



(12) **BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ**

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) 
1-0021477

(51)⁷ **H03H 17/04**

(13) **B**

(21) 1-2014-03745

(22) 17.04.2013

(86) PCT/US2013/036932 17.04.2013

(87) WO2013/169450A1 14.11.2013

(30) 61/645,291 10.05.2012 US

(45) 26.08.2019 377

(43) 25.02.2015 323

(73) DOLBY LABORATORIES LICENSING CORPORATION (US)

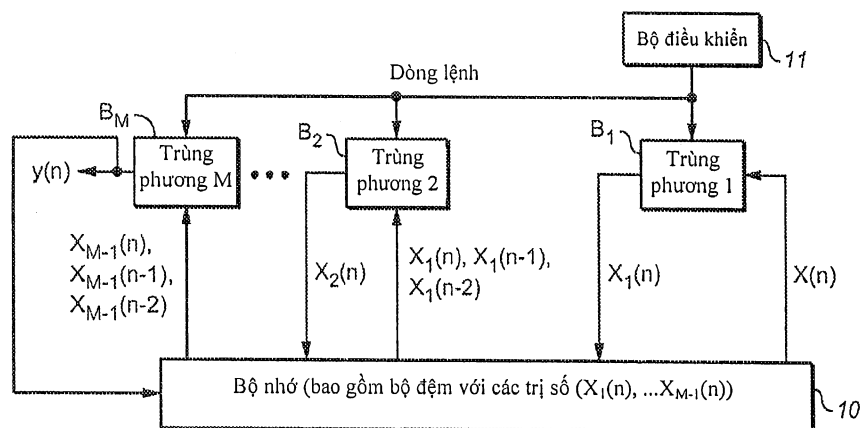
1275 Market Street, San Francisco, California 94103, United States of America.

(72) RATHI, Khushbu P. (IN)

(74) Công ty TNHH Quốc tế D & N (D&N INTERNATIONAL CO.,LTD.)

(54) **BỘ MÃ HÓA ÂM THANH, PHƯƠNG PHÁP MÃ HÓA DỮ LIỆU ÂM THANH, PHƯƠNG PHÁP VÀ THIẾT BỊ XỬ LÝ DÒNG BIT ĐƯỢC MÃ HÓA**

(57) Sáng chế đề cập đến bộ mã hóa âm thanh bao gồm bộ lọc đa tầng có các tầng lọc trùng phương được kết hợp với khoảng thời gian chờ giữa các tầng này và các phương pháp lọc trùng phương đa tầng. Theo các phương án thông thường, tất cả các tầng lọc trùng phương của bộ lọc có thể hoạt động độc lập để thực hiện việc xử lý dữ liệu được song song hóa hoàn toàn. Theo một số phương án, bộ lọc đa tầng theo sáng chế bao gồm bộ nhớ đệm, ít nhất hai tầng lọc trùng phương, và bộ điều khiển được nối và được tạo cấu hình để gán một dòng lệnh duy nhất cho các tầng lọc. Thông thường, bộ lọc đa tầng được tạo cấu hình để thực hiện lọc đa tầng khối các mẫu đầu vào trong một vòng lặp xử lý có sự lặp lại trên chỉ số mẫu nhưng không lặp lại trên chỉ số tầng lọc trùng phương. Sáng chế còn đề cập đến phương pháp mã hóa dữ liệu âm thanh, thiết bị và phương pháp xử lý dòng bit được mã hóa.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các bộ lọc đa tầng bao gồm các tầng lọc trùng phương, và đến việc lọc song song dữ liệu (ví dụ, dữ liệu âm thanh) bằng cách sử dụng các bộ lọc này. Một số phương án của sáng chế đề xuất các phương pháp, hệ thống, và bộ xử lý để lọc dữ liệu âm thanh (sử dụng bộ lọc đa tầng bao gồm các tầng lọc trùng phương) trong khi mã hóa hoặc giải mã dữ liệu theo một trong số các định dạng đã biết như Dolby Digital (AC-3), Dolby Digital Plus (E-AC-3), và Dolby E, hoặc theo một định dạng mã hóa khác. Dolby, Dolby Digital, Dolby Digital Plus, và Dolby E đều là các nhãn hiệu của Dolby Laboratories Licensing Corporation.

Tình trạng kỹ thuật của sáng chế

Trong toàn bộ tài liệu này, bao gồm các điểm yêu cầu bảo hộ, cụm từ thực hiện hoạt động “trên” các tín hiệu hoặc dữ liệu (ví dụ, lọc hoặc chia tỷ lệ tín hiệu hoặc dữ liệu) được sử dụng theo nghĩa rộng để chỉ việc thực hiện hoạt động trực tiếp trên các tín hiệu hoặc dữ liệu, hoặc trên các phiên bản đã qua xử lý của tín hiệu hoặc dữ liệu (ví dụ, trên các phiên bản của tín hiệu đã trải qua quy trình lọc sơ bộ hoặc quy trình xử lý khác trước khi thực hiện hoạt động trên đó).

Trong xử lý tín hiệu, bộ lọc trùng phương số là bộ lọc tuyến tính đệ quy cấp hai, gồm hai điểm cực và hai điểm không. Cách viết tắt bộ lọc "biquad" (hoặc "bi-quad") sẽ được sử dụng trong bản mô tả này để chỉ bộ lọc trùng phương số. Trong miền Z, hàm chuyển của bộ lọc trùng phương là tỷ lệ của hai hàm bậc hai:

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

Các bộ lọc đệ quy cấp cao (các bộ lọc đáp ứng xung vô hạn hay bộ lọc IIR (infinite impulse response) thuộc cấp cao hơn cấp thứ hai) có thể rất nhạy với việc lượng tử hóa các hệ số của chúng, và có thể dễ dàng trở nên không ổn định. Các bộ

lọc đệ quy cấp thứ một và hai cũng có thể có các vấn đề không ổn định thuộc loại này nhưng các vấn đề không ổn định đó ít nghiêm trọng hơn nhiều. Do đó, các bộ lọc đệ quy cấp cao thường được thực hiện dưới dạng các bộ lọc nhiều tầng tuần tự bao gồm tầng tuần tự của các đoạn trùng phương (và tùy ý còn là bộ lọc cấp thứ nhất). Các bộ lọc nhiều tầng tuần tự như vậy đôi khi trong bản mô tả này được gọi là các bộ lọc trùng phương đa tầng, và bao gồm một dãy các bộ lọc trùng phương (đôi khi trong bản mô tả này được gọi là các tầng trùng phương hoặc các đoạn trùng phương).

Ví dụ, các bộ mã hóa thông thường được tạo cấu hình để mã hóa dữ liệu âm thanh theo định dạng AC-3 (Dolby Digital) đã biết rộng rãi, hoặc một trong số các định dạng Dolby Digital Plus và Dolby E đã biết rộng rãi, thiết lập nhiều bộ lọc trùng phương đa tầng. Ví dụ, bộ mã hóa Dolby Digital Plus thông thường sử dụng bộ lọc trùng phương hai tầng (tức là, bộ lọc bao gồm hai bộ lọc trùng phương nhiều tầng) để thực hiện lọc thông cao trong hệ thống con dò tạm thời, bộ lọc trùng phương bốn tầng (tức là, bộ lọc bao gồm bốn bộ lọc trùng phương nhiều tầng) để thực hiện lọc thông thấp trong hệ thống con hiệu ứng tần số thấp (low frequency effect - LFE), và bộ lọc trùng phương ba tầng để thực hiện lọc thông thấp hạn chế băng thông. Bộ mã hóa Dolby E thường sử dụng bộ lọc trùng phương hai tầng (tức là, bộ lọc bao gồm hai bộ lọc trùng phương nhiều tầng) để thực hiện lọc thông cao trong hệ thống con dò tạm thời, và bộ lọc trùng phương bốn tầng (tức là, bộ lọc bao gồm bốn bộ lọc trùng phương nhiều tầng) để thực hiện lọc thông thấp trong các hệ thống con LFE. Bộ giải mã Dolby E thường sử dụng bộ lọc trùng phương ba tầng (tức là, bộ lọc bao gồm ba bộ lọc trùng phương nhiều tầng) thực hiện lọc thông thấp trong hệ thống con LFE.

Ví dụ, Fig.1 là sơ đồ thể hiện bộ lọc trùng phương (thuộc kiểu mà đôi khi được gọi là Dạng Trục tiếp II – Cấu trúc Chuyển vị), bao gồm các phần tử 1, 2, 3, 4, 5, b_0 , b_1 , b_2 , $-a_1$, và $-a_2$, được kết nối như được thể hiện trên hình vẽ. Các phần tử 1, 2, và 3 là các phần tử bổ sung, các phần tử 4 và 5 là các phần tử trễ, và mỗi phần tử khuếch đại b_0 , b_1 , b_2 , $-a_1$, và $-a_2$ áp dụng một khuếch đại tương ứng trong số các khuếch đại b_0 , b_1 , b_2 , $-a_1$, và $-a_2$ cho tín hiệu được gán cho đầu vào của nó. Mặc dù không được thể hiện hoặc mô tả trong bản mô tả này, nhưng các chuyên gia

trong lĩnh vực này đều biết rõ rằng các cấu trúc bộ lọc trùng phương, tương đương, khác có tồn tại, ví dụ như Dạng Trục tiếp I, Dạng Trục tiếp I – Chuyển vị, và Dạng Trục tiếp II. Các cấu trúc bộ lọc trùng phương tương đương như vậy bất kỳ nằm trong phạm vi của sáng chế.

Như được thể hiện trên Fig.1A, nếu bộ lọc trùng phương trên Fig.1 (được gắn nhãn là Trùng phương 1 trên Fig.1A) được xếp tầng với bộ lọc trùng phương có cùng cấu trúc (được gắn nhãn là Trùng phương 2 trên Fig.1A) nhưng các phân tử khuếch đại của nó có thể áp dụng các độ khuếch đại khác so với các độ khuếch đại của bộ lọc trên Fig.1, thì bộ lọc trùng phương đa tầng thu được là một ví dụ về bộ lọc trùng phương hai tầng có thể được sử dụng (ví dụ, để thực hiện việc lọc thông cao trong hệ thống con dò tạm thời của bộ mã hóa âm thanh như nêu trên). Trong bộ lọc trùng phương đa tầng trên Fig.1A, tín hiệu đầu ra, $x_1(n)$, của tầng thứ nhất là tín hiệu đầu vào của tầng thứ hai.

Đối với các bộ lọc trùng phương đa tầng (và một số bộ lọc IIR đa tầng khác), phép tính mẫu đầu ra trong mỗi tầng tại thời điểm “ n ” (tức là, tín hiệu đầu ra $y(n)$ của tầng này) đáp lại các trị số của tín hiệu miền thời gian $x(n)$ (tín hiệu đầu vào hoặc tín hiệu được tạo ra ở một tầng khác của bộ lọc đa tầng) tại thời điểm “ n ” và các thời điểm trước đó, phụ thuộc vào các đầu ra trước đó (tức là, các đầu ra $y(n-1)$ và $y(n-2)$, tại các thời điểm $n-1$ và $n-2$). Hơn nữa, cứ hai tầng liên tiếp (các bộ lọc trùng phương) trong bộ lọc trùng phương đa tầng, đầu ra của mỗi tầng trước là đầu vào của tầng sau đó, sao cho đầu ra của tầng kế tiếp không thể được xác định cho đến khi đầu ra của tầng sớm hơn được xác định. Đây là lý do chính giải thích tại sao quy trình xử lý song song hóa hoàn toàn không được sử dụng (trước sáng chế) để cài đặt bộ lọc trùng phương đa tầng.

Trong nhiều cấu trúc bộ xử lý lỗi hiện đại ngày nay (ví dụ, các cấu trúc bộ xử lý tín hiệu số), có nhiều đơn vị một lệnh, đa dữ liệu (single instruction, multiple data – SIMD) và/hoặc nhiều đơn vị lôgic số học (arithmetic logic unit - ALU) hoặc đơn vị thao tác số học (arithmetic manipulation unit - AMU) mà có thể được dùng để song song hóa nhiều thuật toán và nâng cao hiệu suất. Tuy nhiên, các thuật toán

thông thường để lập trình các bộ xử lý để cài đặt các bộ lọc trùng phương đa tầng không sử dụng các lệnh SIMD và không được song song hóa.

Ví dụ, các bộ mã hóa Dolby Digital Plus (mã hóa dữ liệu âm thanh theo định dạng Dolby Digital Plus) được cài đặt dưới dạng các bộ xử lý Neon ARM được lập trình (mỗi bộ xử lý này là bộ xử lý Cortex ARM có động cơ SIMD Neon cho phép xử lý song song), và dưới dạng các bộ xử lý tín hiệu số Texas Instruments C64 được lập trình. Nhiều bộ mã hóa dữ liệu âm thanh (ví dụ, các bộ mã hóa mã hóa dữ liệu âm thanh theo các định dạng mã hóa AC-3, Dolby Digital Plus, Dolby E, và/hoặc định dạng mã hóa khác) đã được hoặc có thể được cài đặt làm các bộ xử lý được lập trình có cấu trúc bất kỳ trong số các cấu trúc khác nhau, có các đơn vị một lệnh, đa dữ liệu (single instruction, multiple data - SIMD) và/hoặc nhiều đơn vị logic số học (arithmetic logic units - ALU) hoặc đơn vị thao tác số học (arithmetic manipulation units - AMU). Các bộ xử lý như vậy có thể được lập trình để thực hiện các thuật toán khác nhau (được bao gồm trong quy trình mã hóa dữ liệu âm thanh) bằng cách sử dụng quy trình xử lý song song. Tuy nhiên, việc lập trình thông thường mà đã được sử dụng để áp dụng cho các bộ lọc trùng phương đa tầng trong các bộ xử lý như vậy không áp dụng cho xử lý song song.

Các phương án điển hình của sáng chế sử dụng quy trình xử lý song song để áp dụng cho bộ lọc trùng phương đa tầng. Một số phương án sử dụng quy trình xử lý song song để áp dụng cho bộ lọc trùng phương đa tầng thuộc loại được dùng khi mã hóa dữ liệu âm thanh theo định dạng AC-3 (Dolby Digital), định dạng Dolby Digital Plus, hoặc định dạng Dolby E.

Mặc dù sáng chế không bị giới hạn ở sử dụng khi mã hóa dữ liệu âm thanh theo định dạng AC-3, Dolby Digital Plus, hoặc Dolby E, nhưng theo một số phương án, sáng chế còn đề xuất các phương pháp, hệ thống, và bộ xử lý mã hóa âm thanh (ví dụ, để mã hóa dữ liệu âm thanh theo định dạng AC-3, Dolby Digital Plus, hoặc Dolby E) sử dụng ít nhất một bộ lọc trùng phương đa tầng áp dụng (hoặc được thiết kế theo) phương án của sáng chế.

Dòng bit được mã hóa AC-3 bao gồm từ một đến sáu kênh nội dung âm thanh, và siêu dữ liệu chỉ báo ít nhất một đặc tính của nội dung âm thanh. Nội dung

âm thanh là dữ liệu âm thanh đã được nén bằng cách sử dụng kỹ thuật mã hóa âm thanh cảm quan.

Chi tiết về kỹ thuật mã hóa AC-3 (còn được gọi là Dolby Digital) đã được biết đến rộng rãi và được bộc lộ trong nhiều tài liệu tham khảo đã công bố bao gồm các tài liệu sau:

ATSC Standard A52/A: Digital Audio Compression Standard (AC-3), Revision A, Advanced Television Systems Committee, ngày 20 tháng 8 năm 2001;

Flexible Perceptual Coding for Audio Transmission and Storage,” của Craig C. Todd, et al, 96th *Convention of the Audio Engineering Society*, ngày 26 tháng 2 năm 1994, Preprint 3796;

“Design and Implementation of AC-3 Coders,” của Steve Vernon, *IEEE Trans. Consumer Electronics*, Vol. 41, No. 3, tháng 8 năm 1995;

“Dolby Digital Audio Coding Standards,” chương sách của Robert L. Andersen và Grant A. Davidson trong *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,” bởi Bosi và các đồng tác giả, Audio Engineering Society Preprint 3365, 93rd AES Convention, tháng 10 năm 1992; và

Các Patent Mỹ số 5,583,962; 5,632,005; 5,633,981; 5,727,119; và 6,021,386.

Chi tiết về kỹ thuật mã hóa Dolby Digital (AC-3) và Dolby Digital Plus (đôi khi được gọi là AC-3 Nâng cao hoặc “E-AC-3”) được trình bày trong “Introduction to Dolby Digital Plus, an Enhancement to the Dolby Digital Coding System,” AES Convention Paper 6196, 117th AES Convention, ngày 28 tháng 10 năm 2004, và trong Dolby Digital / Dolby Digital Plus Specification (ATSC A/52:2010), có thể truy cập tại <http://www.atsc.org/cms/index.php/standards/published-standards>.

Bản chất kỹ thuật của sáng chế

Theo một nhóm phương án, sáng chế đề xuất bộ lọc đa tầng bao gồm ít nhất hai tầng (mỗi trong số các tầng này là bộ lọc trùng phương), trong đó các tầng được

kết hợp với khoảng thời gian chờ giữa các tầng này, sao cho tất cả các tầng có thể hoạt động một cách độc lập đáp lại một dòng lệnh chung duy nhất, để thực hiện việc xử lý dữ liệu song song hóa hoàn toàn trong các tầng này. Thông thường, bộ lọc đa tầng còn bao gồm bộ điều khiển được nối để gán dòng lệnh chung cho tất cả các tầng, và bộ nhớ dữ liệu được nối với tất cả các tầng, và tất cả các tầng này có thể hoạt động song song để lọc khối trị số dữ liệu đầu vào đáp lại dòng lệnh chung, nhưng với mỗi tầng trong số các tầng hoạt động dựa trên các trị số dữ liệu khác nhau, và với ít nhất một tầng trong số các tầng hoạt động dựa trên các trị số dữ liệu mà bao gồm các trị số đệm, được tạo ra bởi một tầng khác trong số các tầng này đáp lại tập con gồm các trị số dữ liệu đầu vào và được lưu trữ với các khoảng thời gian chờ khác nhau trong bộ nhớ trước khi được khôi phục để xử lý trong một tầng trong số các tầng nêu trên. Do đó, bộ lọc đa tầng theo các phương án này có cấu trúc một lệnh, đa dữ liệu (single instruction, multiple data - SIMD) trong đó các tầng lọc trùng phương riêng lẻ hoạt động một cách độc lập và song song đáp lại một dòng lệnh duy nhất. Ví dụ, bộ lọc đa tầng có thể bao gồm N tầng (trong đó N là số lớn hơn một), và một trong số các tầng này (tầng thứ “M+1” trong dãy) có thể hoạt động dựa trên các trị số dữ liệu được tạo ra bởi một tầng trước đó trong số các tầng (tầng thứ “M” trong dãy) tại các thời điểm khác nhau (ví dụ, đáp lại dãy gồm các trị số dữ liệu đầu vào khác nhau của khối), được lưu trữ trong bộ nhớ đệm (tại các thời điểm khác nhau), và được đọc (bởi tầng thứ “M+1”) từ bộ nhớ đệm sau khi nằm trong bộ nhớ đệm trong các khoảng thời gian chờ khác nhau.

Theo một số phương án, sáng chế đề xuất bộ lọc đa tầng, bao gồm:

bộ nhớ đệm;

ít nhất hai tầng lọc trùng phương, bao gồm tầng lọc trùng phương thứ nhất và tầng lọc trùng phương tiếp theo; và

bộ điều khiển, được nối với các tầng lọc trùng phương và được tạo cấu hình để gán một dòng lệnh duy nhất cho cả tầng lọc trùng phương thứ nhất và tầng lọc trùng phương tiếp theo, trong đó tầng lọc trùng phương thứ nhất đã nêu và tầng lọc trùng phương tiếp theo đã nêu hoạt động một cách độc lập và song song đáp lại dòng lệnh đó,

trong đó tầng lọc trùng phương thứ nhất được nối với bộ nhớ và được tạo cấu hình để thực hiện lọc trùng phương trên khối gồm N mẫu đầu vào đáp lại dòng lệnh để tạo ra các trị số trung gian, và để gán các trị số trung gian này cho bộ nhớ (để lưu trữ trong bộ nhớ này), trong đó các trị số trung gian bao gồm phiên bản được lọc của mỗi trong số ít nhất một tập con gồm các mẫu đầu vào, và

trong đó tầng lọc trùng phương tiếp theo được nối với bộ nhớ và được tạo cấu hình để thực hiện việc lọc trùng phương trên các trị số đệm được khôi phục từ bộ nhớ đáp lại dòng lệnh để tạo ra khối trị số đầu ra, trong đó trị số đầu ra bao gồm trị số đầu ra tương ứng với mỗi mẫu đầu vào trong khối gồm N mẫu đầu vào, và các trị số được đệm bao gồm ít nhất một số trong số các trị số trung gian được tạo ra trong tầng lọc trùng phương thứ nhất đáp lại khối gồm N mẫu đầu vào.

Theo các phương án thông thường, bộ lọc đa tầng được tạo cấu hình để thực hiện lọc đa tầng khối N mẫu đầu vào trong một vòng lặp xử lý duy nhất có sự lặp lại trên chỉ số mẫu mà không lặp lại trên chỉ số tầng lọc trùng phương.

Theo một số phương án, sáng chế đề xuất bộ lọc đa tầng trong đó bộ lọc đa tầng này có M tầng, tầng lọc trùng phương tiếp theo được tạo cấu hình để tạo ra trị số đầu ra tương ứng với mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào đáp lại tập con các trị số đệm được khôi phục từ bộ nhớ, trong đó j là chỉ số nằm trong khoảng từ $M-1$ đến $N-1$, tập con này bao gồm phiên bản đã lọc của mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào, phiên bản đã lọc của mẫu đầu vào thứ “ $j-1$ ” trong số các mẫu đầu vào, và phiên bản đã lọc của mẫu đầu vào thứ “ $j-2$ ” trong số các mẫu đầu vào.

Theo một số phương án của bộ lọc đa tầng theo sáng chế, tầng lọc trùng phương tiếp theo được tạo cấu hình để tạo ra trị số đầu ra tương ứng với mỗi trong số các mẫu đầu vào đáp lại tập con khác gồm các giá trị đệm được khôi phục từ bộ nhớ, mỗi tập con này bao gồm ít nhất ba trong số các trị số trung gian được tạo ra trong tầng lọc trùng phương thứ nhất và được khôi phục từ bộ nhớ sau khi nằm trong bộ nhớ trong các khoảng thời gian chờ khác nhau. Ví dụ, theo một phương án thông thường trong đó bộ lọc đa tầng có M tầng lọc trùng phương, tập con gồm các trị số đã đệm được khôi phục để tạo ra trị số đầu ra tương ứng với mẫu đầu vào thứ

“ j ” trong số các mẫu đầu vào, trong đó j là chỉ số nằm trong khoảng từ $M-1$ đến $N-1$, bao gồm ít nhất một trị số được tạo ra trong tầng lọc trùng phương thứ nhất đáp lại mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào, ít nhất một trị số được tạo ra trong tầng lọc trùng phương thứ nhất đáp lại mẫu đầu vào thứ “ $j-1$ ” trong số các mẫu đầu vào, và ít nhất một trị số được tạo ra trong tầng lọc trùng phương thứ nhất đáp lại mẫu đầu vào thứ “ $j-2$ ” trong số các mẫu đầu vào.

Theo một nhóm phương án khác, sáng chế đề xuất phương pháp thực hiện lọc đa tầng trên khối N mẫu đầu vào, phương pháp này bao gồm các bước:

(a) thực hiện hoạt động lọc trùng phương thứ nhất trên khối N mẫu đầu vào để tạo ra các trị số trung gian, và gán các trị số trung gian này cho bộ nhớ đệm (để lưu trữ trong bộ nhớ này), trong đó các trị số trung gian bao gồm phiên bản được lọc của mỗi trong số ít nhất một tập con các mẫu đầu vào; và

(b) thực hiện hoạt động lọc trùng phương thứ hai trên các trị số đệm được khôi phục từ bộ nhớ, để tạo ra khối các trị số đầu ra, trong đó các trị số đầu ra bao gồm trị số đầu ra tương ứng với mỗi trong số các mẫu đầu vào trong khối N mẫu đầu vào, tập con khác gồm các trị số đã đệm được khôi phục và được lọc để tạo ra trị số đầu ra tương ứng với mỗi mẫu đầu vào trong khối, và mỗi tập con các trị số đệm bao gồm ít nhất hai (ví dụ, ba) trị số trung gian được tạo ra trong khi thực hiện bước (a), các trị số này được khôi phục từ bộ nhớ sau khi nằm trong bộ nhớ này trong các khoảng thời gian chờ khác nhau,

trong đó các bước (a) và (b) được thực hiện đáp lại một dòng lệnh duy nhất, sao cho các bước (a) và (b) được thực hiện độc lập và song song đáp lại một dòng lệnh duy nhất đó. Theo các phương án thông thường, việc lọc đa tầng đối với khối gồm các mẫu đầu vào được thực hiện trong một vòng duy nhất có sự lặp lại trên chỉ số mẫu nhưng lại không lặp lại trên chỉ số tầng lọc trùng phương.

Theo một số phương án của phương pháp của sáng chế trong đó bước lọc được thực hiện trong bộ lọc đa tầng có M tầng, các trị số đệm được khôi phục ở bước (b) để tạo ra trị số đầu ra tương ứng với mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào, trong đó j là chỉ số nằm trong khoảng từ $M-1$ đến $N-1$, bao gồm phiên bản được lọc của mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào được tạo ra ở bước (a),

phiên bản được lọc của mẫu đầu vào thứ “j-1” trong số các mẫu đầu vào được tạo ra ở bước (a), và phiên bản được lọc của mẫu đầu vào thứ “j-2” trong số các mẫu đầu vào được tạo ra ở bước (a).

Theo một nhóm phương án khác, sáng chế đề xuất bộ mã hóa âm thanh được tạo cấu hình để tạo ra dữ liệu âm thanh được mã hóa đáp lại dữ liệu âm thanh đầu vào, bộ mã hóa này bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh (ví dụ, để lọc phiên bản đã được xử lý sơ bộ của dữ liệu âm thanh). Theo một nhóm phương án khác, sáng chế đề xuất phương pháp mã hóa dữ liệu âm thanh để tạo ra dữ liệu âm thanh được mã hóa, bao gồm bước thực hiện phương pháp lọc trùng phương đa tầng theo phương án bất kỳ của sáng chế trên dữ liệu âm thanh (ví dụ, trên phiên bản đã được xử lý sơ bộ của dữ liệu âm thanh). Ví dụ, theo một phương án, sáng chế đề xuất bộ mã hóa âm thanh bao gồm tầng xử lý trước (để xử lý sơ bộ dữ liệu âm thanh đầu vào cần được mã hóa bởi bộ mã hóa), trong đó tầng xử lý trước bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh (ví dụ, dữ liệu đầu vào hoặc phiên bản đã được xử lý sơ bộ của dữ liệu đầu vào). Theo một phương án khác nữa, sáng chế đề xuất bộ xử lý trước (để thực hiện xử lý sơ bộ dữ liệu âm thanh cần được mã hóa bởi bộ mã hóa), trong đó bộ xử lý trước này bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh (ví dụ, dữ liệu được nhập vào bộ xử lý trước này hoặc phiên bản đã được xử lý sơ bộ của dữ liệu đầu vào như vậy).

Theo một nhóm phương án khác, sáng chế đề xuất bộ giải mã âm thanh được tạo cấu hình để tạo ra dữ liệu âm thanh được giải mã đáp lại dữ liệu âm thanh được mã hóa. Theo một số phương án trong nhóm này, bộ giải mã bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh được mã hóa (ví dụ, để lọc phiên bản đã được xử lý sơ bộ của dữ liệu âm thanh được mã hóa). Theo một nhóm phương án khác, sáng chế đề xuất phương pháp giải mã dữ liệu âm thanh được mã hóa để tạo ra dữ liệu âm thanh được giải mã. Theo các phương án này,

bước giải mã bao gồm bước thực hiện phương pháp lọc trùng phương đa tầng theo phương án bất kỳ của sáng chế trên dữ liệu âm thanh được mã hóa (ví dụ, trên phiên bản đã được xử lý sơ bộ của dữ liệu âm thanh được mã hóa). Ví dụ, theo một phương án, sáng chế đề xuất bộ giải mã âm thanh bao gồm tầng xử lý sau (để xử lý sau dữ liệu âm thanh được giải mã mà đã được giải mã bởi bộ giải mã), trong đó tầng xử lý sau bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh (ví dụ, dữ liệu được giải mã hoặc phiên bản đã xử lý của dữ liệu được giải mã). Theo một phương án khác nữa, sáng chế đề xuất bộ xử lý sau (ví dụ, để thực hiện việc xử lý sau dữ liệu âm thanh được giải mã mà đã được giải mã bởi bộ giải mã), trong đó bộ xử lý sau bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh (ví dụ, dữ liệu được giải mã được nhập vào bộ xử lý sau hoặc phiên bản được xử lý của dữ liệu đầu vào như vậy).

Theo các phương án thông thường của sáng chế, các lệnh SIMD (hoặc các lệnh xử lý song song bởi nhiều đơn vị ALU hoặc AMU) được sử dụng để lập trình bộ xử lý (ví dụ, bộ xử lý tín hiệu số hoặc bộ xử lý đa năng) để cài đặt bộ lọc đa tầng. Bộ lọc đa tầng có thể thực hiện lọc băng thông, lọc thông thấp (ví dụ, trong hệ thống con LFE của bộ mã hóa âm thanh), lọc thông cao (ví dụ, trong hệ thống con dò tạm thời của bộ mã hóa âm thanh), hoặc lọc khác.

Các khía cạnh khác của sáng chế bao gồm hệ thống hoặc thiết bị (ví dụ, bộ mã hóa, bộ giải mã, hoặc bộ xử lý) được tạo cấu hình (ví dụ, được lập trình) để thực hiện phương pháp theo phương án bất kỳ của sáng chế, và phương tiện đọc được bởi máy tính (ví dụ, đĩa) lưu trữ mã để thực hiện phương pháp hoặc các bước của nó theo phương án bất kỳ của sáng chế. Ví dụ, hệ thống theo sáng chế có thể là hoặc bao gồm bộ xử lý đa năng lập trình được, bộ xử lý tín hiệu số, hoặc bộ vi xử lý, được lập trình với phần mềm hoặc phần cứng và/hoặc theo cách khác được tạo cấu hình để thực hiện hoạt động bất kỳ trong số các hoạt động trên dữ liệu, bao gồm phương pháp hoặc các bước của nó theo một phương án của sáng chế. Bộ xử lý đa năng như vậy có thể là hoặc bao gồm hệ thống máy tính bao gồm thiết bị nhập, bộ nhớ, và mạch xử lý được lập trình (và/hoặc theo cách khác được tạo cấu hình) để thực hiện

phương pháp (hoặc các bước của nó) theo một phương án của sáng chế đáp lại dữ liệu được gán vào đó.

Theo một số phương án, sáng chế đề xuất các bộ mã hóa (ví dụ, các bộ mã hóa để mã hóa dữ liệu âm thanh theo định dạng Dolby Digital Plus, AC-3, hoặc Dolby E) hoặc các bộ giải mã, được cài đặt làm các bộ xử lý lập trình (ví dụ, các bộ xử lý Neon ARM, mỗi bộ xử lý này là bộ xử lý Cortex ARM có động cơ SIMD Neon cho phép xử lý song song, hoặc các bộ xử lý khác có các đơn vị một lệnh, đa dữ liệu (single instruction, multiple data – SIMD) và/hoặc nhiều đơn vị logic số học (arithmetic logic unit - ALU) hoặc đơn vị thao tác số học (arithmetic manipulation unit - AMU)) hoặc các bộ xử lý tín hiệu số được lập trình (và/hoặc theo cách khác được tạo cấu hình) (ví dụ, các DSP có các đơn vị SIMD và/hoặc nhiều đơn vị ALU hoặc AMU).

Mô tả vắn tắt các hình vẽ

Fig.1 là sơ đồ khối minh họa bộ lọc trùng phương thông thường.

Fig.1A là sơ đồ khối minh họa bộ lọc trùng phương đa tầng thông thường.

Fig.2 là lưu đồ minh họa phương pháp thông thường để thực hiện lọc trong bộ lọc được cài đặt làm các bộ lọc trùng phương nhiều tầng tuần tự (“các đoạn trùng phương nhiều tầng”).

Fig.3 là lưu đồ minh họa phương pháp theo một phương án của sáng chế để thực hiện lọc trong bộ lọc trùng phương đa tầng bao gồm các bộ lọc trùng phương nhiều tầng (“các đoạn trùng phương nhiều tầng”) hoạt động song song đáp lại một dòng lệnh duy nhất.

Fig.4 là sơ đồ khối minh họa bộ lọc đa tầng (ví dụ, được cài đặt bằng cách lập trình DSP hoặc bộ xử lý khác theo một phương án của sáng chế), bộ lọc đa tầng này bao gồm các bộ lọc trùng phương nhiều tầng và có thể thực hiện phương pháp theo kiểu được mô tả dựa vào Fig.3. Trên Fig.4, bộ nhớ 10 bao gồm các vị trí nhớ lưu trữ mỗi khối dữ liệu đầu vào $x(n)$, và các vị trí nhớ đệm lưu trữ tất cả các trị số cần thiết trong số các trị số trung gian $x_1(n), \dots, x_{N-1}(n)$ do các bộ lọc trùng phương tạo ra.

Fig.5 là sơ đồ khối minh họa hệ thống bao gồm bộ mã hóa (bao gồm bộ lọc đa tầng theo một phương án của sáng chế) và bộ giải mã (cũng bao gồm bộ lọc đa tầng theo một phương án của sáng chế).

Fig.6 là lưu đồ minh họa phương pháp theo một phương án khác để thực hiện lọc trong bộ lọc trùng phương đa tầng bao gồm các bộ lọc trùng phương nhiều tầng (“các đoạn trùng phương nhiều tầng”) hoạt động song song đáp lại một dòng lệnh duy nhất.

Fig.7 là lưu đồ minh họa các bước 40, 41, và 42 của phương pháp theo một phương án trên Fig.6.

Fig.8 là lưu đồ minh họa các bước 47, 48, và 49 của phương pháp theo một phương án trên Fig.6.

Fig.9 là sơ đồ minh họa các trị số được tạo ra khi cài đặt hệ thống trên Fig.4 trong đó các phép toán được thực hiện tại chỗ.

Fig.10 là sơ đồ khối minh họa hệ thống bao gồm bộ mã hóa (bao gồm bộ lọc đa tầng theo một phương án của sáng chế) và bộ giải mã mà là bộ giải mã theo một phương án của sáng chế.

Mô tả chi tiết sáng chế

Theo các phương án của sáng chế, phương pháp và hệ thống (ví dụ, bộ mã hóa và bộ giải mã) được tạo cấu hình để thực hiện phương pháp theo sáng chế sẽ được mô tả dựa trên các hình vẽ Fig.3, Fig.4, Fig.5, Fig.6, Fig.7, and Fig.8.

Trước hết, dựa vào trên Fig.2, các tác giả sáng chế mô tả phương pháp thông thường để lọc các mẫu dữ liệu (ví dụ, khối gồm các mẫu dữ liệu âm thanh) bằng bộ lọc đa tầng bao gồm chuỗi M bộ lọc trùng phương (trong đó M là số được gọi là “n đoạn” dưới đây và trên Fig.2). Ví dụ về bộ lọc trùng phương đa tầng thông thường như vậy là bộ lọc của Fig.1A nêu trên.

Trong phương pháp trên Fig.2, mỗi khối mới gồm N mẫu cần được lọc ban đầu được đệm (ở bước 20). Mỗi mẫu trong khối được xác định bởi chỉ số j , trong đó $0 \leq j \leq N - 1$. Mỗi tầng (đoạn) trong bộ lọc đa tầng được xác định bởi chỉ số i , trong đó $0 \leq i \leq M - 1$.

Ở bước 21, chỉ số i được khởi tạo bằng không, và ở bước 22, chỉ số j được khởi tạo bằng không.

Ở bước 23, mẫu đầu vào thứ “ j ” được lọc trong bộ lọc trùng phương thứ “ i ”, và sau đó, ở bước 24, chỉ số j được tăng. Bước 25 xác định xem chỉ số j được tăng (bằng $j + 1$) có nhỏ hơn N hay không. Nếu ở bước 25 xác định rằng chỉ số j được tăng nhỏ hơn N , thì bước 23 lại thực hiện để lọc mẫu thứ “ $(j+1)$ ” tiếp theo trong bộ lọc trùng phương thứ “ i ”.

Nếu ở bước 25 xác định rằng chỉ số j được tăng là bằng N (sao cho tất cả các mẫu trong khối hiện thời đều được lọc trong bộ lọc trùng phương hiện thời, thì ở bước 26, chỉ số i được tăng.

Bước 27 xác định xem chỉ số i được tăng gần nhất (bằng $i + 1$) có nhỏ hơn số “ n đoạn” (mà là bằng M) hay không. Nếu ở bước 27 xác định rằng chỉ số i được tăng gần nhất nhỏ hơn M , thì lần lặp khác của các bước từ 22 đến 26 sau đó được thực hiện để lọc khối các trị số trung gian gần nhất (các đầu ra của bộ lọc trùng phương (thứ “ i ”) trước đó, được tạo ra trong bộ lọc trùng phương trước đó trong lần lặp trước đó của các bước từ 22 đến 25) trong bộ lọc trùng phương (thứ “ $(i+1)$ ”) tiếp theo.

Nếu ở bước 27 xác định rằng chỉ số i được tăng gần nhất bằng M , sao cho việc xử lý tất cả các mẫu trong khối hiện thời trong tất cả các bộ lọc trùng phương được hoàn thành, thì bước 28 được thực hiện. Ở bước 28, N mẫu đã lọc được tạo ra bằng cách lọc khối các mẫu đầu vào hiện thời trong bộ lọc đa tầng được xuất ra. Vào lúc này, khối N mẫu bổ sung bất kỳ cần được lọc được đệm (trong quy trình thực hiện mới của bước 20) và phương pháp trên Fig.2 được lặp lại để lọc khối mẫu mới trong bộ lọc đa tầng.

Đối với quy trình trên Fig.2, trong trường hợp $M = 2$ (tức là, trong trường hợp bộ lọc đa tầng chỉ bao gồm hai bộ lọc trùng phương nhiều tầng), quy trình xử lý được thực hiện trong mỗi tầng của bộ lọc đa tầng còn được mô tả bởi mã giả sau đây, trong đó $N =$ số lượng mẫu đầu ra cần được tạo ra bằng cách lọc khối gồm N mẫu trong bộ lọc đa tầng:

```

for (i = 0; i < 2; i++)
{
    for (j = 0; j < N; j++)
    {
        Output[j]= function(output[j-1], output[j-2], input[j], input[j-1], input[j-2]);
    }
}

```

Trong khi thực hiện phương pháp thông thường trên Fig.2, do đầu ra của mỗi tầng của bộ lọc đa tầng cho mỗi mẫu (“output[j]”) nói chung phụ thuộc vào các đầu ra của tầng cho hai mẫu trước đó (output[j-1] và output[j-2]), và phụ thuộc vào đầu vào hiện thời cho tầng (“input[j]”) và hai đầu vào trước đó cho tầng (input[j-1] và input[j-2]), các hoạt động đối với vòng lặp (các bước 23, 24, và 25 trên Fig.2) được thực hiện bên trong tầng không được song song hóa. Ngoài ra, do đầu ra của tầng thứ nhất ($i = 0$) được đưa vào tầng thứ hai ($i = 1$), nên các hoạt động trên các tầng không được song song hóa khi thực hiện phương pháp thông thường trên Fig.2. Điều này dẫn đến yêu cầu đếm chỉ lệnh cao (đôi khi còn được gọi là MIPS hay Hàng triệu Chỉ lệnh mỗi giây - Millions of Instructions Per Second) đối với bộ lọc đa tầng ngay cả khi nó được cài đặt bởi bộ xử lý mà cấu trúc của nó bao gồm nhiều đơn vị ALU (hoặc AMU) hoặc SIMD.

Tiếp theo, dựa vào Fig.3, các tác giả sáng chế mô tả một phương án của phương pháp lọc khối gồm các mẫu dữ liệu (ví dụ, khối gồm các mẫu dữ liệu âm thanh) với bộ lọc đa tầng bao gồm chuỗi hai bộ lọc trùng phương. Ban đầu, ở bước 30, mỗi khối mới gồm N mẫu cần lọc được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để sẵn sàng sử dụng trong các bước tiếp theo (bao gồm các bước 31, 33 và 34).

Mỗi mẫu trong khối được xác định bởi chỉ số j , trong đó $0 \leq j \leq N - 1$. Mỗi tầng (đoạn) trong bộ lọc đa tầng được xác định bởi chỉ số i , trong đó $0 \leq i \leq 1$.

Ở bước 31, mẫu đầu vào thứ nhất ($j = 0$) được lọc trong bộ lọc trùng phương thứ nhất ($i = 0$). Trị số tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên

Fig.4) để sẵn sàng dùng cho lần sử dụng tiếp theo (ví dụ, khi thực hiện các bước 33 và/hoặc 34 tiếp theo).

Ở bước 32, chỉ số j được thiết lập bằng 1.

Tiếp theo, các bước 33 và 34 được thực hiện song song. Ở bước 33, mẫu đầu vào thứ “ j ” được lọc trong bộ lọc trùng phương thứ nhất ($i = 0$), và ít nhất một (ví dụ, mỗi) trị số (trị số “trung gian”) được tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn dùng trong các bước tiếp theo. Ở bước 34, mẫu đầu vào thứ “ $j-1$ ” của bộ lọc trùng phương thứ hai ($i = 1$) được lọc, và ít nhất một (ví dụ, mỗi) trị số (trị số “trung gian”) tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn dùng trong các bước tiếp theo.

Sau đó, ở bước 35, chỉ số j được tăng. Bước 36 xác định xem chỉ số j đã tăng (bằng $j + 1$) có nhỏ hơn N hay không. Nếu ở bước 36 xác định được rằng chỉ số j đã tăng là nhỏ hơn N , thì các bước 33 và 34 được thực hiện lại để lọc mẫu đầu vào tiếp theo của bộ lọc trùng phương thứ nhất (ở bước 33) và mẫu đầu vào tiếp theo của bộ lọc trùng phương thứ hai (ở bước 34). Ít nhất một (ví dụ, mỗi) trị số (trị số “trung gian”) tạo ra bởi mỗi lần lặp lại mỗi trong số các bước 33 và 34 được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để sẵn sàng sử dụng trong các bước tiếp theo. Ví dụ, một hoặc nhiều trị số trung gian được tạo ra trong một hoặc nhiều lần lặp lại trước đó của bước 33 có thể được khôi phục từ bộ đệm để sử dụng trong việc thực hiện bước 34.

Nếu ở bước 36 xác định rằng chỉ số đã tăng j là bằng N (sao cho tất cả các mẫu đầu vào trong khối hiện thời đều được lọc trong bộ lọc trùng phương thứ nhất, thì bước 37 được thực hiện.

Ở bước 37, mẫu đầu vào cuối cùng ($j = N-1$) của bộ lọc trùng phương thứ hai ($i = 1$) được lọc. Trị số tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để sẵn sàng cho việc sử dụng sau này (ví dụ, cho xuất ra ở bước 38).

Sau đó, ở bước 38, N mẫu đã lọc tạo ra bởi bộ lọc trùng phương thứ hai được xuất ra (dưới dạng đầu ra của bộ lọc đa tầng đáp lại khối hiện thời gồm N mẫu đầu vào). Lúc này, khối bỏ sung bất kỳ gồm N mẫu cần lọc được đệm (trong quy trình

thực hiện mới của bước 30) và phương pháp trên Fig.3 được thực hiện lặp lại để lọc khối mới gồm các mẫu trong bộ lọc đa tầng.

Quy trình xử lý được thực hiện (khi thực hiện phương pháp trên Fig.3) trong mỗi tầng của bộ lọc đa tầng còn được mô tả bởi mã giả sau đây, trong đó N = số lượng các mẫu đầu ra cần được tạo ra bằng cách lọc khối gồm N mẫu trong bộ lọc đa tầng, $\text{outputstage1}[k]$ là đầu ra của tầng thứ nhất của bộ lọc đa tầng đáp lại mẫu đầu vào thứ k , $\text{outputstage2}[k]$ là đầu ra của tầng thứ hai của bộ lọc đa tầng tương ứng với mẫu đầu vào thứ k , và $\text{input}[k]$ là mẫu đầu vào thứ k của tầng thứ nhất của bộ lọc đa tầng:

```
{ xử lý mẫu thứ nhất của tầng thứ nhất}

for ( $j = 1; j < N; j++$ )
{
    Outputstage1[j] = function(outputstage1[j-1],
                               outputstage1[j-2], input[j], input[j-1], input[j-2] );

    Outputstage2[j-1] = function(outputstage2[j-2], outputstage2[j-3],
                                outputstage1[j-1], outputstage1[j-2], outputstage1[j-3] );
}

{ Xử lý mẫu thứ (“N-1”) cuối cùng của tầng thứ hai.}
```

Có thể thấy rõ ràng từ Fig.3 và mã giả tương ứng, quy trình xử lý ở cả hai tầng của bộ lọc đa tầng được kết hợp trong một vòng mẫu lặp (các bước 33, 34, 35 và 36 trên Fig.3). Bằng cách đưa vào khoảng thời gian chờ một mẫu giữa hai tầng (trong trường hợp bộ lọc trùng phương hai tầng) hoặc cụ thể hơn, như được mô tả dưới đây có dựa vào các hình vẽ trên Fig.6, Fig.7, và Fig.8, khoảng thời gian chờ một mẫu giữa mỗi tầng của bộ lọc đa tầng có hai hoặc nhiều tầng lọc trùng phương, quy trình xử lý ở tất cả các tầng của bộ lọc đa tầng có thể được song song hóa hoàn toàn theo sáng chế. Việc xử lý khối gồm các mẫu ở tất cả các tầng lọc trùng phương của bộ lọc đa tầng do đó có thể được song song hóa trong một vòng lặp mẫu (được kết hợp cho tất cả các tầng), theo một phương án được mô tả của sáng chế.

Các biến thể của phương pháp theo phương án trên Fig.3 của sáng chế, để lọc khối gồm các mẫu dữ liệu (ví dụ, khối gồm các mẫu dữ liệu âm thanh) bằng bộ lọc đa tầng gồm chuỗi M bộ lọc trùng phương (trong đó M lớn hơn 2) được xem xét. Các biến thể như vậy thường được thực hiện theo cách được mô tả dựa vào các hình vẽ trên Fig.6, Fig.7 và Fig.8.

Ban đầu, ở bước 40 của lưu đồ trên Fig.6, mỗi khối mới gồm N mẫu cần lọc được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để sẵn sàng cho việc sử dụng tiếp (bao gồm các bước 41, 43-45, và 48).

Mỗi mẫu trong khối được xác định bởi chỉ số j , trong đó $0 \leq j \leq N - 1$. Mỗi tầng (đoạn) trùng phương trong bộ lọc đa tầng được xác định bởi chỉ số i , trong đó $0 \leq i \leq M-1$.

Ở bước 41, việc lọc trước vòng lặp được thực hiện trên M-1 mẫu đầu vào thứ nhất (từ $j = 0$ đến $j = M-2$) trong các tầng lọc trùng phương từ $i = 0$ đến $i = M-2$ (ví dụ, theo cách được mô tả dựa vào Fig.7). Như được thể hiện trên Fig.7, cần lưu ý rằng đối với một số tầng lọc trùng phương, việc lọc trước vòng lặp xảy ra tương ứng với chỉ một tập con gồm M-1 mẫu đầu vào thứ nhất. Các trị số tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để sẵn sàng cho việc sử dụng sau (ví dụ, khi thực hiện các bước 43-45 tiếp theo).

Ở bước 42, chỉ số j được thiết lập bằng M - 1.

Sau đó, các bước từ 43 đến 45 (một bước cho mỗi trong số M tầng) được thực hiện song song. Ở bước 43, mẫu đầu vào thứ “ j ” của bộ lọc trùng phương thứ nhất ($i = 0$) được lọc, và ít nhất một (ví dụ, mỗi) trị số (trị số “trung gian”) tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để sẵn sàng sử dụng trong các bước tiếp theo. Ở bước 44, mẫu đầu vào thứ “ $j-1$ ” của bộ lọc trùng phương thứ hai ($i = 1$) được lọc, và ít nhất một (ví dụ, mỗi) trị số (trị số “trung gian”) tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để sẵn sàng sử dụng trong các bước tiếp theo. Tương tự (trong ít nhất một bước được thực hiện song song với các bước 43 và 44, giả sử M lớn hơn 2), mẫu đầu vào thứ “ $j-2$ ” của bộ lọc trùng phương thứ ba ($i = 2$) được lọc, mẫu đầu vào thứ “ $j-3$ ” của bộ lọc trùng phương thứ tư ($i = 3$) được lọc (giả sử M lớn hơn hoặc bằng 4), và v.v. đối với mỗi bộ lọc bổ

sung trong số các bộ lọc trùng phương từ $i = 4$ đến $i = M-2$, và ít nhất một (ví dụ, mỗi) trị số (trị số “trung gian”) tạo ra bởi mỗi bước như vậy được đệm để có sẵn sử dụng trong các bước tiếp theo. Ở bước 45 (được thực hiện song song với các bước 43 và 44, nếu M lớn hơn 2), thì mẫu đầu vào thứ “ $j-M+1$ ” của bộ lọc trùng phương ($i = M-1$) cuối cùng được lọc, và ít nhất một (ví dụ, mỗi) trị số (trị số “trung gian”) tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn sử dụng trong các bước tiếp theo.

Sau đó, ở bước 46, chỉ số j được tăng, và bước 47 xác định xem chỉ số đã tăng j (bằng $j + 1$) có nhỏ hơn N hay không. Nếu ở bước 47 xác định rằng chỉ số đã tăng j là nhỏ hơn N , thì các bước từ 43-45 (và (các) bước khác bất kỳ được thực hiện song song với các bước từ 43 đến 45) được thực hiện lại để lọc mẫu tiếp theo trong bộ lọc trùng phương thứ nhất (ở bước 43), mẫu tiếp theo trong bộ lọc trùng phương thứ hai (ở bước 44), và v.v. cho mỗi tầng lọc trùng phương bổ sung. Ít nhất một (ví dụ, mỗi) trị số (trị số “trung gian”) tạo ra bởi mỗi vòng lặp của mỗi trong số các bước từ 43 đến 45 được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn sử dụng trong các bước tiếp theo. Ví dụ, một hoặc nhiều trị số trung gian tạo ra trong một hoặc nhiều vòng lặp trước đó của bước 43 có thể được khôi phục từ bộ đệm để sử dụng khi thực hiện bước 44.

Nếu ở bước 47 xác định rằng chỉ số đã tăng j bằng N (sao cho tất cả các mẫu đầu vào trong khối hiện thời được lọc trong một trong số các bộ lọc trùng phương (bộ lọc trong đó $i = 0$), thì bước lọc sau vòng lặp 48 được thực hiện.

Ở bước 48, bước lọc sau vòng lặp được thực hiện trên các mẫu đầu vào chưa được lọc còn lại bất kỳ của các tầng lọc trùng phương từ $i = 1$ đến $i = M-1$ (ví dụ, theo cách được mô tả dựa vào Fig.8). (Các) trị số tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn cho việc sử dụng sau (ví dụ, để xuất ra ở bước 49).

Sau bước 48, ở bước 49, N mẫu được lọc tạo ra bởi bộ lọc trùng phương ($i = M-1$) cuối cùng được xuất ra (làm đầu ra của bộ lọc đa tầng đáp lại khối hiện thời gồm N mẫu đầu vào). Lúc này, khối bổ sung bất kỳ gồm N mẫu cần lọc được đệm

(trong quy trình thực hiện mới của bước 40) và phương pháp trên Fig.6 được thực hiện lặp lại để lọc khối mới gồm các mẫu trong bộ lọc đa tầng.

Fig.7 là lưu đồ minh họa các bước 40 và 42, và mô tả chi tiết về phương án của bước 41, của phương pháp theo phương án của sáng chế trên Fig.6. Các bước từ 50 đến 58 trên Fig.7 là phương án thực hiện của bước lọc trước vòng lặp 41 trên Fig.6. Ở bước 50, mẫu thứ nhất ($j = 0$) của khối hiện thời được lọc trong tầng lọc trùng phương thứ nhất ($i = 0$), và trị số mẫu được lọc tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn cho việc sử dụng sau này (ví dụ, trong các bước 51 và 52). Ở bước 51, mẫu thứ hai của khối ($j = 1$) được lọc trong tầng lọc trùng phương thứ nhất ($i = 0$), và trị số mẫu được lọc tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn cho việc sử dụng sau này (ví dụ, ở bước 53). Ở bước 52, mẫu thứ nhất của khối ($j = 0$) của tầng lọc trùng phương thứ hai ($i = 1$) được lọc, và trị số mẫu được lọc tạo ra bởi bước này tốt hơn là được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn cho việc sử dụng sau này (ví dụ, ở bước 54). Tốt hơn là các bước 51 và 52 được thực hiện song song (đáp lại cùng một chỉ lệnh hoặc chuỗi các chỉ lệnh gán cho các tầng thứ nhất và thứ hai).

Trong các bước được thể hiện theo chiều dọc bên dưới bước 51 trên Fig.7 (bao gồm các bước 53 và 56), mỗi trong số các mẫu từ mẫu thứ ba ($j = 2$) đến mẫu thứ “M-1” ($j = M-2$) của khối hiện thời được lọc trong tầng lọc trùng phương thứ nhất ($i = 0$), và trị số mẫu đã lọc được tạo ra bởi mỗi bước như vậy được đệm để có sẵn cho việc sử dụng sau. Ở bước 53, mẫu thứ ba ($j = 2$) của khối được lọc trong tầng lọc trùng phương thứ nhất ($i = 0$), và trị số mẫu được lọc tạo ra bởi bước như vậy được đệm để có sẵn cho việc sử dụng sau. Ở bước 56, mẫu thứ “M-1” ($j = M-2$) của khối hiện thời được lọc trong tầng lọc trùng phương thứ nhất ($i = 0$), và trị số mẫu đã lọc được tạo ra bởi bước này được đệm để có sẵn cho việc sử dụng sau.

Trong các bước được thể hiện theo chiều dọc bên dưới bước 52 trên Fig.7 (bao gồm các bước 54 và 57), mỗi trong số các mẫu đầu vào từ mẫu đầu vào thứ hai ($j = 1$) đến mẫu đầu vào thứ “M-2” ($j = M-3$) của tầng lọc trùng phương thứ hai ($i = 1$) được lọc, và trị số mẫu đã lọc được tạo ra bởi mỗi bước như vậy được đệm để có sẵn cho việc sử dụng sau. Ở bước 54, mẫu đầu vào thứ hai ($j = 1$) của tầng lọc trùng

phương thứ hai được lọc, và trị số mẫu đã lọc được tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn cho việc sử dụng sau. Ở bước 57, mẫu đầu vào thứ “M-2” ($j = M-3$) của tầng lọc trùng phương thứ hai được lọc, và trị số mẫu đã lọc được tạo ra bởi bước như vậy được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn cho việc sử dụng sau.

Nói chung, đối với tầng lọc trùng phương thứ “ k ”, trong đó k là chỉ số nằm trong khoảng từ 0 đến M-2, chuỗi các bước (cột có các bước trên Fig.7) được thực hiện để lọc mỗi trong số các mẫu đầu vào từ mẫu đầu vào thứ nhất ($j = 0$) của tầng lọc trùng phương thứ “ k ” đến mẫu đầu vào thứ “M-1- k ” ($j = M-2-k$) của tầng lọc trùng phương thứ “ k ”, và (các) trị số mẫu đã lọc được tạo ra bởi mỗi bước như vậy được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn cho việc sử dụng sau.

Do đó, nếu $M = 3$, thì các bước 53, 54, và 55 trên Fig.7 được thực hiện (tốt hơn là song song). Ở bước 53, mẫu đầu vào thứ ba ($j = 2$) của tầng lọc trùng phương thứ nhất ($i = 0$) được lọc, và trị số mẫu đã lọc tạo ra bởi bước này được đệm. Ở bước 54, mẫu đầu vào thứ hai ($j = 1$) của tầng lọc trùng phương thứ hai ($i = 1$) được lọc, và trị số mẫu đã lọc tạo ra bởi bước này được đệm. Ở bước 55, mẫu đầu vào thứ nhất ($j = 0$) của tầng lọc trùng phương thứ ba ($i = 2$) được lọc, và trị số mẫu đã lọc được tạo ra bởi bước này được đệm.

Tương tự, nếu $M > 5$, thì các bước của hàng nhiều bước bên dưới hàng gồm các bước từ 53 đến 55 (như được thể hiện trên Fig.7) được thực hiện (tốt hơn là song song), và sau đó các bước của hàng nhiều bước mà bao gồm các bước 56, 57, và 58 trên Fig.7 được thực hiện (tốt hơn là song song). Ở bước 56, mẫu đầu vào thứ “M-1” ($j = M-2$) của tầng lọc trùng phương thứ nhất ($i = 0$) được lọc, và trị số mẫu đã lọc được tạo ra bởi bước này được đệm. Ở bước 57, mẫu đầu vào thứ “M-2” ($j = M-3$) của tầng lọc trùng phương thứ hai ($i = 1$) được lọc, và trị số mẫu đã lọc được tạo ra bởi bước này được đệm. Ở bước 58, mẫu đầu vào thứ nhất ($j = 0$) của tầng lọc trùng phương thứ “M-1” ($i = M-2$) được lọc, và trị số mẫu đã lọc được tạo ra bởi bước này được đệm.

Các bước của mỗi hàng gồm nhiều bước trên Fig.7 (ví dụ, các bước 51 và 52, hoặc các bước 53, 54, và 55) tốt hơn là được thực hiện song song (đáp lại cùng một lệnh hoặc chuỗi các lệnh được gán cho các tầng liên quan).

Fig.8 là lưu đồ minh họa phương án của các bước 47 và 49, và mô tả chi tiết phương án của bước 48, theo một phương án của phương pháp trên Fig.6. Các bước từ 60 đến 66 trên Fig.8 là phương án thực hiện của bước lọc sau vòng lặp 48 trên Fig.6.

Nói chung, như được thể hiện trên Fig.8, đối với tầng lọc trùng phương thứ “ k ”, trong đó k là chỉ số nằm trong khoảng từ 1 đến $M-1$, chuỗi các bước (cột các bước trên Fig.8) được thực hiện để lọc mỗi mẫu đầu vào trong số các mẫu đầu vào từ mẫu đầu vào thứ “ $N-k+1$ ” ($j = N-k$) của tầng lọc trùng phương thứ “ k ” đến mẫu đầu vào cuối cùng ($j = N-1$) của tầng lọc trùng phương thứ “ k ”, và (các) trị số mẫu đã lọc được tạo ra bởi mỗi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có sẵn cho việc sử dụng sau.

Ví dụ, ở bước 60, mẫu cuối cùng ($j = N-1$) của khối hiện thời được lọc trong tầng lọc trùng phương thứ hai ($i = 1$), và trị số mẫu đã lọc tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có thể sử dụng sau. Ở bước 61, mẫu cuối cùng thứ hai của khối ($j = N-2$) được lọc trong tầng lọc trùng phương thứ ba ($i = 2$), và trị số mẫu đã lọc được tạo ra bởi bước này được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có thể sử dụng sau (ví dụ, ở bước 63). Ở bước 63, mẫu ($j = N-1$) cuối cùng của khối được lọc trong tầng lọc trùng phương thứ ba ($i = 2$), và trị số mẫu đã lọc được tạo ra bởi bước này tốt hơn là được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có thể sử dụng sau.

Ở bước 64, mẫu ($j = N-1$) cuối cùng của khối được lọc trong tầng lọc trùng phương cuối cùng thứ hai ($i = M-2$), và trị số mẫu được lọc tạo ra bởi bước này tốt hơn là được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có thể sử dụng sau.

Ở bước 62, mẫu thứ “ $N-M+1$ ” của khối được lọc trong tầng lọc trùng phương cuối cùng ($i = M-1$), và trị số mẫu đã lọc được tạo ra bởi bước này tốt hơn là được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có thể sử dụng sau (ví dụ, trong các bước theo chiều dọc bên dưới bước 62 trên Fig.8). Ở bước 65, mẫu ($j = N-2$)

cuối cùng thứ hai của khối được lọc trong tầng lọc trùng phương cuối cùng ($i = M-1$), và trị số mẫu đã lọc được tạo ra bởi bước này tốt hơn là được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có thể sử dụng sau (ví dụ, ở bước 66 in Fig.8). Ở bước 66, mẫu ($j = N-1$) cuối cùng của khối được lọc trong tầng lọc trùng phương cuối cùng ($i = M-1$), và trị số mẫu đã lọc được tạo ra bởi bước này tốt hơn là được đệm (ví dụ, trong bộ nhớ 10 trên Fig.4) để có thể sử dụng sau.

Các bước của mỗi hàng gồm nhiều bước trên Fig.8 (ví dụ, các bước 64 và 65, hoặc các bước ở hàng gồm các bước 60, 61, và 62) tốt hơn là được thực hiện song song (đáp lại cùng một lệnh hoặc chuỗi các lệnh được gán cho các tầng liên quan).

Phương pháp trên Fig.3 (và các biến thể của nó để lọc khối gồm các mẫu dữ liệu bằng bộ lọc đa tầng bao gồm chuỗi nhiều hơn hai bộ lọc trùng phương) thực hiện lọc đa tầng đối với khối gồm N mẫu đầu vào trong một vòng lặp có sự lặp lại trên chỉ số mẫu (chỉ số j trên Fig.3) mà không có sự lặp lại trên chỉ số tầng lọc trùng phương. Ngược lại, phương pháp thông thường trên Fig.2 xử lý khối gồm các mẫu dữ liệu trong hai vòng lặp lồng nhau có lặp trên cả chỉ số mẫu (chỉ số j trên Fig.2) và lặp trên chỉ số tầng lọc trùng phương (chỉ số i trên Fig.2).

Theo các phương án thông thường (ví dụ, phương án trên Fig.4 sẽ được mô tả sau đây), các tầng của bộ lọc đa tầng theo sáng chế (mỗi tầng trong số các tầng này là một bộ lọc trùng phương) được kết hợp với khoảng thời gian chờ giữa các tầng, sao cho tất cả các tầng có thể hoạt động độc lập, cho phép song song hóa việc xử lý của các tầng khác nhau. Tất cả các tầng có thể hoạt động song song (để lọc khỏi trị số dữ liệu đầu vào), đáp lại dòng lệnh chung duy nhất từ bộ điều khiển, nhưng với mỗi tầng lại hoạt động trên các trị số dữ liệu khác nhau, với ít nhất một trong số các tầng hoạt động trên các trị số dữ liệu bao gồm các trị số đệm (được tạo ra bởi một tầng khác trong số các tầng đáp lại tập con gồm các trị số dữ liệu đầu vào, và được lưu trữ với khoảng thời gian chờ khác trong bộ nhớ đệm trước khi được khôi phục để xử lý trong một trong số các tầng nêu trên). Do đó, bộ lọc đa tầng có cấu trúc SIMD (một lệnh, đa dữ liệu) trong đó các tầng lọc trùng phương riêng lẻ hoạt động một cách độc lập và song song đáp lại một dòng lệnh duy nhất. Ví dụ, bộ lọc đa tầng có thể bao gồm N tầng, và một trong số các tầng này (tầng thứ

“M+1” trong chuỗi) có thể hoạt động trên các trị số dữ liệu được tạo ra bởi một tầng trước đó trong số các tầng (tầng thứ “M” trong chuỗi) tại các thời điểm khác nhau (ví dụ, đáp lại chuỗi gồm các trị số dữ liệu đầu vào khác nhau của khối), được lưu trữ trong bộ nhớ đệm (tại các thời điểm khác nhau), và đọc (bởi tầng thứ “M+1”) từ bộ nhớ đệm sau khi nằm trong bộ nhớ đệm trong các khoảng thời gian chờ khác nhau.

Tiếp theo, dựa vào Fig.4, các tác giả sáng chế mô tả một nhóm các phương án của bộ lọc đa tầng theo sáng chế. Bộ lọc đa tầng trên Fig.4 bao gồm nhiều bộ lọc trùng phương (M bộ lọc trùng phương, trong đó M là số nguyên lớn hơn một) và được tạo cấu hình để thực hiện phương pháp theo kiểu được mô tả trên Fig.3 (hoặc biến thể của phương pháp này, chẳng hạn như phương pháp được thể hiện trên Fig.6). Bộ lọc trên Fig.4 bao gồm bộ nhớ 10, bộ điều khiển 11, và các bộ lọc trùng phương $B_1, B_2, \dots, B_M$, được kết nối như được thể hiện trên hình vẽ, và được tạo cấu hình để lọc khối gồm N trị số dữ liệu đầu vào $x(n)$, trong đó “n” là chỉ số nằm trong khoảng từ 1 đến N, đáp lại một dòng lệnh duy nhất được gán cho các bộ lọc trùng phương bởi bộ điều khiển 11. Mỗi trong số các trị số dữ liệu đầu vào $x(n)$ có thể là mẫu dữ liệu âm thanh.

Cần hiểu rằng cụm từ “dòng lệnh duy nhất” được gán vào các tầng riêng lẻ (ví dụ, mỗi tầng là một bộ lọc trùng phương) của bộ lọc đa tầng, được dùng trong bản mô tả này theo nghĩa rộng bao gồm cả hai trường hợp: các trường hợp trong đó một dòng lệnh duy nhất được gán cho tất cả các tầng (ví dụ, trên một bus hoặc vật dẫn mà tất cả các tầng được nối vào); và các trường hợp trong đó các dòng lệnh giống nhau (hoặc về cơ bản là giống nhau) được gán đồng thời (hoặc về cơ bản là đồng thời) cho các tầng (ví dụ, mỗi dòng được gán trên bus hoặc vật dẫn khác được nối với một tầng khác trong số các tầng).

Đáp lại khối gồm các trị số dữ liệu đầu vào $x(n)$, bộ lọc B_1 tạo ra N trị số (được lọc trùng phương) trung gian $x_1(n)$, và gán các trị số này cho các vị trí nhớ đệm trong bộ nhớ 10. Khi hoạt động, bộ lọc B_2 khôi phục các trị số trung gian cần thiết $x_1(n)$ từ bộ nhớ 10, tạo ra các trị số (được lọc trùng phương) trung gian $x_2(n)$ để đáp lại các trị số này, và gán các trị số trung gian mà nó tạo ra cho các vị trí nhớ

đệm trong bộ nhớ 10. Tương tự, khi hoạt động, mỗi bộ lọc khác trong số các bộ lọc trùng phương (bộ lọc B_i , trong đó i là chỉ số nằm trong khoảng từ 3 đến M) khôi phục các trị số trung gian $x_{i-1}(n)$ từ bộ nhớ 10, tạo ra các trị số được lọc trùng phương $x_i(n)$ đáp lại các trị số này, và gán các trị số mà nó tạo ra cho các vị trí nhớ đệm trong bộ nhớ 10. Các trị số được lọc trùng phương, $x_M(n) = y(n)$, được tạo ra trong bộ lọc trùng phương cuối cùng (B_M) bao gồm khối gồm N trị số dữ liệu đầu ra được lọc hoàn toàn được tạo ra đáp lại khối gồm các trị số dữ liệu đầu vào $x(n)$.

Bộ nhớ 10 bao gồm các vị trí nhớ lưu trữ mỗi khối dữ liệu đầu vào $x(n)$, và các vị trí nhớ đệm lưu trữ các trị số trung gian $x_1(n), \dots, x_{M-1}(n)$ được tạo ra bởi các bộ lọc trùng phương $B_1, B_2, \dots, B_{M-1}$ (ví dụ, các vị trí đệm lưu trữ các trị số trung gian $x_1(n), \dots, x_{M-1}(n)$ được tạo ra cho mỗi khối gồm dữ liệu đầu vào). Theo một số phương án thực hiện, trong đó các phép toán được thực hiện tại chỗ, các vị trí nhớ giống nhau mà được sử dụng để lưu trữ dữ liệu đầu vào $x(n)$ có thể được dùng để lưu trữ các trị số trung gian $x_1(n), \dots, x_{M-1}(n)$ ngay khi các mẫu dữ liệu đầu vào cụ thể không còn được bộ lọc đa tầng cần đến nữa. Trong các phương án thực hiện như vậy, bộ nhớ 10 thường không cần bao gồm nhiều vị trí nhớ hơn (hoặc nhiều hơn đáng kể) so với bộ nhớ thông thường (để cài đặt phiên bản thông thường, không được song song hóa của bộ lọc đa tầng), do bộ nhớ thông thường như vậy thường bao gồm các vị trí nhớ để lưu trữ mỗi khối gồm dữ liệu đầu vào, $x(n)$, cần lọc, và mỗi trị số đầu ra được tạo ra bởi mỗi trong số các tầng của bộ lọc đa tầng cần thiết cho sự hoạt động của chính tầng đó và/hoặc cho sự hoạt động của mỗi tầng tiếp theo của bộ lọc.

Ví dụ, Fig.9 là sơ đồ minh họa các trị số được tạo ra trong phương án thực hiện của hệ thống trên Fig.4 trong đó các phép tính toán được thực hiện tại chỗ, trong trường hợp $N = 4$ và $M = 2$ (tức là, hoạt động trùng phương hai tầng trên các khối gồm 4 mẫu tại một thời điểm).

Trong ví dụ trên Fig.9, các tác giả sáng chế bắt đầu với bốn mẫu trong bộ đệm đầu vào, $x(0) \dots x(3)$.

Ở bước thứ nhất, mẫu $x(0)$ được lọc qua bộ lọc B0 (trùng phương tầng thứ nhất) để tạo ra mẫu $x1(0)$. Mẫu $x1(0)$ được lưu trữ trong bộ nhớ ở vị trí trước đó bị chiếm bởi mẫu $x(0)$. Tất cả các vị trí nhớ khác không thay đổi.

Ở bước thứ hai, mẫu $x1(0)$ được lọc qua bộ lọc B1 (trùng phương tầng thứ hai) để tạo ra mẫu $y(0)$. Mẫu $y(0)$ được lưu trữ trong bộ nhớ ở vị trí trước đó bị chiếm bởi mẫu $x1(0)$.

Đồng thời, mẫu $x(1)$ được lọc qua bộ lọc B0 để tạo ra mẫu $x1(1)$. Mẫu $x1(1)$ được lưu trữ trong bộ nhớ ở vị trí trước đó bị chiếm bởi mẫu $x(1)$.

Trong các bước tiếp theo, quy trình xử lý tiếp tục cho đến khi tất cả các mẫu đầu vào $x(0) \dots x(3)$ được thay thế bởi các mẫu đầu ra $y(0) \dots y(3)$.

Trong ví dụ trên Fig.9, khi bộ lọc B1 được áp dụng cho mẫu $x1(2)$, các mẫu $x1(1)$ và $x1(0)$ không còn nằm trong bộ đệm nữa (được thay thế bởi $y(0)$ và $y(1)$). Thay vào đó, các mẫu mà được tạo ra bởi bộ lọc B1 từ $x1(1)$ và $x1(0)$ (tức là, các mẫu tương ứng với $s_1(n)$ và $s_2(n)$ trên Fig.1) được chứa trong 2 biến số trạng thái gắn với bộ lọc B1.

Trong ví dụ trên Fig.9, mỗi bộ lọc (ví dụ, mỗi bộ lọc trong số các bộ lọc B0 và B1) cần truy cập vào các vị trí nhớ lưu trữ hai mẫu (tương ứng với $s_1(n)$ và $s_2(n)$ trên Fig.1) mà nó tạo ra (đáp ứng lại các mẫu đầu vào thứ “ $j-1$ ” và “ $j-2$ ” của khối hiện thời của N mẫu đầu vào) dùng để lọc mẫu đầu vào thứ “ j ” của khối hiện thời. Các vị trí nhớ này có thể nằm trong bộ nhớ 10 của hệ thống trên Fig.4 (hoặc có thể là các vị trí nhớ đệm khác).

Đối với mỗi trong số các bộ lọc, mỗi cặp mẫu lưu trữ (tương ứng với $s_1(n)$ và $s_2(n)$ trên Fig.1) mà đã được tạo ra bởi bộ lọc được cập nhật mỗi lần một mẫu đầu vào mới (với chỉ số j được tăng) được gán cho bộ lọc. Các mẫu đã lưu trữ (tương ứng với $s_1(n)$ và $s_2(n)$ trên Fig.1) là các ví dụ về “các trị số trung gian” (cụm từ này cũng được sử dụng trong suốt bản mô tả) mà được tạo ra bởi bộ lọc (bộ lọc này là một tầng của bộ lọc đa tầng) và được đệm cho lần sử dụng tiếp theo sáng chế, nhưng tiếp đó chúng được sử dụng bởi tầng lọc mà tạo ra chúng (chứ không phải bởi một tầng lọc khác của bộ lọc đa tầng).

Mặc dù Fig.9 thể hiện một ví dụ cụ thể về phương pháp theo sáng chế sử dụng kỹ thuật lọc tại chỗ, trong đó kích thước khối (N) là bằng 4 và số lượng tầng lọc trùng phương (M) trong bộ lọc theo sáng chế là bằng 2, nhưng các phương án của phương pháp theo sáng chế sử dụng kỹ thuật lọc tại chỗ này được áp dụng cho mọi trị số M và N phụ thuộc vào sự ràng buộc $M > 1$ và $N > M$. Trong phương án thực hiện theo Fig.9 (và các phương án khác sử dụng kỹ thuật lọc tại chỗ trong đó kích thước khối là khác 4 và/hoặc số lượng tầng trùng phương là khác 2), các vị trí nhớ giống nhau (ví dụ, trong bộ nhớ 10 trên Fig.4) mà được dùng để lưu trữ dữ liệu đầu vào $x(n)$ có thể được dùng để lưu trữ các trị số trung gian $x_1(n), \dots, x_{M-1}(n)$ ngay khi các mẫu dữ liệu đầu vào cụ thể không còn được bộ lọc đa tầng cần đến nữa.

Bộ lọc trên Fig.4 có thể được thực hiện bằng cách lập trình bộ xử lý tín hiệu số (DSP - digital signal processor) hoặc bộ xử lý khác mà bao gồm bộ nhớ (thực hiện chức năng như bộ nhớ 10), bộ điều khiển (thực hiện chức năng như bộ điều khiển 11), và các đơn vị lôgic số học (arithmetic logic unit - ALU) hoặc các đơn vị thao tác số học (arithmetic manipulation unit - AMU), với mỗi trong số các bộ lọc trùng phương $B_1, B_2, \dots, B_M$ được thực hiện như một trong số các ALU hoặc AMU được tạo cấu hình thích hợp.

Do đó, bộ lọc trên Fig.4 bao gồm:

bộ nhớ đệm (các vị trí đệm trong bộ nhớ 10);

ít nhất hai tầng lọc trùng phương (các bộ lọc trùng phương $B_1, B_2, \dots, B_M$), bao gồm tầng lọc trùng phương thứ nhất (ví dụ, bộ lọc trùng phương B_1) và tầng lọc trùng phương tiếp theo (ví dụ, bộ lọc trùng phương B_2); và

bộ điều khiển (bộ điều khiển 11), được nối với các tầng lọc trùng phương và được tạo cấu hình để gán một dòng lệnh duy nhất cho cả tầng lọc trùng phương thứ nhất và tầng lọc trùng phương tiếp theo. Tầng lọc trùng phương thứ nhất và tầng lọc trùng phương tiếp theo (và mỗi tầng lọc trùng phương khác của bộ lọc trên Fig.4) hoạt động một cách độc lập và song song đáp lại dòng lệnh đó.

Tầng lọc trùng phương thứ nhất được nối với bộ nhớ và được tạo cấu hình để thực hiện lọc trùng phương trên khối gồm N mẫu đầu vào đáp lại dòng lệnh để tạo

ra các trị số trung gian (ví dụ, các trị số $x_1(n)$), và để gán các trị số trung gian này cho bộ nhớ (để lưu trữ trong bộ nhớ này). Các trị số trung gian này bao gồm phiên bản được lọc của mỗi trong số các mẫu đầu vào. Theo một số phương án nhất định của sáng chế, không cần phải có nhiều hơn một trị số trung gian $x_1(n)$ trong bộ nhớ 10 tại thời điểm bất kỳ. Tầng lọc trùng phương tiếp theo được nối với bộ nhớ và được tạo cấu hình để thực hiện lọc trùng phương trên các trị số đệm được khôi phục từ bộ nhớ đáp lại dòng lệnh để tạo ra khối gồm các trị số đầu ra (ví dụ, các trị số $x_2(n)$), trong đó các trị số đầu ra bao gồm trị số đầu ra tương ứng với mỗi trong số các mẫu đầu vào trong khối gồm N mẫu đầu vào, và các trị số đệm bao gồm ít nhất một số trong số các trị số trung gian được tạo ra trong tầng lọc trùng phương thứ nhất đáp lại khối gồm N mẫu đầu vào.

Tầng lọc trùng phương tiếp theo (ví dụ, bộ lọc B_2 trên Fig.4), theo một phương án trong đó bộ lọc đa tầng có M tầng, được tạo cấu hình để tạo ra trị số đầu ra tương ứng với mẫu đầu vào thứ “j” trong số các mẫu đầu vào đáp lại tập con các trị số đệm được khôi phục từ bộ nhớ, trong đó j là chỉ số nằm trong khoảng từ M-1 đến N-1, tập con này bao gồm phiên bản được lọc của mẫu đầu vào thứ “j” trong số các mẫu đầu vào, phiên bản được lọc của mẫu đầu vào thứ “j-1” trong số các mẫu đầu vào, và phiên bản được lọc của mẫu đầu vào thứ “j-2” trong số các mẫu đầu vào.

Tầng lọc trùng phương tiếp theo (ví dụ, bộ lọc B_2 trên Fig.4) được tạo cấu hình để tạo ra trị số đầu ra ($x_2(n)$) tương ứng với mỗi trong số các mẫu đầu vào, $x(n)$, đáp lại tập con các giá trị đệm khác nhau được khôi phục từ bộ nhớ, mỗi tập con này bao gồm ít nhất hai (ví dụ, ba) trong số các trị số trung gian được tạo ra trong tầng lọc trùng phương thứ nhất (ví dụ, các trị số $x_1(n)$, $x_1(n-1)$, và $x_1(n-2)$, được thể hiện trên Fig.4) và được khôi phục từ bộ nhớ sau khi nằm trong bộ nhớ này trong các khoảng thời gian chờ khác nhau. Cụ thể hơn, theo một phương án trong đó bộ lọc đa tầng có M tầng, tập con các trị số đệm được khôi phục bởi bộ lọc B_2 để tạo ra trị số đầu ra tương ứng với mẫu đầu vào thứ “j” trong số các mẫu đầu vào, trong đó j là chỉ số nằm trong khoảng từ M-1 đến N-1, bao gồm ít nhất một trị số được tạo ra trong tầng lọc trùng phương thứ nhất đáp lại mẫu đầu vào thứ “j” trong số các mẫu đầu vào, ít nhất một trị số được tạo ra trong tầng lọc trùng phương

thứ nhất đáp lại mẫu đầu vào thứ “ $j-1$ ” trong số các mẫu đầu vào, và ít nhất một trị số được tạo ra trong tầng lọc trùng phương thứ nhất đáp lại mẫu đầu vào thứ “ $j-2$ ” trong số các mẫu đầu vào.

Bộ lọc trên Fig.4 được tạo cấu hình để thực hiện lọc đa tầng trên khối gồm N mẫu đầu vào (các trị số dữ liệu $x(n)$), bao gồm bằng cách thực hiện các bước:

(a) thực hiện thao tác lọc trùng phương thứ nhất trên khối gồm N mẫu đầu vào để tạo ra các trị số trung gian (ví dụ, các trị số $x_1(n)$, $x_1(n-1)$, và $x_1(n-2)$, được biểu thị trên Fig.4), và gán các trị số trung gian này cho bộ nhớ đệm (để lưu trữ trong bộ nhớ này), trong đó các trị số trung gian bao gồm phiên bản được lọc của mỗi trong số ít nhất một tập con của các mẫu đầu vào; và

(b) thực hiện thao tác lọc trùng phương thứ hai trên các trị số đệm khôi phục từ bộ nhớ, để tạo ra khối các trị số đầu ra (ví dụ, các trị số $x_2(n)$ được thể hiện trên Fig.4), trong đó trị số đầu ra bao gồm trị số đầu ra tương ứng với mỗi mẫu đầu vào trong khối gồm N mẫu đầu vào, tập con các trị số được đệm khác nhau được khôi phục và lọc để tạo ra trị số đầu ra tương ứng với mỗi mẫu đầu vào trong khối, và mỗi tập con các trị số đệm bao gồm ít nhất hai (ví dụ, ba) trong số các trị số trung gian được tạo ra khi thực hiện bước (a) (ví dụ, các trị số $x_1(n)$, $x_1(n-1)$, và $x_1(n-2)$, được thể hiện trên Fig.4), các trị số này được khôi phục từ bộ nhớ sau khi nằm trong bộ nhớ này trong các khoảng thời gian chờ khác nhau,

trong đó các bước (a) và (b) được thực hiện đáp lại một dòng lệnh duy nhất, sao cho các bước (a) và (b) được thực hiện độc lập và song song đáp lại dòng lệnh duy nhất đó.

Các trị số đệm được khôi phục ở bước (b) để tạo ra trị số đầu ra tương ứng với mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào theo một phương án trong đó việc lọc được thực hiện trong bộ lọc đa tầng có M tầng, trong đó j là chỉ số nằm trong khoảng từ $M-1$ đến $N-1$, bao gồm phiên bản được lọc của mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào được tạo ra ở bước (a), phiên bản được lọc của mẫu đầu vào thứ “ $j-1$ ” trong số các mẫu đầu vào được tạo ra ở bước (a), và phiên bản được lọc của mẫu đầu vào thứ “ $j-2$ ” trong số các mẫu đầu vào được tạo ra ở bước (a).

Fig.5 là sơ đồ khối minh họa hệ thống bao gồm bộ mã hóa (bộ mã hóa 150) bao gồm bộ lọc đa tầng theo một phương án của sáng chế (“bộ lọc M B” 153). Ví dụ, bộ lọc 153 có thể thuộc loại được thể hiện trên và được mô tả dựa vào Fig.4. Bộ mã hóa 150 tùy ý bao gồm hai hoặc nhiều bộ lọc đa tầng, mỗi bộ lọc đa tầng này là bộ lọc đa tầng theo một phương án của sáng chế. Đáp lại các mẫu dữ liệu âm thanh đầu vào, bộ mã hóa 150 tạo ra dữ liệu âm thanh được mã hóa và gán dữ liệu âm thanh được mã hóa cho hệ thống con phân phối 151.

Hệ thống con phân phối 151 được tạo cấu hình để lưu trữ dữ liệu âm thanh được mã hóa và/hoặc để truyền tín hiệu chỉ báo dữ liệu âm thanh được mã hóa. Bộ giải mã 152 được nối và được tạo cấu hình (ví dụ, được lập trình) để nhận dữ liệu âm thanh được mã hóa từ hệ thống con 151 (ví dụ, bằng cách đọc ra hoặc khôi phục dữ liệu âm thanh được mã hóa từ bộ lưu trữ trong hệ thống con 151, hoặc nhận tín hiệu chỉ báo dữ liệu âm thanh được mã hóa đã được truyền bởi hệ thống con 151).

Bộ giải mã 152 bao gồm bộ lọc đa tầng theo một phương án của sáng chế (“bộ lọc M B” 154). Ví dụ, bộ lọc 154 có thể thuộc loại được thể hiện trên và được mô tả dựa vào Fig.4. Bộ giải mã 152 tùy ý bao gồm hai hoặc nhiều bộ lọc đa tầng, mỗi bộ lọc đa tầng này là bộ lọc đa tầng theo một phương án của sáng chế. Bộ giải mã 152 hoạt động để giải mã dữ liệu âm thanh được mã hóa, do đó tạo ra dữ liệu âm thanh được giải mã.

Hệ thống trên Fig.5 còn bao gồm hệ thống con xử lý trước âm thanh (“bộ xử lý trước”) 155 được tạo cấu hình để thực hiện xử lý sơ bộ dữ liệu âm thanh cần được mã hóa bởi bộ mã hóa 150. Bộ xử lý trước 155 bao gồm bộ lọc đa tầng theo một phương án của sáng chế (“bộ lọc M B” 157). Ví dụ, bộ lọc 157 có thể thuộc loại được thể hiện trên và được mô tả dựa vào Fig.4.

Hệ thống trên Fig.5 còn bao gồm hệ thống con xử lý sau âm thanh (“bộ xử lý sau”) 156 được tạo cấu hình để thực hiện xử lý sau dữ liệu âm thanh giải mã mà đã được giải mã bởi bộ giải mã 154. Bộ xử lý sau 156 bao gồm bộ lọc đa tầng theo một phương án của sáng chế (“bộ lọc M B” 158). Ví dụ, bộ lọc 158 có thể thuộc loại được thể hiện trên và được mô tả dựa vào Fig.4.

Theo một số phương án thực hiện, bộ mã hóa 150 là bộ mã hóa AC-3 (hoặc AC-3 nâng cao, hoặc Dolby E), bộ mã hóa này được tạo cấu hình để tạo ra dòng bit âm thanh được mã hóa AC-3 (hoặc AC-3 nâng cao, hoặc Dolby E) đáp lại dữ liệu âm thanh đầu vào miền thời gian, và bộ giải mã 52 là bộ giải mã AC-3 (hoặc AC-3 nâng cao, hoặc Dolby E).

Theo một nhóm các phương án, sáng chế là bộ mã hóa âm thanh (ví dụ, bộ mã hóa 150 trên Fig.5) được tạo cấu hình để tạo ra dữ liệu âm thanh mã hóa đáp lại dữ liệu âm thanh đầu vào, bộ mã hóa này bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh (ví dụ, để lọc phiên bản được xử lý sơ bộ của dữ liệu âm thanh). Bộ mã hóa 150 được tạo cấu hình để mã hóa dữ liệu âm thanh để tạo ra dữ liệu âm thanh được mã hóa, bao gồm bằng cách thực hiện phương pháp lọc đa tầng trên dữ liệu âm thanh (ví dụ, trên phiên bản được xử lý sơ bộ của dữ liệu âm thanh) theo một phương án của sáng chế.

Theo một nhóm các phương án, sáng chế là bộ giải mã âm thanh (ví dụ, bộ giải mã 152 trên Fig.5) được tạo cấu hình để tạo ra dữ liệu âm thanh được giải mã đáp lại dữ liệu âm thanh được mã hóa, bộ giải mã này bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh được mã hóa (ví dụ, để lọc phiên bản đã xử lý sơ bộ của dữ liệu âm thanh được mã hóa). Bộ giải mã 152 được tạo cấu hình để giải mã dữ liệu âm thanh được mã hóa để tạo ra dữ liệu âm thanh được giải mã, bao gồm bằng cách thực hiện phương pháp lọc đa tầng trên dữ liệu âm thanh được mã hóa (ví dụ, trên phiên bản đã xử lý sơ bộ của dữ liệu âm thanh được mã hóa) theo một phương án của sáng chế.

Theo một phương án khác nữa, sáng chế là bộ xử lý trước (ví dụ, bộ xử lý trước 155 trên Fig.5) để thực hiện quy trình xử lý sơ bộ dữ liệu âm thanh (ví dụ, dữ liệu âm thanh cần được mã hóa bởi bộ mã hóa), trong đó bộ xử lý trước này bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh (ví dụ,

dữ liệu được nhập vào bộ xử lý trước này hoặc phiên bản đã xử lý sơ bộ của dữ liệu đầu vào như vậy).

Theo một phương án khác nữa, sáng chế là bộ xử lý sau (ví dụ, bộ xử lý sau 156 trên Fig.5) để thực hiện việc xử lý sau dữ liệu âm thanh giải mã mà đã được giải mã bởi bộ giải mã, trong đó bộ xử lý sau này bao gồm ít nhất một bộ lọc đa tầng (bộ lọc đa tầng này là bộ lọc đa tầng theo phương án bất kỳ của sáng chế) được nối và được tạo cấu hình để lọc dữ liệu âm thanh (ví dụ, dữ liệu giải mã được nhập vào bộ xử lý sau hoặc phiên bản xử lý của dữ liệu đầu vào như vậy).

Fig.10 là sơ đồ khối minh họa hệ thống bao gồm bộ mã hóa (bộ mã hóa 150) bao gồm bộ lọc đa tầng theo một phương án của sáng chế (“bộ lọc M B” 153). Bộ mã hóa 150 trên Fig.10 là giống với bộ mã hóa 150 trên Fig.5, và có thể được thực hiện theo cách bất kỳ trong số các cách trong đó bộ mã hóa 150 trên Fig.5 có thể được thực hiện. Bộ lọc đa tầng 153 có thể thực hiện phương án bất kỳ của bộ lọc đa tầng theo sáng chế. Đáp lại các mẫu dữ liệu âm thanh đầu vào, bộ mã hóa 150 tạo ra dữ liệu âm thanh được mã hóa (biểu diễn một hoặc nhiều kênh âm thanh đầu vào) và gán dữ liệu âm thanh được mã hóa cho hệ thống con phân phối 151.

Hệ thống con phân phối 151 được tạo cấu hình để lưu trữ dữ liệu âm thanh được mã hóa và/hoặc để truyền tín hiệu chỉ báo dữ liệu âm thanh được mã hóa. Hệ thống con 151 trên Fig.10 là giống với hệ thống con 151 trên Fig.5 và có thể được thực hiện theo cách bất kỳ trong số các cách trong đó hệ thống con 151 trên Fig.5 có thể được thực hiện.

Bộ giải mã 252 trên Fig.10 có đầu vào được nối để nhận dữ liệu âm thanh mã hóa từ hệ thống con 151 (ví dụ, bằng cách đọc hoặc khôi phục dữ liệu âm thanh mã hóa từ bộ nhớ trong hệ thống con 151, hoặc nhận tín hiệu chỉ báo dữ liệu âm thanh mã hóa mà đã được truyền bởi hệ thống con 151).

Bộ giải mã 252 được tạo cấu hình (ví dụ, được lập trình) để trích (từ dòng bit được mã hóa nhận được) dữ liệu mã hóa biểu diễn một hoặc nhiều kênh thông tin âm thanh được xử lý bởi bộ lọc đa tầng 153, và để giải mã dữ liệu mã hóa để cung cấp các dạng biểu diễn giải mã của một hoặc nhiều kênh thông tin âm thanh.

Theo các phương án thông thường của sáng chế, các lệnh SIMD (hoặc các lệnh xử lý song song bởi nhiều đơn vị ALU hoặc AMU) được dùng để lập trình bộ xử lý (ví dụ, bộ xử lý tín hiệu số hoặc bộ xử lý đa năng) để cài đặt bộ lọc đa tầng. Bộ lọc đa tầng có thể thực hiện lọc thông thấp hạn chế băng thông, lọc thông thấp (ví dụ, trong hệ thống con LFE của bộ mã hóa âm thanh), lọc thông cao (ví dụ, trong hệ thống con dò tạm thời của bộ mã hóa âm thanh), hoặc lọc khác.

Theo các khía cạnh khác, sáng chế bao gồm hệ thống hoặc thiết bị (ví dụ, bộ mã hóa, bộ giải mã, hoặc bộ xử lý) được tạo cấu hình (ví dụ, được lập trình) để thực hiện phương pháp theo phương án bất kỳ của sáng chế, và phương tiện đọc được bởi máy tính (ví dụ, đĩa) lưu trữ mã để thực hiện phương pháp hoặc các bước của nó theo phương án bất kỳ của sáng chế. Ví dụ, hệ thống theo sáng chế có thể là hoặc có thể bao gồm bộ xử lý đa năng lập trình được, bộ xử lý tín hiệu số, hoặc bộ vi xử lý, được lập trình với phần mềm hoặc phần sụn và/hoặc theo cách khác được tạo cấu hình để thực hiện hoạt động bất kỳ trong số các hoạt động trên dữ liệu, bao gồm phương pháp hoặc các bước của nó theo một phương án của sáng chế. Bộ xử lý đa năng như vậy có thể là hoặc có thể bao gồm hệ thống máy tính bao gồm thiết bị đầu vào, bộ nhớ, và mạch xử lý được lập trình (và/hoặc theo cách khác được tạo cấu hình) để thực hiện phương pháp (hoặc các bước của nó) theo một phương án của sáng chế để xử lý dữ liệu được gán vào đó.

Theo một số phương án, sáng chế là các bộ mã hóa (ví dụ, các bộ mã hóa mã hóa dữ liệu âm thanh theo định dạng Dolby Digital Plus, AC-3, hoặc Dolby E) hoặc các bộ giải mã, được thực hiện như các bộ xử lý được lập trình (ví dụ, các bộ xử lý Neon ARM, mỗi bộ xử lý này là bộ xử lý Cortex ARM với máy SIMD Neon cho phép xử lý song song, hoặc các bộ xử lý khác có các đơn vị một lệnh, đa dữ liệu (single instruction, multiple data – SIMD) và/hoặc nhiều đơn vị logic số học (arithmetic logic unit - ALU) hoặc đơn vị thao tác số học (arithmetic manipulation unit - AMU)) hoặc các bộ xử lý tín hiệu số được lập trình (và/hoặc theo cách khác được tạo cấu hình) (ví dụ, các DSP có các đơn vị SIMD và/hoặc nhiều đơn vị ALU hoặc AMU).

Việc kết hợp các hoạt động của tất cả các tầng lọc trùng phương của bộ lọc trùng phương đa tầng (trong một vòng lặp mẫu) theo các phương án thông thường của sáng chế giúp cải thiện hiệu quả bằng cách cho phép song song hóa. Các bộ xử lý có các đơn vị SIMD và nhiều đơn vị ALU (hoặc AMU) có thể sử dụng hiệu quả các tài nguyên của chúng bằng cách thực hiện phương pháp theo một phương án của sáng chế.

Theo các phương án thông thường, phương pháp cài đặt bộ lọc trùng phương đa tầng không ảnh hưởng đến độ chính xác của đầu ra hoặc độ ổn định của bộ lọc (so với độ chính xác và độ ổn định có thể đạt được bởi cách cài đặt thông thường của bộ lọc).

Các thử nghiệm được tiến hành bởi tác giả sáng chế đã cho thấy rằng bộ mã hóa, được tạo cấu hình để mã hóa dữ liệu âm thanh theo định dạng Dolby Digital Plus, và được cài đặt làm bộ xử lý tín hiệu số Texas Instrument C64 được lập trình để bao gồm bộ lọc trùng phương hai tầng theo một phương án của sáng chế (thực hiện lọc thông cao trong hệ thống con dò tạm thời của bộ mã hóa) chỉ cần trung bình 1846 vòng để lọc một khối dữ liệu âm thanh thông thường, trái với số vòng trung bình (4141) cần thiết để lọc khối khi bộ mã hóa được lập trình theo cách thông thường để bao gồm phương án thực hiện thông thường (không được song song hóa) của bộ lọc hai tầng.

Các thử nghiệm được thực hiện bởi tác giả sáng chế cũng cho thấy rằng bộ mã hóa, được tạo cấu hình để mã hóa dữ liệu âm thanh theo định dạng Dolby Digital Plus, và được cài đặt làm bộ xử lý tín hiệu số Texas Instrument C64 được lập trình để bao gồm bộ lọc trùng phương bốn tầng theo một phương án của sáng chế (thực hiện lọc thông thấp ở các hệ thống con hiệu ứng tần số thấp (low frequency effects - LFE) của bộ mã hóa) chỉ cần trung bình 5802 vòng để lọc một khối dữ liệu âm thanh thông thường, trái với số lượng vòng trung bình (10375) cần để lọc khối khi bộ mã hóa được lập trình theo cách thông thường để bao gồm phương án thực hiện thông thường (không được song song hóa) của bộ lọc bốn tầng.

Sáng chế được mong đợi cũng có thể mang lại các ưu điểm hiệu suất tương tự khi bộ lọc theo sáng chế được cài đặt bằng cách lập trình thích hợp các bộ xử lý khác (có cấu trúc bộ xử lý lõi khác). Cũng mong đợi rằng mức độ cải thiện hiệu suất sẽ phụ thuộc vào cấu trúc bộ xử lý, số lượng tầng của bộ lọc, và số lượng điểm cực trong bộ lọc.

Sáng chế có thể được cài đặt trong phần cứng, phần sụn, hoặc phần mềm, hoặc kết hợp của hai phần (ví dụ, dưới dạng mảng logic lập trình được). Trừ khi có quy định khác, các thuật toán hoặc các quy trình xử lý được bao gồm như là một phần của sáng chế vốn không liên quan tới máy tính cụ thể hoặc thiết bị khác. Cụ thể, các máy đa năng khác nhau có thể được sử dụng với các chương trình được viết ra theo các phần bộc lộ trong bản mô tả này, hoặc có thể thuận tiện hơn nếu tạo ra thiết bị chuyên môn hóa hơn nữa (ví dụ, mạch tích hợp) để thực hiện các bước của phương pháp được yêu cầu. Do đó, sáng chế có thể được cài đặt trong một hoặc nhiều chương trình máy tính chạy trên một hoặc nhiều hệ thống máy tính lập trình được (ví dụ, hệ thống máy tính cài đặt bộ mã hóa trên Fig.5), mỗi hệ thống này bao gồm ít nhất một bộ xử lý, ít nhất một hệ thống lưu trữ dữ liệu (bao gồm bộ nhớ khả biến và bất khả biến và/hoặc các phần tử lưu trữ), ít nhất một thiết bị hoặc cổng đầu vào, và ít nhất một thiết bị hoặc cổng đầu ra. Mã chương trình được sử dụng để nhập dữ liệu để thực hiện các chức năng được mô tả trong bản mô tả này và tạo ra thông tin đầu ra. Thông tin đầu ra được áp dụng cho một hoặc nhiều thiết bị đầu ra theo cách đã biết.

Mỗi chương trình như vậy có thể được cài đặt bằng ngôn ngữ máy tính mong muốn bất kỳ (bao gồm ngôn ngữ máy, hợp ngữ, hoặc ngôn ngữ thủ tục cấp cao, ngôn ngữ logic, hoặc ngôn ngữ lập trình định hướng vật thể) để giao tiếp với hệ thống máy tính. Trong bất kỳ trường hợp nào, ngôn ngữ này có thể là ngôn ngữ biên dịch hoặc ngôn ngữ diễn dịch.

Ví dụ, khi được thực hiện bởi các chuỗi lệnh phần mềm máy tính, các chức năng và các bước khác nhau theo các phương án của sáng chế có thể được thực hiện bởi các chuỗi lệnh phần mềm đa luồng chạy trong phần cứng xử lý tín hiệu số thích

hợp, trong trường hợp này các thiết bị, các bước, và các chức năng khác nhau theo các phương án có thể tương ứng với các phần của các chỉ lệnh phần mềm.

Mỗi chương trình máy tính như vậy tốt hơn là được lưu trữ trên hoặc được tải về phương tiện hoặc thiết bị lưu trữ (ví dụ, bộ nhớ hoặc phương tiện thể rắn, hoặc phương tiện từ hoặc quang) có thể đọc được bởi máy tính lập trình chuyên dụng hoặc đa dụng, để tạo cấu hình và vận hành máy tính khi phương tiện hoặc thiết bị lưu trữ được đọc bởi hệ thống máy tính để thực hiện các quy trình được mô tả trong bản mô tả này. Hệ thống theo sáng chế cũng có thể được cài đặt làm phương tiện lưu trữ đọc được bởi máy tính, được tạo cấu hình với (tức là, lưu trữ) chương trình máy tính, trong đó phương tiện lưu trữ được tạo cấu hình như vậy khiến cho hệ thống máy tính hoạt động theo cách cụ thể và định trước để thực hiện các chức năng được mô tả trong bản mô tả này.

Dự tính rằng các bộ lọc đa tầng, có các tầng riêng lẻ là các bộ lọc IIR chứ không phải là các bộ lọc trùng phương (như trong các phương án cụ thể được mô tả trong bản mô tả này), có thể được cài đặt theo sáng chế sao cho quy trình xử lý của các tầng riêng biệt của nó được song song hóa (ví dụ, sao cho tất cả các tầng của nó có thể hoạt động độc lập đáp lại dòng lệnh chung duy nhất, để thực hiện việc xử lý dữ liệu song song hóa hoàn toàn trong các tầng này). Ví dụ, bộ lọc đa tầng thuộc loại được mô tả trong Công bố đơn Patent Mỹ số 2012/0019723 A1, công bố ngày 26 tháng 1 năm 2012, có thể được cải biến theo một phương án của sáng chế sao cho việc xử lý các tầng riêng lẻ của nó được song song hóa (ví dụ, sao cho tất cả các tầng của nó có thể hoạt động độc lập đáp lại một dòng lệnh chung duy nhất, để thực hiện việc xử lý dữ liệu song song hóa hoàn toàn trong các tầng này).

Theo một số phương án của phương pháp theo sáng chế, một số hoặc tất cả các bước được mô tả trong bản mô tả này được thực hiện đồng thời hoặc theo một thứ tự khác so với thứ tự nêu trong các ví dụ được mô tả trong bản mô tả này. Mặc dù các bước được thực hiện theo thứ tự cụ thể theo một số phương án của phương pháp theo sáng chế, nhưng một số bước có thể được thực hiện đồng thời hoặc theo thứ tự khác theo các phương án khác.

Nhiều phương án theo sáng chế đã được mô tả. Tuy nhiên, cần hiểu rằng các cải biến khác nhau có thể được thực hiện mà không nằm ngoài mục đích và phạm vi của sáng chế. Có thể thực hiện được rất nhiều cải biến và biến thể của sáng chế dựa trên các bộc lộ trên đây. Cần hiểu rằng trong phạm vi của các điểm yêu cầu bảo hộ đính kèm, sáng chế có thể được thực hiện theo cách bất kỳ như được nêu cụ thể trong bản mô tả này.

YÊU CẦU BẢO HỘ

1. Bộ mã hóa âm thanh được tạo cấu hình để tạo ra dữ liệu âm thanh đã mã hóa đáp lại dữ liệu âm thanh đầu vào, bộ mã hóa này bao gồm ít nhất một bộ lọc đa tầng được nối và tạo cấu hình để lọc dữ liệu âm thanh, trong đó bộ lọc đa tầng này bao gồm:

bộ nhớ đệm;

ít nhất hai tầng lọc trùng phương, bao gồm tầng lọc trùng phương thứ nhất và tầng lọc trùng phương tiếp theo; và

bộ điều khiển, được nối với các tầng lọc trùng phương và được tạo cấu hình để gán một dòng lệnh duy nhất cho cả tầng lọc trùng phương thứ nhất và tầng lọc trùng phương tiếp theo, trong đó tầng lọc trùng phương thứ nhất và tầng lọc trùng phương tiếp theo hoạt động một cách độc lập và song song để đáp lại dòng lệnh đó,

trong đó tầng lọc trùng phương thứ nhất được nối với bộ nhớ và được tạo cấu hình để thực hiện lọc trùng phương trên khối gồm N mẫu đầu vào đáp lại dòng lệnh nhằm tạo ra các trị số trung gian, và để gán các trị số trung gian cho bộ nhớ, trong đó các trị số trung gian bao gồm phiên bản được lọc của mỗi trong số ít nhất một tập con các mẫu đầu vào, và

trong đó tầng lọc trùng phương tiếp theo được nối với bộ nhớ và được tạo cấu hình để thực hiện lọc trùng phương trên các trị số đệm được khôi phục từ bộ nhớ đáp lại dòng lệnh nhằm tạo ra khối các trị số đầu ra, trong đó các trị số đầu ra bao gồm trị số đầu ra tương ứng với mỗi trong số các mẫu đầu vào trong khối gồm N mẫu đầu vào, và các trị số đệm bao gồm ít nhất một số trong số các trị số trung gian được tạo ra ở tầng lọc trùng phương thứ nhất đáp lại khối gồm N mẫu đầu vào;

trong đó bộ lọc đa tầng được tạo cấu hình để thực hiện lọc đa tầng khối N mẫu đầu vào trong một vòng xử lý duy nhất với sự lặp lại trên chỉ số mẫu chứ không lặp lại trên chỉ số lọc trùng phương.

2. Bộ mã hóa theo điểm 1, trong đó bộ lọc đa tầng này có M tầng lọc trùng phương, tầng lọc trùng phương tiếp theo được tạo cấu hình để tạo ra trị số đầu ra tương ứng với mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào đáp lại tập con các trị số đệm

được khôi phục từ bộ nhớ, trong đó j là chỉ số nằm trong phạm vi từ $M-1$ đến $N-1$, tập con này bao gồm phiên bản đã lọc của mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào, phiên bản đã lọc của mẫu đầu vào thứ “ $j-1$ ” trong số các mẫu đầu vào, và phiên bản đã lọc của mẫu đầu vào thứ “ $j-2$ ” trong số các mẫu đầu vào.

3. Bộ mã hóa theo điểm 1, trong đó bộ lọc đa tầng này có M tầng lọc trùng phương, tầng lọc trùng phương tiếp theo được tạo cấu hình để tạo ra trị số đầu ra tương ứng với mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào đáp lại tập con các trị số đệm được khôi phục từ bộ nhớ, trong đó j là chỉ số nằm trong khoảng từ $M-1$ đến $N-1$, tập con này bao gồm phiên bản được lọc của mẫu đầu vào thứ “ j ” trong số các mẫu đầu vào được tạo ra bởi tầng lọc trùng phương thứ nhất, và trị số được tạo ra bởi tầng lọc trùng phương tiếp theo đáp lại mẫu đầu vào thứ “ $j-1$ ” trong số các mẫu đầu vào, và trị số được tạo ra bởi tầng lọc trùng phương tiếp theo đáp lại mẫu đầu vào thứ “ $j-2$ ” trong số các mẫu đầu vào.

4. Bộ mã hóa theo điểm 1, trong đó tầng lọc trùng phương tiếp theo được tạo cấu hình để tạo ra trị số đầu ra tương ứng với mỗi mẫu đầu vào trong số các mẫu đầu vào đáp lại tập con khác gồm các giá trị đệm được khôi phục từ bộ nhớ, mỗi tập con bao gồm ít nhất ba trong số các trị số trung gian được tạo ra trong tầng lọc trùng phương thứ nhất và khôi phục từ bộ nhớ sau khi nằm trong bộ nhớ trong các khoảng thời gian chờ khác nhau.

5. Bộ mã hóa theo điểm 1, trong đó bộ mã hóa là bộ xử lý bao gồm ít nhất một đơn vị một lệnh, đa dữ liệu được lập trình để thực hiện bộ lọc đa tầng.

6. Bộ mã hóa theo điểm 1, trong đó bộ mã hóa là bộ xử lý bao gồm nhiều đơn vị logic số học được lập trình để thực hiện các tầng lọc trùng phương.

7. Bộ mã hóa theo điểm 1, trong đó bộ mã hóa là bộ xử lý bao gồm nhiều đơn vị thao tác số học được lập trình để thực hiện các tầng lọc trùng phương.

8. Phương pháp mã hóa dữ liệu âm thanh để tạo ra dữ liệu âm thanh được mã hóa, bao gồm bằng cách thực hiện lọc đa tầng trên khối gồm N mẫu dữ liệu âm thanh, trong đó lọc đa tầng bao gồm các bước:

(a) thực hiện hoạt động lọc trùng phương thứ nhất trên khối gồm N mẫu để tạo ra các trị số trung gian, và gán các trị số trung gian này cho bộ nhớ đệm, trong đó các trị số trung gian bao gồm phiên bản được lọc của mỗi trong số ít nhất một tập con gồm N mẫu; và

(b) thực hiện hoạt động lọc trùng phương thứ hai trên các trị số đệm được khôi phục từ bộ nhớ, để tạo ra khối các trị số đầu ra, trong đó trị số đầu ra bao gồm trị số đầu ra tương ứng với mỗi trong số các mẫu trong khối gồm N mẫu, tập con khác gồm các trị số đệm được khôi phục và lọc để tạo ra trị số đầu ra tương ứng với mỗi trong số các mẫu trong khối, và mỗi tập con các trị số đệm bao gồm ít nhất hai trong số các trị số trung gian được tạo ra trong khi thực hiện bước (a), các trị số này được khôi phục từ bộ nhớ sau khi nằm trong bộ nhớ này trong các khoảng thời gian chờ khác nhau,

trong đó các bước (a) và (b) được thực hiện đáp lại một dòng lệnh duy nhất, sao cho các bước (a) và (b) được thực hiện độc lập và song song đáp lại dòng lệnh duy nhất đó;

trong đó việc lọc đa tầng khối N mẫu được thực hiện trong một vòng duy nhất với sự lặp lại trên chỉ số mẫu nhưng không có sự lặp lại trên chỉ số tầng lọc trùng phương.

9. Phương pháp theo điểm 8, trong đó bước lọc đa tầng được thực hiện trong bộ lọc đa tầng có M tầng, các trị số đệm được khôi phục ở bước (b) để tạo ra trị số đầu ra tương ứng với mẫu thứ “ j ” trong số các mẫu, trong đó j là chỉ số nằm trong khoảng từ $M-1$ đến $N-1$, bao gồm phiên bản được lọc của mẫu thứ “ j ” trong số các mẫu được tạo ra ở bước (a), phiên bản được lọc của mẫu thứ “ $j-1$ ” trong số các mẫu được tạo ra ở bước (a), và phiên bản được lọc của mẫu thứ “ $j-2$ ” trong số các mẫu được tạo ra ở bước (a).

10. Phương pháp xử lý dòng bit được mã hóa, phương pháp này bao gồm các bước:

nhận dòng bit được mã hóa và trích ra từ đó dữ liệu được mã hóa biểu diễn một hoặc nhiều kênh thông tin âm thanh được xử lý bởi bộ lọc đa tầng; và

giải mã dữ liệu được mã hóa để cung cấp các dạng biểu diễn được giải mã của một hoặc nhiều kênh thông tin âm thanh được xử lý bởi bộ lọc đa tầng, trong đó bộ lọc đa tầng được hoạt động để thực hiện một cách độc lập và song song, đáp lại một dòng lệnh duy nhất:

(a) hoạt động lọc trùng phương thứ nhất trên khối gồm N mẫu để tạo ra các trị số trung gian để gán cho bộ nhớ đệm, trong đó các trị số trung gian bao gồm phiên bản được lọc của mỗi trong số ít nhất một tập con gồm N mẫu; và

(b) hoạt động lọc trùng phương thứ hai trên các trị số đệm được khôi phục từ bộ nhớ để tạo ra khối các trị số đầu ra, trong đó các trị số đầu ra bao gồm trị số đầu ra tương ứng với mỗi trong số các mẫu trong khối gồm N mẫu, một tập con khác gồm các trị số đệm được khôi phục và lọc để tạo ra trị số đầu ra tương ứng với mỗi trong số các mẫu trong khối, và mỗi tập con các trị số đệm bao gồm ít nhất hai trong số các trị số trung gian được tạo ra trong khi thực hiện bước (a), các trị số này được khôi phục từ bộ nhớ sau khi nằm trong bộ nhớ trong các khoảng thời gian chờ khác nhau;

trong đó bộ lọc đa tầng được tạo cấu hình để thực hiện việc lọc đa tầng của khối gồm N mẫu trong một vòng xử lý duy nhất với sự lặp lại trên chỉ số mẫu nhưng không lặp lại trên chỉ số tầng lọc trùng phương.

11. Phương pháp theo điểm 10, trong đó bộ lọc đa tầng là bộ lọc trùng phương ba tầng để thực hiện lọc thông thấp giới hạn băng thông.

12. Phương pháp theo điểm 10, trong đó một trong số một hoặc nhiều kênh là kênh hiệu ứng tần số thấp (low frequency effects – LFE) và bộ lọc đa tầng là bộ lọc trùng phương bốn tầng để thực hiện lọc thông thấp trên kênh LFE.

13. Thiết bị xử lý dòng bit được mã hóa, thiết bị này bao gồm:

đầu vào được nối để nhận dòng bit được mã hóa; và

bộ giải mã được nối với đầu vào, và được tạo cấu hình để trích ra, từ dòng bit được mã hóa, dữ liệu được mã hóa biểu diễn một hoặc nhiều kênh thông tin âm thanh được xử lý bởi bộ lọc đa tầng, và để giải mã dữ liệu được mã hóa để cung cấp các dạng biểu diễn được giải mã của một hoặc nhiều kênh thông tin âm thanh được

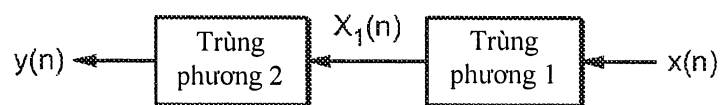
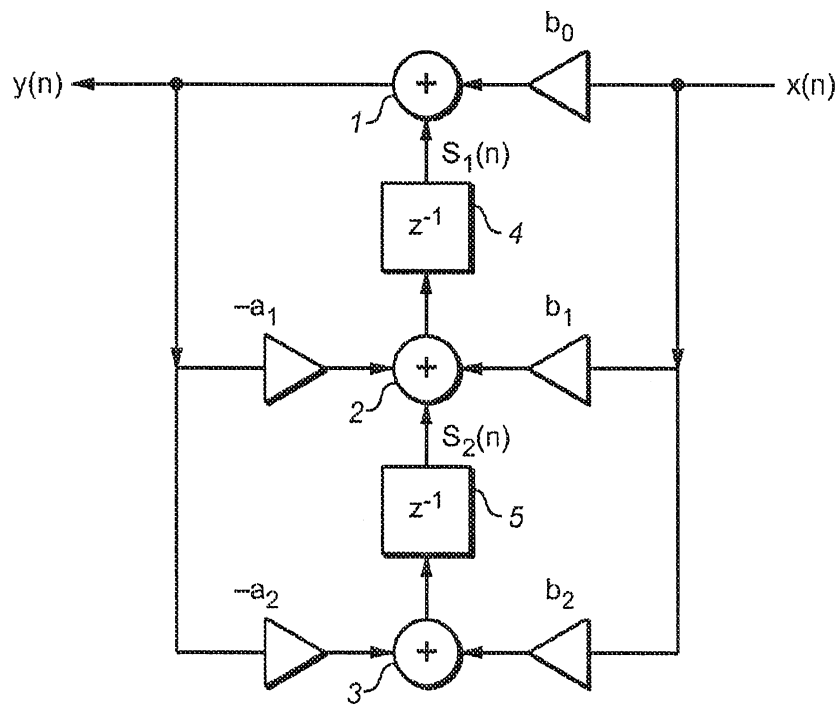
xử lý bởi bộ lọc đa tầng, trong đó bộ lọc đa tầng được hoạt động để thực hiện một cách độc lập và song song, đáp lại một dòng lệnh duy nhất:

- (a) hoạt động lọc trùng phương thứ nhất trên khối gồm N mẫu để tạo ra các trị số trung gian để gán cho bộ nhớ đệm, trong đó các trị số trung gian bao gồm phiên bản được lọc của mỗi trong số ít nhất một tập con gồm N mẫu; và
- (b) hoạt động lọc trùng phương thứ hai trên các trị số đệm được khôi phục từ bộ nhớ để tạo ra khối các trị số đầu ra, trong đó trị số đầu ra bao gồm trị số đầu ra tương ứng với mỗi trong số các mẫu trong khối gồm N mẫu, một tập con khác gồm các trị số đệm được khôi phục và lọc để tạo ra trị số đầu ra tương ứng với mỗi trong số các mẫu trong khối, và mỗi tập con các trị số đệm bao gồm ít nhất hai trong số các trị số trung gian được tạo ra trong khi thực hiện bước (a), các trị số này được khôi phục từ bộ nhớ sau khi nằm trong bộ nhớ với các khoảng thời gian chờ khác nhau;

trong đó bộ lọc đa tầng được tạo cấu hình để thực hiện việc lọc đa tầng của khối N mẫu trong một vòng xử lý duy nhất với sự lặp lại trên chỉ số mẫu nhưng không lặp lại trên chỉ số tầng lọc trùng phương.

14. Thiết bị theo điểm 13, trong đó bộ lọc đa tầng là bộ lọc trùng phương ba tầng để thực hiện lọc thông thấp giới hạn băng thông.

15. Thiết bị theo điểm 13, trong đó một trong số một hoặc nhiều kênh là kênh LFE và bộ lọc đa tầng là bộ lọc trùng phương bốn tầng để thực hiện lọc thông thấp trên kênh LFE.



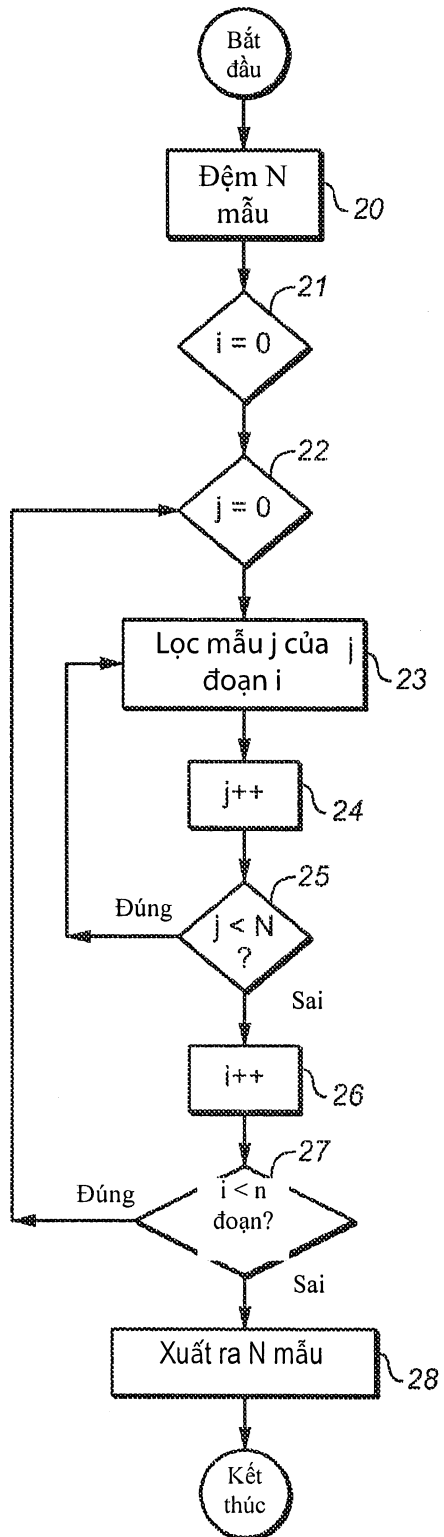


Fig.2

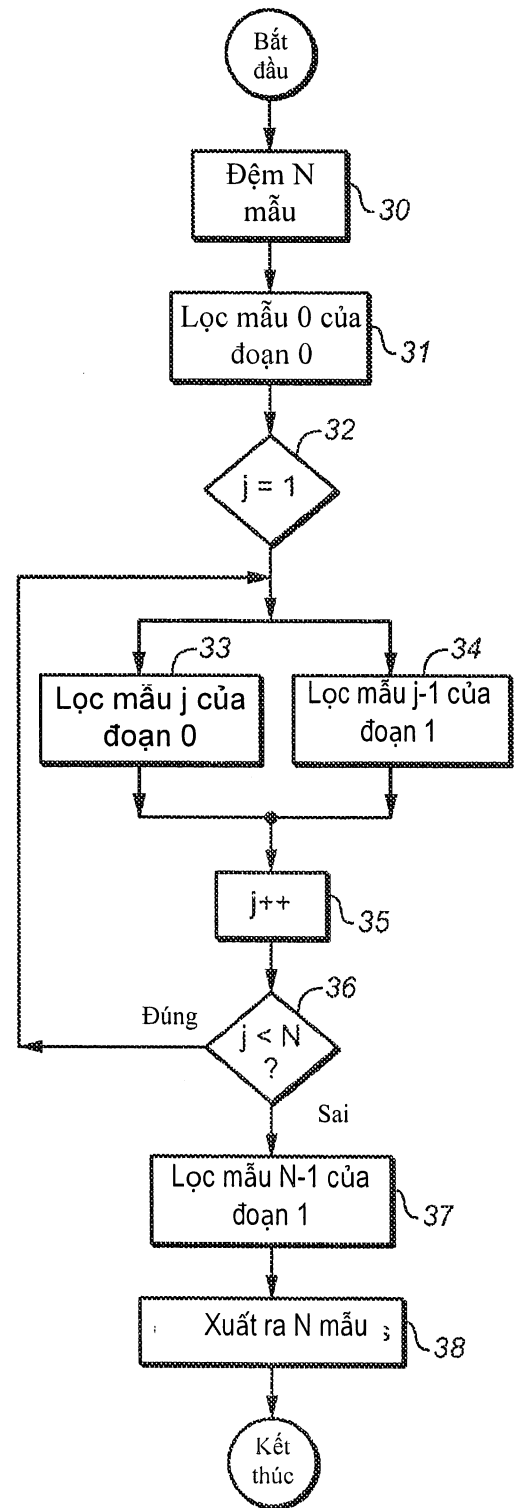


Fig.3

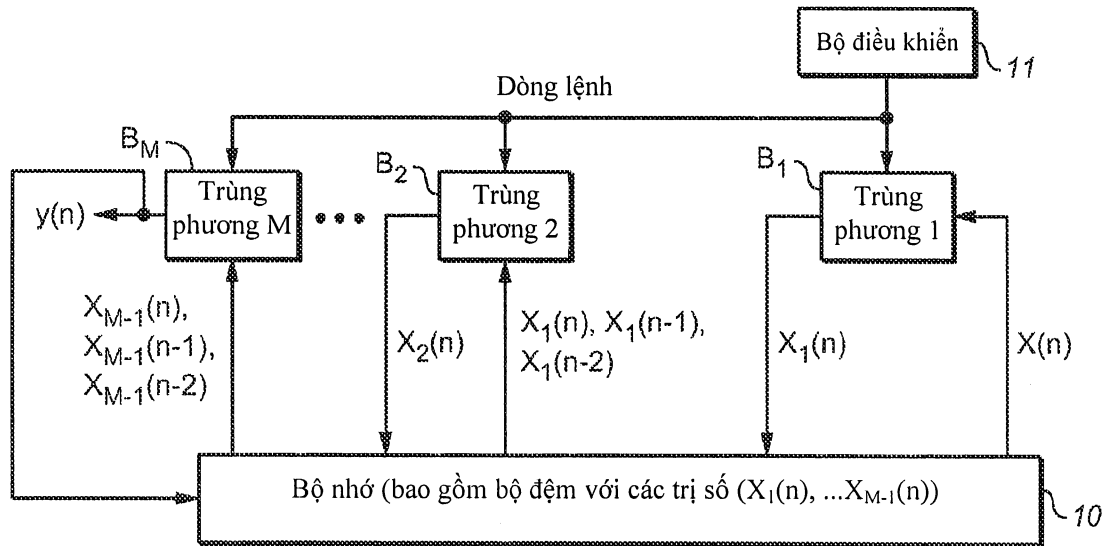


Fig.4

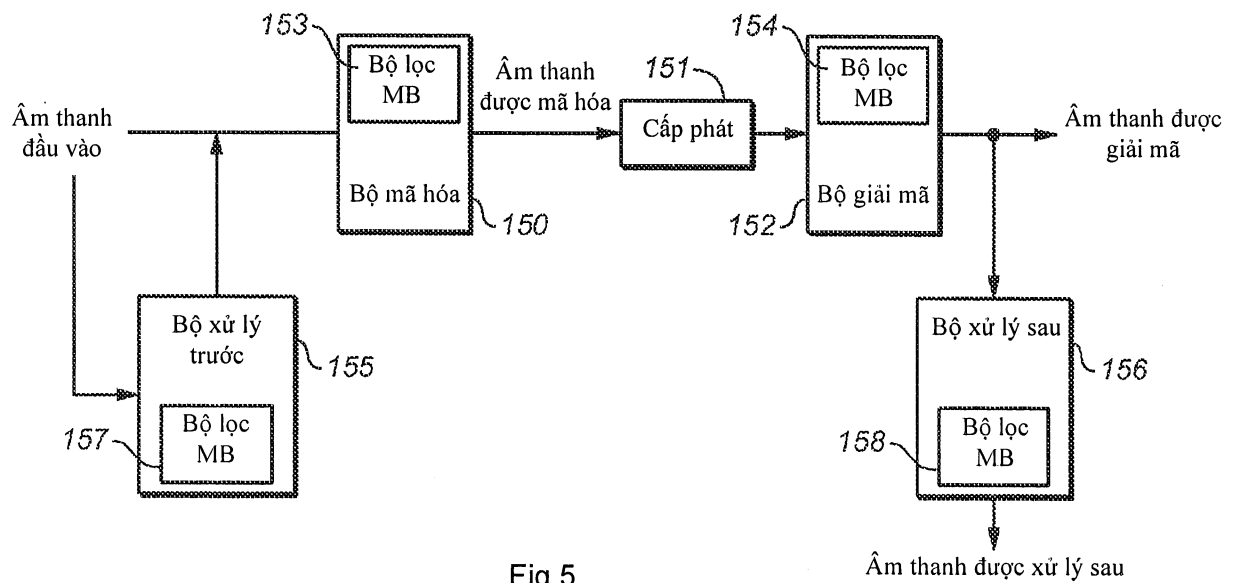


Fig.5

4 / 7

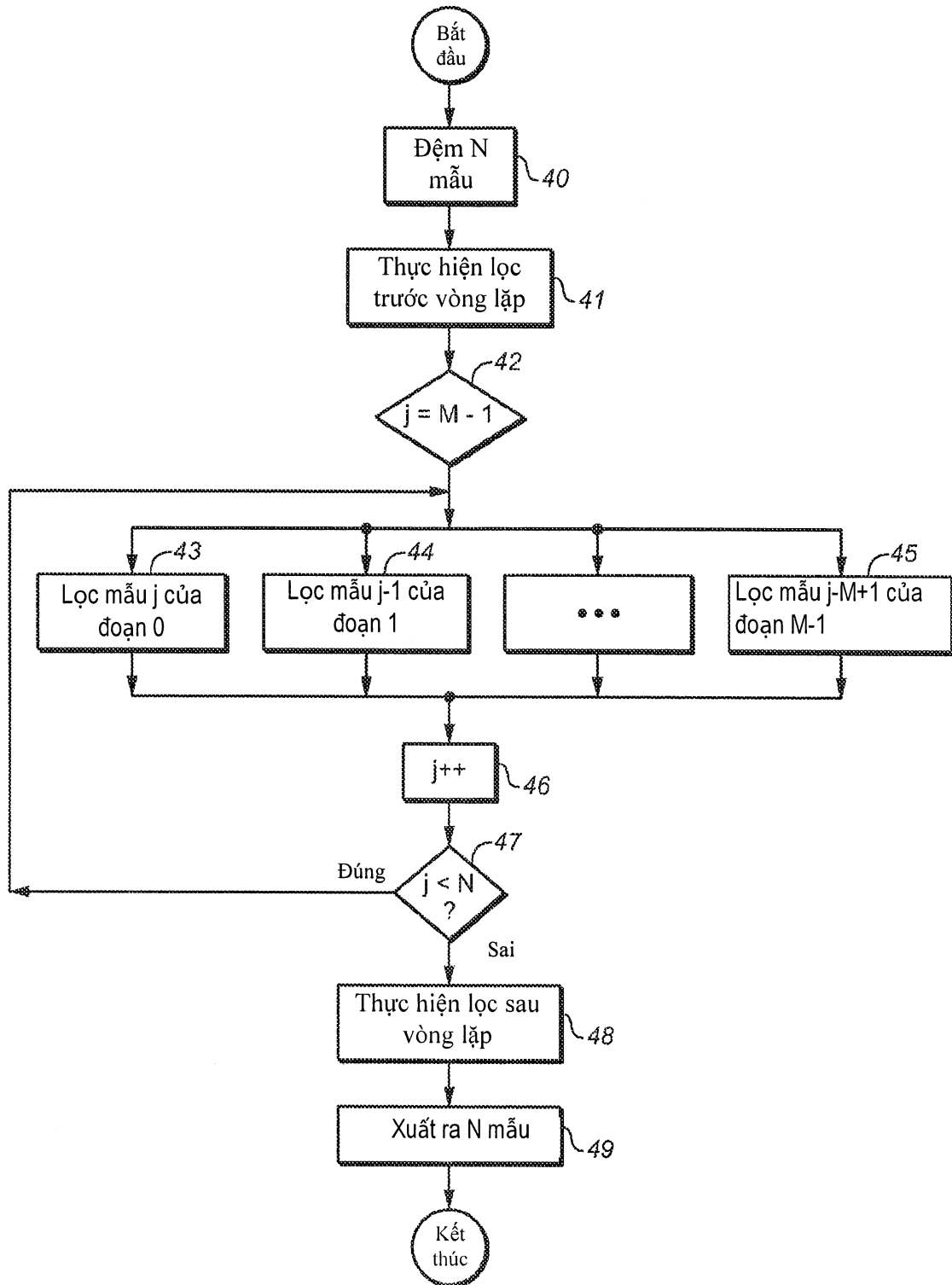


Fig.6

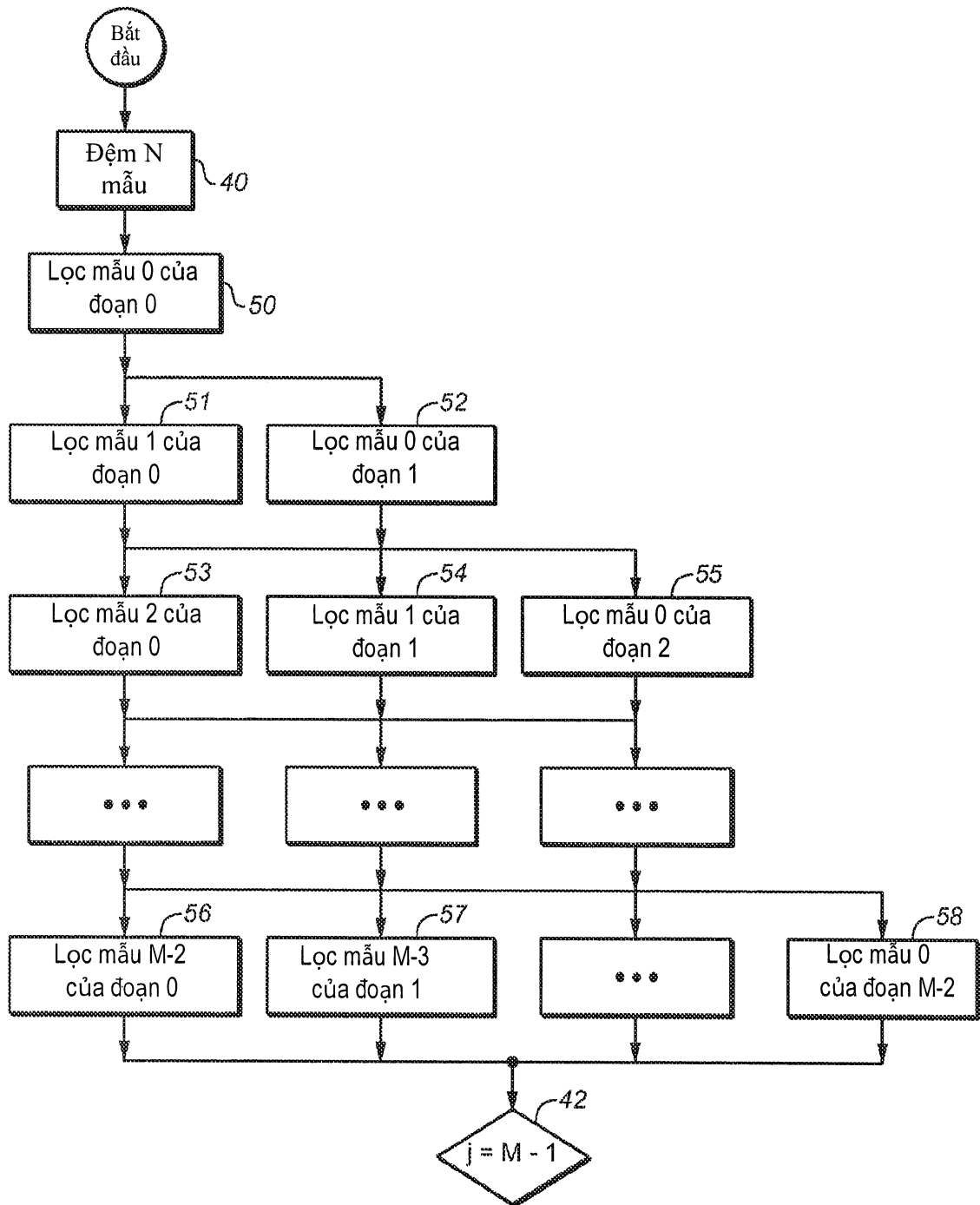


Fig.7

6 / 7

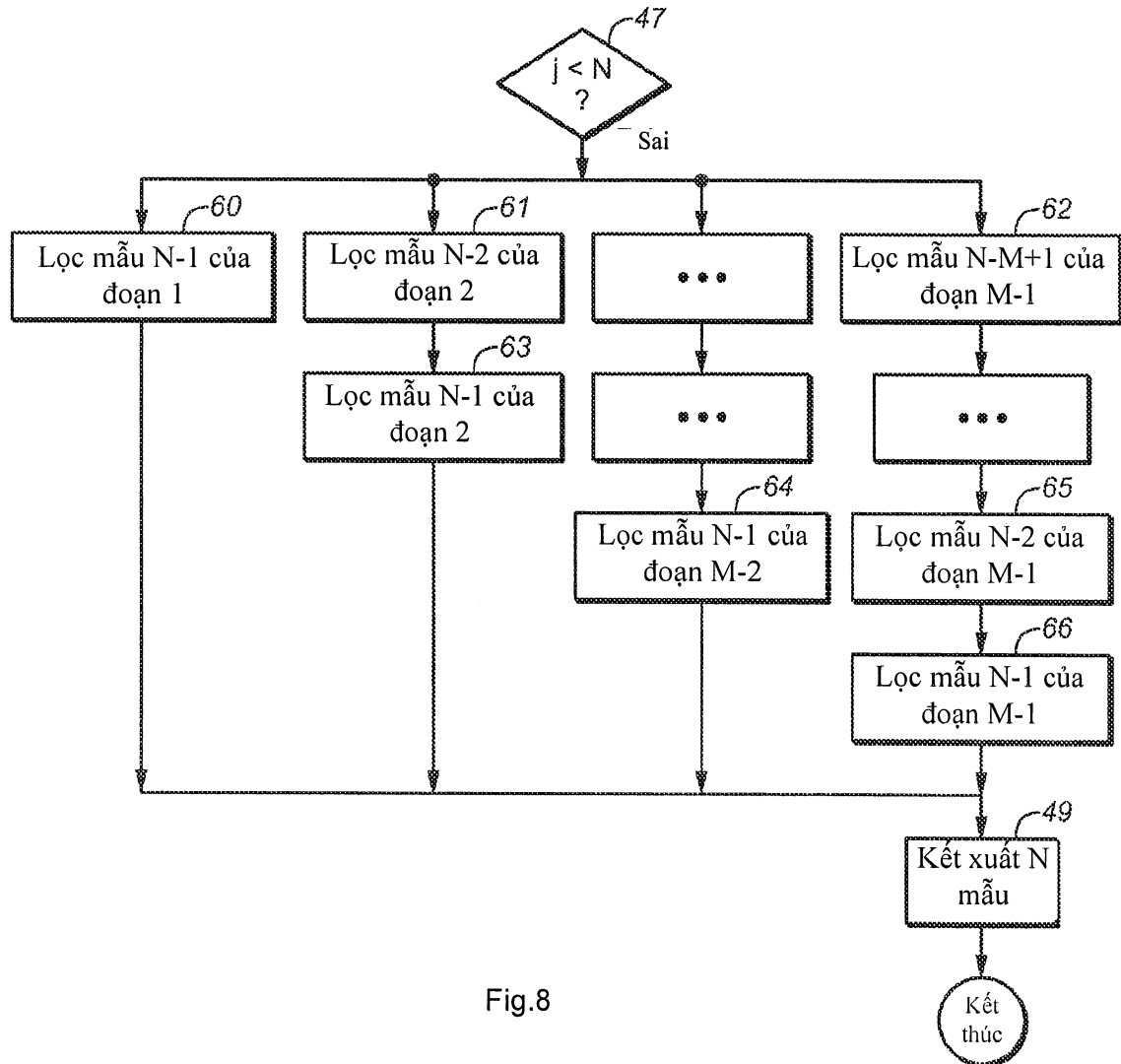


Fig.8

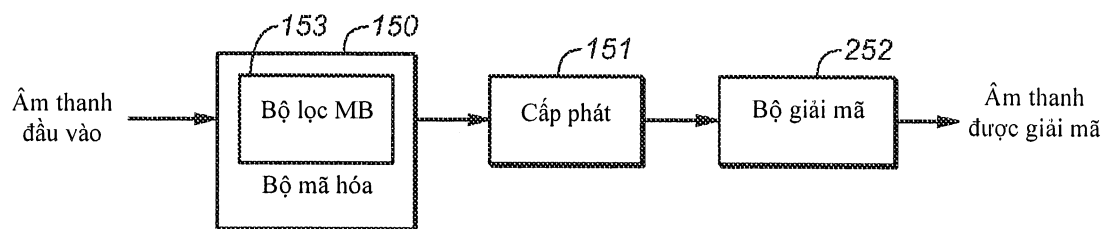


Fig.10

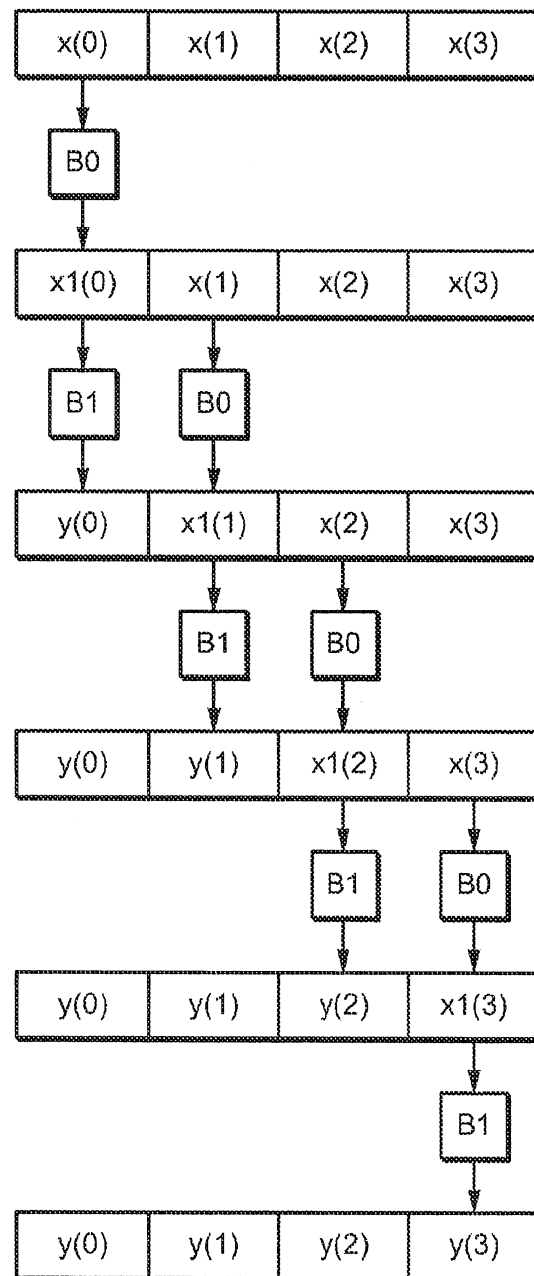


Fig.9