



ДЕРЖАВНА СЛУЖБА  
ІНТЕЛЕКТУАЛЬНОЇ  
ВЛАСНОСТІ  
УКРАЇНИ

УКРАЇНА

(19) **UA**

(11) **112001**

(13) **C2**

(51) МПК

**H03H 17/04** (2006.01)

## (12) ОПИС ДО ПАТЕНТУ НА ВІНАХІД

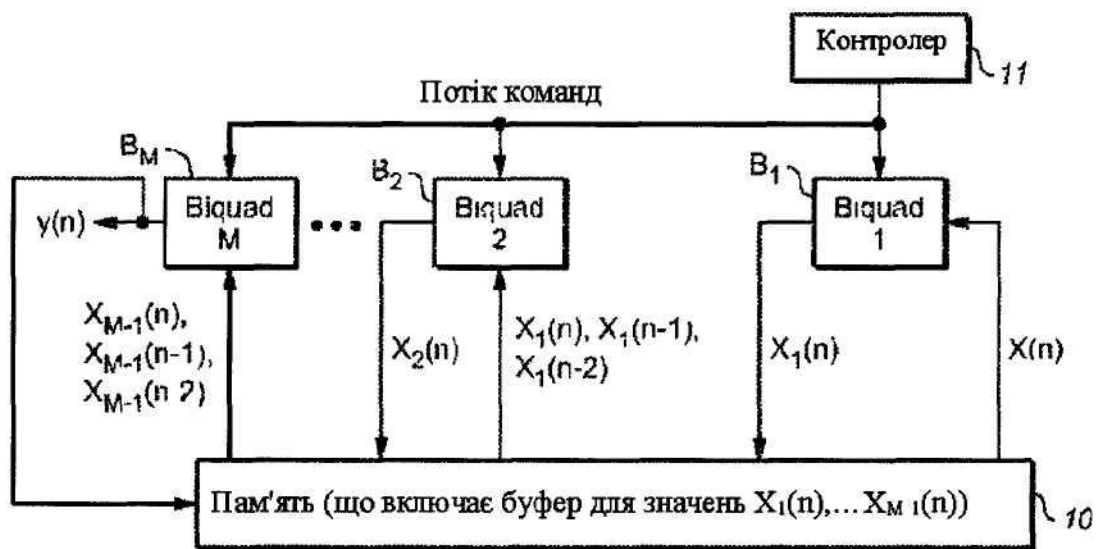
<b>(21)</b> Номер заявки:	<b>а 2014 12039</b>	<b>(72)</b> Винахідник(и):	<b>Ратхі Кхушбу П. (US)</b>
<b>(22)</b> Дата подання заявки:	<b>17.04.2013</b>	<b>(73)</b> Власник(и):	<b>ДОЛБІ ЛАБОРАТОРІС ЛАЙСЕНЗІН</b>
<b>(24)</b> Дата, з якої є чинними права на винахід:	<b>11.07.2016</b>		<b>КОРПОРЕЙШН,</b>
<b>(31)</b> Номер попередньої заявки відповідно до Паризької конвенції:	<b>61/645,291</b>		1275 Market Street, San Francisco, California 94103, USA (US)
<b>(32)</b> Дата подання попередньої заявки відповідно до Паризької конвенції:	<b>10.05.2012</b>	<b>(74)</b> Представник:	<b>Михайлюк Валентин Іванович, реєстр. №1</b>
<b>(33)</b> Код держави-учасниці Паризької конвенції, до якої подано попередню заявку:	<b>US</b>	<b>(56)</b> Перелік документів, взятих до уваги експертизою:	US 2002/138716 A1, 26.09.2002 US 2005/076073 A1, 07.04.2005 US 2011/113081 A1, 12.05.2011
<b>(41)</b> Публікація відомостей про заявку:	<b>10.12.2014, Бюл.№ 23</b>		
<b>(46)</b> Публікація відомостей про видачу патенту:	<b>11.07.2016, Бюл.№ 13</b>		
<b>(86)</b> Номер та дата подання міжнародної заявки, поданої відповідно до Договору РСТ	<b>PCT/US2013/036932, 17.04.2013</b>		

## (54) БАГАТОКАСКАДНИЙ ІІR-ФІЛЬТР І РОЗПАРАЛЕЛЕНА ФІЛЬТРАЦІЯ ДАНИХ ТАКИМ

### (57) Реферат:

У деяких варіантах здійснення винаходу багатокаскадний фільтр, каскади біквадратних фільтрів якого об'єднані з затримкою між каскадами, система (наприклад, аудіокодер або декодер), що включає такий фільтр, і способи багатокаскадної біквадратної фільтрації. У типових варіантах здійснення винаходу всі каскади біквадратних фільтрів фільтра діють незалежно з метою виконання повністю розпаралеленого оброблення даних. У деяких варіантах здійснення винаходу багатокаскадний фільтр згідно з винаходом містить буферну пам'ять, щонайменше два каскади біквадратних фільтрів і контролер, підключений і сконфігурований для затвердження єдиного потоку команд у каскадах фільтрів. Як правило, багатокаскадний фільтр є сконфігурованим для виконання багатокаскадної фільтрації на блоці вхідних дискретних значень у єдиному циклі оброблення даних з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

UA 112001 C2



Фіг. 4

## ПЕРЕДУМОВИ ВИНАХОДУ

Передресне посилання на родинні заявки

Дана заявка заявляє пріоритет попередньої заявки на патент США №61/645291, поданої 10 травня 2012 р., яка посиланням повністю включається в це розкриття.

5 1. Область винаходу

Винахід відноситься до багатокаскадних фільтрів, що включають каскади біквадратних фільтрів, і до розпаралеленої фільтрації даних (наприклад, аудіоданих) з використанням таких фільтрів. Деякі варіанти здійснення винаходу являють собою способи, системи та процесори для фільтрації аудіоданих (з використанням багатокаскадного фільтра, що включає каскади біквадратних фільтрів) у ході кодування або декодування даних відповідно до одного з форматів, відомих як Dolby Digital (AC-3), Dolby Digital Plus (E-AC-3) і Dolby E, або відповідно до іншого формату кодування. Dolby, Dolby Digital, Dolby Digital Plus і Dolby E є торговельними знаками Dolby Laboratories Licensing Corporation.

2. Передумови винаходу

Усюди в цьому розкритті, у тому числі у формулі винаходу, вираз виконання операції "на" сигналах або даних (наприклад, фільтрація або масштабування сигналів або даних) використовується в широкому змісті для позначення операції безпосередньо на сигналах або даних, або на оброблених версіях сигналів або даних (наприклад, на версіях сигналів, які піддали попередній фільтрації або іншій обробці перед виконанням на них зазначеної операції).

У області оброблення сигналів цифровий біквадратний фільтр являє собою рекурсивний лінійний фільтр другого порядку, що містить два полюси та два нулі. Скорочення "biquad" буде використовуватися в цьому розкритті для позначення цифрового біквадратного фільтра. У області Z передатна функція біквадратного фільтра являє собою відношення двох квадратичних функцій:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Рекурсивні фільтри високого порядку (фільтри з нескінченною імпульсною характеристикою, або "IIR-фільтри", з порядком вище другого порядку) можуть мати високу чутливість до квантування їх коефіцієнтів і легко можуть ставати нестійкими. Рекурсивні фільтри першого та другого порядку також можуть мати труднощі, пов'язані з нестійкістю цього типу, але ці пов'язані з нестійкістю труднощі є менш серйозними. Тому рекурсивні фільтри високого порядку, як правило, реалізуються як послідовно з'єднані в каскад фільтри, що включають послідовний каскад з біквадратних ланок (і, необов'язково, також фільтр першого порядку). Такі послідовно з'єднані в каскад фільтри іноді іменуються в цьому розкритті багатокаскадними біквадратними фільтрами та включають послідовність біквадратних фільтрів (іноді іменованих у цьому розкритті біквадратними каскадами або біквадратними ланками).

Наприклад, традиційні кодери, сконфігуровані для кодування аудіоданих відповідно до добре відомого формату AC-3 (Dolby Digital) або добре відомих форматів Dolby Digital Plus і Dolby E, реалізують декілька багатокаскадних біквадратних фільтрів. Наприклад, у кодері Dolby Digital Plus, як правило, для реалізації фільтрації пропущення верхніх частот у підсистемі детектора короточасних сигналів використовується двокаскадний біквадратний фільтр (тобто фільтр, що включає два з'єднані в каскад біквадратні фільтри), чотирьохкаскадний біквадратний фільтр (тобто фільтр, що включає чотири з'єднані в каскад біквадратні фільтри) для реалізації фільтрації пропущення нижніх частот у підсистемі низькочастотних ефектів ("LFE"); і трьохкаскадний біквадратний фільтр для реалізації фільтрації пропущення нижніх частот з обмеженням смуги пропущення. У кодері Dolby E, як правило, використовується двокаскадний біквадратний фільтр (тобто фільтр, що включає два з'єднані в каскад біквадратні фільтри) для реалізації фільтрації пропущення верхніх частот у підсистемі детектора короточасних сигналів, і чотирьохкаскадний біквадратний фільтр (тобто фільтр, що включає чотири з'єднані в каскад біквадратні фільтри) для реалізації фільтрації пропущення нижніх частот у підсистемі низькочастотних ефектів ("LFE"). У кодері Dolby E, як правило, використовується трьохкаскадний біквадратний фільтр (тобто фільтр, що включає три з'єднані в каскад біквадратні фільтри) для реалізації фільтрації пропущення нижніх частот у підсистемі низькочастотних ефектів ("LFE").

Наприклад, фіг. 1 являє собою схему біквадратного фільтра (що відноситься до типу, іноді іменованого "конструкція Direct Form II - Transposed"), що включає елементи 1, 2, 3, 4, 5,  $b_0$ ,  $b_1$ ,  $b_2$ ,  $-a_1$  і  $-a_2$ , з'єднані так, як це показано. Елементи 1, 2 і 3 являють собою елементи додавання, елементи 4 і 5 являють собою елементи затримки, і кожний з елементів посилення  $b_0$ ,  $b_1$ ,  $b_2$ ,  $-a_1$  і  $-a_2$  застосовує до сигналу, затвердженого в його введенні, відповідний один з коефіцієнтів

підсилення  $b_0$ ,  $b_1$ ,  $b_2$ ,  $-a_1$  і  $-a_2$ . І хоча цього не показано та не описано в цьому розкритті, фахівцям у даній області техніки добре відомо, що існують і інші, еквівалентні конструкції біквадратних фільтрів, наприклад "Direct Form I", "Direct Form I - Transposed" і "Direct Form II". Будь-які такі еквівалентні конструкції біквадратних фільтрів перебувають у межах обсягу винаходу.

Як показано на фіг. 1A, якщо біквадратний фільтр за фіг. 1 (позначений на фіг. 1A як Biquad 1) з'єднано в каскад з біквадратним фільтром, що мають таку ж конструкцію (позначеним на фіг. 1A як Biquad 2), але елементи посилення якого можуть застосовувати інші коефіцієнти підсилення, ніж елементи за фіг. 1, то результуючий багатокаскадний біквадратний фільтр буде являти собою один з прикладів двокаскадного біквадратного фільтра, який може бути задіяний (наприклад, для реалізації фільтрації пропущення верхніх частот у підсистемі детектора короточасних сигналів одного з вищезгаданих аудіокодерів). У багатокаскадному біквадратному фільтрі за фіг. 1A вихідний сигнал,  $x_1(n)$ , першого каскаду є вхідним сигналом у другий каскад.

Для багатокаскадних біквадратних фільтрів (і деяких інших багатокаскадних IIR-фільтрів) обчислення в кожному каскаді вихідного дискретного значення в момент часу "n" (тобто вихідного сигналу  $y(n)$  каскаду) у відповідь на значення сигналу  $x(n)$  у тимчасовій області (вхідного сигналу або сигналу, згенерованого в іншому каскаді багатокаскадного фільтра) у момент часу "n" і попередні моменти часу, має залежність від попередніх виводів (тобто виводів  $y(n-1)$  і  $y(n-2)$  у моменти часу  $n-1$  і  $n-2$ ). Також, для кожних двох послідовних каскадів (біквадратних фільтрів) у багатокаскадному біквадратному фільтрі вивід кожного більш раннього каскаду є вводом у наступний каскад, тому вивід наступного каскаду не можна визначити доти, доки не буде визначений вивід більш раннього каскаду. Ці причини є основними причинами того, чому для реалізації багатокаскадного біквадратного фільтра (до цього винаходу) не було задіяно повністю розпаралелене оброблення даних.

У багатьох сучасних архітектурах процесорного ядра (наприклад, в архітектурах процесорів цифрового оброблення сигналів) є блоки SIMD ("один потік команд - багато потоків даних") і/або кілька блоків ALU (арифметичних логічних блоків) або блоків AMU (блоків арифметичного оброблення), які можуть бути використані для розпаралелювання багатьох алгоритмів і підвищення продуктивності. Однак традиційні алгоритми для програмування процесорів з метою реалізації багатокаскадних біквадратних фільтрів не використовують команди SIMD і не є розпаралеленими.

Наприклад, кодери Dolby Digital Plus (які кодують аудіодані відповідно до формату Dolby Digital Plus) були реалізовані як програмувальні процесори ARM неон (кожний з яких являє собою процесор ARM Cortex з механізмом Neon SIMD, що допускають паралельне оброблення даних) і як програмувальні процесори цифрового оброблення сигналів Texas Instruments C64. Багато кодерів аудіоданих (наприклад, кодери, які кодують аудіодані відповідно до AC-3, Dolby Digital Plus, Dolby E і/або іншими форматами кодування) були реалізовані або могли б бути реалізовані як програмувальні процесори, що мають кожну з великої різноманітності архітектур, що містять блоки SIMD ("один потік команд - багато потоків даних") і/або кілька блоків ALU (арифметичних логічних блоків) або блоків AMU (блоків арифметичного оброблення). Такі процесори могли б програмуватися для реалізації різних алгоритмів (включених у кодування аудіоданих) з використанням паралельного оброблення даних. Однак традиційним програмуванням, яке було задіяно для реалізації багатокаскадних біквадратних фільтрів у таких процесорах, не було реалізоване паралельне оброблення даних.

Типові варіанти здійснення цього винаходу задіюють паралельне оброблення даних для реалізації багатокаскадного біквадратного фільтра. Деякі варіанти здійснення винаходу задіюють паралельне оброблення даних для реалізації багатокаскадного біквадратного фільтра, що відноситься до типу, використовуваного при кодуванні аудіоданих відповідно до формату AC-3 (Dolby Digital), формату Dolby Digital Plus або формату Dolby E.

І хоча винахід не обмежується використанням при кодуванні аудіоданих відповідно до формату AC-3, Dolby Digital Plus або Dolby E, деякі варіанти здійснення винаходу являють собою способи, системи та процесори для кодування аудіоданих (наприклад, для кодування аудіоданих відповідно до формату AC-3, Dolby Digital Plus або Dolby E), які задіюють щонайменше один багатокаскадний біквадратний фільтр, що реалізує один з варіантів здійснення винаходу (або спроектований відповідно до нього).

Кодований бітовий потік AC-3 включає від одного до шести каналів звукового вмісту та метадані, що служать ознакою щонайменше однієї характерної особливості цього звукового вмісту. Звуковий вміст являє собою аудіодані, які були стиснуті з використанням перцепційного кодування звуку.

Подробиці кодування AC-3 (також відомого як Dolby Digital) добре відомі та викладені в багатьох опублікованих джерелах, включаючи наступні:

стандарт ATSC A52/A: Digital Audio Compression Standard (AC-3), Revision A, Advanced Television Systems Committee, 20 серпня 2001;

5 "Flexible Perceptual Coding for Audio Transmission and Storage", авторів Craig C. Todd і ін., 96<sup>th</sup> Convention of the Audio Engineering Society, 26 лютого, 1994, Preprint 3796;

"Design and Implementation of AC-3 Coders", автора Steve Vernon, IEEE Trans. Consumer Electronics, Vol. 41, No. 3, серпень 1995;

10 глава "Dolby Digital Audio Coding Standards" авторів Robert L. Andersen і Grant A. Davidson, у довіднику The Digital Signal Processing Handbook, Second Edition, Vijay K. Madisetti, Editor-in-Chief, CRC Press, 2009;

"High Quality, Low-Rate Audio Transform Coding for Transmission and Multimedia Applications" авторів Bosi і ін., Audio Engineering Society Preprint 3365, 93rd AES Convention, жовтень 1992; і патенти США №№5583962; 5632005; 5633981; 5727119 і 6021386.

15 Подробиці кодування Dolby Digital (AC-3) і Dolby Digital Plus (іноді іменованого Enhanced AC-3 або "E-AC-3") викладені в статті "Introduction to Dolby Digital Plus, an Enhancement to the Dolby Digital Coding System", AES Convention Paper 6196, 117<sup>th</sup> AES Convention, 28 жовтня, 2004, і в технічних умовах Dolby Digital/Dolby Digital Plus (ATSC A/52:2010), доступних за адресою:

<http://www.atsc.org/cms/index.php/standards/published-standards>.

## 20 КОРОТКИЙ ОПИС ВИНАХОДУ

У одному з класів варіантів здійснення винаходу винахід являє собою багатокаскадний фільтр, що включає щонайменше два каскади (кожний з яких являє собою біквадратний фільтр), де каскади об'єднані з затримкою між зазначеними каскадами так, що всі каскади є діючими незалежно у відповідь на єдиний, загальний потік команд з метою виконання повністю розпаралеленого оброблення даних на зазначених каскадах. Як правило, багатокаскадний фільтр також включає контролер, підключений для затвердження загального потоку команд у всіх каскадах, і пам'ять даних, з'єднану з усіма каскадами, де всі каскади є діючими паралельно з метою фільтрації блоку вхідних значень даних у відповідь на загальний потік команд, але де кожний з каскадів діє на різні значення даних, і де щонайменше один з каскадів діє на значеннях даних, які включають буферизовані значення, згенеровані іншим з каскадів у відповідь на підмножину вхідних значень даних і збережені в пам'яті з одмінними затримками перед витяганням для оброблення в зазначеному одному з каскадів. Таким чином, багатокаскадний фільтр у цих варіантах здійснення винаходу має архітектуру SIMD ("один потік команд - багато потоків даних"), у якій окремі каскади біквадратних фільтрів діють незалежно та паралельно у відповідь на єдиний потік команд. Наприклад, багатокаскадний фільтр може включати N каскадів (де N - число більше одиниці), і один з каскадів ("M+1"-й каскад у послідовності) може діяти на значеннях даних, генерованих попереднім одним з каскадів ("M"-м каскадом у послідовності) у різний час (наприклад, у відповідь на послідовність різних вхідних значень даних блоку), збережених в буферну пам'ять (у різний час) і зчитуваних ("M+1"-м каскадом) з буферної пам'яті після перебування в буферній пам'яті протягом різного часу очікування.

40 У деяких варіантах здійснення винаходу винахід являє собою багатокаскадний фільтр, що включає:

буферну пам'ять;

45 щонайменше два каскади біквадратних фільтрів, у тому числі перший каскад біквадратного фільтра та наступний каскад біквадратного фільтра; і

контролер, з'єднаний з каскадами біквадратних фільтрів і сконфігурований для затвердження єдиного потоку команд як у першому каскаді біквадратного фільтра, так і в наступному каскаді біквадратного фільтра, де зазначений перший каскад біквадратного фільтра та зазначений наступний каскад біквадратного фільтра діють незалежно та паралельно у відповідь на потік команд,

50 де перший каскад біквадратного фільтра з'єднаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на блоці з N вхідних дискретних значень у відповідь на потік команд з метою генерування проміжних значень і для затвердження цих проміжних значень у пам'яті (для зберігання в зазначеній пам'яті), де проміжні значення включають піддану фільтрації версію кожного з щонайменше підмножини вхідних дискретних значень, і

55 де наступний каскад біквадратного фільтра пов'язаний з пам'яттю й сконфігурований для виконання біквадратної фільтрації на буферизованих значеннях, витягнутих з пам'яті, у відповідь на потік команд із метою генерування блоку вихідних значень, де вихідні значення включають вихідне значення, що відповідає кожному з вхідних дискретних значень у блоці з N вхідних дискретних значень, і буферизовані значення включають щонайменше деякі з

проміжних значень, згенерованих у першому каскаді біквадратного фільтра у відповідь на блок з  $N$  вхідних дискретних значень.

У типових варіантах здійснення винаходу багатокаскадний фільтр є сконфігурованим для виконання багатокаскадної фільтрації на блоці з  $N$  вхідних значень у єдиному циклі оброблення даних з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

У деяких варіантах здійснення багатокаскадного фільтра згідно з винаходом, у якому багатокаскадний фільтр містить  $M$  каскадів, наступний каскад біквадратного фільтра є сконфігурованим для генерування вихідного значення, що відповідає " $j$ "-му із вхідних дискретних значень, у відповідь на підмножину буферизованих значень, витягнутих з пам'яті, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , при цьому зазначена підмножина включає піддану фільтрації версію " $j$ "-го із вхідних дискретних значень, піддану фільтрації версію " $j-1$ "-го із вхідних дискретних значень і піддану фільтрації версію " $j-2$ "-го із вхідних дискретних значень.

У деяких варіантах здійснення багатокаскадного фільтра згідно з винаходом наступний каскад біквадратного фільтра є сконфігурованим для генерування вихідного значення, що відповідає кожному з вхідних дискретних значень, у відповідь на іншу підмножину буферизованих значень, витягнутих з пам'яті, де кожна така підмножина включає щонайменше три із проміжних значень, згенерованих у першому каскаді біквадратного фільтра та витягнутих з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування. Наприклад, у одному з типових варіантів здійснення винаходу, де багатокаскадний фільтр містить  $M$  каскадів біквадратних фільтрів, підмножина буферизованих значень, що витягається для генерування вихідного значення, що відповідає " $j$ "-му з вхідних дискретних значень, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , включає щонайменше одне значення, згенероване в першому каскаді біквадратного фільтра у відповідь на " $j$ "-е з вхідних дискретних значень, щонайменше одне значення, згенероване в першому каскаді біквадратного фільтра у відповідь на " $j-1$ "-е з вхідних дискретних значень, і щонайменше одне значення, згенероване в першому каскаді біквадратного фільтра у відповідь на " $j-2$ "-е з вхідних дискретних значень.

У іншому класі варіантів здійснення винаходу винахід являє собою спосіб виконання багатокаскадної фільтрації на блоці з  $N$  вхідних дискретних значень, при цьому зазначений спосіб включає етапи:

(а) виконання першої операції біквадратної фільтрації на блоці з  $N$  вхідних дискретних значень з метою генерування проміжних значень і затвердження проміжних значень у буферній пам'яті (для зберігання в зазначеній пам'яті), де проміжні значення включають піддану фільтрації версію кожного з щонайменше підмножини вхідних дискретних значень; і

(б) виконання другої операції біквадратної фільтрації на буферизованих значеннях, витягнутих з пам'яті, з метою генерування блоку вихідних значень, де вихідні значення включають вихідне значення, що відповідає кожному з вхідних дискретних значень у блоці з  $N$  вхідних дискретних значень, інша підмножина буферизованих значень витягується та фільтрується для генерування вихідного значення, що відповідає кожному з вхідних дискретних значень у блоці, і кожна така підмножина буферизованих значень включає щонайменше два (наприклад, три) з проміжних значень, згенерованих у ході виконання етапу (а), які витягуються з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування,

де етапи (а) і (б) виконуються у відповідь на єдиний потік команд так, що етапи (а) і (б) виконуються незалежно та паралельно у відповідь на єдиний потік команд. У типових варіантах здійснення винаходу багатокаскадна фільтрація блоку вхідних дискретних значень виконується в єдиному циклі з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

У деяких варіантах здійснення способу згідно з винаходом, де фільтрація виконується в багатокаскадному фільтрі, що містить  $M$  каскадів, буферизовані значення, що витягуються на етапі (б) для генерування вихідного значення, що відповідає " $j$ "-му з вхідних дискретних значень, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , включають піддану фільтрації версію " $j$ "-го з вхідних дискретних значень, згенеровану на етапі (а), піддану фільтрації версію " $j-1$ "-го з вхідних дискретних значень, згенеровану на етапі (а), і піддану фільтрації версію " $j-2$ "-го з вхідних дискретних значень, згенеровану на етапі (а).

У іншому класі варіантів здійснення винаходу винахід являє собою аудіокодер, сконфігурований для генерування кодованих аудіоданих у відповідь на вхідні аудіодані, при цьому зазначений кодер включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації аудіоданих (наприклад, для фільтрації попередньо обробленої версії аудіоданих). У іншому класі варіантів здійснення винаходу винахід являє собою спосіб

кодування аудіоданих з метою генерування кодованих аудіоданих, що полягає у виконанні будь-якого з варіантів здійснення способу згідно з винаходом багатокаскадної біквадратної фільтрації на аудіоданих (наприклад, на попередньо обробленій версії аудіоданих). Наприклад, один з варіантів здійснення винаходу являє собою аудіокодер, що включає каскад попереднього оброблення даних (призначений для попереднього оброблення вхідних аудіоданих, що підлягають кодуванню кодером), де каскад попереднього оброблення даних включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації аудіоданих (наприклад, вхідних даних або попередньо обробленої версії вхідних даних). Інший варіант здійснення винаходу являє собою передпроцесор (призначений для виконання попереднього оброблення аудіоданих, що підлягають кодуванню кодером), де передпроцесор включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації аудіоданих (наприклад, даних, які є вхідними в передпроцесор, або попередньо обробленої версії таких вхідних даних).

У іншому класі варіантів здійснення винаходу винахід являє собою аудіодекодер, сконфігурований для генерування декодованих аудіоданих у відповідь на кодовані аудіодані. У деяких варіантах здійснення винаходу в цьому класі декодер включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації кодованих аудіоданих (наприклад, для фільтрації попередньо обробленої версії кодованих аудіоданих). У іншому класі варіантів здійснення винаходу винахід являє собою спосіб декодування кодованих аудіоданих з метою генерування декодованих аудіоданих. У деяких таких варіантах здійснення винаходу декодування включає виконання будь-якого з варіантів здійснення способу згідно з винаходом багатокаскадної біквадратної фільтрації на кодованих аудіоданих (наприклад, на попередньо обробленій версії кодованих аудіоданих). Наприклад, один з варіантів здійснення винаходу являє собою аудіодекодер, що включає каскад постоброблення (призначений для постоброблення декодованих аудіоданих, які були декодовані декодером), де каскад постоброблення включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації аудіоданих (наприклад, декодованих даних або обробленої версії декодованих даних). Інший варіант здійснення винаходу являє собою постпроцесор (наприклад, призначений для виконання постоброблення декодованих аудіоданих, які були декодовані декодером), де постпроцесор включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації аудіоданих (наприклад, декодованих даних, які є вхідними в постпроцесор, або обробленої версії таких вхідних даних).

Відповідно до типових варіантів здійснення цього винаходу для виконання багатокаскадного фільтра з метою програмування процесора (наприклад, процесора цифрового оброблення сигналів або процесора загального призначення) використовуються команди SIMD (або команди для паралельного оброблення даних декількома блоками ALU або AMU). Багатокаскадний фільтр може реалізовувати смугову фільтрацію, фільтрацію пропущення нижніх частот (наприклад, у підсистемі LFE аудіокодера), фільтрацію пропущення верхніх частот (наприклад, у підсистемі детектора короточасних сигналів аудіокодера) або іншу фільтрацію.

Інші особливості винаходу включають систему або пристрій (наприклад, кодер, декодер або процесор), сконфігуровані (наприклад, запрограмовані) для виконання будь-якого із варіантів здійснення способу згідно з винаходом, і машинопрочитуваний носій даних (наприклад, диск), на якому зберігається код, призначений для реалізації будь-якого з варіантів здійснення способу згідно з винаходом або його етапів. Наприклад, система згідно з винаходом може являти собою або містити в собі програмувальний процесор загального призначення, процесор цифрового оброблення сигналів або мікропроцесор, запрограмовані програмним забезпеченням або апаратно-програмним забезпеченням і/або інакше сконфігуровані для виконання будь-якої із безлічі операцій на даних, що включають один з варіантів здійснення способу згідно з винаходом або його етапів. Такий процесор загального призначення може являти собою або містити в собі комп'ютерну систему, що включає пристрій введення, пам'ять і схему оброблення даних, запрограмовану (і/або інакше сконфігуровану) для виконання одного з варіантів здійснення способу згідно з винаходом (або його етапів) у відповідь на затверджені в ній дані.

Деякі варіанти здійснення винаходу являють собою кодери (наприклад, кодери, які кодують аудіодані відповідно до формату Dolby Digital Plus, AC-3 або Dolby E) або декодери, реалізовані

як програмувальні процесори (наприклад, процесори ARM neop, кожний з яких являє собою процесор ARM Cortex з механізмом Neon SIMD, що допускає паралельне оброблення даних, або інші процесори, що містять блоки SIMD ("один потік команд - багато потоків даних") і/або кілька блоків ALU (арифметичних логічних блоків) або AMU (блоків арифметичного оброблення)) або програмувальні (і/або такі, що конфігуруються інакше) процесори цифрового оброблення сигналів (наприклад, процесори DSP, що містять блоки SIMD і/або кілька блоків ALU або AMU).

#### КОРОТКИЙ ОПИС ГРАФІЧНИХ МАТЕРІАЛІВ

Фіг. 1 - блок-схема традиційного біквадратного фільтра.

Фіг. 1A - блок-схема традиційного багатокаскадного біквадратного фільтра.

Фіг. 2 - схема послідовності операцій традиційного способу виконання фільтрації у фільтрі, реалізованому як послідовно з'єднані в каскад біквадратні фільтри ("з'єднані в каскад біквадратні ланки").

Фіг. 3 - схема послідовності операцій одного з варіантів здійснення способу згідно з винаходом виконання фільтрації в багатокаскадному біквадратному фільтрі, що включає з'єднані в каскад біквадратні фільтри ("з'єднані в каскад біквадратні ланки"), які діють паралельно у відповідь на єдиний потік команд.

Фіг. 4 - блок-схема багатокаскадного фільтра (наприклад, реалізованого програмувальним DSP або іншим процесором відповідно до одного з варіантів здійснення винаходу), який включає з'єднані в каскад біквадратні фільтри і який може виконувати спосіб, що відноситься до типу, описаного з посиланням на фіг. 3. На фіг. 4 пам'ять 10 включає комірки пам'яті, у яких зберігається кожний блок вхідних даних  $x(n)$ , і гнізда буферної пам'яті, у яких зберігаються всі необхідні значення з числа проміжних значень  $x_1(n), \dots, x_{n-1}(n)$ , згенерованих біквадратними фільтрами.

Фіг. 5 - блок-схема системи, що включає кодер (що включає один з варіантів здійснення багатокаскадного фільтра згідно з винаходом) і декодер (що також включає один з варіантів здійснення багатокаскадного фільтра згідно з винаходом).

Фіг. 6 - схема послідовності операцій іншого варіанта здійснення способу згідно з винаходом виконання фільтрації в багатокаскадному біквадратному фільтрі, що включає з'єднані в каскад біквадратні фільтри ("з'єднані в каскад біквадратні ланки"), які діють паралельно у відповідь на єдиний потік команд.

Фіг. 7 - схема послідовності операцій одного з варіантів здійснення етапів 40, 41 і 42 варіанта здійснення способу згідно з винаходом за фіг. 6.

Фіг. 8 - схема послідовності операцій одного з варіантів здійснення етапів 47, 48 і 49 варіанта здійснення способу згідно з винаходом за фіг. 6.

Фіг. 9 - схема значень, що генеруються у реалізації системи за фіг. 4, де обчислення виконуються на місці.

Фіг. 10 - блок-схема системи, що включає кодер (що включає один з варіантів здійснення багатокаскадного фільтра згідно з винаходом) і декодер, який являє собою один з варіантів здійснення декодера згідно з винаходом.

#### ДОКЛАДНИЙ ОПИС ВАРІАНТІВ ЗДІЙСНЕННЯ ВИНАХОДУ

Варіанти здійснення способу згідно з винаходом і систем (наприклад, кодерів і декодерів), сконфігурованих для реалізації способу згідно з винаходом, будуть описані з посиланням на фіг. 3, 4, 5, 6, 7 і 8.

Спочатку, з посиланням на фіг. 2, описано традиційний спосіб фільтрації дискретних значень даних (наприклад, блоків дискретних значень аудіоданих) багатокаскадним фільтром, що включають каскад з  $M$  біквадратних фільтрів (де  $M$  - число, яке нижче й на фіг. 2 іменується "nsections"). Одним з прикладів такого традиційного багатокаскадного біквадратного фільтра є фільтр за вищеописаною фіг. 1A.

У способі за фіг. 2 кожний новий блок з  $N$  дискретних значень, що підлягають фільтрації, спочатку буферизується (на етапі 20). Кожне дискретне значення в блоці орієнтується за індексом  $j$ , де  $0 \leq j \leq N-1$ . Кожний каскад (ланка) у багатокаскадному фільтрі орієнтується за індексом  $i$ , де  $0 \leq i \leq M-1$ .

На етапі 21 індексу  $i$  привласнюється нульове початкове значення, і на етапі 22 індексу  $j$  привласнюється нульове початкове значення.

На етапі 23 "j"-е вхідне дискретне значення фільтрується в "i"-му біквадратному фільтрі, а потім на етапі 24 індекс  $j$  прирощується. Етап 25 визначає, чи є прирощений індекс  $j$  (що дорівнює  $j+1$ ) меншим, ніж  $N$ . Якщо на етапі 25 визначено, що прирощений індекс  $j$  менший, ніж  $N$ , етап 23 виконується знову з метою фільтрації наступного («(j+1)-го») дискретного значення в "i"-му біквадратному фільтрі.



Якщо на етапі 25 визначено, що прирощений індекс  $j$  що дорівнює  $N$  (так що всі дискретні значення в поточному блоці були піддані фільтрації в поточному біквадратному фільтрі), то на етапі 26 прирощується індекс  $i$ .

Етап 27 визначає, чи є найостанніший прирощений індекс  $i$  (що дорівнює  $i+1$ ) менший, ніж число "nsections" (яке дорівнює  $M$ ). Якщо на етапі 27 визначено, що найостанніший прирощений індекс  $i$  менший за  $M$ , то з метою фільтрації найостаннішого блоку проміжних значень (выводів попереднього ("i»-го) біквадратного фільтра, згенерованих у попередньому біквадратному фільтрі на попередньому повторенні етапів 22-25) виконується наступне повторення етапів 22-26 у наступному (" $i+1$ »-м) біквадратному фільтрі.

Якщо на етапі 27 визначено, що найостанніший прирощений індекс  $i$  дорівнює  $M$ , тобто оброблення всіх дискретних значень поточного блоку у всіх біквадратних фільтрах завершено, то виконується етап 28. На етапі 28 виводиться  $N$  підданих фільтрації дискретних значень, згенерованих шляхом фільтрації поточного блоку вхідних дискретних значень у багатокаскадному фільтрі. У цей момент буферизується (при виконанні етапу 20 заново) будь-який додатковий блок з  $N$  дискретних значень, що підлягають фільтрації, і спосіб за фіг. 2 повторюється для фільтрації цього нового блоку дискретних значень у багатокаскадному фільтрі.

Для процесу за фіг. 2 у випадку, коли  $M=2$  (тобто у випадку, коли багатокаскадний фільтр включає лише два з'єднані в каскад біквадратні фільтри), оброблення даних, виконване в кожному каскаді багатокаскадного фільтра, також описується нижченаведеним псевдокодом, у якому  $N$  дорівнює кількості вихідних дискретних значень, що підлягають генеруванню шляхом фільтрації блоку з  $N$  дискретних значень у багатокаскадному фільтрі:

для ( $i=0$ ;  $i<2$ ;  $i++$ )

{

25 для ( $j=0$ ;  $j<N$ ;  $j++$ )

{

Output[j]=функція(output[j-1], output[j-2], input[j], input[j-1], input[j-2]);

}

}

30 У ході виконання традиційного способу за фіг. 2, тому що вивід кожного каскаду багатокаскадного фільтра для кожного дискретного значення (output[j]) залежить, загалом, від виводів каскаду для двох попередніх дискретних значень (output[j-1] і output[j-2]) і від поточного вводу в каскад ("input[j]») і двох попередніх ввідів у каскад (input[j-1] і input[j-2]), операції циклу (етапи 23, 24 і 25 за фіг. 2), реалізовані в межах каскаду, не є розпаралеленими. Також, оскільки вивід першого каскаду ( $i=0$ ) є введенням у другий каскад ( $i=1$ ), операції за цими каскадами при виконанні традиційного способу за фіг. 2 не є розпаралеленими. Це призводить до високих вимог до команд (іноді іменованому MIPS або мільйони команд у секунду) для багатокаскадного фільтра навіть тоді, коли він реалізується процесором, архітектура якого містить кілька блоків ALU (або AMU) або блоки SIMD.

40 Далі, з посиланням на фіг. 3, описано один з варіантів здійснення способу згідно з винаходом для фільтрації блоку дискретних значень даних (наприклад, блоку дискретних значень аудіоданих) багатокаскадним фільтром, що містить каскад з двох біквадратних фільтрів. Спочатку на етапі 30 кожний новий блок з  $N$  дискретних значень, що підлягають фільтрації, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб він був доступний для використання на наступних етапах (включаючи етапи 31, 33 і 34).

45 Кожне дискретне значення в блоці розпізнається за індексом  $j$ , де  $0 \leq j \leq N-1$ . Кожний каскад (ланка) у багатокаскадному фільтрі розпізнається за індексом  $i$ , де  $0 \leq i \leq 1$ .

На етапі 31 перше вхідне дискретне значення ( $j=0$ ) фільтрується в першому ( $i=0$ ) біквадратному фільтрі. Значення, згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, при наступних виконаннях етапів 33 і/або 34).

На етапі 32 індекс  $j$  прирівнюється 1

Потім паралельно виконуються етапи 33 і 34. На етапі 33 "j»-е вхідне дискретне значення фільтрується в першому ( $i=0$ ) біквадратному фільтрі, і щонайменше одне (наприклад, кожне) значення ("проміжне" значення), згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для використання на наступних етапах. На етапі 34 "j-1»-е вхідне дискретне значення в другий ( $i=1$ ) біквадратний фільтр фільтрується, і щонайменше одне (наприклад, кожне) значення ("проміжне" значення), згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для використання на наступних етапах.

Потім на етапі 35 індекс  $j$  прирощується. Етап 36 визначає, чи є прирощений індекс  $j$  (що дорівнює  $j+1$ ) меншим, ніж  $N$ . Якщо на етапі 36 визначено, що прирощений індекс  $j$  менший за  $N$ , то етапи 33 і 34 виконуються знову для фільтрації наступного вхідного дискретного значення в перший біквадратний фільтр (на етапі 33) і наступного вхідного дискретного значення - у другий біквадратний фільтр (етап 34). Щонайменше одне (наприклад, кожне) значення ("проміжне" значення), згенероване кожним повторенням кожного з етапів 33 і 34, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для використання на наступних етапах. Наприклад, одне або кілька проміжних значень, згенерованих в одному або декількох повтореннях в одному або декількох попередніх повтореннях етапу 33, можуть витягуватися з буфера для використання при виконанні етапу 34.

Якщо на етапі 36 визначено, що прирощений індекс  $j$  дорівнює  $N$  (тобто що фільтрації в першому біквадратному фільтрі були піддані всі вхідні дискретні значення в поточному блоці), то виконується етап 37.

На етапі 37 фільтрується останнє вхідне дискретне значення ( $j=N-1$ ) у другий ( $i=1$ ) біквадратний фільтр. Значення, згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, для виводу на етапі 38).

Потім на етапі 38 виводиться (у якості виводу багатокаскадного фільтра у відповідь на поточний блок з  $N$  вхідних дискретних значень)  $N$  підданих фільтрації дискретних значень, згенерованих другим біквадратним фільтром. У цей момент буферизується (при виконанні етапу 30 заново) будь-який додатковий блок з  $N$  дискретних значень, що підлягають фільтрації, і спосіб за фіг. 3 повторюється з метою фільтрації в багатокаскадному фільтрі нового блоку дискретних значень.

Виконуване (у ході виконання способу за фіг. 3) оброблення даних у кожному каскаді багатокаскадного фільтра також описується наступним псевдокодом, у якому  $N$  дорівнює кількості вихідних дискретних значень, що підлягають генеруванню шляхом фільтрації блоку з  $N$  вхідних дискретних значень у багатокаскадному фільтрі,  $outputstage1[k]$  - вивід першого каскаду багатокаскадного фільтра у відповідь на  $k$ -е вхідне дискретне значення,  $outputstage2[k]$  - вивід другого каскаду багатокаскадного фільтра відповідно до  $k$ -го вхідного дискретного значення, а  $input[k]$  -  $k$ -е вхідне дискретне значення у перший каскад багатокаскадного фільтра:

```
{ оброблення першого дискретного значення першого каскаду }
для ( $j=1$ ;  $j<N$ ;  $j++$ )
{
  Outputstage1[j]=функція(outputstage1[j-1],
35  outputstage1[j-2], input[j], input[j-1], input[j-2]);
  Outputstage2[j-1]=функція(outputstage2[j-2], outputstage2[j-3],
  outputstage1[j-1], outputstage1[j-2], outputstage1[j-3]);
}
{Оброблення останнього ("N-1"-го) дискретного значення другого каскаду.}
```

Як видно з фіг. 3 і відповідного псевдокоду, оброблення даних в обох каскадах багатокаскадного фільтра об'єднано в єдиний цикл оброблення дискретних значень (етапи 33, 34, 35 і 36 за фіг. 3). Вносячи затримку на одне дискретне значення між двома каскадами (у випадку двокаскадного біквадратного фільтра) або, більш узагальнено, як це описується нижче, відносно фіг. 6, фіг. 7 і фіг. 8, затримку на одне дискретне значення між кожним каскадом багатокаскадного фільтра, що містить два або більше каскадів біквадратних фільтрів, оброблення даних на всіх каскадах багатокаскадного фільтра можна повністю розпаралелити згідно з винаходом. Оброблення блоку дискретних значень у всіх каскадах біквадратних фільтрів багатокаскадного фільтра, таким чином, відповідно до описаного варіанта здійснення винаходу можна розпаралелити в єдиному циклі оброблення дискретних значень (об'єднаному для всіх каскадів).

Передбачаються зміни варіанта здійснення способу згідно з винаходом за фіг. 3, для фільтрації блоку дискретних значень (наприклад, блоку дискретних значень аудіоданих) багатокаскадним фільтром, що містить каскад з  $M$  біквадратних фільтрів (де  $M$  більше 2). Такі зміни, як правило, реалізуються способом, який буде описуватися з посиланням на фіг. 6, 7 і 8.

Спочатку на етапі 40 схеми послідовності операцій за фіг. 6 кожний новий блок з  $N$  дискретних значень, що підлягають фільтрації, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб він був доступний для використання на наступних етапах (включаючи етапи 41, 43-45 і 48).

Кожне дискретне значення в блоці розпізнається за індексом  $j$ , де  $0 \leq j \leq N-1$ . Кожний біквадратний каскад (ланка) у багатокаскадному фільтрі розпізнається за індексом  $i$ , де  $0 \leq i \leq M-$

1.

На етапі 41 на перших  $M-1$  вхідних дискретних значеннях (від  $j=0$  до  $j=M-2$ ) у каскадах біквадратних фільтрів від  $i=0$  до  $i=M-2$  виконується фільтрація перед циклом (наприклад, способом, який буде описаний з посиланням на фіг. 7). Слід зазначити, що, як показано на фіг. 7, для деяких каскадів біквадратних фільтрів фільтрація перед циклом відбувається у відповідності тільки до підмножини з перших  $M-1$  вхідних дискретних значень. Значення, згенеровані на цьому етапі, буферизуються (наприклад, у пам'яті 10 за фіг. 4) так, щоб вони були доступні для наступного використання (наприклад, при наступних виконаннях етапів 43-45).

На етапі 42 індекс  $j$  прирівнюється до  $M-1$ .

Потім паралельно виконуються етапи 43-45 (по одному етапу на кожний з  $M$  каскадів). На етапі 43 " $j$ "-е вхідне дискретне значення в перший ( $i=0$ ) біквадратний фільтр фільтрується, і щонайменше одне (наприклад, кожне) значення ("проміжне" значення), згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для використання на наступних етапах. На етапі 44 фільтрується " $j-1$ "-е вхідне дискретне значення в другий ( $i=1$ ) біквадратний фільтр, і щонайменше одне (наприклад, кожне) значення ("проміжне" значення), згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для використання на наступних етапах. Аналогічно (на щонайменше одному етапі, виконуваному паралельно з етапами 43 і 44, у припущенні, що  $M$  більше 2) фільтрується " $j-2$ "-е вхідне дискретне значення в третій ( $i=2$ ) біквадратний фільтр, " $j-3$ "-е вхідне дискретне значення в четвертий ( $i=3$ ) біквадратний фільтр (у припущенні, що  $M$  більше або дорівнює 4) і далі подібним чином для кожного додаткового фільтра з числа біквадратних фільтрів від  $i=4$  до  $i=M-2$ , і щонайменше одне (наприклад, кожне) значення ("проміжне" значення), згенероване на кожному з цих етапів, буферизується так, щоб воно було доступним для використання на наступних етапах. На етапі 45 (виконуваному паралельно з етапами 43 і 44, якщо  $M$  більше або дорівнює 2) " $j-M+1$ "-е вхідне дискретне значення в останній ( $i=M-1$ ) біквадратний фільтр фільтрується, і щонайменше одне (наприклад, кожне) значення ("проміжне" значення), згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для використання на наступних етапах.

Потім на етапі 46 прирощується індекс  $j$ , і етап 47 визначає, чи є прирощений індекс  $j$  (що дорівнює  $j+1$ ) меншим, ніж  $N$ . Якщо на етапі 47 визначено, що прирощений індекс  $j$  менший за  $N$ , то етапи 43-45 (і будь-який інший етап (етапи), виконуваний паралельно з етапами 43-45) виконуються знову для фільтрації наступного дискретного значення в першому біквадратному фільтрі (на етапі 43), наступного дискретного значення - у другому біквадратному фільтрі (на етапі 44) і далі подібним чином для біквадратного фільтра кожного додаткового каскаду. Щонайменше одне (наприклад, кожне) значення ("проміжне" значення), згенероване кожним повторенням кожного з етапів 43-45, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для використання на наступних етапах. Наприклад, одне або кілька проміжних значень, згенерованих в одному або декількох попередніх повтореннях етапу 43, можуть витягуватися з буфера для використання при виконанні етапу 44.

Якщо на етапі 47 визначено, що прирощений індекс  $j$  дорівнює  $N$  (так що всі вхідні дискретні значення в поточному блоці були піддані фільтрації в одному з біквадратних фільтрів (у фільтрі, для якого  $i=0$ )), на етапі 48 виконується фільтрація після циклу.

На етапі 48 виконується фільтрація після циклу на будь-яких вхідних дискретних значеннях, які не піддавалися фільтрації, для каскадів біквадратних фільтрів від  $i=1$  до  $i=M-1$  (наприклад, способом, який буде описуватися з посиланням на фіг. 8). Значення (декілька значень), згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, для виводу на етапі 49).

Після етапу 48 на етапі 49  $N$  підданих фільтрації дискретних значень, що генеруються завершальним ( $i=M-1$ ) біквадратним фільтром, виводяться (у якості виводу багатокаскадного фільтра у відповідь на поточний блок з  $N$  вхідних дискретних значень). У цей момент буферизується (при повторному виконанні етапу 40) будь-який додатковий блок з  $N$  дискретних значень, що підлягають фільтрації, і спосіб за фіг. 6 повторюється для фільтрації в багатокаскадному фільтрі нового блоку дискретних значень.

Фіг. 7 являє собою схему послідовності операцій етапів 40 і 42 і подробиці варіанта здійснення етапу 41 варіанта здійснення способу згідно з винаходом за фіг. 6. Етапи 50-58 за фіг. 7 являють собою реалізацію етапу 41 фільтрації перед циклом за фіг. 6. На етапі 50 перше дискретне значення ( $j=0$ ) поточного блоку фільтрується в першому каскаді ( $i=0$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, на етапах 51 і 52). На етапі 51 друге дискретне значення блоку ( $j=1$ )

фільтрується в першому каскаді ( $i=0$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, на етапі 53). На етапі 52 перше дискретне значення блоку ( $j=0$ ) фільтрується в другому каскаді ( $i=1$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, переважно буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, на етапі 54). Етапи 51 і 52 переважно виконуються паралельно (у відповідь на ту саму команду або послідовність команд, затверджену в першому та другому каскадах).

На етапах, показаних на фіг. 7 нижче етапу 51 у вертикальному напрямку (у тому числі на етапах 53 і 56), кожне з дискретних значень поточного блоку від третього ( $j=2$ ) до "М-1"-го ( $j=M-2$ ) фільтрується в першому каскаді ( $i=0$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на кожному з цих етапів, буферизується так, щоб воно було доступним для наступного використання. На етапі 53 третє дискретне значення ( $j=2$ ) блоку фільтрується в першому каскаді ( $i=0$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується так, щоб воно було доступним для наступного використання. На етапі 56 "М-1"-е дискретне значення ( $j=M-2$ ) поточного блоку фільтрується в першому каскаді ( $i=0$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується так, щоб воно було доступним для наступного використання.

На етапах, показаних на фіг. 7 нижче етапу 52 у вертикальному напрямку (у тому числі на етапах 54 і 57), кожне з вхідних дискретних значень поточного блоку від другого ( $j=1$ ) до "М-2"-го ( $j=M-3$ ) у другому каскаді ( $i=1$ ) біквадратного фільтра фільтрується, і значення підданого фільтрації дискретного значення, згенероване на кожному із цих етапів, буферизується так, щоб воно було доступним для наступного використання. На етапі 54 друге вхідне дискретне значення ( $j=1$ ) у другий каскад біквадратного фільтра фільтрується, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання. На етапі 57 "М-2"-е вхідне дискретне значення ( $j=M-3$ ) у другий каскад біквадратного фільтра фільтрується, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання.

Загалом, для "к"-го каскаду біквадратного фільтра, де  $k$  - індекс в інтервалі від 0 до  $M-2$ , для фільтрації кожного з вхідних дискретних значень від першого вхідного дискретного значення ( $j=0$ ) в "к"-й каскад біквадратного фільтра до "М-1-к"-го вхідного дискретного значення ( $j=M-2-k$ ) в "к"-й каскад біквадратного фільтра виконується послідовність етапів (колонка етапів за фіг. 7), і значення (декілька значень) підданих фільтрації дискретних значень, згенероване на кожному з цих етапів, буферизується (буферизуються) (наприклад, у пам'яті 10 за фіг. 4) так, щоб вони були доступні для наступного використання.

Так, якщо  $M=3$ , то виконуються (переважно паралельно) етапи 53, 54 і 55 за фіг. 7. На етапі 53 третє вхідне дискретне значення ( $j=2$ ) у перший каскад ( $i=0$ ) біквадратного фільтра фільтрується, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується. На етапі 54 друге вхідне дискретне значення ( $j=1$ ) у другий каскад ( $i=1$ ) біквадратного фільтра фільтрується, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується. На етапі 55 перше вхідне дискретне значення ( $j=0$ ) у третій каскад ( $i=2$ ) біквадратного фільтра фільтрується, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується.

Аналогічно, коли  $M>5$ , тоді виконуються (переважно паралельно) етапи з рядка етапів нижче рядка, що включає етапи 53-55 (як зазначено на фіг. 7), а потім виконуються (переважно паралельно) етапи з рядка етапів, який включає етапи 56, 57 і 58 за фіг. 7. На етапі 56 "М-1"-е вхідне дискретне значення ( $j=M-2$ ) у перший каскад ( $i=0$ ) біквадратного фільтра фільтрується, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується. На етапі 57 "М-2"-е вхідне дискретне значення ( $j=M-3$ ) у другий каскад ( $i=1$ ) біквадратного фільтра фільтрується, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується. На етапі 58 перше вхідне дискретне значення ( $j=0$ ) в "М-1"-й каскад біквадратного фільтра ( $i=M-2$ ) фільтрується, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується.

Етапи кожного рядка етапів за фіг. 7 (наприклад, етапи 51 і 52 або етапи 53, 54 і 55) переважно виконуються паралельно (у відповідь на ту саму команду або послідовність команд, що затверджені в каскадах, що мають відношення до справи).

Фіг. 8 являє собою послідовність операцій одного з варіантів здійснення етапів 47 і 49 і подробиці одного з варіантів здійснення етапу 48 варіанта здійснення способу згідно з винаходом за фіг. 6. Етапи 60-66 за фіг. 8 являють собою одну з реалізацій етапу 48 фільтрації після циклу за фіг. 6.

Загалом, як показано на фіг. 8, для "к"-го каскаду біквадратного фільтра, де  $k$  - індекс в інтервалі від 1 до  $M-1$ , для фільтрації кожного з дискретних значень від " $N-k+1$ "-го вхідного дискретного значення ( $j=N-k$ ) в "к"-й каскад біквадратного фільтра до останнього вхідного дискретного значення ( $j=N-1$ ) в "к"-й каскад біквадратного фільтра виконується послідовність етапів (колонка етапів за фіг. 8), і значення (декілька значень) підданих фільтрації дискретних значень, згенероване (згенеровані) на кожному з цих етапів, буферизується (буферизуються) (наприклад, у пам'яті 10 за фіг. 4) так, щоб бути доступним (доступними) для наступного використання.

Наприклад, на етапі 60 останнє дискретне значення ( $j=N-1$ ) поточного блоку фільтрується в другому каскаді ( $i=1$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання. На етапі 61 передостаннє дискретне значення ( $j=N-2$ ) цього блоку фільтрується в третьому каскаді ( $i=2$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, на етапі 63). На етапі 63 останнє дискретне значення ( $j=N-1$ ) блоку фільтрується в третьому каскаді ( $i=2$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, переважно, буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання.

На етапі 64 останнє дискретне значення ( $j=N-1$ ) цього блоку фільтрується в передостанньому каскаді ( $i=M-2$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, переважно буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання.

На етапі 62 " $N-M+1$ "-е дискретне значення блоку фільтрується в останньому каскаді ( $i=M-1$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, переважно буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, на етапах нижче етапу 62 у вертикальному напрямку на фіг. 8). На етапі 65 передостаннє дискретне значення ( $j=N-2$ ) блоку фільтрується в останньому каскаді ( $i=M-1$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, переважно буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання (наприклад, на етапі 66 за фіг. 8). На етапі 66 останнє дискретне значення ( $j=N-1$ ) блоку фільтрується в останньому каскаді ( $i=M-1$ ) біквадратного фільтра, і значення підданого фільтрації дискретного значення, згенероване на цьому етапі, переважно буферизується (наприклад, у пам'яті 10 за фіг. 4) так, щоб воно було доступним для наступного використання.

Етапи з кожного рядка етапів за фіг. 8 (наприклад, етапи 64 і 65 або етапи в рядку, що включає етапи 60, 61 і 62) переважно виконуються паралельно (у відповідь на ту саму команду або послідовність команд, що затверджені в каскадах, що мають відношення до справи).

Спосіб за фіг. 3 (і його зміни, призначені для фільтрації блоку дискретних значень даних багатокаскадним фільтром, що включають каскад із двох або більше біквадратних фільтрів) виконує багатокаскадну фільтрацію блоку з  $N$  вхідних дискретних значень у єдиному циклі з повторенням за індексом дискретних значень (індекс  $j$  за фіг. 3), але без повторення за індексом каскадів біквадратних фільтрів. Для порівняння, традиційний спосіб за фіг. 2 обробляє блок дискретних значень даних у двох вкладених циклах з повторенням як за індексом дискретних значень (індекс  $j$  за фіг. 2), так і з повторенням за індексом каскадів біквадратних фільтрів (індекс  $i$  за фіг. 2).

У типових варіантах здійснення винаходу (наприклад, у варіанті здійснення винаходу за фіг. 4, який буде описуватися нижче) каскади багатокаскадного фільтра згідно з винаходом (де кожний з каскадів являє собою біквадратний фільтр) об'єднані з затримкою між каскадами так, що всі каскади можуть діяти незалежно, допускаючи розпаралелювання оброблення даних на різних каскадах. Усі каскади можуть діяти паралельно (для фільтрації блоку вхідних значень даних) у відповідь на єдиний, загальний потік команд з контролера, але кожний каскад діє на різних значеннях даних, де щонайменше один з каскадів діє на значеннях даних, які включають буферизовані значення (згенеровані одним з інших каскадів у відповідь на підмножину вхідних значень даних і збережені з різними затримками в буферній пам'яті перед витягуванням для оброблення в зазначеному одному з каскадів). Таким чином, багатокаскадний фільтр має

архітектуру SIMD ("один потік команд - багато потоків даних"), у якій окремі каскади біквдратних фільтрів діють незалежно та паралельно у відповідь на єдиний потік команд. Наприклад, багатокаскадний фільтр може включати  $N$  каскадів, і один з каскадів (" $M+1$ »-й каскад у послідовності) може діяти на значеннях даних, згенерованих одним з попередніх каскадів (" $M$ »-м каскадом у послідовності) у різний час (наприклад, у відповідь на послідовність різних вхідних значень даних із блоку), збережених у буферній пам'яті (у різний час) і зчитаних (" $M+1$ »-м каскадом) з буферної пам'яті після їхнього перебування в буферній пам'яті з різним часом очікування.

Далі, з посиланням на фіг. 4, описано один із класів варіантів здійснення багатокаскадного фільтра згідно з винаходом. Багатокаскадний фільтр за фіг. 4 містить кілька біквдратних фільтрів ( $M$  біквдратних фільтрів, де  $M$  - ціле число більше одиниці) і є сконфігурованим для виконання способу, що відноситься до типу, описаного з посиланням на фіг. 3 (або зміни такого способу, як, наприклад, зміна за фіг. 6). Фільтр за фіг. 4 містить пам'ять 10, контролер 11 і біквдратні фільтри  $B_1, B_2, \dots, B_M$ , з'єднані так, як це показано, і є сконфігурованим для фільтрації блоку з  $N$  вхідних значень даних  $x(n)$ , де " $n$ " - індекс в інтервалі від 1 до  $N$ , у відповідь на єдиний потік команд, затверджений у біквдратних фільтрах контролером 11. Кожне з вхідних значень даних  $x(n)$  може являти собою вхідне дискретне значення аудіоданих.

Слід прийняти до уваги, що вираз про те, що "єдиний потік команд" затверджений в окремих каскадах (кожний з яких, наприклад, являє собою біквдратний фільтр) багатокаскадного фільтра, використовується в цьому розкритті в широкому змісті, що охоплює як випадки, у яких єдиний потік команд затверджений у всіх каскадах (наприклад, у єдиній шині або провіднику, з якими з'єднані всі каскади), так і випадки, у яких у каскадах одночасно (або по суті одночасно) затверджені однакові (або по суті однакові) потоки команд (наприклад, кожний потік затверджений в окремій шині або провіднику, пов'язаними з одним з каскадів).

У відповідь на блок вхідних значень даних  $x(n)$  фільтр  $B_1$  генерує  $N$  проміжних (підданих біквдратної фільтрації) значень  $x_1(n)$  і затверджує їх у комірках буферної пам'яті в пам'яті 10. При застосуванні фільтр  $B_2$  витягає необхідні проміжні значення  $x_1(n)$  з пам'яті 10, генерує у відповідь на них проміжні (піддані біквдратної фільтрації) значення  $x_2(n)$  і затверджує проміжні значення, які він генерує, у комірках буферної пам'яті в пам'яті 10. Аналогічно, при застосуванні кожний з інших біквдратних фільтрів (фільтр  $B_i$ , де  $i$  - індекс в інтервалі від 3 до  $M$ ) витягає проміжні значення  $x_{i-1}(n)$  з пам'яті 10, генерує у відповідь на них піддані біквдратній фільтрації значення  $x_i(n)$  і затверджує значення, які він генерує, у комірках буферної пам'яті в пам'яті 10. Піддані біквдратній фільтрації значення,  $x_M(n)=y(n)$ , згенеровані в завершальному біквдратному фільтрі ( $B_M$ ), становлять блок з  $N$  повністю підданих фільтрації вихідних значень даних, згенерованих у відповідь на блок вхідних значень даних  $x(n)$ .

Пам'ять 10 містить комірки пам'яті, у яких зберігається кожний блок вхідних даних  $x(n)$ , і комірки буферної пам'яті, у яких зберігаються проміжні значення  $x_1(n), \dots, x_{M-1}(n)$ , згенеровані біквдратними фільтрами  $B_1, B_2, \dots, B_{M-1}$  (наприклад, буферні гнізда, у яких зберігаються проміжні значення  $x_1(n), \dots, x_{M-1}(n)$ , згенеровані для кожного блоку вхідних даних). У деяких реалізаціях, для яких обчислення виконуються на місці, для зберігання проміжних значень  $x_1(n), \dots, x_{M-1}(n)$  можуть використовуватися ті ж комірки пам'яті, які використовувалися для зберігання вхідних даних  $x(n)$ , після того, як конкретні вхідні дискретні значення даних більше не будуть потрібні для багатокаскадного фільтра. У таких реалізаціях пам'ять 10, як правило, необов'язково включає більше (або значно більше) комірок пам'яті, ніж звичайна пам'ять (призначена для реалізації традиційної, нерозпаралеленої версії багатокаскадного фільтра), оскільки така традиційна пам'ять, як правило, може включати комірки пам'яті для зберігання кожного блоку вхідних даних,  $x(n)$ , що підлягають фільтрації, і кожне вихідне значення, згенероване кожним з каскадів багатокаскадного фільтра, які потрібні для роботи самого фільтра і/або для роботи кожного з наступних каскадів фільтра.

Наприклад, фіг. 9 являє собою схему значень, згенерованих у реалізації системи за фіг. 4, де обчислення виконуються на місці, у випадку, коли  $N=4$  і  $M=2$  (тобто у випадку двокаскадної біквдратної операції на блоках з 4 дискретних значень одноразово).

У прикладі за фіг. 9 починають із чотирьох дискретних значень у вхідному буфері,  $x(0) \dots x(3)$ .

На першому етапі дискретне значення  $x(0)$  фільтрується через фільтр  $B_0$  (біквдратний фільтр із першим каскадом), що виробляє дискретне значення  $x_1(0)$ . Дискретне значення  $x_1(0)$  зберігається в пам'яті в комірці, раніше зайнятій дискретним значенням  $x(0)$ . Усі інші комірки пам'яті залишаються незмінними.

На другому етапі дискретне значення  $x_1(0)$  фільтрується через фільтр  $B_1$  (біквдратний фільтр із другим каскадом), що виробляє дискретне значення  $y(0)$ . Дискретне значення  $y(0)$

зберігається в пам'яті в комірці, раніше зайнятій дискретним значенням  $x_1(0)$ .

Паралельно дискретне значення  $x(1)$  фільтрується через фільтр  $B_0$ , що виробляє дискретне значення  $x_1(1)$ . Дискретне значення  $x_1(1)$  зберігається в пам'яті в комірці, раніше зайнятій дискретним значенням  $x(1)$ .

5 На наступних етапах оброблення даних триває доти, доки всі вхідні дискретні значення  $x(0) \dots x(3)$  не будуть заміщені вихідними дискретними значеннями  $y(0) \dots y(3)$ .

У прикладі за фіг. 9, коли фільтр  $B_1$  застосовується до дискретного значення  $x_1(2)$ , дискретні значення  $x_1(1)$  і  $x_1(0)$  уже не перебувають у буфері (були заміщені  $y(0)$  і  $y(1)$ ). Замість цього у двох змінних станах, пов'язаних з фільтром  $B_1$ , утримуються дискретні значення, отримані фільтром  $B_1$  з  $x_1(1)$  і  $x_1(0)$  (тобто дискретні значення, що відповідають  $s_1(n)$  і  $s_2(n)$  на фіг. 1).

У прикладі за фіг. 9 кожний фільтр (наприклад, кожний з фільтрів  $B_0$  і  $B_1$ ) вимагає для використання при фільтрації "j"-го вхідного дискретного значення поточного блоку доступу до комірок пам'яті, у яких зберігаються два дискретні значення (що відповідають  $s_1(n)$  і  $s_2(n)$  на фіг. 1), які він генерує (у відповідь на "j-1"-е й "j-2"-е вхідні дискретні значення поточного блоку з  $N$  вхідних дискретних значень). Ці комірки пам'яті можуть перебувати в пам'яті 10 системи за фіг. 4 (або можуть являти собою комірки іншої буферної пам'яті).

Для кожного з фільтрів кожна пара збережених дискретних значень (відповідних  $s_1(n)$  і  $s_2(n)$  на фіг. 1), які були згенеровані фільтром, обновляється щоразу, коли у фільтрі затверджується нове вхідне дискретне значення (з прирощеним індексом  $j$ ). Збережені дискретні значення (відповідні  $s_1(n)$  і  $s_2(n)$  на фіг. 1) являють собою приклади "проміжних значень" (у тому розумінні, у якому ця фраза використовується будь-де у цьому розкритті), які генеруються фільтром (що представляють собою один каскад багатокаскадного фільтра) і буферизуються для наступного використання згідно з винаходом, однак вони згодом використовуються каскадом фільтра, який їх генерує (а не іншим каскадом багатокаскадного фільтра).

Незважаючи на те що фіг. 9 показує конкретний приклад способу згідно з винаходом, що використовує фільтрацію на місці, де довжина блоку ( $N$ ) дорівнює 4, а кількість каскадів ( $M$ ) біквадратних фільтрів у фільтрі згідно з винаходом дорівнює 2, варіанти здійснення способу згідно з винаходом, що використовує фільтрацію на місці, передбачаються для будь-яких значень  $M$  і  $N$  за умови, що  $M > 1$  і  $N > M$ . У реалізації за фіг. 9 (і в інших варіантах здійснення винаходу, що використовують фільтрацію на місці, де довжина блоку відмінна від 4 і/або кількість біквадратних каскадів відмінна від 2), для зберігання проміжних значень  $x_1(n), \dots, x_{M-1}(n)$  можуть використовувати ті ж комірки пам'яті (наприклад, у пам'яті 10 за фіг. 4), які використовуються для зберігання вхідних даних  $x(n)$ , якщо конкретні вхідні дискретні значення даних більше не потрібні для багатокаскадного фільтра.

Фільтр за фіг. 4 може бути реалізований шляхом програмування процесора цифрового оброблення сигналів (DSP) або іншого процесора, який включає пам'ять (виконуючу функцію пам'яті 10), контролер (виконуючий функцію контролера 11) і блоки ALU (арифметичні логічні блоки) або блоки AMU (блоки арифметичного оброблення), де кожний з біквадратних фільтрів  $B_1, B_2, \dots, B_M$  реалізовано як відповідно сконфігурований один із блоків ALU або AMU.

Таким чином, фільтр за фіг. 4 містить:

буферну пам'ять (буферні комірки в пам'яті 10);

щонайменше два каскади біквадратних фільтрів (біквадратні фільтри  $B_1, B_2, \dots, B_M$ ), що включають перший каскад біквадратного фільтра (наприклад, біквадратний фільтр  $B_1$ ) і наступний каскад біквадратного фільтра (наприклад, біквадратний фільтр  $B_2$ ); і

контролер (контроллер 11), з'єднаний з каскадами біквадратних фільтрів і сконфігурований для затвердження єдиного потоку команд як у перший каскад біквадратного фільтра, так і в наступний каскад біквадратного фільтра. Перший каскад біквадратного фільтра та наступний каскад біквадратного фільтра (і кожний з інших каскадів біквадратних фільтрів фільтра за фіг. 4) діють незалежно та паралельно у відповідь на потік команд.

Перший каскад біквадратного фільтра пов'язаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на блоці з  $N$  вхідних дискретних значень у відповідь на потік команд з метою генерування проміжних значень (наприклад, значень  $x_1(n)$ ) і для затвердження цих проміжних значень у пам'яті (для зберігання в зазначеній пам'яті). Ці проміжні значення включають піддану фільтрації версію кожного з вхідних дискретних значень. У деяких варіантах здійснення винаходу необхідно, щоб у кожний момент часу в пам'яті 10 було присутнє не більше одного проміжного значення  $x_1(n)$ . Наступний каскад біквадратного фільтра пов'язаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на буферизованих значеннях, витягнутих з пам'яті у відповідь на потік команд з метою генерування блоку вихідних значень (наприклад, значень  $x_2(n)$ ), де вихідні значення включають вихідне значення, що

відповідає кожному з вхідних дискретних значень у блоці з  $N$  вхідних дискретних значень, і буферизовані значення включають щонайменше деякі з проміжних значень, згенерованих у першому каскаді біквадратного фільтра у відповідь на блок з  $N$  вхідних дискретних значень.

У варіанті здійснення винаходу, де багатокаскадний фільтр містить  $M$  каскадів, наступний каскад біквадратного фільтра (наприклад, фільтр  $B_2$  за фіг. 4) сконфігурований для генерування вихідного значення, що відповідає "j"-му з вхідних дискретних значень у відповідь на підмножину буферизованих значень, витягнутих з пам'яті, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , при цьому зазначена підмножина включає піддану фільтрації версію "j"-го з вхідних дискретних значень, піддану фільтрації версію "j-1"-го з вхідних дискретних значень і піддану фільтрації версію "j-2"-го з вхідних дискретних значень.

Цей наступний каскад біквадратного фільтра (наприклад, фільтр  $B_2$  за фіг. 4) сконфігурований для генерування вихідного значення ( $x_2(n)$ ), що відповідає кожному з вхідних дискретних значень,  $x(n)$ , у відповідь на іншу підмножину буферизованих значень, витягнутих з пам'яті, де кожна така підмножина включає щонайменше два (наприклад, три) з проміжних значень, згенерованих у першому каскаді біквадратного фільтра (наприклад, значення  $x_1(n)$ ,  $x_1(n-1)$  і  $x_1(n-2)$ ), що зазначені на фіг. 4) і витягнуті з пам'яті після перебування в зазначеній пам'яті з різним часом очікування. Конкретніше, в одному з варіантів здійснення винаходу, де багатокаскадний фільтр містить  $M$  каскадів, підмножина буферизованих значень, що витягаються фільтром  $B_2$  для генерування вихідного значення, що відповідає "j"-му з вхідних дискретних значень, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , включає щонайменше одне значення, згенероване в першому каскаді біквадратного фільтра у відповідь на "j"-е з вхідних дискретних значень, щонайменше одне значення, згенероване в першому каскаді біквадратного фільтра у відповідь на "j-1"-е з вхідних дискретних значень, і щонайменше одне значення, згенероване в першому каскаді біквадратного фільтра у відповідь на "j-2"-е з вхідних дискретних значень,

Фільтр за фіг. 4 сконфігурований для виконання багатокаскадної фільтрації на блоці з  $N$  вхідних дискретних значень (значень даних  $x(n)$ ), яка включає виконання етапів:

(а) виконання першої операції біквадратної фільтрації на блоці з  $N$  вхідних дискретних значень з метою генерування проміжних значень (наприклад, значень  $x_1(n)$ ,  $x_1(n-1)$  і  $x_1(n-2)$ ), зазначених на фіг. 4) і затвердження проміжних значень у буферній пам'яті (для зберігання в зазначеній пам'яті), де проміжні значення включають піддану фільтрації версію кожного із щонайменше підмножини вхідних дискретних значень; і

(б) виконання другої операції біквадратної фільтрації на буферизованих значеннях, витягнутих з пам'яті, з метою генерування блоку вихідних значень (наприклад, значень  $x_2(n)$ ), зазначених на фіг. 4), де вихідні значення включають вихідне значення, що відповідає кожному з вхідних дискретних значень у блоці з  $N$  вхідних дискретних значень, і інша підмножина буферизованих значень витягується та фільтрується для генерування вихідного значення, що відповідає кожному з вхідних дискретних значень у блоці, і кожна зазначена підмножина буферизованих значень включає щонайменше два (наприклад, три) з проміжних значень, згенерованих шляхом виконання етапу (а) (наприклад, значення  $x_1(n)$ ,  $x_1(n-1)$  і  $x_1(n-2)$ ), що зазначені на фіг. 4), які витягнуті з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування,

де етапи (а) і (б) виконуються у відповідь на єдиний потік команд так, що етапи (а) і (б) виконуються у відповідь на цей єдиний потік команд незалежно та паралельно.

У варіанті здійснення винаходу, де фільтрація виконується в багатокаскадному фільтрі, що містить  $M$  каскадів, буферизовані значення, витягнуті на етапі (б) для генерування вихідного значення, що відповідає "j"-му з вхідних дискретних значень, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , включають піддану фільтрації версію "j"-го з вхідних дискретних значень, згенерованих на етапі (а), піддану фільтрації версію "j-1"-го з вхідних дискретних значень, згенерованих на етапі (а), і піддану фільтрації версію "j-2"-го з вхідних дискретних значень, згенерованих на етапі (а).

Фіг. 5 являє собою блок-схему системи, що містить кодер (кодер 150), що містить один з варіантів здійснення багатокаскадного фільтра згідно з винаходом ("фільтр МВ" 153). Наприклад, фільтр 153 може відноситись до типу, показаного та описаного з посиланням на фіг. 4. Кодер 150, необов'язково, включає два або більше багатокаскадних фільтрів, кожний з яких являє собою один з варіантів здійснення багатокаскадного фільтра згідно з винаходом. У відповідь на вхідні дискретні значення аудіоданих кодер 150 генерує кодовані аудіодані та затверджує кодовані аудіодані в підсистемі 151 доставки.

Підсистема 151 доставки сконфігурована для зберігання кодованих аудіоданих і/або для передачі сигналу, що служить ознакою кодованих аудіоданих. Декодер 152 підключений і сконфігурований (наприклад, запрограмований) для приймання кодованих аудіоданих з



підсистеми 151 (наприклад, шляхом зчитування або витягування кодованих аудіоданих з пам'яті в підсистемі 151 або шляхом приймання сигналу, що служить ознакою кодованих аудіоданих і був переданий підсистемою 151).

Декодер 152 містить один з варіантів здійснення багатокаскадного фільтра згідно з винаходом ("фільтр MB" 154). Наприклад, фільтр 154 може відноситись до типу, показаного та описаного з посиланням на фіг. 4. Декодер 152, необов'язково, містить два або більше багатокаскадних фільтрів, кожний з яких являє собою один з варіантів здійснення багатокаскадного фільтра згідно з винаходом. Декодер 152 діє для декодування кодованих аудіоданих, за допомогою чого генеруються декодовані аудіодані.

Система за фіг. 5 також містить підсистему 155 попереднього оброблення аудіоданих ("передпроцесор"), сконфігуровану для виконання попереднього оброблення аудіоданих, що підлягають кодуванню кодером 150. Передпроцесор 155 включає один з варіантів здійснення багатокаскадного фільтра згідно з винаходом ("фільтр MB" 157). Наприклад, фільтр 157 може відноситись до типу, показаного та описаного з посиланням на фіг. 4.

Система за фіг. 5 також включає підсистему 156 постоброблення аудіоданих ("постпроцесор"), сконфігуровану для виконання постоброблення декодованих аудіоданих, які були декодовані декодером 154. Постпроцесор 156 включає один з варіантів здійснення багатокаскадного фільтра згідно з винаходом ("фільтр MB" 158). Наприклад, фільтр 158 може відноситись до типу, показаного та описаного з посиланням на фіг. 4.

У деяких реалізаціях кодер 150 являє собою кодер AC-3 (або вдосконалений AC-3 або Dolby E), який сконфігурований для генерування кодованого бітового потоку аудіоданих AC-3 (або вдосконаленого AC-3 або Dolby E) у відповідь на вхідні аудіодані в тимчасовій області, і декодер 52 являє собою декодер AC-3 (або вдосконалений AC-3 або Dolby E).

У одному з класів варіантів здійснення винаходу винахід являє собою аудіокодер (наприклад, кодер 150 за фіг. 5), сконфігурований для генерування кодованих аудіоданих у відповідь на вхідні аудіодані, де зазначений кодер включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації аудіоданих (наприклад, для фільтрації попередньо обробленої версії аудіоданих). Кодер 150 сконфігурований для кодування аудіоданих з метою генерування кодованих аудіоданих, що полягає у виконанні одного з варіантів здійснення способу багатокаскадної фільтрації згідно з винаходом на аудіоданих (наприклад, на попередньо обробленій версії аудіоданих).

У одному з класів варіантів здійснення винаходу винахід являє собою аудіодекодер (наприклад, декодер 152 за фіг. 5), сконфігурований для генерування декодованих аудіоданих у відповідь на кодовані аудіодані, де зазначений декодер включає щонайменше один багатокаскадний фільтр (який являє собою будь-який із варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації кодованих аудіоданих (наприклад, для фільтрації попередньо обробленої версії кодованих аудіоданих). Декодер 152 сконфігурований для декодування кодованих аудіоданих з метою генерування декодованих аудіоданих, що полягає у виконанні одного з варіантів здійснення способу багатокаскадної фільтрації згідно з винаходом на кодованих аудіоданих (наприклад, на попередньо обробленій версії кодованих аудіоданих).

Інший варіант здійснення винаходу являє собою передпроцесор (наприклад, передпроцесор 155 за фіг. 5), призначений для виконання попереднього оброблення аудіоданих (наприклад, аудіоданих, що підлягають кодуванню кодером), де передпроцесор включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації аудіоданих (наприклад, даних, які являють собою введення в передпроцесор, або попередньо обробленої версії таких вхідних даних).

Інший варіант здійснення винаходу являє собою постпроцесор (наприклад, постпроцесор 156 за фіг. 5), призначений для виконання постоброблення декодованих аудіоданих, які були декодовані декодером, де постпроцесор включає щонайменше один багатокаскадний фільтр (який являє собою будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом), підключений і сконфігурований для фільтрації аудіоданих (наприклад, декодованих даних, які являють собою введення в постпроцесор, або попередньо обробленої версії таких вхідних даних).

Фіг. 10 являє собою блок-схему системи, що включає кодер (кодер 150), що включає один з варіантів здійснення багатокаскадного фільтра згідно з винаходом ("фільтр MB" 153). Кодер 150 за фіг. 10 є таким же, як кодер 150 за фіг. 5, і він може бути реалізований кожним зі способів, якими може бути реалізований кодер 150 за фіг. 5. Багатокаскадний фільтр 153 може

реалізовувати будь-який з варіантів здійснення багатокаскадного фільтра згідно з винаходом. У відповідь на вхідні дискретні значення аудіоданих кодер 150 генерує кодовані аудіодані (що представляють один або кілька вхідних звукових каналів) і затверджує кодовані аудіодані в підсистемі 151 доставки.

5 Підсистема 151 доставки сконфігурована для зберігання кодованих аудіоданих і/або для передачі сигналу, що служить ознакою кодованих аудіоданих. Підсистема 151 за фіг. 10 є такою ж, як підсистема 151 за фіг. 5, і вона може бути реалізована кожним зі способів, яким може бути реалізована підсистема 151 за фіг. 5.

10 Декодер 252 за фіг. 10 містить ввід, підключений для приймання кодованих аудіоданих з підсистеми 151 (наприклад, шляхом зчитування або витягання кодованих аудіоданих з пам'яті підсистеми 151 або шляхом приймання сигналу, що служить ознакою кодованих аудіоданих, який був переданий підсистемою 151).

15 Декодер 252 сконфігурований (наприклад, запрограмований) для витягання (з прийнятого кодованого бітового потоку) кодованих даних, що представляють один або кілька каналів звукової інформації, обробленої багатокаскадним фільтром 153, і для декодування кодованих даних з метою створення декодованих представлень одного або декількох каналів звукової інформації.

20 Згідно з типовими варіантами здійснення винаходу для програмування процесора (наприклад, процесора цифрового оброблення сигналів або процесора загального призначення) з метою реалізації багатокаскадного фільтра використовуються команди SIMD (або команди на паралельне оброблення декількома блоками ALU або AMU). Багатокаскадний фільтр може реалізовувати фільтрацію пропущення нижніх частот з обмеженням смуги пропущення, фільтрацію пропущення нижніх частот (наприклад, у підсистемі LFE аудіокодера), фільтрацію пропущення верхніх частот (наприклад, у підсистемі детектора короткочасних сигналів аудіокодера) або іншу фільтрацію.

25 Інші особливості винаходу включають систему або пристрій (наприклад, кодер, декодер або процесор), сконфігуровані (наприклад, запрограмовані) для виконання будь-якого з варіантів здійснення способу згідно з винаходом, і машинопрочитуваний носій даних (наприклад, диск), на якому зберігається код для реалізації будь-якого з варіантів здійснення способу згідно з винаходом або його етапів. Наприклад, система згідно з винаходом може являти собою або містити в собі програмувальний процесор загального призначення, процесор цифрового оброблення сигналів або мікропроцесор, запрограмовані програмним забезпеченням або програмно-апаратним забезпеченням і/або інакше сконфігурований для виконання однієї з безлічі операцій на даних, у тому числі одного з варіантів здійснення способу згідно з винаходом або його етапів. Такий процесор загального призначення може являти собою або містити в собі комп'ютерну систему, що включає пристрій введення, пам'ять і схему оброблення даних, запрограмовану (і/або інакше сконфігуровану) для виконання одного з варіантів здійснення способу згідно з винаходом (або його етапів) у відповідь на затверджені в ній дані.

40 Деякі варіанти здійснення винаходу являють собою кодери (наприклад, кодери, які кодують аудіодані відповідно до формату Dolby Digital Plus, AC-3 або Dolby E) або декодери, реалізовані як запрограмовані процесори (наприклад, процесори ARM neon, кожний з яких являє собою процесор ARM Cortex з механізмом Neon SIMD, що допускає паралельне оброблення даних, або інші процесори, що містять блоки SIMD ("один потік команд - багато потоків даних") і/або кілька блоків ALU (арифметичних логічних блоків) або AMU (блоків арифметичного оброблення)) або запрограмовані (і/або інакше сконфігуровані) процесори цифрового оброблення сигналів (наприклад, процесори DSP, що містять блоки SIMD і/або кілька блоків ALU або AMU).

50 Об'єднання операцій усіх каскадів біквадратних фільтрів багатокаскадного біквадратного фільтра (у єдиному циклі оброблення дискретних значень) згідно з типовими варіантами здійснення винаходу підвищує продуктивність, роблячи можливою паралельність. Шляхом реалізації одного з варіантів здійснення способу згідно з винаходом, процесори із блоками SIMD і декількома блоками ALU (або AMU) можуть ефективно використовувати свої ресурси.

55 Типові варіанти здійснення способу згідно з винаходом, призначені для реалізації багатокаскадного біквадратного фільтра, не стосуються точності виводу або стійкості фільтра (відносно точності та стійкості, що досягаються традиційною реалізацією фільтра).

60 Випробування, виконані автором винаходу, показали, що кодер, сконфігурований для кодування аудіоданих відповідно до формату Dolby Digital Plus і реалізований як процесор цифрового оброблення сигналів Texas Instruments C64, запрограмований для включення одного з варіантів здійснення двокаскадного біквадратного фільтра згідно з винаходом (пропущення, що реалізує фільтрацію верхніх частот у підсистемі детектора короткочасних сигналів кодера),

вимагає для фільтрації типового блоку аудіоданих лише 1846 циклів, на відміну від середньої кількості циклів (4141), необхідних для фільтрації блоку, коли кодер замість цього запрограмований традиційно так, щоб він включав традиційну (не розпаралелену) реалізацію двокаскадного фільтра.

Випробування, виконані автором винаходу, також показали, що кодер, сконфігурований для кодування аудіоданих відповідно до формату Dolby Digital Plus і реалізований як процесор цифрового оброблення сигналів Texas Instruments C64, запрограмований для включення одного з варіантів чотирьохкаскадного біквадратного фільтра згідно з винаходом (пропущення, що реалізує фільтрацію нижніх частот у підсистемі низькочастотних ефектів ("LFE") кодера), вимагає для фільтрації типового блоку аудіоданих у середньому лише 5802 циклів, на відміну від середньої кількості циклів (10375), необхідних для фільтрації блоку, коли кодер замість цього запрограмований традиційно так, щоб він включав традиційну (не розпаралелену) реалізацію чотирьохкаскадного фільтра.

Очікується, що винахід також може забезпечувати аналогічні переваги відносно продуктивності, коли винахідницький фільтр реалізується відповідним програмуванням інших процесорів (що мають інші архітектури процесорного ядра). Також очікується, що ступінь підвищення продуктивності буде залежати від архітектури процесора, кількості каскадів фільтра та кількості полюсів у фільтрі.

Винахід може бути реалізований в апаратному забезпеченні, апаратно-програмному забезпеченні або програмному забезпеченні, а також як їхній комбінації (наприклад, як програмувальна логічна матриця). Якщо не обумовлене інше, алгоритми або процеси, включені як частина винаходу, по суті не відносяться до якого-небудь конкретного комп'ютера або іншого пристрою. Зокрема, різні машини загального призначення можуть використовуватися з програмами, написаними відповідно до ідей цього розкриття, або для виконання необхідних етапів способу може бути більш зручним створення більш спеціалізованих пристроїв (наприклад, інтегральних мікросхем). Таким чином, винахід може бути реалізований в одній або декількох комп'ютерних програмах, що виконуються на одній або декількох програмувальних комп'ютерних системах (наприклад, на комп'ютерній системі, яка реалізує кодер за фіг. 5), кожна з яких включає щонайменше один процесор, щонайменше одну систему зберігання даних (у тому числі енергозалежну та енергонезалежну пам'ять і/або запам'ятовувальні елементи), щонайменше один пристрій або порт введення та щонайменше один пристрій або порт виведення. Програмний код застосовується до вхідних даних для виконання функцій, описаних у цьому розкритті та для генерування вихідної інформації. Вихідна інформація відомим чином застосовується до одного або декількох пристроїв виведення.

Для встановлення зв'язку з комп'ютерною системою кожна така програма може бути реалізована на будь-якій бажаній комп'ютерній мові (включаючи машинні, асемблерні або вискорівневі процедурні, логічні або об'єктно-орієнтовані мови програмування). У кожному разі, мова може бути компільовуваною або інтерпретовуваною мовою.

Наприклад, різні функції та етапи варіантів здійснення винаходу, коли вони реалізуються послідовностями команд комп'ютерного програмного забезпечення, можуть бути реалізовані багатопотоковими послідовностями команд програмного забезпечення, що запускаються в підходящому апаратному забезпеченні цифрового оброблення сигналів, і, у цьому випадку, різні пристрої, етапи та функції варіантів здійснення винаходу можуть відповідати частинам команд програмного забезпечення.

Кожна така комп'ютерна програма переважно зберігається або завантажується на носій даних або пристрій (наприклад, твердотільну пам'ять або носії, або магнітні, або оптичні носії), придатні для зчитування спеціалізованим програмувальним комп'ютером або програмувальним комп'ютером загального призначення з метою конфігурування та приведення в дію комп'ютера, коли носій даних або пристрій зчитується комп'ютерною системою для виконання процедур, описуваних у цьому розкритті. Система згідно з винаходом також може бути реалізована як машинопрочитуваний носій даних, що сконфігурований (наприклад, що зберігається в пам'яті) комп'ютерною програмою, де носій даних, сконфігурований таким чином, викликає дію комп'ютерної системи обумовленим і попередньо визначеним чином з метою виконання функцій, описуваних у цьому розкритті.

Передбачається, що багатокаскадні фільтри, окремі каскади яких являють собою IIR-фільтри, але не біквадратні фільтри (якими вони є в приватних варіантах здійснення винаходу, описаних у цьому розкритті), можуть бути реалізовані згідно з винаходом так, щоб оброблення даних їхніх окремих каскадів було розпаралелено (наприклад, щоб усі їхні каскади були здатні діяти незалежно у відповідь на єдиний, загальний потік команд для виконання повністю розпаралеленого оброблення даних на зазначених каскадах). Наприклад, багатокаскадний

фільтр, що відноситься до типу, описаного в опублікованій заявці на патент США №2012/0019723 A1, опублікованої 26 січня 2012 р., може бути модифікований відповідно до одного з варіантів здійснення цього винаходу так, щоб оброблення даних його окремих каскадів було розпаралелене (наприклад, щоб усі його каскади діяли незалежно у відповідь на єдиний, загальний потік команд з метою виконання повністю розпаралеленого оброблення даних у зазначених каскадах).

У деяких варіантах здійснення способу згідно з винаходом деякі або всі етапи, описані в цьому розкритті, виконуються одночасно або в іншому порядку, відмінному від зазначеного в прикладах, описаних у цьому розкритті. І хоча в деяких варіантах здійснення способу згідно з винаходом етапи виконуються в конкретному порядку, деякі етапи можуть виконуватися одночасно або в іншому порядку в інших варіантах здійснення винаходу.

Було описано кілька варіантів здійснення винаходу. Проте слід розуміти, що без відступу від суті й обсягу винаходу можуть здійснюватися різні модифікації. У світлі наведених вище ідей можливі численні модифікації й зміни цього винаходу. Слід розуміти, що в межах обсягу прикладеної формули винаходу винахід може застосовуватися на практиці інакше, ніж це описано конкретно в цьому розкритті.

#### ФОРМУЛА ВИНАХОДУ

1. Багатокаскадний фільтр, що містить: буферну пам'ять; щонайменше два каскади біквадратних фільтрів, що містять перший каскад біквадратного фільтра та наступний каскад біквадратного фільтра; і контролер, пов'язаний з каскадами біквадратних фільтрів і сконфігурований для затвердження єдиного потоку команд як у перший каскад біквадратного фільтра, так і в наступний каскад біквадратного фільтра, при цьому зазначений перший каскад біквадратного фільтра та зазначений наступний каскад біквадратного фільтра діють незалежно та паралельно у відповідь на потік команд, при цьому перший каскад біквадратного фільтра пов'язаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на блоці з N вхідних дискретних значень у відповідь на потік команд з метою генерування проміжних значень і для затвердження проміжних значень у пам'яті, при цьому проміжні значення включають піддану фільтрації версію кожного з щонайменше підмножини вхідних дискретних значень, і при цьому наступний каскад біквадратного фільтра пов'язаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на буферизованих значеннях, що витягаються з пам'яті у відповідь на потік команд, з метою генерування блока вихідних значень, при цьому вихідні значення містять вихідне значення, що відповідає кожному з вхідних дискретних значень у блоці з N вхідних дискретних значень, і буферизовані значення містять щонайменше деякі з проміжних значень, згенерованих у першому каскаді біквадратного фільтра у відповідь на блок з N вхідних дискретних значень.

2. Багатокаскадний фільтр за п. 1, який **відрізняється** тим, що зазначений багатокаскадний фільтр сконфігуровано для виконання багатокаскадної фільтрації на блоці з N вхідних дискретних значень у єдиному циклі оброблення даних з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

3. Багатокаскадний фільтр за п. 1, який **відрізняється** тим, що багатокаскадний фільтр містить M каскадів біквадратних фільтрів, при цьому наступний каскад біквадратного фільтра сконфігуровано для генерування вихідного значення, що відповідає "j"-му з вхідних дискретних значень, у відповідь на підмножину буферизованих значень, витягнутих з пам'яті, де j - індекс в інтервалі від M-1 до N-1, при цьому зазначена підмножина містить піддану фільтрації версію "j"-го з вхідних дискретних значень, піддану фільтрації версію "j-1"-го з вхідних дискретних значень і піддану фільтрації версію "j-2"-го з вхідних дискретних значень.

4. Багатокаскадний фільтр за п. 1, який **відрізняється** тим, що багатокаскадний фільтр містить M каскадів біквадратних фільтрів, при цьому наступний каскад біквадратного фільтра сконфігуровано для генерування вихідного значення, що відповідає "j"-му з вхідних дискретних значень, у відповідь на підмножину буферизованих значень, витягнутих з пам'яті, де j - індекс в інтервалі від M-1 до N-1, при цьому зазначена підмножина містить піддану фільтрації версію "j"-го з вхідних дискретних значень, згенеровану першим каскадом біквадратного фільтра, і значення, згенероване наступним каскадом біквадратного фільтра у відповідь на "j-1"-е з вхідних дискретних значень, а також значення, згенероване наступним каскадом біквадратного фільтра у відповідь на "j-2"-е з вхідних дискретних значень.

5. Багатокаскадний фільтр за п. 1, який **відрізняється** тим, що наступний каскад біквадратного фільтра сконфігуровано для генерування вихідного значення, що відповідає кожному з вхідних дискретних значень у відповідь на іншу підмножину буферизованих значень, витягнутих з пам'яті, при цьому кожна зазначена підмножина містить щонайменше три з проміжних значень, згенерованих у першому каскаді біквадратного фільтра та витягнутих з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування.

6. Спосіб виконання багатокаскадної фільтрації на блоці з N вхідних дискретних значень, при цьому зазначений спосіб включає етапи:

(а) виконання першої операції біквадратної фільтрації на блоці з N вхідних дискретних значень з метою генерування проміжних значень і затвердження проміжних значень у буферній пам'яті, при цьому проміжні значення містять піддану фільтрації версію кожного з щонайменше підмножини вхідних дискретних значень; і

(b) виконання другої операції біквадратної фільтрації на буферизованих значеннях, витягнутих з пам'яті, з метою генерування блока вихідних значень, при цьому вихідні значення містять вихідне значення, що відповідає кожному з вхідних дискретних значень у блоці з N вхідних дискретних значень, іншу підмножину буферизованих значень витягають і фільтрують з метою генерування вихідного значення, що відповідає кожному з вхідних дискретних значень у блоці, і кожна зазначена підмножина буферизованих значень містить щонайменше два з проміжних значень, згенерованих у ході виконання етапу (а), які витягають з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування,

при цьому етапи (а) і (b) виконують у відповідь на єдиний потік команд так, що етапи (а) і (b) виконують незалежно та паралельно у відповідь на єдиний потік команд.

7. Спосіб за п. 6, який **відрізняється** тим, що багатокаскадну фільтрацію блока вхідних дискретних значень виконують у єдиному циклі з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

8. Спосіб за п. 6, який **відрізняється** тим, що багатокаскадну фільтрацію виконують у багатокаскадному фільтрі, що містить M каскадів, буферизовані значення, що витягають на етапі (b) для генерування вихідного значення, що відповідає "j"-му з вхідних дискретних значень, де "j" - індекс в інтервалі від M-1 до N-1, містять піддану фільтрації версію "j"-го з вхідних дискретних значень, згенеровану на етапі (а), піддану фільтрації версію "j-1"-го з вхідних дискретних значень, згенеровану на етапі (а), і піддану фільтрації версію "j-2"-го з вхідних дискретних значень, згенеровану на етапі (а).

9. Аудіокодер, сконфігурований для генерування кодованих аудіоданих у відповідь на вхідні аудіодані, при цьому зазначений кодер містить щонайменше один багатокаскадний фільтр, підключений і сконфігурований для фільтрації аудіоданих, при цьому багатокаскадний фільтр містить:

буферну пам'ять;

щонайменше два каскади біквадратних фільтрів, що містять перший каскад біквадратного фільтра й наступний каскад біквадратного фільтра; і

контролер, пов'язаний з каскадами біквадратних фільтрів і сконфігурований для затвердження єдиного потоку команд як у перший каскад біквадратного фільтра, так і в наступний каскад біквадратного фільтра, при цьому зазначений перший каскад біквадратного фільтра й зазначений наступний каскад біквадратного фільтра діють незалежно та паралельно у відповідь на потік команд,

при цьому перший каскад біквадратного фільтра пов'язаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на блоці з N вхідних дискретних значень у відповідь на потік команд з метою генерування проміжних значень і для затвердження проміжних значень у пам'яті, при цьому проміжні значення містять піддану фільтрації версію кожного з щонайменше підмножини вхідних дискретних значень, і

при цьому наступний каскад біквадратного фільтра пов'язаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на буферизованих значеннях, що витягаються з пам'яті у відповідь на потік команд, з метою генерування блока вихідних значень, при цьому вихідні значення містять вихідне значення, що відповідає кожному з вхідних дискретних значень у блоці з N вхідних дискретних значень, і буферизовані значення містять щонайменше деякі з проміжних значень, згенерованих у першому каскаді біквадратного фільтра у відповідь на блок з N вхідних дискретних значень.

10. Кодер за п. 9, який **відрізняється** тим, що багатокаскадний фільтр сконфігурований для виконання багатокаскадної фільтрації на блоці з N вхідних дискретних значень у єдиному циклі оброблення даних з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

11. Кодер за п. 9, який **відрізняється** тим, що багатокаскадний фільтр містить М каскадів біквадратних фільтрів, при цьому наступний каскад біквадратного фільтра сконфігурований для генерування вихідного значення, що відповідає "j"-му з вхідних дискретних значень, у відповідь на підмножину буферизованих значень, витягнутих з пам'яті, де j - індекс в інтервалі від М-1 до N-1, при цьому зазначена підмножина містить піддану фільтрації версію "j"-го з вхідних дискретних значень, піддану фільтрації версію "j-1"-го з вхідних дискретних значень і піддану фільтрації версію "j-2"-го з вхідних дискретних значень.

12. Кодер за п. 9, який **відрізняється** тим, що багатокаскадний фільтр містить М каскадів біквадратних фільтрів, при цьому наступний каскад біквадратного фільтра сконфігурований для генерування вихідного значення, що відповідає "j"-му з вхідних дискретних значень, у відповідь на підмножину буферизованих значень, витягнутих з пам'яті, де j - індекс в інтервалі від М-1 до N-1, при цьому зазначена підмножина містить піддану фільтрації версію "j"-го з вхідних дискретних значень, згенеровану першим каскадом біквадратного фільтра, і значення, згенероване наступним каскадом біквадратного фільтра у відповідь на "j-1"-е з вхідних дискретних значень, а також значення, згенероване наступним каскадом біквадратного фільтра у відповідь на "j-2"-е з вхідних дискретних значень.

13. Кодер за п. 9, який **відрізняється** тим, що наступний каскад біквадратного фільтра сконфігурований для генерування вихідного значення, що відповідає кожному з вхідних дискретних значень у відповідь на іншу підмножину буферизованих значень, витягнутих з пам'яті; при цьому кожна зазначена підмножина містить щонайменше три із проміжних значень, згенерованих у першому каскаді біквадратного фільтра та витягнутих з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування.

14. Кодер за п. 9, який **відрізняється** тим, що зазначений кодер являє собою процесор, що містить щонайменше один блок "один потік команд - багато потоків даних", запрограмований для реалізації багатокаскадного фільтра.

15. Кодер за п. 9, який **відрізняється** тим, що зазначений кодер являє собою процесор, що містить кілька арифметичних логічних блоків, запрограмованих для реалізації каскадів біквадратних фільтрів.

16. Кодер за п. 9, який **відрізняється** тим, що зазначений кодер являє собою процесор, що містить кілька блоків арифметичного оброблення, запрограмованих для реалізації каскадів біквадратних фільтрів.

17. Спосіб кодування аудіоданих для генерування кодованих аудіоданих, що включає виконання багатокаскадної фільтрації на блоці з N дискретних значень аудіоданих, при цьому багатокаскадна фільтрація включає етапи:

(а) виконання першої операції біквадратної фільтрації на блоці з N дискретних значень з метою генерування проміжних значень і твердження цих проміжних значень у буферній пам'яті, при цьому проміжні значення містять піддану фільтрації версію кожного з щонайменше підмножини N дискретних значень; і

(b) виконання другої операції біквадратної фільтрації на буферизованих значеннях, витягнутих з пам'яті, з метою генерування блока вихідних значень, при цьому вихідні значення містять вихідне значення, що відповідає кожному з вхідних дискретних значень у блоці з N дискретних значень, іншу підмножину буферизованих значень витягають і фільтрують з метою генерування вихідного значення, що відповідає кожному з дискретних значень у блоці, і кожна зазначена підмножина буферизованих значень містить щонайменше два з проміжних значень, згенерованих у ході виконання етапу (а), які витягають з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування,

при цьому етапи (а) і (b) виконують у відповідь на єдиний потік команд так, що етапи (а) і (b) виконують незалежно та паралельно у відповідь на цей єдиний потік команд.

18. Спосіб за п. 17, який **відрізняється** тим, що багатокаскадну фільтрацію блока дискретних значень виконують у єдиному циклі з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

19. Спосіб за п. 17, який **відрізняється** тим, що багатокаскадна фільтрація виконується в багатокаскадному фільтрі, що містить М каскадів, при цьому буферизовані значення, що витягаються на етапі (b) для генерування вихідного значення, що відповідає "j"-му з дискретних значень, де j - індекс в інтервалі від М-1 до N-1, містять піддану фільтрації версію "j"-го з дискретних значень, згенеровану на етапі (а), піддану фільтрації версію "j-1"-го з дискретних значень, згенеровану на етапі (а), і піддану фільтрації версію "j-2"-го з дискретних значень, згенеровану на етапі (а).

20. Аудіодекодер, сконфігурований для генерування декодованих аудіоданих у відповідь на кодовані аудіодані, при цьому зазначений декодер містить щонайменше один багатокаскадний

фільтр, підключений і сконфігурований для фільтрації кодованих аудіоданих, при цьому багатокаскадний фільтр містить:

буферну пам'ять;

щонайменше два каскади біквадратних фільтрів, що містять перший каскад біквадратного фільтра та наступний каскад біквадратного фільтра; і

контролер, пов'язаний з каскадами біквадратних фільтрів і сконфігурований для затвердження єдиного потоку команд як у перший каскад біквадратного фільтра, так і в наступний каскад біквадратного фільтра, при цьому зазначений перший каскад біквадратного фільтра та зазначений наступний каскад біквадратного фільтра діють незалежно та паралельно у відповідь на потік команд,

при цьому перший каскад біквадратного фільтра пов'язаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на блоці з  $N$  вхідних дискретних значень у відповідь на потік команд з метою генерування проміжних значень і для затвердження проміжних значень у пам'яті, при цьому проміжні значення містять піддану фільтрації версію кожного з щонайменше підмножини вхідних дискретних значень, і

при цьому наступний каскад біквадратного фільтра пов'язаний з пам'яттю та сконфігурований для виконання біквадратної фільтрації на буферизованих значеннях, що витягаються з пам'яті у відповідь на потік команд, з метою генерування блока вихідних значень, при цьому вихідні значення містять вихідне значення, що відповідає кожному з вхідних дискретних значень у блоці з  $N$  вхідних дискретних значень, і буферизовані значення містять щонайменше деякі з проміжних значень, згенерованих у першому каскаді біквадратного фільтра у відповідь на блок з  $N$  вхідних дискретних значень.

21. Декодер за п. 20, який **відрізняється** тим, що багатокаскадний фільтр сконфігурований для виконання багатокаскадної фільтрації на блоці з  $N$  вхідних дискретних значень у єдиному циклі оброблення даних з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

22. Декодер за п. 20, який **відрізняється** тим, що багатокаскадний фільтр містить  $M$  каскадів біквадратних фільтрів, при цьому наступний каскад біквадратного фільтра сконфігурований для генерування вихідного значення, що відповідає " $j$ "-му з вхідних дискретних значень, у відповідь на підмножину буферизованих значень, витягнутих з пам'яті, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , при цьому зазначена підмножина містить піддану фільтрації версію " $j$ "-го з вхідних дискретних значень, піддану фільтрації версію " $j-1$ "-го з вхідних дискретних значень і піддану фільтрації версію " $j-2$ "-го з вхідних дискретних значень.

23. Декодер за п. 20, який **відрізняється** тим, що багатокаскадний фільтр містить  $M$  каскадів біквадратних фільтрів, при цьому наступний каскад біквадратного фільтра сконфігурований для генерування вихідного значення, що відповідає " $j$ "-му з вхідних дискретних значень, у відповідь на підмножину буферизованих значень, витягнутих з пам'яті, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , при цьому зазначена підмножина містить піддану фільтрації версію " $j$ "-го з вхідних дискретних значень, згенеровану першим каскадом біквадратного фільтра, і значення, згенероване наступним каскадом біквадратного фільтра у відповідь на " $j-1$ "-е з вхідних дискретних значень, і значення, згенероване наступним каскадом біквадратного фільтра у відповідь на " $j-2$ "-е з вхідних дискретних значень.

24. Декодер за п. 20, який **відрізняється** тим, що наступний каскад біквадратного фільтра сконфігурований для генерування вихідного значення, що відповідає кожному з вхідних дискретних значень у відповідь на іншу підмножину буферизованих значень, витягнутих з пам'яті, при цьому кожна зазначена підмножина містить щонайменше три з проміжних значень, згенерованих у першому каскаді біквадратного фільтра та витягнутих з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування.

25. Декодер за п. 20, який **відрізняється** тим, що зазначений декодер являє собою процесор, що містить щонайменше один блок "один потік команд - багато потоків даних", запрограмований для реалізації багатокаскадного фільтра.

26. Декодер за п. 20, який **відрізняється** тим, що зазначений декодер являє собою процесор, що містить кілька арифметичних логічних блоків, запрограмованих для реалізації каскадів біквадратних фільтрів.

27. Декодер за п. 20, який **відрізняється** тим, що зазначений декодер являє собою процесор, що містить кілька блоків арифметичного оброблення, запрограмованих для реалізації каскадів біквадратних фільтрів.

28. Спосіб декодування кодованих аудіоданих з метою генерування декодованих аудіоданих, що включає виконання багатокаскадної фільтрації на блоці з  $N$  дискретних значень аудіоданих, при цьому багатокаскадна фільтрація включає етапи:

(а) виконання першої операції біквадратної фільтрації на блоці з  $N$  дискретних значень з метою генерування проміжних значень і затвердження проміжних значень у буферній пам'яті, при цьому проміжні значення містять піддану фільтрації версію кожного з щонайменше підмножини  $N$  дискретних значень; і

5 (b) виконання другої операції біквадратної фільтрації на буферизованих значеннях, витягнутих з пам'яті, з метою генерування блока вихідних значень, при цьому вихідні значення містять вихідне значення, що відповідає кожному з дискретних значень у блоці з  $N$  дискретних значень, іншу підмножину буферизованих значень витягають і фільтрують з метою генерування вихідного значення, що відповідає кожному з дискретних значень у блоці, і при цьому кожна  
10 зазначена підмножина буферизованих значень містить щонайменше два з проміжних значень, згенерованих у ході виконання етапу (а), які витягають з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування,  
при цьому етапи (а) і (b) виконують у відповідь на єдиний потік команд так, що етапи (а) і (b) виконують незалежно та паралельно у відповідь на єдиний потік команд.

15 29. Спосіб за п. 28, який **відрізняється** тим, що багатокаскадну фільтрацію блока дискретних значень виконують у єдиному циклі з повторенням за індексом дискретних значень, але без повторення за індексом каскадів біквадратних фільтрів.

30. Спосіб за п. 28, який **відрізняється** тим, що багатокаскадну фільтрацію виконують у багатокаскадному фільтрі, що містить  $M$  каскадів, при цьому буферизовані значення, що  
20 витягаються на етапі (b) для генерування вихідного значення, що відповідає "j"-му з дискретних значень, де  $j$  - індекс в інтервалі від  $M-1$  до  $N-1$ , включають піддану фільтрації версію "j"-го з дискретних значень, згенеровану на етапі (а), піддану фільтрації версію "j-1"-го з дискретних значень, згенеровану на етапі (а), і піддану фільтрації версію "j-2"-го з дискретних значень, згенеровану на етапі (а).

25 31. Спосіб оброблення кодованого бітового потоку, що включає:

приймання кодованого бітового потоку та витягання з нього кодованих даних, що представляють один або кілька каналів звукової інформації, обробленої багатокаскадним фільтром; і

30 декодування кодованих даних з метою створення декодованих представлень одного або декількох каналів звукової інформації, обробленої багатокаскадним фільтром, при цьому багатокаскадний фільтр діє для незалежного та паралельного виконання у відповідь на єдиний потік команд:

(а) першої операції біквадратної фільтрації на блоці з  $N$  дискретних значень з метою генерування проміжних значень для затвердження в буферній пам'яті, при цьому проміжні  
35 значення містять піддану фільтрації версію кожного з щонайменше підмножини з  $N$  дискретних значень; і

(b) другої операції біквадратної фільтрації на буферизованих значеннях, що витягаються з пам'яті, з метою генерування блока вихідних значень, при цьому вихідні значення містять вихідне значення, що відповідає кожному з дискретних значень у блоці з  $N$  дискретних значень,  
40 іншу підмножину буферизованих значень витягають і буферизують з метою генерування вихідного значення, що відповідає кожному з дискретних значень у блоці, і при цьому кожна зазначена підмножина буферизованих значень містить щонайменше два з проміжних значень, згенерованих в ході виконання етапу (а), які витягають з пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування.

45 32. Спосіб за п. 31, який **відрізняється** тим, що багатокаскадний фільтр являє собою трикаскадний біквадратний фільтр, призначений для фільтрації пропущення нижніх частот з обмеженням смуги пропущення.

33. Спосіб за п. 31, який **відрізняється** тим, що один з одного або декількох каналів являє собою канал LFE і багатокаскадний фільтр являє собою чотирикаскадний біквадратний фільтр,  
50 призначений для виконання фільтрації пропущення нижніх частот на каналі LFE.

34. Пристрій, призначений для оброблення даних кодованого бітового потоку, що містить:

ввід, підключений для приймання кодованого бітового потоку; і

35 декодер, пов'язаний з вводом і сконфігурований для витягання з кодованого бітового потоку кодованих даних, що представляють один або кілька каналів звукової інформації, обробленої багатокаскадним фільтром, і для декодування кодованих даних з метою створення декодованих виражень одного або декількох каналів звукової інформації, обробленої багатокаскадним  
фільтром, при цьому багатокаскадний фільтр діє для незалежного та паралельного виконання у відповідь на єдиний потік команд:

60 (а) першої операції біквадратної фільтрації на блоці з  $N$  дискретних значень з метою генерування проміжних значень для затвердження в буферній пам'яті, при цьому проміжні

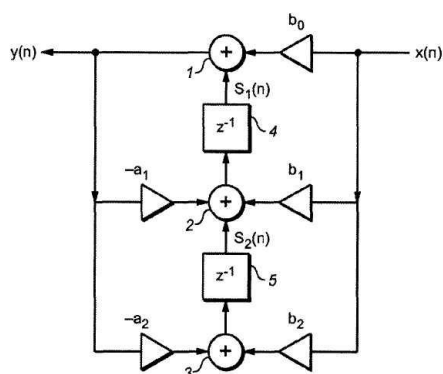


значення містять піддану фільтрації версію кожного з щонайменше підмножини з  $N$  дискретних значень; і

(b) другої операції біквадратної фільтрації на буферизованих значеннях, витягнутих з пам'яті, з метою генерування блока вихідних значень, при цьому вихідні значення містять вихідне значення, що відповідає кожному з дискретних значень у блоці з  $N$  дискретних значень, іншу підмножину буферизованих значень витягають і фільтрують з метою генерування вихідного значення, що відповідає кожному з дискретних значень у блоці, і при цьому кожна зазначена підмножина буферизованих значень містить щонайменше два з проміжних значень, згенерованих в ході виконання етапу (a), які витягають із пам'яті після перебування в зазначеній пам'яті протягом різного часу очікування.

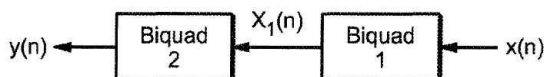
35. Пристрій за п. 34, який **відрізняється** тим, що багатокаскадний фільтр являє собою трикаскадний біквадратний фільтр, призначений для фільтрації пропущення нижніх частот з обмеженням смуги пропущення.

36. Пристрій за п. 34, який **відрізняється** тим, що один з одного або декількох каналів являє собою канал LFE і багатокаскадний фільтр являє собою чотирикаскадний біквадратний фільтр, призначений для виконання фільтрації пропущення нижніх частот на каналі LFE.



Фіг. 1

(відомий рівень техніки)



Фіг. 1A

(відомий рівень техніки)

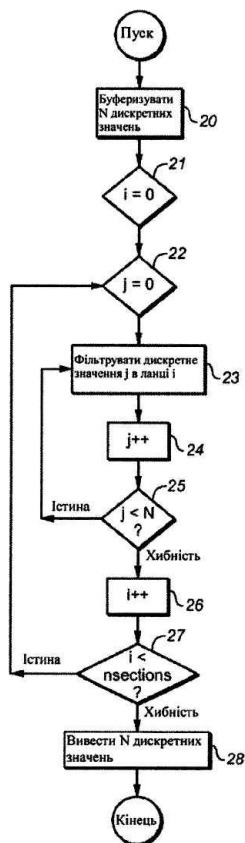


Fig. 2

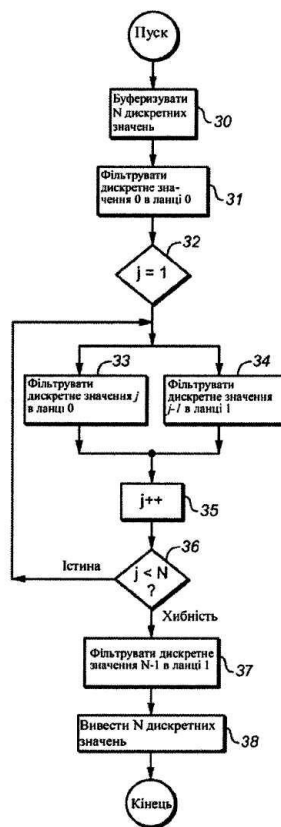


Fig. 3

(відомий рівень техніки)

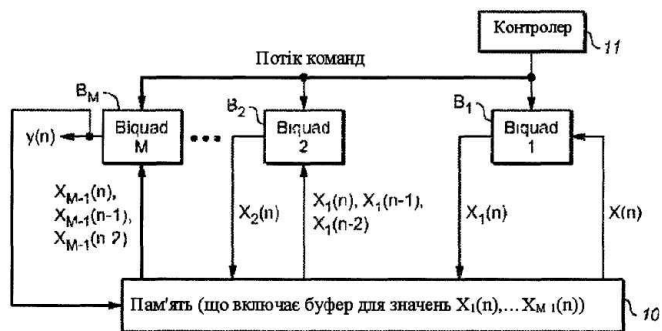
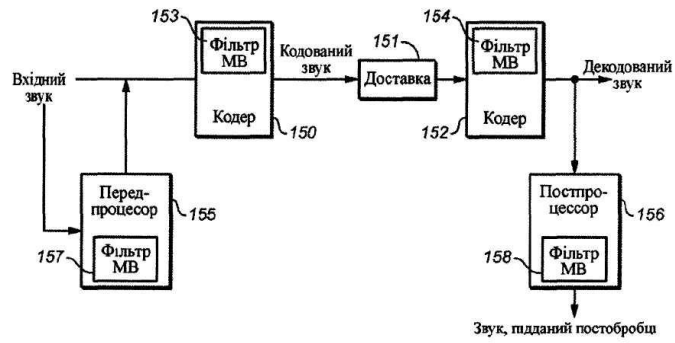
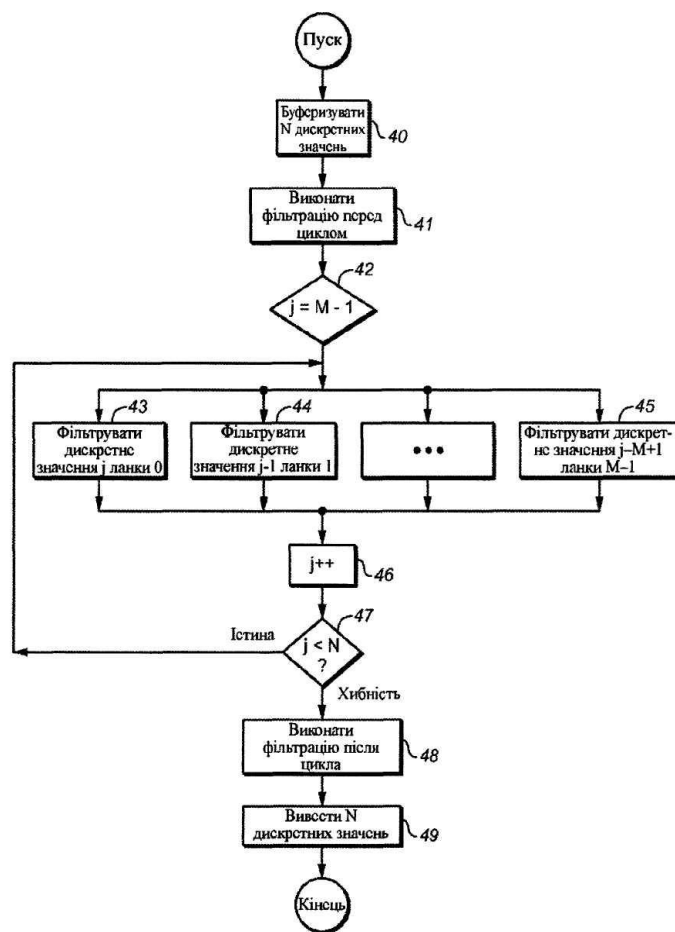


Fig. 4



Фіг. 5



Фіг. 6

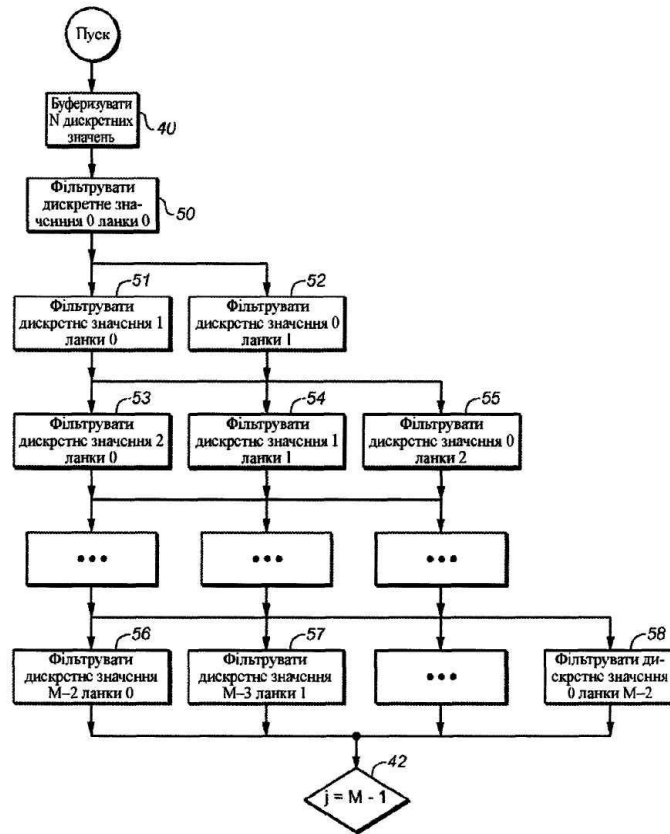


Fig. 7

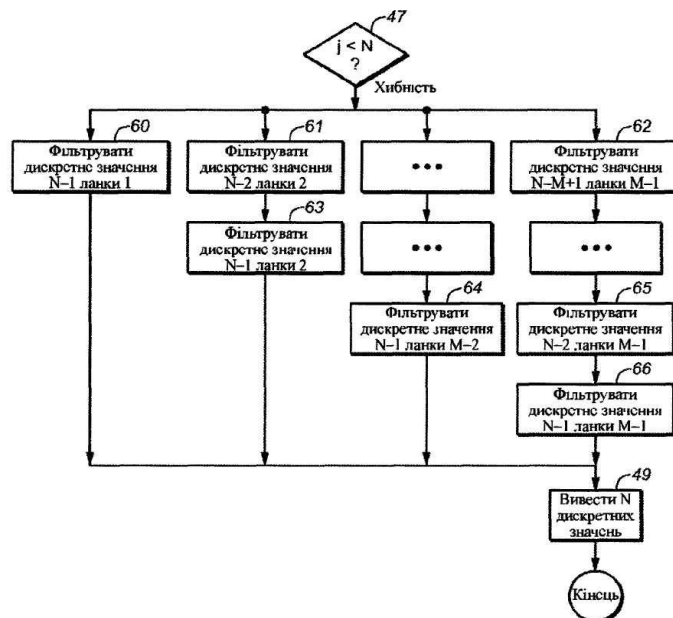
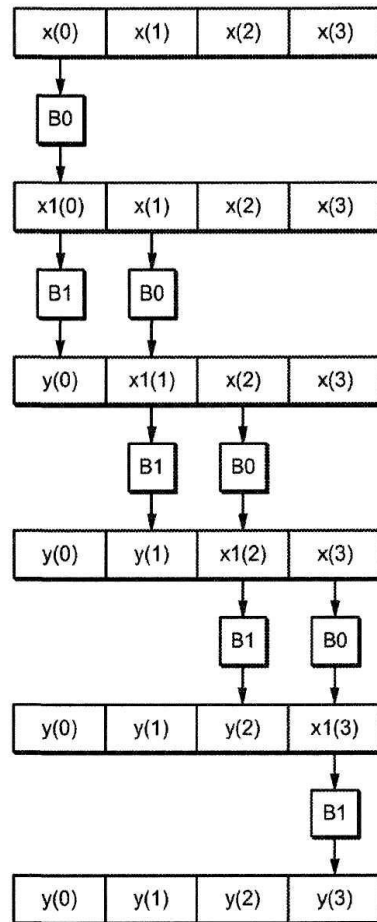
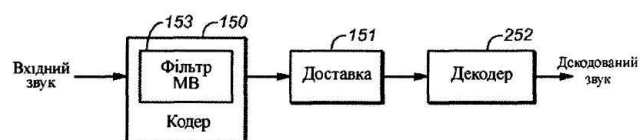


Fig. 8



Фиг. 9



Фиг. 10