

Даний винахід відноситься до пристрою і способу модуляції, пристрою і способу демодуляції, а також до програмоносія. Зокрема, даний винахід відноситься до переважного пристрою модуляції і переважного способу модуляції, переважного пристрою демодуляції і переважного способу демодуляції, а також до переважного програмоносія, що використовується в операції запису даних на носій при високій густині запису, і зчитування даних, записаних на носій при високій густині запису.

При передачі даних по лінії передачі або при записуванні на носій, такий, як магнітний, оптичний або магнітооптичний диск, дані перед передачею або записом модулюються шляхом перетворення в код, відповідний лінії передачі або носієві, на який виконується запис. Відомим способом модуляції є блокове кодування. При блоковому кодуванні рядок даних розбивається на блоки, кожний з яких містить $m \times i$ бітів. Кожний з блоків, званих далі словами даних, згодом перетворюється в кодове слово, що містить $n \times i$ бітів, відповідно до правила кодування. При $i = 1$ це кодове слово є кодом фіксованої довжини. У випадку, якщо i приймає множину значень, вибраних в діапазоні від 1 до i_{\max} (максимального значення i), кодове слово, що отримується, є кодом змінної довжини. У загальному випадку код, що отримується при блоковому кодуванні, означається як код змінної довжини (d, k, m, n, i, g).

Тут i називається довжиною кодового обмеження, а g дорівнює i_{\max} , максимальній довжині кодового обмеження, d - це мінімальна кількість нулів, що з'являються між двома послідовними одиницями, d називається мінімальною серією нулів. З іншого боку, k - це максимальна кількість нулів, що з'являються між двома послідовними одиницями, k називається максимальною серією нулів.

При записуванні коду змінної довжини, отриманого за допомогою описаного вище блокового кодування, на носій даних, такий, як оптичний або магнітооптичний диск, компакт-диск (CD) або міні-диск (MD), код змінної довжини проходить NRZI - модуляцію (без повернення до нуля з інверсією), при якій кожна "1" коду змінної довжини інтерпретується як інверсія, тоді як "0" інтерпретується як відсутність інверсії. Після цього відбувається запис коду змінної довжини, що пройшов NRZI - модуляцію. Код змінної довжини, що пройшов NRZI - модуляцію, називається послідовністю сигналів запису. На магнітооптичні диски, відповідні раннім специфікаціям ISO, що передбачають невисоку густину запису, послідовність бітів, що пройшли модуляцію перед записом, записується без змін і без NRZI - модуляції.

Нехай T_{\min} і T_{\max} означають відповідно мінімальний і максимальний періоди інверсії послідовності сигналів запису. У цьому випадку, для запису послідовності сигналів з високою густиною в напрямі лінійної швидкості переважним є тривалий мінімальний період інверсії T_{\min} або велика мінімальна серія d . При цьому для синхронізації бажано мати короткий максимальний період інверсії T_{\max} або невелику максимальну серію k . Для задоволення цим вимогам був запропонований цілий ряд способів модуляції.

Так, для оптичних, магнітних і магнітооптичних дисків запропоновані або в цей час використовуються способи модуляції для генерації коду змінної довжини RLL(1 - 7), який також означається як (1, 7; m, n, i, g), коду змінної довжини RLL(2 - 7), що означається також як (2, 7; m, n, i, g), і коду фіксованої довжини RLL(1 - 7), що означається також як (1, 7; $m, n, i, 1$), що використовується в ISO - специфікації MO. Для дискових пристроїв, що розробляються в даний час, таких, як оптичні диски і магнітооптичні диски з високою густиною запису, широко використовується RLL - код (код з обмеженням довжини серій) з мінімальною серією d , що дорівнює одиниці.

Нижче приведений приклад таблиці перетворення RLL - коду (1 - 7) змінної довжини.

Таблиця 1

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

	Дані	Код	
$i = 1$	11	00x	
	10	010	
	01	10x	
$i = 2$	0011	000	00x
	0010	000	010
	0001	100	00x
	0000	100	010

Символ x , що використовується в таблиці перетворення, має значення "1" для наступного біта каналу "0" або має значення "00" для наступного біта каналу "1". Максимальна довжина кодового обмеження g дорівнює 2.

Параметри коду змінної довжини RLL(1 - 7) дорівнюють (1, 7; 2, 3; 2). Мінімальний період інверсії T_{\min} , який може бути виражений як $(d + 1)T$, дорівнює, таким чином, $2(= 1 + 1)T$, де T - проміжок між бітами в послідовності сигналів запису. Мінімальний період інверсії T_{\min} , який також може бути виражений як $(m/n) \times 2T_{data}$, дорівнює, таким чином, $1,33(= 2/3 \times 2)T_{data}$, де T_{data} - це проміжок між бітами в рядку даних. Максимальний період інверсії T_{\max} , який також може бути виражений як $(k + 1)T$, дорівнює, таким чином, $(7 + 1)T = 8T = 8 \times (m/n)T_{data} = 8 \times 2/3 T_{data} = 5,33 T_{data}$. Ширина вікна визначення T_w , яка також може бути виражена як $(m/n)T_{data}$, дорівнює, таким чином, $0,67(= 2/3)T_{data}$.

До речі, в послідовності бітів каналу, що пройшли RLL - модуляцію (1 - 7), як показано в Таблиці 1, частота генерації, відповідна періоду $2T$, що дорівнює мініальному періоду інверсії T_{\min} , частіше за все перевищує частоту генерації, відповідним періодам $3T$ і $4T$. Той факт, що велика кількість граничної інформації генерується через короткі інтервали, такі, як $2T$ і $3T$, в багатьох випадках є вигідним для

генерації синхросигналів.

Проте, з подальшим зростанням лінійної густини запису, мінімальні серії створюють проблеми. Тобто якщо мінімальні серії 2Т генеруються послідовно, виникають спотворення послідовності сигналів запису. Це відбувається тому, що потужність сигналу 2Т менша, ніж потужність інших сигналів, і тому на цей сигнал сильно впливають такі чинники, як розфокусування і нахил дотичної. Крім того, при високій лінійній густині, на запис послідовних мінімальних міток (2Т) також сильно впливають перешкоди, такі як шум. Таким чином, при зчитуванні даних також можуть виникати помилки. У цьому випадку комбінація помилок при відтворенні даних виявляється в багатьох випадках як зсуви передніх і задніх меж мінімальних міток. У результаті довжина згенерованої помилки біта збільшується.

Як було описано вище, при передачі даних по лінії передачі або при записуванні на носій дані перед передачею або записом модулюються шляхом перетворення в код, відповідний лінії передачі або носієві. Якщо код, отриманий після модуляції, містить постійну складову, то виникають різні сигнали помилок, такі, як помилки стеження, що генеруються для управління системою стеження дисководом, схильною до коливань або "тремтіння". З цієї причини бажано вжити всі можливі заходи для запобігання появі постійної складової в модульованому коді.

Для запобігання появі постійної складової в модульованому коді запропонований контроль DSV (цифрового значення суми). DSV - це контрольна сума, яка обчислюється складанням значень послідовності бітів (символів даних), де значення +1 і -1 присвоюються, відповідно, значенням бітів "1" і "0", що отримується після NRZI - модуляції (тобто кодування по рівню) послідовності бітів каналу. DSV є індикатором постійної складової, що міститься в послідовності бітів. Зменшення абсолютного значення DSV за допомогою контролю DSV еквівалентно пригніченню величини постійної складової, що міститься в послідовності кодів.

Контроль DSV не застосовується для модуляції коду, що генерується відповідно до таблиці RLL - коду (1 - 7) змінної довжини, такої, як приведена вище Таблиця 1. Контроль DSV в такому випадку виконується обчисленням DSV послідовності кодованих бітів (послідовності бітів каналу) після модуляції протягом заданого періоду часу і вставкою заданої кількості контрольних бітів DSV в послідовність кодованих бітів (послідовність бітів каналу).

У будь-якому випадку, контрольні біти DSV є, по суті, надлишковими розрядами. Якщо треба враховувати ефективність перетворення коду, то бажано скоротити кількість контрольних бітів DSV до найменшого можливого значення.

Крім того, якщо вставляються контрольні біти DSV, бажано також зберігати незмінними мінімальну серію d і максимальну серію k . Це пояснюється тим, що зміна (d , k) впливає на характеристики запису/читання.

Як було описано вище, при записуванні RLL - коду з високою лінійною густиною або при зчитуванні RLL - коду, записаного з високою лінійною густиною, виникає проблема, що полягає в тому, що комбінація послідовних мінімальних серій d часто викликає генерацію помилки великої тривалості.

Крім того, у випадку RLL - коду, такого, як RLL - код (1 - 7), контроль DSV вимагає вставки контрольних бітів DSV в довільну частину рядка кодових слів (послідовності бітів каналу). Оскільки контрольні біти DSV є, по-суті, надлишковими розрядами, бажано скоротити кількість контрольних бітів DSV до найменшого можливого значення. Проте, для збереження мінімальної і максимальної серій постійними, кількість контрольних бітів DSV повинна бути не менше за 2. Таким чином, бажано скоротити кількість контрольних бітів DSV навіть до ще меншого значення.

Даний винахід вирішує описані вище проблеми. Задача даного винаходу - уможливити виконання контролю DSV з генерацією високоефективних контрольних бітів на RLL - коді (d , k ; m , n), де мінімальна серія $d = 1$, тобто на RLL - коді (1 , 7 ; 2 , 3), таким чином, щоб кількість послідовних мінімальних серій скорочувалася, в той час як мінімальна серія і максимальна серія зберігалися.

Ще однією задачею даного винаходу є запобігання наростанню поширення помилки демодуляції шляхом використання таблиці перетворення, що має найпростішу з можливих конфігурацій.

Пристрій модуляції, згідно з пунктом 1, відрізняється наявністю засобу перетворення вхідних даних в код відповідно до таблиці перетворення, де вказана таблиця перетворення задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

основні коди для $d = 1$, $k = 7$, $m = 2$ і $n = 3$, де d - мінімальна серія і k - обмеження довжини серії;
коди першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d ; і
коди другої підстановки для збереження вказаного обмеження довжини серії k .

Спосіб модуляції, згідно з пунктом 25, відрізняється включенням кроку перетворення вхідних даних в код відповідно до таблиці перетворення, де вказана таблиця перетворення задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

основні коди для $d = 1$, $k = 7$, $m = 2$ і $n = 3$, де d - мінімальна серія і k - обмеження довжини серії;
коди першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d ; і
коди другої підстановки для збереження вказаного обмеження довжини серії k .

Програмоносій, згідно з пунктом 26, для представлення програми, що реалізовує обробку, що включає

крок перетворення вхідних даних в код відповідно до даних таблиці перетворення в пристрої модуляції для перетворення даних з довжиною основних даних m біт в код змінної довжини $(d, k; m, n; r)$ з довжиною основного коду n біт, відрізняється тим, що вказана таблиця перетворення задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

- основні коди для $d = 1, k = 7, m = 2 \text{ і } n = 3$, де d - мінімальна серія і k - обмеження довжини серії;
- коди першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d ; і
- коди другої підстановки для збереження вказаного обмеження довжини серії k .

Пристрій демодуляції, згідно з пунктом 27, відрізняється наявністю засобу перетворення вхідного коду в дані відповідно до таблиці перетворення, де вказана таблиця задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

- основні коди для $d = 1, k = 7, m = 2 \text{ і } n = 3$, де d - мінімальна серія і k - обмеження довжини серії;
- коди першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d ; і
- коди другої підстановки для збереження вказаного обмеження довжини серії k .

Спосіб демодуляції, згідно з пунктом 30, відрізняється наявністю кроку перетворення вхідного коду в дані відповідно до таблиці перетворення, де вказана таблиця перетворення задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

- основні коди для $d = 1, k = 7, m = 2 \text{ і } n = 3$, де d - мінімальна серія і k - обмеження довжини серії;
- коди першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d ; і
- коди другої підстановки для збереження вказаного обмеження довжини серії k .

Програмоносій, згідно з пунктом 31, для представлення програми, що включає крок перетворення вхідного коду в дані відповідно до таблиці перетворення в пристрої демодуляції для перетворення коду змінної довжини $(d, k; m, n; r)$ з довжиною основного коду n біт в дані з довжиною основних даних m біт, відрізняється тим, що вказана таблиця перетворення задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

- основні коди для $d = 1, k = 7, m = 2 \text{ і } n = 3$, де d - мінімальна серія і k - обмеження довжини серії;
- коди першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d ; і
- коди другої підстановки для збереження вказаного обмеження довжини серії k .

Відповідно до опису пристрою модуляції в пункті 1, опису способу модуляції в пункті 25, опису програмоносія в пункті 26, опису пристрою демодуляції в пункті 27, опису способу демодуляції в пункті 30 і опису програмоносія в пункті 31, перетворення виконується на основі таблиці перетворення, що задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

- основні коди для $d = 1, k = 7, m = 2 \text{ і } n = 3$;
- коди першої підстановки для обмеження кількості послідовних появ мінімальної серії d ; і
- коди другої підстановки для збереження вказаного обмеження довжини серії k .

Перед початком пояснення деяких переважних варіантів здійснення даного винаходу, для того, щоб прояснити відносини, зв'язуючі засоби даного винаходу, описані в пунктах формули винаходу, з реалізаціями, прийнятими у варіантах здійснення, в наступному описі, що описує даний винахід, кожний з засобів супроводжується типовою реалізацією, взятою в дужки, у вигляді "засіб(що реалізовується, наприклад, типовою реалізацією)". Проте типова реалізація не повинна тлумачитися в обмеженому значенні. Тобто, засіб не обов'язково обмежується типовою реалізацією, пов'язаною з даним засобом.

Пристрій модуляції, згідно з пунктом 1, відрізняється наявністю засобу перетворення(що реалізовується, наприклад, блоком модуляції, показаним на фіг. 12) для перетворення вхідних даних в код відповідно до таблиці перетворення(що реалізовується, наприклад, Таблицею 2), де вказана таблиця перетворення задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

- основні коди для $d = 1, k = 7, m = 2 \text{ і } n = 3$, де d - мінімальна серія і k - обмеження довжини серії;
- коди першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d ; і
- коди другої підстановки для збереження вказаного обмеження довжини серії k .

Згідно з пунктом 12, пристрій модуляції, заявлений як пункт 1, відрізняється, крім того, наявністю засобу вставки сигналу синхронізації(що реалізовується, наприклад, блоком 212 вставки сигналів синхронізації, показаним на фіг. 9) для вставки сигналу синхронізації, що містить унікальну комбінацію, що не входить у вказані коди перетворення вказаної таблиці перетворення, в будь-яку довільну позицію

вказаного рядка кодових слів.

Згідно з пунктом 23, пристрій модуляції, заявлений як пункт 1, відрізняється, крім того, наявністю засобу контролю DSV(що реалізовується, наприклад, блоком 11 визначення/вставки контрольних бітів DSV, показаним на фіг. 1) для контролю значень DSV вхідних даних і видачі вказаних значень DSV на вказаний засіб перетворення.

Згідно з пунктом 24, пристрій модуляції, заявлений як пункт 1, відрізняється, крім того, тим, що вказаний засіб перетворення включає:

засіб виявлення кодів першої підстановки(що реалізовується, наприклад, блоком 33 виявлення коду обмеження послідовних появ мінімальних серій, показаним на фіг. 3) для виявлення вказаних кодів першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d; і

засіб виявлення кодів другої підстановки(що реалізовується, наприклад, засобом 34 виявлення коду гарантування максимальної серії, показаним на фіг. 3) для виявлення вказаних кодів другої підстановки для збереження обмеження довжини серії.

Згідно з пунктом 27, пристрій демодуляції відрізняється наявністю засобу перетворення(що реалізовується, наприклад, блоком демодуляції 111, показаним на фіг. 5) для перетворення вхідного коду в дані відповідно до таблиці перетворення, де вказана таблиця перетворення(що реалізовується, наприклад, Таблицею 2) задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення вказаного рядка даних, і коди перетворення вказаної таблиці включають:

основні коди для $d = 1$, $k = 7$, $m = 2$ і $p = 3$, де d - мінімальна серія і k - обмеження довжини серії;

коди першої підстановки для обмеження кількості послідовних появ вказаної мінімальної серії d; і

коди другої підстановки для збереження вказаного обмеження довжини серії k.

Згідно з пунктом 28, пристрій демодуляції, заявлений як пункт 27, відрізняється, крім того, наявністю засобу видалення бітів(що реалізовується, наприклад, блоком 112 видалення контрольних бітів DSV, показаним на фіг. 5) для видалення надлишкових розрядів, що вставляються через задані інтервали часу у вказаний код.

Переважні варіанти здійснення описуються нижче. Для полегшення розуміння в нижчевикладеному описі масив бітів '0' і '1' даних до перетворення, тобто рядок даних, що підлягає перетворенню, визначається як послідовність бітів, взята в дужки "(", як, наприклад, (000011). З іншого боку, масив бітів '0' і '1' коду, що отримується внаслідок перетворення, тобто рядок слів після перетворення, означається як послідовність бітів, відділених парою символів ", як, наприклад, "000100100". Таблиці 2 і 3, приведені нижче, є прикладами таблиць перетворення даних в код відповідно до даного винаходу.

Таблиця 2

17PP RML.32

Дані	Код
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(наступні 010)
00001000	000 100 100 100
00000000	010 100 100 100

Якщо $xx1$ тоді $*0* = 000$, $xx0$ тоді $*0* = 101$

Синхронізація(Sync) і завершення

#01 000 000 001(12 бітів каналу) або

#01 001 000 000 001 000 000 001(24 біта каналу)

= 0: Відсутність завершення

= 1: Завершення

00

000

0000

010 100

"110111 001 000 000(наступні 010)

Коли наступними бітами каналу є '010', перетворити '11 01 11' в '001 000 000' після використання основної таблиці і таблиці завершення.

Як показано в Таблиці 2, таблиця перетворення містить коди, що отримуються внаслідок перетворення, в тому числі основні коди, коди підстановки і коди завершення. Виконання перетворення не

може відбуватися без основного коду. У таблиці перетворення основними кодами є коди, що отримуються внаслідок перетворення рядків даних з(11) до(000000). Перетворення може виконуватися навіть якщо код підстановки не існує. Проте, якщо код підстановки існує, перетворення може бути виконано більш ефективно. У таблиці перетворення кодами підстановки є коди, що отримуються внаслідок перетворення рядків даних(100111), (00001000) і (00000000). Код завершення використовується для завершення коду, що отримується внаслідок перетворення в будь-якій довільній позиції. Кодами завершення в таблиці є коди, що отримуються внаслідок перетворення рядків даних(00) і (0000). Крім того, таблиця перетворення визначає також сигнали синхронізації.

У Таблиці 2 мінімальна серія дорівнює 1, тоді як максимальна серія дорівнює 7. Одним з елементів основних кодів є невизначений код, тобто код, позначений символом '*'. Біт, представлений символом '*' невизначеного коду, може визначатися як "0" або "1" для збереження значень мінімальної серії d і максимальної серії k безвідносно безпосередньо попереднього або наступного рядка кодів слів. Більш детально: якщо 2 - бітовий рядок даних для перетворення дорівнює(11), код, що отримується внаслідок перетворення, може бути "000" або "101" в залежності від безпосередньо попереднього рядка. Точніше, якщо єдиний біт каналу безпосередньо попереднього рядка кодів слів дорівнює "1", 2 - бітовий рядок даних(11) перетворюється в код "000" для збереження значення мінімальної серії d. З іншого боку, якщо єдиний біт каналу безпосередньо попереднього рядка кодів слів дорівнює "0", 2 - бітовий рядок даних(11) перетворюється в код "101" для збереження максимальної серії k.

Основні коди, показані в таблиці перетворення Таблиці 2, мають структуру змінної довжини. Кількість основних кодів з довжиною кодового обмеження і, що дорівнює 1, дорівнює 3 - значенню, яке менше, ніж необхідна кількість, що дорівнює $4 (= 2^m = 2^2)$. Цими 3 основними кодами є "0*", "001" і "010". У результаті, в операції перетворення рядка даних зустрічається рядок даних, який не може бути перетворений тільки з довжиною кодового обмеження і, що дорівнює 1. З цієї причини для перетворення всіх рядків даних необхідно звертатися до основних кодів з довжиною кодового обмеження і аж до 3 в Таблиці 2. Тобто основні коди з довжиною кодового обмеження і аж до 3 включені в Таблицю 2 для того, щоб Таблиця 2 могла служити доконаною таблицею перетворення.

Крім того, таблиця перетворення Таблиці 2 також включає коди підстановки для обмеження послідовних появ мінімальної серії d. Якщо рядок даних дорівнює(110111) і рядок кодів слів, наступний за кодом, що отримується внаслідок перетворення рядка даних, дорівнює "010", рядок даних перетворюється в кодове слово "010 000 000". З іншого боку, якщо рядок кодів слів, наступний за кодом, що отримується внаслідок перетворення рядка даних, не дорівнює "010", рядок даних(110111) перетворюється 2 - бітовими блоками. Більш детально: 2 - бітові послідовності(11), (01) і (11) рядки даних перетворюються в рядок кодів слів "0* 010 і *0*". У результаті, послідовні появи мінімальної серії d в рядку кодів слів, що отримується внаслідок перетворення рядка даних, можуть бути обмежені, при цьому кількість мінімальних серій, що повторюються обмежується до максимального значення, що дорівнює 6.

Таблиця перетворення Таблиці 2 задає також правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних. Тобто якщо кількість "1" елемента рядка даних парна, кількість "1" елемента рядка кодів слів також парна, і, з іншого боку, якщо кількість "1" елемента рядка даних непарна, кількість "1" елемента рядка кодів слів також непарна. Наприклад, рядок даних(000001) перетворюється в рядок кодів слів "010 100 000". У цьому випадку залишок від ділення на 2 кількості "1" елемента рядка даних дорівнює 1, що дорівнює залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних. Тобто кількість "1" як рядка даних, так і рядка кодів слів непарна. Як інший приклад рядок даних(000000) перетворюється в рядок кодів слів "010 100 100". У цьому випадку залишок від ділення на 2 кількості "1" елемента рядка даних дорівнює 0, що дорівнює залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних. Тобто кількість "1" як рядка даних, так і рядка кодів слів парна.

Крім того, максимальна довжина кодового обмеження g таблиці перетворення Таблиці 2 дорівнює 4. Коди в таблиці з довжиною кодового обмеження і, що дорівнює 4, є кодами підстановки для реалізації значення 7 максимальної серії k. Такий код підстановки називається кодом гарантування максимальної серії. Тобто рядок даних(00001000) перетворюється в рядок кодів слів "000 100 100 100", тоді як рядок даних(00000000) перетворюється в рядок кодів слів "010 100 100 100". Потрібно зазначити, що в цьому випадку значення мінімальної серії d також буде дорівнювати 1.

Якщо таблиця перетворення Таблиці 2 не містить кодів підстановки, що мають довжину кодового обмеження, що дорівнює 4, максимальна довжина кодового обмеження g для таблиці дорівнює 3, при цьому генерується код з максимальною серією k, що дорівнює 8. Проте оскільки таблиця включає основні коди з довжиною кодового обмеження і, що дорівнює 4, може бути згенерований код з максимальною серією k, що дорівнює 7.

Взагалі, чим більше максимальна серія k, тим складніша генерація синхросигналів і, таким чином, гірша стабільність системи. Тому, шляхом зменшення значення максимальної серії k з 8 до 7, відповідно можуть бути поліпшені характеристики системи із зменшенням максимальної серії k.

Тобто якщо таблиця перетворення Таблиці 2 виконана таким чином, що містить тільки основні коди, максимальна довжина кодового обмеження g для такої таблиці дорівнює 3. У цьому випадку можлива генерація коду, що має мінімальну серію d, що дорівнює 1, і максимальну серію k, що дорівнює 8. Крім того,

залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних.

Якщо таблиця перетворення створена таким чином, щоб містити також коди підстановки для обмеження послідовних появ мінімальної серії d в доповнення до основних кодів, максимальна довжина кодового обмеження g для такої таблиці також дорівнює 3. У цьому випадку, проте, можлива генерація коду, що має мінімальну серію d , що дорівнює 1, і максимальну серію k , що дорівнює 8, тоді як кількість послідовних мінімальних серій d обмежена значенням верхньої межі. Крім того, залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних.

Якщо таблиця перетворення створюється створена таким чином, щоб містити також коди підстановки для гарантування максимальної серії k , що дорівнює 7, в доповнення до кодів підстановки для обмеження послідовних появ мінімальної серії d і основних кодів, максимальна довжина кодового обмеження g для такої таблиці дорівнює 4. У цьому випадку можлива генерація коду, що має мінімальну серію d , що дорівнює 1, і максимальну серію k , що дорівнює 7, тоді як кількість послідовних мінімальних серій d обмежена значенням верхньої межі. Крім того, залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних.

Проте в загальному випадку, чим більша максимальна довжина кодового обмеження g , тим гірша характеристика поширення помилки демодуляції, що генерується у випадку зсуву розряду, тобто помилки, що генерується через зсув граничного біту уперед або назад на 1 розряд від його нормальної позиції.

Порівняння Таблиці 1 і Таблиці 2 показує, що максимальна довжина кодового обмеження g в першій з них дорівнює 2, тоді як у другій вона дорівнює 4. Тому Таблиця 2 повинна забезпечувати гірші характеристики, ніж Таблиця 1. Проте результати моделювання, які будуть описані нижче, з посиланням на Таблицю 7, показують, що характеристики Таблиці 2 ненабагато гірше в порівнянні з Таблицею 1. Наприклад, як показано в Таблиці 7, середня частота появи помилок для Таблиці 1 становить 1,014 байт, в той час як для Таблиці 2 вона становить 1,67 байт - значення, яке є ненабагато більшим значення для Таблиці 1. Відмінність середньої частоти появи помилок може бути пояснена тим фактом, що кількість кодових груп в Таблиці 1 на 2 менше, ніж в Таблиці 2.

До речі, для випадку вставки сигналу синхронізації в будь-яку довільну позицію рядка кодів слів(тобто послідовності бітів каналу), що генерується внаслідок перетворення, що виконується відповідно до таблиці перетворення Таблиці 2, таблиця перетворення видає код зі структурою змінної довжини. Це відбувається тому, що таблиця перетворення включає таблицю завершення, що визначає коди завершення для завершення коду, що отримується внаслідок перетворення в будь-якій довільній позиції. Код завершення використовується по мірі необхідності.

Передбачимо, наприклад, що сигнал синхронізації вставляється в певну позицію в коді, що отримується внаслідок перетворення. У цьому випадку, передусім в точці переходу між рядком кодів слів, безпосередньо попередній даній позиції, і рядком кодів слів, наступним безпосередньо за даною позицією, встановлюються біти переходу, в той час як мінімальна серія d і максимальна серія k зберігаються, а між бітами переходу встановлюється унікальна комбінація, що представляє сигнал синхронізації. Розглянемо комбінацію сигналу синхронізації, що порушує значення 7 максимальної серії k . У цьому випадку комбінація сигналу синхронізації мінімальної довжини є 12 - бітовим кодовим словом(тобто 12 бітів каналу), що задається таким чином:

"#01 000 000 001"

Символ "#" на початку комбінації сигналу синхронізації є бітом переходу, що приймає значення "0" або "1", який буде описаний нижче. Другий біт каналу, наступний за бітом "#", дорівнює "0" для збереження мінімальної серії d . Третій біт каналу і наступні біти встановлюються в значення, що формують унікальну комбінацію 9Т, кодову комбінацію, не передбачену в Таблиці 2, для забезпечення максимальної серії k , що дорівнює 8. Як показано в приведеній вище комбінації, між третім бітом каналу і останнім бітом каналу знаходяться 8 послідовних бітів "0". Потрібно зазначити, що навіть якщо останній біт каналу в комбінації сигналу синхронізації встановлюється у "1", використанням таблиці перетворення Таблиці 2 мінімальна серія d може бути збережена.

Далі пояснюються таблиця завершення і біт переходу "#" в бітовій комбінації сигналу синхронізації. Як показано в Таблиці 2, таблиця завершення будується таким чином:

00	000	
0000	010	100

Таблиця завершення необхідна для основних кодів з довжиною кодового обмеження i , що забезпечує кількість пар, кожна з яких складається з рядка даних і рядка кодів слів, менше, ніж необхідна кількість $4(= 2^m = 2^2)$

Більш детально: у випадку Таблиці 2, для довжини кодового обмеження $i = 1$, оскільки кількість пар, кожна з яких складається з рядка даних і рядка кодів слів, дорівнює 3, необхідна таблиця завершення. Для довжини кодового обмеження $i = 2$, оскільки кількість пар, кожна з яких складається з рядка даних і рядка кодів слів, дорівнює також 3, необхідна таблиця завершення. Для довжини кодового обмеження $i = 3$, кількість пар, кожна з яких складається з рядка даних і рядка кодів слів, дорівнює 5, включаючи пару, що має код підстановки. Інші - це 4 пари, кожна з яких містить основний код. Оскільки необхідна кількість, що дорівнює 4, є наявною, таблиця завершення не потрібна. Для довжини кодового обмеження i

= 4, оскільки всі рядки кодових слів є кодами підстановки, немає необхідності брати до уваги код завершення. Тому таблиця завершення необхідна для довжини кодового обмеження $i = 1$, коли код завершення використовується для рядка даних (00). Аналогічно, таблиця завершення необхідна для довжини кодового обмеження $i = 2$, коли код завершення використовується для рядка даних (0000). Згідно з таблицею завершення, рядки даних (00) і (0000) перетворюються в рядки кодових слів "000" і "010100" відповідно. У результаті в операції вставки сигналу синхронізації можна уникнути ситуації, в якій дані, попередні комбінації сигналу синхронізації, більше не можуть бути перетворені. Тобто система перетворення може виключити ситуації, в яких код, безпосередньо попередній сигналу синхронізації, більше не може служити кодом завершення.

Біт "#" комбінації сигналу синхронізації використовується для розрізнення випадку використання таблиці завершення і випадку, в якому таблиця завершення не використовується. Точніше, перший біт каналу "#" на початку комбінації сигналу синхронізації встановлюється в '1' для позначення того, що код завершення використовується, або встановлюється в '0' для позначення того, що код завершення не використовується. Таким чином, можна правильно визначити, використовується таблиця завершення чи ні, тобто використовується код завершення чи ні.

Як було описано вище, комбінація сигналу синхронізації мінімальної довжини є 12 - бітовим кодовим словом(тобто 12 бітів каналу). Оскільки комбінація сигналу синхронізації, що порушує значення 7 максимальної серії k, але така, що забезпечує максимальну серію k, що дорівнює 8(9T) є прийнятною, може бути побудована будь-яка інша комбінація сигналу синхронізації, що формує кодове слово довжиною не менше за 12 біт. У разі формування 15 - бітового кодового слова, можуть бути згенеровані, наприклад, такі сигнали синхронізації:

"#01 000 000 001 010"

"#01 000 000 001 001"

У разі формування 21 - бітового кодового слова, може бути згенерований, наприклад, такий сигнал синхронізації:

"#01 000 000 001 000 000 001"

Вищенаведений 21 - бітовий сигнал синхронізації включає 2 послідовних комбінації, кожна з яких забезпечує максимальну серію k, що дорівнює 8(9T). Такий сигнал синхронізації може бути виявлений з більш високим ступенем надійності. Тоді у разі формування 24 - бітового кодового слова може бути згенерований такий сигнал синхронізації:

"#01 001 000 000 001 000 000 001"

Вищенаведений сигнал синхронізації - комбінація, що має форму "3T - 9T - 9T" - зменшує імовірність того, що довга серія(T) з'явиться до і/або після двох послідовних комбінацій, що забезпечують максимальну серію k, що дорівнює 8(9T), а також збільшує здатність виявлення. Можна вибрати, яку здатність виявлення повинен забезпечувати сигнал синхронізації відповідно до вимог до системи.

Таблиця 3 є ще однією типовою таблицею, що передбачається даним винаходом.

Таблиця 3

17PP. RML.52

i = 1 Основна таблиця:	
Дані	Код
00	101
01	100
10	001
11	000
i = 2 Таблиця підстановки A (d обмежене до 1):	
0000	100 010
0001	101 010
1000	000 010
1001	001 010
i = 3 Таблиця підстановки B (k обмежене до 8):	
111111	000 010 010
111110	001 010 010
011110	101 010 010
011111	100 010 010
i = 4 Таблиця підстановки C (RMTR (кількість повторень мінімальної серії) обмежене до 6):	

	00010001	100 010 010 010
канал – 0	10010001	100 000 010 010
канал - 1	10010001	000 010 010 010
i = 4 Таблиця підстановки D (k обмежене до 7):		
канал 010	11100000	000 001 010 010
канал 010	11100010	100 001 010 010
канал 010	11100001	001 010 010 010
канал 010	11100011	101 010010010

Синхронізація(Sync)

дані: x1 - 0x

кан.: xx0 100 000 000 10x (12 бітів каналу)

дані: x1 - 0x

кан.: xx0 100 000 000 100 000 000 10x (24 біта каналу)

Завершення:

Додати біти даних '01' або '11' в початок, i '00' або '01' в кінець

Таблиця перетворення Таблиці 3 має структуру, в якій для мінімальної серії d = 1, максимальній серії k = 7 і довжини кодового обмеження i = 1, передбачаються 4 (= 2^m = 2²) основних кодів. Тобто для довжини кодового обмеження i = 1 в основній таблиці знаходяться 4 основних кодів. Для довжини кодового обмеження i = 2 або більше передбачаються таблиці кодів підстановки для обмеження параметрів, таких, як мінімальна серія d і максимальна серія k. Точніше, Таблиця A для довжини кодового обмеження i = 2 передбачає коди підстановки для обмеження мінімальної серії d до 1. Таблиця B для довжини кодового обмеження i = 3 передбачає коди підстановки для обмеження максимальної серії k до верхньої межі, що дорівнює 8. Таблиця C для довжини кодового обмеження i = 4 передбачає коди підстановки для обмеження послідовних появ мінімальної серії d, що має значення 1. Таблиця D для довжини кодового обмеження i = 4 передбачає коди підстановки для обмеження максимальної серії k до верхньої межі, що дорівнює 7. Таким чином, в таблиці перетворення Таблиці 3 межа максимального обмеження g = 4.

Як було описано вище, таблиця перетворення Таблиці 3 включає коди підстановки для обмеження послідовних появ мінімальної серії d. Наприклад, рядок даних (0001001) перетворюється в рядок кодових слів "100 010 010 010". А для рядка даних (10010001) перевіряється безпосередньо попереднє кодове слово для визначення того, чи є безпосередньо попередній біт каналу "0" або "1". Якщо безпосередньо попередній біт каналу дорівнює "0", рядок даних перетворюється в рядок кодових слів "100 000 010 010". З іншого боку, якщо безпосередньо попередній біт каналу дорівнює "1", рядок даних перетворюється в рядок кодових слів "000 010 010 010". У результаті рядок кодових слів, що отримується внаслідок перетворення даних, має кількість мінімальних серій, що послідовно повторюються, обмежену до максимального значення, що дорівнює 6.

Крім того, таблиця перетворення Таблиці 3 задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення цього рядка даних. Тобто якщо кількість "1" елемента рядка даних парна, кількість "1" елемента рядка кодових слів також парна, і, з іншого боку, якщо кількість "1" елемента рядка даних непарна, кількість "1" елемента рядка кодових слів також непарна. Наприклад, рядок даних (1000) перетворюється в рядок кодових слів "000 010". У цьому випадку залишок від ділення на 2 кількості "1" елемента рядка даних дорівнює 1, що дорівнює залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення цього рядка даних. Тобто кількість "1" як рядка даних, так і рядка кодових слів непарна. Як інший приклад рядок даних (11111) перетворюється в рядок кодових слів "000 010 010". У цьому випадку залишок від ділення на 2 кількості "1" елемента рядка даних дорівнює 0, що дорівнює залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення цього рядка даних. Тобто кількість "1" як рядка даних, так і рядка кодових слів парна.

Крім того, коди в таблиці перетворення Таблиці 3 з довжиною кодового обмеження i = 4, що дорівнює максимальній довжині кодового обмеження g, є кодами підстановки для реалізації значення 7 максимальної серії k. У разі перетворення з використанням коду підстановки перевіряється безпосередньо попередній рядок кодових слів. Точніше, якщо безпосередньо попередній рядок кодових слів дорівнює "010", то здійснюється перетворення. Якщо рядок даних дорівнює (11100000) і безпосередньо попередній рядок кодових слів дорівнює, наприклад, "010", то здійснюється перетворення, результатом якого є рядок кодових слів "000 001 010 010". Як інший приклад, якщо рядок даних дорівнює (11100010) і безпосередньо попередній рядок кодових слів дорівнює "010", то рядок даних перетворюється в рядок кодових слів "100 001 010 010".

Вищенаведена таблиця перетворення Таблиці 3 для реалізації RLL - кодування не може бути побудована тільки з основних кодів. RLL - код з гарантованою мінімальною серією d і гарантованою максимальною серією k може бути отриманий з використанням основних кодів в основній таблиці, а також кодів підстановки в Таблиці A для довжини кодового обмеження i, що дорівнює 2, і в Таблиці B для довжини кодового обмеження i, що дорівнює 3. У цьому випадку максимальна довжина кодового обмеження g дорівнює 3, і можлива генерація коду, що має мінімальну серію d, що дорівнює 1, і

максимальну серію k , що дорівнює 8. Крім того, залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних.

Якщо Таблиця С, що передбачає коди підстановки для обмеження послідовних появ мінімальної серії d , включена в конфігурацію таблиці перетворення Таблиці 3 в доповнення до основної таблиці і Таблиць А і В, то максимальна довжина кодового обмеження g дорівнює 4, і можлива генерація коду, що має мінімальну серію d , що дорівнює 1, максимальну серію k , що дорівнює 8, і обмежену кількість послідовних появ мінімальних серій d . Крім того, залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних. Потрібно зазначити, що в цьому випадку не завжди необхідно перевіряти безпосередньо попередній рядок кодів слів, як у разі використання Таблиці С Таблиці 3.

Якщо Таблиця D, що передбачає коди підстановки для гарантування максимальної серії k , що дорівнює 7, включена в конфігурацію Таблиці 3 в доповнення до основної таблиці і Таблиці А, Таблиці В і Таблиці С, що передбачає коди підстановки для обмеження послідовних появ мінімальної серії d , то максимальна довжина кодового обмеження g дорівнює 4, і можлива генерація коду, що має мінімальну серію d , що дорівнює 1, максимальну серію k , що дорівнює 7, і обмежену кількість послідовних появ мінімальних серій d . Крім того, залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних.

Таблиця завершення Таблиці 2 не потрібна для обробки вставки сигналу синхронізації в певну позицію в рядку кодів слів(або послідовність бітів каналу), що отримується внаслідок перетворення з використанням таблиці перетворення Таблиці 3. Це пояснюється тим, що згідно з таблицею перетворення Таблиці 3, код може завершуватися в позиції, відповідній довжині кодового обмеження i , що дорівнює 1.

Для того, щоб вставка комбінації сигналу синхронізації забезпечувала як можна більш ефективний код, дана комбінація визначається таким чином. Як 3 - бітове кодове слово, безпосередньо попереднє унікальній комбінації(або рядку кодів слів), що служить сигналом синхронізації, так і 3 - бітове кодове слово, безпосередньо наступне за унікальною комбінацією, мають формат, в якому біти даних і біти переходу комбінуються таким чином.

Передусім, 3 біти безпосередньо попереднього кодового слова визначаються таким чином. Перевіряється m - бітовий блок слова даних до перетворення, де $m = 2$. Перший біт 2 - бітового блоку на початку слова даних до перетворення є інформаційним бітом, тоді як другий біт має значення Т для позначення сигналу синхронізації. 2 - бітовий блок на початку слова даних перетворюється в кодове слово(біти каналу) відповідно до Таблиці 3. Більш конкретно: m бітів(2 біти) слова даних(x_1) перетворюються в n бітів(3 біти) кодового слова "xx0".

Тоді 3 біти безпосередньо наступного кодового слова визначаються таким чином. Аналогічно, перевіряється m - бітовий блок слова даних до перетворення, де $m = 2$. У цьому випадку, проте, перший біт 2 - бітового блоку на початку слова даних до перетворення має значення '0' для позначення сигналу синхронізації, тоді як другий біт є інформаційним бітом. 2 - бітовий блок на початку слова даних перетворюється в кодове слово(біти каналу) відповідно до Таблиці 3. Більш конкретно: m бітів(2 біти) слова даних(0x) перетворюються в n бітів(3 біти) кодового слова "10x".

Коли унікальна комбінація сигналу синхронізації задається як комбінація, що порушує значення 7 максимальної серії k , комбінація сигналу синхронізації з найменшою можливою довжиною, яка може бути реалізована, містить кодове слово унікальної комбінації довжиною принаймні 12 бітів(12 бітів каналу), що задається таким чином:

"xx0 100 000 000 10x"

де значення "x" залежить від таблиці перетворення. Приведене вище кодове слово включає 3 біти "x". 2 біти "x" знаходяться на початку кодового слова, тоді як 1 біт "x" знаходиться в його кінці. 3 біти "x" представляють 2 - бітовий блок слова даних до перетворення. Інші 12 бітів каналу кодового слова є надлишковою частиною, що фактично представляє комбінацію сигналу синхронізації. Третій біт каналу кодового слова встановлюється в "0" для збереження мінімальної серії d . Що стосується інших бітів, починаючи з четвертого біту каналу, то 9Т встановлюється як комбінація сигналу синхронізації для забезпечення максимальної серії k , що дорівнює 8. Більш детально: 8 "0" вставляються послідовно між "1" і "1".

Як було описано вище, унікальна комбінація сигналу синхронізації з мінімальною довжиною - це 12 - бітове кодове слово(тобто 12 бітів каналу). Оскільки прийнятною є комбінація сигналу синхронізації, що порушує значення 7 максимальної серії k , але така, що забезпечує максимальну серію k , що дорівнює 8(9Т), може бути побудована будь-яка інша комбінація сигналу синхронізації, що формує кодове слово унікальної комбінації довжиною не менше за 12 біт. У разі формування кодового слова унікальної послідовності довжиною 15 біт, може бути згенерований, наприклад, такий сигнал синхронізації:

"xx0 100 000 000 100 10x"

У разі формування кодового слова довжиною 21 біт може бути згенерований такий сигнал синхронізації:

"xx0 100 000 000 100 000 000 10x"

Вищенаведений сигнал синхронізації довжиною 21 біт включає 2 послідовні комбінації, кожна з яких забезпечує максимальну серію k , що дорівнює 8(9Т). Відповідно до сигналу синхронізації здатність виявлення може бути збільшена. Можна вибрати, яку здатність виявлення повинен забезпечувати сигнал

синхронізації відповідно до вимог до системи.

Подібно стандартному способу, після перетворення рядка даних з використанням таблиці перетворення, такої, як Таблиця 2 або 3, контроль DSV може виконуватися додаванням контрольних бітів DSV через задані інтервали в послідовність бітів каналу, що отримується внаслідок перетворення. Проте завдяки використанню зв'язку між рядком даних і рядком кодових слів, що отримується внаслідок перетворення, заснованого на Таблицях 2 і 3, контроль DSV може виконуватися навіть з ще більш високим рівнем ефективності.

Більш детально: правило перетворення виконується таким чином, що залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення цього рядка даних. Таким чином, вставка контрольних бітів DSV, що мають значення "1" для позначення інверсії і значення "0" для позначення відсутності інверсії, в послідовність бітів каналу еквівалентна вставці контрольних бітів DSV, що мають значення "1" для позначення інверсії і значення "0" для позначення відсутності інверсії, в послідовність бітів даних.

Розглянемо, наприклад, випадок, в якому 3 біти(001) даних перетворюються відповідно до Таблиці 2. Нехай контрольний біт DSV x , наступний за 3 бітами, знаходиться між бітами даних. У цьому випадку даними, що містять контрольний біт DSV, будуть (001- x), де x - це 1 контрольний біт, що має значення '0' або '1'. При $x = 0$ дані, що містять контрольний біт DSV, перетворюються відповідно до Таблиці 2 таким чином:

Рядок даних	Рядок кодових слів
0010	010 000

З іншого боку, при $x = 1$ дані, що містять контрольний біт DSV, перетворюються відповідно до Таблиці 2 таким чином:

Рядок даних	Рядок кодових слів
0011	010 100

При застосуванні кодування по рівню з використанням NRZI - модуляції до рядка кодових слів, що отримується внаслідок перетворення з використанням Таблиці 2, виходить такий рядок, закодований по рівню:

Рядок даних	Рядок кодових слів	Закодований кодовий рядок
0010	010000	011111
0011	010 100	011000

Як показано у вищевказаній таблиці, останні 3 біти першого закодованого кодового рядка є інвертованими бітами останніх 3 бітів другого закодованого кодового рядка. Вищевказані результати означають, що вибираючи (1) або (0) значенням контрольного біта DSV, контроль DSV може виконуватися всередині рядка даних.

Далі, розглянемо надлишковість, що вводиться контролем DSV. Виконання контролю DSV вставкою 1 контрольного біту DSV в рядок даних відповідає виконанню контролю DSV вставкою 1,5 контрольних бітів DSV в послідовність бітів каналу, де значення 1,5 є зворотною величиною коефіцієнта перетворення $m/n = 2/3$ Таблиць 2 і 3. Для виконання контролю DSV для таблиці RLL(1 - 7), такої, як Таблиця 1, необхідно застосовувати контроль в послідовності бітів каналу. У цьому випадку для збереження мінімальної серії d потрібно, принаймні, 2 біти каналу, що приводить до високої відносної надлишковості в порівнянні з контролем DSV, що застосовується до рядка даних, для Таблиць 2 і 3. Інакше кажучи, в даній системі ефективність контролю DSV може бути підвищена виконанням контролю DSV в рядку даних.

Далі, посилаючись на фіг. 1, пояснюється варіант здійснення, що реалізовує пристрій модуляції, що пропонується даним винаходом. У даному варіанті здійснення рядок даних перетворюється в код змінної довжини($d, k; m, n; g$) = (1, 7; 2, 3; 4) з використанням Таблиці 2.

Як показано на фіг. 1, пристрій модуляції 1 включає блок 11 визначення/вставки контрольних бітів DSV для визначення значення '1' або '0' контрольного біта DSV і для вставки контрольних бітів DSV через будь-які довільні інтервали в рядок даних, що подається на вхід, блок модуляції 12 для модуляції рядка даних з вставленими в неї бітами DSV, і блок NRZI - кодування 13 для перетворення виходу блоку модуляції 12 в послідовність сигналів запису. Крім того, пристрій модуляції 1 містить також блок управління синхронізацією 14 для генерації тактових сигналів і подачі цих сигналів на різні компоненти.

Фіг. 2 - це пояснювальна діаграма, що використовується для опису обробки, що виконується блоком 11 визначення/вставки контрольних бітів DSV. Як показано на фіг. 2, значення контрольних бітів DSV визначаються і контрольні біти DSV вставляються в рядок даних через будь-які довільні інтервали. Для вставки контрольного біту DSV в позицію між блоками даних DATA1 і DATA2 поступаючого рядка даних, наприклад, блок 11 визначення/вставки контрольних бітів DSV обчислює кумулятивну суму DSV для даних аж до DATA1. Контрольна сума DSV обчислюється виконанням наступних кроків:

- перетворенням DATA1 в послідовність бітів каналу;
- виконанням NRZI - модуляції послідовності бітів;
- присвоєнням значення +1 рівню H(високому) (1) і значення -1 рівню L(низькому) (0) результату NRZI - модуляції; і

- складанням значень, присвоєних рівням результату NRZI - модуляції.

Крім того, блок 11 визначення/вставки контрольних бітів DSV обчислює контрольну суму DSV для сегмента DATA2, наступного за DATA1. Нехай x_1 буде контрольним бітом, який треба вставити в позицію між блоками даних DATA1 і DATA2. Блок 11 визначення/вставки контрольних бітів DSV визначає значення контрольного біта DSV x_1 таким чином, щоб абсолютне значення суми значень DSV для DATA1, x_1 і

DATA2 наближалось до нуля.

Якщо контрольний біт DSV x1 встановлюється в (1), коди рівня сегмента DATA2, наступного за DATA1, інвертуються. З іншого боку, якщо контрольний біт DSV x1 встановлюється в (0), коди рівня сегмента DATA2, наступного за DATA1, не інвертуються. Це відбувається тому, що в кожному елементі таблиць перетворення Таблиць 2 і 3 залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних. Таким чином, вставка біту(1) в рядок даних повинна супроводитися вставкою "1" в рядок кодів слів, що отримується внаслідок перетворення рядка даних, що означає інверсію.

Після того, як значення контрольного біта DSV x1, показаного на фіг. 2, визначено, як було описано вище, між DATA2 і DATA3 вставляється контрольний біт DSV x2, що забезпечує заданий інтервал даних між x1 і x2 для здійснення контролю DSV таким самим способом. У цьому випадку кумулятивна сума DSV є сумою значень кумулятивних сум DSV для даних аж до DATA1, значення DSV для x1 і даних DSV для сегмента DATA2.

Як було описано вище, контрольні біти DSV вставляються в рядок даних заздалегідь перед тим, як рядок даних модулюється блоком модуляції 12 для генерації послідовності бітів каналу.

Фіг. 3 - це структурна схема, на якій показана типова конфігурація блоку модуляції 12. Як показано на фіг. 3, регістр зсуву 31 зсуває дані, що зберігаються в ньому, на 2 біти за один раз, при цьому його вихід подається на блок 32 визначення довжини кодового обмеження, блок 33 виявлення коду обмеження послідовних появ мінімальних серій, блок 34 виявлення коду гарантування обмеження довжини серії і всі блоки перетворення з 35 - 1 по 35 - 4. Регістр зсуву 31 видає стільки бітів, скільки потрібно для обробки, на кожний з компонентів з 32 по 35.

Блок 32 визначення довжини кодового обмеження визначає довжину кодового обмеження і даних і видає цю довжину і на мультиплексор 36. Коли блок 33 виявлення коду обмеження послідовних появ мінімальних серій виявляє спеціальне слово даних, що підлягає обмеженню послідовних появ мінімальної серії d, блок 33 визначення коду обмеження послідовних появ мінімальних серій видає сигнал виявлення цього слова(i = 3), що вказує довжину кодового обмеження і, блоку 32 визначення довжини кодового обмеження. У разі Таблиці 2, спеціальне слово даних дорівнює (110111). Крім того, коли блок 34 виявлення коду гарантування обмеження довжини серії виявляє спеціальне слово даних, що вимагає гарантування максимальної серії k, блок 34 виявлення коду гарантування обмеження довжини серії видає сигнал виявлення цього слова(1 = 4), що вказує довжину кодового обмеження і, блоку 32 визначення довжини кодового обмеження. У разі Таблиці 2, спеціальне слово даних дорівнює (00001000) або (00000000).

Коли спеціальне слово даних виявляється блоком 33 виявлення коду обмеження послідовних появ мінімальних серій або блоком 34 виявлення коду гарантування обмеження довжини серії, блок 32 визначення довжини кодового обмеження передає довжину кодового обмеження і спеціального слова даних далі на мультиплексор 36. У цей час блок 32 визначення довжини кодового обмеження може також визначити інше значення для довжини кодового обмеження сам. Однак в цьому випадку блок 32 визначення довжини кодового обмеження дозволяє довжині кодового обмеження, виданій блоком 33 виявлення коду обмеження послідовних появ мінімальних серій або блоком 34 виявлення коду гарантування обмеження довжини серії мати пріоритет над кодом, визначеним ним самим. Іншими словами, вибирається більша довжина кодового обмеження.

Кожний з блоків перетворення з 35 - 1 по 35 - 4 визначає, занесений чи ні основний код для даних, що подаються на ці блоки, в таблицю перетворення, вбудовану в ці блоки. Якщо визначається, що основний код занесений в таблицю, дані перетворюються в кодове слово, що представляється основним кодом, і кодове слово, що отримується внаслідок перетворення, подається на мультиплексор 36. З іншого боку, якщо основний код для даних не занесений в таблицю перетворення, блоки перетворення з 35 - 1 по 35 - 4 виключають ці дані.

Потрібно зазначити, що оскільки пристрій модуляції 12 призначений для роботи з такою таблицею перетворення, як Таблиця 2, кожний з блоків перетворення 35 - і призначений для виконання перетворення даних з довжиною кодового обмеження і аж до 4. Тобто кожний з блоків перетворення 35 - і призначений для виконання перетворення даних аж до максимальної довжини кодового обмеження g, що дорівнює 4.

Мультиплексор 36 вибирає код, що отримується внаслідок перетворення, виконаного одним з блоків перетворення 35 - і, відповідним довжині кодового обмеження і, виданій блоком 32 визначення довжини кодового обмеження. Вибраний код потім виводиться через буфер 37 у вигляді послідовних даних.

Синхронізація роботи кожного компонента керується тактовими сигналами, що генеруються блоком управління синхронізацією 14.

Далі описується робота варіанту здійснення.

Передусім, регістр зсуву 31 видає стільки бітів даних, скільки потрібно для обробки, такої, як визначення, 2 - бітовими блоками на блок 32 визначення довжини кодового обмеження, блок 33 виявлення коду обмеження послідовних появ мінімальних серій, блок 34 виявлення коду гарантування обмеження довжини серії і блоки перетворення з 35 - 1 по 35 - 4.

Обладнаний вбудованою таблицею перетворення, такою, як Таблиця 2, блок 32 визначення довжини кодового обмеження визначає значення довжини кодового обмеження і, звертаючись до таблиці перетворення, і видає значення на мультиплексор 36.

У блок 33 виявлення коду обмеження послідовних появ мінімальних серій вбудоване слово даних, яке повинно замінятися кодом підстановки для обмеження послідовних появ мінімальної серії d Таблиці 2 в

тому випадку, якщо наступне кодове слово дорівнює "010". У разі Таблиці 2, це слово даних дорівнює (110111). Коли дані, що вимагають обмеження послідовних появ мінімальної серії d, виявляються внаслідок звернення до цієї частини таблиці перетворення, блок 33 виявлення коду обмеження послідовних появ мінімальних серій видає сигнал виявлення, який вказує, що довжина кодового обмеження $i = 3$, блокові 32 визначення довжини кодового обмеження.

З іншого боку, в блок 34 виявлення коду гарантування обмеження довжини серії вбудовані слова даних, які повинні замінятися кодами підстановки для гарантування обмеження довжини серії Таблиці 2. У разі Таблиці 2 ці слова даних дорівнюють (00001000) і (00000000). Коли слово даних, що вимагає гарантування обмеження довжини серії, виявляється внаслідок звернення до цієї частини таблиці перетворення, блок 34 виявлення коду гарантування обмеження довжини серії видає сигнал виявлення, який вказує, що довжина кодового обмеження $i = 4$, блокові 32 визначення довжини кодового обмеження.

При отриманні від блоку 33 виявлення коду обмеження послідовних появ мінімальних серій сигналу виявлення, що вказує, що довжина кодового обмеження $i = 3$ у випадку Таблиці 2, блок 32 визначення довжини кодового обмеження передає значення довжини кодового обмеження $i = 3$ далі на мультиплексор 36, навіть якщо в цей час сам блок 32 визначення довжини кодового обмеження визначає значення довжини кодового обмеження i , замість видачі на мультиплексор 36 значення, визначеного ним самим. Крім того, при отриманні від блоку 34 виявлення коду гарантування обмеження довжини серії сигналу виявлення, що вказує, що довжина кодового обмеження $i = 4$ у випадку Таблиці 2, блок визначення довжини кодового обмеження передає значення довжини кодового обмеження $i = 4$ далі на мультиплексор 36, навіть якщо в цей час сам блок 32 визначення довжини кодового обмеження визначає значення довжини кодового обмеження i , замість видачі на мультиплексор 36 значення, визначеного ним самим.

Це означає, що блок 32 визначення довжини кодового обмеження передає значення довжини кодового обмеження i , отримане від блоку 33 виявлення коду обмеження послідовних появ мінімальних серій або блоку 34 виявлення коду гарантування обмеження довжини серії, далі на мультиплексор замість видачі значення, визначеного ним самим, якщо виявляється відмінність значення довжини кодового обмеження i , визначеного блоком 33 виявлення коду обмеження послідовних появ мінімальних серій або блоком 34 виявлення коду гарантування обмеження довжини серії, від значення, визначеного ним самим. Іншими словами, для передачі на мультиплексор 36 вибирається більша довжина кодового обмеження.

Фіг. 4 - це діаграма, що ілюструє процес обробки, що виконується блоком 32 визначення довжини кодового обмеження, блоком 33 виявлення коду обмеження послідовних появ мінімальних серій і блоком 34 виявлення коду гарантування обмеження довжини серії, на конкретному прикладі.

Як було описано вище, в блок 34 виявлення коду гарантування обмеження довжини серії вбудовані слова даних (00001000) і (00000000) Таблиці 2 як частини його функції визначення значення довжини кодового обмеження i . Коли 8 - бітові дані, співпадаючі зі словами даних (00001000) або (00000000), подаються на блок 34 виявлення коду гарантування обмеження довжини серії, блок 34 виявлення коду гарантування обмеження довжини серії видає сигнал виявлення, що вказує, що довжина кодового обмеження $i = 4$, блокові 32 визначення довжини кодового обмеження.

З іншого боку, в блок 33 виявлення коду обмеження послідовних появ мінімальних серій вбудоване слово даних (110111) Таблиці 2 як його функція визначення значення довжини кодового обмеження i . Коли 6 - бітові дані, співпадаючі зі словом даних (110111), подаються на блок 33 виявлення коду обмеження послідовних появ мінімальних серій, і 3 - бітове кодове слово, що отримується внаслідок перетворення, наступне за словом даних, дорівнює "010", блок 33 виявлення коду обмеження послідовних появ мінімальних серій видає сигнал виявлення, що вказує, що довжина кодового обмеження $i = 3$, блокові 32 визначення довжини кодового обмеження. Потрібно зазначити, що 3 - бітове кодове слово "010" є результатом перетворення рядка даних, що до перетворення має значення (01), (001) або (0000). Іншими словами, функція включає рядок даних (110111) + (01/001/00000). Коли виявляються 6 - бітові дані, співпадаючі зі словом даних (110111), дані аж до 5 бітів, наступні за 6 - бітовими даними, порівнюються зі словом даних (01), (001) або (0000) для визначення того, чи співпадають вони один з одним. Якщо поступаючі дані дорівнюють (11011101), (11011001) або (11011100000), блок 33 виявлення коду обмеження послідовних появ мінімальних серій видає сигнал виявлення, що вказує, що довжина кодового обмеження $i = 3$, блокові 32 визначення довжини кодового обмеження.

У блок 32 визначення довжини кодового обмеження вбудовані рядки даних таблиці перетворення Таблиці 2. Якщо 6 - бітові дані, співпадаючі зі словом даних (000011), (000010), (000001) або (000000), подаються на блок 32 визначення довжини кодового обмеження, блок 32 визначення довжини кодового обмеження визначає, що значення довжини кодового обмеження i дорівнює 3. Якщо 4 - бітові дані, співпадаючі зі словом даних (0011), (0010) або (0001), подаються на блок 32 визначення довжини кодового обмеження, блок 32 визначення довжини кодового обмеження визначає, що значення довжини кодового обмеження i дорівнює 2. Якщо 2 - бітові дані, співпадаючі зі словом даних (11), (10) або (01), подаються на блок 32 визначення довжини кодового обмеження, блок 32 визначення довжини кодового обмеження визначає, що значення довжини кодового обмеження i дорівнює 1.

Передбачимо, що подаються 6 - бітові дані (000010). У цьому випадку блок 32 визначення довжини кодового обмеження визначає, що значення довжини кодового обмеження i дорівнює 3. Також передбачимо, що за 6 - бітовими даними слідує 2 біти (00). У результаті 8 - бітові дані, співпадаючі зі словом даних (00001000) подаються на блок 34 виявлення коду гарантування обмеження довжини серії, примушуючи блок 34 виявлення коду гарантування обмеження довжини серії видати сигнал виявлення, що вказує, що довжина кодового обмеження $i = 4$, блокові 32 визначення довжини кодового обмеження. У цьому випадку блок 32 визначення довжини кодового обмеження дозволяє сигналу виявлення, що передає

значення 4 від блоку 34 виявлення коду гарантування обмеження довжини серії, мати пріоритет над значенням 3, визначеним ним самим, при цьому визначаючи довжину кодового обмеження і як таку, що має значення 4.

Як було описано вище, довжина кодового обмеження даних, що містять послідовність одиниць і нулів, може бути визначена відповідно до таблиці перетворення Таблиці 2 зверненням до слова даних, що подається, довжиною до 8 бітів, відповідному максимальній довжині кодового обмеження, і, при необхідності, до 3 - бітового кодового слова. Як альтернатива, довжина кодового обмеження даних, що містять послідовність одиниць і нулів, може бути визначена зверненням тільки до слова даних, що подається, довжиною до 11 бітів.

Блок 32 визначення довжини кодового обмеження видає значення довжини кодового обмеження і, визначеного таким способом, на мультиплексор 36.

Потрібно зазначити, що блок 32 визначення довжини кодового обмеження може також визначати значення довжини кодового обмеження і в зростаючому порядку значень і, починаючи з найменшого значення, тобто в порядку $i = 1, i = 2, i = 3$ і $i = 4$, в протилежність порядку, показаному на фіг. 4.

Кожний з блоків перетворення з 35 - 1 по 35 - 4 має таблицю перетворення, відповідну присвоєному йому значенню довжини кодового обмеження. Точніше, блоки перетворення з 35 - 1 по 35 - 4 мають таблиці перетворення для $i = 1, i = 2, i = 3$ і $i = 4$ відповідно. Якщо правило перетворення для даних, що подаються на будь-який з блоків перетворення з 35 - 1 по 35 - 4, занесене в таблицю блоків перетворення, 2х бітів даних, що подаються, перетворюються в 3х бітів коду відповідно до занесеного правила перетворення. Код, що отримується в результаті, подається потім на мультиплексор 36.

Мультиплексор 36 вибирає код, що отримується внаслідок перетворення, виконаного одним з блоків перетворення 35 - і, відповідним довжині кодового обмеження і, виданий блоком 32 визначення довжини кодового обмеження. Вибраний код потім виводиться через буфер 37 у вигляді послідовних даних.

Як показано в Таблиці 2, для довжини кодового обмеження $i = 3$ таблиця перетворення не містить код підстановки для рядка даних (110111), що вимагає обмеження послідовних появ мінімальної серії d, що повторюються. Припустимо, що подається такий рядок даних:

(1101110111011101)

У цьому випадку обробка перетворення виконується в такому порядку слів даних: (11), (01), (11), (01) і т.д. Внаслідок перетворення генерується наступний рядок кодових слів (послідовність бітів каналу): "010 010 101 010 101 010 101 010"

Потім звичайно до згенерованого рядка кодових слів застосовується NRZI модуляція для виконання кодування по рівню. Оскільки логічна інверсія відбувається при появі '1' в сигналі, вищенаведений рядок кодових слів перетворюється в такий рядок кодових слів:

'110 011 001 100 110 011'

де мінімальні інтервали інверсії 2T зберігаються у всьому рядку. При записуванні або зчитуванні при високій лінійній густині такий кодовий рядок стає комбінацією, що легко викликає помилку в операції запису або зчитування.

Припустимо, що таблиця перетворення Таблиці 2 також передбачає код підстановки для рядка даних (110111), що вимагає обмеження послідовних появ мінімальної серії d, що повторюються. Нехай тепер подається такий рядок даних:

(1101110111011101)

У цьому випадку перше слово даних (11011101) в рядку даних містить слово даних (110111), за яким слідує слово даних (01), яке буде перетворене в рядок кодових слів "010". Таким чином, перше слово даних перетворюється в такий рядок кодових слів:

"001 000 000 010"

Аналогічно друге слово даних (11011101) в рядку даних також містить слово даних (110111), за яким слідує слово даних (01), яке буде перетворене в рядок кодових слів "010". Таким чином, перше слово даних перетворюється в такий рядок кодових слів:

"001 000 000 010".

У результаті рядок даних перетворюється в такий рядок кодових слів:

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

в якому запобігаються послідовні появи мінімальної серії d, що повторюються. Тобто виключається комбінація, що легко викликає помилку в операції запису або зчитування. Потрібно зазначити, що у вищеописаному перетворенні рядка даних в рядок кодових слів зберігаються відповідні значення мінімальної серії d і максимальної серії k.

Як було описано вище, перетворення, що виконується пристроєм модуляції 1, засноване на таблиці перетворення Таблиці 2. Потрібно зазначити, що перетворення також може виконуватися з використанням таблиці перетворення Таблиці 3. У цьому випадку блок 33 виявлення коду обмеження послідовних появ мінімальних серій, що застосовується в блоці модуляції 12, показаному на фіг. 3, забезпечується Таблицею С для довжини кодового обмеження $i = 4$ Таблиці 3. З іншого боку, блок 34 виявлення коду гарантування обмеження довжини серії забезпечується Таблицею А для довжини кодового обмеження $i = 2$, Таблицею В для довжини кодового обмеження $i = 3$ і Таблицею D для довжини кодового обмеження $i = 4$ Таблиці 3.

До речі, в Таблицях 2 і 3 склад кожної пари, що складається з рядка даних і рядка кодових слів всередині групи з однаковою довжиною кодового обмеження, може мінятися. Наприклад, у разі групи з довжиною кодового обмеження $i = 1$ Таблиці 2 початковий склад кожної пари показаний нижче:

Дані Код

i = 1	11	*0*
	10	001
	01	010
Склад пари може бути змінений таким чином:		
	Дані 11	Код
i = 1	11	*0*
	10	010
	01	001

Навіть із зміненим складом пар, залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодів слів, що отримується внаслідок перетворення цього рядка даних.

Далі з посиланням на фіг. 5 пояснюється варіант здійснення, що реалізовує пристрій демодуляції, що передбачається даним винаходом. У даному варіанті здійснення код змінної довжини $(d, k; m, n; r) = (1, 7; 2, 3; 4)$ демодулюється назад в дані з використанням Таблиці 2.

Як показано на фіг. 5, пристрій демодуляції 100 містить блок демодуляції 111 для демодуляції сигналу, що отримується з лінії передачі або сигналу, що зчитується з носія даних, з використанням таблиці демодуляції або зворотної таблиці перетворення, і блоку 112 видалення контрольних бітів DSV для видалення контрольних бітів DSV, вставлених через довільні інтервали в рядок даних, що отримується внаслідок модуляції, з рядка даних для відновлення початкового рядка даних. Буфер 113 використовується для тимчасового зберігання послідовних даних, що генеруються блоком 112 видалення контрольних бітів DSV. Дані, що зберігаються в буфері 113, потім зчитуються із заданою швидкістю передачі для формування вихідного сигналу. Блок 114 управління синхронізацією генерує тактові сигнали і подає ці сигнали на різні компоненти для управління синхронізацією їх роботи за часом.

Фіг. 6 - це структурна схема, на якій показана конфігурація блоку демодуляції 111. Як показано на фіг. 6, блок демодуляції 111 має компаратор 121 для перетворення сигналу, що отримується з лінії передачі або сигналу, що зчитується з носія даних, в двійкові дані. Якщо сигнал, що подається на компаратор 121, є сигналом, що пройшов NRZI - модуляцію (тобто кодування по рівню), сигнал проходить процес зворотного NRZI - кодування (тобто процес кодування фронтів). Блок 122 визначення довжини кодового обмеження визначає довжину кодового обмеження і цифрового сигналу, що отримується від компаратора 121. Коли блок 123 виявлення коду обмеження послідовних появ мінімальних серій виявляє в цифровому сигналі, що генерується компаратором 121, спеціальний код для обмеження послідовних появ мінімальної серії d, блок 123 виявлення коду обмеження послідовних появ мінімальних серій видає сигнал виявлення цього коду ($i = 3$), що вказує довжину кодового обмеження і, блокові 122 визначення довжини кодового обмеження. У разі Таблиці 2, спеціальний код дорівнює "001 000 000". Крім того, коли блок 124 виявлення коду гарантування обмеження довжини серії виявляє спеціальний код для гарантування максимальної серії k, блок 124 видає сигнал виявлення цього коду ($i = 4$), що вказує довжину кодового обмеження і, блокові 122 визначення довжини кодового обмеження. У разі Таблиці 2, спеціальний код дорівнює "000 100 100 100" або "010 100 100 100".

Кожний з блоків зворотного перетворення з 125 - 1 по 125 - 4 має таблицю, що використовується для зворотного перетворення $n \times i$ - бітового коду змінної довжини в $m \times i$ - бітові дані. У разі Таблиці 2 блоки зворотного перетворення з 125 - 1 по 125 - 4 мають таблиці зворотного перетворення для довжини кодового обмеження від $i = 1$ до 4 відповідно, які по суті є тими ж таблицями, що і таблиці перетворення, вбудовані в блоки перетворення з 35 - 1 по 35 - 4, описані раніше. Мультиплексор 126 вибирає один з виходів, що генеруються блоками зворотного перетворення з 125 - 1 по 125 - 4, в залежності від результату визначення, отриманого від блоку 122 визначення довжини кодового обмеження.

Далі пояснюється робота блоку демодуляції 111, показаного на фіг. 6. Сигнал, що отримується з лінії передачі або такий, що зчитується з носія даних, подається на компаратор 121 для виконання порівняння. Сигнал, що видається компаратором 121, є цифровим сигналом зворотного NRZI - коду, тобто коду з "1", вказуючою фронт. Потім цифровий сигнал подається на блок 122 визначення довжини кодового обмеження для визначення довжини кодового обмеження і сигналу з використанням таблиці перетворення (строго кажучи, таблиці зворотного перетворення) Таблиці 2. Результат визначення, тобто значення довжини кодового обмеження і, виданий блоком 122 визначення довжини кодового обмеження, подається на мультиплексор 126.

Крім того, цифровий сигнал, що видається компаратором 121, подається також на блок 123 виявлення коду обмеження послідовних появ мінімальних серій. Блок 123 виявлення коду обмеження послідовних появ мінімальних серій має вбудовану частину зворотного перетворення, що містить код підстановки таблиці зворотного перетворення Таблиці 2, для обмеження послідовних появ мінімальної серії d. У разі Таблиці 2, кодом підстановки є кодове слово "001 000 000". Коли в цифрових даних виявляється код "001 000 000 ні 100", занесений в таблицю зворотного перетворення для обмеження послідовних появ мінімальної серії d, блок 123 виявлення коду обмеження послідовних появ мінімальних серій видає довжину кодового обмеження $i = 3$ на блок 122 визначення довжини кодового обмеження.

Крім того, цифровий сигнал, що видається компаратором 121, подається також на блок 124 виявлення коду гарантування обмеження довжини серії. Блок 124 виявлення коду гарантування обмеження довжини серії має вбудовану частину зворотного перетворення, що містить код підстановки таблиці зворотного перетворення Таблиці 2, для збереження максимальної серії k. У разі Таблиці 2 кодами підстановки є кодові слова "000 100 100 100" і "010 100 100 100". Коли в цифрових даних виявляється кодове слово "000 100 100 100" або "010 100 100 100", занесене в таблицю зворотного перетворення для збереження

максимальної серії k , блок 124 виявлення коду гарантування обмеження довжини серії видає довжину кодового обмеження $i = 4$ на блок 122 визначення довжини кодового обмеження.

Фіг. 7 - це діаграма, на якій показане зведення обробки визначення довжини кодового обмеження і модульованого коду, поданого на пристрій демодуляції 100. Як показано на фіг. 7, блок 124 виявлення коду гарантування обмеження довжини серії має вбудовану частину зворотного перетворення, що містить кодові слова "000 100 100 100" і "010 100 100 100" таблиці зворотного перетворення Таблиці 2. Коли 12 - бітовий рядок кодових слів, що подається на блок 124 виявлення коду гарантування обмеження довжини серії, співпадає з одним з кодових слів в частині зворотного перетворення, блок 124 виявлення коду гарантування обмеження довжини серії видає довжину кодового обмеження $i = 4$ на блок 122 визначення довжини кодового обмеження.

Крім того, блок 123 виявлення коду обмеження послідовних появ мінімальних серій має вбудовану частину зворотного перетворення, що містить кодове слово "001 000 000" таблиці зворотного перетворення Таблиці 2. Коли 12 - бітовий рядок кодових слів, що подається на блок 123 виявлення коду обмеження послідовних появ мінімальних серій співпадає з "001 000 000 не 100", блок 123 виявлення коду обмеження послідовних появ мінімальних серій видає довжину кодового обмеження $i = 3$ на блок 122 визначення довжини кодового обмеження. Потрібно зазначити, що виявлені 12 бітів рядка кодових слів фактично дорівнюють "001 000 000 010", хоч це не має відношення до визначення довжини кодового обмеження і зокрема.

Блок 122 визначення довжини кодового обмеження має вбудовану таблицю зворотного перетворення Таблиці 2. Якщо 9 - бітовий рядок кодових слів, що подається на блок 122 визначення довжини кодового обмеження, дорівнює "000 100 100" або "010 100 100", або якщо 12 - бітовий рядок кодових слів, що подається на цей блок, дорівнює "000 100 000 не 100" або "010 100 000 не 100", блок 122 визначення довжини кодового обмеження визначає, що довжина кодового обмеження i дорівнює 3. Якщо 6 - бітовий рядок кодових слів, що подається на блок 122 визначення довжини кодового обмеження, дорівнює "010 100" або "000 100", або якщо, з іншого боку, 9 - бітовий рядок кодових слів, що подається на цей блок, дорівнює "010 000 не 100", блок 122 визначення довжини кодового обмеження визначає, що довжина кодового обмеження i дорівнює 2. Інакше, якщо 3 - бітовий рядок кодових слів, що подається на блок 122 визначення довжини кодового обмеження, дорівнює "000", "101", "001" або "010", блок 122 визначення довжини кодового обмеження визначає, що довжина кодового обмеження i дорівнює 1.

Потрібно зазначити, що кожний з блоків 122 визначення довжини кодового обмеження, 123 виявлення коду обмеження послідовних появ мінімальних серій і 124 виявлення коду гарантування обмеження довжини серії можуть виконувати обробку в зростаючому порядку значень i , починаючи з найменшого значення, тобто в порядку $i = 1, i = 2, i = 3$ і $i = 4$, в протилежності порядку, показаному на фіг. 7.

Припустимо, що блок 122 визначення довжини кодового обмеження може також визначати значення довжини кодового обмеження i в порядку $i = 2/i = 3$ і $i = 4$, і на блок 122 визначення довжини кодового обмеження подається рядок кодових слів "000 100 100 100". Блок 122 визначення довжини кодового обмеження порівнює рядок кодових слів, що подається на нього, з рядком кодових слів у вбудованій таблиці в зростаючому порядку значень довжини кодового обмеження i , починаючи з найменшого значення, для визначення того, співпадає чи ні цей рядок кодових слів з кодовими словами в таблиці. Рядок кодових слів "000 100 100 100", що подається на блок 122 визначення довжини кодового обмеження, співпадає з одним з кодових слів для всіх довжин кодового обмеження $i = 1, i = 2, i = 3$ і $i = 4$. У цьому випадку, як правило визначення, вибирається найбільша довжина кодового обмеження i подається на мультимплексор 126.

Таблиця зворотного перетворення блоку зворотного перетворення 125 - 1 реалізовується у вигляді запам'ятовуючого пристрою, в якому блок даних (11) зберігається за адресами "101" і "000", тоді як блоки даних (10) і (01) зберігаються за адресами "001" і "010" відповідно. Кожна з таблиць зворотного перетворення блоків зворотного перетворення 125 - 2 і 125 - 4 реалізовується у вигляді запам'ятовуючого пристрою для зберігання даних таким же чином, як і в блоці зворотного перетворення 125 - 1. Рядок кодових слів довжиною $3x$ бітів, що подається на блок зворотного перетворення 125 - i , перетворюється назад в рядок даних довжиною $2x$ бітів, який згодом подається на мультимплексор 126.

Мультимплексор 126 вибирає один з рядків даних, що видаються блоками перетворення з 125 - 1 по 125 - 4 відповідно до результату визначення довжини кодового обмеження i , виданої блоком 122 визначення довжини кодового обмеження.

Таблиця 4 - це таблиця зворотного перетворення для Таблиці 2.

Таблиця 4

Таблиця зворотного перетворення (1, 7; 2, 3; 4)

Рядок кодових слів		Демодульований рядок даних
$i = 1$	101	11
	000	11
	001	10
	010	01
$i = 2$	010 100	0011
	010 000 (не 100)	0010
	000 100	0001

I = 3	000 100 100	000011
	000 100 000 (не 100)	000010
	010 100 100	000001
	010 100 000 (не 100)	000000

i = 3: Заборона довжини серії мінімального переходу
001 000 000(не 100) 110111

i = 4: k обмежене до 7
000 100 100 100 00001000
010 100 100 100 00000000

Фіг. 8 - це блок - схема, що використовується як довідкова інформація при поясненні операцій, що виконуються блоком 112 видалення контрольних бітів DSV. Блок 112 видалення контрольних бітів DSV обладнаний внутрішнім лічильником. Як показано на фіг. 8, блок - схема починається з кроку S1, на якому внутрішнім лічильником підраховується кількість бітів в рядку даних, виданому блоком демодуляції 111. Потім потік обробки переходить до кроку S2 для визначення того, досягла чи ні кількість бітів значення, що представляє заданий інтервал даних, через який вставляється контрольний біт DSV. Якщо результат визначення вказує, що кількість бітів не відповідає довільному інтервалу даних, потік обробки переходить до кроку S3, на якому дані, видані блоком демодуляції 111, виводяться в буфер 113 без обробки. З іншого боку, якщо результат визначення вказує, що кількість бітів відповідає заданому інтервалу даних, вказуючому, що поточний біт є контрольним бітом DSV, крок обробки S3 пропускається. Тобто в цьому випадку поточний біт рядка даних видаляється замість того, щоб бути виданим в буфер 113.

У будь-якому випадку потік обробки переходить до кроку S4, на якому виконується обробка введення наступного рядка даних. Потім потік обробки переходить до кроку S5 для визначення того, завершена чи ні обробка всіх даних. Якщо ще є дані для обробки, потік обробки повертається до кроку S1 для повторного виконання обробки. З іншого боку, якщо результат визначення, виконаного на кроці S5, вказує, що всі дані були оброблені, обробка завершується.

У результаті контрольні біти DSV видаляються з даних, виданих блоком 112 видалення контрольних бітів. Згодом дані виводяться через буфер 113.

Згідно з вищенаведеним описом, блок демодуляції 111 використовує таблицю перетворення Таблиці 2, або, строго кажучи, таблицю зворотного перетворення Таблиці 4. Потрібно зазначити, що аналогічна обробка може бути виконана з використанням перетворення Таблиці 3, або, строго кажучи, таблиці зворотного перетворення Таблиці 5, приведеної нижче. У цьому випадку блок 123 виявлення коду обмеження послідовних появ мінімальних серій, що застосовується в блоці демодуляції 111, показаному на фіг. 6, забезпечується Таблицею С для довжини кодового обмеження i = 4 Таблиці 3. З іншого боку, блок 124 виявлення коду гарантування обмеження довжини серії забезпечується Таблицею А для довжини кодового обмеження i = 2, Таблицею В для довжини кодового обмеження i = 3 і Таблицею D для довжини кодового обмеження i = 4 Таблиці 3.

Таблиця 5

Таблиця зворотного перетворення(1, 7; 2, 3; 4)

Рядок кодових слів	Демодульований рядок даних
r = 1 Основна таблиця	
101	00
100	01
001	10
000	11
r = 2 Таблиця підстановки А (d обмежене до 1)	
100010	0000
101 010	0001
000 010	1000
001 010	1001
r = 3 Таблиця підстановки В (k обмежене до 8)	
000 010 010	111111
001 010 010	111110
101 010 010	011110
100 010 010	011111

r = 4 Таблица підстановки C (RMTR обмежене до 6)	
100 010 010 010	00010001
100 000 010 010	10010001
000 010 010 010	10010001
r = 4 Таблица підстановки D (к обмежене до 7)	
000 001 010 010	11100000
100 001 010 010	11100010
001 010 010 010	11100001
101 010010010	11100011

До речі, є випадки, в яких в дані необхідно вставляти сигнал синхронізації(Sync). Далі описуються варіанти здійснення, що реалізують пристрій модуляції 1 і пристрій демодуляції 100, з посиланнями на фіг. 9 і фіг. 10 відповідно, які виконані з можливістю функціонування з даними з вставленими сигналами синхронізації. Також у разі цих варіантів здійснення рядок даних модулюється в код змінної довжини(d, k; m, n; r) = (l, 7; 2, 3; 4).

У ще одному пристрої модуляції даного винаходу, показаному на фіг. 9, в якому сигнали синхронізації вставляються через задані інтервали, вихід блоку 11 визначення/вставки контрольних бітів DSV подається на блок 211 визначення сигналів синхронізації. Також на блок 211 визначення сигналів синхронізації подається вихід блоку модуляції 12. Коли блок 211 визначення сигналів синхронізації визначає сигнал синхронізації з сигналів, що подаються блоком 11 визначення/вставки контрольних бітів DSV і блоком модуляції 12, блок 211 визначення сигналів синхронізації видає сигнал синхронізації на блок 212 вставки сигналів синхронізації. Блок 212 вставки сигналів синхронізації вставляє сигнал синхронізації, що подається блоком 211 визначення сигналів синхронізації, в модульований сигнал, що подається блоком модуляції 12, і подає вихід цього блоку на блок NRZI - кодування 13. Інша частина конфігурації така ж, що і в пристрої модуляції 1, показаному на фіг. 1.

У випадку, коли комбінація 24 - бітового кодового слова служить сигналом синхронізації, сигнал синхронізації перетворюється блоком 211 визначення сигналів синхронізації відповідно до Таблиці 2 в наступний код:

"#01 001 000 000 001 000 000 001"

де символ # означає біт, що залежить від безпосередньо попереднього рядка даних, включаючи контрольний біт DSV, якщо він є, відділеного вставкою сигналу синхронізації. Точніше, коли таблиця завершення використовується для завершення в операції модуляції відділеного рядка даних з використанням таблиці перетворення, "#" = "1". З іншого боку, коли Таблиця 2 використовується для завершення замість таблиці завершення, "#" = "0". Таким чином, блок модуляції 12 видає "#" = "1" або "#" = "0" на блок 211 визначення сигналів синхронізації, коли таблиця перетворення відповідно використовується або не використовується. При отриманні значення "#" від блоку модуляції 12 блок 211 визначення сигналів синхронізації додає значення "#" в початок сигналу синхронізації, а потім видає сигнал синхронізації на блок 212 вставки сигналів синхронізації.

Блок 212 вставки сигналів синхронізації вставляє сигнал синхронізації, що подається блоком 211 визначення сигналів синхронізації, в модульований сигнал, що подається блоком модуляції 12, і подає вихід цього блоку на блок NRZI - кодування 13. Інша частина конфігурації така ж, що і в пристрої модуляції, показаному на фіг. 1.

Перший блок даних, наступний за вставленим сигналом синхронізації, перетворюється з його початку без урахування даних, безпосередньо попередніх сигналу синхронізації. Кожний з блоків 12 модуляції і 211 визначення сигналів синхронізації обладнаний лічильником для підрахунку кількості заданих інтервалів, через які вставляються сигнали синхронізації. Вміст лічильника використовується для визначення позиції сигналу синхронізації.

Як було описано вище, варіант здійснення, показаний на фіг. 9, використовує таблицю перетворення Таблиці 2. Потрібно зазначити, що також може бути використана таблиця перетворення Таблиці 3. У цьому випадку блок 211 визначення сигналів синхронізації приймає приведене нижче 12 - бітове кодове слово як комбінацію сигналу синхронізації:

"xx0 100 000 000 10x"

де символ "x" означає біт, що залежить від безпосередньо попереднього і наступного рядків даних, включаючи контрольний біт DSV, якщо він є, відділених вставкою сигналу синхронізації. З біта на початку і 3 біта в кінці сигналу синхронізації визначаються Таблицею 3 таким чином. Нехай (p) буде останнім рядком даних, відділеним вставкою сигналу синхронізації, і (q) буде першим рядком даних, безпосередньо наступним за сигналом синхронізації. Рядок даних (p_i) перетворюється в 3 біти на початку сигналу синхронізації, тоді як рядок даних (0q) перетворюється в 3 біти в кінці сигналу синхронізації з використанням Таблиці 3. Між 3 бітами на початку і 3 бітами в кінці сигналу синхронізації, що отримується внаслідок перетворення, розміщені біти "100 000 000", утворюючи комбінацію. Таким чином, може бути згенерований сигнал синхронізації, що порушує необхідну максимальну серію k, але такий, що завжди зберігає її значення k = 8(9T).

Фіг. 10 - це структурна схема, на якій показана типова конфігурація варіанту здійснення, що реалізовує

ще один пристрій демодуляції 100 для демодуляції коду, що отримується внаслідок перетворення, що виконується пристроєм модуляції 1, показаним на фіг. 9. Як показано на фіг. 10, в даному варіанті здійснення поступаючий сигнал, що передається через заданий канал передачі, подається на блок демодуляції 111 і блок 221 ідентифікації сигналів синхронізації. Блок 221 ідентифікації сигналів синхронізації використовує поступаючий сигнал і сигнал, що отримується від блоку демодуляції 111, для ідентифікації сигналу синхронізації, видаючи сигнал синхронізації на блок 222 видалення сигналів синхронізації. Блок 222 видалення сигналів синхронізації видаляє сигнал синхронізації з демодульованого сигналу, що подається блоком демодуляції 111, відповідно до сигналу, що видається блоком 221 ідентифікації сигналів синхронізації. Демодульований сигнал з видаленими сигналами синхронізації потім подається на блок 112 видалення контрольних бітів DSV. Інша частина конфігурації така ж, що і в пристрої демодуляції 100, показаному на фіг. 5.

Блок 221 ідентифікації сигналів синхронізації має вбудований лічильник для підрахунку кількості кодових слів. Вміст лічильника використовується для визначення позиції кожного з сигналів синхронізації, які вставляються в рядок слів даних через задані інтервали. Після того, як позиція комбінації сигналу синхронізації визначена, зчитується біт "#", визначений при модуляції. Тобто зчитується біт на початку сигналу синхронізації і виводиться на блок демодуляції 111. Якщо початковий біт дорівнює "1", блок демодуляції 111 використовує таблицю завершення Таблиці 2 при демодуляції коду, безпосередньо попереднього сигналу синхронізації. З іншого боку, якщо початковий біт дорівнює "0", блок демодуляції 111 використовує таблицю кодів перетворення Таблиці 2 при демодуляції коду, безпосередньо попереднього сигналу синхронізації. Інші біти сигналу синхронізації видаляються, оскільки вони не несуть ніякої інформації.

Блок 221 ідентифікації сигналів синхронізації видає сигнал ідентифікації для ідентифікації бітів, що складають сигнал синхронізації, на блок 222 видалення сигналів синхронізації. Блок 222 видалення сигналів синхронізації видаляє сигнал синхронізації з демодульованого сигналу, що подається блоком демодуляції 111, відповідно до сигналу ідентифікації, що видається блоком 221 ідентифікації сигналів синхронізації. Демодульований сигнал з видаленими сигналами синхронізації потім подається на блок 112 видалення контрольних бітів DSV.

Як було описано вище, пристрій демодуляції 100, показаний на фіг. 10, використовує таблицю перетворення Таблиці 2. Потрібно зазначити, що також може бути використана Таблиця 3. У цьому випадку, наприклад, блок 221 ідентифікації сигналів синхронізації використовує вміст лічильника для визначення позиції кожного з сигналів синхронізації, які вставляються в рядок слів даних через задані інтервали. Після того, як позиція комбінації сигналу синхронізації ідентифікована, блок 221 ідентифікації сигналів синхронізації видає сигнали, що задають 3 - бітові кодові слова на початку і в кінці комбінації сигналу синхронізації, на блок демодуляції 111 для запиту блоку демодуляції 111 про те, щоб кожне з цих кодових слів також було демодульоване, оскільки кожне з них містить рядок даних.

Блок 221 ідентифікації сигналів синхронізації видає сигнал, що задає біти унікальної комбінації сигналу синхронізації, виключаючи кодові слова, що містять рядки даних, на блок 222 видалення сигналів синхронізації. Таким чином, блок 222 видалення сигналів синхронізації може видаляти тільки біти сигналу синхронізації, тобто біти унікальної комбінації, що задається сигналом, що отримується від блоку 221 ідентифікації сигналів синхронізації.

Фіг. 11 - це діаграма, на якій показаний приклад коду для запису з вставленими в нього сигналами синхронізації і контрольними бітами DSV. У цьому прикладі як сигнал синхронізації використовується 24 - бітове кодове слово. Контроль DSV виконується через інтервали в 56 бітів даних, а сигнал синхронізації вставляється через кожні 5 виконань контролю DSV. Таким чином, кількість кодових слів, тобто кількість бітів каналу на кожний сигнал синхронізації, дорівнює:

$$24 + (1 + 56 + 1 + 56 + 1 + 56 + 1 + 56 + 1 + 56 + 1) \times 1,5 = 453 \text{ кодових слова(біта каналу)}.$$

Відносна надлишковість, що вводиться в слова даних, складає біля 7,3%, що очевидно з наступних обчислень:

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

$$\text{Таким чином, відносна надлишковість} = 1 - 0,927 = 0,0728 = 7,3\%$$

Винахідники і деякі інші люди виконали моделювання з використанням вищеописаних таблиць перетворення для отримання результатів модуляції. Нижче описані результати модуляції рядка даних, що містить вставлені контрольні біти DSV, з обмеженими появами T_{min}. При моделюванні були використані Таблиці 2 і 3. З метою порівняння було також виконано моделювання з використанням Таблиці 1 для стандартної RLL - модуляції(1 - 7).

При моделюванні виконувався контроль DSV вставкою 1 контрольного біта DSV на кожні 56 бітів випадкових даних, що складаються з 13 107 200 бітів, згенерованих випадковим способом, і дані потім перетворювалися в рядок кодових слів(або послідовність бітів каналу) з використанням таблиці перетворення коду Таблиць 2 або 3. При іншому моделюванні випадкові дані, що складаються з 13 107 200 бітів, згенерованих випадковим способом, перетворювалися в рядок кодових слів(або послідовність бітів каналу) з використанням таблиці перетворення коду Таблиці 1, і згодом 2 біти каналу вставлялися як контрольні біти DSV на кожні 112 бітів коду, або 112 бітів каналу рядка кодових слів, що виходить внаслідок, для виконання контролю DSV.

Причиною, з якої при моделюванні з використанням Таблиць 2 або 3 вставлявся 1 контрольний біт DSV на кожні 56 бітів даних, тоді як при моделюванні з використанням Таблиці 1 вставлялися 2 контрольних біта DSV на кожні 112 кодових бітів, є забезпечення однакової відносної надлишковості, викликані контрольними бітами DSV, для кожного моделювання. Якщо кількість бітів, необхідних для

контролю DSV в одному випадку відрізняється від кількості бітів в іншому випадку, і повинна бути забезпечена однакова відносна надлишковість в обох випадках, Таблиця 2 або 3, що дозволяє виконувати контроль DSV з високим рівнем ефективності, забезпечує хороші низькочастотні характеристики в порівнянні з Таблицею 1.

Чисельні значення результатів моделювання обчислюються таким чином: Кіл_сер[від 1 до 10]: Кількість появ від 1 мінімальної серії до 10 послідовних мінімальних серій. Т_розм[від 2 до 10]: Кількість появ серій від 2Т до 10Т.

Сума: Кількість бітів.

Загальне: Кількість довжин серій, тобто загальна кількість появ серії 2Т, серії 3Т, і т.д.

Середня серія: (Сума/Загальне)

Чисельні значення розподілу серій: $(T_розм.[i] \times i) / \text{Сума}$, де $i = 2, 3, 4, \dots, 10$

Чисельні значення в рядках з 2Т по 10Т Таблиці 6 є чисельними значеннями розподілу серій.

Чисельні значення розподілу послідовних мінімальних серій:

$(\text{Кіл_сер}[i] \times i) / T_розм.[2Т]$, де $i = 1, 2, 3, 4, \dots, 10$.

Чисельні значення в рядках з RMTR(1) по RMTR(9) Таблиці 6 є чисельними значеннями розподілу послідовних мінімальних серій. Max_RMTR: Максимальна кількість повторень мінімальної серії. Пік_DSV: Піки обчислених значень DSV на позитивній і негативній сторонах, що спостерігаються в процесі виконання контролю DSV на послідовності бітів каналу.

Обчислення відносної надлишковості, викликаной вставкою 1 біта DSV на кожні 56 бітів даних, засноване на тому факті, що на кожні 56 бітів даних існує 1 біт DSV. Таким чином, відносна надлишковість обчислюється таким чином:

Відносна надлишковість $= 1 / (1 + 56) = 1,75\%$

Обчислення відносної надлишковості, викликаной вставкою 2 бітів DSV на кожні 112 бітів коду, засноване на тому факті, що на кожні 112 бітів кодових слів існують 2 біти DSV. Таким чином, відносна надлишковість обчислюється таким чином:

Відносна надлишковість $= 2 / (2 + 112) = 1,75\%$

Таким чином, в обох випадках виходить однакова відносна надлишковість.

Таблиця 6

PP17 Порівняння

	Таблиця 2	Таблиця 3	Таблиця 1	
	17PP-32	17PP-52	+2 біта -постійна складова (Контроль DSV)	Без контролю постійної складової (Немає контролю DSV)
Середня серія	3,3665	3,4048	3,3016	3,2868
Сума	20011947	20011947	20011788	19660782
Загальне	5944349	5877654	6061150	5981807
2Т	0,2256	0,2246	0,2417	0,1419
3Т	0,2217	0,2069	0,2234	0,2281
4Т	0,1948	0,1935	0,1902	0,1915
5Т	0,1499	0,1491	0,1502	0,1511
6Т	0,1109	0,1904	0,1135	0,1141
7Т	0,0579	0,0814	0,0561	0,0544
8Т	0,0392	0,0351	0,0218	0,0188
9Т		0,0023		
10Т		0,0009		
RMTR(1)	0,3837	0,3890	0,3628	0,3641
RMTR(2)	0,3107	0,3137	0,2884	0,2883
RMTR(3)	0,1738	0,1906	0,1717	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	
Max_RMTR	6	6	18	

Пік DSV # від 36 до 36 # від 35 до 40 * від 46 до 43 * від 1783 до 3433(#: 56 бітів даних + 1 біт постійної складової, 1,75%) (**: 112 бітів коду + 2 біта постійної складової, 1,75%)

Вищенаведені результати підтверджують, що з використанням Таблиць 2 і 3 реалізовується система RLL(1 - 7), причому в той же час зберігаються мінімальна і максимальна серії, а кількість послідовних появ мінімальної серії обмежена до 6. Крім того, результати DSV підтверджують, що контроль DSV може бути виконаний над рядом даних(тобто значення піка DSV залишаються в заданому діапазоні), і в цьому

випадку, оскільки ефективність контрольних бітів DSV висока, можливе отримання низькочастотних компонентів, які є більш задовільними, ніж в стандартному способі вставки бітів DSV в рядок кодових слів(послідовність бітів каналу). Результати DSV підтверджують, що у випадку Таблиці 1 різниця між позитивним і негативним піками DSV дорівнює $89(= 464 - 43)$, тоді як у випадку Таблиць 2 і 3 різниця становить $72(= 36 + 36)$ і $75(= 35 + 40)$ відповідно, що в будь-якому випадку менше, ніж значення для Таблиці 1.

З вищенаведеного опису очевидно, що в порівнянні зі стандартною системою RLL(1 - 7), тобто системою, заснованою на Таблиці 1, так звана система 17PP з використанням Таблиці 2 або 3, здатна обмежувати кількість повторень мінімальної серії до значення, що не перевищує 6. У результаті можна чекати поліпшення характеристики помилок при високій лінійній густині.

Крім того, оскільки ефективність контролю DSV чудова, виконання контролю DSV в системі 17PP при такій самій відносній надлишковості в 1,75%, що і в стандартній системі RLL(1 - 7), приводить до меншої різниці між значеннями негативного і позитивного піків. У результаті, оскільки низькочастотні компоненти можуть бути пригнічені, можливе стабільне здійснення операцій запису/зчитування даних.

Крім того, було також виконано моделювання для дослідження поширення помилки демодуляції, викликаной зсувом розряду в послідовності бітів каналу, згенерованої з тих же випадкових даних, що і у вищеописаному випадку. Результат дослідження показує, що найгірше поширення помилки в системі 17PP становить 3 байти. Однак результат також підтверджує, що частота фактичної генерації помилки майже дорівнює 0, значенню, що вказує на не дуже значне погіршення в порівнянні зі стандартною системою RLL(1 - 7). Підтверджені середні частоти появи помилок байта, дорівнюють 1,014 байту для Таблиці 1, 1,167 байту для Таблиці 2 і 1,174 байту для Таблиці 3. Потрібно зазначити, що для таблиць перетворення, що передбачаються даним винаходом, чисельні значення результатів частоти появи помилок включають контрольні біти DSV, але для стандартної системи RLL(1 - 7) чисельне значення не включає контрольні біти DSV. Тобто не можна сказати, що вимірювання були виконані при однакових умовах. Різниця в умовах вимірювань може впливати на чисельні значення, і тому при порівнянні необхідно врахувати вплив даної різниці на ці значення.

Таблиця 7

Характеристика за помилками зсуву

	<Таблиця 2>	<Таблиця 3>	<Таблиця 1>
	17PP-32	17PP-52	+2 біта - постійна
Найгірший випадок	3 байта	3 байта	складова 2 байта
(біти постійної складової)	Є	Є	Немає
Помилка байта (0)	0,028	0,096	0,080
Помилка байта (1)	0,777	0,0635	0,826
Помилка байта (2)	0,195	0,268	0,094
Помилка байта (3)	0,000	0,001	

Середня частота появи помилок байта 1,167 байту, 1,174 байту, 1,014 байт

Як було описано вище, в даному варіанті реалізації, таблиці перетворення з мінімальною серією d, що дорівнює 1, максимальною серією k, що дорівнює 7, і коефіцієнтом перетворення m/n, що дорівнює 2/3, включають коди підстановки для обмеження кількості послідовних появ мінімальних серій, що забезпечує таку перевагу:

поліпшуються характеристики запису і зчитування при високій лінійній густині, а також стійкість до нахилу дотичної;

уможливорюється зменшення кількості низькорівневих частин для збільшення точності обробки сигналів, такої, як AGC(автоматичне регулювання посилення, АРУ) і PLL(фазове автопідстроювання частоти, ФАПЧ), і, отже, поліпшення загальних характеристик;

у порівнянні зі стандартною системою, можлива конструкція з a small path memory length of bit abi code or the like and, hence, to reduce the size of the circuit;

Крім того, залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення цього рядка даних, забезпечуючи наступні додаткові переваги:

може бути зменшена кількість надлишкових розрядів для контролю DSV;

при мінімальній серії d, що дорівнює 1, і параметрах перетворення(m, n), що дорівнюють(2, 3), контроль DSV може виконуватися з 1,5 - бітовим кодовим словом;

у доповнення до низької відносної надлишковості, можуть зберігатися мінімальна і максимальна серії d і k;

Крім того, таблиці перетворення спеціально містять коди підстановки для збереження обмеження довжини серії, із забезпеченням таких додаткових переваг:

таблиці є компактними;

поширення помилки модуляції, викликаной зсувом розряду, є таким самим, що і в стандартній системі, заснованій на Таблиці 1.

Потрібно зазначити, що як програмоносії для зберігання комп'ютерної програми, що виконується для здійснення вищеописаної обробки, можуть використовуватися CD - ROM і напівпровідникові

запам'ятовуючі пристрої, а також мережеві і супутникові засоби зв'язку в доповнення до носія даних, такому, як магнітний диск.

Як було описано вище, відповідно до опису пристрою модуляції в пункті 1, описом способу модуляції в пункті 25, описом програмоносія в пункті 26, описом пристрою демодуляції в пункті 27, описом способу демодуляції в пункті 30 і описом програмоносія в пункті 31, обробка перетворення виконується на основі таблиці перетворення, що задає правило перетворення, згідно з яким залишок від ділення на 2 кількості "1" елемента рядка даних, що має значення 0 або 1, завжди повинен дорівнювати залишку від ділення на 2 кількості "1" елемента рядка кодових слів, що отримується внаслідок перетворення цього рядка даних, і коди перетворення вказаної таблиці включають:

основні коди для $d = 1$, $k = 7$, $m = 2$ і $n = 3$;

коди першої підстановки для обмеження кількості послідовних появ мінімальної серії d ; і

коди другої підстановки для збереження обмеження довжини серії k .

У результаті контроль DSV може виконуватися з використанням невеликої кількості надлишкових розрядів, і рядок кодових слів може записуватися і зчитуватися з малим числом помилок при високій лінійній густині. Крім того, може бути пригнічено наростання поширення помилки демодуляції, викликаного зсувом розряду.

Варіанти здійснення даного винаходу були описані з посиланнями на наступні малюнки, де:

фіг. 1 - це структурна схема, на якій показана типова конфігурація варіанту здійснення, що реалізовує пристрій модуляції, що передбачається даним винаходом;

фіг. 2 - це пояснювальна діаграма, що використовується для опису обробки, що виконується блоком 11 визначення/вставки контрольних бітів DSV, що застосовується в пристрої модуляції, показаному на фіг. 1;

фіг. 3 - це структурна схема, на якій показана типова конфігурація блоку модуляції 12, що застосовується в пристрої модуляції, показаному на фіг. 1;

фіг. 4 - це діаграма, що пояснює на прикладі обробку, що виконується блоком модуляції 12, показаним на фіг. 3;

фіг. 5 - це структурна схема, на якій показана типова конфігурація варіанту здійснення, що реалізовує пристрій демодуляції, що передбачається даним винаходом;

фіг. 6 - це структурна схема, на якій показана типова конфігурація варіанту здійснення, що реалізовує блок демодуляції 111, що застосовується в пристрої демодуляції, показаному на фіг. 5;

фіг. 7 - це пояснювальна діаграма, що використовується для опису обробки, що виконується блоком демодуляції 111, показаним на фіг. 6;

фіг. 8 - це блок - схема, що використовується як довідкова інформація при поясненні операцій, що виконуються блоком 112 видалення контрольних бітів DSV, що використовується в пристрої демодуляції, показаному на фіг. 5;

Фіг. 9 - це структурна схема, на якій показана ще одна типова конфігурація варіанту здійснення, що реалізовує пристрій модуляції, що передбачається даним винаходом;

Фіг. 10 - це структурна схема, на якій показана ще одна типова конфігурація варіанту здійснення, що реалізовує пристрій демодуляції, що передбачається даним винаходом; і

Фіг. 11 - це діаграма, на якій показаний приклад коду для запису з вставленими в нього сигналами синхронізації і контрольними бітами DSV.

Перелік основних цифрових позначень

11 Блок визначення/вставки контрольних бітів DSV

12 Блок модуляції

13 Блок NRZI - кодування

31 Регістр зсуву

32 Блок визначення довжини кодового обмеження

33 Блок виявлення коду обмеження послідовних появ мінімальних серій

34 Блок виявлення коду гарантування обмеження довжини серії

35 - 1, 35 - 4 Блоки перетворення

36 Мультиплексор

37 Буфер

111 Блок демодуляції

112 Блок видалення контрольних бітів DSV

121 Компаратор

122 Блок визначення довжини кодового обмеження

123 Блок виявлення коду обмеження послідовних появ мінімальних серій

124 Блок виявлення коду гарантування обмеження довжини серії

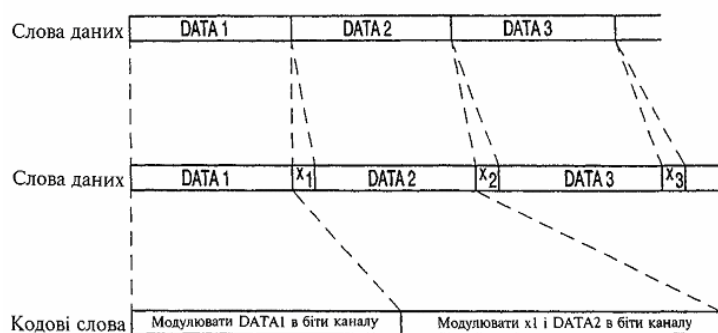
125 - 1, 125 - 4 Блоки зворотного перетворення

126 Мультиплексор

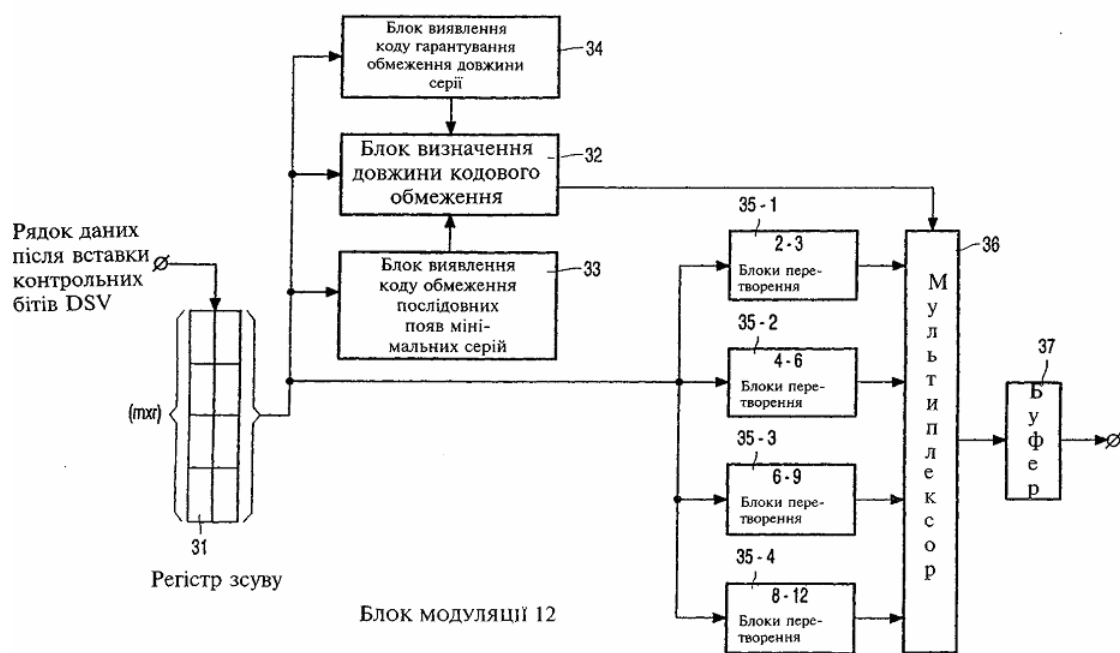


ПРИСТРІЙ МОДУЛЯЦІЇ 1

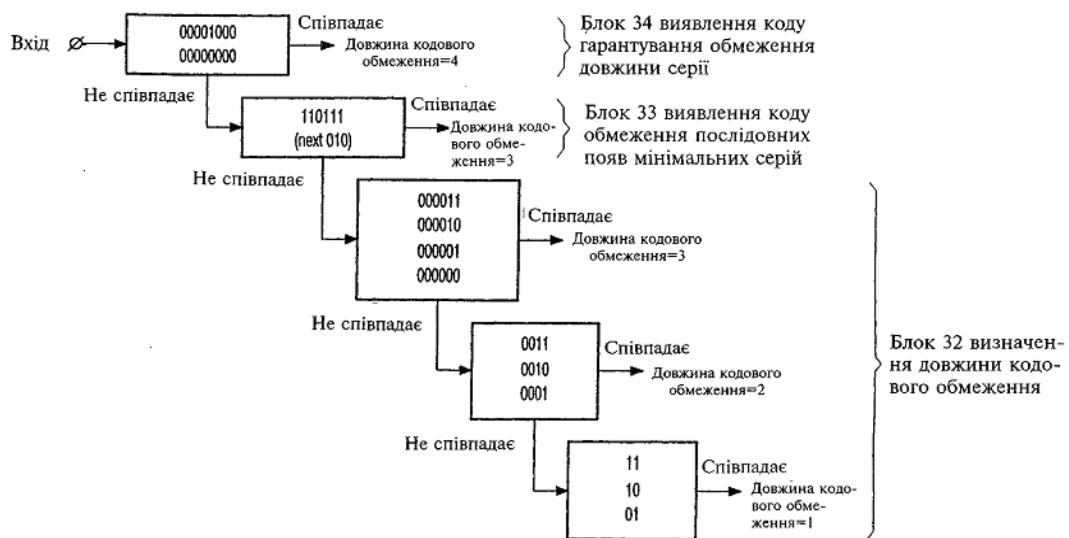
Фіг. 1



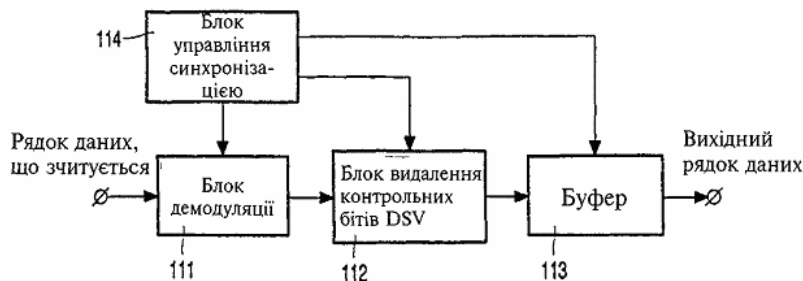
Фіг. 2



Фіг. 3

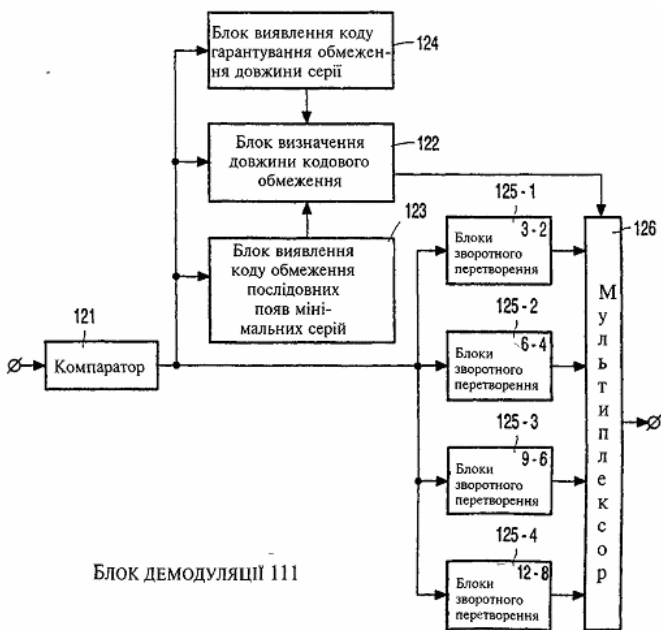


Фіг. 4

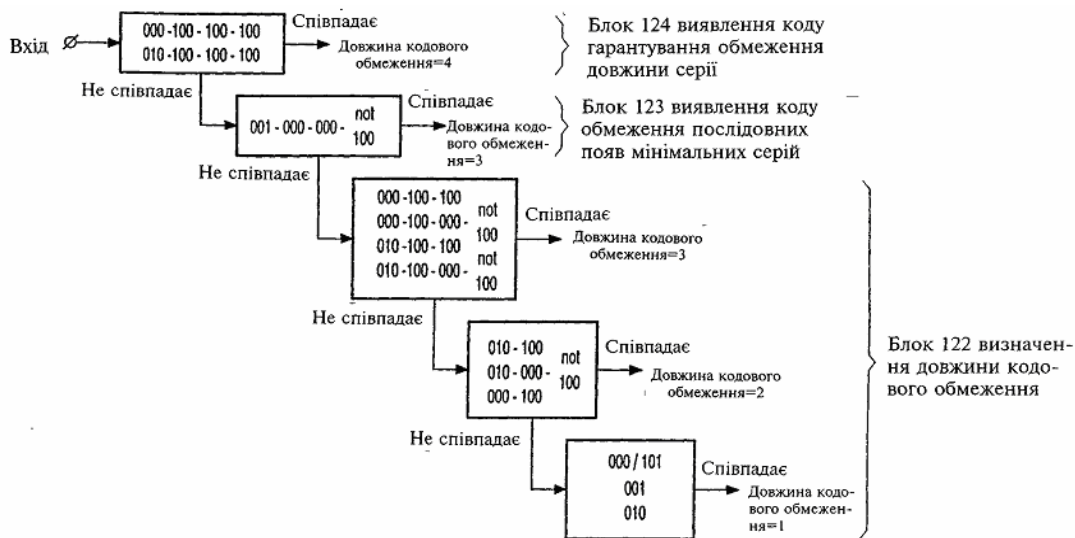


ПРИСТРІЙ ДЕМОДУЛЯЦІЇ 100

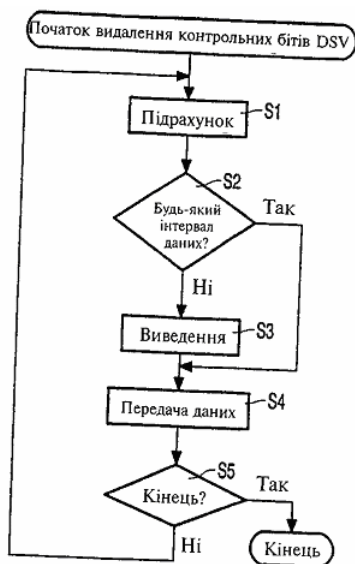
Фіг. 5



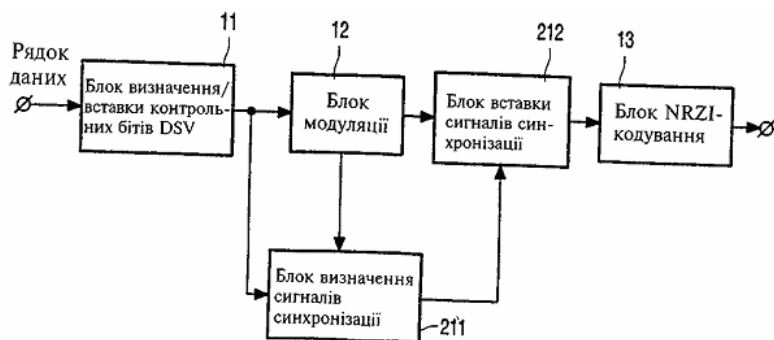
Фіг. 6



Фіг. 7



Фіг. 8



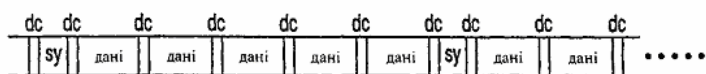
ПРИСТРІЙ МОДУЛЯЦІЇ 1

Фіг. 9



Фіг. 10

dc: Контрольний біт DSV
sy: Сигнал синхронізації



Фіг. 11