



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

(19) Cộng hòa xã hội chủ nghĩa Việt Nam (VN) (11)
CỤC SỞ HỮU TRÍ TUỆ



1-0043488

(51)^{2020.01} G10L 19/26; G10L 19/005; G10L 19/08 (13) B

(21) 1-2020-02794

(22) 06/11/2018

(86) PCT/EP2018/080350 06/11/2018

(87) WO 2019/091980 16/05/2019

(30) 17201099.3 10/11/2017 EP

(45) 25/02/2025 443

(43) 25/08/2020 389A

(73) Fraunhofer-Gesellschaft zur Foerderung der angewandten Forschung e. V. (DE)
Hansastrasse 27c, 80686 Muenchen, Germany

(72) RAVELLI, Emmanuel (FR); TOMASEK, Adrian (DE); LUTZKY, Manfred (DE);
BENNDORF, Conrad (DE).

(74) CÔNG TY LUẬT TRÁCH NHIỆM HỮU HẠN AMBYS HÀ NỘI (AMBYS
HANOI)

(54) THIẾT BỊ VÀ PHƯƠNG PHÁP GIẢI MÃ TÍN HIỆU ÂM THANH

(21) 1-2020-02794

(57) Sáng chế đề cập đến thiết bị và phương pháp giải mã tín hiệu âm thanh. Phía bộ mã hóa có thể xác định xem khung tín hiệu có hữu ích cho việc lọc sau dải hạn (LTPF) và/hoặc che mất gói tin (PLC) hay không và có thể mã hóa thông tin theo các kết quả xác định. Phía bộ giải mã có thể áp dụng LTPF và/hoặc PLC theo thông tin thu được từ bộ mã hóa.

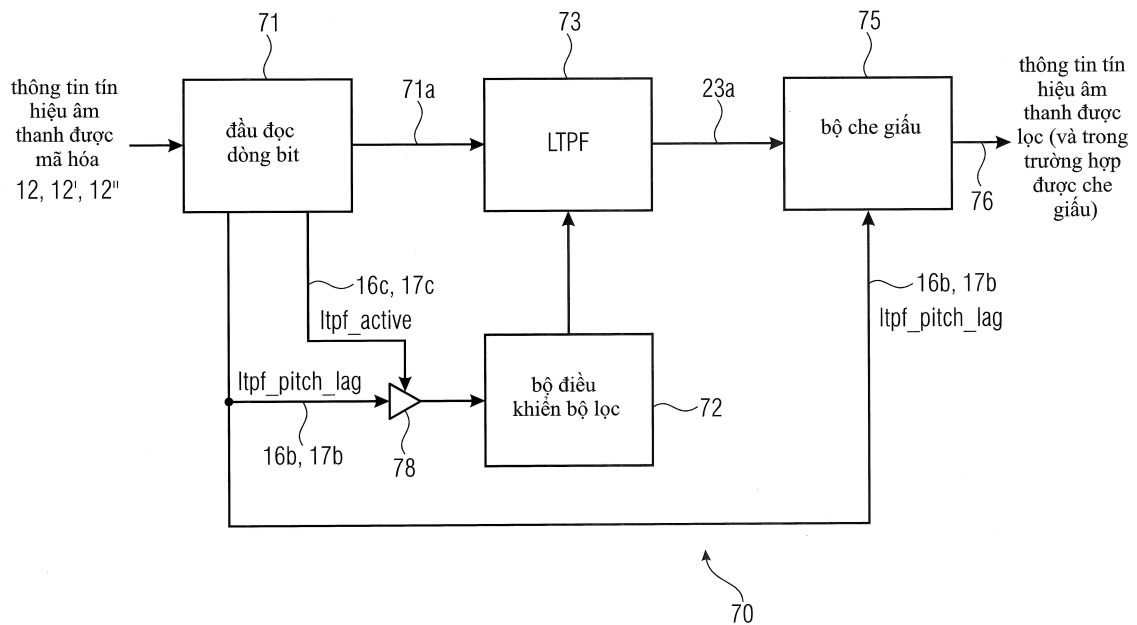


Fig. 7

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến các phương pháp và thiết bị để mã hóa/giải mã thông tin tín hiệu âm thanh.

Tình trạng kỹ thuật của sáng chế

Các bộ mã hóa-giải mã âm thanh trên cơ sở biến đổi nói chung đưa vào nhiều âm liên sóng hài khi xử lý các tín hiệu âm thanh sóng hài, cụ thể là ở độ trễ thấp và tốc độ bit thấp. Nhiều âm liên sóng hài này thường được nhận biết như thành phần lạ rất khó chịu, việc giảm đáng kể hiệu suất của bộ mã hóa-giải mã âm thanh trên cơ sở biến đổi khi được ước lượng một cách chủ quan trên vật liệu âm thanh có âm cao.

Lọc sau dài hạn (long term post filtering - LTPF) là công cụ để mã hóa âm thanh dựa trên biến đổi giúp giảm nhiều âm liên sóng hài này. Nó dựa trên bộ lọc sau mà được áp dụng lên tín hiệu miền thời gian sau khi giải mã biến đổi. Bộ lọc sau này về cơ bản là bộ lọc đáp ứng xung vô hạn (infinite impulse response - IIR) với đáp ứng tần số kiểu răng lược được điều khiển bởi các tham số như thông tin bước ghi (ví dụ, độ trễ bước ghi).

Để có độ mạnh tốt hơn, các tham số của bộ lọc sau (độ trễ bước ghi và, trong một số ví dụ, độ khuếch đại trên mỗi khung) được ước lượng ở phía bộ mã hóa và được mã hóa trong dòng bit, ví dụ, khi độ khuếch đại khác không. Trong các ví dụ, trường hợp độ khuếch đại bằng không được báo hiệu bằng một bit và tương ứng với bộ lọc sau không hoạt động, được sử dụng khi tín hiệu không chứa phần sóng hài.

LTPF lần đầu tiên được giới thiệu trong tiêu chuẩn 3GPP EVS [1] và sau đó được tích hợp với tiêu chuẩn âm thanh 3D MPEG-H [2]. Các tài liệu sáng chế tương ứng là [3] và [4].

Trong tình trạng kỹ thuật, các chức năng khác tại bộ giải mã có thể sử dụng thông tin bước ghi. Một ví dụ là che mất gói tin (packet loss concealment - PLC) hoặc che lỗi. PLC được sử dụng trong các bộ mã hóa-giải mã âm thanh để che các gói tin bị mất hoặc bị hỏng trong quá trình truyền từ bộ mã hóa tới bộ giải mã. Trong kỹ thuật trước, PLC có thể được thực hiện ở phía bộ giải mã và ngoại suy tín hiệu được giải mã

trong miền biến đổi hoặc trong miền thời gian. Lý tưởng nhất là tín hiệu được che phải không là tín hiệu nhân tạo và phải có cùng các đặc điểm phổ với tín hiệu bị mất. Mục tiêu này đặc biệt khó đạt được khi tín hiệu sẽ được che chứa cấu trúc sóng hài.

Trong trường hợp này, các kỹ thuật PLC dựa trên bước ghi có thể tạo ra các kết quả chấp nhận được. Các phương pháp này giả định rằng tín hiệu đứng yên cục bộ và khôi phục tín hiệu bị mất bằng cách tổng hợp tín hiệu định kỳ bằng cách sử dụng khoảng bước ghi ngoại suy. Các kỹ thuật này có thể được sử dụng trong mã hóa tiếng nói dựa trên CELP (xem ví dụ ITU-T G.718 [5]). Chúng cũng có thể được sử dụng để mã hóa PCM (ITU-T G.711 [6]). Và gần đây, chúng được áp dụng cho mã hóa âm thanh dựa trên MDCT, ví dụ tốt nhất là che miền thời gian TCX (TCX TD-PLC) trong tiêu chuẩn 3GPP EVS [7].

Thông tin bước ghi (có thể là độ trễ bước ghi) là tham số chính được sử dụng trong PLC dựa trên bước ghi. Tham số này có thể được ước lượng ở phía bộ mã hóa và được mã hóa vào dòng bit. Trong trường hợp này, độ trễ bước ghi của các khung tốt cuối cùng được sử dụng để che khung bị mất hiện thời (như trong ví dụ [5] và [7]). Nếu không có độ trễ bước ghi trong dòng bit, nó có thể được ước lượng ở phía bộ giải mã bằng cách chạy thuật toán phát hiện bước ghi trên tín hiệu được giải mã (như trong ví dụ tài liệu [6]).

Trong tiêu chuẩn 3GPP EVS (xem tài liệu [1] và [7]), cả LTPF và PLC dựa trên bước ghi đều được sử dụng trong cùng một bộ mã hóa-giải mã âm thanh TCX dựa trên MDCT. Cả hai công cụ có cùng tham số độ trễ bước ghi. Bộ mã hóa LTPF ước lượng và mã hóa tham số độ trễ bước ghi. Độ trễ bước ghi này có mặt trong dòng bit khi độ khuếch đại khác không. Ở phía bộ giải mã, bộ giải mã sử dụng thông tin này để lọc tín hiệu được giải mã. Trong trường hợp mất gói tin, PLC dựa trên bước ghi được sử dụng khi độ khuếch đại LTPF của khung tốt cuối cùng ở trên ngưỡng nhất định và các điều kiện khác được đáp ứng (chi tiết xem tài liệu [7]). Trong trường hợp đó, độ trễ bước ghi có trong dòng bit và nó có thể được sử dụng trực tiếp bởi môđun PLC.

Cú pháp dòng bit của kỹ thuật trước được cho bởi:

Syntax	No. of bits	Mnemonic
<pre> ltpf_data() { ltpf_active; if (ltpf_active) { ltpf_pitch_lag; ltpf_gain; } } </pre>	1 9 2	uimsbf uimsbf uimsbf

Tuy nhiên, một số vấn đề có thể phát sinh.

Tham số độ trễ bước ghi không được mã hóa trong dòng bit cho mọi khung. Khi độ khuếch đại bằng không trong khung (LTPF không hoạt động), không có thông tin độ trễ bước ghi nào xuất hiện trong dòng bit. Điều này có thể xảy ra khi nội dung sóng hài của tín hiệu không chiếm ưu thế và/hoặc đủ ổn định.

Theo đó, bằng cách phân biệt mã hóa độ trễ bước ghi trên cơ sở độ khuếch đại, không thể thu được độ trễ bước ghi nào bằng các chức năng khác (ví dụ, PLC).

Ví dụ, có những khung mà tín hiệu hơi hài, không đủ cho LTPF, nhưng đủ để sử dụng PLC dựa trên bước ghi. Trong trường hợp đó, cần tham số độ trễ ở phía bộ giải mã mặc dù nó không có trong dòng bit.

Có giải pháp là thêm bộ phát hiện bước ghi thứ hai ở phía bộ giải mã, nhưng điều này sẽ gây ra sự phức tạp đáng kể, đó là một vấn đề đối với các bộ mã hóa-giải mã âm thanh nhắm vào các thiết bị năng lượng thấp.

Tài liệu tham khảo

[1] 3GPP TS 26.445; Codec for Enhanced Voice Services (EVS); Detailed algorithmic description.

[2] ISO/IEC 23008-3:2015; Information technology -- High efficiency coding and media delivery in heterogeneous environments -- Part 3: 3D audio.

[3] Ravelli et al. "Apparatus and method for processing an audio signal using a harmonic post-filter." U.S. Patent Application No. 2017/0140769 A1. 18 May. 2017.

[4] Markovic et al. "Harmonicity-dependent controlling of a harmonic filter tool." U.S. Patent Application No. 2017/0133029 A1. 11 May. 2017.

[5] ITU-T G.718: Frame error robust narrow-band and wideband embedded variable bit-rate coding of speech and audio from 8-32 kbit/s.

[6] ITU-T G.711 Appendix I: A high quality low-complexity algorithm for packet loss concealment with G.711.

[7] 3GPP TS 26.447; Codec for Enhanced Voice Services (EVS); Error concealment of lost packets.

Bản chất kỹ thuật của sáng chế

Theo các ví dụ, mục đích của sáng chế là đề xuất thiết bị giải mã thông tin tín hiệu âm thanh liên quan đến tín hiệu âm thanh được chia thành chuỗi các khung, thiết bị bao gồm:

đầu đọc dòng bit được tạo cấu hình để đọc thông tin tín hiệu âm thanh được mã hóa có:

sự biểu diễn được mã hóa của tín hiệu âm thanh cho khung thứ nhất và khung thứ hai;

thông tin bước ghi thứ nhất cho khung thứ nhất và mục dữ liệu điều khiển thứ nhất có giá trị thứ nhất; và

thông tin bước ghi thứ hai cho khung thứ hai và mục dữ liệu điều khiển thứ hai có giá trị thứ hai khác với giá trị thứ nhất; và

bộ điều khiển được tạo cấu hình để điều khiển bộ lọc sau dải hạn LTPF để:

lọc sự biểu diễn được giải mã của tín hiệu âm thanh trong khung thứ hai bằng cách sử dụng thông tin bước ghi thứ hai khi mục dữ liệu điều khiển thứ hai có giá trị thứ hai; và

vô hiệu hóa LTPF cho khung thứ nhất khi mục dữ liệu điều khiển thứ nhất có giá trị thứ nhất.

Theo đó, thiết bị có thể phân biệt giữa các khung phù hợp với LTPF và các khung không phù hợp với LTPF, trong khi sử dụng các khung để che lỗi ngay cả khi LTPF không phù hợp. Ví dụ, trong trường hợp độ hài hòa cao hơn, thiết bị có thể sử dụng thông tin bước ghi (ví dụ, độ trễ bước ghi) cho LTPF. Trong trường hợp độ hài hòa thấp hơn, thiết bị có thể tránh sử dụng thông tin bước ghi cho LTPF, nhưng có thể sử dụng thông tin bước ghi cho các chức năng khác (ví dụ, che).

Theo các ví dụ, đầu đọc dòng bit được tạo cấu hình để đọc khung thứ ba, khung thứ ba có mục dữ liệu điều khiển biểu thị sự có mặt hay vắng mặt của thông tin bước ghi thứ nhất và/hoặc thông tin bước ghi thứ hai.

Theo các ví dụ, khung thứ ba có định dạng thiếu thông tin bước ghi thứ nhất, mục dữ liệu điều khiển thứ nhất, thông tin bước ghi thứ hai và mục dữ liệu điều khiển thứ hai.

Theo các ví dụ, mục dữ liệu điều khiển thứ ba được mã hóa trong một bit đơn có giá trị phân biệt khung thứ ba với khung thứ nhất và khung thứ hai.

Theo các ví dụ, trong thông tin tín hiệu âm thanh được mã hóa, đối với khung thứ nhất, một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ nhất và trường dữ liệu cố định được dành riêng cho thông tin bước ghi thứ nhất.

Theo các ví dụ, trong thông tin tín hiệu âm thanh được mã hóa, đối với khung thứ hai, một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ hai và trường dữ liệu cố định được dành riêng cho thông tin bước ghi thứ hai.

Theo các ví dụ, mục dữ liệu điều khiển thứ nhất và mục dữ liệu điều khiển thứ hai được mã hóa trong cùng một phần hoặc trường dữ liệu trong thông tin tín hiệu âm thanh được mã hóa.

Theo các ví dụ, thông tin tín hiệu âm thanh được mã hóa bao gồm một bit báo hiệu thứ nhất mã hóa mục dữ liệu điều khiển thứ ba; và trong trường hợp giá trị của mục dữ liệu điều khiển thứ ba (18e) biểu thị sự có mặt của thông tin bước ghi thứ nhất (16b) và/hoặc thông tin bước ghi thứ hai (17b), bit báo hiệu thứ hai mã hóa mục dữ liệu điều khiển thứ nhất (16c) và mục dữ liệu điều khiển thứ hai (17c).

Theo các ví dụ, thiết bị có thể còn bao gồm bộ che được tạo cấu hình để sử dụng thông tin bước ghi thứ nhất và/hoặc thứ hai để che khung âm thanh được giải mã không chính xác sau đó.

Theo các ví dụ, bộ che có thể được tạo cấu hình để, trong trường hợp xác định giải mã khung không hợp lệ, kiểm tra xem thông tin bước ghi liên quan đến khung được giải mã chính xác trước đó có được lưu trữ hay không, để che khung được giải mã không hợp lệ với khung thu được bằng cách sử dụng thông tin bước ghi được lưu trữ.

Theo đó, có thể thu được sự che tốt mỗi khi tín hiệu âm thanh phù hợp với việc che, và không chỉ khi tín hiệu âm thanh phù hợp với LTPF. Khi thu được thông tin bước ghi, không cần thiết phải ước tính độ trễ bước ghi, do đó làm giảm độ phức tạp.

Theo các ví dụ, sáng chế đề xuất thiết bị mã hóa các tín hiệu âm thanh, thiết bị bao gồm:

- bộ ước lượng bước ghi được tạo cấu hình để thu được thông tin bước ghi liên quan đến bước ghi của tín hiệu âm thanh;

- bộ phân tích tín hiệu được tạo cấu hình để thu được thông tin độ hài hòa liên quan đến sự hài hòa của tín hiệu âm thanh; và

- bộ tạo dòng bit được tạo cấu hình để chuẩn bị thông tin tín hiệu âm thanh được mã hóa mã hóa các khung để đưa vào dòng bit;

- sự biểu diễn được mã hóa của tín hiệu âm thanh cho khung thứ nhất, khung thứ hai, và khung thứ ba;

- thông tin bước ghi thứ nhất cho khung thứ nhất và mục dữ liệu điều khiển thứ nhất có giá trị thứ nhất;

- thông tin bước ghi thứ hai cho khung thứ hai và mục dữ liệu điều khiển thứ hai có giá trị thứ hai khác với giá trị thứ nhất; và

- mục dữ liệu điều khiển thứ ba cho khung thứ nhất, thứ hai và thứ ba,

- trong đó giá trị thứ nhất và giá trị thứ hai phụ thuộc vào các tiêu chí thứ hai liên quan đến thông tin độ hài hòa, và

giá trị thứ nhất biểu thị sự không đáp ứng các tiêu chí thứ hai về độ hài hòa của tín hiệu âm thanh trong khung thứ nhất, và

giá trị thứ hai biểu thị sự đáp ứng các tiêu chí thứ hai về độ hài hòa của tín hiệu âm thanh trong khung thứ hai,

trong đó các tiêu chí thứ hai bao gồm ít nhất một điều kiện được đáp ứng khi ít nhất một số đo độ hài hòa thứ hai lớn hơn ít nhất một ngưỡng thứ hai,

mục dữ liệu điều khiển thứ ba được mã hóa trong một bit đơn có giá trị phân biệt khung thứ ba với khung thứ nhất và khung thứ hai, khung thứ ba được mã hóa trong trường hợp không đáp ứng các tiêu chí thứ nhất và các khung thứ nhất và thứ hai được mã hóa trong trường hợp đáp ứng các tiêu chí thứ nhất, trong đó các tiêu chí thứ nhất bao gồm ít nhất một điều kiện được đáp ứng khi có ít nhất một số đo độ hài hòa thứ nhất lớn hơn ít nhất một ngưỡng thứ nhất,

trong đó trong dòng bit, đối với khung thứ nhất, một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ nhất và trường dữ liệu cố định được dành riêng cho thông tin bước ghi thứ nhất.

trong đó trong dòng bit, đối với khung thứ hai, một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ hai và trường dữ liệu cố định được dành riêng cho thông tin bước ghi thứ hai, và

trong đó trong dòng bit, đối với khung thứ ba, không có bit nào được dành riêng cho trường dữ liệu cố định và/hoặc cho mục điều khiển thứ nhất và thứ hai.

Theo đó, bộ giải mã có thể phân biệt giữa các khung hữu ích cho LTPF, các khung chỉ hữu ích cho PLC, và các khung vô ích cho cả LTPF và PLC.

Theo các ví dụ, các tiêu chí thứ hai bao gồm điều kiện bổ sung được đáp ứng khi có ít nhất một số đo độ hài hòa của khung trước lớn hơn ít nhất một ngưỡng thứ hai.

Theo các ví dụ, bộ phân tích tín hiệu được tạo cấu hình để xác định xem tín hiệu có ổn định giữa hai khung liên tiếp làm điều kiện cho các tiêu chí thứ hai hay không.

Theo đó, bộ giải mã có thể phân biệt, ví dụ, giữa tín hiệu ổn định và tín hiệu không ổn định. Trong trường hợp tín hiệu không ổn định, bộ giải mã có thể tránh sử dụng thông tin bước ghi cho LTPF, nhưng có thể sử dụng thông tin bước ghi cho các chức năng khác (ví dụ, che).

Theo các ví dụ, các số đo độ hài hòa thứ nhất và thứ hai thu được ở các mức lấy mẫu khác nhau.

Theo các ví dụ, thông tin bước ghi bao gồm thông tin độ trễ hoặc bản được xử lý của nó.

Theo các ví dụ, thông tin độ hài hòa bao gồm ít nhất một trong các giá trị tự tương quan và/hoặc giá trị tự tương quan chuẩn hóa và/hoặc phiên bản được xử lý của chúng.

Theo các ví dụ, sáng chế là đề xuất phương pháp giải mã thông tin tín hiệu âm thanh liên quan đến tín hiệu âm thanh được chia thành chuỗi các khung, phương pháp bao gồm các bước:

đọc thông tin tín hiệu âm thanh được mã hóa bao gồm:

sự biểu diễn được mã hóa của tín hiệu âm thanh cho khung thứ nhất và khung thứ hai;

thông tin bước ghi thứ nhất cho khung thứ nhất và mục dữ liệu điều khiển thứ nhất (16c) có giá trị thứ nhất;

thông tin bước ghi thứ hai cho khung thứ hai và mục dữ liệu điều khiển thứ hai có giá trị thứ hai khác với giá trị thứ nhất,

khi xác định rằng mục dữ liệu điều khiển thứ nhất có giá trị thứ nhất, sẽ sử dụng thông tin bước ghi thứ nhất cho bộ lọc sau dài hạn LTPF, và

khi xác định giá trị thứ hai của mục dữ liệu điều khiển thứ hai (17c), sẽ vô hiệu hóa LTPF.

Theo các ví dụ, phương pháp còn bao gồm bước, khi xác định rằng mục dữ liệu điều khiển thứ nhất hoặc thứ hai có giá trị thứ nhất hoặc thứ hai, sử dụng thông tin bước ghi thứ nhất hoặc thứ hai cho chức năng che lỗi.

Theo các ví dụ, sáng chế đề xuất phương pháp mã hóa thông tin tín hiệu âm thanh liên quan đến tín hiệu được chia thành các khung, phương pháp bao gồm các bước:

thu các số đo từ tín hiệu âm thanh;

xác minh việc đáp ứng các tiêu chí thứ hai, các tiêu chí thứ hai dựa trên các số đo và bao gồm ít nhất một điều kiện được đáp ứng khi ít nhất một số đo độ hài hòa thứ hai lớn hơn ngưỡng thứ hai;

tạo thông tin tín hiệu âm thanh được mã hóa có các khung bao gồm:

sự biểu diễn được mã hóa của tín hiệu âm thanh cho khung thứ nhất và khung thứ hai và khung thứ ba;

thông tin bước ghi thứ nhất cho khung thứ nhất và mục dữ liệu điều khiển thứ nhất có giá trị thứ nhất và mục dữ liệu điều khiển thứ ba;

thông tin bước ghi thứ hai cho khung thứ hai và mục dữ liệu điều khiển thứ hai có giá trị thứ hai khác với giá trị thứ nhất và mục dữ liệu điều khiển thứ ba,

trong đó giá trị thứ nhất và giá trị thứ hai phụ thuộc vào các tiêu chí thứ hai, và giá trị thứ nhất biểu thị sự không đáp ứng các tiêu chí thứ hai trên cơ sở độ hài hòa của tín hiệu âm thanh trong khung thứ nhất, và giá trị thứ hai biểu thị sự đáp ứng các tiêu chí thứ hai trên cơ sở độ hài hòa của tín hiệu âm thanh trong khung thứ hai,

mục dữ liệu điều khiển thứ ba là một bit đơn có giá trị phân biệt khung thứ ba với khung thứ nhất và khung thứ hai liên quan đến sự đáp ứng các tiêu chí thứ nhất, để xác định khung thứ ba khi mục dữ liệu điều khiển thứ ba biểu thị sự không đáp ứng các tiêu chí thứ nhất trên cơ sở ít nhất một điều kiện được đáp ứng khi có ít nhất một số đo độ hài hòa thứ nhất lớn hơn ít nhất một ngưỡng thứ nhất,

trong đó thông tin tín hiệu âm thanh được mã hóa được tạo sao cho, đối với khung thứ nhất, một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ nhất và trường dữ liệu cố định cho thông tin bước ghi thứ nhất, và

trong đó thông tin tín hiệu âm thanh được mã hóa được tạo sao cho, đối với khung thứ hai, một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ hai và trường dữ liệu cố định cho thông tin bước ghi thứ hai, và

trong đó thông tin tín hiệu âm thanh được mã hóa được tạo sao cho, đối với khung thứ ba, không có bit nào được dành riêng cho trường dữ liệu cố định và không có bit nào được dành riêng cho mục dữ liệu điều khiển thứ nhất và mục dữ liệu điều khiển thứ hai.

Theo các ví dụ, sáng chế đề xuất phương pháp bao gồm các bước:

mã hóa tín hiệu âm thanh;

truyền thông tin tín hiệu âm thanh được mã hóa đến bộ giải mã hoặc lưu trữ thông tin tín hiệu âm thanh được mã hóa;

giải mã thông tin tín hiệu âm thanh.

Theo các ví dụ, sáng chế đề xuất phương pháp mã hóa/giải mã các tín hiệu âm thanh, phương pháp bao gồm các bước:

tại bộ mã hóa, mã hóa tín hiệu âm thanh và lấy thông tin độ hài hòa và/hoặc thông tin bước ghi;

tại bộ mã hóa, xác định xem thông tin độ hài hòa và/hoặc thông tin bước ghi có phù hợp với ít nhất một chức năng che LTPF và/hoặc lỗi hay không;

truyền từ bộ giải mã đến bộ mã hóa và/hoặc lưu trữ trong bộ nhớ dòng bit chứa sự biểu diễn kỹ thuật số của tín hiệu âm thanh và thông tin liên quan đến độ hài hòa và báo hiệu xem thông tin bước ghi có phù hợp với LTPF và/hoặc che lỗi hay không;

tại bộ giải mã, giải mã sự biểu diễn kỹ thuật số của tín hiệu âm thanh và sử dụng thông tin bước ghi cho LTPF và/hoặc che lỗi theo mẫu báo hiệu của bộ mã hóa.

Trong các ví dụ, bộ mã hóa theo bất kỳ ví dụ nào ở trên hoặc bên dưới, và/hoặc bộ giải mã theo bất kỳ ví dụ nào ở trên hoặc bên dưới, và/hoặc mã hóa theo các ví dụ ở trên hoặc bên dưới và/hoặc giải mã theo các ví dụ ở trên hoặc bên dưới.

Theo các ví dụ, sáng chế đề xuất bộ nhớ không tạm thời lưu trữ các lệnh mà, khi được chạy bởi bộ xử lý, sẽ thực hiện phương pháp như ở trên hoặc bên dưới.

Do đó, bộ mã hóa có thể xác định xem khung tín hiệu có hữu ích cho việc lọc sau dải hạn (LTPF) và/hoặc che việc mất gói tin (PLC) hay không và có thể mã hóa thông tin theo các kết quả xác định. Bộ giải mã có thể áp dụng LTPF và/hoặc PLC theo thông tin thu được từ bộ mã hóa.

Mô tả vắn tắt các hình vẽ

Fig.1 và Fig.2 thể hiện thiết bị mã hóa thông tin tín hiệu âm thanh.

Các hình vẽ từ Fig.3 đến Fig.5 thể hiện các định dạng của thông tin tín hiệu được mã hóa mà có thể được mã hóa bởi thiết bị theo Fig.1 hoặc Fig.2.

Fig.6a và Fig.6b thể hiện các phương pháp mã hóa thông tin tín hiệu âm thanh.

Fig.7 thể hiện thiết bị giải mã thông tin tín hiệu âm thanh.

Fig.8a và Fig.8b thể hiện các định dạng của thông tin tín hiệu âm thanh được mã hóa.

Fig.9 thể hiện thiết bị giải mã thông tin tín hiệu âm thanh.

Fig.10 thể hiện phương pháp giải mã thông tin tín hiệu âm thanh.

Fig.11 và Fig.12 thể hiện các hệ thống mã hóa/giải mã thông tin tín hiệu âm thanh.

Fig.13 thể hiện phương pháp mã hóa/giải mã.

Mô tả chi tiết sáng chế

1. Phía bộ mã hóa

Fig.1 thể hiện thiết bị 10. Thiết bị 10 có thể là để mã hóa các tín hiệu (bộ mã hóa). Ví dụ, thiết bị 10 có thể mã hóa các tín hiệu âm thanh 11 để tạo thông tin tín hiệu âm thanh được mã hóa (ví dụ, thông tin 12, 12', 12'', với thuật ngữ được sử dụng bên dưới).

Thiết bị 10 có thể chứa thành phần (không được thể hiện trên các hình vẽ) để thu được (ví dụ, bằng cách lấy mẫu tín hiệu âm thanh gốc) sự biểu diễn kỹ thuật số của tín hiệu âm thanh, để xử lý nó ở dạng kỹ thuật số. Tín hiệu âm thanh có thể được chia thành các khung (ví dụ, tương ứng với chuỗi các khoảng thời gian) hoặc khung con (có

thể là các phân vùng của các khung). Ví dụ, mỗi khoảng có thể dài 20ms (khung con có thể dài 10ms). Mỗi khung có thể chứa một số lượng mẫu hữu hạn (ví dụ, 1024 hoặc 2048 mẫu cho khung 20ms) trong miền thời gian (time domain - TD). Trong các ví dụ, khung hoặc bản sao hoặc phiên bản đã xử lý của chúng có thể được chuyển đổi (một phần hoặc hoàn toàn) thành sự biểu diễn miền tần số (frequency domain - FD). Thông tin tín hiệu âm thanh được mã hóa có thể là, ví dụ, của dự đoán tuyến tính kích thích mã (code-excited linear prediction - CELP), hoặc loại CELP đại số (algebraic CELP - ACELP), và/hoặc loại TCX. Trong các ví dụ, thiết bị 10 có thể chứa (không được thể hiện trên các hình vẽ) bộ giảm mẫu để giảm số lượng mẫu trên mỗi khung. Trong các ví dụ, thiết bị 10 có thể chứa bộ lấy mẫu lại (có thể thuộc bộ tăng mẫu, bộ lọc thông thấp, và loại bộ tăng mẫu).

Trong các ví dụ, thiết bị 10 có thể cung cấp thông tin tín hiệu âm thanh được mã hóa cho bộ phận truyền thông. Bộ phận truyền thông có thể bao gồm phần cứng (ví dụ, ít nhất là có ăng-ten) để truyền thông với các thiết bị khác (ví dụ, để truyền thông tin tín hiệu âm thanh được mã hóa đến các thiết bị khác). Các bộ phận truyền thông có thể thực hiện các truyền thông theo giao thức cụ thể. Các truyền thông có thể là không dây. Sự truyền theo tiêu chuẩn Bluetooth có thể được thực hiện. Trong các ví dụ, thiết bị 10 có thể bao gồm (hoặc lưu trữ thông tin tín hiệu âm thanh được mã hóa trên) thiết bị lưu trữ.

Thiết bị 10 có thể bao gồm bộ ước lượng bước ghi 13 có thể ước lượng và cung cấp thông tin bước ghi đầu ra 13a cho tín hiệu âm thanh 11 trong khung (ví dụ, trong khoảng thời gian). Thông tin bước ghi 13a có thể bao gồm độ trễ bước ghi hoặc phiên bản được xử lý của nó. Thông tin bước ghi 13a có thể thu được, ví dụ, bằng cách tính sự tự tương quan của tín hiệu âm thanh 11. Thông tin bước ghi 13a có thể được biểu diễn trong trường dữ liệu nhị phân (ở đây được biểu thị bằng "ltpf_pitch_lag"), mà có thể được biểu diễn, ví dụ, với số lượng bit nằm trong khoảng giữa 7 và 11 (ví dụ, 9 bit).

Thiết bị 10 có thể bao gồm bộ phân tích tín hiệu 14 có thể phân tích tín hiệu âm thanh 11 cho khung (ví dụ, trong khoảng thời gian). Bộ phân tích tín hiệu 14 có thể, ví dụ, thu thông tin độ hài hòa 14a liên quan đến tín hiệu âm thanh 11. Thông tin độ hài

hòa có thể bao gồm hoặc dựa trên, ví dụ, ít nhất một trong số hoặc tổ hợp của thông tin độ tương quan (ví dụ, thông tin độ tự tương quan), thông tin độ khuếch đại (ví dụ, thông tin độ khuếch đại bộ lọc sau), thông tin có tính chu kỳ, thông tin dự đoán, v.v. Ít nhất một giá trị trong số các giá trị này ví dụ có thể được chuẩn hóa hoặc xử lý.

Trong các ví dụ, thông tin độ hài hòa 14a có thể bao gồm thông tin mà có thể được mã hóa trong một bit (ở đây được biểu thị với "ltpf_active"). Thông tin độ hài hòa 14a có thể mang thông tin về sự hài hòa của tín hiệu. Thông tin độ hài hòa 14a có thể dựa trên việc tín hiệu đáp ứng một các tiêu chí ("các tiêu chí thứ hai"). Thông tin độ hài hòa 14a có thể phân biệt, ví dụ, giữa việc đáp ứng các tiêu chí thứ hai (có thể liên quan đến tính có chu kỳ cao hơn và/hoặc khả năng dự đoán cao hơn và/hoặc độ ổn định của tín hiệu), và không đáp ứng các tiêu chí thứ hai (có thể liên quan đến độ hài hòa thấp hơn và/hoặc khả năng dự đoán thấp hơn và/hoặc độ không ổn định tín hiệu). Độ hài hòa thấp hơn nói chung liên quan đến tiếng ồn. Ít nhất một trong các dữ liệu trong thông tin độ hài hòa 14a có thể dựa trên việc xác minh các tiêu chí thứ hai và/hoặc xác minh ít nhất một trong (các) điều kiện được thiết lập bởi các tiêu chí thứ hai. Ví dụ, các tiêu chí thứ hai có thể bao gồm việc so sánh ít nhất một số đo liên quan đến độ hài hòa (ví dụ, một hoặc tổ hợp của độ tự tương quan, độ hài hòa, độ khuếch đại, khả năng dự đoán, tính có chu kỳ, v.v., cũng có thể được chuẩn hóa và/hoặc xử lý), hoặc phiên bản được xử lý của chúng, với ít nhất một ngưỡng. Ví dụ, ngưỡng có thể là "ngưỡng thứ hai" (có thể có nhiều hơn một ngưỡng). Trong một số ví dụ, các tiêu chí thứ hai bao gồm việc xác minh các điều kiện trên khung trước đó (ví dụ, khung ngay trước khung hiện thời). Trong một số ví dụ, thông tin độ hài hòa 14a có thể được mã hóa trong một bit. Trong một số ví dụ khác, chuỗi các bit, (ví dụ, một bit cho "ltpf_active" và một số bit khác, ví dụ, để mã hóa thông tin độ khuếch đại hoặc thông tin độ hài hòa khác).

Như được biểu thị bởi bộ chọn 26, thông tin độ hài hòa đầu ra 21a có thể kiểm soát việc mã hóa thực tế của thông tin bước ghi 13a. Ví dụ, trong trường hợp độ hài hòa cực thấp, thông tin bước ghi 13a có thể bị ngăn việc bị mã hóa theo dòng bit.

Như được biểu thị bởi bộ chọn 25, giá trị của thông tin độ hài hòa đầu ra 21a ("ltpf_pitch_lag_present") có thể kiểm soát việc mã hóa thực tế của thông tin độ hài

hòa 14a. Do đó, trong trường hợp phát hiện độ hài hòa cực kỳ thấp (ví dụ, trên cơ sở các tiêu chí khác với các tiêu chí thứ hai), thông tin độ hài hòa 14a có thể được ngăn chặn việc bị mã hóa theo dòng bit.

Thiết bị 10 có thể bao gồm bộ tạo dòng bit 15. Bộ tạo dòng bit 15 có thể cung cấp thông tin tín hiệu âm thanh được mã hóa (được biểu thị bằng 12, 12', hoặc 12'') của tín hiệu âm thanh 11 (ví dụ, trong khoảng thời gian). Cụ thể, bộ tạo dòng bit 15 có thể tạo dòng bit chứa ít nhất là phiên bản kỹ thuật số của tín hiệu âm thanh 11, thông tin bước ghi 13a (ví dụ, "ltpf_pitch_lag"), và thông tin độ hài hòa 14a (ví dụ, "ltpf_active"). Thông tin tín hiệu âm thanh được mã hóa có thể được cung cấp cho bộ giải mã. Thông tin tín hiệu âm thanh được mã hóa có thể là dòng bit mà ví dụ có thể được lưu trữ và/hoặc truyền đến bộ thu (do đó, có thể giải mã thông tin âm thanh được mã hóa bởi thiết bị 10).

Thông tin bước ghi 13a trong thông tin tín hiệu âm thanh được mã hóa có thể được sử dụng, ở phía bộ giải mã, cho bộ lọc sau dài hạn (LTPF). LTPF có thể hoạt động trong TD. Trong các ví dụ, khi thông tin độ hài hòa 14a biểu thị độ hài hòa cao hơn, LTPF sẽ được kích hoạt ở phía bộ giải mã (ví dụ sử dụng thông tin bước ghi 13a). Khi thông tin độ hài hòa 14a biểu thị độ hài hòa (trung gian) thấp hơn (hoặc dù sao thì độ hài hòa không phù hợp với LTPF), LTPF sẽ bị vô hiệu hóa hoặc bị suy giảm ở phía bộ giải mã (ví dụ, không sử dụng thông tin bước ghi 13a, ngay cả khi thông tin bước ghi vẫn còn được mã hóa theo dòng bit). Khi thông tin độ hài hòa 14a bao gồm trường "ltpf_active" (mà có thể được mã hóa trong một bit), ltpf_active=0 có thể có nghĩa là "không sử dụng LTPF tại bộ giải mã", trong khi ltpf_active=1 có thể có nghĩa là "sử dụng LTPF tại bộ giải mã". Ví dụ, ltpf_active=0 có thể liên quan đến độ hài hòa mà thấp hơn độ hài hòa liên quan đến ltpf_active=1, ví dụ, sau khi so sánh số đo độ hài hòa với ngưỡng thứ hai. Mặc dù theo các quy ước trong tài liệu này, ltpf_active=0 đề cập đến độ hài hòa thấp hơn độ hài hòa liên quan đến ltpf_active=1, quy ước khác (ví dụ, dựa trên ý nghĩa khác nhau của các giá trị nhị phân) có thể được cung cấp. Các tiêu chí và/hoặc điều kiện bổ sung hoặc thay thế có thể được sử dụng để xác định giá trị của ltpf_active. Ví dụ, để khẳng định ltpf_active=1, cũng có thể kiểm tra xem tín hiệu có ổn định hay không (ví dụ, bằng cách kiểm tra số đo độ hài hòa liên quan đến khung trước đó).

Ngoài chức năng LTPF, thông tin bước ghi 13a có thể được sử dụng, ví dụ, để thực hiện thao tác che mắt gói tin (PLC) tại bộ giải mã. Trong các ví dụ, bất kể thông tin độ hài hòa 14a (ví dụ, ngay cả khi $ltpf_active=0$), PLC sẽ không được thực hiện. Do đó, trong các ví dụ, trong khi thông tin bước ghi 13a sẽ luôn được sử dụng bởi chức năng PLC của bộ giải mã, thì thông tin bước ghi 13a tương tự sẽ chỉ được sử dụng bởi chức năng LTPF tại bộ giải mã theo điều kiện được thiết lập bởi thông tin độ hài hòa 14a.

Cũng có thể xác minh việc đáp ứng hoặc không đáp ứng "các tiêu chí thứ nhất" (có thể khác với các tiêu chí thứ hai), ví dụ, để xác định xem việc truyền thông tin độ hài hòa 13a có phải là thông tin có giá trị cho bộ giải mã hay không.

Trong các ví dụ, khi bộ phân tích tín hiệu 14 phát hiện ra rằng sự hài hòa (ví dụ, số đo cụ thể của sự hài hòa) không đáp ứng các tiêu chí thứ nhất (các tiêu chí thứ nhất được đáp ứng, ví dụ, về điều kiện của sự hài hòa, và đặc biệt là số đo sự hài hòa cao hơn "ngưỡng thứ nhất" cụ thể), sau đó việc lựa chọn việc mã hóa không có thông tin bước ghi 13a có thể được thực hiện bởi thiết bị 10. Ví dụ, trong trường hợp đó, bộ giải mã sẽ sử dụng dữ liệu trong khung được mã hóa không phải cho chức năng LTPF cũng như cho chức năng PLC (ít nhất, trong một số ví dụ, bộ giải mã sẽ sử dụng chiến lược che không dựa trên thông tin bước ghi, mà sử dụng các kỹ thuật che khác nhau, chẳng hạn như kỹ thuật ước lượng dựa trên bộ giải mã, kỹ thuật che FD, hoặc các kỹ thuật khác).

Ngưỡng thứ nhất và ngưỡng thứ hai được thảo luận ở trên có thể được chọn, trong một số ví dụ, sao cho:

- ngưỡng thứ nhất và/hoặc các tiêu chí thứ nhất phân biệt giữa tín hiệu âm thanh phù hợp với PLC và tín hiệu âm thanh không phù hợp với PLC; và
- ngưỡng thứ hai và/hoặc các tiêu chí thứ hai phân biệt giữa tín hiệu âm thanh phù hợp với LTPF và tín hiệu âm thanh không phù hợp với LTPF.

Trong các ví dụ, ngưỡng thứ nhất và ngưỡng thứ hai có thể được chọn sao cho, giả sử rằng các số đo độ hài hòa được so sánh với ngưỡng thứ nhất và ngưỡng thứ hai có giá trị từ 0 đến 1 (trong đó 0 có nghĩa là: không phải là tín hiệu sóng hài và 1 có

nghĩa là: hoàn toàn là tín hiệu sóng hài), sau đó giá trị của ngưỡng thứ nhất thấp hơn giá trị của ngưỡng thứ hai (ví dụ, độ hài hòa liên quan đến ngưỡng thứ nhất thấp hơn độ hài hòa liên quan đến ngưỡng thứ hai).

Trong số các điều kiện được đặt ra cho các tiêu chí thứ hai, cũng có thể kiểm tra xem sự phát triển theo thời gian của tín hiệu âm thanh 11 có thể sử dụng tín hiệu cho LTPF hay không. Ví dụ, có thể kiểm tra xem, đối với khung trước đó, đã đạt đến ngưỡng tương tự (hoặc giống) chưa. Trong các ví dụ, các kết hợp (hoặc kết hợp có trọng số) của các số đo độ hài hòa (hoặc các phiên bản được xử lý của chúng) có thể được so sánh với một hoặc nhiều ngưỡng. Các số đo độ hài hòa khác nhau (ví dụ, thu được ở các tốc độ lấy mẫu khác nhau) có thể được sử dụng.

Fig.5 thể hiện các ví dụ về các khung 12" (hoặc các phần khung) của thông tin tín hiệu âm thanh được mã hóa có thể được chuẩn bị bởi thiết bị 10. Các khung 12" có thể được phân biệt giữa các khung thứ nhất 16", các khung thứ hai 17", và các khung thứ ba 18". Trong quá trình phát triển theo thời gian của tín hiệu âm thanh 11, các khung thứ nhất 16" có thể được thay thế bằng các khung thứ hai 17" và/hoặc các khung thứ ba, và ngược lại, ví dụ, theo các tính năng (ví dụ độ hài hòa) của tín hiệu âm thanh trong các khoảng thời gian cụ thể (ví dụ, trên cơ sở tín hiệu đáp ứng hoặc không đáp ứng các tiêu chí thứ nhất và/hoặc thứ hai và/hoặc độ hài hòa lớn hơn hoặc nhỏ hơn ngưỡng thứ nhất và/hoặc ngưỡng thứ hai).

Khung thứ nhất 16" có thể là khung liên quan đến sự hài hòa được cho là phù hợp với PLC nhưng không nhất thiết phù hợp với LTPF (các tiêu chí thứ nhất được đáp ứng, các tiêu chí thứ hai không được đáp ứng). Ví dụ, số đo độ hài hòa có thể thấp hơn ngưỡng thứ hai hoặc các điều kiện khác không được đáp ứng (ví dụ, tín hiệu không ổn định giữa khung trước và khung hiện thời). Khung thứ nhất 16" có thể bao gồm sự biểu diễn được mã hóa 16a của tín hiệu âm thanh 11. Khung thứ nhất 16" có thể bao gồm thông tin bước ghi thứ nhất 16b (ví dụ, "ltpf_pitch_lag"). Thông tin bước ghi thứ nhất 16b có thể mã hóa hoặc dựa trên, ví dụ, thông tin bước ghi 13a thu được từ bộ ước lượng bước ghi 13. Khung thứ nhất 16" có thể bao gồm mục dữ liệu điều khiển thứ nhất 16c (ví dụ, "ltpf_active", với giá trị "0" theo quy ước hiện tại), có thể bao gồm hoặc dựa trên, ví dụ, thông tin độ hài hòa 14a thu được bởi bộ phân tích tín

hiệu 14. Khung thứ nhất 16" này có thể chứa (trong trường 16a) đủ thông tin để giải mã, ở phía bộ giải mã, tín hiệu âm thanh, và hơn nữa, để sử dụng thông tin bước ghi 13a (được mã hóa trong 16b) cho PLC, trong trường hợp cần thiết. Trong các ví dụ, bộ giải mã sẽ không sử dụng thông tin bước ghi 13a cho LTPF, do độ hài hòa không đáp ứng các tiêu chí thứ hai (ví dụ, số đo độ hài hòa thấp của tín hiệu và/hoặc tín hiệu không ổn định giữa hai khung liên tiếp).

Khung thứ hai 17" có thể là khung liên quan đến độ hài hòa được giữ lại đủ cho LTPF (ví dụ, nó đáp ứng các tiêu chí thứ hai, ví dụ độ hài hòa theo số đo cao hơn ngưỡng thứ hai và/hoặc khung trước đó cũng lớn hơn ít nhất một ngưỡng cụ thể). Khung thứ hai 17" có thể bao gồm sự biểu diễn được mã hóa 17a của tín hiệu âm thanh 11. Khung thứ hai 17" có thể bao gồm thông tin bước ghi thứ hai 17b (ví dụ "ltpf_pitch_lag"). Thông tin bước ghi thứ hai 17b có thể mã hóa hoặc dựa trên, ví dụ, thông tin bước ghi 13a thu được từ bộ ước lượng bước ghi 13. Khung thứ hai 17" có thể bao gồm mục dữ liệu điều khiển thứ hai 17c (ví dụ, "ltpf_active", với giá trị "1" theo quy ước hiện tại), mà có thể bao gồm hoặc dựa trên, ví dụ, thông tin độ hài hòa 14a thu được bởi bộ phân tích tín hiệu 14. Khung thứ hai 17" này có thể chứa đủ thông tin sao cho ở phía bộ giải mã, tín hiệu âm thanh 11 được giải mã, và hơn nữa, thông tin bước ghi 17b (từ đầu ra 13a của bộ ước lượng bước ghi) có thể được sử dụng cho PLC trong trường hợp cần thiết. Hơn nữa, bộ giải mã sẽ sử dụng thông tin bước ghi 17b (13a) cho LTPF, nhờ việc đáp ứng các tiêu chí thứ hai, cụ thể là dựa trên độ hài hòa cao của tín hiệu (như được biểu thị bởi ltpf_active=1 theo quy ước hiện tại).

Trong các ví dụ, các khung thứ nhất 16" và các khung thứ hai 17" được xác định bằng giá trị của các mục dữ liệu điều khiển 16c và 17c (ví dụ, bằng giá trị nhị phân của "ltpf_active").

Trong các ví dụ, khi được mã hóa trong dòng bit, khung thứ nhất và thứ hai có mặt, cho thông tin bước ghi thứ nhất và thứ hai (16b, 17b) và cho các mục dữ liệu điều khiển thứ nhất và thứ hai (16c, 17c), định dạng sao cho:

- một bit đơn được dành riêng để mã hóa các mục dữ liệu điều khiển thứ nhất 16c và thứ hai 17c; và

- trường dữ liệu cố định được dành riêng cho từng thông tin bước ghi thứ nhất 16b và thứ hai 17b.

Theo đó, một mục dữ liệu thứ nhất 16c đơn có thể được phân biệt với một mục dữ liệu thứ hai 17c đơn bằng giá trị của bit trong một phần cụ thể (ví dụ cố định) trong khung. Ngoài ra, thông tin bước ghi thứ nhất và thứ hai cũng có thể được chèn vào một số bit cố định ở vị trí dành riêng (ví dụ, vị trí cố định).

Trong các ví dụ (ví dụ, như được thể hiện trên Fig.4 và/hoặc Fig.5), thông tin độ hài hòa 14a không chỉ đơn giản là phân biệt giữa việc đáp ứng và không đáp ứng các tiêu chí thứ hai, ví dụ, không chỉ đơn giản là phân biệt giữa độ hài hòa cao hơn và độ hài hòa thấp hơn. Trong một số trường hợp, thông tin hài hòa có thể bao gồm thông tin độ hài hòa bổ sung chẳng hạn như thông tin độ khuếch đại (ví dụ, độ khuếch đại bộ lọc sau) và/hoặc thông tin độ tương quan (tự tương quan, tương quan chuẩn hóa), và/hoặc phiên bản được xử lý của chúng. Trong một số trường hợp, việc tham chiếu ở đây đã tạo ra độ khuếch đại hoặc thông tin độ hài hòa khác có thể được mã hóa trong 1 đến 4 bit (ví dụ, 2 bit) và có thể đề cập đến độ khuếch đại bộ lọc sau như thu được bởi bộ phân tích tín hiệu 14.

Trong các ví dụ trong đó thông tin độ hài hòa bổ sung được mã hóa, bộ giải mã, bằng việc thừa nhận `ltpf_active=1` (ví dụ, khung thứ hai 17' hoặc 17''), có thể hiểu rằng trường tiếp theo của khung thứ hai 17' hoặc 17'' mã hóa thông tin độ hài hòa bổ sung 17d. Ngược lại, bằng cách xác định `ltpf_active=0` (ví dụ, khung thứ nhất 16' hoặc 16''), bộ giải mã có thể hiểu rằng không có trường thông tin độ hài hòa bổ sung 17d nào được mã hóa trong khung 17' hoặc 17''.

Trong các ví dụ (ví dụ Fig.5), khung thứ ba 18'' có thể được mã hóa trong dòng bit. Khung thứ ba 18'' có thể được xác định để có định dạng thiếu thông tin bước ghi và thông tin độ hài hòa. Cấu trúc dữ liệu của nó không cung cấp bit nào để mã hóa dữ liệu 16b, 16c, 17b, 17c. Tuy nhiên, khung thứ ba 18'' vẫn có thể bao gồm sự biểu diễn được mã hóa 18a của tín hiệu âm thanh và/hoặc dữ liệu điều khiển khác hữu ích cho bộ mã hóa.

Trong các ví dụ, khung thứ ba 18'' được phân biệt với khung thứ nhất và khung thứ hai bởi dữ liệu điều khiển thứ ba 18e ("`ltpf_pitch_lag_present`"), có thể có giá trị

trong khung thứ ba khác với giá trị ở khung thứ nhất 16" và khung thứ hai 17". Ví dụ, mục dữ liệu điều khiển thứ ba 18e có thể là "0" để xác định khung thứ ba 18" và "1" để xác định khung thứ nhất 16" và khung thứ hai 17".

Trong các ví dụ, khung thứ ba 18" có thể được mã hóa khi tín hiệu thông tin sẽ không hữu ích cho LTPF và cho PLC (ví dụ, do tính hài hòa rất thấp, ví dụ, khi nhiều đang chiếm ưu thế). Do đó, mục dữ liệu điều khiển 18e ("ltpf_pitch_lag_present") có thể là "0" để báo hiệu cho bộ giải mã rằng sẽ không có thông tin có giá trị nào trong độ trễ bước ghi và do đó, không có ý nghĩa gì khi mã hóa nó. Đây có thể là kết quả của quá trình xác minh dựa trên các tiêu chí thứ nhất.

Theo quy ước hiện tại, khi mục dữ liệu điều khiển thứ ba 18e là "0", các số đo độ hài hòa có thể thấp hơn ngưỡng thứ nhất liên quan đến độ hài hòa thấp (đây có thể là một kỹ thuật để xác minh việc đáp ứng các tiêu chí thứ nhất).

Fig.3 và Fig.4 thể hiện các ví dụ về khung thứ nhất 16, 16' và khung thứ hai 17, 17' mà mục điều khiển thứ ba 18e không được cung cấp (khung thứ hai 17' mã hóa thông tin độ hài hòa bổ sung, mà có thể là tùy chọn trong một số ví dụ). Trong một số ví dụ, những khung này không được sử dụng. Tuy nhiên, đáng chú ý là trong một số ví dụ, ngoài việc không có mục điều khiển thứ ba 18e, các khung 16, 16', 17, 17' có các trường giống nhau của các khung 16" và 17" của Fig.5.

Fig.2 thể hiện ví dụ về thiết bị 10' mà có thể là một triển khai cụ thể của thiết bị 10. Các thuộc tính của thiết bị 10 (các tính năng của tín hiệu, mã, các tính năng truyền/lưu trữ, triển khai Bluetooth, v.v.) do đó không được lặp lại ở đây. Thiết bị 10' có thể chuẩn bị thông tin tín hiệu âm thanh được mã hóa (ví dụ các khung 12, 12', 12") của tín hiệu âm thanh 11. Thiết bị 10' có thể bao gồm bộ ước lượng bước ghi 13, bộ phân tích tín hiệu 14, và bộ tạo dòng bit 15, có thể giống như (hoặc rất giống với) thiết bị 10. Thiết bị 10' cũng có thể bao gồm các thành phần để lấy mẫu, lấy mẫu lại, và lọc như thiết bị 10.

Bộ ước lượng bước ghi 13 có thể xuất ra thông tin bước ghi 13a (ví dụ, độ trễ bước ghi, chẳng hạn như "ltpf_pitch_lag").

Bộ phân tích tín hiệu 14 có thể xuất ra thông tin độ hài hòa 24c (14a) mà trong một số ví dụ có thể được tạo bởi nhiều giá trị (ví dụ, vector bao gồm nhiều giá trị). Bộ phân tích tín hiệu 14 có thể bao gồm bộ đo độ hài hòa 24 có thể tạo ra các số đo độ hài hòa 24a. Các phép đo độ hài hòa 24a có thể bao gồm thông tin tương quan/tự tương quan chuẩn hóa hoặc không được chuẩn hóa, thông tin độ khuếch đại (ví dụ, độ khuếch đại bộ lọc sau), thông tin về tính có chu kỳ, thông tin dự đoán, thông tin liên quan đến độ ổn định và/hoặc sự phát triển của tín hiệu, phiên bản được xử lý của chúng, v.v. Ký hiệu tham chiếu 24a có thể đề cập đến nhiều giá trị, mà ít nhất một số (hoặc tất cả) trong số đó, tuy nhiên, có thể giống nhau hoặc có thể khác nhau, và/hoặc các phiên bản được xử lý của cùng một giá trị và/hoặc thu được ở các tốc độ lấy mẫu khác nhau.

Trong các ví dụ, các số đo độ hài hòa 24a có thể bao gồm số đo độ hài hòa thứ nhất 24a' (có thể được đo ở tốc độ lấy mẫu thứ nhất, ví dụ, 6,4KHz) và số đo độ hài hòa thứ hai 24a (có thể được đo ở tốc độ lấy mẫu thứ hai, ví dụ, 12,8KHz). Trong các ví dụ khác, số đo tương tự có thể được sử dụng.

Ở khối 21 xác minh xem các số đo độ hài hòa 24a (ví dụ, số đo độ hài hòa thứ nhất 24a') đáp ứng các tiêu chí thứ nhất, ví dụ, chúng vượt quá ngưỡng thứ nhất mà có thể được lưu trữ trong phần tử bộ nhớ 23.

Ví dụ, ít nhất một số đo độ hài hòa 24a (ví dụ, số đo độ hài hòa thứ nhất 24a') có thể được so sánh với ngưỡng thứ nhất. Ngưỡng thứ nhất có thể được lưu trữ, ví dụ, trong phần tử bộ nhớ 23 (ví dụ, phần tử bộ nhớ không tạm thời). Khối 21 (có thể được xem là sự so sánh của số đo độ hài hòa thứ nhất 24a' với ngưỡng thứ nhất) có thể xuất ra thông tin độ hài hòa 21a cho biết liệu độ hài hòa của tín hiệu âm thanh 11 có vượt quá ngưỡng thứ nhất hay không (và cụ thể, liệu số đo độ hài hòa thứ nhất 24a' có vượt quá ngưỡng thứ nhất hay không).

Trong các ví dụ, `ltpf_pitch_present` có thể là, ví dụ,

$$\text{ltpf_pitch_present} = \begin{cases} 1 & \text{nếu } \text{normcorr}(x_{6.4}, N_{6.4}, T_{6.4}) > \text{first_threshold} \\ 0 & \text{ngược lại} \end{cases}$$

trong đó $x_{6.4}$ là tín hiệu âm thanh ở tốc độ lấy mẫu 6,4kHz, $N_{6.4}$ là độ dài của khung hiện thời và $T_{6.4}$ là độ trễ thu được bởi bộ ước lượng bước ghi cho khung hiện

thời và $\text{normcorr}(x, L, T)$ là mối tương quan chuẩn hóa của tín hiệu x của độ dài L ở độ trễ T

$$\text{normcorr}(x, L, T) = \frac{\sum_{n=0}^{L-1} x(n)x(n-T)}{\sqrt{\sum_{n=0}^{L-1} x^2(n) \sum_{n=0}^{L-1} x^2(n-T)}}$$

Trong một số ví dụ, các tốc độ lấy mẫu khác hoặc các mối tương quan khác có thể được sử dụng. Trong các ví dụ, ngưỡng thứ nhất có thể là 0,6. Trên thực tế, cần lưu ý rằng đối với các số đo độ hài hòa trên 0,6, PLC có thể được thực hiện một cách đáng tin cậy. Tuy nhiên, không phải lúc nào cũng đảm bảo rằng, ngay cả đối với các giá trị nhỏ hơn 0,6, LTPF có thể được thực hiện một cách đáng tin cậy.

Do đó, đầu ra 21a từ khối 21 có thể là giá trị nhị phân (ví dụ, "ltpf_pitch_lag_present") có thể là "1" nếu độ hài hòa vượt quá ngưỡng thứ nhất (ví dụ, nếu số đo độ hài hòa thứ nhất 24a' vượt quá ngưỡng thứ nhất), và có thể là "0" nếu độ hài hòa dưới ngưỡng thứ nhất. Thông tin độ hài hòa 21a (ví dụ, "ltpf_pitch_lag_present") có thể kiểm soát việc mã hóa thực tế của đầu ra 13a: nếu (ví dụ, với số đo thứ nhất 24a' như được thể hiện ở trên) độ hài hòa nằm dưới ngưỡng thứ nhất (ltpf_pitch_lag_present=0) hoặc các tiêu chí thứ nhất không được đáp ứng, không có thông tin bước ghi 13a nào được mã hóa; nếu độ hài hòa vượt quá ngưỡng thứ nhất (ltpf_pitch_lag_present=1) hoặc các tiêu chí thứ nhất được đáp ứng, thông tin bước ghi thực sự được mã hóa. Đầu ra 21a ("ltpf_pitch_lag_present") có thể được mã hóa. Do đó, đầu ra 21a có thể được mã hóa thành mục điều khiển thứ ba 18e (ví dụ, để mã hóa khung thứ ba 18" khi đầu ra 21a là "0", và khung thứ hai hoặc khung thứ ba khi đầu ra 21a là "1").

Bộ đo độ hài hòa 24 có thể tùy ý xuất ra số đo độ hài hòa 24b mà có thể, ví dụ, là thông tin độ khuếch đại (ví dụ, "ltpf_gain") có thể được mã hóa trong thông tin tín hiệu âm thanh được mã hóa 12, 12', 12" bởi bộ tạo dòng bit 15. Các tham số khác có thể được cung cấp. Thông tin độ hài hòa khác 24b có thể được sử dụng, trong một số ví dụ, cho LTPF ở phía bộ giải mã.

Như được biểu thị bởi khối 22, việc xác minh sự đáp ứng các tiêu chí thứ hai có thể được thực hiện trên cơ sở ít nhất một số đo độ hài hòa 24a (ví dụ, số đo độ hài hòa thứ hai 24a").

Một điều kiện dựa trên các tiêu chí thứ hai có thể là so sánh ít nhất một số đo độ hài hòa 24a (ví dụ, số đo độ hài hòa thứ hai 24a") với ngưỡng thứ hai. Ngưỡng thứ hai có thể được lưu trữ, ví dụ, trong phần tử bộ nhớ 23 (ví dụ, ở vị trí bộ nhớ khác với việc lưu trữ ngưỡng thứ nhất).

Các tiêu chí thứ hai cũng có thể dựa trên các điều kiện khác (ví dụ, về việc đáp ứng đồng thời hai điều kiện khác nhau). Một điều kiện bổ sung có thể, ví dụ, dựa trên khung trước đó. Ví dụ, có thể so sánh ít nhất một số đo độ hài hòa 24a (ví dụ, số đo độ hài hòa thứ hai 24a") với ngưỡng.

Theo đó, khối 22 có thể xuất ra thông tin độ hài hòa 22a mà có thể dựa trên ít nhất một điều kiện hoặc dựa trên nhiều điều kiện (ví dụ, một điều kiện trên khung hiện thời và một điều kiện trên khung trước).

Khối 22 có thể xuất ra (ví dụ, là kết quả của quá trình xác minh các tiêu chí thứ hai) thông tin độ hài hòa 22a cho biết liệu độ hài hòa của tín hiệu âm thanh 11 (đối với khung hiện thời và/hoặc đối với khung trước) có vượt quá ngưỡng thứ hai hay không (và, ví dụ, liệu số đo độ hài hòa thứ hai 24a" có vượt quá ngưỡng thứ hai hay không). Thông tin độ hài hòa 22a có thể là giá trị nhị phân (ví dụ, "ltpf_active") có thể là "1" nếu độ hài hòa vượt quá ngưỡng thứ hai (ví dụ, số đo độ hài hòa thứ hai 24a" vượt quá ngưỡng thứ hai), và có thể là "0" nếu độ hài hòa (của khung hiện thời và/hoặc khung trước) nằm dưới ngưỡng thứ hai (ví dụ, số đo độ hài hòa thứ hai 24a" nằm dưới ngưỡng thứ hai).

Thông tin độ hài hòa 22a (ví dụ, "ltpf_active") có thể kiểm soát (nếu được cung cấp) việc mã hóa thực tế của giá trị 24b (trong các ví dụ trong đó giá trị 24b thực sự được cung cấp): nếu độ hài hòa thứ hai (ví dụ, số đo độ hài hòa thứ hai 24a") không đáp ứng các tiêu chí thứ hai (ví dụ, nếu độ hài hòa nằm dưới ngưỡng thứ hai và ltpf_active=0), không có thêm thông tin độ hài hòa 24b nào (ví dụ, không có thông tin độ hài hòa bổ sung nào) được mã hóa; nếu độ hài hòa (ví dụ, số đo độ hài hòa thứ hai

24a") đáp ứng các tiêu chí thứ hai (ví dụ, nó vượt quá ngưỡng thứ hai và $ltpf_active=1$), thông tin độ hài hòa bổ sung 24b thực sự được mã hóa.

Đáng chú ý, các tiêu chí thứ hai có thể dựa trên các điều kiện khác nhau và/hoặc bổ sung. Ví dụ, có thể xác minh xem tín hiệu có ổn định theo thời gian không (ví dụ, nếu sự tương quan chuẩn hóa có tính chất tương tự trong hai khung liên tiếp).

(Các) ngưỡng thứ hai có thể được xác định để được liên kết với nội dung hài vượt quá nội dung hài liên quan đến ngưỡng thứ nhất. Trong các ví dụ, ngưỡng thứ nhất và ngưỡng thứ hai có thể được chọn sao cho, giả sử rằng các số đo độ hài hòa được so sánh với ngưỡng thứ nhất và ngưỡng thứ hai có giá trị từ 0 đến 1 (trong đó 0 có nghĩa là: không phải là tín hiệu sóng hài và 1 có nghĩa là: hoàn toàn là tín hiệu sóng hài), sau đó giá trị của ngưỡng thứ nhất thấp hơn giá trị của ngưỡng thứ hai (ví dụ, độ hài hòa liên quan đến ngưỡng thứ nhất thấp hơn độ hài hòa liên quan đến ngưỡng thứ hai).

Giá trị 22a (ví dụ, " $ltpf_active$ ") có thể được mã hóa, ví dụ, để trở thành mục dữ liệu điều khiển thứ nhất 16c hoặc thứ hai 17c (Fig.4). Việc mã hóa thực tế của giá trị 22a có thể được kiểm soát bởi giá trị 21a (ví dụ, sử dụng bộ chọn 25): ví dụ, " $ltpf_active$ " chỉ có thể được mã hóa nếu $ltpf_pitch_lag_present=1$, trong khi " $ltpf_active$ " không được cung cấp cho bộ tạo dòng bit 15 khi $ltpf_pitch_lag_present=0$ (để mã hóa khung thứ ba 18"). Trong trường hợp đó, không cần thiết phải cung cấp thông tin bước ghi cho bộ giải mã: độ hài hòa có thể quá thấp, mà bộ giải mã sẽ không sử dụng thông tin bước ghi cho cả PLC và LTPF. Ngoài ra, thông tin độ hài hòa, chẳng hạn như " $ltpf_active$ " có thể vô ích trong trường hợp đó: vì không có thông tin bước ghi nào được cung cấp cho bộ giải mã, không có khả năng bộ giải mã sẽ cố gắng thực hiện LTPF.

Ví dụ để thu được giá trị $ltpf_active$ (16c, 17c, 22a) được đưa ra ở đây. Các chiến lược thay thế khác có thể được thực hiện.

Mối tương quan chuẩn hóa có thể được tính trước tiên như sau

$$nc = \frac{\sum_{n=0}^{127} x_i(n,0)x_i(n - pitch_int, pitch_fr)}{\sqrt{\sum_{n=0}^{127} x_i^2(n,0) \sum_{n=0}^{127} x_i^2(n - pitch_int, pitch_fr)}}$$

với pitch_int là phần nguyên của độ trễ bước ghi, pitch_fr là phần thập phân của độ trễ bước ghi, và

$$x_i(n, d) = \sum_{k=-2}^2 x_{12.8}(n+k) h_i(4k-d)$$

với $x_{12.8}$ là tín hiệu đầu vào được lấy mẫu lại ở 12,8kHz (ví dụ) và h_i là đáp ứng xung của bộ lọc thông thấp FIR được cho bởi

$$h_i(n) = \begin{cases} \text{tab_ltpf_interp_x12k8}(n+7) & , \text{if } -8 < n < 8 \\ 0 & , \text{otherwise} \end{cases}$$

với $\text{tab_ltpf_interp_x12k8}$ được chọn, ví dụ, từ các giá trị sau:

```
double tab_ltpf_interp_x12k8[15] = {
+6.698858366939680e-03, +3.967114782344967e-02, +1.069991860896389e-01
+2.098804630681809e-01, +3.356906254147840e-01, +4.592209296082350e-01
+5.500750019177116e-01, +5.835275754221211e-01, +5.500750019177116e-01
+4.592209296082350e-01, +3.356906254147840e-01, +2.098804630681809e-01
+1.069991860896389e-01, +3.967114782344967e-02, +6.698858366939680e-03};
```

Bit kích hoạt LTPF ("ltpf_active") sau đó có thể thu được theo quy trình sau:

```
if (
(mem_ltpf_active==0 && mem_nc>0.94 && nc>0.94) ||
(mem_ltpf_active==1 && nc>0.9) ||
(mem_ltpf_active==1 && abs(pit-mem_pit)<2 && (nc-mem_nc)>-0.1 &&
nc>0.84)
)
{
ltpf_active = 1;
}
else
{
ltpf_active = 0;
}
```

trong đó mem_ltpf_active là giá trị của ltpf_active trong khung trước (nó là 0 nếu $\text{ltpf_pitch_present}=0$ trong khung trước), mem_nc là giá trị của nc trong khung trước (nó là 0 nếu $\text{ltpf_pitch_present}=0$ trong khung trước), $\text{pit}=\text{pitch_int}+\text{pitch_fr}/4$ và mem_pit là giá trị của pit trong khung trước (nó là 0 nếu $\text{ltpf_pitch_present}=0$ trong khung trước). Quy trình này được thể hiện, ví dụ, trên Fig.6b (xem thêm bên dưới).

Điều quan trọng cần lưu ý là việc sơ đồ hóa của Fig.2 hoàn toàn thuộc về cách trình bày. Thay vì các khối 21, 22 và các bộ chọn, các bộ phận phần cứng và/hoặc phần mềm khác nhau có thể được sử dụng. Trong các ví dụ, ít nhất hai trong số các thành phần chẳng hạn như các khối 21 và 22, bộ ước lượng bước ghi, bộ phân tích tín hiệu và/hoặc bộ đo độ hài hòa và/hoặc bộ tạo dòng bit có thể được triển khai một thành phần duy nhất.

Trên cơ sở các phép đo được thực hiện, có thể phân biệt giữa:

- trạng thái thứ ba, trong đó:

- các tiêu chí thứ nhất không được đáp ứng;
- cả hai đầu ra 21a và 22a của khối 21 và khối 22 đều là "0";
- các đầu ra 13a (ví dụ, "ltpf_pitch_lag"), 24b (ví dụ, thông tin độ hài hòa bổ sung, tùy chọn), và 22a (ví dụ, "ltpf_active") không được mã hóa;
- chỉ có giá trị "0" (ví dụ, "ltpf_pitch_lag_present") của đầu ra 21a được mã hóa;
- khung thứ ba 18" được mã hóa với mục điều khiển thứ ba "0" (ví dụ, từ "ltpf_pitch_lag_present") và sự biểu diễn tín hiệu của tín hiệu âm thanh, nhưng không có bất kỳ bit nào mã hóa thông tin bước ghi và/hoặc mục điều khiển thứ nhất và thứ hai;
- theo đó, bộ giải mã sẽ hiểu rằng không có thông tin bước ghi và thông tin độ hài hòa nào có thể được sử dụng cho LTPF và PLC (ví dụ, do độ hài hòa cực thấp);

- trạng thái thứ nhất, trong đó:

- các tiêu chí thứ nhất được đáp ứng và các tiêu chí thứ hai không được đáp ứng;
- đầu ra 21a của khối 21 là "1" (ví dụ, do đáp ứng các tiêu chí thứ nhất, ví dụ, do số đo thứ nhất 24a' lớn hơn ngưỡng thứ nhất), trong khi đầu ra 22a của khối 22 là "0" (ví dụ, do không đáp ứng các tiêu chí thứ hai, ví dụ, do số đo thứ hai 24a", đối với khung hiện thời hoặc khung trước, nằm dưới ngưỡng thứ hai);

- giá trị "1" của đầu ra 21a (ví dụ, "ltpf_pitch_lag_present") được mã hóa trong 18e;
- đầu ra 13a (ví dụ, "ltpf_pitch_lag") được mã hóa trong 16b;
- giá trị "0" của đầu ra 22a (ví dụ, "ltpf_active") được mã hóa trong 16c;
- đầu ra tùy chọn 24b (ví dụ, thông tin độ hài hòa bổ sung) không được mã hóa;
- khung thứ nhất 16" được mã hóa với mục dữ liệu điều khiển thứ ba bằng "1" (ví dụ, từ "ltpf_pitch_lag_present" 18e), với một bit đơn mã hóa mục dữ liệu điều khiển thứ nhất bằng "0" (ví dụ, từ "ltpf_active" 16c), và một lượng bit cố định (ví dụ, ở một vị trí cố định) để mã hóa thông tin bước ghi thứ nhất 16b (ví dụ, được lấy từ "ltpf_pitch_lag");
- theo đó, bộ giải mã sẽ hiểu rằng sẽ sử dụng thông tin bước ghi 13a (ví dụ, độ trễ bước ghi được mã hóa trong 16b) chỉ cho PLC, mà sẽ không sử dụng thông tin bước ghi hoặc thông tin độ hài hòa cho LTPF;
- trạng thái thứ hai, trong đó:
 - tiêu chuẩn thứ nhất và tiêu chuẩn thứ hai được đáp ứng;
 - cả hai đầu ra 21a và 22a của khối 21 và khối 22 là "1" (ví dụ, do đáp ứng các tiêu chí thứ nhất, ví dụ, do số đo thứ nhất 24a' lớn hơn ngưỡng thứ hai và số đo thứ hai 24a" đáp ứng các tiêu chí thứ hai, ví dụ, số đo thứ hai 24a" lớn hơn ngưỡng thứ hai trong khung hiện thời hoặc trong khung trước);
 - giá trị "1" của đầu ra 21a (ví dụ, "ltpf_pitch_lag_present") được mã hóa;
 - đầu ra 13a (ví dụ, "ltpf_pitch_lag") được mã hóa;
 - giá trị "1" của đầu ra 22a (ví dụ, "ltpf_active") được mã hóa;
 - khung thứ hai 17" được mã hóa với mục dữ liệu điều khiển thứ ba bằng "1" (ví dụ, từ "ltpf_pitch_lag_present" trong 18e), với một bit đơn mã hóa mục dữ liệu điều khiển thứ hai bằng "1" (ví dụ, từ "ltpf_active" trong 17c), và một lượng bit cố định (ví dụ, ở một vị trí cố định) để mã hóa thông tin bước ghi thứ hai (ví dụ, được lấy

từ "ltpf_pitch_lag") trong 17b, và một cách tùy chọn, thông tin bổ sung (chẳng hạn như thông tin độ hài hòa bổ sung) trong 17d;

- theo đó, bộ giải mã sẽ sử dụng thông tin bước ghi 13a (ví dụ, độ trễ bước ghi) cho PLC, và cũng sẽ sử dụng thông tin bước ghi và (trong trường hợp) thông tin độ hài hòa bổ sung cho LTPF (ví dụ, giả sử rằng độ hài hòa là đủ cho cả LTPF và PLC).

Do đó, tham chiếu đến Fig.5, các khung 12" được thể hiện có thể được cung cấp bởi bộ tạo dòng bit 15, ví dụ, trong thiết bị 10'. Đặc biệt có thể được mã hóa:

- trong trường hợp trạng thái thứ ba, khung thứ ba 18" với các trường:

- mục dữ liệu điều khiển thứ ba 18e (ví dụ, "ltpf_pitch_lag_present", thu được từ 21a) với giá trị "0"; và

- sự biểu diễn được mã hóa 18a của tín hiệu âm thanh 11;

- trong trường hợp trạng thái thứ nhất, khung thứ nhất 16" với các trường:

- mục dữ liệu điều khiển thứ ba 18e (ví dụ, "ltpf_pitch_lag_present", thu được từ 21a) với giá trị "1";

- sự biểu diễn được mã hóa 16a của tín hiệu âm thanh 11;

- thông tin bước ghi thứ nhất 16b (ví dụ, "ltpf_pitch_lag", thu được từ 13a) trong trường dữ liệu cố định của khung thứ nhất 16"; và

- mục dữ liệu điều khiển thứ nhất 16c (ví dụ, "ltpf_active", thu được từ 22a) với giá trị "0"; và

- trong trường hợp trạng thái thứ hai, khung thứ hai "17" với các trường:

- mục dữ liệu điều khiển thứ ba 18e (ví dụ, "ltpf_pitch_lag_present", thu được từ 21a) với giá trị "1";

- sự biểu diễn được mã hóa 17a của tín hiệu âm thanh 11;

- thông tin bước ghi thứ hai 17b (ví dụ, "ltpf_pitch_lag", thu được từ 13a) khung thứ hai 17";

- mục dữ liệu điều khiển thứ hai 17c (ví dụ, "ltpf_active", thu được từ 22a) với giá trị "1"; và
- trong đó được cung cấp, thông tin độ hài hòa 17d (tùy chọn) (ví dụ, thu được từ 24b).

Trong các ví dụ, khung thứ ba 18" không hiển thị trường dữ liệu cố định cho thông tin bước ghi thứ nhất hoặc thứ hai và không hiển thị bất kỳ bit nào mã hóa mục dữ liệu điều khiển thứ nhất và mục dữ liệu điều khiển thứ hai.

Từ mục dữ liệu điều khiển thứ ba 18e và các mục dữ liệu điều khiển thứ nhất và thứ hai 16c và 17c, bộ giải mã sẽ hiểu liệu:

- bộ giải mã sẽ không triển khai LTPF và PLC với thông tin bước ghi và thông tin độ hài hòa trong trường hợp trạng thái thứ ba,
- bộ giải mã sẽ không triển khai LTPF mà sẽ triển khai PLC với thông tin bước ghi chỉ trong trường hợp trạng thái thứ nhất, và
- bộ giải mã sẽ thực hiện cả LTPF sử dụng cả thông tin bước ghi và PLC sử dụng thông tin bước ghi trong trường hợp trạng thái thứ hai.

Như có thể thấy trên Fig.5, trong một số ví dụ:

- khung thứ ba 18 có thể có định dạng thiếu thông tin bước ghi thứ nhất 16b, mục dữ liệu điều khiển thứ nhất 16c, thông tin bước ghi thứ hai 17b, và mục dữ liệu điều khiển thứ hai 17c;
- mục dữ liệu điều khiển thứ ba 18e có thể được mã hóa trong một bit đơn có giá trị phân biệt khung thứ ba 18" với khung thứ nhất 16" và khung thứ hai 17"; và/hoặc
- trong thông tin tín hiệu âm thanh được mã hóa, đối với khung thứ nhất 16", một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ nhất 16c và trường dữ liệu cố định 16b được dành riêng cho thông tin bước ghi thứ nhất; và/hoặc
- trong thông tin tín hiệu âm thanh được mã hóa, đối với khung thứ hai 17", một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ hai 17c và trường dữ liệu cố định 17b được dành riêng cho thông tin bước ghi thứ hai; và/hoặc

- mục dữ liệu điều khiển thứ nhất 16c và mục dữ liệu điều khiển thứ hai 17c được mã hóa trong cùng một phần hoặc trường dữ liệu trong thông tin tín hiệu âm thanh được mã hóa; và/hoặc

- thông tin tín hiệu âm thanh được mã hóa có thể bao gồm một bit báo hiệu thứ nhất mã hóa mục dữ liệu điều khiển thứ ba 18e; và/hoặc trong trường hợp giá trị của mục dữ liệu điều khiển thứ ba biểu thị sự có mặt của thông tin bước ghi thứ nhất và/hoặc thông tin bước ghi thứ hai, bit báo hiệu thứ hai mã hóa mục dữ liệu điều khiển thứ nhất và mục dữ liệu điều khiển thứ hai.

Fig.6a thể hiện phương pháp 60 theo các ví dụ. Phương pháp này có thể được thực hiện, ví dụ, sử dụng thiết bị 10 hoặc 10'. Ví dụ, phương pháp có thể mã hóa các khung 16", 17", 18" như giải thích ở trên.

Phương pháp 60 có thể bao gồm bước S60 thu được (tại một khoảng thời gian cụ thể) (các) số đo độ hài hòa (ví dụ 24a) từ tín hiệu âm thanh 11, ví dụ, sử dụng bộ phân tích tín hiệu 14, và cụ thể là bộ đo độ hài hòa 24. Các số đo độ hài hòa (thông tin độ hài hòa) có thể bao gồm hoặc dựa trên, ví dụ, ít nhất một hoặc tổ hợp của thông tin độ tương quan (ví dụ, thông tin độ tự tương quan), thông tin độ khuếch đại (ví dụ, thông tin độ khuếch đại bộ lọc sau), thông tin tính có chu kỳ, thông tin dự đoán, được áp dụng cho tín hiệu âm thanh 11 (ví dụ, cho khoảng thời gian). Trong các ví dụ, có thể thu được số đo độ hài hòa thứ nhất 24a' (ví dụ, ở 6,4KHz) và có thể thu được số đo độ hài hòa thứ hai 24a" (ví dụ, ở 12,8KHz). Trong các ví dụ khác nhau, các số đo độ hài hòa tương tự có thể được sử dụng.

Phương pháp có thể bao gồm việc xác minh việc đáp ứng các tiêu chí thứ nhất, ví dụ, sử dụng khối 21. Ví dụ, có thể thực hiện việc so sánh (các) số đo độ hài hòa với ngưỡng thứ nhất. Nếu ở S61 các tiêu chí thứ nhất không được đáp ứng (ví dụ, độ hài hòa nằm dưới ngưỡng thứ nhất, ví dụ, khi số đo thứ nhất 24a' nằm dưới ngưỡng thứ nhất), thì tại S62, khung thứ ba 18" có thể được mã hóa, khung thứ ba 18" biểu thị giá trị "0" trong mục dữ liệu điều khiển thứ ba 18e (ví dụ, "ltpf_pitch_lag_present"), ví dụ, không dành bất kỳ bit nào để mã hóa các giá trị chẳng hạn như thông tin bước ghi và thông tin độ hài hòa bổ sung. Do đó, bộ giải mã sẽ không thực hiện LTPF cũng như PLC dựa trên thông tin bước ghi và thông tin độ hài hòa được cung cấp bởi bộ mã hóa.

Nếu tại S61 xác định rằng các tiêu chí thứ nhất được đáp ứng (ví dụ, độ hài hòa đó lớn hơn ngưỡng thứ nhất và do đó không ở mức độ hài hòa thấp hơn), thì tại các bước S63 và S65 sẽ kiểm tra liệu các tiêu chí thứ hai có được đáp ứng hay không. Các tiêu chí thứ hai có thể bao gồm, ví dụ, sự so sánh số đo độ hài hòa, cho khung hiện thời, với ít nhất một ngưỡng.

Ví dụ, ở bước S63, độ hài hòa (ví dụ, số đo độ hài hòa thứ hai 24a") được so sánh với ngưỡng thứ hai (trong một số ví dụ, ngưỡng thứ hai được thiết lập sao cho nó có liên quan đến hàm lượng sóng hài lớn hơn hàm lượng sóng hài liên quan đến ví dụ ngưỡng thứ nhất, với giả định rằng số đo độ hài hòa nằm giữa giá trị 0 liên quan đến tín hiệu hoàn toàn không hài hòa, và giá trị 1 liên quan đến tín hiệu hoàn toàn hài hòa).

Nếu tại S63 xác định rằng độ hài hòa không lớn hơn ngưỡng thứ hai (ví dụ, trong một số trường hợp có thể liên quan đến mức độ hài hòa trung gian), thì tại S64, khung thứ nhất 16, 16', 16" được mã hóa. Khung thứ nhất (biểu thị độ hài hòa trung gian) có thể được mã hóa để bao gồm mục dữ liệu điều khiển thứ ba 18e (ví dụ, "ltpf_pitch_lag_present") có thể là "1", mục dữ liệu điều khiển thứ nhất 16b (ví dụ "ltpf_active") có thể là "0", và giá trị của thông tin bước ghi thứ nhất 16b, chẳng hạn như độ trễ bước ghi ("ltpf_pitch_lag"). Do đó, khi nhận được khung thứ nhất 16, 16', 16", bộ giải mã sẽ sử dụng thông tin bước ghi thứ nhất 16b cho PLC, mà sẽ không sử dụng thông tin bước ghi thứ nhất 16b cho LTPF.

Đáng chú ý là, sự so sánh được thực hiện tại S61 và tại S62 có thể dựa trên các số đo độ hài hòa khác nhau, mà ví dụ có thể thu được ở các tốc độ lấy mẫu khác nhau.

Nếu ở S63 xác định rằng độ hài hòa lớn hơn ngưỡng thứ hai (ví dụ, số đo độ hài hòa thứ hai vượt quá ngưỡng thứ hai), thì tại bước S65 có thể kiểm tra xem liệu tín hiệu âm thanh là tín hiệu tạm thời hay không, ví dụ, liệu cấu trúc của tín hiệu âm thanh 11 có thay đổi theo thời gian hay không (hoặc liệu một điều kiện khác trên khung trước có được đáp ứng hay không). Ví dụ, có thể kiểm tra xem khung trước có đáp ứng điều kiện vượt quá ngưỡng thứ hai hay không. Nếu điều kiện trên khung trước cũng thỏa mãn (không tạm thời), thì tín hiệu được coi là ổn định và có thể kích hoạt bước S66. Mặt khác, phương pháp tiếp tục bước S64 để mã hóa khung thứ nhất 16, 16' hoặc 16" (xem ở trên).

Ở bước S66, khung thứ hai 17, 17', 17" có thể được mã hóa. Khung thứ hai 17" có thể bao gồm mục dữ liệu điều khiển thứ ba 18e (ví dụ "ltpf_pitch_lag_present") với giá trị "1" và mục dữ liệu điều khiển thứ hai 17c (ví dụ "ltpf_active") có thể là "1". Theo đó, thông tin bước ghi 17b (chẳng hạn như "pitch_lag" và một cách tùy chọn, cũng là thông tin độ hài hòa bổ sung 17d) có thể được mã hóa. Bộ giải mã sẽ hiểu rằng cả PLC có thông tin bước ghi và LTPF với thông tin bước ghi (và một cách tùy chọn, cũng là thông tin độ hài hòa) có thể được sử dụng.

Tại S67, khung được mã hóa có thể được truyền đến bộ giải mã (ví dụ thông qua kết nối Bluetooth), được lưu trữ trên bộ nhớ, hoặc được sử dụng theo cách khác.

Trong các bước S63 và S64, số đo tương quan chuẩn hóa nc (số đo thứ hai 24a") có thể là số đo tương quan chuẩn hóa nc thu được ở 12,8KHz (xem thêm ở trên và bên dưới). Trong bước S61, tương quan chuẩn hóa (số đo thứ nhất 24a') có thể là tương quan chuẩn hóa ở 6,4KHz (xem thêm ở trên và bên dưới).

Fig.6b thể hiện phương pháp 60b cũng có thể được sử dụng. Fig.6b thể hiện rõ ràng các ví dụ về các tiêu chí thứ hai 600 có thể được sử dụng để xác định giá trị của ltpf_active.

Như có thể thấy, các bước S60, S61, và S62 giống như trong phương pháp 60 và do đó không được lặp lại.

Ở bước S610 có thể được kiểm tra liệu:

- đối với khung trước đã thu được ltpf_active=0 (được biểu thị bởi mem_ltpf_active=0); và
- đối với khung trước, số đo tương quan chuẩn hóa nc (24a") lớn hơn ngưỡng thứ ba (ví dụ, giá trị nằm trong khoảng từ 0,92 đến 0,96, chẳng hạn như 0,94); và
- đối với khung hiện thời, số đo tương quan chuẩn hóa nc (24a") lớn hơn ngưỡng thứ ba (ví dụ, giá trị nằm trong khoảng từ 0,92 đến 0,96, chẳng hạn như 0,94).

Nếu kết quả là đúng, ltpf_active được đặt bằng 1 tại bước S614 và các bước S66 (mã hóa khung thứ hai 17, 17', 17") và S67 (truyền hoặc lưu trữ khung được mã hóa) được kích hoạt.

Nếu điều kiện thiết lập ở bước S610 không được xác minh, nó có thể được kiểm tra ở bước S611:

- đối với khung trước thu được $ltpf_active=1$ (được biểu thị bởi $mem_ltpf_active=1$);
- đối với khung hiện thời, số đo tương quan chuẩn hóa nc (24a") lớn hơn ngưỡng thứ tư (ví dụ, giá trị nằm trong khoảng từ 0,85 đến 0,95, ví dụ 0,9).

Nếu kết quả là đúng, $ltpf_active$ được đặt bằng 1 tại bước S614 và các bước S66 (mã hóa khung thứ hai 17, 17', 17") và S67 (truyền hoặc lưu trữ khung được mã hóa) được kích hoạt.

Nếu điều kiện thiết lập ở bước S611 không được xác minh, nó có thể được kiểm tra ở bước S612 liệu:

- đối với khung trước thu được $ltpf_active=0$ (được biểu thị bởi $mem_ltpf_active=0$);
- đối với khung hiện thời, khoảng cách giữa bước ghi hiện thời và bước ghi trước nhỏ hơn ngưỡng thứ năm (ví dụ, giá trị nằm trong khoảng từ 1,8 đến 2,2, chẳng hạn như 2); và
- chênh lệch giữa số đo tương quan chuẩn hóa nc (24a") của khung hiện thời và số đo tương quan chuẩn hóa mem_nc của khung trước lớn hơn ngưỡng thứ sáu (ví dụ, giá trị nằm trong khoảng từ -0,15 đến -0,05, chẳng hạn như -0,1); và
- đối với khung hiện thời, số đo tương quan chuẩn hóa nc (24a") lớn hơn ngưỡng thứ bảy (ví dụ, giá trị nằm trong khoảng từ 0,82 đến 0,86, chẳng hạn như 0,84).

(Trong một số ví dụ về các bước từ S610 đến S612, một số điều kiện ở trên có thể bỏ qua trong khi một số điều kiện có thể được duy trì.)

Nếu kết quả kiểm tra tại bước S612 là đúng, $ltpf_active$ được đặt bằng 1 tại S614 và các bước S66 (mã hóa khung thứ hai 17, 17', 17") và S67 (truyền hoặc lưu trữ khung được mã hóa) được kích hoạt.

Mặt khác, nếu không có sự kiểm tra nào ở các bước từ S610 đến S612 được xác minh, ltpf_active được đặt bằng 0 cho khung hiện thời tại S613 và bước S64 được kích hoạt, để mã hóa khung thứ nhất 16, 16', 16".

Trong các bước từ S610 đến S612, số đo tương quan chuẩn hóa nc (phép đo thứ hai 24a") có thể là số đo tương quan chuẩn hóa thu được tại 12,8KHz (xem ở trên). Trong bước S61, tương quan chuẩn hóa (số đo thứ nhất 24a') có thể là tương quan chuẩn hóa tại 6,4KHz (xem ở trên).

Như có thể thấy, một vài số liệu liên quan đến khung hiện thời và/hoặc khung trước có thể được tính. Do đó, việc đáp ứng các tiêu chí thứ hai có thể được xác minh bằng cách kiểm tra xem một vài số đo (ví dụ liên quan đến khung hiện thời và/hoặc khung trước) tương ứng ở trên hay ở dưới một vài ngưỡng (ví dụ, ít nhất là một số ngưỡng từ thứ ba đến thứ bảy của các bước từ S610 đến S612).

Một số ví dụ về cách lấy tham số cho LTPF ở phía bộ mã hóa được đưa ra sau đây.

Ví dụ về kỹ thuật lấy mẫu lại được thảo luận sau đây (các kỹ thuật khác có thể được sử dụng).

Tín hiệu đầu vào ở tốc độ lấy mẫu f_s được lấy mẫu lại với tốc độ lấy mẫu cố định là 12,8kHz. Việc lấy mẫu lại được thực hiện bằng cách sử dụng phương pháp tăng mẫu+lọc thông thấp+giảm mẫu có thể được xây dựng như sau

$$x_{12.8}(n) = P \sum_{k=-\frac{120}{P}}^{\frac{120}{P}} x\left(\left\lfloor \frac{15n}{P} \right\rfloor + k - \frac{120}{P}\right) h_{6.4}(Pk - 15n \bmod P) \quad \text{với } n = 0..127$$

với $x(n)$ là tín hiệu đầu vào, $x_{12.8}(n)$ là tín hiệu được lấy mẫu lại tại 12,8kHz, $P = \frac{192\text{kHz}}{f_s}$

là hệ số tăng mẫu và $h_{6.4}$ là đáp ứng xung của bộ lọc thông thấp FIR được cho bởi

$$h_{6.4}(n) = \begin{cases} \text{tab_resamp_filter}[n + 119] & , \text{ nếu } -120 < n < 120 \\ 0 & , \text{ ngược lại} \end{cases}$$

Ví dụ về tab_resamp_filter được đưa ra là:

```
double tab_resamp_filter[239] = {
-2.043055832879108e-05, -4.463458936757081e-05, -7.163663994481459e-05,
-1.001011132655914e-04, -1.283728480660395e-04, -1.545438297704662e-04,
-1.765445671257668e-04, -1.922569599584802e-04, -1.996438192500382e-04,
-1.968886856400547e-04, -1.825383318834690e-04, -1.556394266046803e-04,
-1.158603651792638e-04, -6.358930335348977e-05, +2.810064795067786e-19,
+7.292180213001337e-05, +1.523970757644272e-04, +2.349207769898906e-04,
+3.163786496265269e-04, +3.922117380894736e-04, +4.576238491064392e-04,
+5.078242936704864e-04, +5.382955231045915e-04, +5.450729176175875e-04,
+5.250221548270982e-04, +4.760984242947349e-04, +3.975713799264791e-04,
+2.902002172907180e-04, +1.563446669975615e-04, -5.818801416923580e-19,
-1.732527127898052e-04, -3.563859653300760e-04, -5.411552308801147e-04,
-7.184140229675020e-04, -8.785052315963854e-04, -1.011714513697282e-03,
-1.108767055632304e-03, -1.161345220483996e-03, -1.162601694464620e-03,
-1.107640974148221e-03, -9.939415631563015e-04, -8.216921898513225e-04,
-5.940177657925908e-04, -3.170746535382728e-04, +9.746950818779534e-19,
+3.452937604228947e-04, +7.044808705458705e-04, +1.061334465662964e-03,
+1.398374734488549e-03, +1.697630799350524e-03, +1.941486748731660e-03,
+2.113575906669355e-03, +2.199682452179964e-03, +2.188606246517629e-03,
+2.072945458973295e-03, +1.849752491313908e-03, +1.521021876908738e-03,
+1.093974255016849e-03, +5.811080624426164e-04, -1.422482656398999e-18,
-6.271537303228204e-04, -1.274251404913447e-03, -1.912238389850182e-03,
-2.510269249380764e-03, -3.037038298629825e-03, -3.462226871101535e-03,
-3.758006719596473e-03, -3.900532466948409e-03, -3.871352309895838e-03,
-3.658665583679722e-03, -3.258358512646846e-03, -2.674755551508349e-03,
-1.921033054368456e-03, -1.019254326838640e-03, +1.869623690895593e-18,
+1.098415446732263e-03, +2.231131973532823e-03, +3.348309272768835e-03,
+4.397022774386510e-03, +5.323426722644900e-03, +6.075105310368700e-03,
+6.603520247552113e-03, +6.866453987193027e-03, +6.830342695906946e-03,
+6.472392343549424e-03, +5.782375213956374e-03, +4.764012726389739e-03,
+3.435863514113467e-03, +1.831652835406657e-03, -2.251898372838663e-18,
-1.996476188279370e-03, -4.082668858919100e-03, -6.173080374929424e-03,
-8.174448945974208e-03, -9.988823864332691e-03, -1.151698705819990e-02,
-1.266210056063963e-02, -1.333344579518481e-02, -1.345011199343934e-02,
-1.294448809639154e-02, -1.176541543002924e-02, -9.880867320401294e-03,
-7.280036402392082e-03, -3.974730209151807e-03, +2.509617777250391e-18,
+4.586044219717467e-03, +9.703248998383679e-03, +1.525124770818010e-02,
+2.111205854013017e-02, +2.715337236094137e-02, +3.323242450843114e-02,
+3.920032029020130e-02, +4.490666443426786e-02, +5.020433088017846e-02,
+5.495420172681558e-02, +5.902970324375908e-02, +6.232097270672976e-02,
+6.473850225260731e-02, +6.621612450840858e-02, +6.671322871619612e-02,
+6.621612450840858e-02, +6.473850225260731e-02, +6.232097270672976e-02,
+5.902970324375908e-02, +5.495420172681558e-02, +5.020433088017846e-02,
+4.490666443426786e-02, +3.920032029020130e-02, +3.323242450843114e-02,
+2.715337236094137e-02, +2.111205854013017e-02, +1.525124770818010e-02,
+9.703248998383679e-03, +4.586044219717467e-03, +2.509617777250391e-18,
-3.974730209151807e-03, -7.280036402392082e-03, -9.880867320401294e-03,
-1.176541543002924e-02, -1.294448809639154e-02, -1.345011199343934e-02,
-1.333344579518481e-02, -1.266210056063963e-02, -1.151698705819990e-02,
-9.988823864332691e-03, -8.174448945974208e-03, -6.173080374929424e-03,
-4.082668858919100e-03, -1.996476188279370e-03, -2.251898372838663e-18,
+1.831652835406657e-03, +3.435863514113467e-03, +4.764012726389739e-03,
+5.782375213956374e-03, +6.472392343549424e-03, +6.830342695906946e-03,

```

```
+6.866453987193027e-03, +6.603520247552113e-03, +6.075105310368700e-03,
+5.323426722644900e-03, +4.397022774386510e-03, +3.348309272768835e-03,
+2.231131973532823e-03, +1.098415446732263e-03, +1.869623690895593e-18,
-1.019254326838640e-03, -1.921033054368456e-03, -2.674755551508349e-03,
-3.258358512646846e-03, -3.658665583679722e-03, -3.871352309895838e-03,
-3.900532466948409e-03, -3.758006719596473e-03, -3.462226871101535e-03,
-3.037038298629825e-03, -2.510269249380764e-03, -1.912238389850182e-03,
-1.274251404913447e-03, -6.271537303228204e-04, -1.422482656398999e-18,
+5.811080624426164e-04, +1.093974255016849e-03, +1.521021876908738e-03,
+1.849752491313908e-03, +2.072945458973295e-03, +2.188606246517629e-03,
+2.199682452179964e-03, +2.113575906669355e-03, +1.941486748731660e-03,
+1.697630799350524e-03, +1.398374734488549e-03, +1.061334465662964e-03,
+7.044808705458705e-04, +3.452937604228947e-04, +9.746950818779534e-19,
-3.170746535382728e-04, -5.940177657925908e-04, -8.216921898513225e-04,
-9.939415631563015e-04, -1.107640974148221e-03, -1.162601694464620e-03,
-1.161345220483996e-03, -1.108767055632304e-03, -1.011714513697282e-03,
-8.785052315963854e-04, -7.184140229675020e-04, -5.411552308801147e-04,
-3.563859653300760e-04, -1.732527127898052e-04, -5.818801416923580e-19,
+1.563446669975615e-04, +2.902002172907180e-04, +3.975713799264791e-04,
+4.760984242947349e-04, +5.250221548270982e-04, +5.450729176175875e-04,
+5.382955231045915e-04, +5.078242936704864e-04, +4.576238491064392e-04,
+3.922117380894736e-04, +3.163786496265269e-04, +2.349207769898906e-04,
+1.523970757644272e-04, +7.292180213001337e-05, +2.810064795067786e-19,
-6.358930335348977e-05, -1.158603651792638e-04, -1.556394266046803e-04,
-1.825383318834690e-04, -1.968886856400547e-04, -1.996438192500382e-04,
-1.922569599584802e-04, -1.765445671257668e-04, -1.545438297704662e-04,
-1.283728480660395e-04, -1.001011132655914e-04, -7.163663994481459e-05,
-4.463458936757081e-05, -2.043055832879108e-05};
```

Ví dụ về kỹ thuật lọc thông cao được thảo luận sau đây (các kỹ thuật khác có thể được sử dụng).

Tín hiệu được lấy mẫu lại có thể được lọc thông cao bằng cách sử dụng bộ lọc IIR bậc 2 có chức năng truyền có thể được cho bởi

$$H_{50}(z) = \frac{0.9827947082978771 - 1.965589416595754z^{-1} + 0.9827947082978771z^{-2}}{1 - 1.9652933726226904z^{-1} + 0.9658854605688177z^{-2}}$$

Ví dụ về kỹ thuật phát hiện bước ghi được thảo luận sau đây (các kỹ thuật khác có thể được sử dụng).

Tín hiệu $x_{12.8}(n)$ có thể được giảm mẫu theo hệ số 2 bằng cách sử dụng

$$x_{6.4}(n) = \sum_{k=0}^4 x_{12.8}(2n+k-3)h_2(k) \quad \text{với } n = 0..63$$

với $h_2 = \{0,1236796411180537, 0,2353512128364889, 0,2819382920909148, 0,2353512128364889, 0,1236796411180537\}$.

Độ tự tương quan của $x_{6.4}(n)$ có thể được tính là

$$R_{6.4}(k) = \sum_{n=0}^{63} x_{6.4}(n)x_{6.4}(n-k) \quad \text{với } k = k_{\min} \dots k_{\max}$$

với $k_{\min} = 17$ và $k_{\max} = 114$ là độ trễ tối thiểu và độ trễ tối đa.

Độ tự tương quan có thể được gán trọng số bằng cách sử dụng

$$R_{6.4}^w(k) = R_{6.4}(k)w(k) \quad \text{với } k = k_{\min} \dots k_{\max}$$

với $w(k)$ được xác định như sau:

$$w(k) = 1 - 0.5 \frac{(k - k_{\min})}{(k_{\max} - k_{\min})} \quad \text{với } k = k_{\min} \dots k_{\max}$$

Ước lượng thứ nhất về độ trễ bước ghi T_1 có thể là độ trễ tối đa hóa độ tự tương quan có trọng số

$$T_1 = \underset{k=k_{\min} \dots k_{\max}}{\operatorname{argmax}} R_{6.4}^w(k)$$

Ước lượng thứ hai về độ trễ bước ghi T_2 có thể là độ trễ tối đa hóa độ tự tương quan không có trọng số trong vùng lân cận của độ trễ bước ghi được ước lượng trong khung trước

$$T_2 = \underset{k=k'_{\min} \dots k'_{\max}}{\operatorname{argmax}} R_{6.4}(T)$$

với $k'_{\min} = \max(k_{\min}, T_{\text{prev}} - 4)$, $k'_{\max} = \min(k_{\max}, T_{\text{prev}} + 4)$ và T_{prev} là độ trễ bước ghi cuối cùng được ước lượng trong khung trước.

Ước lượng cuối cùng của độ trễ bước ghi trong khung hiện thời có thể được cho bởi

$$T_{\text{curr}} = \begin{cases} T_1 & \text{nếu } \text{normcorr}(x_{6.4}, 64, T_2) \leq 0.85 \cdot \text{normcorr}(x_{6.4}, 64, T_1) \\ T_2 & \text{ngược lại} \end{cases}$$

với $\text{normcorr}(x, L, T)$ là độ tương quan chuẩn hóa của tín hiệu x của độ dài L ở độ trễ T

$$\text{normcorr}(x, L, T) = \frac{\sum_{n=0}^{L-1} x(n)x(n-T)}{\sqrt{\sum_{n=0}^{L-1} x^2(n) \sum_{n=0}^{L-1} x^2(n-T)}}$$

Độ tương quan chuẩn hóa có thể là ít nhất một trong các số đo độ hài hòa thu được từ bộ phân tích tín hiệu 14 và/hoặc bộ đo độ hài hòa 24. Đây là một trong những số đo độ hài hòa có thể được sử dụng, ví dụ, để so sánh với ngưỡng thứ nhất.

Ví dụ để có được kỹ thuật dòng bit LTPF được thảo luận sau đây (các kỹ thuật khác có thể được sử dụng).

Bit thứ nhất của dòng bit LTPF báo hiệu sự có mặt của tham số độ trễ bước ghi trong dòng bit. Nó thu được bởi

$$\text{ltf_pitch_present} = \begin{cases} 1 & \text{nếu } \text{normcorr}(x_{6,4}, 64, T_{\text{curr}}) > 0.6 \\ 0 & \text{ngược lại} \end{cases}$$

Nếu ltf_pitch_present bằng 0, không có thêm bit nào được mã hóa, dẫn đến dòng bit LTPF chỉ có một bit (xem khung thứ ba 18").

Nếu ltf_pitch_present bằng 1, sẽ có thêm hai tham số được mã hóa, một tham số độ trễ bước ghi (ví dụ được mã hóa trên 9 bit), và một bit để báo hiệu sự kích hoạt LTPF (xem các khung 16" và 17"). Trong trường hợp đó, dòng bit LTPF (khung) có thể được tạo bởi 11 bit.

$$\text{nbits}_{\text{LTPF}} = \begin{cases} 1 & , \text{nếu } \text{ltf_pitch_present} = 0 \\ 11 & , \text{ngược lại} \end{cases}$$

Tham số độ trễ bước ghi và bit kích hoạt thu được như mô tả trong các phần sau.

Các dữ liệu này có thể được mã hóa trong các khung 12, 12', 12" theo các phương thức đã thảo luận ở trên.

Ví dụ để thu được tham số độ trễ LTPF được thảo luận sau đây (các kỹ thuật khác có thể được sử dụng).

Phần nguyên của tham số độ trễ bước ghi LTPF có thể được cho bởi

$$\text{ltf_pitch_int} = \underset{k=k''_{\min} \dots k''_{\max}}{\text{argmax}} R_{12,8}(k)$$

với

$$R_{12,8}(k) = \sum_{n=0}^{127} x_{12,8}(n)x_{12,8}(n-k)$$

và $k''_{\min} = \max(32, 2T_{\text{curr}} - 4)$, $k''_{\max} = \min(228, 2T_{\text{curr}} + 4)$.

Phần thập phân của độ trễ bước ghi LTPF sau đó có thể được cho bởi

$$\text{pitch_fr} = \begin{cases} 0 & \text{nếu pitch_int} \geq 157 \\ \underset{d=-2,0,2}{\text{argmax}} \text{interp}(R_{12,8}, \text{pitch_int}, d) & \text{nếu } 157 > \text{pitch_int} \geq 127 \\ \underset{d=-3 \dots 3}{\text{argmax}} \text{interp}(R_{12,8}, \text{pitch_int}, d) & \text{nếu } 127 > \text{pitch_int} > 32 \\ \underset{d=0 \dots 3}{\text{argmax}} \text{interp}(R_{12,8}, \text{pitch_int}, d) & \text{nếu pitch_int} = 32 \end{cases}$$

với

$$\text{interp}(R, T, d) = \sum_{k=-4}^4 R(T+k)h_4(4k-d)$$

và h_4 là đáp ứng xung của bộ lọc thông thấp FIR được cho bởi

$$h_4(n) = \begin{cases} \text{tab_ltpf_interp_R}(n+15) & , \text{nếu } -16 < n < 16 \\ 0 & , \text{ngược lại} \end{cases}$$

Các giá trị `tab_ltpf_interp_R` có thể là, ví dụ:

```
double | tab_ltpf_interp_[31] = {
-2.874561161519444e-03, -3.001251025861499e-03, +2.745471654059321e-03
+1.535727698935322e-02, +2.868234046665657e-02, +2.950385026557377e-02
+4.598334491135473e-03, -4.729632459043440e-02, -1.058359163062837e-01
-1.303050213607112e-01, -7.544046357555201e-02, +8.357885725250529e-02
+3.301825710764459e-01, +6.032970076366158e-01, +8.174886856243178e-01
+8.986382851273982e-01, +8.174886856243178e-01, +6.032970076366158e-01
+3.301825710764459e-01, +8.357885725250529e-02, -7.544046357555201e-02
-1.303050213607112e-01, -1.058359163062837e-01, -4.729632459043440e-02
+4.598334491135473e-03, +2.950385026557377e-02, +2.868234046665657e-02
+1.535727698935322e-02, +2.745471654059321e-03, -3.001251025861499e-03
-2.874561161519444e-03};
```

Nếu $\text{pitch_fr} < 0$ thì cả pitch_int và pitch_fr được biến đổi theo

$$\begin{aligned} \text{pitch_int} &= \text{pitch_int} - 1 \\ \text{pitch_fr} &= \text{pitch_fr} + 4 \end{aligned}$$

Cuối cùng, chỉ số tham số độ trễ bước ghi được cho bởi

$$\text{pitch_index} = \begin{cases} \text{pitch_int} + 283 & \text{nếu pitch_int} \geq 157 \\ 2\text{pitch_int} + \frac{\text{pitch_fr}}{2} + 126 & \text{nếu } 157 > \text{pitch_int} \geq 127 \\ 4\text{pitch_int} + \text{pitch_fr} - 128 & \text{nếu } 127 > \text{pitch_int} \end{cases}$$

Độ tương quan chuẩn hóa có thể được tính trước tiên như sau

$$nc = \frac{\sum_{n=0}^{127} x_i(n, 0) x_i(n - \text{pitch_int}, \text{pitch_fr})}{\sqrt{\sum_{n=0}^{127} x_i^2(n, 0) \sum_{n=0}^{127} x_i^2(n - \text{pitch_int}, \text{pitch_fr})}}$$

với

$$x_i(n, d) = \sum_{k=-2}^2 x_{12.8}(n+k) h_i(4k-d)$$

và h_i là đáp ứng xung của bộ lọc thông thấp FIR được cho bởi

$$h_i(n) = \begin{cases} \text{tab_ltpf_interp_x12k8}(n+7) & , \text{ nếu } -8 < n < 8 \\ 0 & , \text{ ngược lại} \end{cases}$$

với `tab_ltpf_interp_x12k8` được chọn, ví dụ, từ các giá trị sau:

```
double tab_ltpf_interp_x12k8[15] = {
+6.698858366939680e-03, +3.967114782344967e-02, +1.069991860896389e-01
+2.098804630681809e-01, +3.356906254147840e-01, +4.592209296082350e-01
+5.500750019177116e-01, +5.835275754221211e-01, +5.500750019177116e-01
+4.592209296082350e-01, +3.356906254147840e-01, +2.098804630681809e-01
+1.069991860896389e-01, +3.967114782344967e-02, +6.698858366939680e-03};
```

Bit kích hoạt LTPF ("`ltpf_active`") sau đó có thể được thiết lập theo:

```
if (
(mem_ltpf_active==0 && mem_nc>0.94 && nc>0.94) ||
(mem_ltpf_active==1 && nc>0.9) ||
(mem_ltpf_active==1 && abs(pit-mem_pit)<2 && (nc-mem_nc)>-0.1 &&
nc>0.84)
)
{
ltpf_active = 1;
}
else
{
ltpf_active = 0;
}
```

trong đó `mem_ltpf_active` là giá trị của `ltpf_active` trong khung trước (nó bằng 0 nếu `pitch_present=0` trong khung trước), `mem_nc` là giá trị của `nc` trong khung trước (nó bằng 0 nếu `pitch_present=0` trong khung trước), `pit=pitch_int+pitch_fr/4` và `mem_pit` là giá trị của `pit` trong khung trước (nó bằng 0 nếu `pitch_present=0` trong khung trước).

2. Phía bộ giải mã

Fig.7 thể hiện thiết bị 70. Thiết bị 70 có thể là bộ giải mã. Thiết bị 70 có thể thu được dữ liệu như thông tin tín hiệu âm thanh được mã hóa 12, 12', 12". Thiết bị 70 có thể thực hiện các thao tác được mô tả ở trên và/hoặc bên dưới. Thông tin tín hiệu âm thanh được mã hóa 12, 12', 12" có thể đã được tạo ví dụ bởi bộ mã hóa như thiết bị 10 hoặc 10' hoặc bằng cách thực hiện phương pháp 60. Ví dụ, thông tin tín hiệu âm thanh được mã hóa 12, 12', 12" có thể đã được tạo ví dụ bởi bộ mã hóa khác với thiết bị 10 hoặc 10' hoặc không thực hiện phương pháp 60. Thiết bị 70 có thể tạo thông tin tín hiệu âm thanh được giải mã được lọc 76.

Thiết bị 70 có thể bao gồm (hoặc nhận dữ liệu từ) bộ phận truyền thông (ví dụ sử dụng ăng-ten) để thu thông tin tín hiệu âm thanh được mã hóa. Truyền thông Bluetooth có thể được thực hiện. Thiết bị 70 có thể bao gồm (hoặc nhận dữ liệu từ) bộ lưu trữ (ví dụ sử dụng bộ nhớ) để thu thông tin tín hiệu âm thanh được mã hóa. Thiết bị 70 có thể bao gồm các thiết bị hoạt động trong TD và/hoặc FD.

Thiết bị 70 có thể bao gồm đầu đọc dòng bit 71 (hoặc "bộ phân tích dòng bit", hoặc "bộ biến dạng dòng bit", hoặc "bộ phân tách dòng bit") có thể giải mã thông tin tín hiệu âm thanh được mã hóa 12, 12', 12". Đầu đọc dòng bit 71 có thể bao gồm, ví dụ, bộ máy trạng thái để thể hiện dữ liệu thu được dưới dạng dòng bit. Đầu đọc dòng bit 71 có thể xuất ra sự biểu diễn được giải mã 71a của tín hiệu âm thanh 11.

Sự biểu diễn 71a được giải mã có thể trải qua một hoặc nhiều kỹ thuật xử lý xuôi dòng tới đầu đọc dòng bit (không thể hiện ở đây cho đơn giản).

Thiết bị 70 có thể chứa LTPF 73 có thể lần lượt cung cấp thông tin tín hiệu âm thanh được giải mã được lọc 73'.

Thiết bị 70 có thể bao gồm bộ điều khiển bộ lọc 72 có thể điều khiển LTPF 73.

Cụ thể, LTPF 73 có thể được điều khiển bởi thông tin độ hài hòa bổ sung (ví dụ thông tin độ khuếch đại), khi được cung cấp bởi đầu đọc dòng bit 71 (đặc biệt, khi có mặt trong trường 17d, "ltpf_gain", trong khung 17' hoặc 17").

Thêm vào đó hoặc ngoài ra, LTPF 73 có thể được điều khiển bởi thông tin bước ghi (ví dụ độ trễ bước ghi). Thông tin bước ghi có thể có trong các trường 16b hoặc

17b của các khung 16, 16', 16'', 17, 17', 17''. Tuy nhiên, như được chỉ định bởi bộ chọn 78, thông tin bước ghi không phải lúc nào cũng được sử dụng để điều khiển LTPF: khi mục dữ liệu điều khiển 16c ("ltpf_active") bằng 0, thì thông tin bước ghi không được sử dụng cho LTPF (do độ hài hòa quá thấp cho LTPF).

Thiết bị 70 có thể bao gồm bộ che 75 để thực hiện chức năng PLC để cung cấp thông tin âm thanh 76. Khi có mặt trong khung được giải mã, thông tin bước ghi có thể được sử dụng cho PLC.

Ví dụ về LTPF tại thiết bị 70 được thảo luận trong các đoạn sau.

Fig.8a và Fig.8b thể hiện các ví dụ về cú pháp cho các khung có thể được sử dụng. Các trường khác nhau cũng được biểu thị.

Như được thể hiện trên Fig.8a, đầu đọc dòng bit 71 có thể tìm kiếm giá trị thứ nhất ở một vị trí (trường) cụ thể của khung đang được mã hóa (với giả thiết rằng khung là một trong các khung 16'', 17'' và 18'' của Fig.5). Vị trí cụ thể có thể được hiểu, ví dụ, như vị trí liên quan đến mục điều khiển thứ ba 18e trong khung 18'' (ví dụ "ltpf_pitch_lag_present").

Nếu giá trị "ltpf_pitch_lag_present" 18e bằng "0", đầu đọc dòng bit 71 hiểu rằng không có thông tin nào khác cho LTPF và PLC (ví dụ, không có "ltpf_active", "ltpf_pitch_lag", "ltpf_gain").

Nếu giá trị "ltpf_pitch_lag_present" 18e bằng "1", bộ đọc 71 có thể tìm kiếm trường (ví dụ trường 1 bit) chứa dữ liệu điều khiển 16c hoặc 17c (ví dụ "ltpf_active"), biểu thị thông tin độ hài hòa (ví dụ 14a, 22a). Ví dụ, nếu "ltpf_active" bằng "0" thì có thể hiểu rằng khung là khung thứ nhất 16'' chỉ ra độ hài hòa không có giá trị đối với LTPF nhưng có thể được sử dụng cho PLC. Nếu "ltpf_active" bằng "1", có thể hiểu rằng khung là khung thứ hai 17'' có thể mang thông tin có giá trị cho cả LTPF và PLC.

Đầu đọc 71 cũng tìm kiếm trường (ví dụ trường 9 bit) chứa thông tin bước ghi 16b hoặc 17b (ví dụ "ltpf_pitch_lag"). Thông tin bước ghi này có thể được cung cấp cho bộ che 75 (cho PLC). Thông tin bước ghi này có thể được cung cấp cho bộ điều khiển bộ lọc 72/LTPF 73, nhưng chỉ khi "ltpf_active" bằng "1" (ví dụ, độ hài hòa cao hơn) như được biểu thị trên Fig.7 bởi bộ chọn 78.

Thao tác tương tự được thực hiện trong ví dụ của Fig.8b mà thêm vào đó, độ khuếch đại 17d có thể được mã hóa một cách tùy chọn.

3. Ví dụ về LTPF ở phía bộ giải mã

Tín hiệu được giải mã sau việc tổng hợp MDCT (modified discrete cosine transformation: biến đổi cosin rời rạc cải biên), việc tổng hợp MDST (modified discrete sine transformation: biến đổi sin rời rạc cải biên), hoặc việc tổng hợp dựa trên một phép biến đổi khác, có thể được lọc sau trong miền thời gian bằng bộ lọc IIR có các tham số phụ thuộc vào dữ liệu dòng bit LTPF “pitch_index” và “ltpf_active”. Để tránh gián đoạn khi các tham số thay đổi từ khung này sang khung tiếp theo, cơ chế chuyển đổi có thể được áp dụng vào phần tư thứ nhất của khung hiện thời.

Trong các ví dụ, bộ lọc LTPF IIR có thể được triển khai bằng cách sử dụng

$$\widehat{x}_{ltpf}(n) = \widehat{x}(n) - \sum_{k=0}^{L_{num}} c_{num}(k) \widehat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x}_{ltpf}\left(n - p_{int} + \frac{L_{den}}{2} - k\right)$$

với $\widehat{x}(n)$ là tín hiệu đầu vào của bộ lọc (nghĩa là tín hiệu được giải mã sau khi tổng hợp MDCT) và $\widehat{x}_{ltpf}(n)$ là tín hiệu đầu ra của bộ lọc.

Phần nguyên p_{int} và phần thập phân p_{fr} của độ trễ bước ghi LTPF có thể được tính như sau. Trước tiên độ trễ bước ghi tại 12,8kHz được khôi phục bằng cách sử dụng

$$\text{pitch_int} = \begin{cases} \text{pitch_index} - 283 & \text{nếu pitch_index} \geq 440 \\ \left\lfloor \frac{\text{pitch_index}}{2} \right\rfloor - 63 & \text{nếu } 440 > \text{pitch_index} \geq 380 \\ \left\lfloor \frac{\text{pitch_index}}{4} \right\rfloor + 32 & \text{nếu } 380 > \text{pitch_index} \end{cases}$$

$$\text{pitch_fr} = \begin{cases} 0 & \text{nếu pitch_index} \geq 440 \\ 2 * \text{pitch_index} - 4 * \text{pitch_int} + 508 & \text{nếu } 440 > \text{pitch_index} \geq 380 \\ \text{pitch_index} - 4 * \text{pitch_int} + 128 & \text{nếu } 380 > \text{pitch_index} \end{cases}$$

$$\text{pitch} = \text{pitch_int} + \frac{\text{pitch_fr}}{4}$$

Độ trễ bước ghi sau đó có thể được thu nhỏ theo tốc độ lấy mẫu đầu ra f_s và được chuyển đổi thành phần nguyên và phần thập phân bằng cách sử dụng

$$\text{pitch}_{f_s} = \text{pitch} * \frac{f_s}{12800}$$

$$p_{\text{up}} = \text{nint}(\text{pitch}_{f_s} * 4)$$

$$p_{\text{int}} = \left\lfloor \frac{p_{\text{up}}}{4} \right\rfloor$$

$$p_{\text{fr}} = p_{\text{up}} - 4 * p_{\text{int}}$$

trong đó f_s là tốc độ lấy mẫu.

Các hệ số bộ lọc $c_{\text{num}}(k)$ và $c_{\text{den}}(k, p_{\text{fr}})$ có thể được tính như sau

$$c_{\text{num}}(k) = 0.85 * \text{gain_ltpf} * \text{tab_ltpf_num_fs}[\text{gain_ind}][k] \quad \text{với } k = 0..L_{\text{num}}$$

$$c_{\text{den}}(k, p_{\text{fr}}) = \text{gain_ltpf} * \text{tab_ltpf_den_fs}[p_{\text{fr}}][k] \quad \text{với } k = 0..L_{\text{den}}$$

với

$$L_{\text{den}} = \max\left(4, \frac{f_s}{4000}\right)$$

$$L_{\text{num}} = L_{\text{den}} - 2$$

và gain_ltpf và gain_ind có thể thu được theo

```
fs_idx = min(4, (nbits/8000-1));
if (nbits < 320 + fs_idx*80)
{
    gain_ltpf = 0.4;
    gain_ind = 0;
}
else if (nbits < 400 + fs_idx*80)
{
    gain_ltpf = 0.35;
    gain_ind = 1;
}
else if (nbits < 480 + fs_idx*80)
{
    gain_ltpf = 0.3;
    gain_ind = 2;
}
else if (nbits < 560 + fs_idx*80)
{
    gain_ltpf = 0.25;
    gain_ind = 3;
}
else
{
```

```
gain_ltpf = 0;
}
```

và các bảng `tab_ltpf_num_fs[gain_ind][k]` và `tab_ltpf_den_fs[p_fr][k]` được xác định trước.

Các ví dụ về `tab_ltpf_num_fs[gain_ind][k]` được đưa ra ở đây (thay vì “fs”, tốc độ lấy mẫu được biểu thị):

```
double tab_ltpf_num_8000[4][3] = {
{6.023618207009578e-01, 4.197609261363617e-01, -1.883424527883687e-02},
{5.994768582584314e-01, 4.197609261363620e-01, -1.594928283631041e-02},
{5.967764663733787e-01, 4.197609261363617e-01, -1.324889095125780e-02},
{5.942410120098895e-01, 4.197609261363618e-01, -1.071343658776831e-02}};

double tab_ltpf_num_16000[4][3] = {
{6.023618207009578e-01, 4.197609261363617e-01, -1.883424527883687e-02},
{5.994768582584314e-01, 4.197609261363620e-01, -1.594928283631041e-02},
{5.967764663733787e-01, 4.197609261363617e-01, -1.324889095125780e-02},
{5.942410120098895e-01, 4.197609261363618e-01, -1.071343658776831e-02}};

double tab_ltpf_num_24000[4][5] = {
{3.989695588963494e-01, 5.142508607708275e-01, 1.004382966157454e-01, -
1.278893956818042e-02, -1.572280075461383e-03},
{3.948634911286333e-01, 5.123819208048688e-01, 1.043194926386267e-01, -
1.091999960222166e-02, -1.347408330627317e-03},
{3.909844475885914e-01, 5.106053522688359e-01, 1.079832524685944e-01, -
9.143431066188848e-03, -1.132124620551895e-03},
{3.873093888199928e-01, 5.089122083363975e-01, 1.114517380217371e-01, -
7.450287133750717e-03, -9.255514050963111e-04}};

double tab_ltpf_num_32000[4][7] = {
{2.982379446702096e-01, 4.652809203721290e-01, 2.105997428614279e-
01, 3.766780380806063e-02, -1.015696155796564e-02, -2.535880996101096e-03, -
3.182946168719958e-04},
{2.943834154510240e-01, 4.619294002718798e-01, 2.129465770091844e-
01, 4.066175002688857e-02, -8.693272297010050e-03, -2.178307114679820e-03, -
2.742888063983188e-04},
{2.907439213122688e-01, 4.587461910960279e-01, 2.151456974108970e-
01, 4.350104772529774e-02, -7.295495347716925e-03, -1.834395637237086e-03, -
2.316920186482416e-04},
{2.872975852589158e-01, 4.557148886861379e-01, 2.172126950911401e-
01, 4.620088878229615e-02, -5.957463802125952e-03, -1.502934284345198e-03, -
1.903851911308866e-04}};

double tab_ltpf_num_48000[4][11] = {
{1.981363739883217e-01, 3.524494903964904e-01, 2.513695269649414e-
01, 1.424146237314458e-01, 5.704731023952599e-02, 9.293366241586384e-03, -
7.226025368953745e-03, -3.172679890356356e-03, -1.121835963567014e-03, -
2.902957238400140e-04, -4.270815593769240e-05},
```

```
{1.950709426598375e-01, 3.484660408341632e-01, 2.509988459466574e-01, 1.441167412482088e-01, 5.928947317677285e-02, 1.108923827452231e-02, -6.192908108653504e-03, -2.726705509251737e-03, -9.667125826217151e-04, -2.508100923165204e-04, -3.699938766131869e-05},
{1.921810055196015e-01, 3.446945561091513e-01, 2.506220094626024e-01, 1.457102447664837e-01, 6.141132133664525e-02, 1.279941396562798e-02, -5.203721087886321e-03, -2.297324511109085e-03, -8.165608133217555e-04, -2.123855748277408e-04, -3.141271330981649e-05},
{1.894485314175868e-01, 3.411139251108252e-01, 2.502406876894361e-01, 1.472065631098081e-01, 6.342477229539051e-02, 1.443203434150312e-02, -4.254449144657098e-03, -1.883081472613493e-03, -6.709619060722140e-04, -1.749363341966872e-04, -2.593864735284285e-05}};
```

Các ví dụ về `tab_ltpf_den_fs[pfr][k]` được đưa ra ở đây (thay vì “fs”, tốc độ lấy mẫu được biểu thị):

```
double_tab_ltpf_den_8000[4][5] = {
{0.000000000000000e+00, 2.098804630681809e-01, 5.835275754221211e-01, 2.098804630681809e-01, 0.000000000000000e+00},
{0.000000000000000e+00, 1.069991860896389e-01, 5.500750019177116e-01, 3.356906254147840e-01, 6.698858366939680e-03},
{0.000000000000000e+00, 3.967114782344967e-02, 4.592209296082350e-01, 4.592209296082350e-01, 3.967114782344967e-02},
{0.000000000000000e+00, 6.698858366939680e-03, 3.356906254147840e-01, 5.500750019177116e-01, 1.069991860896389e-01}};
```

```
double_tab_ltpf_den_16000[4][5] = {
{0.000000000000000e+00, 2.098804630681809e-01, 5.835275754221211e-01, 2.098804630681809e-01, 0.000000000000000e+00},
{0.000000000000000e+00, 1.069991860896389e-01, 5.500750019177116e-01, 3.356906254147840e-01, 6.698858366939680e-03},
{0.000000000000000e+00, 3.967114782344967e-02, 4.592209296082350e-01, 4.592209296082350e-01, 3.967114782344967e-02},
{0.000000000000000e+00, 6.698858366939680e-03, 3.356906254147840e-01, 5.500750019177116e-01, 1.069991860896389e-01}};
```

```
double_tab_ltpf_den_24000[4][7] = {
{0.000000000000000e+00, 6.322231627323796e-02, 2.507309606013235e-01, 3.713909428901578e-01, 2.507309606013235e-01, 6.322231627323796e-02, 0.000000000000000e+00},
{0.000000000000000e+00, 3.459272174099855e-02, 1.986515602645028e-01, 3.626411726581452e-01, 2.986750548992179e-01, 1.013092873505928e-01, 4.263543712369752e-03},
{0.000000000000000e+00, 1.535746784963907e-02, 1.474344878058222e-01, 3.374259553990717e-01, 3.374259553990717e-01, 1.474344878058222e-01, 1.535746784963907e-02},
{0.000000000000000e+00, 4.263543712369752e-03, 1.013092873505928e-01, 2.986750548992179e-01, 3.626411726581452e-01, 1.986515602645028e-01, 3.459272174099855e-02}};
```

```
double_tab_ltpf_den_32000[4][9] = {
```

```

{0.0000000000000000e+00, 2.900401878228730e-02, 1.129857420560927e-01,
2.212024028097570e-01, 2.723909472446145e-01, 2.212024028097570e-01,
1.129857420560927e-01, 2.900401878228730e-02, 0.0000000000000000e+00},
{0.0000000000000000e+00, 1.703153418385261e-02, 8.722503785537784e-02,
1.961407762232199e-01, 2.689237982237257e-01, 2.424999102756389e-01,
1.405773364650031e-01, 4.474877169485788e-02, 3.127030243100724e-03},
{0.0000000000000000e+00, 8.563673748488349e-03, 6.426222944493845e-02,
1.687676705918012e-01, 2.587445937795505e-01, 2.587445937795505e-01,
1.687676705918012e-01, 6.426222944493845e-02, 8.563673748488349e-03},
{0.0000000000000000e+00, 3.127030243100724e-03, 4.474877169485788e-02,
1.405773364650031e-01, 2.424999102756389e-01, 2.689237982237257e-01,
1.961407762232199e-01, 8.722503785537784e-02, 1.703153418385261e-02}};

double_tab_ltpf_den_48000[4][13] = {
{0.0000000000000000e+00, 1.082359386659387e-02, 3.608969221303979e-02,
7.676401468099964e-02, 1.241530577501703e-01, 1.627596438300696e-01,
1.776771417779109e-01, 1.627596438300696e-01, 1.241530577501703e-01,
7.676401468099964e-02, 3.608969221303979e-02, 1.082359386659387e-02,
0.0000000000000000e+00},
{0.0000000000000000e+00, 7.041404930459358e-03, 2.819702319820420e-02,
6.547044935127551e-02, 1.124647986743299e-01, 1.548418956489015e-01,
1.767122381341857e-01, 1.691507213057663e-01, 1.352901577989766e-01,
8.851425011427483e-02, 4.499353848562444e-02, 1.557613714732002e-02,
2.039721956502016e-03},
{0.0000000000000000e+00, 4.146998467444788e-03, 2.135757310741917e-02,
5.482735584552816e-02, 1.004971444643720e-01, 1.456060342830002e-01,
1.738439838565869e-01, 1.738439838565869e-01, 1.456060342830002e-01,
1.004971444643720e-01, 5.482735584552816e-02, 2.135757310741917e-02,
4.146998467444788e-03},
{0.0000000000000000e+00, 2.039721956502016e-03, 1.557613714732002e-02,
4.499353848562444e-02, 8.851425011427483e-02, 1.352901577989766e-01,
1.691507213057663e-01, 1.767122381341857e-01, 1.548418956489015e-01,
1.124647986743299e-01, 6.547044935127551e-02, 2.819702319820420e-02,
7.041404930459358e-03}}

```

Tham chiếu đến việc xử lý chuyển tiếp, năm trường hợp khác nhau được xem xét.

Trường hợp thứ nhất: ltpf_active=0 và mem_ltpf_active=0

$$\widehat{x}_{\text{ltpf}}(n) = \hat{x}(n) \quad \text{với } n = 0.. \frac{N_F}{4}$$

Trường hợp thứ hai: ltpf_active=1 và mem_ltpf_active=0

$$\widehat{x}_{\text{ltpf}}(n) = \hat{x}(n) - \frac{n}{\frac{N_F}{4}} \left[\sum_{k=0}^{L_{\text{num}}} c_{\text{num}}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{\text{den}}} c_{\text{den}}(k, p_{\text{fr}}) \widehat{x}_{\text{ltpf}} \left(n - p_{\text{int}} + \frac{L_{\text{den}}}{2} - k \right) \right]$$

với $n = 0.. \frac{N_F}{4}$

Trường hợp thứ ba: ltpf_active=0 và mem_ltpf_active=1

$$\widehat{x}_{ltpf}(n) = \hat{x}(n) - \left(1 - \frac{n}{N_F}\right) \left[\sum_{k=0}^{L_{num}} c_{num}^{mem}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}^{mem}(k, p_{fr}^{mem}) \widehat{x}_{ltpf}\left(n - p_{int}^{mem} + \frac{L_{den}}{2} - k\right) \right]$$

với $n = 0.. \frac{N_F}{4}$

với c_{num}^{mem} , c_{den}^{mem} , p_{int}^{mem} và p_{fr}^{mem} là các tham số bộ lọc được tính trong khung trước.

Trường hợp thứ tư: ltpf_active=1 và mem_ltpf_active=1 và $p_{int} = p_{int}^{mem}$ và $p_{fr} = p_{fr}^{mem}$

$$\widehat{x}_{ltpf}(n) = \hat{x}(n) - \sum_{k=0}^{L_{num}} c_{num}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x}_{ltpf}\left(n - p_{int} + \frac{L_{den}}{2} - k\right)$$

với $n = 0.. \frac{N_F}{4}$

Trường hợp thứ năm: ltpf_active=1 và mem_ltpf_active=1 và ($p_{int} \neq p_{int}^{mem}$ hoặc $p_{fr} \neq p_{fr}^{mem}$)

$$\widehat{x}_{ltpf}'(n) = \hat{x}(n) - \left(1 - \frac{n}{N_F}\right) \left[\sum_{k=0}^{L_{num}} c_{num}^{mem}(k) \hat{x}(n-k) + \sum_{k=0}^{L_{den}} c_{den}^{mem}(k, p_{fr}^{mem}) \widehat{x}_{ltpf}'\left(n - p_{int}^{mem} + \frac{L_{den}}{2} - k\right) \right]$$

với $n = 0.. \frac{N_F}{4}$

$$\widehat{x}_{ltpf}(n) = \widehat{x}_{ltpf}'(n) - \frac{n}{N_F} \left[\sum_{k=0}^{L_{num}} c_{num}(k) \widehat{x}_{ltpf}'(n-k) + \sum_{k=0}^{L_{den}} c_{den}(k, p_{fr}) \widehat{x}_{ltpf}'\left(n - p_{int} + \frac{L_{den}}{2} - k\right) \right]$$

với $n = 0.. \frac{N_F}{4}$

4. Che mất gói tin

Các ví dụ về che mất gói tin (PLC) hoặc che lỗi được đưa ra ở đây.

4.1 Thông tin chung

Khung bị hỏng không cung cấp đầu ra âm thanh chính xác và sẽ bị loại bỏ.

Đối với mỗi khung được giải mã, tính hợp lệ của nó có thể được xác minh. Ví dụ, mỗi khung có thể có trường mang mã dư chu trình (cyclical redundancy code - CRC) được xác minh bằng cách thực hiện các thao tác được xác định trước được cung

cấp bởi thuật toán được xác định trước. Đầu đọc 71 (hoặc một thành phần logic khác, chẳng hạn như bộ che 75) có thể lặp lại thuật toán và xác minh xem kết quả tính được có tương ứng với giá trị trên trường CRC hay không. Nếu khung chưa được giải mã chính xác, có thể giả định rằng một số lỗi đã ảnh hưởng đến nó. Do đó, nếu xác minh được là kết quả giải mã không chính xác, khung được xem là giải mã không chính xác (không hợp lệ, bị lỗi).

Khi khung được xác định là được giải mã không chính xác, chiến lược che có thể được sử dụng để cung cấp đầu ra nghe được: nếu không, có thể sẽ nghe thấy một cái gì đó giống như một lỗ hổng âm thanh khó chịu. Do đó, cần phải tìm một số dạng khung mà "lấp đầy khoảng trống" được giữ mở bởi khung được giải mã không chính xác. Mục đích của quy trình che mất khung là để che ảnh hưởng của bất kỳ khung không có sẵn hoặc bị hỏng cho việc giải mã.

Quy trình che mất khung có thể bao gồm các phương pháp che cho các loại tín hiệu khác nhau. Hiệu suất bộ mã hóa-giải mã tốt nhất có thể trong các tình huống dễ bị hỏng với sự mất khung có thể đạt được thông qua việc chọn phương pháp phù hợp nhất. Một trong những phương pháp che mất gói tin có thể là, ví dụ, che miền thời gian TCX.

4.2 Che miền thời gian TCX

Phương pháp che miền thời gian TCX là kỹ thuật PLC dựa trên bước ghi hoạt động trong miền thời gian. Nó phù hợp nhất cho các tín hiệu có cấu trúc sóng hài chiếm ưu thế. Ví dụ về quy trình như sau: tín hiệu tổng hợp của các khung được giải mã cuối cùng được lọc ngược với bộ lọc LP như được mô tả trong phần 4.2.1 để thu được tín hiệu có chu kỳ như được mô tả trong phần 4.2.2. Tín hiệu ngẫu nhiên được tạo bởi một bộ tạo ngẫu nhiên có sự phân phối gần như đồng đều trong phần 4.2.3. Hai tín hiệu kích thích được tổng hợp để tạo thành tín hiệu kích thích tổng thể như được mô tả trong phần 4.2.4, được làm suy giảm một cách thích ứng với hệ số suy giảm được mô tả trong phần 4.2.6 và cuối cùng được lọc bằng bộ lọc LP để thu được tín hiệu thời gian được che tổng hợp. Nếu LTPF đã hoạt động trong khung tốt cuối cùng, LTPF cũng được áp dụng trên tín hiệu thời gian được che tổng hợp như được mô tả

trong phần 4.3. Để có được sự chồng lấp phù hợp với khung tốt thứ nhất sau khung bị mất, tín hiệu triệt tiêu răng cưa miền thời gian được tạo trong phần 4.2.5.

4.2.1 Tính toán tham số LPC

Phương pháp che miền thời gian TCX đang hoạt động trong miền kích thích. Hàm tự tương quan có thể được tính trên 80 dải miền tần số tương đương. Năng lượng được cường điệu trước với hệ số cường điệu trước cố định μ

f_s	μ
8000	0.62
16000	0.72
24000	0.82
32000	0.92
48000	0.92

Hàm tự tương quan được tạo cửa sổ độ trễ bằng cách sử dụng cửa sổ sau đây

$$w_{\text{lag}}(i) = \exp \left[-\frac{1}{2} \left(\frac{120\pi i}{f_s} \right)^2 \right], \quad \text{với } i = 1 \dots 16$$

trước khi nó được chuyển thành miền thời gian bằng cách sử dụng DFT xếp chồng ngược nhau. Cuối cùng, thao tác Levinson Durbin có thể được sử dụng để có được bộ lọc LP $a_E(k)$ cho khung được che. Ví dụ được đưa ra dưới đây:

```

e = R_L(0)
a^0(0) = 1
for k = 1 to N_L do
    rc =  $\frac{-\sum_{n=0}^{k-1} a^{k-1}(n)R_L(k-n)}{e}$ 
    a^k(0) = 1
    for n = 1 to k-1 do
        a^k(n) = a^{k-1}(n) + rc.a^{k-1}(k-n)
    a^k(k) = rc
    e = (1 - rc^2)e

```

Bộ lọc LP chỉ được tính trong khung bị mất thứ nhất sau khung tốt và vẫn còn trong các khung bị mất sau đó.

4.2.2 Xây dựng phần có chu kỳ của sự kích thích

Các mẫu thời gian được giải mã $N_L + T_c + \frac{N}{2}$ cuối cùng được cường điệu trước với hệ số cường điệu trước từ phần 4.2.1 bằng cách sử dụng bộ lọc

$$H_{\text{pre-emph}}(z) = 1 - \mu z^{-1}$$

để thu được tín hiệu $x_{\text{pre}}(k)$, trong đó T_c là giá trị độ trễ bước ghi pitch_int hoặc $\text{pitch_int} + 1$ nếu $\text{pitch_fr} > 0$. Các giá trị pitch_int và pitch_fr là các giá trị độ trễ bước ghi được truyền trong dòng bit.

Tín hiệu được cường điệu trước $x_{\text{pre}}(k)$ còn được lọc với bộ lọc LP nghịch đảo được tính toán để thu được tín hiệu kích thích trước đó $\text{exc}'_p(k)$. Để xây dựng tín hiệu kích thích $\text{exc}_p(k)$ cho khung bị mất hiện thời, $\text{exc}'_p(k)$ được sao chép liên tục với T_c như sau

$$\text{exc}_p(k) = \text{exc}'_p(E - T_c + k), \quad \text{với } k = 0 \dots N - 1$$

trong đó E tương ứng với mẫu cuối cùng trong $\text{exc}'_p(k)$. Nếu hệ số ổn định θ thấp hơn 1, chu kỳ bước ghi thứ nhất của $\text{exc}'_p(k)$ trước tiên được lọc thông thấp bằng bộ lọc FIR pha tuyến tính 11 nhánh được mô tả trong bảng bên dưới

f_s	Các hệ số bộ lọc FIR thông thấp
8000 - 16000	{0.0053, 0.0000, -0.0440, 0.0000, 0.2637, 0.5500, 0.2637, 0.0000, -0.0440, 0.0000, 0.0053}
24000 - 48000	{-0.0053, -0.0037, -0.0140, 0.0180, 0.2668, 0.4991, 0.2668, 0.0180, -0.0140, -0.0037, -0.0053}

Độ khuếch đại của bước ghi g'_p được tính như sau

$$g'_p = \frac{\sum_{k=0}^{N/2} x_{\text{pre}}(N_L + k) \cdot x_{\text{pre}}(N_L + T_c + k)}{\sum_{k=0}^{N/2} x_{\text{pre}}(N_L + k)^2}$$

Nếu $\text{pitch_fr} = 0$ thì $g_p = g'_p$. Ngược lại, độ khuếch đại thứ hai của bước ghi g''_p được tính như sau

$$g_p'' = \frac{\sum_{k=0}^{N/2} x_{pre}(N_L + 1 + k) \cdot x_{pre}(N_L + T_c + k)}{\sum_{k=0}^{N/3} x_{pre}(N_L + 1 + k)^2}$$

và $g_p = \max(g_p', g_p'')$. Nếu $g_p'' > g_p'$ thì T_c giảm đi một để xử lý thêm.

Cuối cùng, g_p được giới hạn bởi $0 \leq g_p \leq 1$.

Sự kích thích có chu kỳ được tạo $exc_p(k)$, được suy giảm từng mẫu một trong suốt khung bắt đầu bằng một và kết thúc bằng hệ số suy giảm α , để thu được $\widehat{exc}_p(k)$. Độ khuếch đại của bước ghi chỉ được tính trong khung bị mất thứ nhất sau khung tốt và được đặt bằng α đối với các sự mất khung liên tiếp.

4.2.3 Xây dựng phần ngẫu nhiên của sự kích thích

Phần ngẫu nhiên của sự kích thích có thể được tạo bằng bộ tạo ngẫu nhiên với sự phân phối gần như đồng đều như sau

$$exc_{n,FB}(k) = \text{extract}(exc_{n,FB}(k-1) \cdot 12821 + 16831), \quad \text{với } k = 0 \dots N-1$$

trong đó $exc_{n,FB}(-1)$ được khởi tạo với 24607 cho khung thứ nhất được che bằng phương pháp này và $\text{extract}()$ trích xuất 16 LSB của giá trị. Đối với các khung tiếp theo, $exc_{n,FB}(N-1)$ được lưu trữ và sử dụng như $exc_{n,FB}(-1)$ tiếp theo.

Để chuyển nhiều nhiều hơn sang các tần số cao hơn, tín hiệu kích thích được lọc thông cao bằng bộ lọc FIR pha tuyến tính 11 nhánh được mô tả trong bảng dưới đây để có $exc_{n,HP}(k)$.

f_s	Các hệ số bộ lọc FIR thông cao
8000 - 16000	{0, -0.0205, -0.0651, -0.1256, -0.1792, 0.8028, -0.1792, -0.1256, -0.0651, -0.0205, 0}
24000 - 48000	{-0.0517, -0.0587, -0.0820, -0.1024, -0.1164, 0.8786, -0.1164, -0.1024, -0.0820, -0.0587, -0.0517}

Để đảm bảo nhiễu có thể giảm dần thành nhiễu toàn dải với tốc độ giảm dần phụ thuộc vào hệ số suy giảm α , phần ngẫu nhiên của sự kích thích $\text{exc}_n(k)$ được tạo thành thông qua phép nội suy tuyến tính giữa toàn dải $\text{exc}_{n,\text{FB}}(k)$ và phiên bản được lọc thông cao $\text{exc}_{n,\text{HP}}(k)$ là

$$\text{exc}_n(k) = (1 - \beta) \cdot \text{exc}_{n,\text{FB}}(k) + \beta \cdot \text{exc}_{n,\text{HP}}(k), \quad \text{với } k = 0 \dots N - 1$$

trong đó $\beta = 1$ đối với khung bị mất thứ nhất sau khung tốt và

$$\beta = \beta_{-1} \cdot \alpha$$

đối với các sự mất khung thứ hai và liên tiếp khác nữa, trong đó β_{-1} là β của khung được che trước đó.

Để điều chỉnh mức nhiễu, độ khuếch đại của nhiễu g'_n được tính là

$$g'_n = \sqrt{\frac{\sum_{k=0}^{N/2-1} \left(\text{exc}'_p(E - N/2 + 1 + k) - g_p \cdot \text{exc}'_p(E - N/2 + 1 - T_c + k) \right)^2}{N/2}}$$

Nếu $T_c = \text{pitch_int}$ sau phần 4.2.2, thì $g_n = g'_n$. Ngược lại, độ khuếch đại thứ hai của nhiễu g''_n được tính như trong phương trình trên, nhưng với T_c là pitch_int . Sau đây, $g_n = \min(g'_n, g''_n)$.

Để xử lý thêm, g_n trước tiên được chuẩn hóa và sau đó nhân với $(1.1 - 0.75g_p)$ để có $\widehat{g}_n$.

Sự kích thích ngẫu nhiên được tạo $\text{exc}_n(k)$, được suy giảm đồng đều với $\widehat{g}_n$ từ mẫu thứ nhất đến mẫu năm và theo từng mẫu một trong suốt khung bắt đầu bằng $\widehat{g}_n$ và kết thúc bằng $\widehat{g}_n \cdot \alpha$ để thu được $\widehat{\text{exc}}_n(k)$. Độ khuếch đại của nhiễu g_n chỉ được tính trong khung bị mất thứ nhất sau khung tốt và được đặt thành $g_n \cdot \alpha$ cho các sự mất khung liên tiếp tiếp theo.

4.2.4 Xây dựng kích thích tổng, tổng hợp và xử lý sau

Sự kích thích ngẫu nhiên $\widehat{\text{exc}}_n(k)$ được thêm vào kích thích có chu kỳ $\widehat{\text{exc}}_p(k)$ để tạo thành tín hiệu kích thích tổng $\text{exc}_t(k)$. Tín hiệu được tổng hợp cuối cùng cho khung

được che thu được bằng cách lọc kích thích tổng với bộ lọc LP từ phần 4.2.1 và được xử lý sau với bộ lọc chỉnh giảm.

4.2.5 Triệt tiêu răng cưa miền thời gian

Để có được sự bổ sung chồng lấp thích hợp trong trường hợp khung tiếp theo là khung tốt, phần triệt tiêu răng cưa miền thời gian $\mathbf{x}_{\text{TDAc}}(k)$ có thể được tạo. Để làm được điều đó, $N - Z$ mẫu bổ sung được tạo giống như mô tả ở trên để thu được tín hiệu $\mathbf{x}(k)$ for $k = 0 \dots 2N - Z$. Trên đó, phần triệt tiêu răng cưa miền thời gian được tạo bởi các bước sau:

Không điền vào bộ đệm miền thời gian tổng hợp $\mathbf{x}(k)$

$$\hat{\mathbf{x}}(k) = \begin{cases} 0, & 0 \leq k < Z \\ \mathbf{x}(k - Z), & Z \leq k < 2N \end{cases}$$

Tạo cửa sổ $\hat{\mathbf{x}}(k)$ bằng cửa sổ MDCT $w_N(k)$

$$\hat{\mathbf{x}}_w(k) = w_N(k) \cdot \hat{\mathbf{x}}(k), \quad 0 \leq k < 2N$$

Định hình lại từ $2N$ đến N

$$y(k) = \begin{cases} -\hat{\mathbf{x}}_w\left(\frac{3N}{2} + k\right) - \hat{\mathbf{x}}_w\left(\frac{3N}{2} - 1 - k\right), & 0 \leq k < \frac{N}{2} \\ \hat{\mathbf{x}}_w\left(-\frac{N}{2} + k\right) - \hat{\mathbf{x}}_w\left(\frac{3N}{2} - 1 - k\right), & \frac{N}{2} \leq k < N \end{cases}$$

Định hình lại từ N đến $2N$

$$\hat{y}(k) = \begin{cases} y\left(\frac{N}{2} + k\right), & 0 \leq k < \frac{N}{2} \\ -y\left(\frac{3N}{2} - 1 - k\right), & \frac{N}{2} \leq k < N \\ -y\left(\frac{3N}{2} - 1 - k\right), & N \leq k < \frac{3N}{2} \\ -y\left(-\frac{3N}{2} + k\right), & \frac{3N}{2} \leq k < 2N \end{cases}$$

Tạo cửa sổ $\hat{y}(k)$ bằng cửa sổ MDCT được lật $w_N(k)$

$$\mathbf{x}_{\text{TDAc}}(k) = w_N(2N - 1 - k) \cdot \hat{y}(k), \quad 0 \leq k < 2N$$

4.2.6 Xử lý việc mất nhiều khung

Tín hiệu được xây dựng giảm dần về không. Tốc độ giảm dần được kiểm soát bởi hệ số suy giảm α phụ thuộc vào hệ số suy giảm trước đó α_{-1} , độ khuếch đại của bước ghi g_p được tính trên khung nhận chính xác cuối cùng, số khung bị xóa liên tiếp $nbLostCmpt$, và độ ổn định θ . Quy trình sau đây có thể được sử dụng để tính hệ số suy giảm α

```

if (nbLostCmpt == 1)
     $\alpha = \sqrt{g_p}$ 
    if ( $\alpha > 0.98$ )
         $\alpha = 0.98$ 
    else if ( $\alpha < 0.925$ )
         $\alpha = 0.925$ 
else if (nbLostCmpt == 2)
     $\alpha = (0.63 + 0.35 \theta) \cdot \alpha_{-1}$ 
    if  $\alpha < 0.919$ 
         $\alpha = 0.919$ ;
else if (nbLostCmpt == 3)
     $\alpha = (0.652 + 0.328 \theta) \cdot \alpha_{-1}$ 
else if (nbLostCmpt == 4)
     $\alpha = (0.674 + 0.3 \theta) \cdot \alpha_{-1}$ 
else if (nbLostCmpt == 5) {
     $\alpha = (0.696 + 0.266 \theta) \cdot \alpha_{-1}$ 
else
     $\alpha = (0.725 + 0.225 \theta) \cdot \alpha_{-1}$ 
 $g_p = \alpha$ 

```

Hệ số θ (độ ổn định của hai vector hệ số tỷ lệ liên kề cuối cùng $scf_{-2}(k)$ và $scf_{-1}(k)$) có thể được thu được ví dụ là:

$$\theta = 1.25 - \frac{1}{25} \sum_{k=0}^{15} \left(scf_{-1}(k) - scf_{-2}(k) \right)^2$$

trong đó $scf_{-2}(k)$ và $scf_{-1}(k)$ là các vector hệ số tỷ lệ của hai khung liên kề cuối cùng. Hệ số θ được giới hạn bởi $0 \leq \theta \leq 1$, với các giá trị lớn hơn θ tương ứng với các tín hiệu ổn định hơn. Điều này giới hạn các dao động năng lượng và đường bao phổ. Nếu không có hai vector hệ số tỷ lệ liên kề, hệ số θ được đặt bằng 0,8.

Để ngăn việc tăng năng lượng cao nhanh chóng, phổ được lọc thông thấp với $X_s(0) = X_s(0) \cdot 0.2$ và $X_s(1) = X_s(1) \cdot 0.5$.

4.3 Thao tác che liên quan đến LTPE

Nếu `mem_ltpf_active=1` trong khung bị che, `ltpf_active` được đặt bằng 1 nếu phương pháp che là sự lặp lại khung MDCT với sự xáo trộn ký hiệu hoặc che miền thời gian TCX. Do đó, bộ lọc sau dài hạn được áp dụng trên tín hiệu miền thời gian tổng hợp như được mô tả trong phần 1, nhưng với

$$\text{gain_ltpf} = \text{gain_ltpf_past} \cdot \alpha$$

trong đó `gain_ltpf_past` là độ khuếch đại LTPF của khung trước và α là hệ số suy giảm. Các giá trị bước ghi `pitch_int` và `pitch_fr` được sử dụng cho LTPF được sử dụng lại từ khung cuối cùng.

5. Bộ giải mã của Fig.9

Fig.9 thể hiện sơ đồ khối của bộ giải mã âm thanh 300 theo ví dụ (mà có thể là, ví dụ, triển khai của thiết bị 70).

Bộ giải mã âm thanh 300 có thể được tạo cấu hình để nhận thông tin tín hiệu âm thanh được mã hóa 310 (mà có thể là, ví dụ, thông tin tín hiệu âm thanh được mã hóa 12, 12', 12'') và để cung cấp thông tin âm thanh được giải mã 312 trên cơ sở đó.

Bộ giải mã âm thanh 300 có thể bao gồm bộ phân tích dòng bit 320 (mà có thể cũng được chỉ ra như "bộ biến dạng dòng bit" hoặc "bộ phân tích dòng bit") có thể tương ứng với đầu đọc dòng bit 71. Bộ phân tích dòng bit 320 có thể nhận thông tin tín hiệu âm thanh được mã hóa 310 và cung cấp, trên cơ sở của nó, sự biểu diễn miền tần số 322 và thông tin điều khiển 324.

Thông tin điều khiển 324 có thể bao gồm thông tin bước ghi 16b, 17b (ví dụ "ltpf_pitch_lag"), và thông tin độ hài hòa bổ sung, chẳng hạn như thông tin độ hài hòa bổ sung hoặc thông tin độ khuếch đại (ví dụ "ltpf_gain"), cũng như các mục dữ liệu điều khiển chẳng hạn như 16c, 17c, 18c liên quan đến độ hài hòa của tín hiệu âm thanh 11 tại bộ giải mã.

Thông tin điều khiển 324 cũng có thể bao gồm các mục điều khiển dữ liệu (ví dụ 16c, 17c). Bộ chọn 325 (ví dụ, tương ứng với bộ chọn 78 của Fig.7) thể hiện thông tin bước ghi được cung cấp cho thành phần LTPF 376 dưới sự điều khiển của các mục điều khiển (lần lượt được điều khiển bởi thông tin độ hài hòa thu được tại bộ mã hóa): nếu độ hài hòa của thông tin tín hiệu âm thanh được mã hóa 310 quá thấp (ví dụ dưới ngưỡng thứ hai được thảo luận ở trên), thành phần LTPF 376 không nhận được thông tin bước ghi.

Sự biểu diễn miền tần số 322 có thể, ví dụ, bao gồm các giá trị phổ được mã hóa 326, các hệ số tỷ lệ được mã hóa 328, và một cách tùy chọn, thông tin phụ bổ sung 330 mà có thể, ví dụ, điều khiển các bước xử lý cụ thể như, ví dụ, điền đầy nhiễu âm, xử lý ngay lập tức hoặc xử lý sau. Bộ giải mã âm thanh 300 cũng bao gồm việc giải mã giá trị phổ 340 mà có thể được tạo cấu hình để nhận các giá trị phổ được mã hóa 326, và để cung cấp, trên cơ sở của nó, tập hợp các giá trị phổ được giải mã 342. Bộ giải mã âm thanh 300 có thể cũng bao gồm việc giải mã hệ số tỷ lệ 350, mà có thể được tạo cấu hình để nhận các hệ số tỷ lệ được mã hóa 328 và để cung cấp, trên cơ sở của nó, tập hợp các hệ số tỷ lệ được giải mã 352.

Ngoài ra, để giải mã hệ số tỷ lệ, có thể sử dụng thành phần chuyển đổi LPC thành hệ số định tỉ lệ 354, ví dụ, trong trường hợp mà thông tin âm thanh được mã hóa bao gồm thông tin LPC được mã hóa thay vì thông tin hệ số tỷ lệ. Tuy nhiên, trong một số chế độ mã hóa (ví dụ, trong chế độ giải mã TCX của bộ giải mã âm thanh USAC hoặc trong bộ giải mã âm thanh EVS) tập hợp các hệ số LPC có thể được sử dụng để suy ra tập hợp các hệ số tỷ lệ tại phía bộ giải mã âm thanh. Chức năng này có thể đạt được bởi thành phần chuyển đổi LPC thành hệ số tỷ lệ 354.

Bộ giải mã âm thanh 300 cũng có thể bao gồm khối xử lý tùy chọn 366 để thực hiện xử lý tín hiệu tùy chọn (chẳng hạn như lấp đầy nhiễu; và/hoặc tạo hình nhiễu theo thời gian; TNS, v.v.) mà có thể được áp dụng cho các giá trị phổ được giải mã 342. Phiên bản được xử lý 366' của các giá trị phổ được giải mã 342 có thể được xuất ra bởi khối xử lý 366.

Bộ giải mã âm thanh 300 có thể cũng bao gồm bộ định tỉ lệ 360, mà có thể được tạo cấu hình để áp dụng tập hợp các hệ số được định tỉ lệ 352 cho tập hợp các giá trị

phổ 342 (hoặc các phiên bản được xử lý của chúng 366') để nhờ đó thu được tập hợp các giá trị được định tỷ lệ 362. Ví dụ, dải tần thứ nhất bao gồm nhiều giá trị phổ được giải mã 342 (hoặc các phiên bản được xử lý của chúng 366') có thể được định tỷ lệ bằng cách sử dụng hệ số tỷ lệ thứ nhất, và dải tần thứ hai bao gồm nhiều giá trị phổ được giải mã 342 có thể được định tỷ lệ bằng cách sử dụng hệ số tỷ lệ thứ hai. Theo đó, thu được tập hợp các giá trị được định tỷ lệ 362.

Bộ giải mã âm thanh 300 có thể cũng bao gồm phép biến đổi miền tần số thành miền thời gian 370, mà được tạo cấu hình để nhận các giá trị được định tỷ lệ 362, và để cung cấp sự biểu diễn miền thời gian 372 liên quan đến tập hợp các giá trị được định tỷ lệ 362. Ví dụ, phép biến đổi miền tần số thành miền thời gian 370 có thể cung cấp sự biểu diễn miền thời gian 372, mà liên quan đến khung hoặc khung con của nội dung âm thanh. Ví dụ, phép biến đổi miền tần số thành miền thời gian có thể nhận tập hợp các hệ số MDCT (hoặc MDST) (mà có thể được xem là các giá trị phổ được giải mã được định tỷ lệ) và cung cấp, trên cơ sở của nó, khối các mẫu miền thời gian có thể tạo ra sự biểu diễn miền thời gian 372.

Bộ giải mã âm thanh 300 cũng bao gồm thành phần LTPF 376 có thể tương ứng với bộ điều khiển bộ lọc 72 và LTPF 73. Thành phần LTPF 376 có thể nhận sự biểu diễn miền thời gian 372 và một cách nào đó cải biên sự biểu diễn miền thời gian 372, để nhờ đó thu được phiên bản được xử lý sau 378 của sự biểu diễn miền thời gian 372.

Bộ giải mã âm thanh 300 cũng có thể bao gồm thành phần che lỗi 380 mà ví dụ có thể tương ứng với bộ che 75 (để thực hiện chức năng PLC). Thành phần che lỗi 380 có thể, ví dụ, nhận sự biểu diễn miền thời gian 372 từ phép biến đổi miền tần số thành miền thời gian 370 và có thể, ví dụ, cung cấp thông tin âm thanh che lỗi 382 cho một hoặc nhiều khung âm thanh bị mất. Nói cách khác, nếu khung âm thanh bị mất sao cho, ví dụ, không có các giá trị phổ được mã hóa 326 nào có sẵn cho khung âm thanh đã nêu (hoặc khung âm thanh con), thành phần che lỗi 380 có thể cung cấp thông tin âm thanh che lỗi trên cơ sở sự biểu diễn miền thời gian 372 liên quan đến một hoặc nhiều khung âm thanh đứng trước khung âm thanh bị mất. Thông tin âm thanh che lỗi có thể thường là sự biểu diễn miền thời gian của nội dung âm thanh.

Về sự che lỗi, cần lưu ý rằng sự che lỗi không xảy ra cùng lúc với sự giải mã khung. Ví dụ nếu khung n tốt thì thực hiện giải mã bình thường, và cuối cùng lưu một số biến sẽ giúp ích nếu phải che khung tiếp theo, sau đó nếu $n+1$ bị mất, thì sẽ gọi hàm che đưa ra biến đến từ khung tốt trước đó. Ta cũng sẽ cập nhật một số biến số để giúp cho việc mất khung tiếp theo hoặc cho việc khôi phục khung tốt tiếp theo.

Do đó, thành phần che lỗi 380 có thể được kết nối với thành phần lưu trữ 327 trong đó các giá trị 16b, 17b, 17d được lưu trữ theo thời gian thực để sử dụng trong tương lai. Chúng sẽ chỉ được sử dụng nếu các khung tiếp theo được thừa nhận là được giải mã một cách không sạch. Mặt khác, các giá trị được lưu trữ trên thành phần lưu trữ 327 sẽ được cập nhật theo thời gian thực với các giá trị mới 16b, 17b, 17d.

Trong các ví dụ, thành phần che lỗi 380 có thể thực hiện sự lặp lại độ phân giải khung MDCT (hoặc MDST) với sự xáo trộn tín hiệu, và/hoặc che miền thời gian TCX, và/hoặc pha ECU. Trong các ví dụ, có thể chủ động thừa nhận kỹ thuật tốc hành thích hợp hơn và sử dụng nó.

Bộ giải mã âm thanh 300 cũng có thể bao gồm thành phần kết hợp tín hiệu 390 mà có thể được tạo cấu hình để nhận sự biểu diễn miền thời gian được lọc (xử lý sau) 378. Hơn nữa, sự kết hợp tín hiệu 390 có thể nhận thông tin âm thanh che lỗi 382 mà cũng là sự biểu diễn miền thời gian của tín hiệu âm thanh che lỗi được cung cấp cho khung âm thanh bị mất. Sự kết hợp tín hiệu 390 có thể, ví dụ, kết hợp các sự biểu diễn miền thời gian liên quan đến các khung âm thanh tiếp theo. Trong trường hợp có các khung âm thanh được giải mã hợp lệ tiếp theo, sự kết hợp tín hiệu 390 có thể kết hợp (ví dụ, chồng lấp và cộng) các sự biểu diễn miền thời gian liên quan đến các khung âm thanh được giải mã hợp lệ tiếp theo này. Tuy nhiên, nếu khung âm thanh bị mất, sự kết hợp tín hiệu 390 có thể kết hợp (ví dụ, chồng lấp và cộng) sự biểu diễn miền thời gian liên quan đến khung âm thanh được giải mã hợp lệ đứng trước khung âm thanh bị mất và thông tin âm thanh che lỗi liên quan đến khung âm thanh bị mất, để nhờ đó có sự chuyển tiếp mịn giữa khung âm thanh được nhận hợp lệ và khung âm thanh bị mất. Tương tự, sự kết hợp tín hiệu 390 có thể được tạo cấu hình để kết hợp (ví dụ, chồng lấp và cộng) thông tin âm thanh che lỗi liên quan đến khung âm thanh bị mất và sự biểu diễn miền thời gian liên quan đến một khung âm thanh được giải mã hợp lệ khác

theo sau khung âm thanh bị mất (hoặc một thông tin âm thanh che lỗi khác liên quan đến một khung âm thanh bị mất khác trong trường hợp nhiều khung âm thanh liên tiếp bị mất).

Theo đó, sự kết hợp tín hiệu 390 có thể cung cấp thông tin âm thanh được giải mã 312, sao cho sự biểu diễn miền thời gian 372, hoặc phiên bản được xử lý sau 378 của nó, được cung cấp cho các khung âm thanh được giải mã hợp lệ, và sao cho thông tin âm thanh che lỗi 382 được cung cấp cho khung âm thanh bị mất, trong đó thao tác chồng lấp và cộng thường được thực hiện giữa thông tin âm thanh (bất kể là nó có được cung cấp bởi phép biến đổi miền tần số thành miền thời gian 370 hoặc bởi thành phần che lỗi 380 hay không) của các khung âm thanh tiếp theo. Vì một số bộ mã hóa-giải mã có một số răng cưa trên phần chồng lấp và cộng mà cần được xóa bỏ, một cách tùy ý, ta có thể tạo ra một số răng cưa nhân tạo trên nửa khung mà ta đã tạo ra để thực hiện việc cộng chồng lấp.

Đáng chú ý là, thành phần che 380 có thể nhận được ở đầu vào thông tin bước ghi và/hoặc thông tin độ khuếch đại (16b, 17b, 17d) ngay cả khi thông tin độ khuếch đại không được cung cấp cho thành phần LTPF: điều này là do thành phần che 380 có thể hoạt động với độ hài hòa thấp hơn so với độ hài hòa mà tại đó thành phần LTPF 370 sẽ hoạt động. Như đã giải thích ở trên, trong đó độ hài hòa vượt quá ngưỡng thứ nhất nhưng ở dưới ngưỡng thứ hai, chức năng che có thể hoạt động ngay cả khi chức năng LTPF bị vô hiệu hóa hoặc bị giảm.

Đáng chú ý là, các triển khai khác có thể được chọn. Cụ thể, các thành phần khác với các thành phần 340, 350, 354, 360 và 370 có thể được sử dụng.

Đáng chú ý là, trong các ví dụ được cung cấp rằng khung thứ ba 18" có thể được sử dụng (ví dụ, không có các trường 16b, 17b, 16c, 17c), khi thu được khung thứ ba 18", không có thông tin nào từ khung thứ ba 18" được sử dụng cho thành phần LTPF 376 và cho thành phần che lỗi 380.

6. Phương pháp theo Fig.10

Phương pháp 100 được thể hiện trên Fig.10. Ở bước S101, khung (12, 12', 12'') có thể được giải mã bởi đầu đọc (71, 320). Trong các ví dụ, khung có thể được nhận (ví dụ, thông qua kết nối Bluetooth) và/hoặc thu được từ bộ lưu trữ.

Ở bước S102, tính hợp lệ của khung được kiểm tra (ví dụ với CRC, tính chẵn lẻ, v.v.). Nếu khung không hợp lệ, việc che được thực hiện (xem bên dưới).

Ngược lại, nếu khung hợp lệ, tại bước S103 sẽ kiểm tra xem thông tin bước ghi có được mã hóa trong khung không. Ví dụ, giá trị của trường 18e ("ltpf_pitch_lag_present") trong khung 12'' được kiểm tra. Trong các ví dụ, thông tin bước ghi chỉ được mã hóa nếu độ hài hòa vượt quá ngưỡng thứ nhất (ví dụ, bởi khối 21 và/hoặc ở bước S61). Tuy nhiên, bộ giải mã không thực hiện việc so sánh.

Nếu tại S103 biết rằng thông tin bước ghi thực sự được mã hóa (ví dụ, `ltpf_pitch_lag_present=1` với quy ước hiện tại), thì thông tin bước ghi được giải mã (ví dụ từ trường mã hóa thông tin bước ghi 16b hoặc 17b, "ltpf_pitch_lag") và được lưu trữ ở bước S104. Ngược lại, chu trình kết thúc và khung mới có thể được giải mã tại S101.

Sau đó, ở bước S105 sẽ kiểm tra xem LTPF có được bật hay không, tức là liệu có thể sử dụng thông tin bước ghi cho LTPF hay không. Việc xác minh này có thể được thực hiện bằng cách kiểm tra mục điều khiển tương ứng (ví dụ, 16c, 17c, "ltpf_active"). Điều này có thể có nghĩa là độ hài hòa vượt quá ngưỡng thứ hai (ví dụ, như được thừa nhận bởi khối 22 và/hoặc ở bước S63) và/hoặc sự phát triển theo thời gian không quá phức tạp (tín hiệu đủ phẳng trong khoảng thời gian). Tuy nhiên, (các) so sánh không được thực hiện bởi bộ giải mã.

Nếu xác minh rằng LTPF đang hoạt động, thì LTPF được thực hiện ở bước S106. Nếu không, LTPF bị bỏ qua. Chu kỳ kết thúc. Khung mới có thể được giải mã ở S101.

Tham chiếu đến sự che, nó có thể được chia nhỏ thành nhiều bước. Ở bước S107 xác minh xem thông tin bước ghi của khung trước (hoặc thông tin bước ghi của một trong các khung trước) có được lưu trong bộ nhớ hay không (tức là ở sự bố trí).

Nếu xác minh được rằng thông tin bước ghi được tìm kiếm được lưu trữ, thì việc che lỗi có thể được thực hiện (ví dụ, bằng thành phần 75 hoặc 380) ở bước S108. Việc lặp lại độ phân giải khung MDCT (hoặc MDST) với sự xáo trộn tín hiệu và/hoặc che miền thời gian TCX, và/hoặc pha ECU có thể được thực hiện.

Mặt khác, nếu tại S107 xác minh được rằng không có thông tin bước ghi mới nào được lưu trữ (do hệ quả của việc các khung trước liên quan đến độ hài hòa cực thấp hoặc độ biến thiên tín hiệu cực cao), kỹ thuật che khác được biết và không ngụ ý sử dụng về thông tin bước ghi được cung cấp bởi bộ mã hóa có thể được sử dụng ở bước S109. Một số kỹ thuật này có thể dựa trên việc ước lượng thông tin bước ghi và/hoặc thông tin độ hài hòa khác tại bộ giải mã. Trong một số ví dụ, không có kỹ thuật che nào có thể được thực hiện trong trường hợp này.

Sau khi thực hiện việc che, chu trình kết thúc và khung mới có thể được giải mã tại S101.

7. Thảo luận về giải pháp

Giải pháp được đề xuất có thể được xem là chỉ giữ một bộ dò bước ghi ở phía bộ mã hóa và gửi tham số độ trễ bước ghi bất cứ khi nào LTPF hoặc PLC cần thông tin này. Một bit được sử dụng để báo hiệu rằng thông tin bước ghi có mặt hay không trong dòng bit. Một bit bổ sung được sử dụng để báo hiệu rằng LTPF có hoạt động hay không.

Bằng cách sử dụng hai bit báo hiệu thay vì một, giải pháp được đề xuất có thể cung cấp trực tiếp thông tin độ trễ bước ghi cho cả hai mô đun mà không có thêm bất kỳ sự phức tạp nào, ngay cả trong trường hợp PLC dựa trên bước ghi hoạt động mà không phải LTPF.

Theo đó, có thể thu được sự kết hợp của LTPF và PLC dựa trên bước ghi có độ phức tạp thấp.

7.1 Bộ mã hóa

a. Một độ trễ bước ghi cho mỗi khung được ước lượng bằng cách sử dụng thuật toán phát hiện bước ghi. Điều này có thể được thực hiện trong 3 bước để giảm độ phức tạp và cải thiện độ chính xác. Độ trễ bước ghi thứ nhất được ước lượng thô bằng

cách sử dụng "phép phân tích bước ghi vòng lặp mở" ở tốc độ lấy mẫu giảm (xem ví dụ tài liệu [1] hoặc [5]). Phần nguyên của độ trễ bước ghi sau đó được tinh chỉnh bằng cách tối đa hóa hàm tương quan với tốc độ lấy mẫu cao hơn. Bước thứ ba là ước lượng phần thập phân của độ trễ bước ghi ví dụ bằng cách tối đa hóa hàm tương quan nội suy.

b. Quyết định được đưa ra để mã hóa hoặc không mã hóa độ trễ bước ghi trong dòng bit. Phép đo về độ hài hòa của tín hiệu có thể được sử dụng ví dụ sự tương quan chuẩn hóa. Sau đó, bit `ltpf_pitch_lag_present` được đặt bằng 1 nếu độ hài hòa của tín hiệu ở trên ngưỡng và bằng 0 nếu ở dưới ngưỡng. Độ trễ bước ghi `ltpf_pitch_lag` được mã hóa trong dòng bit nếu `ltpf_pitch_lag_present` bằng 1.

c. Trong trường hợp `ltpf_pitch_lag_present` bằng 1, quyết định thứ hai được đưa ra để kích hoạt hoặc không kích hoạt công cụ LTPF trong khung hiện thời. Quyết định này cũng có thể dựa trên độ hài hòa của tín hiệu, ví dụ sự tương quan chuẩn hóa, nhưng với ngưỡng cao hơn và thêm vào đó là cơ chế trễ để đưa ra quyết định ổn định. Quyết định này thiết lập bit `ltpf_active`.

d. (Tùy chọn) trong trường hợp `ltpf_active` bằng 1, độ khuếch đại LTPF được ước lượng và được mã hóa trong dòng bit. Độ khuếch đại LTPF có thể được ước lượng bằng cách sử dụng hàm dựa trên sự tương quan và được lượng tử hóa bằng cách sử dụng phép lượng tử hóa đồng nhất.

7.2 Dòng bit

Cú pháp dòng bit được thể hiện trên Fig.8a và Fig.8b theo các ví dụ.

7.3 Bộ giải mã

Nếu bộ giải mã nhận một cách chính xác khung không bị hỏng:

a. Dữ liệu LTPF được giải mã từ dòng bit

b. Nếu `ltpf_pitch_lag_present` bằng 0 hoặc `ltpf_active` bằng 0, thì bộ giải mã LTPF được gọi với độ khuếch đại LTPF bằng 0 (trong trường hợp đó không có độ trễ bước ghi nào).

c. Nếu `ltpf_pitch_lag_present` bằng 1 và `ltpf_active` bằng 1, thì bộ giải mã LTPF được gọi với độ trễ bước ghi được giải mã và độ khuếch đại được giải mã.

Nếu bộ giải mã nhận được khung bị hỏng hoặc nếu khung bị mất:

a. Quyết định được đưa ra là có sử dụng PLC dựa trên bước ghi để che khung bị mất/bị hỏng hay không. Quyết định này dựa trên dữ liệu LTPF của khung tốt cuối cùng cộng với các thông tin khác.

b. Nếu `ltpf_pitch_lag_present` của khung tốt cuối cùng bằng 0, thì PLC dựa trên bước ghi không được sử dụng. Một phương pháp PLC khác được sử dụng trong trường hợp đó, chẳng hạn như sự lặp lại khung với sự xáo trộn dấu hiệu (xem tài liệu [7]).

c. Nếu `ltpf_pitch_lag_present` của khung tốt cuối cùng bằng 1 và có thể các điều kiện khác được đáp ứng, thì PLC dựa trên bước ghi được sử dụng để che khung bị mất/bị hỏng. Môđun PLC sử dụng độ trễ bước ghi `ltpf_pitch_lag` được giải mã từ dòng bit của khung tốt cuối cùng.

8. Các ví dụ khác

Fig.11 thể hiện hệ thống 110 có thể triển khai thiết bị mã hóa 10 hoặc 10' và/hoặc thực hiện phương pháp 60. Hệ thống 110 có thể bao gồm bộ xử lý 111 và bộ nhớ không tạm thời 112 lưu trữ các lệnh mà khi được chạy bởi bộ xử lý 111, có thể khiến bộ xử lý 111 thực hiện việc ước lượng bước ghi 113 (ví dụ để triển khai bộ ước lượng bước ghi 13), phân tích tín hiệu 114 (ví dụ để triển khai bộ phân tích tín hiệu 14 và/hoặc bộ đo độ hài hòa 24), và tạo dòng bit 115 (ví dụ để triển khai bộ tạo dòng bit 15 và/hoặc các bước S62, S64 và/hoặc S66). Hệ thống 110 có thể bao gồm bộ phận đầu vào 116 mà có thể thu được tín hiệu âm thanh (ví dụ tín hiệu âm thanh 11). Do đó, bộ xử lý 111 có thể thực hiện các quy trình để thu được sự biểu diễn được mã hóa (ví dụ ở định dạng khung 12, 12', 12'') của tín hiệu âm thanh. Sự biểu diễn được mã hóa này có thể được cung cấp cho các bộ phận bên ngoài bằng cách sử dụng bộ phận đầu ra 117. Bộ phận đầu ra 117 có thể bao gồm, ví dụ, bộ phận truyền thông để truyền thông với các thiết bị bên ngoài (ví dụ bằng cách sử dụng truyền thông không dây, chẳng hạn như Bluetooth) và/hoặc các không gian lưu trữ bên ngoài. Bộ xử lý 111 có

thể lưu sự biểu diễn được mã hóa của tín hiệu âm thanh trong không gian lưu trữ cục bộ 118.

Fig.12 thể hiện hệ thống 120 có thể triển khai thiết bị giải mã 70 hoặc 300 và/hoặc thực hiện phương pháp 100. Hệ thống 120 có thể bao gồm bộ xử lý 121 và bộ nhớ không tạm thời 122 lưu trữ các lệnh mà khi được chạy bởi bộ xử lý 121, có thể khiến bộ xử lý 121 thực hiện việc đọc dòng bit 123 (ví dụ, để triển khai đầu đọc bước ghi 71 và/hoặc 320 và/hoặc bước S101 bộ phận 75 hoặc 380 và/hoặc các bước từ S107 đến S109), điều khiển bộ lọc 124 (ví dụ để triển khai LTPF 73 hoặc 376 và/hoặc bước S106), và che 125 (ví dụ để triển khai). Hệ thống 120 có thể bao gồm bộ phận đầu vào 126 mà có thể thu được sự biểu diễn được giải mã của tín hiệu âm thanh (ví dụ ở dạng các khung 12, 12', 12"). Do đó, bộ xử lý 121 có thể thực hiện các quy trình để thu được sự biểu diễn được giải mã của tín hiệu âm thanh. Sự biểu diễn được giải mã này có thể được cung cấp cho các bộ phận bên ngoài bằng cách sử dụng bộ phận đầu ra 127. Bộ phận đầu ra 127 có thể bao gồm, ví dụ, bộ phận truyền thông để truyền thông với các thiết bị bên ngoài (ví dụ bằng cách sử dụng truyền thông không dây, chẳng hạn như Bluetooth) và/hoặc các không gian lưu trữ bên ngoài. Bộ xử lý 121 có thể lưu sự biểu diễn được giải mã của tín hiệu âm thanh trong không gian lưu trữ cục bộ 128.

Trong các ví dụ, các hệ thống 110 và 120 có thể là cùng một thiết bị.

Fig.13 thể hiện phương pháp 1300 theo các ví dụ. Ở phía bộ mã hóa, ở bước S130, phương pháp có thể cung cấp việc mã hóa tín hiệu âm thanh (ví dụ, theo bất kỳ phương pháp nào ở trên hoặc sử dụng ít nhất một số thiết bị thảo luận ở trên) và suy ra thông tin độ hài hòa và/hoặc thông tin bước ghi.

Ở phía bộ mã hóa, ở bước S131, phương pháp có thể cung cấp việc xác định (ví dụ, trên cơ sở thông tin độ hài hòa chẳng hạn như các phép đo độ hài hòa) liệu thông tin bước ghi có phù hợp với ít nhất LTPF và/hoặc chức năng che lỗi được vận hành ở phía bộ giải mã hay không.

Ở phía bộ mã hóa, ở bước S132, phương pháp có thể cung cấp sự truyền từ bộ mã hóa (ví dụ không dây, ví dụ sử dụng Bluetooth) và/hoặc việc lưu trữ trong bộ nhớ dòng bit chứa sự biểu diễn kỹ thuật số của tín hiệu âm thanh và thông tin liên quan đến độ hài hòa. Bước này cũng có thể cung cấp việc báo hiệu cho bộ giải mã xem thông tin

bước ghi có được làm thích ứng cho LTPF và/hoặc che lỗi hay không. Ví dụ, mục điều khiển thứ ba 18e ("ltpf_pitch_lag_present") có thể báo hiệu rằng thông tin bước ghi (được mã hóa trong dòng bit) được làm thích ứng hoặc không được làm thích ứng cho ít nhất là che lỗi theo giá trị được mã hóa trong mục điều khiển thứ ba 18e. Ví dụ, mục điều khiển thứ nhất 16a (ltpf_active=0) có thể báo hiệu rằng thông tin bước ghi (được mã hóa trong dòng bit là "ltpf_pitch_lag") được làm thích ứng cho che lỗi nhưng không được làm thích ứng cho LTPF (ví dụ, do độ hài hòa trung gian của nó). Ví dụ, mục điều khiển thứ hai 17a (ltpf_active=1) có thể báo hiệu rằng thông tin bước ghi (được mã hóa trong dòng bit là "ltpf_pitch_lag") được làm thích ứng cho cả che lỗi và LTPF (ví dụ, do độ hài hòa cao hơn của nó).

Tại phía bộ giải mã, phương pháp có thể đề xuất, ở bước S134, việc giải mã sự biểu diễn kỹ thuật số của tín hiệu âm thanh và sử dụng thông tin bước ghi cho LTPF và/hoặc che lỗi theo mẫu báo hiệu của bộ mã hóa.

Tùy thuộc vào các yêu cầu triển khai nhất định, các ví dụ có thể được triển khai trong phần cứng. Việc triển khai có thể được thực hiện bằng cách sử dụng phương tiện lưu trữ số, ví dụ đĩa mềm, đĩa đa năng kỹ thuật số (DVD), đĩa Blu-Ray, đĩa compac (CD), bộ nhớ chỉ đọc (ROM), bộ nhớ chỉ đọc có thể lập trình (PROM), bộ nhớ chỉ đọc có thể xóa và có thể lập trình (EPROM), bộ nhớ chỉ đọc có thể lập trình xóa được bằng điện (EEPROM) hoặc bộ nhớ flash, có các tín hiệu điều khiển có thể đọc được bằng điện tử được lưu trữ trên đó, mà kết hợp (hoặc có khả năng kết hợp) với hệ thống máy tính có thể lập trình sao cho phương pháp tương ứng được thực hiện. Do đó, phương tiện lưu trữ số có thể đọc được bằng máy tính.

Nói chung, các ví dụ có thể được triển khai như sản phẩm chương trình máy tính với các lệnh chương trình, các lệnh chương trình có tác dụng để thực hiện một trong số các phương pháp khi sản phẩm chương trình máy tính chạy trên máy tính. Các lệnh chương trình có thể ví dụ được lưu trữ trên phương tiện có thể đọc được bằng máy.

Các ví dụ khác bao gồm chương trình máy tính để thực hiện một trong số các phương pháp được mô tả ở đây, được lưu trữ trên vật mang có thể đọc được bằng máy. Do đó, nói cách khác, ví dụ của phương pháp là chương trình máy tính có các lệnh

chương trình để thực hiện một trong số các phương pháp được mô tả ở đây, khi chương trình máy tính chạy trên máy tính.

Do đó, ví dụ khác của các phương pháp là vật mang dữ liệu (hoặc vật ghi lưu trữ số, hoặc vật ghi có thể đọc được bằng máy tính) chứa chương trình máy tính đã được ghi lại trên đó để thực hiện một trong số các phương pháp được mô tả ở đây. Vật mang dữ liệu, phương tiện lưu trữ kỹ thuật số hoặc phương tiện được ghi lại là hữu hình và/hoặc không tạm thời, thay vì các tín hiệu vô hình và tạm thời.

Ví dụ khác bao gồm bộ xử lý, ví dụ máy tính, hoặc thiết bị logic lập trình được thực hiện một trong số các phương pháp được mô tả ở đây.

Ví dụ khác bao gồm máy tính có chương trình máy tính được cài đặt trên đó để thực hiện một trong số các phương pháp được mô tả ở đây.

Ví dụ khác bao gồm thiết bị hoặc hệ thống truyền tải (ví dụ, băng điện tử hoặc quang học) chương trình máy tính để thực hiện một trong số các phương pháp được mô tả ở đây đến bộ nhận. Bộ nhận có thể là, ví dụ, máy tính, thiết bị di động, thiết bị nhớ hoặc tương tự. Thiết bị hoặc hệ thống có thể, ví dụ, gồm có máy chủ tập tin để chuyển chương trình máy tính đến bộ nhận.

Theo một số ví dụ, thiết bị logic lập trình được (ví dụ, mảng cổng lập trình được dạng trường) có thể được sử dụng để thực hiện một số hoặc tất cả các chức năng của các phương pháp được mô tả ở đây. Trong một số ví dụ, mảng cổng lập trình được dạng trường có thể kết hợp với bộ vi xử lý để thực hiện một trong số các phương pháp được mô tả ở đây. Thông thường, các phương pháp tốt hơn là được thực hiện bởi thiết bị phần cứng thích hợp bất kỳ.

Các ví dụ được mô tả ở trên chỉ mang tính minh họa cho các nguyên tắc được thảo luận ở trên. Cần hiểu rằng các biến thể và biến đổi của các cách sắp xếp và các chi tiết được mô tả ở đây sẽ là rõ ràng. Do đó, mục đích của sáng chế chỉ bị giới hạn bởi phạm vi của các điểm yêu cầu bảo hộ sắp đưa ra dưới đây và không bị giới hạn bởi các chi tiết cụ thể được biểu diễn bằng cách mô tả và giải thích của các ví dụ ở đây.

YÊU CẦU BẢO HỘ

1. Thiết bị giải mã thông tin tín hiệu âm thanh liên quan đến tín hiệu âm thanh được chia thành chuỗi các khung, mỗi khung trong số chuỗi các khung là một khung trong số khung thứ nhất, khung thứ hai, và khung thứ ba, thiết bị bao gồm:

đầu đọc dòng bit được tạo cấu hình để đọc thông tin tín hiệu âm thanh được mã hóa bao gồm:

sự biểu diễn được mã hóa của tín hiệu âm thanh cho khung thứ nhất, khung thứ hai và khung thứ ba;

thông tin bước ghi thứ nhất cho khung thứ nhất và mục dữ liệu điều khiển thứ nhất bao gồm giá trị thứ nhất; và

thông tin bước ghi thứ hai cho khung thứ hai và mục dữ liệu điều khiển thứ hai có giá trị thứ hai khác với giá trị thứ nhất, trong đó mục dữ liệu điều khiển thứ nhất và mục dữ liệu điều khiển thứ hai nằm trong cùng một trường; và

mục dữ liệu điều khiển thứ ba cho khung thứ nhất, khung thứ hai, và khung thứ ba, mục dữ liệu điều khiển thứ ba biểu thị sự có mặt hay vắng mặt của thông tin bước ghi thứ nhất và/hoặc thông tin bước ghi thứ hai, mục dữ liệu điều khiển thứ ba được mã hóa trong một bit đơn bao gồm giá trị phân biệt khung thứ ba từ khung thứ nhất và thứ hai, khung thứ ba bao gồm định dạng thiếu thông tin bước ghi thứ nhất, mục dữ liệu điều khiển thứ nhất, thông tin bước ghi thứ hai, và mục dữ liệu điều khiển thứ hai;

bộ điều khiển được tạo cấu hình để điều khiển bộ lọc sau dài hạn LTPF và để:

kiểm tra mục dữ liệu điều khiển thứ ba để xác minh xem khung có phải là khung thứ ba không, và trong trường hợp xác minh rằng khung không phải là khung thứ ba, sẽ kiểm tra mục dữ liệu thứ nhất và mục dữ liệu điều khiển thứ hai để xác minh xem khung là khung thứ nhất hay khung thứ hai, để:

lọc sự biểu diễn được giải mã của tín hiệu âm thanh trong khung thứ hai bằng cách sử dụng thông tin bước ghi thứ hai, và lưu trữ thông tin bước ghi thứ hai để che khung âm thanh được giải mã không chính xác

sau đó, trong trường hợp xác minh được rằng mục dữ liệu điều khiển thứ hai bao gồm giá trị thứ hai;

vô hiệu hóa LTPF cho khung thứ nhất, nhưng lưu trữ thông tin bước ghi thứ nhất để che khung âm thanh được giải mã không chính xác sau đó, trong trường hợp xác minh được rằng mục dữ liệu điều khiển thứ nhất bao gồm giá trị thứ nhất; và

cả việc vô hiệu hóa LTPF và lưu trữ thông tin bước ghi để che khung âm thanh được giải mã không chính xác sau đó, trong trường hợp xác minh từ mục dữ liệu điều khiển thứ ba rằng khung là khung thứ ba.

2. Thiết bị theo điểm 1, trong đó:

trong thông tin tín hiệu âm thanh được mã hóa, đối với khung thứ nhất, một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ nhất và trường dữ liệu cố định được dành riêng cho thông tin bước ghi thứ nhất.

3. Thiết bị theo điểm 1, trong đó:

trong thông tin tín hiệu âm thanh được mã hóa, đối với khung thứ hai, một bit đơn được dành riêng cho mục dữ liệu điều khiển thứ hai và trường dữ liệu cố định được dành riêng cho thông tin bước ghi thứ hai.

4. Thiết bị theo điểm 1, còn bao gồm:

bộ che được tạo cấu hình để sử dụng thông tin bước ghi thứ nhất và/hoặc thứ hai để che khung âm thanh được giải mã không chính xác sau đó.

5. Thiết bị theo điểm 4, bộ che được tạo cấu hình để:

trong trường hợp xác định việc giải mã khung không hợp lệ, sẽ kiểm tra xem thông tin bước ghi liên quan đến khung được giải mã chính xác trước đó có được lưu trữ không,

để che một khung được giải mã không hợp lệ với khung thu được bằng cách sử dụng thông tin bước ghi được lưu trữ.

6. Phương pháp giải mã thông tin tín hiệu âm thanh liên quan đến tín hiệu âm thanh được chia thành chuỗi các khung, trong đó mỗi khung là một khung trong số khung thứ nhất, khung thứ hai, và khung thứ ba, phương pháp bao gồm các bước:

đọc thông tin tín hiệu âm thanh được mã hóa bao gồm:

sự biểu diễn được mã hóa của tín hiệu âm thanh cho khung thứ nhất và khung thứ hai;

thông tin bước ghi thứ nhất cho khung thứ nhất và mục dữ liệu điều khiển thứ nhất bao gồm giá trị thứ nhất;

thông tin bước ghi thứ hai cho khung thứ hai và mục dữ liệu điều khiển thứ hai bao gồm giá trị thứ hai khác với giá trị thứ nhất, trong đó mục dữ liệu điều khiển thứ nhất và mục dữ liệu điều khiển thứ hai nằm trong cùng một trường; và

mục dữ liệu điều khiển thứ ba cho khung thứ nhất, khung thứ hai, và khung thứ ba, mục dữ liệu điều khiển thứ ba biểu thị sự có mặt hoặc vắng mặt của thông tin bước ghi thứ nhất và/hoặc thông tin bước ghi thứ hai, mục dữ liệu điều khiển thứ ba được mã hóa trong một bit đơn bao gồm giá trị phân biệt khung thứ ba từ khung thứ nhất và thứ hai, khung thứ ba bao gồm định dạng thiếu thông tin bước ghi thứ nhất, mục dữ liệu điều khiển thứ nhất, thông tin bước ghi thứ hai, và mục dữ liệu điều khiển thứ hai,

khi xác định rằng mục dữ liệu điều khiển thứ nhất bao gồm giá trị thứ nhất, sẽ sử dụng thông tin bước ghi thứ nhất cho bộ lọc sau dài hạn LTPF, và cho chức năng che lỗi;

khi xác định giá trị thứ hai của mục dữ liệu điều khiển thứ hai, sẽ vô hiệu hóa LTPF nhưng sử dụng thông tin bước ghi thứ hai cho chức năng che lỗi; và

khi xác định rằng khung là khung thứ ba, sẽ vô hiệu hóa LTPF và vô hiệu hóa việc sử dụng sự biểu diễn được mã hóa của tín hiệu âm thanh cho chức năng che lỗi.

7. Phương tiện lưu trữ kỹ thuật số không tạm thời có chương trình máy tính được lưu trữ trên đó để thực hiện phương pháp giải mã thông tin tín hiệu âm thanh liên quan đến tín hiệu âm thanh được chia thành một chuỗi khung, trong đó mỗi khung là một

khung trong số khung thứ nhất, khung thứ hai, và khung thứ ba, phương pháp bao gồm các bước:

đọc thông tin tín hiệu âm thanh được mã hóa bao gồm:

sự biểu diễn được mã hóa của tín hiệu âm thanh cho khung thứ nhất và khung thứ hai;

thông tin bước ghi thứ nhất cho khung thứ nhất và mục dữ liệu điều khiển thứ nhất bao gồm giá trị thứ nhất;

thông tin bước ghi thứ hai cho khung thứ hai và mục dữ liệu điều khiển thứ hai bao gồm giá trị thứ hai khác với giá trị thứ nhất, trong đó mục dữ liệu điều khiển thứ nhất và mục dữ liệu điều khiển thứ hai nằm trong cùng một trường; và

mục dữ liệu điều khiển thứ ba cho khung thứ nhất, khung thứ hai, và khung thứ ba, mục dữ liệu điều khiển thứ ba biểu thị sự có mặt hoặc vắng mặt của thông tin bước ghi thứ nhất và/hoặc thông tin bước ghi thứ hai, mục dữ liệu điều khiển thứ ba được mã hóa trong một bit đơn bao gồm giá trị phân biệt khung thứ ba từ khung thứ nhất và thứ hai, khung thứ ba bao gồm định dạng thiếu thông tin bước ghi thứ nhất, mục dữ liệu điều khiển thứ nhất, thông tin bước ghi thứ hai, và mục dữ liệu điều khiển thứ hai,

khi xác định rằng mục dữ liệu điều khiển thứ nhất bao gồm giá trị thứ nhất, sẽ sử dụng thông tin bước ghi thứ nhất cho bộ lọc sau dải hạn LTPF, và cho chức năng che lỗi;

khi xác định giá trị thứ hai của mục dữ liệu điều khiển thứ hai, sẽ vô hiệu hóa LTPF nhưng sử dụng thông tin bước ghi thứ hai cho chức năng che lỗi; và

khi xác định rằng khung là khung thứ ba, sẽ vô hiệu hóa LTPF và vô hiệu hóa việc sử dụng sự biểu diễn được mã hóa của tín hiệu âm thanh cho chức năng che lỗi, khi chương trình máy tính đã nêu được chạy bởi máy tính.

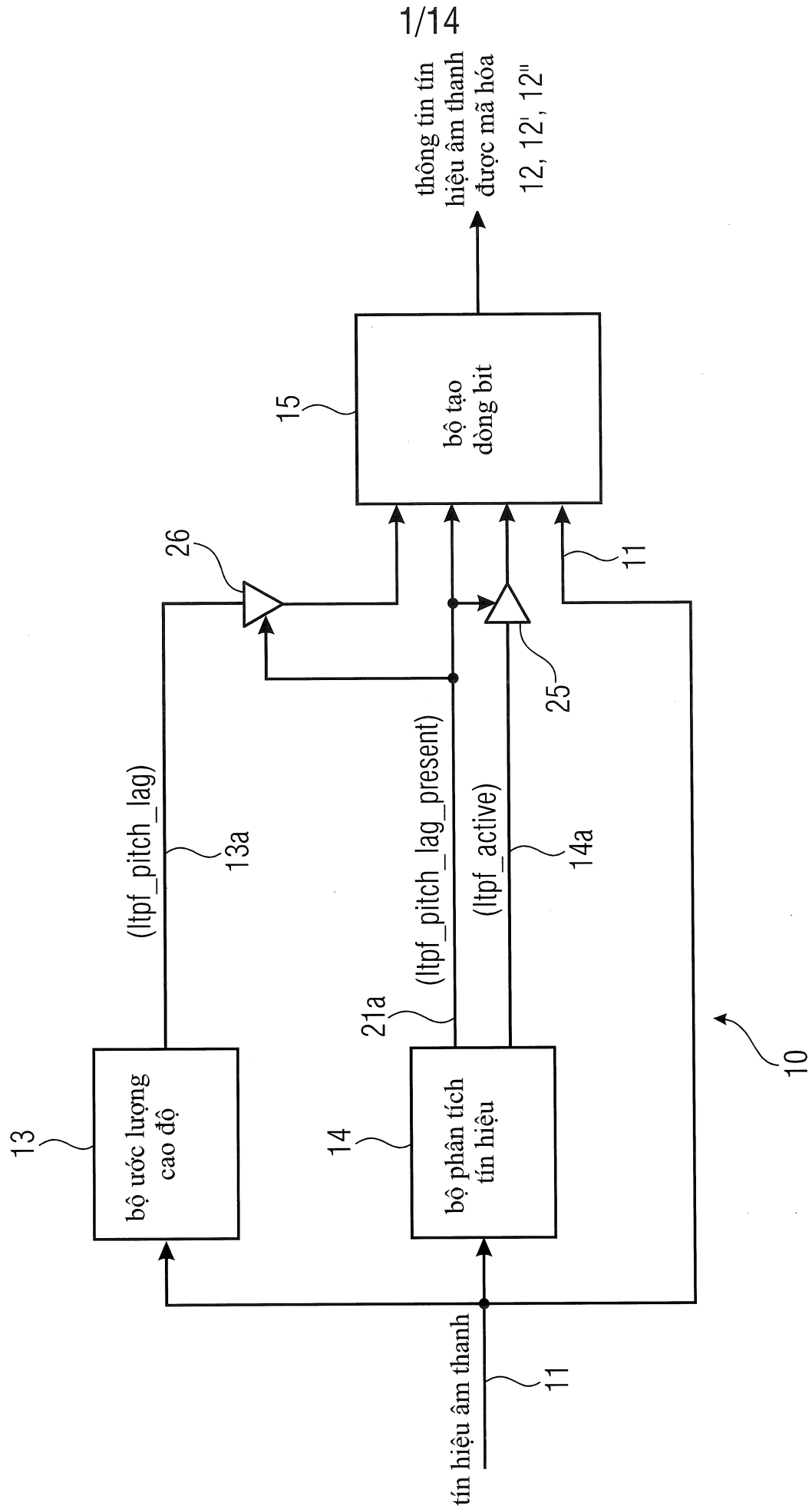


Fig. 1

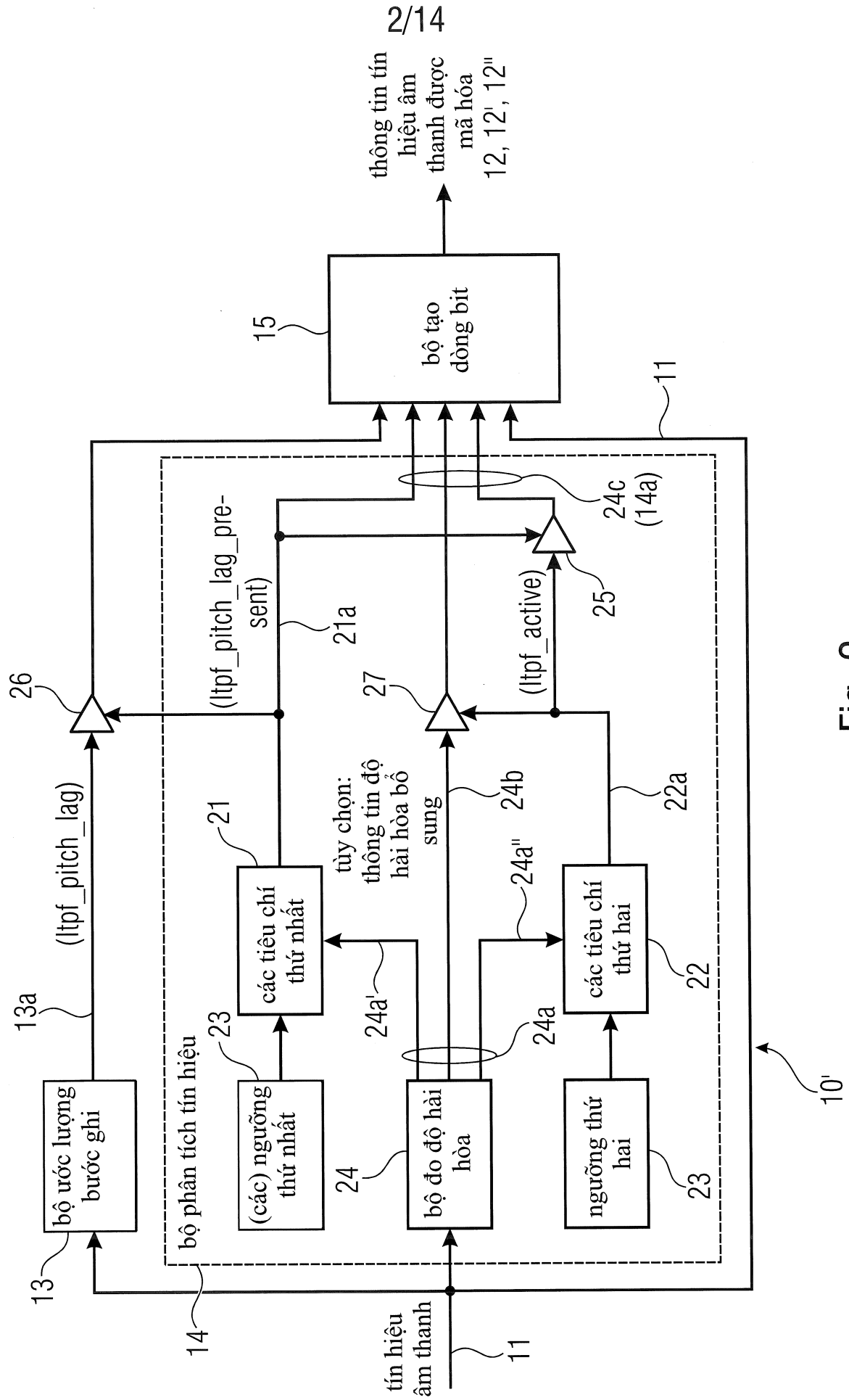


Fig. 2

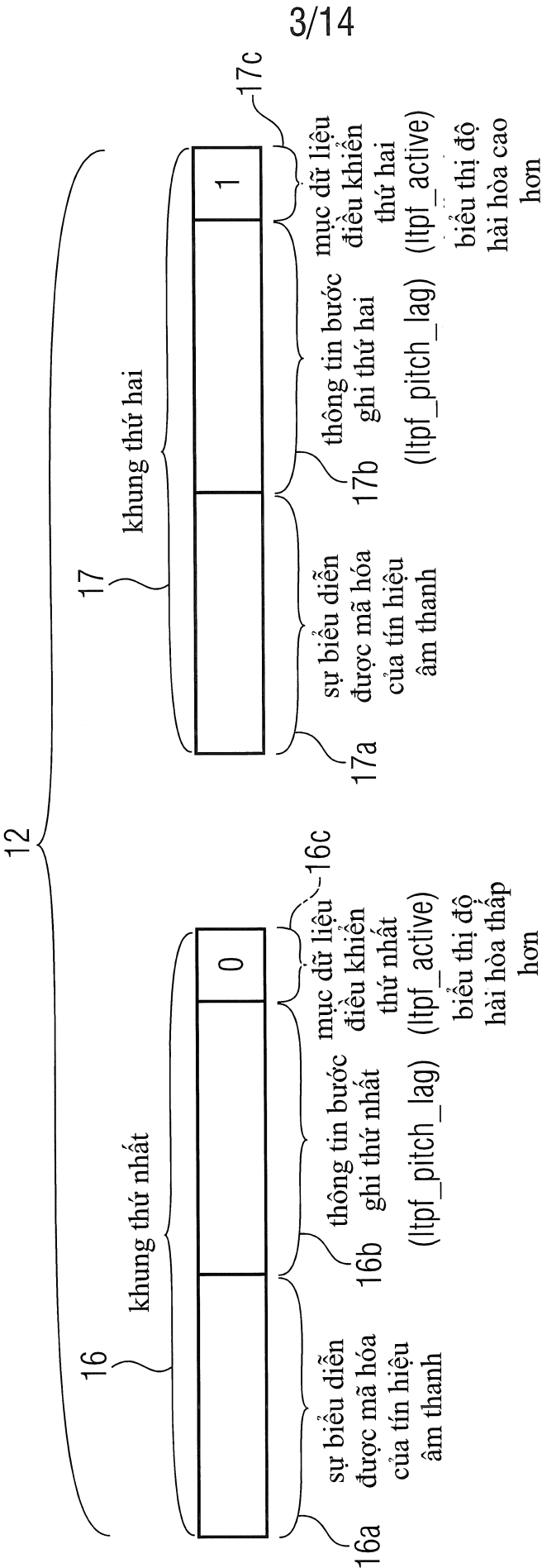


Fig. 3

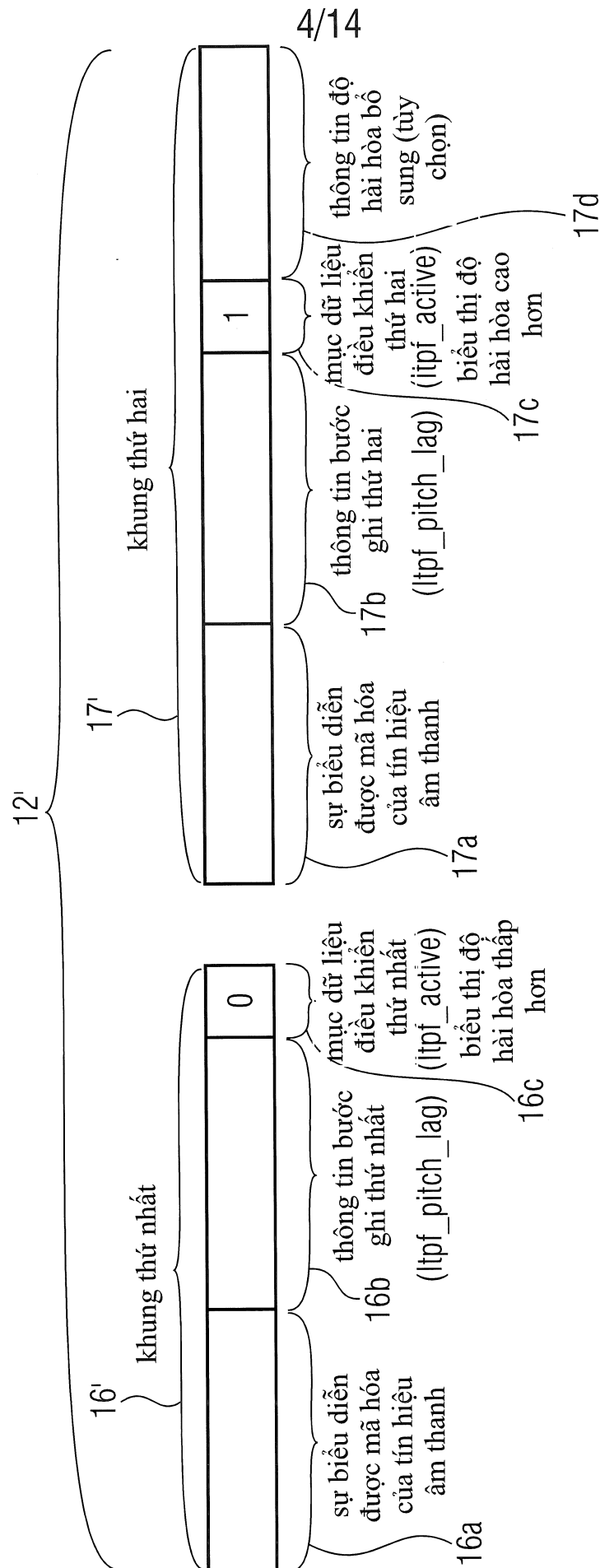


Fig. 4

6/14

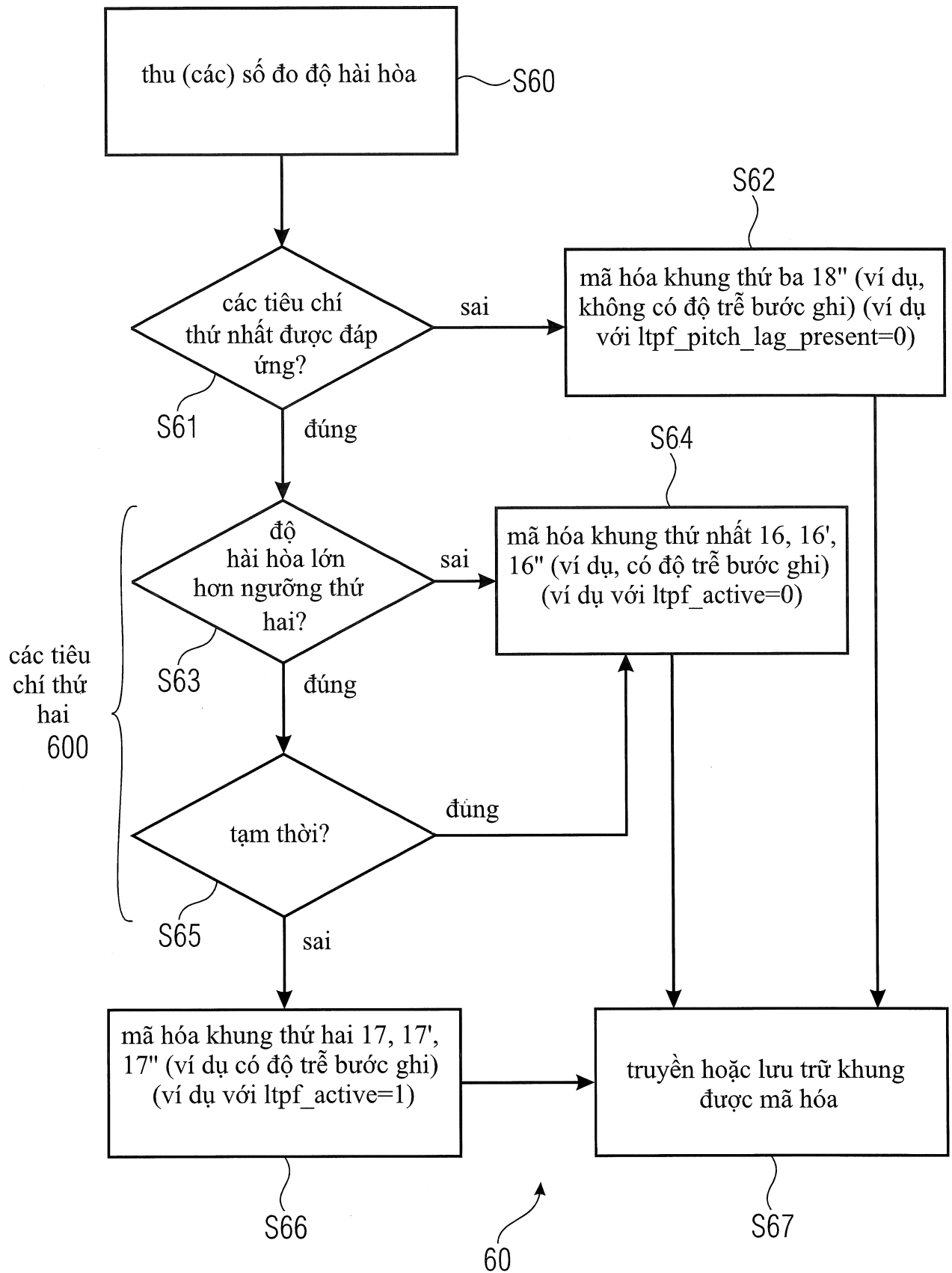


Fig. 6a

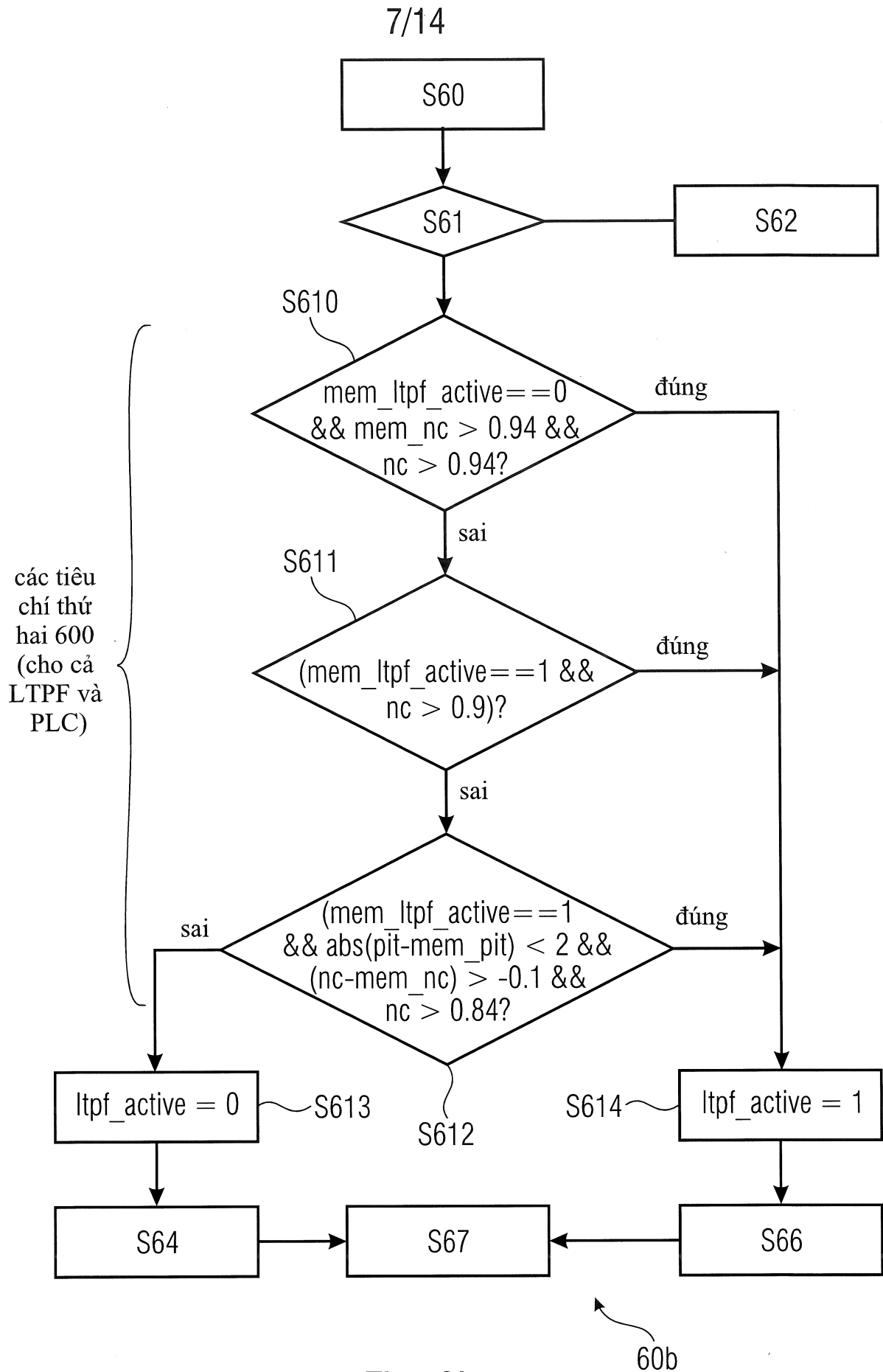


Fig. 6b

8/14

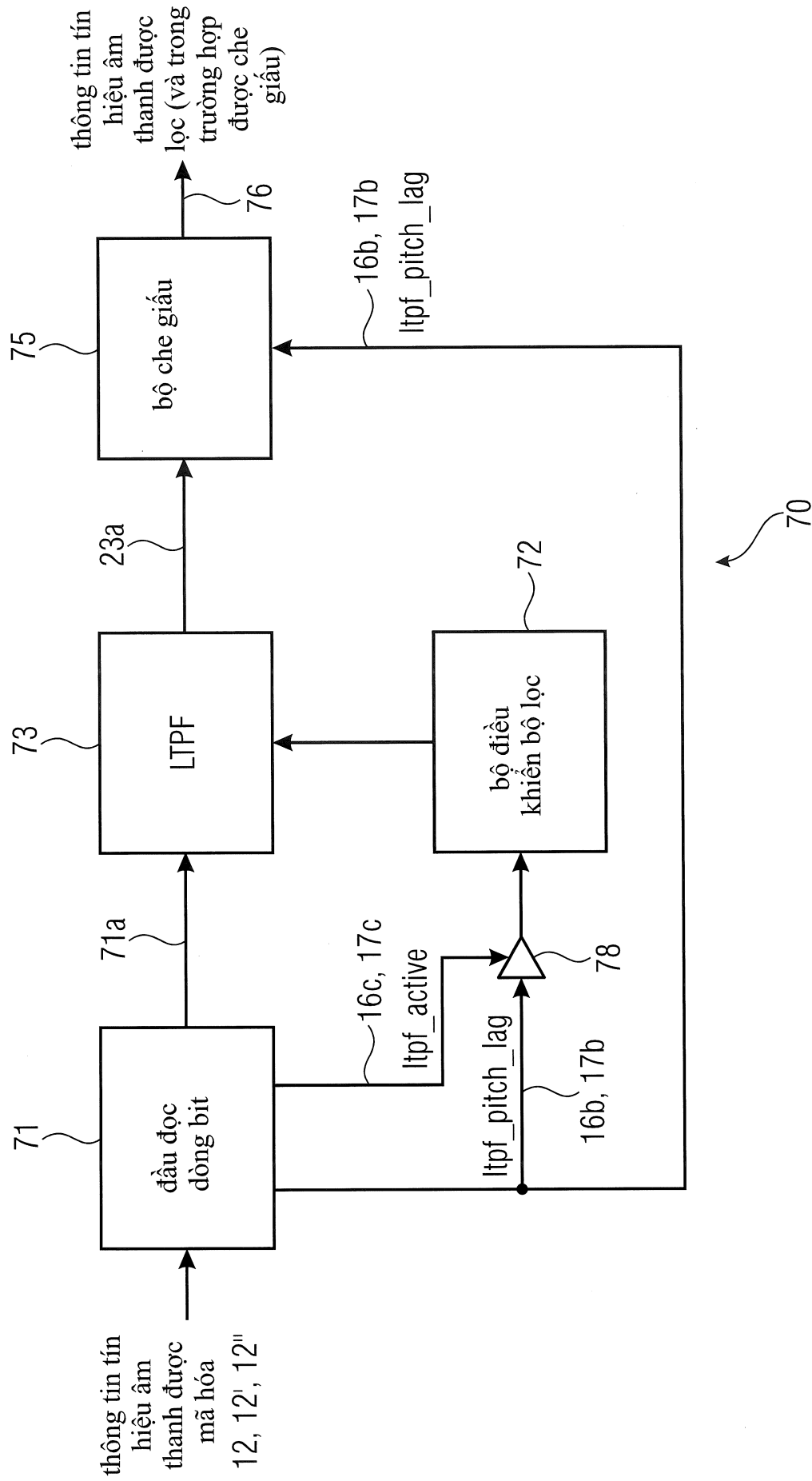


Fig. 7

9/14

	cú pháp	số bit	mnemonic
16a, 17a, 18a	ltpf data() {		
18e	ltpf_pitch_lag_present;	1	uimsbf
16c, 17c	if (ltpf_pitch_lag_present) {		
16b, 17b	ltpf_active;	1	uimsbf
	ltpf_pitch_lag;	9	uimsbf
	}		
	}		

Fig. 8a

	cú pháp	số bit	mnemonic
16a, 17a, 18a	ltpf data() {		
18e	ltpf_pitch_lag_present;	1	uimsbf
16c, 17c	if (ltpf_pitch_lag_present) {		
16b, 17b	ltpf_active;	1	uimsbf
	ltpf_pitch_lag;	9	uimsbf
17d	if (ltpf_active) {		
	ltpf_gain;	2	uimsbf
	}		
	}		
	}		

Fig. 8b

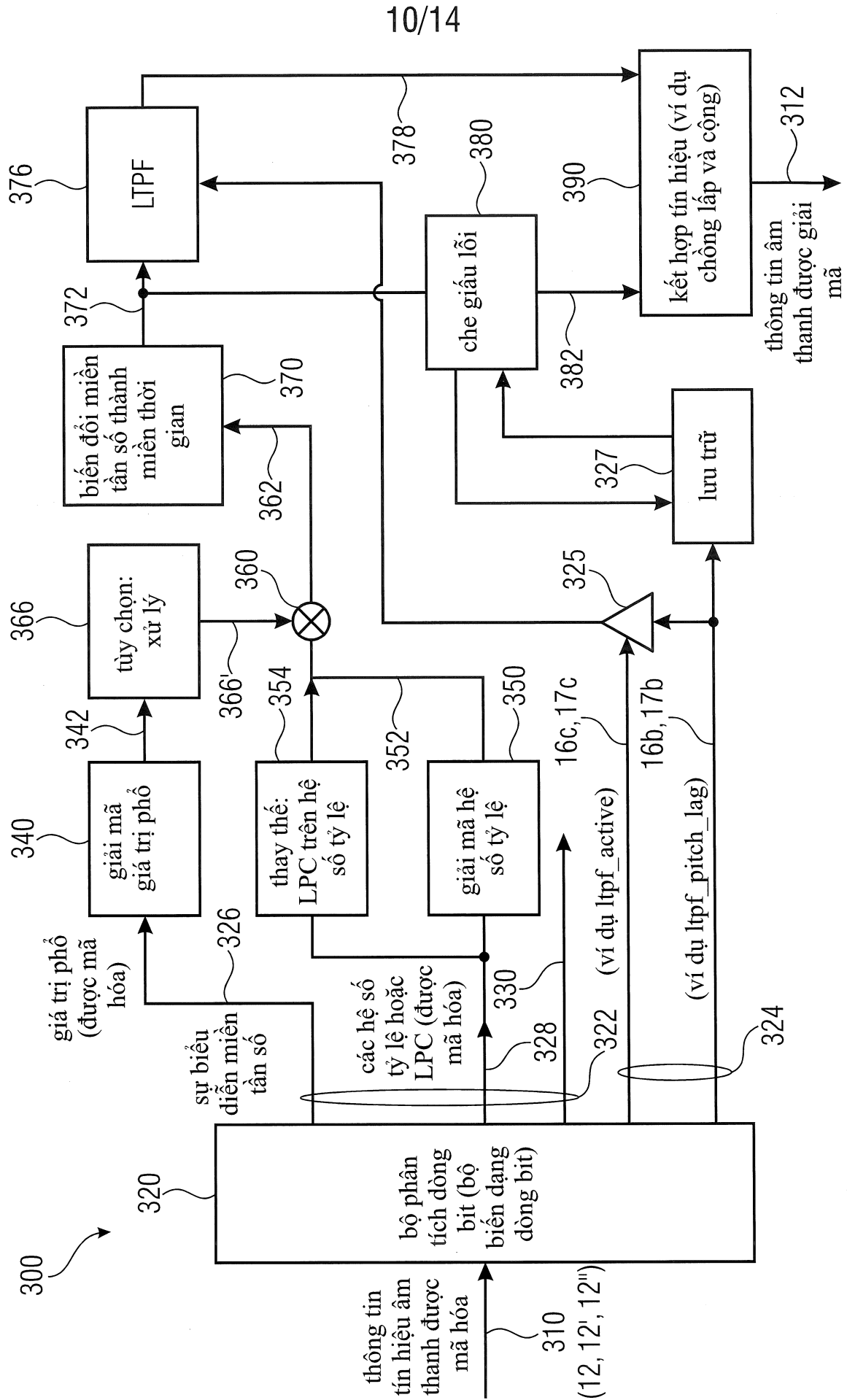


Fig. 9

11/14

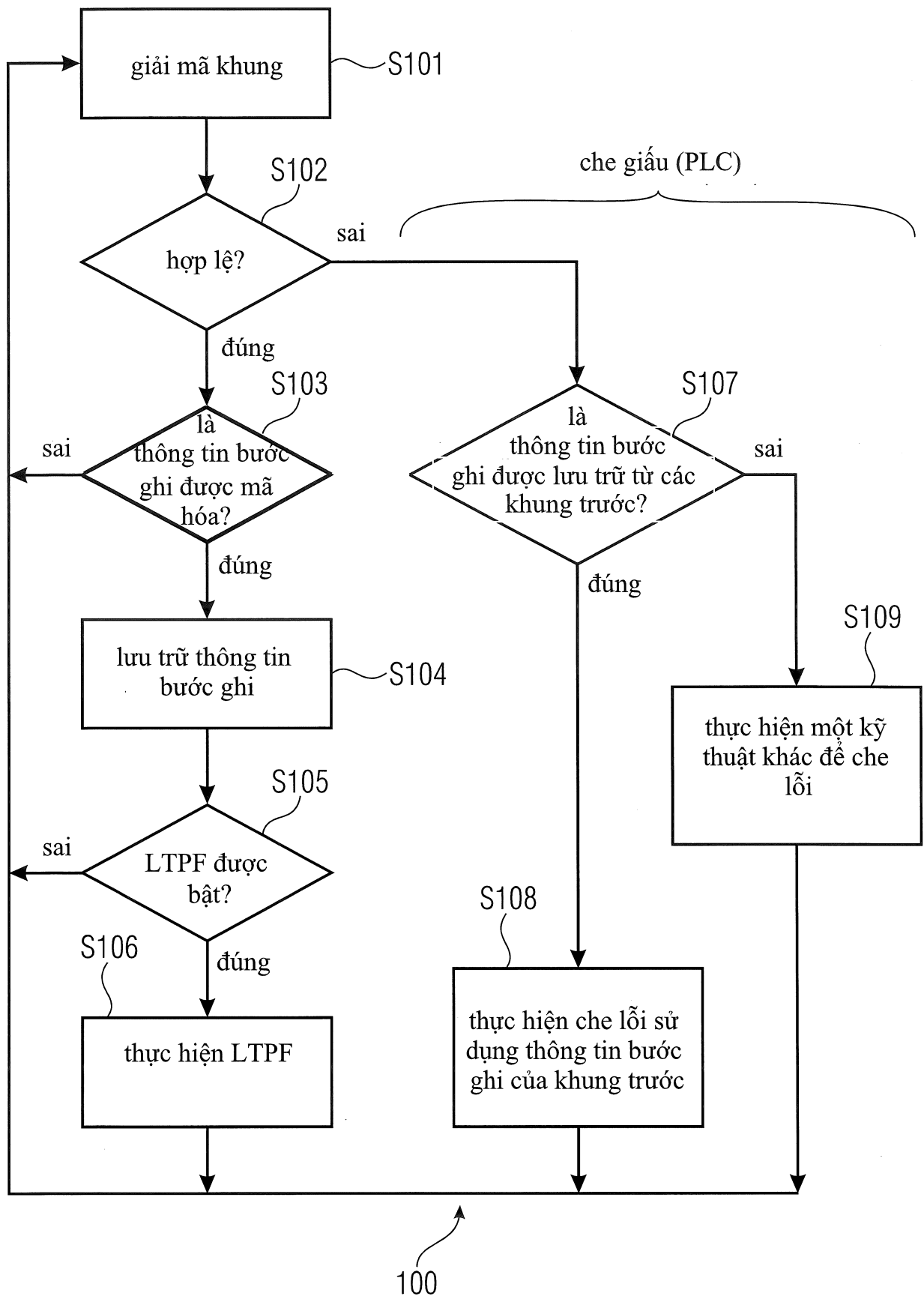


Fig. 10

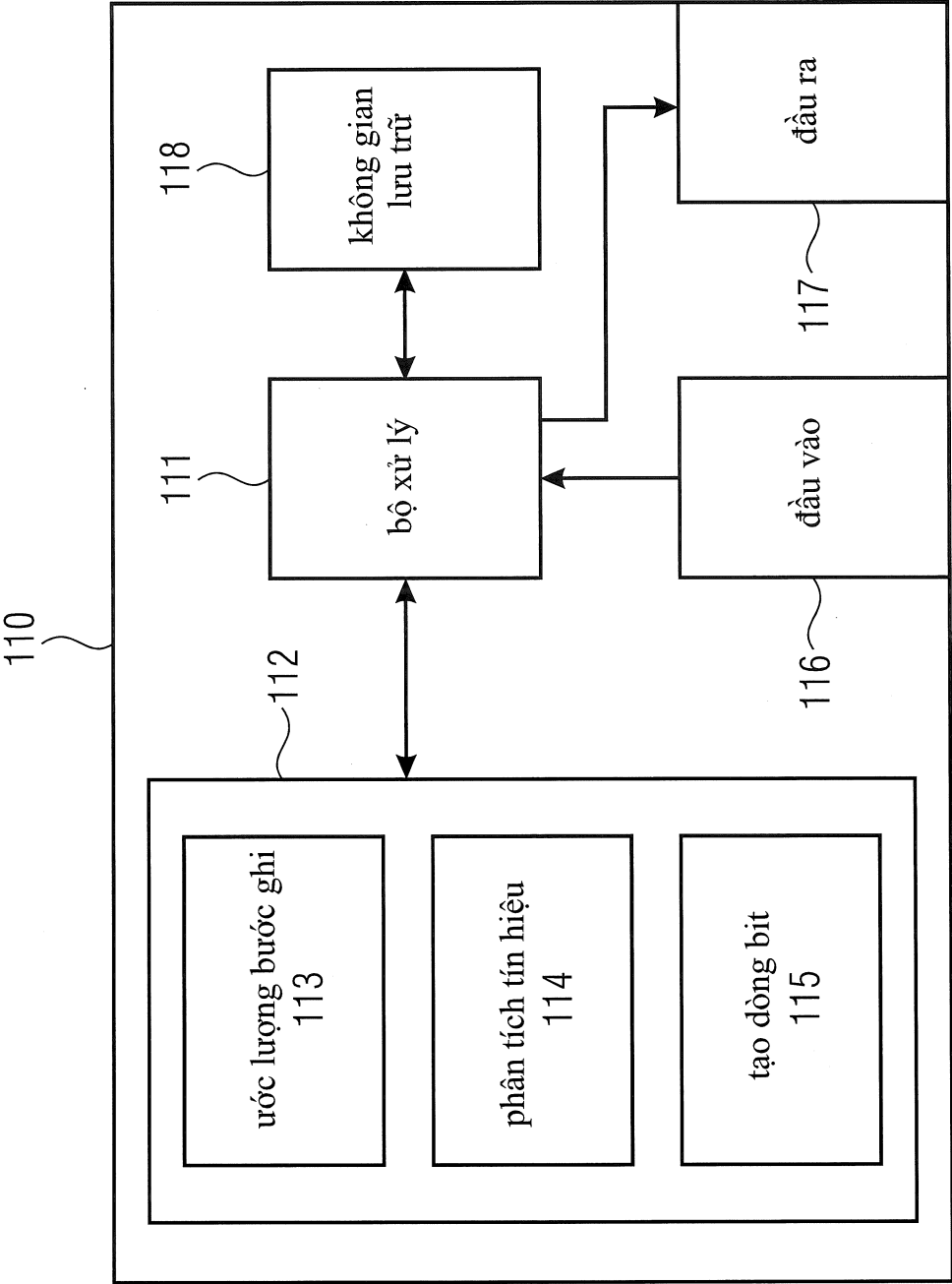


Fig. 11

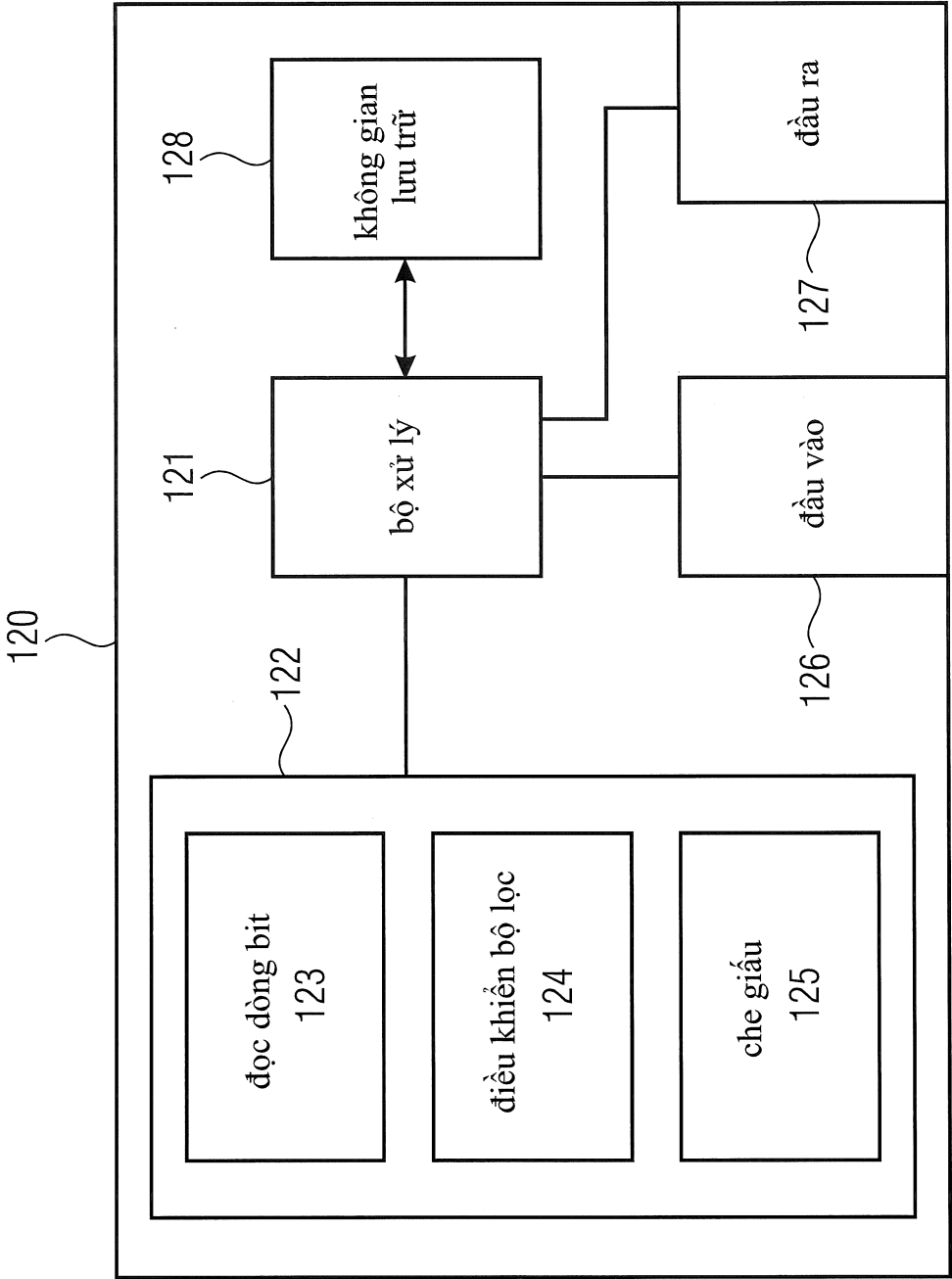


Fig. 12

14/14

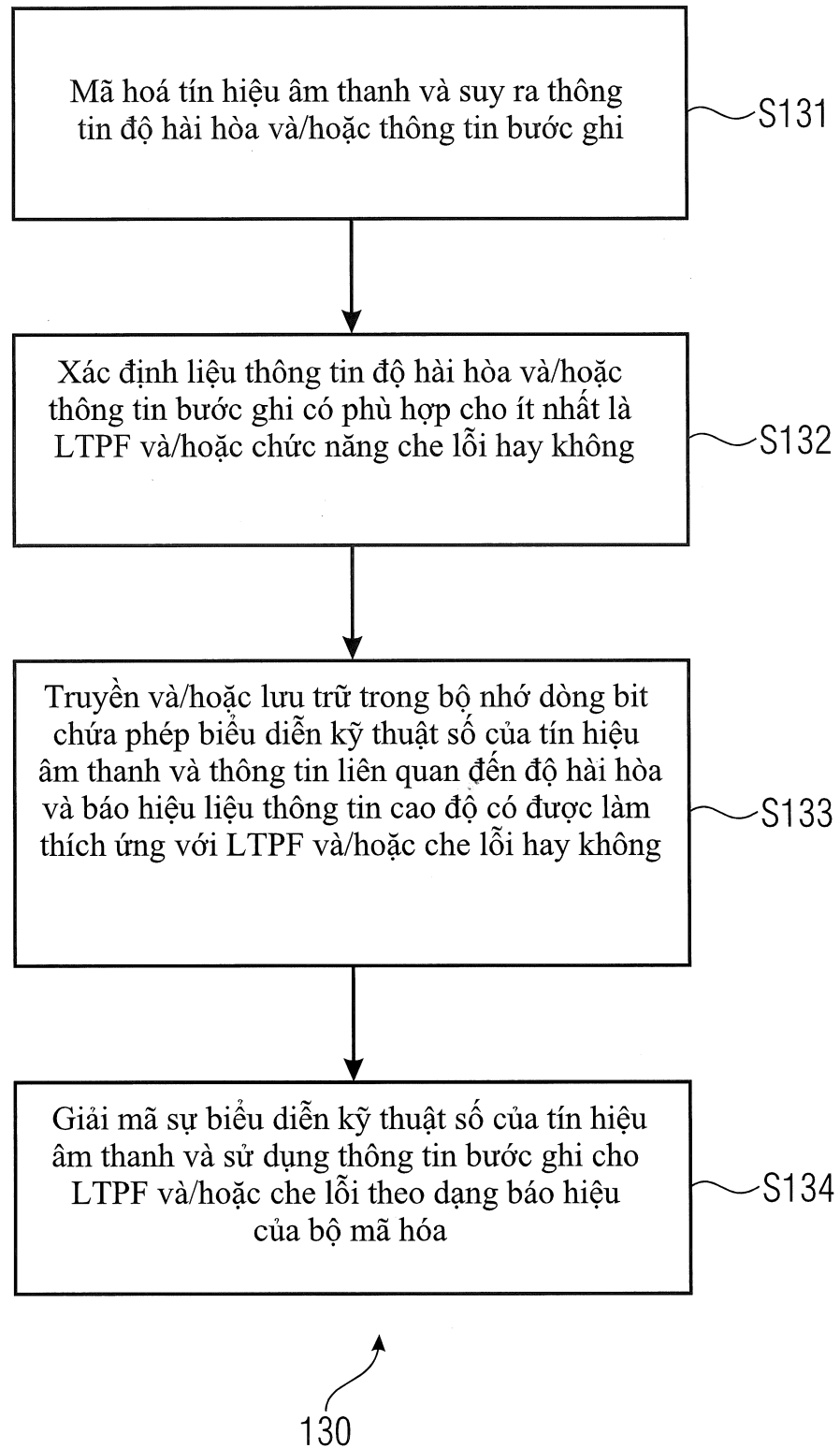


Fig. 13