konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$, kur d yra minimali seka, o k yra sekos trukmės riba;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą k .

Programos laikmena pagal apibrėžties 29 punktą, skirta programai, apimančiai konversijos etapą, konvertuojantį įvesties duomenis į kodą pagal konversijos lentelės duomenis moduliavimo įrenginyje, skirtą konvertuoti kintamo

ilgio kodą ($d, k; m, n; r$), turintį n bitų ilgio pagrindinį kodą, į duomenis, turinčius m bitų ilgio pagrindinius duomenis, pasižymi tuo, kad konversijos lentelė įgalina konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$, kur d yra minimali seka, o k yra sekos trukmės riba;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą k .

Pagal moduliavimo įrenginį, apibūdintą apibrėžties 1 punkte, moduliavimo būdą, apibūdintą apibrėžties 23 punkte, programos laikmeną, apibūdintą apibrėžties 24 punkte, demoduliavimo įrenginį, apibūdintą apibrėžties 25 punkte, demoduliavimo būdą, apibūdintą apibrėžties 28 punkte, ir programos laikmeną, apibūdintą apibrėžties 29 punkte, konversija vykdoma pagal konversijos lentelę, įgalinančią konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą.

Prieš pradėdant aiškinti tinkamiausius išradimo įgyvendinimo variantus, siekiant išaiškinti šio išradimo sąryšio priemones, po kiekvienos priemonės apibūdinimo seka jos tipinis aprašymas skliaustuose, t.y. "priemonė (pavyzdžiui, tipinė priemonė)". Neverta net užsiminti, kad tipinė priemonė nėra susiaurinantis terminas. T.y. priemonė nebūtinai apribota tipiniu įdiegimu, susietu su priemone.

Moduliavimo įrenginys pagal apibrėžties 1 punktą pasižymi tuo, kad turi konversijos priemonę (pavyzdžiui, fig.12 pavaizduotąjį moduliavimo įrenginį), skirtą konvertuoti įvesties duomenis į kodą pagal konversijos lentelę (pavyzdžiui, 2

lentelę), kur ši konversijos lentelė įgalina konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$, kur d yra minimali seka, o k yra sekos trukmės riba;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą k .

Moduliavimo įrenginys pagal apibrėžties 10 punktą dar pasižymi tuo, kad turi sinchronizavimo signalų įterpimo priemonę (pavyzdžiui, sinchronizavimo signalų įterpimo įrenginį 25, parodytą fig.9), skirtą įterpti sinchronizavimo signalą, turintį unikalų derinį, kurio neturi šios konversijos lentelės konversijos kodai, į bet kurią arbitražinę poziciją kodinių žodžių eilutėje.

Pagal apibrėžties 21 punktą, moduliavimo įrenginys, apibūdintas apibrėžties 1 punkte, dar pasižymi tuo, kad turi DSV kontrolės priemonę (pavyzdžiui, DSV bitų detekcijos/įterpimo kontrolės įrenginį 2, parodytą fig.1), skirtą kontroliuoti įvesties duomenų DSV (skaitmeninės sumos reikšmės) ir tiekti šias DSV reikšmes į konversijos priemonę.

Pagal apibrėžties 22 punktą, moduliavimo įrenginys, apibūdintas apibrėžties 1 punkte, pasižymi tuo, kad konversijos priemonė turi:

pirmąją kodų detekcijos priemonę (pavyzdžiui, minimalios sekos nuoseklių atvejų ribojimo kodų detekcijos įrenginį 8, pavaizduotą fig.3), skirtą aptikti pirmuosius pakaitos kodus ir apriboti minimalios sekos d nuoseklių atvejų skaičių; ir

antrąją kodų detekcijos priemonę (pavyzdžiui, užtikrinančių maksimalią seką kodų detekcijos priemonę 9, pavaizduotą fig.3), skirtą aptikti antruosius pakaitos kodus ir išlaikyti sekos trukmės apribojimą.

Demoduliavimo įrenginys pagal apibrėžties 25 punktą pasižymi tuo, kad turi konversijos priemones (pavyzdžiui, demoduliavimo įrenginį 2, pavaizduotą fig.5), skirtą konvertuoti įvesties duomenis į kodą pagal konversijos lentelę, kur ši konversijos lentelė (pavyzdžiui, 2 lentelė) įgalina konversijos taisyklę, pagal kurią '1'

skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$, kur d yra minimali seka, o k yra sekos trukmės riba;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos trukmės ribą k .

Demoduliacinio įrenginio, apibūdintas apibrėžties 25 punkte, pagal apibrėžties 26 punktą dar pasižymi tuo, kad turi bitų pašalinimo priemonę (pavyzdžiui, DSV kontrolės bitų pašalinimo įrenginį 15, pavaizduotą fig.5), skirtą pašalinti perteklinius bitus, įterptus iš anksto nustatytais intervalais į šį kodą.

Toliau aprašyti tinkamiausi išradimo įgyvendinimo variantai. Paaiškinimo supaprastinimui duomenų bitų '0' ir '1' masyvas prieš konversiją, t.y. prieš prieškonversinę duomenų eilutę, pateikiama kaip bitų vora skliaustuose $\{()$, pavyzdžiui, $\{000011\}$. Iš kitos pusės, kodo, gautojo konversijos metu, bitų '0' ir '1' masyvas, t.y. pokonversinės eilutės žodžių eilutė, pateikiama kaip bitų, apibrėžtų pora simbolių $\{, \}$, vora, pavyzdžiui, $\{000100100\}$. Žemiau pateiktosios 2 ir 3 lentelės yra konversijos lentelės, skirtos konvertuoti duomenis į kodą pagal šį išradimą, pavyzdžiai.

2 LENTELĖ

17PP.RML.32

Duomenys	Kodas
11	*0*
10	001
01	010
0011	010 100
0010	010 000
0001	000 100

000011	000 100 100
000010	000 100 000
000001	010 100 100
000000	010 100 000

"110111	001 000 000 (sekantis 010)
00001000	000 100 100 100
00000000	010 100 100 100

jei xx1, tuomet *0*=000

xx0, tuomet *0*=101

sinchronizavimas & baigtis

#01 000 000 001 (12 kanalo bitų)

arba

#01 001 000 000 001 000 000 001 (24 kanalo bitai)

#=0: be baigties

#=1: su baigtimi

Baigties lentelė

00	000
0000	010 100

"110111	001 000 000 (sekantis 010)
---------	----------------------------

Kuomet sekantys kanalo bitai yra '010', konvertuoti '11 01 11' į '001 000 000', baigus naudotis pagrindine lentele ir baigties lentele.

Kaip parodyta 2 lentelėje, konversijos lentelė pateikia kodus, gautus konversijos metu, turinčius pagrindinius kodus, pakaitos kodus ir baigties kodus. Konversija negali būti atlikta be pagrindinio kodo. Konversijos lentelėje pagrindiniai kodai yra kodai, gautieji, konvertavus duomenų eilutes (11) į (000000). Konversija gali būti atlikta net ir tuo atveju, jei nėra pakaitos kodo. Tačiau, jei yra pakaitos kodas, konversija gali būti atlikta efektyviau. Konversijos lentelėje pakaitos kodai yra

kodai, gautieji duomenų eilučių (110111), (00001000) ir (00000000) konversijos metu. Baigties kodas yra naudojamas nutraukti kodą, gautąjį konversijos bet kurioje arbitražinėje pozicijoje metu. Baigties kodai lentelėje yra kodai, gautieji duomenų eilučių (00) ir (0000) konversijos metu. Be to, konversijos lentelė taip pat nusako sinchronizavimo signalus.

2 lentelėje minimali seka d yra 1, kai, tuo tarpu, maksimali seka k yra 7. Vienas iš pagrindinių kodų elementų turi neapibrėžtą kodą, t.y. kodą, pažymėtą žvaigždute '*'. Bitas, atvaizduotas neapibrėžto kodo simboliu '*', gali būti nustatytas kaip arba "0", arba "1", siekiant, kad minimalios sekos d ir maksimalios sekos k reikšmės nepriklausytų nuo betarpiškai ankstesnės ar vėlesnės kodinių žodžių eilutės. Konkrečiau, jei konvertuojama 2 bitų duomenų eilutė yra (11), kodas, gautasis konversijos metu, gali būti "000" ar "101", priklausomai nuo betarpiškai ankstesnės kodinių žodžių eilutės. Dar konkrečiau, jei betarpiškai ankstesnės kodinių žodžių eilutės vienas kanalo bitas yra "1", 2 bitų eilutė (11) konvertuojama į kodą "000", siekiant turėti minimalią seką d . Jei betarpiškai ankstesnės kodinių žodžių eilutės vienas kanalo bitas yra "0", 2 bitų eilutė (11) konvertuojama į kodą "101", siekiant turėti maksimalią seką d .

Pagrindiniai kodai, pavaizduoti 2 lentelės konversijos lentelėje, turi kintamo ilgio struktūrą. Pagrindinių kodų, kurių ribinis ilgis i lygus 1, yra 3, reikšmė, kuri yra mažesnė, nei reikalingas skaičius 4 ($=2^m=2^2$). Šie trys pagrindiniai kodai yra "*0*", "001" ir "010". Štai kodėl, konvertuojant duomenų eilutę, susiduriama su duomenų eilute, kuri negali būti konvertuota, išlaikant tik ribotą ilgį i , lygų 1. Todėl, norint konvertuoti visas duomenų eilutes, būtina remtis pagrindiniais kodais, kurių ribinis ilgis i 2 lentelėje siekia iki 3. T.y., pagrindiniai kodai, kurių ribinis ilgis i siekia iki 3, įtraukti į 2 lentelę tam, kad 2 lentelė galėtų funkcionuoti kaip pakankama konversijos lentelė.

Be to, 2 lentelės konversijos lentelė taip pat turi pakaitos kodus, skirtus apriboti minimalios sekos d nuoseklius atvejus. Jei duomenų eilutė yra (110111), o kodinių žodžių eilutė, einanti po kodo, gautojo konvertavus duomenų eilutę, yra "010", duomenų eilutė konvertuojama į kodinį žodį "010 000 000". Jei kodinių žodžių eilutė, einanti po kodo, gautojo konvertavus duomenų eilutę, yra kitokia, nei "010", tuomet duomenų eilutė (110111) konvertuojama į 2 bitų blokus. Jei detaliau,

2 bitų voros (11), (01) ir (11) duomenų eilutėje yra konvertuojamos į kodinių žodžių “*0*010” ir *0*” eilutę. To rezultate minimalios sekos d nuoseklių atvejų kodinių žodžių eilutėje, gautoje konvertavus duomenų eilutę, skaičius gali būti fiksuotas, taip apribojant pakartotinių minimalių sekų skaičių iki maksimalaus skaičiaus 6.

Be to, 2 lentelės konversijos lentelė įgalina konversijos taisyklę, pagal kurią “1” skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus 1,111 skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui. T.y., jei elemento “1” skaičius duomenų eilutėje yra lyginis, elemento “1” skaičius kodinių žodžių eilutėje taip pat yra lyginis, ir jei elemento “1” skaičius duomenų eilutėje yra nelyginis, tuomet elemento “1” skaičius kodinių žodžių eilutėje taip pat yra nelyginis. Pavyzdžiui, duomenų eilutė (000001) yra konvertuota į kodinių žodžių eilutę “010 100 000”. Šiuo atveju, “1” skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis yra 1, kuris yra lygus “1” skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus duomenų eilutę, dalybos iš 2 likučiui. T.y., “1” skaičius tiek duomenų eilutėje, tiek kodinių žodžių eilutėje yra nelyginis. Pagal kitą pavyzdį duomenų eilutė (000000) yra konvertuota į kodinių žodžių eilutę “010 100 100”. Šiuo atveju, “1” skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis yra 0, kuris yra lygus “1” skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus duomenų eilutę, dalybos iš 2 likučiui. T.y., “1” skaičius tiek duomenų eilutėje, tiek kodinių žodžių eilutėje yra lyginis.

Be to, maksimalus ribinis ilgis r 2 lentelės konversijos lentelėje yra 4. Kodai lentelėje, turintys ribotą ilgį i lygų 4, yra pakaitos kodai, realizuojantys maksimalios sekos k reikšmę 7. Toks pakaitos kodas yra maksimalios sekos užtikrinimo kodas. T.y., duomenų eilutė (0000100) yra konvertuojama į kodinių žodžių eilutę “000 100 100 100”, kai, tuo tarpu, duomenų eilutė (00000000) yra konvertuojama į kodinių žodžių eilutę “010 100 100 100”. Reikia pažymėti, kad šiuo atveju minimalios sekos d reikšmė taip pat yra 1.

Jei 2 lentelės konversijos lentelė neturi pakaitos kodų, kurių ribinis ilgis i yra lygus 4, maksimalus ribinis ilgis r lentelėje yra 3, tuo būdu generuojamas kodas, kurio maksimali seka k yra lygi 8. Kadangi lentelėje yra pagrindiniai kodai, kurių

ribinis ilgis i yra lygus 4, gali būti generuotas kodas, kurios maksimali seka k yra lygi 7.

Paprastai, kuo didesnė maksimali seka k , tuo nepatogiau yra generuoti sinchronizavimo signalą ir tuo nestabilesnė yra sistema. Tuo būdu, sumažinus maksimalios sekos k reikšmę nuo 8 iki 7, sistemos charakteristika gali būti pagerinta proporcingai maksimalios sekos k sumažinimui.

T.y., jei 2 lentelės konversijos lentelė turi tik pagrindinius kodus, maksimalus ribinis ilgis r tokiai lentelei yra 3. Šiuo atveju, yra galima generuoti kodą, kurio minimali seka d yra 1, o maksimali seka k yra 8. Be to, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui.

Jei sukurta konversijos lentelė taip pat turi pakaitos kodus, skirtus riboti ne tik pagrindinius kodus, bet ir minimalios sekos d nuoseklius atvejus, maksimalus ribinis ilgis r tokiai lentelei taip pat yra 3. Šiuo atveju, yra įmanoma generuoti kodą, kur minimali seka d yra 1, o maksimali seka k yra 8, kai, tuo tarpu, nuoseklių minimalių sekų d skaičius apribotas viršutine ribine reikšme. Be to, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui.

Jei sukurta konversijos lentelėje taip pat turi pakaitos kodus, skirtus užtikrinti maksimalią seką k lygią 7, o ne tik pakaitos kodus, ribojančius minimalios sekos d ir pagrindinių kodų nuoseklius atvejus, maksimalus ribinis ilgis tokiai lentelei yra 4. Šiuo atveju, yra įmanoma generuoti kodą, kurio minimali seka d yra 1, o maksimali seka k yra 7, kai, tuo tarpu, nuoseklių minimalių sekų d skaičius apribotas viršutine ribine reikšme. Be to, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui.

Paprastai, kuo didesnis maksimalus ribinis ilgis r , tuo prastesnė demoduliacijos klaidos, generuotos bito postūmio metu, t.y. klaidos, kuri yra generuojama dėl to, kad kraštinis bitas yra perstumiamas pirmyn arba atgal per 1 bitą iš jo normalios padėties, sklaidimo charakteristika.

Lyginant 1 lentelę su 2 lentele matyti, kad maksimalus ribinis pirmosios ilgis yra 2, o antrosios – 4. Tuo būdu, 2 lentelės charakteristikos yra prastesnės, nei 1 lentelės. Tačiau modeliavimo rezultatai, kurie bus aprašyti vėliau su nuoroda į 7 lentelę, nurodo, kad 2 lentelės charakteristikos nėra tokios prastos, lyginant su 1 lentele. Pavyzdžiui, kaip parodyta 7 lentelėje, 1 lentelės vidutinis bitų klaidos dydis yra 1,014 bitų, kai, tuo tarpu, 2 lentelės – 1,67, reikšmė, kuri nėra žymiai didesnė už 1 lentelės reikšmę. Galima manyti, kad vidutinių bitų klaidos dydžių skirtumas priklauso nuo fakto, kad 1 lentelės konversijos kodų grupių skaičius yra mažesnis, nei 2 lentelėje, 2.

Beje, tuo atveju, kuomet sinchronizavimo signalas yra įterpiamas į bet kurią kodinių žodžių eilutės (t.y. kanalo bitų voros), generuotos konversijos, atliktos pagal 2 lentelės konversijos lentelę, metu, arbitražinę poziciją, konversijos lentelė sukuria kintamo ilgio konstrukcijos kodą. Taip yra dėl to, kad konversijos lentelė turi baigties lentelę, nusakančią baigties kodus, skirtus baigti kodą, gautą konversijos bet kurioje arbitražinėje pozicijoje metu. Baigties kodas naudojamas tuomet, kuomet tai yra būtina.

Tarkime, pavyzdžiui, kad sinchronizavimo signalas yra įterpiamas į konkrečią kodo, gautojo konversijos metu, poziciją. Šiuo atveju, pirmiausia, į sandūros tarp kodinių žodžių eilutės, einančios betarpiškai prieš konkrečią padėtį, ir kodinių žodžių eilutės, einančios betarpiškai po konkrečios padėties, tašką yra įterpiami sandūros bitai, išlaikant minimalią seką d ir maksimalią seką k , o unikalūs derinys, išreiškiantis sinchronizavimo signalą, yra įterpiamas tarp sandūros bitų. Tarkime, kad turime sinchronizavimo signalo derinį, kuris pertraukia maksimalios sekos k reikšmę 7. Šiuo atveju, minimalaus ilgio sinchronizavimo signalo derinys yra 12 bitų kodinis žodis (t.y. 12 kanalo bitų), pavaizduotas taip:

"#01 000 000 001"

Simbolis "#" sinchronizavimo signalo derinio pradžioje yra sandūros bitas, kuris gali būti '0' arba '1', kas bus paaiškinta vėliau. Antrasis kanalo bitas, einantis po "#" bito, yra "0", skirtas išlaikyti minimalią seką d . Trečiasis kanalo bitas ir kiti bitai yra reikšmių, kurios sudaro unikalų 9T derinį, kodo derinį, neapibūdintą 2 lentelėje, rinkinys, skirtas sudaryti maksimalią seką k , lygią 6. Kaip matyti aukščiau pateiktame derinyje, tarp antrojo ir paskutiniojo kanalo bitų yra 8 nuoseklūs "0"

bitai. Reikėtų pažymėti, kad, nors paskutinis kanalo bitas sinchronizavimo signalo derinyje yra nustatytas "1", naudojant 2 lentelės konversijos lentelę galima išlaikyti minimalią seką d.

Toliau yra apibūdinti baigties lentelė ir sandūros bitas "#" sinchronizavimo signalo bitų derinyje. Kaip parodyta 2 lentelėje, baigties lentelė yra sudaryta taip:

00	000	
0000	010	100

Baigties lentelė yra reikalinga pagrindiniams kodams, turintiems ribotą ilgį i, kuri sudaro skaičių porų, susidedančių iš duomenų eilutės ir kodinių žodžių eilutės, mažesnį, nei reikalingas skaičius $4 (=2^m=2^2)$.

Pavyzdžiui, kaip matyti 2 lentelėje, tuo atveju, kuomet ribinis ilgis $i=1$, o skaičius porų, kiekviena iš kurių susideda iš duomenų eilutės ir kodinių žodžių eilutės, yra 3, - baigties lentelė yra reikalinga. Kuomet ribinis ilgis $i=2$, o skaičius porų, kiekviena iš kurių susideda iš duomenų eilutės ir kodinių žodžių eilutės, taip pat yra 3, - baigties lentelė yra reikalinga. Kuomet ribinis ilgis $i=3$, skaičius porų, kiekviena iš kurių susideda iš duomenų eilutės ir kodinių žodžių eilutės, yra 5, įskaitant porą, turinčią pakaitos kodą. Kiekviena iš likusių 4 porų turi pagrindinį kodą. Kadangi turime reikalingą skaičių 4, baigties lentelė nereikalinga. Kuomet ribinis ilgis $i=4$, o visos kodinių žodžių eilutės yra pakaitos kodai, nėra būtina atsižvelgti į baigties kodą. Tuo būdu, baigties lentelė yra reikalinga, esant ribiniam ilgiui $i=1$, kur baigties kodas yra naudojamas duomenų eilutėje (00). Pagal tą patį požymį baigties lentelė yra reikalinga, esant ribiniam ilgiui $i=2$, baigties kodas yra naudojamas duomenų eilutėje (0000). Remiantis baigties lentele, duomenų eilutės (00) ir (0000) yra konvertuojamos į kodinių žodžių eilutes "000" ir "010100", atitinkamai. Todėl sinchronizavimo signalo įterpimo metu yra įmanoma išvengti situacijos, kuomet duomenys, einantys prieš sinchronizavimo signalo derinį, negali būti daugiau konvertuojami. T.y., konversijos sistema gali išvengti situacijos, kurioje daugiau nei įmanoma leisti kodui, einančiam betarpiškai prieš sinchronizavimo signalą, tarnauti baigtimi.

Sinchronizavimo signalo derinio bitas "#" yra naudojamas atskirti tą atvejį, kuomet yra naudojama baigties lentelė, nuo to atvejo, kuomet baigties lentelė nenaudojama. Jei detaliau, pirmasis sinchronizavimo signalo kanalo bitas "#"

nustatomas '1' pažymėti, kad baigties kodas yra naudojamas, arba '0' pažymėti, kad baigties kodas nenaudojamas. Tokiu būdu yra įmanoma tiksliai nustatyti, naudojama ar nenaudojama baigties lentelė, t.y. naudojamas ar nenaudojamas baigties kodas.

Kaip aprašyta aukščiau, trumpiausias sinchronizavimo signalo derinys yra 12 bitų žodis (t.y. 12 kanalo bitų). Kadangi yra priimtinas sinchronizavimo signalo derinys, kuris pertraukia maksimalios sekos k reikšmę 7, bet suteikia maksimalią seką $k=8$ (9T), gali būti sukonstruotas bet koks kitas sinchronizavimo signalo derinys, sudarantis mažiausiai 12 bitų kodinį žodį. Tuo atveju, jei yra formuojamas 15 bitų kodinis žodis, gali būti sukurti, pavyzdžiui, tokie 2 sinchronizavimo signalai:

"#01 000 000 001 010"

"#01 000 000 001 001"

Tuo atveju, jei yra formuojamas 21 bitų kodinis žodis, gali būti sukurtas toks sinchronizavimo signalas:

"#01 000 000 001 000 000 001"

Šis 21 bitų sinchronizavimo signalas turi 2 nuoseklius derinius, kiekvienas iš kurių sudaro maksimalią seką $k=8$ (9T). Tikimybė, kad toks sinchronizavimo signalas bus aptiktas, yra didesnė. Tuomet, jei yra formuojamas 24 bitų kodinis žodis, gali būti sukurtas toks sinchronizavimo signalas:

"#01 001 000 000 001 000 000 001"

Šio sinchronizavimo signalo derinys, turintis "3T-9T-9T" formą, sumažina tikimybę, kad didelė seka (T) atsiras prieš ir/arba po dviejų nuoseklių derinių, kiekviena iš kurių sudaro maksimalią seką $k=8$ (9T), o taip pat padidina detekcijos gebą. Yra įmanoma parinkti detekcijos gebą, kurią sinchronizavimo signalas turėtų suteikti pagal sistemos reikalavimus.

3 lentelė yra kita tipinė konversijos lentelė, atitinkanti šį išradimą.

3 LENTELE

17PP.RML.52

i=1 Pagrindinė lentelė:

Duomenys	Kodas
00	101
01	100

10	001
----	-----

11	000
----	-----

i=2 Pakaitos lentelė A (riboja d iki 1)

0000	100 010
------	---------

0001	101 010
------	---------

1000	000 010
------	---------

i=3 Pakaitos lentelė B (riboja k iki 8)

111111	000 010 010
--------	-------------

111110	001 010 010
--------	-------------

011110	101 010 010
--------	-------------

011111	100 010 010
--------	-------------

i=4 Pakaitos lentelė C (riboja RMTR iki 6)

	00010001	100 010 010 010
--	----------	-----------------

kanalas --0	10010001	100 000 010 010
-------------	----------	-----------------

kanalas --1	10010001	000 010 010 010
-------------	----------	-----------------

r=4 Pakaitos lentelė D (riboja k iki 7)

kanalas 010	11100000	000 001 010 010
-------------	----------	-----------------

kanalas 010	11100010	100 001 010 010
-------------	----------	-----------------

kanalas 010	11100001	001 010 010 010
-------------	----------	-----------------

kanalas 010	11100011	101 010 010 010
-------------	----------	-----------------

Sinchronizacija

duomenys: x1 -----0x

kanalas: xx0 100 000 000 10x (12 kanalo bitų)

duomenys: x1 -----0x

kanalas: xx0 100 000 000 100 000 000 10x (24 kanalo bitai)

Baigtis:

prideda pradžioje duomenų bitus '01' arba '11',

ir gale '00' arba '01'

3 lentelės konversijos lentelės konstrukcija yra tokia, kad, kuomet minimali seka $d=1$, maksimali seka $k=7$, ribinis ilgis $i=1,4(=2^m=2^2)$, yra numatyti pagrindiniai kodai. T.y., kuomet ribinis ilgis $i=1$, į pagrindinę lentelę įvedami 4 pagrindiniai kodai. Kuomet ribinis ilgis $i=2$ ar daugiau, yra numatytos pakaitos kodų

lentelės, skirtos apriboti tokius parametrus, kaip minimali seka d ir maksimali seka k . Konkrečiau, lentelė A, kuomet ribinis ilgis $i=2$, nustato pakaitos kodus, skirtus apriboti minimalią seką d iki 1. Lentelė B, kuomet ribinis ilgis $i=3$, nustato pakaitos kodus, skirtus apriboti maksimalią seką k viršutine riba 8. Lentelė C, kuomet ribinis ilginis $i=4$, nustato pakaitos kodus, skirtus apriboti minimalios sekos d , turinčios reikšmę 1, nuoseklius atvejus. Lentelė D, kuomet ribinis ilgis $i=4$, nustato pakaitos kodus, skirtus apriboti maksimalią seką k viršutine reikšme 7. Tuo būdu, 3 lentelės konversijos lentelėje maksimali ribinė riba $r=4$.

Kaip aprašyta aukščiau, 3 lentelės konversijos lentelė turi pakaitos kodus, skirtus riboti minimalios sekos d nuoseklius atvejus. Pavyzdžiui, duomenų eilutė (0001001) yra konvertuojama į kodinių žodžių eilutę "100 010 010 010". Duomenų eilutei (10010001) yra tuoj pat paskiriamas kodinis žodinis, siekiant nustatyti, ar betarpiškai einantis prieš kanalo bitas yra "0" ar "1". Jei betarpiškai prieš einantis kanalo bitas yra "011", duomenų eilutė yra konvertuojama į kodinių žodžių eilutę "100 000 010 010". Iš kitos pusės, jei betarpiškai prieš einantis kanalo bitas yra "1", duomenų eilutė yra konvertuojama į kodinių žodžių eilutę "000 010 010 010". Šiuo atveju, kodinių žodžių eilutėje, gautoje konvertavus duomenis, nuosekliai pasikartojančių minimalių sekų skaičius yra apribotas iki maksimalaus skaičiaus 6.

Be to, 3 lentelės konversijos lentelė įgalina taisyklę, pagal kurią "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui. T.y., "1" skaičius elemento duomenų eilutėje yra lyginis, "1" skaičius elemento kodinių žodžių eilutėje taip pat yra lyginis, ir, iš kitos pusės, jei "1" skaičius elemento duomenų eilutėje yra nelyginis, "1" skaičius elemento kodinių žodžių eilutėje taip pat yra nelyginis. Pavyzdžiui, duomenų eilutė (1000) yra konvertuojama į kodinių žodžių eilutę "000 010". Šiuo atveju, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis yra 1, kas yra lygu "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus duomenų eilutę, dalybos iš 2 likučiui. T.y., "1" skaičius tiek duomenų eilutėje, tiek kodinių žodžių eilutėje, yra nelyginis. Arba, pavyzdžiui, duomenų eilutė (111111) yra konvertuojama į kodinių žodžių eilutę "000 010 010". Šiuo atveju, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis yra 0, kas yra lygu "1" skaičiaus

elemento duomenų eilutėje, gautoje konvertavus duomenų eilutę, dalybos iš 2 likučiui. T.y., "1" skaičius tiek duomenų eilutėje, tiek kodinių žodžių eilutėje yra lyginis.

Be to, 3 lentelės konversijos lentelės kodai, kurių ribinis ilgis i yra 4, kas yra lygu maksimaliam ribiniam ilgiui r , yra pakaitos kodai, skirti įdiegti maksimalios sekos k reikšmę 7. Konversijos, naudojant tokius pakaitos kodus, metu yra nurodoma einanti betarpiškai prieš kodinių žodžių eilutę. Konkrečiau, jei einanti betarpiškai prieš kodinių žodžių eilutę yra "010", konversija vykdoma. Pavyzdžiui, jei duomenų eilutė yra (11100000), o einanti betarpiškai prieš kodinių žodžių eilutę yra "010", vykdoma konversija, kurios rezultatas yra kodinių žodžių eilutė "000 001 010 010". Arba, pavyzdžiui, jei duomenų eilutė yra (11100010), o einanti betarpiškai prieš kodinių žodžių eilutę yra "010", duomenų eilutė yra konvertuojama į kodinių žodžių eilutę "100 001 010 010".

Jei lentelė C, nusakanti pakaitos kodus, skirtus riboti minimalios sekos d nuoseklius atvejus, yra įtraukta į 3 lentelės konversijos lentelės konfigūraciją kartu su pagrindine lentele ir lentelėmis A ir B, maksimalus ribinis ilgis r yra 4 ir yra įmanoma generuoti kodą, kurio minimali seka d yra 1, maksimali seka k yra 8, ir kuris turi ribotą minimalių sekų d nuoseklių atvejų skaičių. Be to, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui. Reikėtų pažymėti, kad, šiuo atveju, ne visuomet yra būtina nurodyti einančią betarpiškai prieš kodinių žodžių eilutę, kaip tai yra naudojant 3 lentelės lentelę C.

Jei lentelė D, nusakanti pakaitos kodus, skirtus užtikrinti maksimalią seką k , lygią 8, yra įtraukta į 3 lentelės konfigūraciją kartu su pagrindine lentele ir lentelėmis A ir B ir lentele C, nusakančia pakaitos kodus, maksimalus ribinis ilgis r yra 4 ir yra įmanoma generuoti kodą, kurio minimali seka d yra 1, maksimali seka k yra 7, ir kuris turi ribotą minimalių sekų d nuoseklių atvejų skaičių. Be to, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui.

2 lentelės baigties lentelė nereikalinga, įterpiant sinchronizavimo signalą į konkrečią kodinių žodžių eilutės (ar kanalo bitų voros), gautos konvertavimo, naudojant 3 lentelės konversijos lentelę, metu, poziciją. Taip yra todėl, kad pagal 3 lentelės konversijos lentelę kodas gali būti pertrauktas, esant tam tikram ribiniam ilgiui i , lygiam 1.

Siekiant įterpti sinchronizavimo signalo rezultato efektyviausiame kode derinį, derinys yra apibūdinamas taip: 3 bitų kodinis žodis, einantis betarpiškai prieš unikalų derinį (ar kodinio žodžio eilutę), veikiantį kaip sinchronizavimo signalas, ir 3 bitų kodinis žodis, einantis betarpiškai po unikalaus derinio, yra kodiniai žodžiai, gautieji konversijos pagal 3 lentelę metu. Kiekvienas 3 kodinių žodžių bitas, einantis betarpiškai prieš ir po unikalaus derinio, turi formatą, kuriame duomenų bitai ir sandūros bitai yra sumaišyti, kas paaiškinta žemiau.

Pirmiausia, yra apibrėžiami einančio betarpiškai prieš kodinio žodžio 3 bitai. Ištiriamas duomenų žodžio m bitų elementas prieš konversiją, kur $m=2$. 2 bitų elemento pirmasis bitas duomenų žodžio pradžioje prieš konversiją yra informacijos bitas, kai, tuo tarpu, antrojo bito reikšmė yra '1' ir yra skirta nurodyti sinchronizavimo signalą. 2 bitų elementas duomenų žodžio pradžioje yra konvertuojamas į kodinį žodį (kanalo bitus) pagal 3 lentelę. Tiksliau, duomenų žodžio ($x1$) m bitų (2 bitai) yra konvertuojami į kodinio žodžio "xx0" n bitų (3 bitus).

Tuomet apibrėžiami einančio betarpiškai po kodinio žodžio 3 bitai. Pagal tą patį požymį ištiriamas duomenų žodžio m bitų elementas prieš konversiją, kur $m=2$. Tačiau šiuo atveju 2 bitų elemento pirmasis bitas, esantis duomenų žodžio pradžioje, prieš konversiją turi reikšmę '0', kuri skirta nurodyti sinchronizavimo signalą, kai, tuo tarpu, antrasis bitas yra informacijos bitas. 2 bitų elementas duomenų žodžio pradžioje yra konvertuojamas į kodinį žodį (kanalo bitus) pagal 3 lentelę. Tiksliau, duomenų žodžio (0x) m bitų (2 bitai) konvertuojami į kodinio žodžio "10x" n bitų (3 bitus).

Kuomet sinchronizavimo signalo unikalus derinys yra nustatytas kaip derinys, pertraukiantis maksimalios sekos k reikšmę 7, sinchronizavimo signalo derinys, kuris gali būti realizuotas kuo trumpesnis, turi unikalaus derinio kodinį žodį, susidedantį mažiausiai iš 12 bitų (12 kanalo bitų), kuris atrodo taip:

"xx0 100 000 000 10x"

kur "x" reikšmė priklauso nuo konversijos lentelės. Aukščiau parašytas 15 bitų žodis turi 3 "x" bitus. 2 "x" bitai yra kodinio žodžio pradžioje, kai, tuo tarpu, 1 "x" bitas yra jo gale. 3 "x" bitai reiškia 2 bitų kodinio žodžio elementą prieš konversiją. Likę 12 kodinio žodžio kanalo bitų yra perteklinė dalis, iš tikrųjų reiškianti sinchronizavimo signalo derinį. Trečiasis kodinio žodžio kanalo bitas yra nustatytas "0", siekiant išlaikyti minimalią seką d. Kalbant apie likusius, pradedant ketvirtuoju kanalo bitu, 9T yra nustatytas kaip sinchronizavimo signalo derinys, užtikrinantis maksimalią seką k, lygią 8. Tiksliau, aštuoni "0" yra išdėstyti nuosekliai tarp "1" ir "1".

Kaip aprašyta aukščiau, minimalaus ilgio sinchronizavimo signalo unikalus derinys yra 12 bitų kodinis žodis (t.y. 12 kanalo bitų). Kadangi sinchronizavimo signalo derinys, kuris pertraukia maksimalios sekos k reikšmę 7, bet užtikrina maksimalią seką, lygią 8 (9T), yra priimtinas, gali būti sukurtas bet koks kitas sinchronizavimo signalo derinys, sudarantis mažiausiai 12 bitų unikalaus derinio kodinį žodį. Tuo atveju, jei, pavyzdžiui, formuojamas 15 bitų unikalaus derinio kodinis žodis, gali būti sukurtas toks sinchronizavimo signalas:

"xx0 100 000 000 100 10x"

Formuojant 21 bitų kodinį žodį, gali būti sukurtas toks sinchronizavimo signalas:

"xxx0 100 000 000 100 000 000 10x"

Aukščiau parašytasis 21 bitų sinchronizavimo signalas turi 2 nuoseklius derinius, kiekvienas iš kurių suteikia maksimalią seką k, lygią 8 (9T). Pagal sinchronizavimo signalą gali būti padidinta detekcijos galia. Yra įmanoma parinkti detekcijos galia, kurią sinchronizavimo signalas turėtų suteikti pagal sistemos reikalavimus.

Kaip ir įprastame būde, konvertavus duomenų eilutę, naudojantis 2 ar 3 lentelės konversijos lentele, gali būti įvykdyta DSV kontrolė, pridant iš anksto nustatytais intervalais DSV kontrolės bitus į kanalo bitų, gautų konversijos metu, vorą. Tačiau, panaudojant ryšį tarp duomenų eilutės ir kodinių žodžių eilutės, gautos konversijos pagal 2 ir 3 lenteles metu, DSV kontrolė gali būti įvykdyta netgi efektyviau.

Tiksliau, konversijos taisyklė vykdoma taip, kad "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1"

skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui. Tuo būdu, DSV kontrolės bitų, turinčių reikšmę "1", ir bitų, turinčių reikšmę "0", įterpimas į kanalo bitų vorą žymėti, atitinkamai, inversiją ir neinversiją yra ekvivalentiškas DSV kontrolės bitų, turinčių reikšmę "1", ir bitų, turinčių reikšmę "0", įterpimas į duomenų bitų vorą žymėti, atitinkamai, inversiją ir neinversiją.

Išnagrinėjime, pavyzdžiui, atvejį, kuomet kuriame 3 duomenų bitai (001) yra konvertuojami pagal 2 lentelę. Įterpkime DSV kontrolės bitą x , einančiam po 3 bitų, į duomenis. Šiuo atveju, duomenys, turintys DSV kontrolės bitą, tampa (000- x), kur x yra 1 bito kontrolės bitas, turintis reikšmę '0' arba '1'. Kai $x=0$, duomenys, turintys DSV kontrolės bitą, konvertuojami pagal 2 lentelę taip:

Duomenų eilutė	Kodinio žodžio eilutė
0010	010 000

Iš kitos pusės, kai $x=1$, duomenys, turintys DSV kontrolės bitą, konvertuojami pagal 2 lentelę taip:

Duomenų eilutė	Kodinio žodžio eilutė
0011	010 100

Taikant kodinių žodžių eilutei, gautajai konversijos pagal 2 lentelę metu, lygmens kodavimą NRZI moduliavimu, gaunama tokio lygmens koduota kodinė eilutė:

Duomenų eilutė	Kodinio žodžio eilutė	Koduota kodinė eilutė
0010	010 000	011111
0011	010 100	011000

Kaip parodyta aukščiau lentelėje, pirmosios koduotos kodinės eilutės paskutiniai 3 bitai yra invertuoti antrosios koduotos kodinės eilutės paskutiniai 3 bitai. Rezultatai parodo, kad, parenkant DSV kontrolės bito reikšmę (1) ar (0), DSV kontrolė gali būti atlikta pačioje duomenų eilutėje.

Išnagrinėjimo portoklumą, įterptą DSV kontrolę. DSV kontrolės vykdymas, įterpiant 1 DSV kontrolės bitą į duomenų eilutę, atitinka DSV kontrolės vykdymą,

įterpiant 1,5 DSV kontrolės bitų į kanalo bitų vorą, kur reikšmė 1,5 yra 2 ir 3 lentelių konversijos greičio $m/n=2/3$ atvirkštinis dydis. Siekiant vykdyti panašią kaip ir 1 lentelėje DSV kontrolę RLL(1-7) lentelei, reikia taikyti kontrolę kanalo bitų voroje. Šiuo atveju, siekiant išlaikyti minimalią seką d, reikalingi mažiausiai 2 kanalo bitai, sudarant aukštą, lyginant su DSV kontrole, pritaikytą 2 ir 3 lentelių duomenų eilutei, santykinį perteklumą. Kitaip tariant, vykdant DSV kontrolę duomenų eilutėje, DSV kontrolės efektyvumas šioje sistemoje gali būti pagerintas.

Toliau aprašyti išradimo realizavimo variantai pagal brėžinius, kuriuose:

Fig.1 yra diagrama, vaizduojanti tipinę išradimo objekto, naudojančio moduliavimo įrenginį, konfigūraciją;

Fig.2 yra pavyzdinė diagrama, naudojama aprašyti procesą, atliekamą DSV kontrolės bito nustatymo/įterpimo įrenginiu 2, naudojamu moduliavimo įrenginyje, pavaizduotame fig.1;

Fig.3 yra diagrama, vaizduojanti tipinę moduliavimo įtaiso 3, naudojamo moduliavimo įrenginyje, pavaizduotame fig.1, konfigūraciją;

Fig.4 yra diagrama, vaizduojanti procesą, atliekamą fig.3 moduliavimo įtaisu 3;

Fig.5 yra diagrama, vaizduojanti tipinę išradimo objekto, naudojančio demoduliavimo įrenginį, konfigūraciją;

Fig.6 yra diagrama, vaizduojanti tipinę demoduliavimo įtaiso 14, naudojamo demoduliavimo įrenginyje, pavaizduotame fig.1, konfigūraciją;

Fig.7 yra pavyzdinė diagrama, naudojama aprašyti procesą, atliekamą demoduliavimo įrenginiu 14, pavaizduotu fig.6;

Fig.8 yra schema, naudojama paaiškinti operacijas, atliekamas DSV kontrolės bito pašalinimo įrenginiu 15, naudojamu demoduliavimo įrenginyje, pavaizduotame fig.5;

Fig.9 yra diagrama, vaizduojanti kitą išradimo įgyvendinimo konfigūraciją, naudojant šio išradimo moduliavimo įrenginį;

Fig.10 yra diagrama, diagrama, vaizduojanti dar kitą išradimo įgyvendinimo konfigūraciją, naudojant šio išradimo demoduliavimo įrenginį; ir

Fig.11 yra diagrama, vaizduojanti įrašo kodą su įterptais į jį sinchronizavimo signalais ir DSV kontrolės bitais.

Šio išradimo realizavimas, naudojant jame pateiktą moduliavimo įrenginį, yra paaiškintas su nuoroda į fig.1. Šiame įgyvendinimo variante duomenų eilutė yra konvertuojama į kintamo ilgio kodą $(d, k; m, n; r) = (1, 7; 2, 3; 4)$, naudojant 2 lentelę.

Kaip parodyta fig.1, moduliavimo įrenginys 1 turi DSV kontrolės bito nustatymo/įterpimo įrenginį 2, skirtą nustatyti, ar DSV kontrolės bito reikšmė yra '1' ar '0', ir įterpti DSV kontrolės bitus bet kokias arbitražiniais intervalais į pateiktą duomenų eilutę, moduliavimo įrenginį 3, skirtą moduluoti duomenų eilutę įterptais į ją DSV bitais, ir NRZI kodavimo įrenginį 4, skirtą konvertuoti moduliavimo įrenginio 3 išvestį į įrašymo bangos vorą. Be to, moduliavimo įrenginys 1 dar turi sinchronizavimo įrenginį 5, skirtą generuoti sinchronizavimo signalus ir tiekti juos įvairiems komponentams.

Fig.2 yra paaiškinamoji schema, naudojama aprašyti DSV kontrolės bitų nustatymo/įterpimo įrenginiu 2 vykdomą procesą. Kaip parodyta brėžinyje, yra nustatomos DSV kontrolės bitų reikšmės ir DSV kontrolės bitai yra įterpiami į duomenų eilutę bet kokiais arbitražiniais intervalais. Siekiant įterpti DSV kontrolės bitus, pavyzdžiui, tarp gaunamos duomenų eilutės duomenų blokų DATA1 ir DATA2, DSV kontrolės bitų nustatymo/įterpimo įrenginys 2 apskaičiuoja kumuliacinę DSV duomenims iki DATA1. Bendra DSV apskaičiuojama palaipsniui:

konvertuojant DATA1 į kanalo bitų vorą;
atliekant bitų voros NRZI moduliaciją;
priskiriant reikšmę +1 NRZI moduliacijos rezultato aukštam (H) lygmeniui (1) ir reikšmę -1 NRZI moduliacijos rezultato žemam (L) lygmeniui (0); ir
sudedant reikšmes, priskirtas NRZI moduliacijos rezultato lygmenims.

Pagal tą patį požymį DSV kontrolės bitų nustatymo/įterpimo įrenginys 2 apskaičiuoja bendrą DSV segmentui DATA2, einančiam po DATA1. Tarkime, kad x_1 yra DSV kontrolės bitas, kuris bus įterpiamas tarp duomenų DATA1 ir DATA2 blokų. DSV kontrolės bitų nustatymo/įterpimo įrenginys parenka DSV kontrolės bito x_1 reikšmę taip, kad absoliuti DSV sumos reikšmė duomenų blokui DATA1, x_1 ir DATA2 priartėja prie nulio.

Jei DSV kontrolės bitas x_1 yra (1), segmento DATA2, einančio po DATA1, lygmens kodai yra invertuojami. Iš kitos pusės, jei DSV kontrolės bitas x_1 yra (0),

segmento DATA2, einančio po DATA1, lygmens kodai neinvertuojami. Taip yra todėl, kad kiekviename 2 ir 3 lentelių konversijos lentelių elemente "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui. Tuo būdu, (1) bito įterpimas į duomenų eilutę turi būti palydimas "1" įterpimu į kodinių žodžių eilutę, gautąją konvertavus duomenų eilutę, kuomet inversija yra numanoma.

Nustatius aukščiau aprašytu būdu DSV kontrolės bito x_1 , pavaizduoto fig.2, reikšmę, DSV kontrolės bitas x_2 įterpiamas tarp DATA2 ir DATA3, suteikiant iš anksto apibrėžtą duomenų intervalą tarp x_1 ir x_2 , skirtą atlikti DSV kontrolę tuo pačiu būdu. Šiuo atveju, kumuliacinė DSV yra kumuliacinės DSV duomenų iki DATA1, DSV reikšmės kontrolės bitui x_1 ir DSV duomenų segmentui DATA2 suma.

Kaip aprašyta aukščiau, DSV kontrolės bitai įterpiami į duomenų eilutę iš anksto prieš duomenų eilutės moduliavimą moduliavimo įrenginiu 3, generuojant kanalo bitų vorą.

Fig.3 yra blokinė schema, vaizduojanti tipinę moduliavimo įrenginio 3 konfigūraciją. Kaip parodyta brėžinyje, postūmio registras 6 perstumia jame saugomus duomenis vienu metu per 2 bitus, persiunčiant savo išvestį į ribinio ilgio įvertinimo įrenginį 7, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginį 8, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginį 9 ir visus konversijos įrenginius, pradedant 10-1 ir baigiant 10-4. Postūmio registras 6 perduoda tiek bitų, kiek yra reikalinga apdorojimui kiekviename komponente 7-10.

Ribinio ilgio įvertinimo įrenginys 7 nustato duomenų ribinį ilgį i ir perduoda jį į multiplekserį 11. Kuomet minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 8 aptinka specialų duomenų žodį, kuriam reikia apriboti minimalios sekos d nuoseklius atvejus, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos 8 perduoda jo detekcijos signalą ($i=3$), nurodantį ribotą ilgį i , į riboto ilgio įvertinimo įrenginį 7. Kaip matyti 2 lentelėje, specialus duomenų žodis yra (110111). Pagal tą patį požymį, kuomet sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 9 aptinka specialų duomenų žodį, kuriam reikia užtikrinti maksimalią seką k , sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 9 perduoda jo detekcijos signalą ($i=4$), nurodantį ribotą ilgį i , į ribinio ilgio įvertinimo

įrenginį 7. Kaip matyti 2 lentelėje, specialus duomenų žodis yra (00001000) arba (00000000).

Kuomet minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 8 ar sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 9 aptinka specialų duomenų žodį, ribinio ilgio įvertinimo įrenginys 7 persiunčia specialaus duomenų žodžio ribotą ilgį i į multiplekserį 11. Tuo metu ribinio ilgio įvertinimo įrenginys 7 taip pat gali pats nustatyti kitą ribinio ilgio reikšmę. Tačiau, šiuo atveju, ribinio ilgio įvertinimo įrenginys 7 teikia pirmenybę ribiniam ilgiui, tiekiamam minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginiu 8 ar sekos ilgio ribos užtikrinimo kodo detekcijos įrenginiu 9, o ne aptiktam paties. Kitaip tariant, parenkamas didesnis ribinis ilgis.

Kiekvienas konversijos įrenginys 10-1...10-4 priima sprendimą įrašyti arba neįrašyti į jame esančią konversijos lentelę jam tiekiamų duomenų pagrindinį kodą. Jei randama, kad pagrindinis kodas yra įtrauktas į katalogą, duomenys konvertuojami į kodinį žodį, išreikštą pagrindiniu kodu, ir gautasis konversijos metu kodinis žodis persiunčiamas į multiplekserį 11. Iš kitos pusės, jei duomenų pagrindinis kodas nėra įtrauktas į katalogą konversijos lentelėje, konversijos įrenginiai 10-1...10-4 atmeta duomenis.

Reikia pažymėti, kad, kadangi moduliavimo įrenginys 3 yra skirtas 2 lentelės konversijos lentelei, kiekvienas konversijos įrenginys 10-i yra skirtas konvertuoti duomenis, kurių ribinis ilgis i yra 4. T.y., kiekvienas konversijos įrenginys 10-i yra skirtas konvertuoti duomenis, kurių maksimalus ribinis ilgis r yra 4.

Multiplekseris 11 parenka kodą, gautąjį konversijos, atliktos vienu iš konversijos įrenginių 10-i, metu, atitinkantį ribiniam ilgiui i , kurį tiekia ribinio ilgio įvertinimo įrenginys 7. Po to parinktas kodas išvedamas buferiu 12 kaip seriniai duomenys.

Kiekvieno komponentas sinchronizuojamas signalais, generuotais sinchronizavimo valdymo įrenginiu 5.

Toliau aprašomas išradimo objekto veikimas.

Pirmiausia, postūmio registras 6 tiekia tiek duomenų bitų, kiek jų reikia procesui, tokiam kaip įvertinimo formavimas 2 bitų elementais, į ribinio ilgio įvertinimo įrenginį 7, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos

įrenginį 8, sekos ilgio ribojimo užtikrinimo kodo detekcijos įrenginį 9 ir visus konversijos įrenginius 10-1...10-4.

Turėdamas tokią konversijos lentelę, kaip 2 lentelė, ribinio ilgio įvertinimo įrenginys 7 nustato ribinio ilgio i reikšmę, naudodant konversijos lentelę, ir tiekia reikšmę į multiplekserį 11.

Minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginyje 8 yra įterptas duomenų žodis, kuris turi būti pakeistas pakaitos kodu, skirtu riboti 2 lentelės minimalios sekos d nuoseklius atvejus, su sąlyga, kad sekantis kodinis žodis yra "010". 2 lentelėje duomenų žodis yra (110111). Kuomet duomenys, kur reikia apriboti minimalios sekos d nuoseklius atvejus, aptikti kaip nuorodos į šią konversijos lentelės dalį rezultatas, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 8 išveda detekcijos signalą, nurodantį, kad ribinis ilgis $i=3$, į įvertinimo įrenginį 7.

Iš kitos pusės, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginyje 9 yra įterpti duomenų žodžiai, kurie turi būti pakeisti pakaitos kodais, skirtais užtikrinti 2 lentelės sekos ilgio ribą. 2 lentelėje duomenų žodžiai yra (00001000) ir (00000000). Kuomet duomenys, kur reikia apriboti minimalios sekos d nuoseklius atvejus, aptikti kaip nuorodos į šią konversijos lentelės dalį rezultatas, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 9 išveda detekcijos signalą, nurodantį, kad ribinis ilgis $i=4$, į įvertinimo įrenginį 7.

Gavęs iš minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginio 8 detekcijos signalą, nurodantį, kad ribinis ilgis $i=3$, turint 2 lentelę, ribinio ilgio įvertinimo įrenginys 7 persiunčia reikšmę $i=3$ į multiplekserį 11, ribinio ilgio įvertinimo įrenginys 7, net ir pats nustatęs ribinio ilgio i reikšmę, nepersiunčia jos į multiplekserį 11. Pagal tą patį požymį, gavęs iš sekos ilgio ribos užtikrinimo kodo detekcijos įrenginio 9 detekcijos signalą, nurodantį, kad ribinis ilgis $i=4$, turint 2 lentelę, ribinio ilgio įvertinimo įrenginys 7 persiunčia reikšmę $i=4$ į multiplekserį 11, ribinio ilgio įvertinimo įrenginys 7, net ir pats nustatęs ribinio ilgio i reikšmę, nepersiunčia jos į multiplekserį 11.

Tai reiškia, kad ribinio ilgio įvertinimo įrenginys 7 persiunčia ribinio ilgio i reikšmę, gautą iš minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginio 8 ar sekos ilgio ribos užtikrinimo kodo detekcijos įrenginio 9, į

multiplekserį 11, vietoje to, kad tiektų paties nustatytą reikšmę, jei ribinio ilgio reikšmė i , nustatyta minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginiu 8 ar sekos ilgio ribos užtikrinimo kodo detekcijos įrenginiu 9, skiriasi nuo reikšmės, nustatytos paties. Kitaip tariant, siuntimui į multiplekserį 11 parenkamas didesnis ribinis ilgis.

Fig.4 yra diagrama, vaizduojanti konkrečiais terminais procesą, vykdomą ribinio ilgio įvertinimo įrenginiu 7, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginiu 8 ir sekos ilgio ribos užtikrinimo kodo detekcijos įrenginiu 9.

Kaip aprašyta aukščiau, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginyje 9 2 lentelės duomenų žodžiai (00001000) ir (00000000) įterpiami kaip dalis jos funkcijos, siekiant nustatyti ribinio ilgio i reikšmę. Kuomet 8 bitų informacija, atitinkanti duomenų žodį (00001000) ar (00000000), tiekiama į sekos ilgio ribos užtikrinimo kodo detekcijos įrenginį 9, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 9 išveda detekcijos signalą, nurodantį, kad ribinis ilgis $i=4$, į ribinio ilgio įvertinimo įrenginį 7.

Iš kitos pusės, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginyje 8 2 lentelės duomenų žodis (110111) yra įterpiamas kaip jos funkcija, siekiant nustatyti ribinio ilgio i reikšmę. Kuomet 6 bitų informacija, atitinkanti duomenų žodį (110111), yra tiekiama į minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginį 8, o 3 bitų kodinis žodis, gautasis konversijos metu, einantis po duomenų žodžio, yra "010", minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 8 išveda detekcijos signalą, nurodantį, kad ribinis ilgis $i=3$, į ribinio ilgio įvertinimo įrenginį 7. Reikia pažymėti, kad 3 bitų kodinis žodis "010" yra duomenų eilutės, kurios reikšmė prieš konversiją yra (01), (001) ar (00000), konversijos rezultatas. Kitaip tariant, funkcijos sudėtyje yra duomenų eilutė (110111)+(01/01/00000). Kuomet aptinkama 6 bitų informacija, atitinkanti duomenų žodį (110111), 5 bitų informacija, einanti po 6 bitų informacijos, palyginama su duomenų žodžiu (01), (001) ar (00000), siekiant nustatyti, ar jos atitinka viena kitą. Jei įeinanti informacija yra (11011101), (11011001) ar (11011100000), minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 8 išveda detekcijos signalą, nurodantį, kad ribinis ilgis $i=3$, į ribinio ilgio įvertinimo įrenginį 7.

Ribinio ilgio įvertinimo įrenginyje 7 yra įterptos 2 lentelės konversijos lentelės duomenų eilutės. Jei 6 bitų informacija, atitinkanti duomenų žodį (000011), (000010), (000001) ar (000000), yra tiekama į ribinio ilgio įvertinimo įrenginį 7, pastarasis nustato, kad ribinio ilgio i reikšmė yra 3. Jei 4 bitų informacija, atitinkanti duomenų žodį (0011), (0010) ar (0001), yra tiekama į ribinio ilgio įvertinimo įrenginį 7, pastarasis nustato, kad ribinio ilgio i reikšmė yra 2. Jei 2 bitų informacija, atitinkanti duomenų žodį (11), (10) ar (01), yra tiekama į ribinio ilgio įvertinimo įrenginį 7, pastarasis nustato, kad ribinio ilgio i reikšmė yra 1.

Tarkime, kad tiekama 6 bitų informacija (000010). Šiuo atveju, ribinio ilgio įvertinimo įrenginys 7 nustato, kad ribinio ilgio i reikšmė yra 3. Taip pat tarkime, kad 2 bitai (00) seka po 6 bitų informacijos. Dėl šios priežasties 8 bitų informacija, atitinkanti duomenų žodį (00001000), tiekama į sekos ilgio ribos užtikrinimo kodo detekcijos įrenginį 9 ir priverčia jį išvesti detekcijos signalą, nurodantį, kad ribinis ilgis $i=4$, į ribinio ilgio įvertinimo įrenginį 7. Šiuo atveju, ribinio ilgio įvertinimo įrenginys 7 teikia pirmenybę detekcijos signalui, pernešančiam reikšmę 4 iš sekos ilgio ribos užtikrinimo kodo detekcijos įrenginio 9, prieš paties nustatytą reikšmę 3, nustatant ribinio ilgio i reikšmę, lygią 4.

Kaip aprašyta aukščiau, duomenų, susidedančių iš (1) ir (0) voros, ribinis ilgis gali būti nustatytas pagal 2 lentelės konversijos lentelę, įvertinant tiekiamą net iki 8 bitų dydžio duomenų žodį, atitinkantį maksimalų ribinį ilgį, ir, reikalui esant, 3 bitų kodinį žodį. Alternatyviai gali būti nustatytas duomenų, susidedančių iš (1) ir (0) voros, ribinis ilgis, įvertinant tik tiekiamą iki 11 bitų dydžio duomenų žodį.

Ribinio ilgio įvertinimo įrenginys 7 teikia nustatytą šiuo būdu ribinio ilgio i reikšmę į multiplekserį 11.

Reikia pažymėti, kad ribinio ilgio įvertinimo įrenginys 7 taip pat gali nustatyti ribinio ilgio i reikšmę didėjančia i reikšmių tvarka, pradedant mažiausia reikšme, t.y. $i=1$, $i=2$, $i=3$ ir $i=4$ tvarka, priešingai, nei parodyta fig.4.

Kiekvienas konversijos įrenginys 10-1...10-4 turi konversijos lentelę, atitinkančią priskirto jam ribinio ilgio reikšmę. Jei konkrečiau, konversijos įrenginiai 10-1...10-4 turi konversijos lenteles reikšmėms $i=1$, $i=2$, $i=3$ ir $i=4$, atitinkamai. Jei konversijos taisyklė duomenims, tiekiamiems į bet kurį konversijos įrenginį 10-1...10-4, kataloguojama į konversijos įrenginių lentelę, tiekiamų duomenų $2 \times i$

bitai konvertuojami į $3 \times i$ kodo bitus pagal kataloguotą konversijos taisyklę. Gautasis kodas tuomet tiekiamas į multiplekserį 11.

Multiplekseris 11 parenka kodą, gautą iš konversijos, atliktos vienu iš konversijos įrenginių 10-i, metu, atitinkantį ribinį ilgį i , tiekiamą ribinio ilgio įvertinimo įrenginiu 7. Tuomet parinktas kodas išvedamas buferiu 12 kaip seriniai duomenys.

Kaip parodyta 2 lentelėje, kuomet ribinis ilgis $i=3$, konversijos lentelėje nėra pakaitos kodų duomenų eilutei (110111), kuriai reikia apriboti minimalios sekos d nuoseklius kartotinius atvejus. Tarkime, kad yra tiekiamas tokia duomenų eilutė:

(1101110111011101)

Šiuo atveju, konversija atliekama tokia duomenų žodžių tvarka: (11), (01), (11), (01) ir t.t. Konversijai pasibaigus generuojama tokia kodinių žodžių eilutė (kanalo bitų vora):

"101 010 101 010 010 010 101 010".

Tuomet generuota kodinių žodžių eilutė paveikiama NRZI moduliacija, siekiant atlikti lygmens kodavimą. Kadangi logikos inversija vyksta sinchronizuojant '1' signale, aukščiau minėta kodinių žodžių eilutė yra konvertuojama į tokia kodinių žodžių eilutę:

'110 011 001 100 110 011',

kur minimalūs inversijos intervalai $2T$ tęsiasi per visą eilutę. Įrašyta ar atkurta dideliu linijiniu tankiu tokia kodinė eilutė tampa deriniu, kuris lengvai iššaukia klaidą įrašymo ar atkūrimo metu.

Tarkime, kad 2 lentelės konversijos lentelėje yra numatytas pakaitos kodas duomenų eilutei (110111), kuriai reikia apriboti nuoseklius kartotinius minimalios sekos d atvejus. Tiekime tokia duomenų eilutę:

(1101110111011101).

Šiuo atveju, pirmasis duomenų žodis (11011101) duomenų eilutėje taip pat turi duomenų žodį (110111), po kurio seka duomenų žodis (01), kuris bus konvertuotas į kodinių žodžių eilutę "010". Tuo būdu, pirmasis duomenų žodis yra konvertuojamas į tokia kodinių žodžių eilutę:

"001 000 000 010".

Panašiai ir antrasis duomenų žodis (11011101) duomenų eilutėje turi duomenų žodį (110111), po kurio seka duomenų žodis (01), kuris bus konvertuotas

į kodinių žodžių eilutę "010". Tuo būdu, pirmasis duomenų žodis yra konvertuojamas į tokią kodinių žodžių eilutę:

"001 000 000 010".

Taip duomenų eilutė yra konvertuojama į tokią kodinių žodžių eilutę:

"001 000 000 010 001 000 000 010 ...",

kur išvengiama kartotinių nuoseklių minimalios sekos d atvejų. T.y., pašalinamas derinys, kuris lengvai iššaukia klaidą įrašymo ar atkūrimo dideliu linijiniu tankiu operacijos metu. reikia pažymėti, kad, konvertuojant duomenų eilutę į aukščiau aprašytąją kodinių žodžių eilutę, išlaikomos atitinkamos minimalios sekos d ir maksimalios sekos k reikšmės.

Kaip aprašyta aukščiau, konversija, atlikta moduliavimo įrenginiu 1, yra pagrįsta 2 lentelės konversijos lentele. Reikia pažymėti, kad konversija gali būti atlikta ir pagal 3 lentelės konversijos lentelę. Šiuo atveju, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 8, naudojamas moduliavimo įrenginyje 3, pavaizduotame fig.3, turi lentelę C, skirtą ribiniam ilgiui $i=3$. Iš kitos pusės, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 9 turi lentelę A, skirtą 3 lentelės ribiniam ilgiui $i=2$, lentelę B ribiniam ilgiui $i=3$ ir lentelę D ribiniam ilgiui $i=4$.

Beje, 2 ir 3 lentelėse kiekvienos duomenų eilutės ir kodinių žodžių eilutės poros sudėtis to paties ribinio ilgio grupės viduje gali būti pakeista. Jei tai yra, pavyzdžiui, 2 lentelės ribinio ilgio $i=1$ grupė, pradinė kiekvienos poros sudėtis yra tokia:

	Duomenys	Kodas
$i=1$	11	*0*
	10	001
	01	010

Poros sudėtis gali būti pakeista taip:

	Duomenys	Kodas
$i=1$	11	*0*
	10	010
	01	001

Net ir pakeitus poros sudėtį, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus "1" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui.

Toliau su nuoroda į fig.5 bus paaiškintas šio išradimo demoduliavimo įrenginio pritaikymas. Pagal šį išradimą, kintamo ilgio kodas $(d, k; m, n; r) = (1, 7; 2, 3; 4)$ yra demoduliuojamas atgal į duomenų eilutę, naudojant 2 lentelę.

Kaip parodyta fig.5, demoduliavimo įrenginys 13 turi demoduliavimo įrenginį 14, skirtą demoduliuoti signalą, gautą iš perdavimo linijos, ar signalą, atkurtą iš įrašo laikmenos, naudojant demoduliavimo lentelę ar inversijos konversijos lentelę, ir DSV kontrolės bitų pašalinimo įrenginį 15, išimantį DSV kontrolės bitus, įterptus į duomenų eilutę, gautą demoduliavimo arbitražiniais intervalais metu, iš duomenų eilutės, siekiant atstatyti pirminę duomenų eilutę. Buferis 16 naudojamas laikinam serinių duomenų, generuotų DSV kontrolės bitų pašalinimo įrenginiu 15, saugojimui. Duomenys, saugomi buferyje 16, vėliau nuskaitomi iš anksto nustatyto perdavimo greičiu, sukuriant išvesties signalą. Sinchronizavimo kontrolės įrenginys 17 generuoja sinchronizavimo signalus ir tiekia juos įvairiems komponentams tam, kad valdytų jų operacijų sinchronizavimą.

Fig.6 yra diagrama, vaizduojanti demoduliavimo įrenginio 14 konfigūraciją. Kaip parodyta brėžinyje, demoduliavimo įrenginys 14 turi komparatorių 18, skirtą konvertuoti signalą, gautą iš perdavimo linijos, ar signalą, atkurtą iš įrašo laikmenos, į dvejetainius duomenis. Jei signalas, tiekiamas į komparatorių 18, yra signalas, užbaigiantis NRZI moduliaciją (t.y. lygmens kodavimą), jį paveikia inversinis NRZI kodavimo procesas (t.y. frontalinis kodavimo procesas). ribinio ilgio įvertinimo įrenginys 19 nustato skaitmeninio signalo, priimto iš komparatoriaus 18, ribinį ilgį i . Kuomet minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 20 aptinka specialų kodą, skirtą riboti minimalios sekos d nuoseklius atvejus skaitmeniniame signale, generuotame komparatoriumi 18, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 20 tiekia jo detekcijos signalą ($i=3$), nurodantį ribinį ilgį i , į ribinio ilgio įvertinimo įrenginį 19. 2 lentelei specialus kodas yra "001 000 000". Pagal tą patį pozymį, kuomet sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 21 aptinka specialų kodą, skirtą užtikrinti maksimalią

seką k, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 21 tiekia jo detekcijos signalą ($i=4$), nurodantį ribinį ilgį i, į ribinio ilgio įvertinimo įrenginį 19. 2 lentelėi specialus kodas yra "000 100 100 100" arba "010 100 100 100".

Kiekvienas inversinės konversijos įrenginys 22-1...22-4 turi lentelę, naudojamą $n \times i$ bitų kintamo ilgio kodo inversinei konversijai atgal į $m \times i$ bitų duomenis. Jei naudojama 2 lentelė, inversinės konversijos įrenginiai 22-1...22-4 turi inversinės konversijos lenteles ribiniam ilgiui $i=1...4$, atitinkamai, kurios yra iš esmės tokios pačios, kaip ir konversijos lentelės, įterptos į anksčiau aprašytuosius konversijos įrenginius 10-1...10-4. Multiplekseris 23 parenka vieną iš išvesčių, generuotų inversinės konversijos įrenginiais 22-1...22-4, priklausomai nuo įvertinimo rezultatų, gautų iš ribinio ilgio įvertinimo įrenginio 19.

Toliau su nuoroda į fig.6 yra paaiškintas demoduliacijos įrenginys 14. Signalas, priimtas iš perdavimo linijos, ar signalas, atkurtas iš įrašo laikmenos, yra tiekiamas į komparatorių 18 sulyginimui. Komparatoriaus 18 signalo išvestis yra inversinio NRZI kodo, t.y. kodo, kur "1" žymi pradžią, skaitmeninis signalas. Po to skaitmeninis signalas yra tiekiamas į ribinio ilgio įvertinimo įrenginį 19 nustatyti signalo ribinį ilgį i, naudojant 2 lentelės konversijos lentelę (jei tiksliai – inversinės konversijos lentelę). Įvertinimo rezultatas, t.y. ribinio ilgio i reikšmė, generuota ribinio ilgio įvertinimo įrenginiu 19, yra tiekiama į multiplekserį 23.

Be to, komparatorius 18 siunčia skaitmeninio signalo išvestį į minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginį 20. Minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 20 turi įterptą inversinės konversijos dalį su 2 lentelės inversijos lentelės pakaitos kodu, skirtu riboti minimalios sekos d nuoseklius atvejus. 2 lentelėje pakaitos kodas yra kodinis žodis "001 000 000". Kuomet kodas "001 000 000 ne 100", kataloguotas į inversinės konversijos lentelę, skirtą riboti minimalios sekos d nuoseklius atvejus, aptinkamas skaitmeniniuose duomenyse, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 20 perduoda ribinį ilgį $i=3$ į ribinio ilgio įvertinimo įrenginį 19.

Be to, komparatorius 18 siunčia skaitmeninio signalo išvestį į minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginį 21. Sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 21 turi įterptą inversinės konversijos dalį su 2 lentelės inversijos lentelės pakaitos kodais, skirtais išlaikyti maksimalią seką k.

lentelėje pakaitos kodai yra kodiniai žodžiai "000 100 100 100" ir "010 100 100 100". Kuomet kodinis žodis "000 100 100 100" ar "010 100 100 100", kataloguotas į inversinės konversijos lentelę, skirtą išlaikyti maksimalią seką k , yra aptinkamas skaitmeniniuose duomenyse, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 21 siunčia ribinį ilgį $i=4$ į ribinio ilgio įvertinimo įrenginį 19.

Fig.7 yra diagrama, vaizduojanti moduliavimo kodo, tiekiamo į demoduliacijos įtaisą 13, ribinio ilgio i bendrą nustatymo procesą. Kaip parodyta brėžinyje, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 21 turi įterptą inversinės konversijos dalį su 2 lentelės inversijos lentelės kodiniais žodžiais "000 010 100 100" ir "010 100 100 100". Kuomet 12 bitų kodinių žodžių eilutė, tiekama į sekos ilgio ribos užtikrinimo kodo detekcijos įrenginį 21, sutampa su bet kuriuo kodiniu žodžiu inversinės konversijos dalyje, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 21 siunčia ribinį ilgį $i=4$ į ribinio ilgio įvertinimo įrenginį 19.

Pagal tą patį požymį minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 20 turi įterptą inversinės konversijos dalį su 2 lentelės inversijos lentelės kodiniu žodžiu "001 000 000". Kuomet 12 bitų kodinių žodžių eilutė, tiekama į minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginį 20, sutampa su "001 000 000 ne 100", minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 20 siunčia ribinį ilgį $i=3$ į ribinio ilgio įvertinimo įrenginį 19. Reikia pažymėti, kad aptikti kodinių žodžių eilutės 12 bitų yra iš tiesų "001 000 000 010", nors tai ir neturi nieko bendro konkrečiai su ribinio ilgio i nustatymu.

Ribinio ilgio įvertinimo įrenginys 19 turi įterptą 2 lentelės inversinės konversijos lentelę. Jei 9 bitų kodinių žodžių eilutė, tiekama į ribinio ilgio įvertinimo įrenginį 19, yra "000 100 100" ar "010 100 100", ar 12 bitų kodinių žodžių eilutė yra "000 100 000 ne 100" ar "010 100 000 ne 100", ribinio ilgio įvertinimo įrenginys 19 nustato, kad ribinis ilgis i yra 3. Kita vertus, jei 6 bitų kodinių žodžių eilutė, tiekama į ribinio ilgio įvertinimo įrenginį 19, yra "010 100" ar "000 100", arba jei 9 bitų kodinių žodžių eilutė yra "010 000 ne 100", ribinio ilgio įvertinimo įrenginys 19 nustato, kad ribinis ilgis i yra 2. Priešingu atveju, jei 3 bitų kodinių žodžių eilutė, tiekama į ribinio ilgio įvertinimo įrenginį 19, yra "000", "101" ar "010", ribinio ilgio įvertinimo įrenginys 19 nustato, kad ribinis ilgis i yra 1.

Reikia pažymėti, kad tiek ribinio ilgio įvertinimo įrenginys 19, tiek minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 20, tiek sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 21 taip pat gali atlikti dorojimą i reikšmių didėjančia tvarka, pradedant mažiausia reikšme, t.y. $i=1$, $i=2$, $i=3$ ir $i=4$ tvarka, priešingai, nei parodyta fig.7.

Tarkime, kad ribinio ilgio įvertinimo įrenginys 19 taip pat gali nustatyti ribinio ilgio i reikšmę $i=2/i=3$ ir $i=4$ tvarka ir kodinių žodžių eilutę "000 100 100 100" yra tiekama į ribinio ilgio įvertinimo įrenginį 19. Ribinio ilgio įvertinimo įrenginys 19 palygina jam pateiktą kodinių žodžių eilutę su kodiniais žodžiais, įterptais į konversijos lentelę ribinio ilgio i reikšmių didėjančia tvarka, pradedant mažiausia reikšme, ir įvertina, ar kodinių žodžių eilutė sutampa ar nesutampa su kodiniais žodžiais. Kodinių žodžių eilutę "000 100 100 100", tiekama į ribinio ilgio įvertinimo įrenginį 19, sutampa su vienu iš kodinių žodžių visiems ribiniams ilgiams $i=1$, $i=2$, $i=3$ ir $i=4$. Tokiu atveju, kaip nustatymo taisyklė parenkamas ir tiekiamas į multiplekserį 23 didžiausias ribinis ilgis.

Inversinės konversijos įrenginio 22-1 inversinės konversijos lentelė yra atmintis, kurioje duomenų blokas (11) saugomas adresais "101" ir "000", kai, tuo tarpu, duomenų blokai (10) ir (01) yra saugomi adresais "001" ir "010", atitinkamai. Kiekvieno inversinės konversijos įrenginio 22-2 ir 22-4 inversinės konversijos lentelės yra atmintys, skirtos saugoti duomenis tokiu pačiu būdu, kaip ir inversinės konversijos įrenginys 22-1. $3 \times i$ bitų kodinių žodžių eilutė, tiekama į inversinės konversijos įrenginį 22-i, yra konvertuojama atgal į $2 \times i$ bitų duomenų eilutę, kuri tuomet yra tiekama į multiplekserį 23.

Multiplekseris 23 parenka vieną iš duomenų eilučių, tiekiamų inversinės konversijos įrenginiais 22-1...22-4, pagal ribinio ilgio i, išvedamo ribinio ilgio įvertinimo įrenginiu 19, reikšmės įvertinimo rezultatą.

4 lentelė yra 2 lentelės inversinės konversijos lentelė.

4 LENTELĖ

Inversinės konversijos lentelė (1, 7 ; 2, 3 ; 4)

Kodinio žodžio eilutė	Demoduluota duomenų eilutė
$i=1$ 101	11
000	11

	001	10
	010	01
i=2	010 100	0011
	010 000 (ne 100)	0010
	000 100	0001
i=3	000 100 100	000011
	000 100 000 (ne 100)	000010
	010 100 100	000001
	010 100 000 (ne 100)	000000

i=3: Draudžia minimalų perdavimo sekos ilgį

001 000 000 (ne 100)	110111
----------------------	--------

i=4: Riboja k iki 7

000 100 100 100	00001000
010 100 100 100	00000000

Fig.8 yra diagrama, skirta paaiškinti operacijas, atliekamas DSV kontrolės bito pašalinimo įrenginiu 15. DSV kontrolės bito pašalinimo įrenginys 15 turi vidinį skaitiklį. Kaip parodyta brėžinyje, diagrama pradedama etapu S1, kuriame bitų skaičius duomenų eilutėje, tiekiamoje demoduliavimo įrenginiu 2, nustatomas vidiniu skaitikliu. Po to procesas pereina į etapą S2, kuriame įvertinama, ar bitų skaičius pasiekė reikšmę, nusakančią iš anksto nustatytą duomenų intervalą, kuriame įterpiamas DSV kontrolės bitas. Jei įvertinimo rezultatas nurodo, kad bitų skaičius neatitinka arbitražinį duomenų intervalą, procesas pereina į etapą S3, kuriame duomenys, tiekiami demoduliavimo įrenginiu 2, perduodami į buferį 16. Kita vertus, jei įvertinimo rezultatas rodo, kad bitų skaičius atitinka iš anksto nustatytą duomenų intervalą, nurodantį, kad šis bitas yra DSV kontrolės bitas, etapas S3 praleidžiamas. T.y., šiuo atveju šis duomenų eilutės bitas yra pašalinamas, o ne porsiuočiomas į buferį 16.

Bet kuriuo atveju, procesas pereina į etapą S4, kuriame įvedama sekanti duomenų eilutė. Po to procesas tęsiamas etape S5, kuriame įvertinama, ar jau apdoroti visi duomenys. Jei dar liko duomenų, kuriuos reikia apdoroti, procesas pakartotinai grįžta į etapą S1. Kita vertus, jei įvertinimo, suformuoto etape S5, rezultatas rodo, kad visi duomenys yra apdoroti, - procesas baigiamas.

Išdavoje DSV kontrolės bitų pašalinimo įrenginiu 15 pašalinami kontrolės bitai iš duomenų išvesties. Po to duomenys išvedami buferiu 16.

Remiantis aukščiau pateiktu aprašymu, demoduliavimo įrenginys 14 naudoja 2 lentelės konversijos lentelę arba, tiksliau sakant, 4 lentelės konversijos lentelę. Reikia pažymėti, kad panašus procesas gali būti atliktas, naudojant 3 lentelės konversijos lentelę arba, tiksliau sakant, žemiau pateiktą 5 lentelės konversijos lentelę. Šiuo atveju, minimalios sekos nuoseklių atvejų ribojimo kodo detekcijos įrenginys 20, naudojamas fig.6 pavaizduotame moduliavimo įrenginyje 14, turi lentelę C, skirtą 3 lentelės ribiniam ilgiui $i=4$. Kita vertus, sekos ilgio ribos užtikrinimo kodo detekcijos įrenginys 21 turi lentelę A, skirtą 3 lentelės ribiniam ilgiui $i=2$, lentelę B, skirtą 3 lentelės ribiniam ilgiui $i=3$ ir lentelę D, skirtą 3 lentelės ribiniam ilgiui $i=4$.

5 LENTELĖ

Inversijos konversijos lentelė 1, 7 ; 2, 3 ; 4)

Kodinio žodžio eilutė	Demoduliuota duomenų eilutė
r=1 Pagrindinė lentelė	
101	00
100	01
001	10
000	11

r=2 Pakaitos lentelė A (riboja d iki 1)

100 010	0000
101 010	0001
000 010	1000
001 010	1001

r=3 Pakaitos lentelė B (riboja k iki 8)

000 010 010	111111
001 010 010	111110
101 010 010	011110
100 010 010	011111

r=4 Pakaitos lentelė C (riboja RMTR iki 6)

100 010 010 010	00010001
100 000 010 010	10010001
000 010 010 010	10010001

r=4 Pakaitos lentelė D (riboja k iki 7)

000 001 010 010	11100000
100 001 010 010	11100010
001 010 010 010	11100001
101 010 010 010	11100011

Beje, kai kuriais atvejais yra būtina įterpti į duomenis sinchronizavimo (Sync) signalą. Išradimo įgyvendinimo variantai, kuomet naudojamas moduliavimo įtaisas 1 ir demoduliavimo įtaisas 13, kurie gali kopijuoti duomenis su įterptais sinchronizavimo signalais, yra aprašyti su nuoroda į fig.9 ir 10, atitinkamai. Taip pat šiais atvejais duomenų eilutė yra moduluojama į kintamo ilgio kodą (d, k ; m, n ; r)=(1, 7 ; 2, 3 ; 4).

Kitame šio išradimo moduliavimo įtaise, pavaizduotame fig.9, kur sinchronizavimo signalai yra įterpiami iš anksto nustatytais intervalais, DSV kontrolės bitų nustatymo/įterpimo įrenginio 2 išvestis tiekama į sinchronizavimo signalo nustatymo įrenginį 24. Taip pat į sinchronizavimo signalo nustatymo įrenginį 24 yra tiekama moduliavimo įrenginio 3 išvestis. Kuomet sinchronizavimo signalo nustatymo įrenginys 24 nustato sinchronizavimo signalą iš signalų, tiekiamų DSV kontrolės bitų nustatymo/įterpimo įrenginiu 2 ir moduliavimo įrenginiu 3, sinchronizavimo signalo nustatymo įrenginys 24 siunčia sinchronizavimo signalą į sinchronizavimo signalo įterpimo įrenginį 25. Sinchronizavimo signalo įterpimo įrenginys 25 įterpia sinchronizavimo signalą, tiekiamą sinchronizavimo signalo

nustatymo įrenginiu 24, į moduliuotą signalą, tiekiamą moduliavimo įrenginiu 3, ir tiekia jo išvestį į NRZI kodavimo įrenginį 4. Likusioji konfigūracijos dalis yra tokia pati, kaip ir fig.1 pavaizduotojo moduliavimo įtaiso.

Tuo atveju, kuomet 24 bitų kodinio žodžio derinys veikia kaip sinchronizavimo signalas, pastarasis yra konvertuojamas sinchronizavimo signalo nustatymo įrenginiu 24 pagal 2 lentelę į tokį kodą:

"#01 001 000 000 001 000 000 001",

kur simbolis # žymi bitą, priklausantį nuo betarpiškai prieš tai esančios duomenų eilutės, turinčios DSV kontrolės bitą, delimituotą sinchronizavimo signalo įterpimu. Konkrečiau, kuomet baigties lentelė yra naudojama baigti operaciją ir moduliuoti delimituotą duomenų eilutę, naudojant konversijos lentelę, "#"="1". Kita vertus, kuomet baigties procedūrai naudojama 2 lentelė, o ne baigties lentelė, "#"="0". Tuo būdu, moduliavimo įrenginys 3 išveda "#"="1" ar "#"="0" į sinchronizavimo signalo nustatymo įrenginį 24, kuomet baigties lentelė yra, atitinkamai, arba naudojama, arba nenaudojama. Gavęs iš moduliavimo įrenginio 3 reikšmę "#", sinchronizavimo signalo nustatymo įrenginys 24 prideda sinchronizavimo signalo pradžioje reikšmę "#" ir po to išveda sinchronizavimo signalą į sinchronizavimo signalo įterpimo įrenginį 25.

Sinchronizavimo signalo įterpimo įrenginys 25 įterpia sinchronizavimo signalą, tiekiamą sinchronizavimo signalo nustatymo įrenginiu 24, į moduliuotą signalą, tiekiamą moduliavimo įrenginiu 3, ir tiekia jo išvestį į NRZI kodavimo įrenginį 4. Likusioji proceso dalis yra tokia pati, kaip ir moduliavimo įtaiso 1, pavaizduoto fig.1.

Pirmieji duomenys, einantys po įterpto sinchronizavimo signalo, yra konvertuojami, pradedant jų pradžia, neatsižvelgiant į duomenis, einančius betarpiškai prieš sinchronizavimo signalą. Tiek moduliavimo įrenginys 3, tiek sinchronizavimo signalo nustatymo įrenginys 24 turi skaitiklį, skirtą skaičiuoti iš anksto nustatytus intervalus, kuriais įterpiami sinchronizavimo signalai. Skaitiklio parodymai naudojami nustatyti sinchronizavimo signalo poziciją.

Kaip aprašyta aukščiau, išradimo įgyvendinimo variante, pavaizduotame fig.9, naudojamos 2 lentelės konversijos lentelės. Reikia pažymėti, kad gali būti panaudota ir 3 lentelės konversijos lentelė. Šiuo atveju, sinchronizavimo signalo

nustatymo įrenginys 24 priima pateiktą žemiau 12 bitų kodinį žodį kaip sinchronizavimo signalo derinį:

"xx0 100 000 000 10x",

kur simbolis "x" žymi bitą, priklausantį nuo betarpiškai einančias prieš ir po duomenų eilutes, turinčias DSV kontrolės bitą, delimituotas sinchronizavimo signalo įterpimu. 3 bitai sinchronizavimo signalo pradžioje ir 3 bitai sinchronizavimo signalo gale yra nustatomi 3 lentele sekančiu būdu. Tarkime, kad (p) yra paskutinė duomenų eilutė, delimituota sinchronizavimo signalo įterpimu, ir (q) yra pirmoji duomenų eilutė, einanti betarpiškai po sinchronizavimo signalo. Duomenų eilutė (p1) yra konvertuojama į 3 bitus sinchronizavimo signalo pradžioje, kai, tuo tarpu, duomenų eilutė (0q) yra konvertuojama į 3 bitus sinchronizavimo signalo gale, naudojant 3 lentelę. Gauti sinchronizacijos metu 3 bitai sinchronizavimo signalo pradžioje ir 3 bitai sinchronizavimo signalo gaubia vidurinius bitus "100 000 000", sukuriant derinį. Dėl šios priežasties gali būti generuotas sinchronizavimo signalas, kuris pertraukia reikalingą maksimalią seką k, bet visuomet išlaiko $k=8$ (9T).

Fig.10 yra diagrama, vaizduojanti tipinę išradimo įgyvendinimo varianto, naudojant kitą demoduliavimo įtaisą 13, skirtą demoduliuoti kodą, gautą moduliacijos, atliktos fig.9 pavaizduotu moduliavimo įtaisu 1, metu, konfigūraciją. Kaip pavaizduota fig.10, gaunamas signalas, perduotas iš anksto nustatytu perdavimo keliu, yra tiekiamas į demoduliavimo įrenginį 14 ir sinchronizavimo signalo identifikavimo įrenginį 26. Sinchronizavimo signalo identifikavimo įrenginys 26 naudoja gaunamą signalą ir signalą, gautą iš demoduliavimo įrenginio 2, identifikuoti sinchronizavimo signalą, išvedant sinchronizavimo signalą į sinchronizavimo signalo pašalinimo įrenginį 27. Sinchronizavimo signalo pašalinimo įrenginys 27 pašalina sinchronizavimo signalą iš demoduliuoto signalo, tiekiamo demoduliavimo įrenginiu 14, pagal sinchronizavimo signalo identifikavimo įrenginio 26 signalo išvestį. Tuomet demoduliavimo signalas su pašalintais sinchronizavimo signalais tiekiamas į DSV kontrolės bitų pašalinimo įrenginį 15. Likusioji konfigūracijos dalis yra tokia pati, kaip ir fig.5 pavaizduotame demoduliavimo įtaise 13.

Sinchronizavimo signalo identifikavimo įrenginys 26 turi įterptą skaitiklį, skirtą skaičiuoti kodinius žodžius. Skaitiklio parodymai yra naudojami nustatyti kiekvieno

sinchronizavimo signalo, kurie yra įterpiami į duomenų žodžių eilutę iš anksto nustatytais intervalais, poziciją. Identifikavus sinchronizavimo signalo derinio poziciją, nuskaitomas nustatytas moduliacijos metu “#” bitas. T.y., nuskaitomas bitas sinchronizavimo signalo pradžioje ir išvedamas į demoduliavimo įrenginį 14. Jei pradžios bitas yra “1”, demoduliavimo įrenginys 14 naudoja 2 lentelės baigties lentelę demoduliuojant kodą, einantį betarpiškai prieš sinchronizavimo signalą. Kita vertus, jei pradžios bitas yra “0”, demoduliavimo įrenginys 14 naudoja 2 lentelės konversijos kodų lentelę demoduliavimo kode, einančiame betarpiškai prieš sinchronizavimo signalą. Likusieji sinchronizavimo signalo bitai yra pašalinami, nes jie neneša jokios informacijos.

Sinchronizavimo signalo identifikavimo įrenginys 26 išveda identifikavimo signalą identifiкуoti bitus, sudarančius sinchronizavimo signalą, į sinchronizavimo signalo pašalinimo įrenginį 27. Sinchronizavimo signalo pašalinimo įrenginys 27 pašalina sinchronizavimo signalą iš demoduliuoto signalo, tiekiamo demoduliavimo įrenginiu 14, pagal identifikavimo signalą, išvestą sinchronizavimo signalo identifikavimo įrenginiu 26. Tuomet demoduliuotas signalas su pašalintais sinchronizavimo signalais tiekiamas į DSV kontrolės bitų pašalinimo įrenginį 15.

Kaip aprašyta aukščiau, fig.10 pavaizduotasis demoduliavimo įrenginys 13 naudoja 2 lentelės konversijos lentelę. reikia pažymėti, kad gali būti panaudota ir 3 lentelė. Šiuo atveju, pavyzdžiui, sinchronizavimo signalo identifikavimo įrenginys 26 naudoja skaitiklio parodymus nustatyti kiekvieno sinchronizavimo signalo, kurie yra įterpiami į kodinių žodžių eilutę iš anksto nustatytais intervalais, poziciją. Nustačius sinchronizavimo signalo derinio poziciją, sinchronizavimo signalo identifikavimo įrenginys 26 išveda signalus, nurodančius 3 bitų kodinius žodžius sinchronizavimo signalo derinio pradžioje ir gale, į demoduliavimo įrenginį 14, užklausti demoduliavimo įrenginį 14, kad šie kodiniai žodžiai taip pat yra demoduliuoti, nes kiekvienas iš jų turi duomenų eilutę.

Sinchronizavimo signalo identifikavimo įrenginys 26 išveda signalą, nurodantį sinchronizavimo signalo unikalaus derinio bitus, išskyrus kodinius žodžius, kurie turi duomenų eilutes, į sinchronizavimo signalo pašalinimo įrenginį 27. Tuo būdu, sinchronizavimo signalo pašalinimo įrenginys 27 gali pašalinti tik

sinchronizavimo signalo bitus, t.y. unikalius derinio bitus, nurodytus signalu, gautu iš sinchronizavimo signalo identifikavimo įrenginio 26.

Fig.11 yra diagrama, vaizduojanti pavyzdį kodo, skirtą įrašyti su įterptais į jį sinchronizavimo signalais ir DSV kontrolės bitais. Šiame pavyzdyje sinchronizavimo signalas yra 24 bitų kodinis žodis. DSV kontrolė atliekama 56 duomenų bitų intervalais, o sinchronizavimo signalas yra įterpiamas 5 DSV kontrolės operacijoms. Tuo būdu, kiekviename sinchronizacijos signalo kodinių žodžių, t.y. kanalo bitų yra :

$24 + (1 + 56 + 1 + 56 + 1 + 56 + 1 + 56 + 1 + 56 + 1) \times 1,5 = 453$ kodiniai žodžiai (kanalo bitai).

Santykinis perteklumas, įterptas į duomenų žodžius, yra apie 7,3%, kas matyti iš toliau pateikto skaičiavimo:

$$\text{Duomenų kiekis} = (56 \times 5) \times 1,5 / 453 = 420 / 453 = 0,927$$

$$\text{Tuo būdu, santykinis perteklumas} = 1 - 0,927 = 0,0728 = 7,3\%.$$

Išradėjai ir kiti atlieka modeliavimą, naudojant aukščiau aprašytas konversijos lenteles, gauti moduliacijos rezultatus. Duomenų eilutės, turinčios įterptus DSV kontrolės bitus su ribotais $T_{\min}$ nuosekliais atvejais, moduliavimo rezultatai yra aprašyti žemiau. Modeliavimui naudojamos 2 ir 3 lentelės. Taip pat palyginimui buvo atliktas modeliavimas, naudojant 1 lentelę įprastai RLL(1-7) moduliacijai.

Šiuose modeliavimuose DSV kontrolė buvo įvykdyta, įterpiant po 1 DSV kontrolės bitą kiekvieniems atsitiktinių duomenų, susidedančių iš 13107200 arbitražiškai sukurtų bitų, 56 bitams, po to duomenys buvo konvertuoti į kodinių žodžių eilutę (ar kanalo bitų vorą), naudojant 2 ar 3 lentelės konversijos lentelę. Atliekant kitą modeliavimą, atsitiktiniai duomenys, susidedantys iš 13107200 arbitražiškai sukurtų bitų, buvo konvertuoti į kodinių žodžių eilutę (ar kanalo bitų vorą), naudojant 1 lentelės konversijos lentelę, ir po to 2 kanalo bitai kaip DSV kontrolės bitai buvo terpti į gautosios kodinių žodžių eilutės kiekvienus 112 kodinių žodžių ar 112 kanalo bitų DSV kontrolei atlikti.

Priežastis, kodėl modeliavime, naudojant 2 ar 3 lentelę, buvo įterptas 1 DSV kontrolės bitas į kiekvienus 56 duomenų bitus, o modeliavime, naudojant 1 lentelę, buvo įterpti 2 DSV kontrolės bitai į kiekvienus 112 kodinių žodžių, yra tai, kad

siekama DSV kontrolės bitais sukurti santykinį perteklumą, vienodą abiejuose modeliavimuose. Jei bitų, reikalingų DSV kontrolei, skaičius vienu atveju yra kitoks, nei kitu atveju, o santykinis perteklumas turi būti sukurtas vienodas abiem atvejais, 2 ar 3 lentelės, kurios leidžia atlikti efektyvią DSV kontrolę, suteikia gerą žemo dažnio juostos charakteristiką, lyginant su 1 lentele.

Skaitmeninės modeliavimo rezultatų reikšmės apskaičiuojamos taip:

Ren_cnt [nuo 1 iki 10] Atvejų nuo 1 minimalios sekos iki 10 nuoseklių minimalių sekų skaičius.

T_dydis [nuo 2 iki 10] Atvejų nuo 2T sekos iki 10T sekos skaičius.

Suma: Bitų skaičius

Viso: Sekų ilgių skaičius, t.y. bendras sekų 2T, 3T ir t.t. skaičius.

Vidutinė seka: (Suma/Viso)

Sekos pasiskirstymo skaitmeninės reikšmės: $(T_dydis[i] * (i) / \text{Suma})$, kur $i=2, 3, 4, \dots, 10$.

Skaitmeninės reikšmės 6 lentelės eilutėse 2T ... 10T yra sekos pasiskirstymo skaitmeninės reikšmės.

Nuoseklių minimalių sekų pasiskirstymo skaitmeninės reikšmės:

$(Ren_cnt[i] * (i)) / T_dydis[2T]$, kur $i=1, 2, 3, 4, \dots, 10$.

Skaitmeninės reikšmės 6 lentelės eilutėse RMTR (1) ... RMTR (10) yra nuoseklių minimalių sekų pasiskirstymo skaitmeninės reikšmės.

Max_RMTR: Minimalios sekos kartotinių pikų DSV maksimalus skaičius: apskaičiuotų DSV reikšmių pikų skaičius teigiamose ir neigiamose pusėse, pastebėtas DSV kontrolės, vykdomos kanalo bitų voroje, procese.

Santykinio perteklumo, gauto, įterpiant 1 DSV bitą kiekvieniams 56 duomenų bitams, apskaičiavimas yra pagrįstas tuo, kad 1 DSV bitas skirtas kiekvieniams 56 duomenų bitams. Tuo būdu, santykinis perteklumas apskaičiuojamas taip:

$$\text{Santykinis perteklumas} = 1/(1+56) = 1,75\%$$

Santykinio perteklumo, gauto, įterpiant 2 DSV bitus kiekvieniams 112 kodinių žodžių bitams, apskaičiavimas yra pagrįstas tuo, kad 2 DSV bitas skirti kiekvieniams 112 kodinių žodžių bitams. Tuo būdu, santykinis perteklumas apskaičiuojamas taip:

$$\text{Santykinis perteklumas} = 2/(2+112) = 1,75\%$$

Tuo būdu, abiem atvejais gautas tas pats perteklumas.

6 LENTELĖ

PP17 Palyginimas

	2 LENTELĖ	3 LENTELĖ	1 LENTELĖ	
	17PP-32	17PP-52	+2 bitai – DC (DSV kontrolė)	Be – DCC (Nėra DSV kontrolės)

Vidurkis

Seka	3,3665	3,4048	3,3016	3,2868
Suma	20011947	20011947	20011788	19660782
Viso	5944349	5877654	6061150	5981807

2T	0,2256	0,2246	0,2417	0,1419
3T	0,2217	0,2246	0,2417	1,1419
4T	0,1948	0,1935	0,1902	0,1915
5T	0,1499	0,1491	0,1502	0,1511
6T	0,1109	0,1904	0,1135	0,1141
7T	0,0579	0,0814	0,0561	0,0544
8T	0,0392	0,0351	0,0218	0,0188
9T	-----	0,0023	-----	-----
10T	-----	0,0009	-----	-----

RMTR (1)	0,3837	0,3890	0,2628	0,3641
RMTR (2)	0,3107	0,3137	0,2884	0,2883
RMTR (3)	0,1738	0,1906	0,7117	0,1716
RMTR (4)	0,0938	0,0806	0,0909	0,0907
RMTR (5)	0,0299	0,0228	0,0456	0,0452
RMTR (6)	0,0081	0,0033	0,0219	0,0217
RMTR (7)	-----	-----	0,0100	0,0099
RMTR (8)	-----	-----	0,0047	0,0046
RMTR (9)	-----	-----	0,0022	0,0022
Max_RMTR	6	6	18	18

DSV pikas # -36 iki 36 # -35 iki 40 * -46 iki 43 * -1783 iki 3433

"#": 56 duomenų bitai + 1 dc bitas, 1,75%) ("*": 112 c bitų + 2 dc bitai, 1,75%)

Aukščiau pateikti rezultatai patvirtina, kad, naudojant 2 ir 3 lenteles, gaunama RLL(1, 7) sistema, kai tuo pat metu yra išlaikomos minimali ir maksimali sekos, o minimalios sekos nuoseklių atvejų skaičius apribojamas iki 6. Be to, DSV rezultatai patvirtina, kad DSV kontrolė gali būti įvykdyta duomenų eilutėje (t.y. DSV piko reikšmės išlaikomos iš anksto nustatytame diapazone) ir, šiuo atveju, kadangi DSV kontrolės bitų efektyvumas yra didelis, yra įmanoma gauti žemo dažnio juostos komponentus, kurie yra labiau tinkami, nei įprastas DSV bitų įterpimo į kodinių žodžių eilutę (kanalo bitų vorą) būdas. DSV rezultatai patvirtina, kad, turint 2 lentelę, skirtumas tarp DSV teigiamo ir neigiamo pikų yra 89 ($=46+43$), kai, tuo tarpu, turint 2 ir 3 lenteles, skirtumai yra 72 ($=36+36$ ir 75 ($=35+40$), atitinkamai, ir jie abu yra mažesni už 1 lentelės reikšmę.

Iš aukščiau pateikto aprašymo aišku, kad, lyginant su įprasta RLL(1-7) sistema, t.y. sistema, pagrįsta 1 lentele, vadinamoji 17PP sistema, naudojanti 2 ar 3 lenteles, gali apriboti minimalios sekos pasikartojimų skaičių daugiausiai iki 6. Todėl galima tikėtis klaidos charakteristikos, esant aukštam linijiniam tankiui, patobulinimo.

Be to, kadangi DSV kontrolės efektyvumas yra puikus, DSV kontrolės vykdymas 17PP sistemoje tuo pačiu 1,75% santykinu pertekliumu kaip ir įprastoje RLL(1-7) sistemoje pasireiškia mažesniu skirtumu tarp neigiamų ir teigiamų piko reikšmių. Dėl tos priežasties, kad žemo dažnio juostos komponentai gali būti nuslopinti, gali būti atliktos stabilios duomenų įrašymo/atkūrimo operacijos.

Modeliavimas dar buvo atliktas, siekiant nustatyti demoduliacijos klaidos, iššauktos bito postūmiu kanalo bitų, generuotų pagal tuos pačius atsitiktinius duomenis, kaip ir aukščiau aprašytuoju atveju, voroje, sklidimą. Tyrimo rezultatas rodo, kad blogiausias klaidos sklidimas 17PP sistemoje yra 3 baitai. Tačiau rezultatas taip pat patvirtina, kad klaidos esamos generacijos dažnis yra visoks, išskyrus 0, reikšmę, pažymint ne tokį ryškų pablogėjimą, lyginant su įprasta RLL(1-7) sistema. Nustatyta vidutinė baito klaida 1 lentelei yra 1,014 baitų, 2 lentelei – 1,167 baitų ir 3 lentelei 1,174 baitų. Reikia pažymėti, kad šio išradimo konversijos lentelėse klaidos koeficiento skaitmeninės reikšmės turi DSV kontrolės bitus, tačiau įprastos RLL(1-7) sistemos skaitmeninė reikšmė neturi DSV kontrolės bitų. T.y. matavimai nebūtinai turi būti atlikti, esant tokioms pačioms sąlygoms. Matavimo

sąlygų skirtumas gali paveikti skaitmenines reikšmes, todėl palyginimo metu yra būtina priimti domėn skirtumo poveikį.

7 LENTELĖ

Postūmio klaidos charakteristikos

	2 lentelė	3 lentelė	1 lentelė
	17PP-32	17PP-52	+2 bitai-DC
Blogiausias atvejis	3 baitai	3 baitai	2 baitai
(dc bitai)	Yra	Yra	Nėra
Baito klaida (0)	0,028	0,096	0,080
Baito klaida (1)	0,777	0,0635	0,826
Baito klaida (2)	0,195	0,268	0,094
Baito klaida (3)	0,000	0,001	-----

Vidurkis

Baito klaidos koeficientas 1,167 baitai 1,174 baitai 1,014 baitai

Kaip aprašyta aukščiau, šio išradimo konversijos lentelės su minimalia seka d, lygia 1, maksimalia seka k, lygia 7, ir konversijos sparta m/n, lygia 2/3, turi pakaitos kodus, ribojančius minimalios sekos nuoseklių atvejų skaičių, iššaukiant tokius efektus:

- (1) Įrašymo ir atkūrimo dideliu linijiniu tankiu ir tolerancijos tangentiniam pokrypiui pagerinimą.
- (2) Yra įmanoma sumažinti žemo lygio dalių skaičių, padidinti banginio apdorojimo, tokio kaip AGC ir PLL, tikslumą ir, vadinasi, pagerinti bendrą charakteristiką.
- (3) Lyginant su įprasta sistema, yra įmanoma gauti konstrukciją, turinčią trumpą bito kodo atminties takelį ar pan., ir, vadinasi, sumažinti kanalo dydį.

Be to, "1" skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus 111" skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, suteikiant tokius efektus:

- (4) Gali būti sumažintas DSV kontrolės perteklinių bitų skaičius.

(5) Turint minimalią seką d , lygią 1, ir konversijos parametrus (m, n) , lygius (2, 3), DSV kontrolė gali būti atlikta su 1,5 bitų kodiniu žodžiu.

(6) Minimali ir maksimali sekos d ir k gali būti išlaikytos, esant mažam santykiniam perteklumui.

Be to, į konversijos lenteles specialiai įtraukti pakaitos kodai išlaikyti sekos ilgio ribą, iššaukiant tokius efektus:

(7) Lentelės yra kompaktiškos.

(8) Moduliavimo klaidos, iššauktos bito postūmiu, sklidimo režimas gali būti toks pat kaip ir įprastos, pagrastos 1 lentele, sistemos.

Reikia pažymėti, kad programos laikmena, skirta kompiuterių programai, vykdomai atlikti aukščiau aprašytą procesą, gali būti ne tik tokia įrašo laikmena, kaip magnetinis diskas, bet ir CD-ROM bei kieta atmintis, komunikacijos priemonė, pavyzdžiui, tinklas ir palydovas.

Kaip aprašyta aukščiau, remiantis moduliavimo įrenginiu, apibūdintu apibrėžties 1 punkte, moduliavimo būdu, apibūdintu apibrėžties 23 punkte, programos laikmena, apibūdinta apibrėžties 24 punkte, demoduliavimo įrenginiu, apibūdintu apibrėžties 25 punkte, demoduliavimo būdu, apibūdintu apibrėžties 28 punkte, ir programos laikmena, apibūdinta apibrėžties 29 punkte, atliekama konversija, paremta konversijos lentele, įgalinančia konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$;

pirmuosius pakaitos kodus apriboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus išlaikyti sekos ilgio ribą k .

Štai kodėl gali būti atlikta DSV kontrolė, naudojant mažą perteklinių bitų kiekį, o kodinių žodžių eilutė gali būti įrašyta ir atkurta dideliu tankiu tik su keliomis klaidomis. Be to, demoduliavimo klaidos, iššauktos bito postūmiu, sklidimas gali būti apribotas.

IŠRADIMO APIBRĖŽTIS

1. Moduliavimo įrenginys, susidedantis iš konversijos priemonių ir skirtas konvertuoti duomenis, kurių pagrindinis ilgis yra m bitų, į kintamo ilgio kodą (d , k ; m , n ; r), kurio pagrindinis ilgis yra n bitų, kur d yra minimali seka, o k yra sekos ilgio riba, besiskiriantis tuo, kad turi konversijos priemonę, pritaikytą konvertuoti įvesties duomenis į kodą pagal konversijos lentelę, įgalinančią konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:
 - pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$;
 - pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir
 - antruosius pakaitos kodus, skirtus išlaikyti sekos ilgio ribą k .
2. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad porų, susidedančių iš duomenų eilutės ir kodinės eilutės, sudarančių pagrindinius kodus ribiniam ilgiui i , lygiam 1, skaičius yra mažesnis nei 4 ($=2^m=2^2$).
3. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad konversijos lentelių pagrindiniai kodai turi kintamo ilgio struktūrą.
4. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad konversijos lentelių pagrindiniai kodai turi kodą '*0*', kuriame simbolis * yra neapibrėžtumo kodas, kuris yra '0', jei einantis betarpiškai prieš arba po kodinis žodis yra '1', ir '1', jei einantis betarpiškai prieš arba po kodinis žodis yra '0', turint omenyje, kad kodas '*0*' yra arba '000', arba '101'.
5. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad konversijos lentelių konversijos kodai turi kodus, nustatomus pagal sekančią betarpiškai kodinių žodžių eilutę ar sekančią betarpiškai duomenų eilutę.

6. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad konversijos lentelių konversijos kodai turi kodus, nustatomus pagal sekančią betarpiškai kodinių žodžių eilutę ar specifinės rūšies kodinių žodžių eilutę.
7. Moduliavimo įrenginys pagal 5 punktą, besiskiriantis tuo, kad turi kodus, nustatomus pagal sekančią betarpiškai kodinių žodžių eilutę ar sekančią betarpiškai pirmojo ir antrojo pakaitos kodų duomenų eilutę.
8. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad porų, susidedančių iš duomenų eilutės ir kodinės eilutės, sudarančių pagrindinius kodus ribiniam ilgiui i , lygiam 1, skaičius yra lygus 4 ($=2^m = 2^2$).
9. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad, kuomet ribinis ilgis i yra 2 ar didesnis, visi konversijos kodai yra pirmieji ir antrieji pakaitos kodai.
10. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad konversijos kodai ribiniam ilgiui i , lygiam 2, yra kodai, skirti išlaikyti minimalią seką d ties 1.
11. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad konversijos lentelių konversijos kodai turi kodus, nustatomus pagal einančią betarpiškai prieš kodinių žodžių eilutę.
12. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad turi sinchronizavimo signalo įterpimo priemonę, skirtą įterpti sinchronizavimo signalą su unikaliu deriniu, kurio neturi konversijos lentelės konversijos kodai, į kodinių žodžių eilutės arbitražinę poziciją.
13. Moduliavimo įrenginys pagal 12 punktą, besiskiriantis tuo, kad unikalus derinys yra derinys, pertraukiantis maksimalią seką k .

14. Moduliavimo įrenginys pagal 12 punktą, besiskiriantis tuo, kad unikalus derinys yra derinys, išlaikantis minimalią seką d.
15. Moduliavimo įrenginys pagal 12 punktą, besiskiriantis tuo, kad sinchronizavimo signale esantis unikalus derinys turi 1 kodinį žodį jo pradžioje, tarnaujantį jungties su kodiniu žodžiu, gautuoju konversijos link einančių betarpiškai prieš duomenų metu, bitu, antrąjį bitą, skirtą išlaikyti minimalią seką d, ir trečiąjį bitą.
16. Moduliavimo įrenginys pagal 12 punktą, besiskiriantis tuo, kad sinchronizavimo signalo dydis yra mažiausiai 12 kodinių žodžių.
17. Moduliavimo įrenginys pagal 12 punktą, besiskiriantis tuo, kad sinchronizavimo signalas, kurio dydis yra mažiausiai 21 kodinis žodis, turi mažiausiai 2 derinius su maksimalia seka k, lygia 8.
18. Moduliavimo įrenginys pagal 12 punktą, besiskiriantis tuo, kad konversijos lentelės konversijos kodai turi baigties kodus, užbaigiančius gautąjį konversijos metu kodą.
19. Moduliavimo įrenginys pagal 18 punktą, besiskiriantis tuo, kad baigties kodai yra priskirti pagrindiniams ribinio ilgio i kodams, kuriems porų, susidedančių iš duomenų eilutės ir kodinės eilutės, sudarančių šiuos pagrindinius kodus, skaičius yra mažesnis nei 4 ($=2^m=2^2$), ir įgalina konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui.
20. Moduliavimo įrenginys pagal 18 punktą, besiskiriantis tuo, kad, siekiant identifikuoti baigties kodą, 1 kodinis žodis sinchronizavimo signalo pradžioje,

tarnaujantis jungties bitu, nustatomas '1', kuomet baigties kodas yra naudojamas, ir '0', kuomet baigties kodas nenaudojamas.

21. Moduliavimo įrenginys pagal 12 punktą, besiskiriantis tuo, kad unikalus derinys yra įterptas tarp 3 bitų sinchronizavimo signalo pradžioje ir 3 bitų sinchronizavimo signalo pabaigoje, tiek 3 bitai pradžioje, tiek 3 bitai pabaigoje yra naudojami kaip sandūra, turinti mišrius duomenis ir jungties bitus.
22. Moduliavimo įrenginys pagal 12 punktą, besiskiriantis tuo, kad:
 - pirmojo iš 3 bitų sinchronizavimo signalo pradžioje reikšmė atspindi duomenų žodžius prieš konversiją m bitų elementais;
 - antrasis iš 3 bitų yra nustatytas ties '1' nurodyti sinchronizavimo signalą;
 - pirmasis iš 3 bitų sinchronizavimo signalo pabaigoje yra nustatytas ties '0' nurodyti sinchronizavimo signalą; ir
 - antrojo iš 3 bitų sinchronizavimo signalo pabaigoje reikšmė atspindi duomenų žodžius prieš konversiją m bitų elementais.
23. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad turi DSV kontrolės priemonę, skirtą kontroliuoti įvesties duomenų DSV ir tiekti šią DSV į konversijos priemonę.
24. Moduliavimo įrenginys pagal 1 punktą, besiskiriantis tuo, kad konversijos priemonė turi:
 - pirmąją kodo detekcijos priemonę, skirtą aptikti pirmuosius pakaitos kodus, ribojančius minimalios sekos d nuoseklių atvejų skaičių; ir
 - antrąją kodo detekcijos priemonę, skirtą aptikti antruosius pakaitos kodus, išlaikančius sekos ilgio ribą.
25. Moduliavimo būdas, pritaikomas moduliavimo įrenginyje, skirtame konvertuoti duomenis, kurių pagrindinis ilgis yra m bitų, į kintamo ilgio kodą (d, k ; m, n ; r), kurio pagrindinis ilgis yra n bitų, kur d yra minimali seka, o k yra sekos ilgio riba, besiskiriantis tuo, kad konvertuoja įvesties duomenis į kodą pagal

konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos ilgio ribą k .

26. Programos laikmena, skirta perduoti programą, atliekančią duomenų dorojimą, konvertuojant konversijos etapo metu įvesties duomenis į kodą pagal konversijos lentelės duomenis moduliavimo įrenginyje, konvertuojančiame duomenis, kurių pagrindinis ilgis yra m bitų, į kintamo ilgio kodą ($d, k ; m, n ; r$), kurio pagrindinis ilgis yra n bitų, kur d yra minimali seka, o k yra sekos ilgio riba, besiskirianti tuo, kad turi konversijos lentelę, įgalinančią konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos ilgio ribą k .

27. Demoduliavimo įrenginys, susidedantis iš konversijos priemonių ir skirtas konvertuoti kintamo ilgio kodą ($d, k ; m, n ; r$), kurio pagrindinis ilgis yra n bitų, į duomenis, kurių pagrindinis ilgis yra n bitų, kur d yra minimali seka, o k yra sekos ilgio riba, besiskirianti tuo, kad turi konversijos priemonę, pritaikytą konvertuoti įvesties duomenis į kodą pagal konversijos lentelę, įgalinančią konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:
- pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos ilgio ribą k .

28. Demoduliavimo įrenginys pagal 27 punktą, besiskiriantis tuo, kad turi bitų pašalinimo priemonę, kuri skirta pašalinti perteklinius bitus, įterptus į kodą iš anksto nustatytais intervalais.

29. Demoduliavimo įrenginys pagal 28 punktą, besiskiriantis tuo, kad pertekliniai bitai yra DSV bitai ar sinchronizavimo signalai.

30. Demoduliavimo būdas, taikomas demoduliavimo įrenginyje, apimantis kintamo ilgio kodo $(d, k; m, n; r)$, kurio pagrindinis ilgis yra n bitų, konvertavimo į duomenis, kurių pagrindinis ilgis yra m bitų, etapą, kur d yra minimali seka, o k yra sekos ilgio riba, besiskiriantis tuo, kad konversijos etapo metu konvertuoja įvesties kodą į duomenis pagal konversijos lentelę, įgalinančią taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos ilgio ribą k .

31. Programos laikmena, skirta perduoti programą, konvertuojančią įvesties duomenis į kodą pagal konversijos lentelę demoduliavimo įrenginyje, konvertuojančiame kintamo ilgio kodą $(d, k; m, n; r)$, kurio pagrindinis ilgis yra n bitų, į duomenis, kurių pagrindinis ilgis yra m bitų, kur d yra minimali seka, o k yra sekos ilgio riba, besiskirianti tuo, kad turi konversijos lentelę, įgalinančią konversijos taisyklę, pagal kurią '1' skaičiaus elemento duomenų eilutėje dalybos iš 2 likutis, turintis reikšmę 0 ar 1, bus visuomet lygus '1' skaičiaus

elemento kodinių žodžių eilutėje, gautoje konvertavus šią duomenų eilutę, dalybos iš 2 likučiui, o konversijos lentelės konversijos kodai turi:

pagrindinius kodus, kai $d=1$, $k=7$, $m=2$ ir $n=3$;

pirmuosius pakaitos kodus, skirtus riboti minimalios sekos d nuoseklių atvejų skaičių; ir

antruosius pakaitos kodus, skirtus išlaikyti sekos ilgio ribą k .

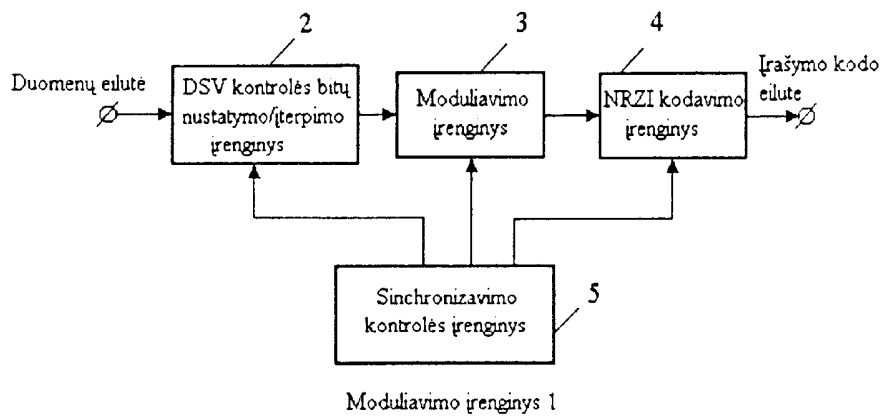


Fig.1

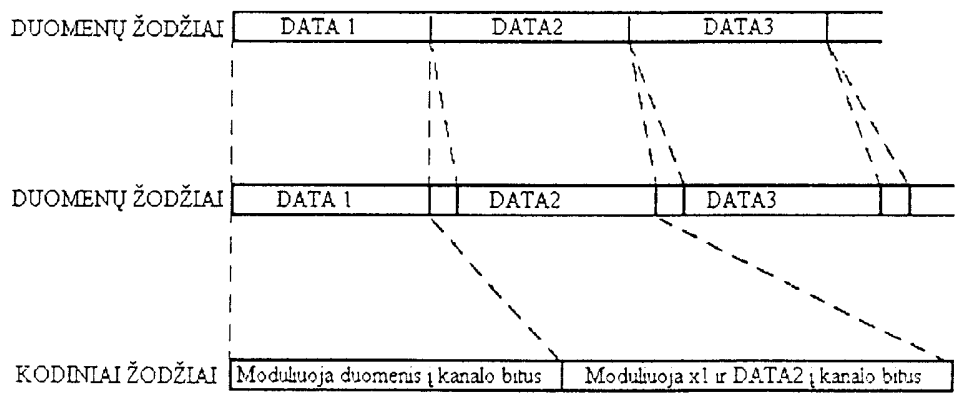


Fig.2

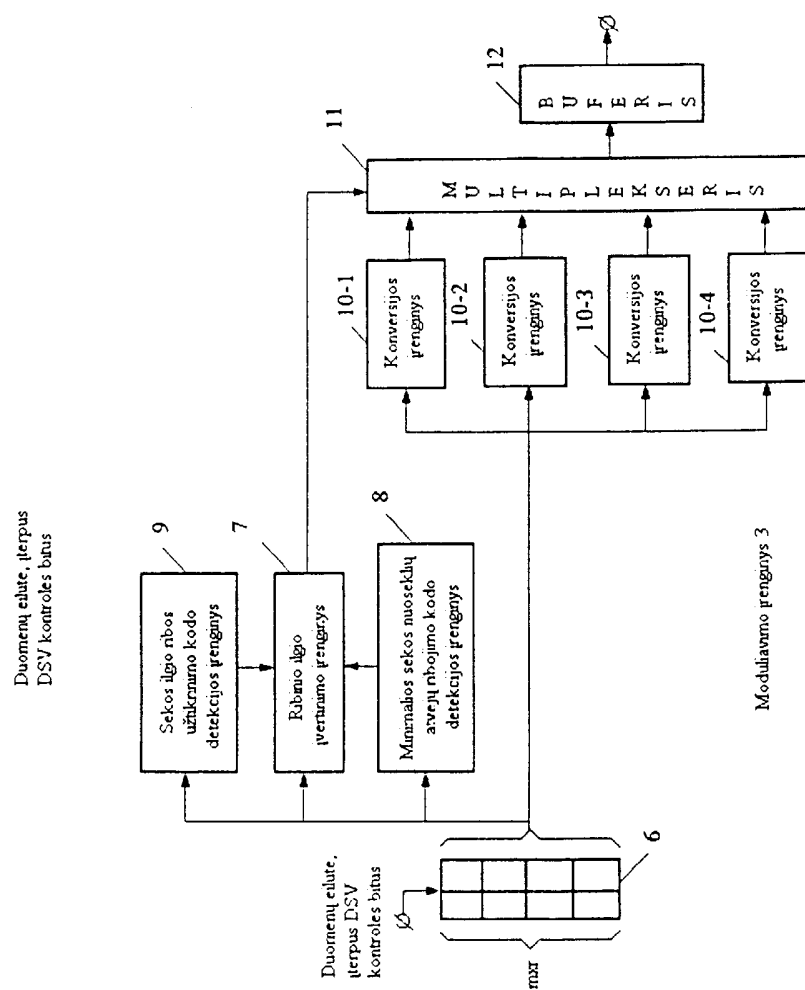


Fig.3

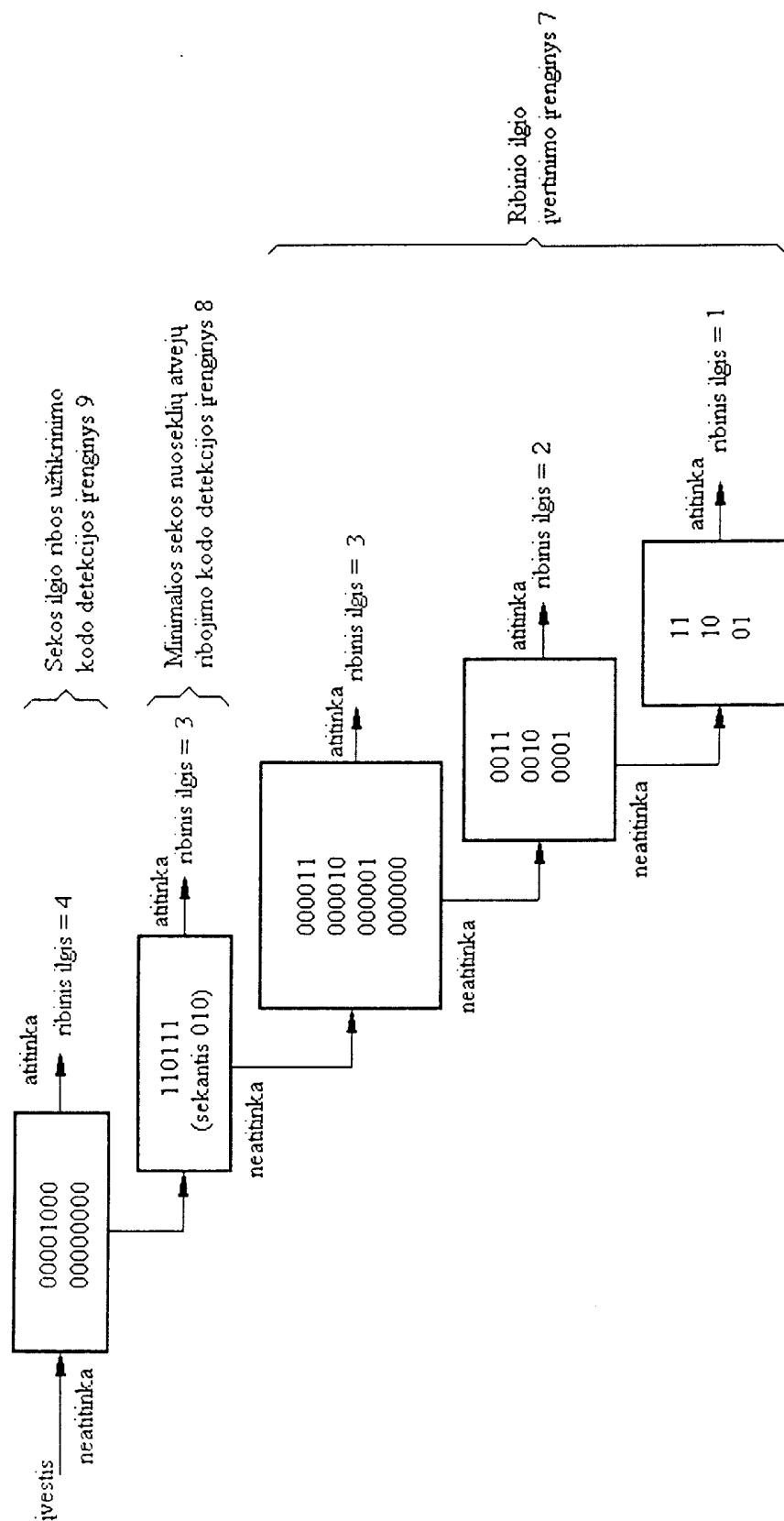


Fig.4

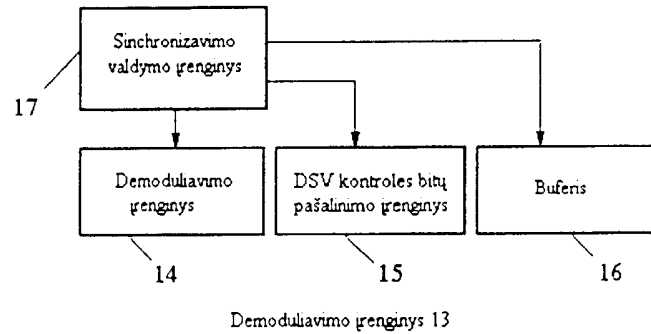


Fig.5

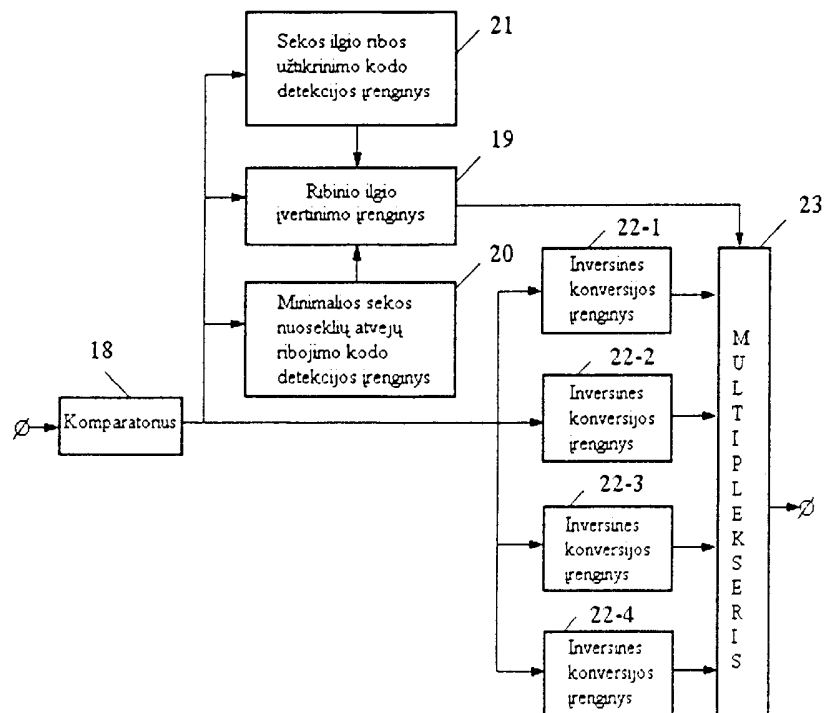


Fig.6

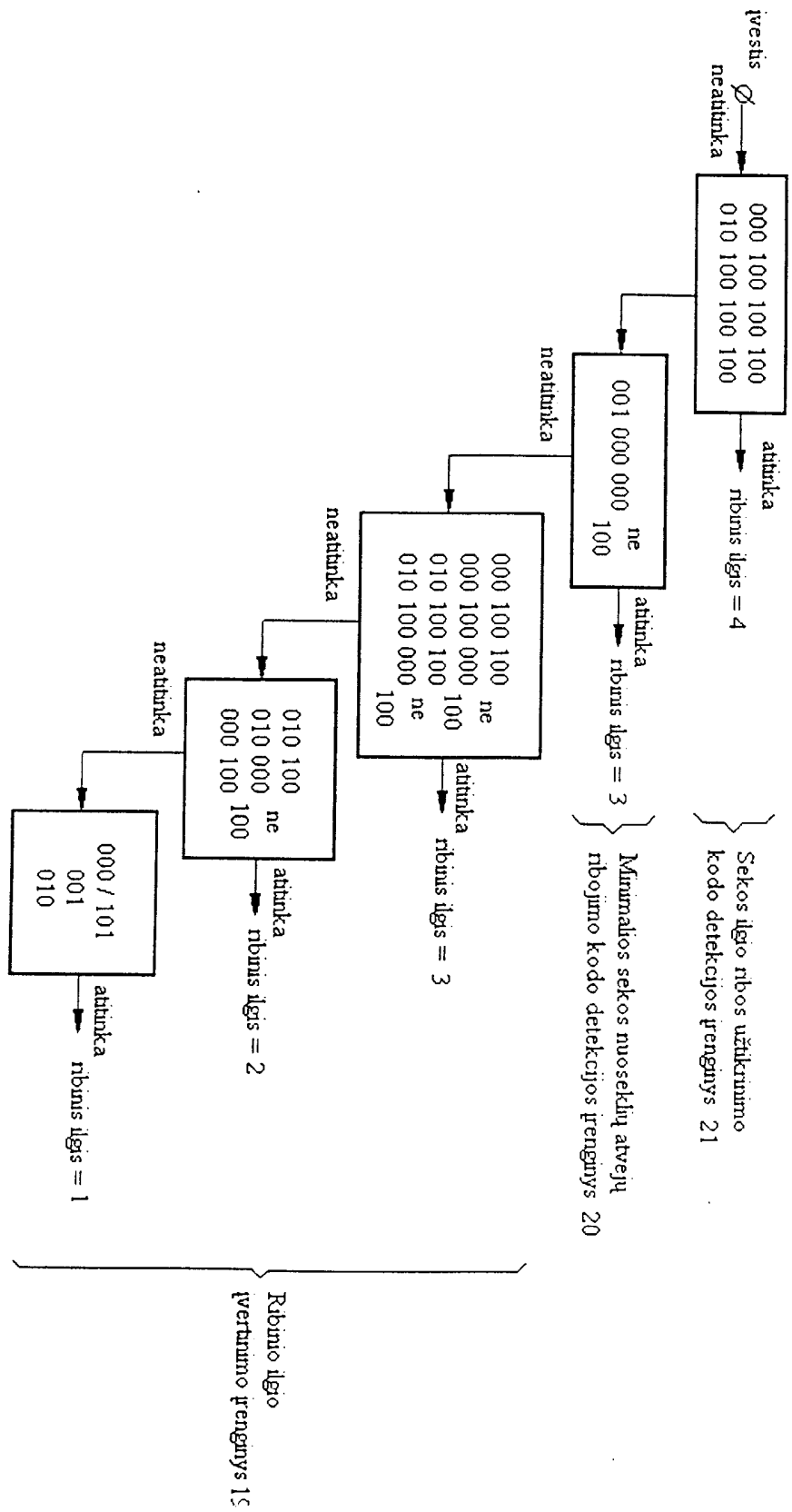


Fig.7

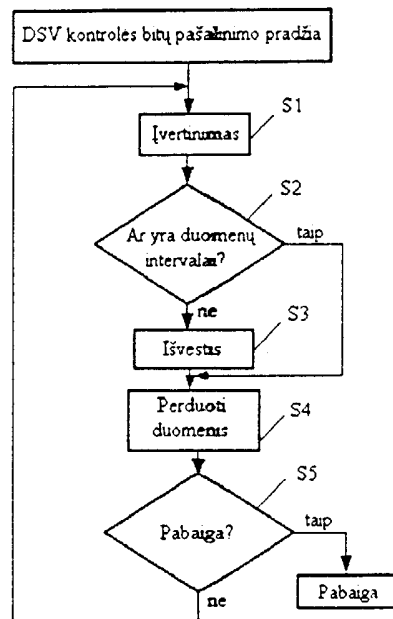


Fig.8

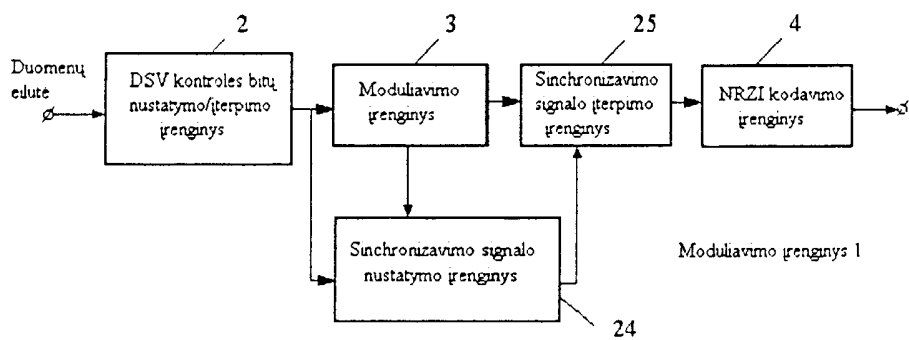


Fig.9

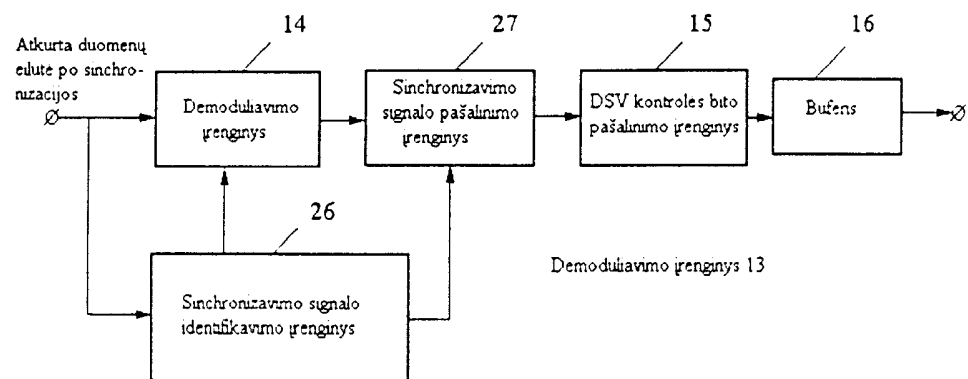


Fig.10

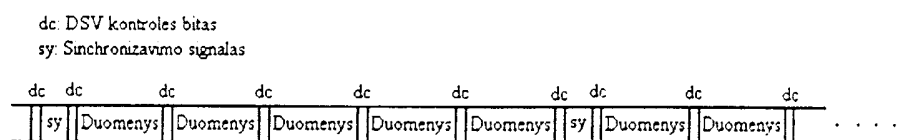


Fig.11