



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

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

(11)



1-0025909

(51)⁷ G10L 19/008

(13) B

(21) 1-2016-00587

(22) 16/07/2014

(86) PCT/EP2014/065283 16/07/2014

(87) WO 2015/010996 29/01/2015

(30) EP13177365 22/07/2013 EP; EP13177367 22/07/2013 EP; EP13177378 22/07/2013 EP; EP13189279 18/10/2013 EP

(45) 26/10/2020 391

(43) 25/05/2016 338A

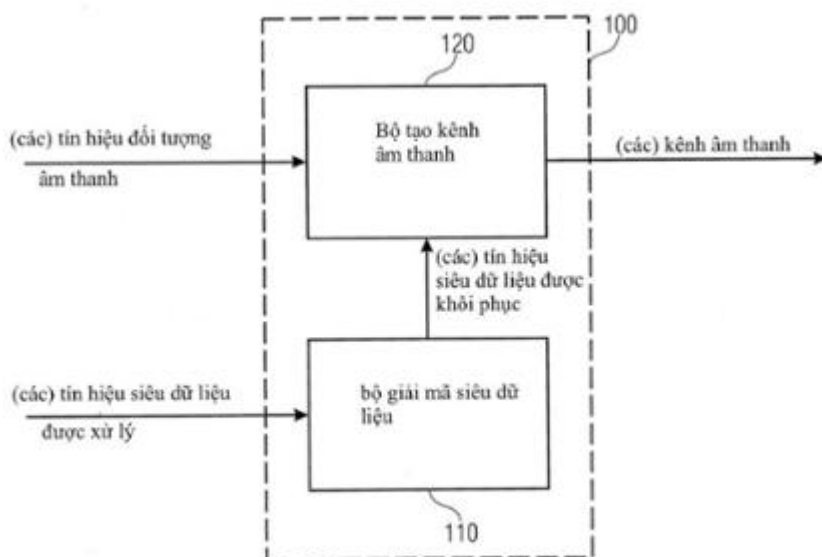
(73) Fraunhofer-Gesellschaft zur Foerderung der angewandten Forschung e.V. (DE)
Hansastrasse 27c, 80686 Muenchen, Germany

(72) BORSS, Christian (DE); ERTEL, Christian (DE); HILPERT, Johannes (DE).

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

(54) THIẾT BỊ VÀ PHƯƠNG PHÁP TẠO RA MỘT HOẶC NHIỀU KÊNH ÂM THANH, THIẾT BỊ VÀ PHƯƠNG PHÁP TẠO RA THÔNG TIN ÂM THANH ĐƯỢC MÃ HÓA, THIẾT BỊ GIẢI MÃ DỮ LIỆU ÂM THANH ĐƯỢC MÃ HÓA VÀ THIẾT BỊ MÃ HÓA DỮ LIỆU ĐÀU VÀO ÂM THANH

(57) Sáng chế đề xuất thiết bị và phương pháp tạo ra một hoặc nhiều kênh âm thanh, thiết bị và phương pháp tạo ra thông tin âm thanh được mã hóa, thiết bị giải mã dữ liệu âm thanh được mã hóa và phương pháp mã hóa dữ liệu đầu vào âm thanh. Thiết bị tạo ra một hoặc nhiều kênh âm thanh (100) bao gồm bộ giải mã siêu dữ liệu (110) để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) từ một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$) phụ thuộc vào tín hiệu điều khiển (b), trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh, trong đó bộ giải mã siêu dữ liệu (110) được tạo cấu hình để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) bằng cách xác định nhiều mẫu siêu dữ liệu được khôi phục ($x_i'(n), \dots, x_N'(n)$) cho mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$). Ngoài ra, thiết bị bao gồm bộ tạo kênh âm thanh (120) để tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$). Bộ giải mã siêu dữ liệu (110) được tạo cấu hình để nhận nhiều mẫu siêu dữ liệu được xử lý ($z_1(n), \dots, z_N(n)$) của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$). Ngoài ra, bộ giải mã siêu dữ liệu (110) được tạo cấu hình để nhận tín hiệu điều khiển (b). Hơn nữa bộ giải mã siêu dữ liệu (110) được tạo cấu hình để xác định từng mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) trong số nhiều mẫu siêu dữ liệu được khôi phục ($x_i'(1), \dots, x_i'(n-1), x_i'(n)$) của mỗi tín hiệu siêu dữ liệu được khôi phục (x_i') trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) đã nêu là tổng của một trong số các mẫu siêu dữ liệu được xử lý ($z_i'(n)$) của một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (z_i) và mẫu siêu dữ liệu được khôi phục đã được tạo ra khác ($x_i'(n-1)$) của tín hiệu siêu dữ liệu được khôi phục đã nêu (x_i'), và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) đã nêu là một mẫu ($z_i(n)$) đã nêu trong số các mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n)$) của một tín hiệu (z_i) đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$). Ngoài ra thiết bị tạo ra thông tin âm thanh được mã hóa (250) được đề xuất.



Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến việc mã hóa/giải mã âm thanh, cụ thể đề cập đến mã hóa âm thanh không gian và mã hóa đối tượng âm thanh không gian, và cụ thể hơn, đề cập đến thiết bị và phương pháp mã hóa siêu dữ liệu đối tượng có hiệu quả.

Tình trạng kỹ thuật của sáng chế

Công cụ mã hóa âm thanh không gian đã được biết đến rộng rãi trong tình trạng kỹ thuật, ví dụ, được tiêu chuẩn hóa trong tiêu chuẩn MPEG-âm thanh nổi. Việc mã hóa âm thanh không gian bắt đầu từ các kênh đầu vào ban đầu, ví dụ, năm hoặc bảy kênh, mà được nhận biết bằng sự sắp đặt trong thiết lập tái tạo, tức là, kênh bên trái, kênh ở giữa, kênh bên phải, kênh âm thanh nổi bên trái, kênh âm thanh nổi bên phải và kênh nâng cao tần số thấp (low frequency enhancement - LFE). Bộ mã hóa âm thanh không gian thông thường có thể rút ra một hoặc nhiều kênh trộn giảm từ các kênh ban đầu và, ngoài ra, có thể được rút dữ liệu tham số liên quan đến các ký hiệu điều khiển không gian như các chênh lệch mức liên kênh trong các giá trị kết hợp kênh, các chênh lệch pha liên kênh, các chênh lệch thời gian liên kênh, v.v. Một hoặc nhiều kênh trộn giảm được truyền dẫn cho nhau với thông tin phụ theo tham số biểu thị các ký hiệu điều khiển không gian cho bộ giải mã âm thanh trong không gian để giải mã các kênh trộn giảm và dữ liệu tham số được kết hợp để cuối cùng thu được các kênh đầu ra mà là phiên bản được làm gần đúng của các kênh đầu vào ban đầu. Sự sắp đặt các kênh trong sự thiết lập đầu ra có thể được cố định một cách thông thường, ví dụ, định dạng 5.1, định dạng 7.1, v.v.

Các định dạng âm thanh trên cơ sở kênh đã được sử dụng rộng rãi để lưu trữ hoặc truyền dẫn nội dung âm thanh đa kênh mà tại đó mỗi kênh liên quan đến loa phát thanh cụ thể tại vị trí đã định. Sự tái tạo trung thành của các loại định dạng này yêu cầu thiết lập loa phát thanh mà tại đó các loa được đặt tại các vị trí như nhau như các loa mà đã được sử dụng trong quá trình tạo ra các tín hiệu âm thanh. Trong khi đó việc tăng số

lượng loa phát thanh cải thiện sự tái tạo của các ngữ cảnh âm thanh 3D nhúng chìm thực sự, càng trở nên khó khăn hơn để thực hiện yêu cầu này - đặc biệt là trong một môi trường trong nhà giống như phòng khách.

Sự cần thiết của việc phải thiết lập loa phát thanh cụ thể có thể được khắc phục bằng cách tiếp cận dựa trên cơ sở đối tượng mà các tín hiệu loa phát thanh được kết xuất đặc biệt cho các thiết lập phát lại.

Ví dụ, các công cụ mã hóa đối tượng âm thanh không gian được biết đến rộng rãi trong lĩnh vực kỹ thuật và được tiêu chuẩn hóa, trong tiêu chuẩn mã hóa đối tượng âm thanh trong không gian (spatial audio object coding - SAOC) MPEG. Trái với sự mã hóa âm thanh không gian bắt đầu từ các kênh ban đầu, sự mã hóa đối tượng âm thanh không gian bắt đầu từ các đối tượng âm thanh mà không tự động dành riêng cho sự thiết lập tái tạo kết xuất nhất định. Thay vào đó, sự sắp đặt các đối tượng âm thanh trong ngữ cảnh tái tạo là linh hoạt và có thể được xác định bởi người dùng bằng cách nhập thông tin kết xuất nhất định vào bộ giải mã mã hóa đối tượng âm thanh không gian. Bằng cách bổ sung hoặc thay thế, thông tin kết xuất, tức là, thông tin mà tại vị trí trong thiết lập tái tạo đối tượng âm thanh nhất định được đặt một cách thông thường trên thời gian có thể được truyền dẫn như thông tin phụ bổ sung hoặc siêu dữ liệu. Để thu được sự nén dữ liệu nhất định, số lượng của các đối tượng âm thanh được mã hóa bởi bộ mã hóa SAOC mà tính toán, từ các đối tượng đầu vào, một hoặc nhiều kênh vận chuyển bằng cách trộn giảm các đối tượng phù hợp với thông tin trộn giảm nhất định. Hơn nữa, bộ mã hóa SAOC tính toán thông tin phụ theo tham số biểu diễn các ký hiệu điều khiển liên-đối tượng như các chênh lệch mức đối tượng (object level differences-OLD), các giá trị kết hợp đối tượng, v.v. Như trong Mã hóa âm thanh không gian (Spatial Audio Coding-SAC), dữ liệu theo tham số liên đối tượng được tính toán, tức là, cho các ô thời gian/tần số riêng lẻ của tín hiệu âm thanh bao gồm, ví dụ, các mẫu 1024 hoặc 2048, 24, 32 hoặc 64, v.v., các băng tần được xem xét sao cho, cuối cùng, dữ liệu theo tham số tồn tại với mỗi khung và mỗi băng tần. Ví dụ, khi đoạn âm thanh có 20 khung và khi mỗi khung được chia thành 32 băng tần, sau đó số ô thời gian/tần số là 640.

Trong cách tiếp cận trên cơ sở đối tượng, miền âm được mô tả bởi các đối tượng âm thanh rời rạc. Điều này đòi hỏi siêu dữ liệu đối tượng mà mô tả trong số vị trí thời gian khác nhau khác của mỗi nguồn âm trong không gian 3D.

Khái niệm mã hóa siêu dữ liệu thứ nhất thuộc tình trạng kỹ thuật là định dạng liên kênh mô tả âm không gian (spatial sound description interchange format - SpatDIF), định dạng mô tả ngữ cảnh âm thanh mà vẫn đang được phát triển [1]. Được thiết kế như định dạng liên kênh cho các ngữ cảnh âm trên cơ sở đối tượng và không cung cấp phương pháp nén bất kỳ cho các quỹ đạo đối tượng. SpatDIF sử dụng định dạng Kiểm soát âm mở (Open Sound Control-OSC) trên cơ sở kí tự cho kết cấu siêu dữ liệu đối tượng [2]. Tuy nhiên, sự biểu diễn trên cơ sở kí tự đơn giản không là sự lựa chọn cho sự truyền dẫn được nén của các quỹ đạo đối tượng.

Khái niệm siêu dữ liệu khác thuộc tình trạng kỹ thuật là Định dạng Mô tả Ngữ cảnh Âm thanh (Audio Scene Description Format-ASDF) [3], giải pháp trên cơ sở kí tự mà có bất lợi tương tự. Dữ liệu được kết cấu bằng việc mở rộng Ngôn ngữ Tích hợp Đa phương tiện Đồng bộ hóa (Synchronized Multimedia Integration Language-SMIL) mà là thiết lập phụ của Ngôn ngữ Đánh dấu Có thể mở rộng (Extensible Markup Language-XML) [4,5].

Khái niệm siêu dữ liệu nữa thuộc tình trạng kỹ thuật là định dạng nhịp đôi âm thanh cho các ngữ cảnh (AudioBIFS), định dạng nhịp đôi mà là một phần của đặc điểm kỹ thuật MPEG-4 [6,7]. Nó liên quan gần gũi với Ngôn ngữ Mô hình hóa thực tế ảo (Virtual Reality Modeling Language-VRML) trên cơ sở XML mà được phát triển cho mô tả các ngữ cảnh 3D âm thanh ảo và các ứng dụng thực tế ảo tương tác [8]. Đặc điểm kỹ thuật AudioBIFS phức hợp sử dụng các biểu đồ ngữ cảnh để định tuyến cụ thể của các chuyển động của đối tượng. Nhược điểm chính của AudioBIFS là không thiết kế cho vận hành thời gian thực mà truy cập ngẫu nhiên và độ trễ hệ thống bị giới hạn vào dòng dữ liệu là một yêu cầu. Hơn nữa, việc mã hóa các vị trí đối tượng không khai thác đặc tính vị trí bị giới hạn của người nghe là con người. Với vị trí người nghe được cố định trong ngữ cảnh âm thanh ảo, dữ liệu đối tượng có thể được lượng tử hóa với số lượng bit thấp hơn nhiều [9]. Do đó, việc mã hóa siêu dữ liệu đối tượng mà được áp dụng trong AudioBIFS là không có hiệu quả đối với việc nén dữ liệu.

Do đó được đánh giá cao nếu các khái niệm mã hóa siêu dữ liệu đối tượng có hiệu quả nếu được cải thiện được đề xuất.

Tài liệu tham khảo

- [1] Peters, N., Lossius, T. and Schacher J. C., "SpatDIF: Principles, Specification, and Examples", 9th Sound and Music Computing Conference, Copenhagen, Denmark, Jul. 2012.
- [2] Wright, M., Freed, A., "Open Sound Control: A New Protocol for Communicating with Sound Sy_Nthesizers", International Computer Music Conference, Thessaloniki, Greece, 1997.
- [3] Matthias Geier, Jens Ahrens, and Sascha Spors. (2010), "Object-based audio reproduction and the audio scene description format", Org. Sound, Vol. 15, No. 3, pp. 219-227, December 2010.
- [4] W3C, "Sy_Nchronized Multimedia Integration Language (SMIL 3.0)", Dec. 2008.
- [5] W3C, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", Nov. 2008.
- [6] MPEG, "ISO/IEC International Standard 14496-3 - Coding of audio-visual objects, Part 3 Audio", 2009.
- [7] Schmidt, J.; Schroeder, E. F. (2004), "New and Advanced Features for Audio Presentation in the MPEG-4 Standard", 116th AES Convention, Berlin, Germany, May 2004
- [8] Web3D, "International Standard ISO/IEC 14772-1:1997 - The Virtual Reality Modeling Language (VRML), Part 1: Functional specification and UTF-8 encoding", 1997.
- [9] Sporer, T. (2012), "Codierung räumlicher Audiosignale mit leichtgewichtigen Audio-Objekten", Proc. Annual Meeting of the German Audiological Society (DGA), Erlangen, Germany, Mar. 2012.

[10] Cutler, C. C. (1950), "Differential Quantization of Communication Signals", US Patent US2605361, Jul. 1952.

[11] Ville Pulkki, "Virtual Sound Source Positioning Using Vector Base Amplitude Panning", J. Audio Eng. Soc., Volume 45, Issue 6, pp. 456-466, June 1997.

Bản chất kỹ thuật của sáng chế

Mục đích của sáng chế là đề xuất các khái niệm được cải thiện để mã hóa siêu dữ liệu đối tượng. Mục đích của sáng chế được giải quyết bằng thiết bị theo điểm 1, bằng phương pháp mã hóa theo điểm 6, bằng hệ thống theo điểm 12, bằng phương pháp theo điểm 13, bằng phương pháp theo điểm 14 và bằng chương trình máy tính theo điểm 15.

Thiết bị tạo ra một hoặc nhiều kênh âm thanh được đề xuất. Thiết bị bao gồm bộ giải mã siêu dữ liệu để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) từ một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$) phụ thuộc vào tín hiệu điều khiển (b), trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh, trong đó bộ giải mã siêu dữ liệu được tạo cấu hình để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) bằng cách xác định nhiều mẫu siêu dữ liệu được khôi phục ($x_1'(n), \dots, x_N'(n)$) cho mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$). Ngoài ra, thiết bị bao gồm bộ tạo kênh âm thanh để tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$). Bộ giải mã siêu dữ liệu được tạo cấu hình để nhận nhiều mẫu siêu dữ liệu được xử lý ($z_1(n), \dots, z_N(n)$) của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$). Ngoài ra, bộ giải mã siêu dữ liệu được tạo cấu hình để nhận tín hiệu điều khiển (b). Hơn nữa, bộ giải mã siêu dữ liệu được tạo cấu hình để xác định từng mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) trong số nhiều mẫu siêu dữ liệu được khôi phục ($x_i'(1), \dots, x_i'(n-1), x_i'(n)$) của mỗi tín hiệu siêu dữ liệu được khôi phục trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($x_i'(n)$)

đã nêu là tổng của một trong số các mẫu siêu dữ liệu được xử lý ($z_i(n)$) của một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (z_i) và mẫu siêu dữ liệu được khôi phục đã được tạo ra khác ($x_i'(n-1)$) của tín hiệu siêu dữ liệu được khôi phục đã nêu (x_i'), và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) đã nêu là một mẫu ($z_i(n)$) đã nêu trong số các mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n)$) của một tín hiệu (z_i) đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$).

Ngoài ra, thiết bị tạo ra thông tin âm thanh được mã hóa bao gồm một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý được đề xuất. Thiết bị bao gồm bộ mã hóa siêu dữ liệu để nhận một hoặc nhiều tín hiệu siêu dữ liệu ban đầu và để xác định một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu bao gồm nhiều mẫu siêu dữ liệu ban đầu, trong đó các mẫu siêu dữ liệu ban đầu của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh.

Ngoài ra, thiết bị bao gồm bộ mã hóa âm thanh để mã hóa một hoặc nhiều tín hiệu đối tượng âm thanh để thu được một hoặc nhiều tín hiệu âm thanh được mã hóa.

Bộ giải mã siêu dữ liệu được tạo cấu hình để xác định từng mẫu siêu dữ liệu được xử lý ($z_i(n)$) trong số nhiều mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n-1), z_i(n)$) của mỗi tín hiệu siêu dữ liệu được xử lý (z_i) trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($z_i(n)$) đã nêu biểu thị chênh lệch hoặc chênh lệch được lượng tử hóa giữa một trong số nhiều mẫu siêu dữ liệu ban đầu ($x_i(n)$) và mẫu siêu dữ liệu được xử lý đã được tạo ra khác của tín hiệu siêu dữ liệu được xử lý (z_i) đã nêu, và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được xử lý ($z_i(n)$) đã nêu là một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$) của một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (x_i), hoặc là sự biểu diễn

được lượng tử hóa ($q_i(n)$) của một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$).

Theo các phương án, các khái niệm nén dữ liệu cho siêu dữ liệu đối tượng được đề xuất, mà cơ cấu nén có hiệu quả đạt được cho các kênh truyền dẫn với tốc độ dữ liệu được giới hạn. Không bỏ sung độ trễ được đưa vào lần lượt bởi bộ mã hóa và bộ giải mã. Ngoài ra, tốc độ nén tốt cho những thay đổi phương vị thuận túy, ví dụ, sự quay máy quay, đạt được. Hơn nữa, các khái niệm được đề xuất hỗ trợ cho những quỹ đạo không liên tục, ví dụ, nhảy vị trí. Ngoài ra, phức hợp giải mã thấp được thực hiện. Hơn nữa, truy nhập ngẫu nhiên với thời gian tái khởi tạo bị giới hạn đạt được.

Hơn nữa, phương pháp tạo ra một hoặc nhiều kênh âm thanh được đề xuất. Phương pháp gồm có:

- Tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) từ một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$) phụ thuộc vào tín hiệu điều khiển (b), trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh trong số một hoặc nhiều tín hiệu đối tượng âm thanh, trong đó tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) được điều khiển bằng cách xác định nhiều mẫu siêu dữ liệu được khôi phục ($x_1'(n), \dots, x_N'(n)$) cho mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$). Và:

- Tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$).

Tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) được điều khiển bằng cách nhận nhiều mẫu siêu dữ liệu được xử lý ($z_1(n), \dots, z_N(n)$) của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), bằng cách nhận tín hiệu điều khiển (b), và bằng cách xác định từng mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) trong số nhiều mẫu siêu dữ liệu được khôi phục ($x_i'(1), \dots, x_i'(n-1), x_i'(n)$) của mỗi tín hiệu siêu dữ liệu được khôi phục (x_i') trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) đã nêu là tổng của một

trong số các mẫu siêu dữ liệu được xử lý ($z_i(n)$) của một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (z_i) và mẫu siêu dữ liệu được khôi phục đã được tạo ra khác ($x_i'(n-1)$) của tín hiệu siêu dữ liệu được khôi phục đã nêu (x_i'), và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) đã nêu là một mẫu ($z_i(n)$) đã nêu trong số các mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n)$) của một tín hiệu (z_i) đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$).

Ngoài ra, phương pháp tạo ra thông tin âm thanh được mã hóa bao gồm một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý được đề xuất. Phương pháp gồm có:

- Nhận một hoặc nhiều tín hiệu siêu dữ liệu ban đầu.
- Xác định một hoặc nhiều tín hiệu siêu dữ liệu được xử lý. Và:
- Mã hóa một hoặc nhiều tín hiệu đối tượng âm thanh để thu được một hoặc nhiều tín hiệu âm thanh được mã hóa.

Mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu bao gồm nhiều mẫu siêu dữ liệu ban đầu, trong đó các mẫu siêu dữ liệu ban đầu của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh trong số một hoặc nhiều tín hiệu đối tượng âm thanh. Xác định một hoặc nhiều tín hiệu siêu dữ liệu được xử lý bao gồm xác định mỗi mẫu siêu dữ liệu được xử lý ($z_i(n)$) trong số nhiều mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n-1), z_i(n)$) của mỗi tín hiệu siêu dữ liệu được xử lý (z_i) trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($z_i(n)$) đã nêu biểu thị chênh lệch hoặc chênh lệch được lượng tử hóa giữa một trong số nhiều mẫu siêu dữ liệu ban đầu ($x_i(n)$) của một trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu (x_i) và mẫu siêu dữ liệu được xử lý đã được tạo ra khác của tín hiệu siêu dữ liệu được xử lý (z_i) đã nêu, và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được xử lý ($z_i(n)$) đã nêu là một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$) của một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (x_i), hoặc là sự biểu diễn được

lượng tử hóa ($q_i(n)$) của một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$).

Hơn nữa, chương trình máy tính để thực hiện phương pháp được mô tả ở trên khi được chạy trên máy tính hoặc bộ xử lý tín hiệu được đề xuất.

Mô tả vắn tắt các hình vẽ

Sau đây, các phương án của sáng chế được mô tả chi tiết hơn với sự tham chiếu đến các hình vẽ, trong đó:

Fig.1 minh họa thiết bị tạo ra một hoặc nhiều kênh âm thanh theo phương án,

Fig.2 minh họa thiết bị tạo ra thông tin âm thanh được mã hóa theo phương án,

Fig.3 minh họa hệ thống theo phương án,

Fig.4 minh họa vị trí của đối tượng âm thanh trong không gian ba chiều từ gốc được biểu thị bởi phương vị, độ cao và bán kính,

Fig.5 minh họa các vị trí của các đối tượng âm thanh và thiết lập loa phát thanh được giả định bởi bộ tạo kênh âm thanh,

Fig.6 minh họa bộ mã hóa Điều Biên Mã Xung Vi Phân,

Fig.7 minh họa bộ giải mã Điều Biên Mã Xung Vi Phân,

Fig.8a minh họa bộ mã hóa siêu dữ liệu theo phương án,

Fig.8b minh họa bộ mã hóa siêu dữ liệu theo phương án khác,

Fig.9a minh họa bộ giải mã siêu dữ liệu theo phương án,

Fig.9b minh họa khối con của bộ giải mã siêu dữ liệu theo phương án,

Fig.10 minh họa phương án thứ nhất của bộ mã hóa âm thanh 3D,

Fig.11 minh họa phương án thứ nhất của bộ giải mã âm thanh 3D,

Fig.12 minh họa phương án thứ hai của bộ mã hóa âm thanh 3D,

Fig.13 minh họa phương án thứ hai của bộ giải mã âm thanh 3D,

Fig.14 minh họa phương án thứ ba của bộ mã hóa âm thanh 3D, và

Fig.15 minh họa phương án thứ ba của bộ giải mã âm thanh 3D.

Mô tả chi tiết sáng chế

Fig.2 minh họa thiết bị tạo ra thông tin âm thanh được mã hóa 250 bao gồm một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý theo phương án.

Thiết bị 250 bao gồm bộ mã hóa siêu dữ liệu 210 để nhận một hoặc nhiều tín hiệu siêu dữ liệu ban đầu và để xác định một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu bao gồm nhiều mẫu siêu dữ liệu ban đầu, trong đó các mẫu siêu dữ liệu ban đầu của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh.

Ngoài ra, thiết bị 250 bao gồm bộ mã hóa âm thanh 220 để mã hóa một hoặc nhiều tín hiệu đối tượng âm thanh để thu được một hoặc nhiều tín hiệu âm thanh được mã hóa.

Bộ mã hóa siêu dữ liệu 210 được tạo cấu hình để xác định từng mẫu siêu dữ liệu được xử lý ($z_i(n)$) trong số nhiều mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n-1), z_i(n)$) của mỗi tín hiệu siêu dữ liệu được xử lý (z_i) trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($z_i(n)$) đã nêu biểu thị chênh lệch hoặc chênh lệch được lượng tử hóa giữa một trong số nhiều mẫu siêu dữ liệu ban đầu ($x_i(n)$) của một trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu (x_i) và mẫu siêu dữ liệu được xử lý đã được tạo ra khác của tín hiệu siêu dữ liệu được xử lý (z_i) đã nêu, và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được xử lý ($z_i(n)$) đã nêu là một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$) của một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (x_i), hoặc là sự biểu diễn được lượng tử hóa ($q_i(n)$) của một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$).

Fig.1 minh họa thiết bị tạo ra một hoặc nhiều kênh 100 theo phương án.

Thiết bị 100 bao gồm bộ giải mã siêu dữ liệu 110 để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ từ một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $(z_1, \dots, z_N)$ phụ thuộc vào tín hiệu điều khiển (b), trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh, trong đó bộ giải mã siêu dữ liệu 110 được tạo cấu hình để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ bằng cách xác định nhiều mẫu siêu dữ liệu được khôi phục $(x_1'(n), \dots, x_N'(n))$ cho mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$.

Ngoài ra, thiết bị 100 bao gồm bộ tạo kênh âm thanh 120 để tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$.

Bộ giải mã siêu dữ liệu 110 được tạo cấu hình để nhận nhiều mẫu siêu dữ liệu được xử lý $(z_1(n), \dots, z_N(n))$ của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $(z_1, \dots, z_N)$. Ngoài ra, bộ giải mã siêu dữ liệu 110 được tạo cấu hình để nhận tín hiệu điều khiển (b).

Hơn nữa, bộ giải mã siêu dữ liệu 110 được tạo cấu hình để xác định từng mẫu siêu dữ liệu được khôi phục $(x_i'(n))$ trong số nhiều mẫu siêu dữ liệu được khôi phục $(x_i'(1), \dots, x_i'(n-1), x_i'(n))$ của mỗi tín hiệu siêu dữ liệu được khôi phục (x_i') trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$, sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục $(x_i'(n))$ đã nêu là tổng của một trong số các mẫu siêu dữ liệu được xử lý $(z_i(n))$ của một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (z_i) và mẫu siêu dữ liệu được khôi phục đã được tạo ra khác $(x_i'(n-1))$ của tín hiệu siêu dữ liệu được khôi phục đã nêu (x_i') , và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được khôi phục $(x_i'(n))$ đã nêu là một mẫu $(z_i(n))$ đã nêu trong số các mẫu siêu dữ liệu được xử lý $(z_i(1), \dots, z_i(n))$ của một tín hiệu (z_i) đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $(z_1, \dots, z_N)$.

Khi kết xuất các mẫu siêu dữ liệu, cần lưu ý rằng, mẫu siêu dữ liệu được đặc trưng bởi giá trị mẫu siêu dữ liệu của nó, nhưng cũng bởi thời điểm của thời gian mà

nó liên quan. Ví dụ, thời điểm của thời gian này có thể liên quan đến điểm bắt đầu của chuỗi âm thanh hoặc tương tự. Ví dụ, chỉ số n hoặc k có thể nhận biết vị trí của mẫu siêu dữ liệu trong tín hiệu siêu dữ liệu và nhờ đó, thời điểm của thời gian (liên quan) (liên quan đến thời gian bắt đầu) được biểu thị. Cần lưu ý rằng khi hai mẫu siêu dữ liệu liên quan đến các thời điểm của thời gian khác, hai mẫu siêu dữ liệu này khác các mẫu siêu dữ liệu, thậm chí khi các giá trị mẫu siêu dữ liệu của nó là bằng, mà đôi khi có thể xảy ra.

Các phương án trên được dựa trên phát hiện ra rằng thông tin siêu dữ liệu (được chứa bởi tín hiệu siêu dữ liệu) mà được kết hợp với tín hiệu đối tượng âm thanh được thay đổi chậm.

Ví dụ, tín hiệu siêu dữ liệu có thể biểu thị thông tin vị trí trên đối tượng âm thanh (ví dụ, góc phương vị, góc độ cao hoặc bán kính định rõ vị trí của đối tượng âm thanh). Có thể được giả định rằng, ở hầu hết thời điểm, vị trí của đối tượng âm thanh hoặc không thay đổi hoặc chỉ thay đổi chậm.

Hoặc, tín hiệu siêu dữ liệu có thể, ví dụ, biểu thị âm lượng (ví dụ, độ khuếch đại) của đối tượng âm thanh, và cũng có thể được giả định rằng, tại hầu hết thời điểm, âm lượng của đối tượng âm thanh thay đổi chậm.

Với lý do này, không cần thiết để truyền dẫn thông tin siêu dữ liệu (trộn vện) tại mọi thời điểm của thời gian.

Thay vào đó, thông tin siêu dữ liệu (trộn vện), có thể, ví dụ, theo một số phương án, chỉ được truyền dẫn tại các thời điểm nhất định của thời gian, ví dụ, một cách định kỳ, ví dụ, tại mọi thời điểm thứ N của thời gian, ví dụ, tại các điểm trong thời gian 0 , N , $2N$, $3N$, v.v.

Ví dụ, trong các phương án, ba tín hiệu siêu dữ liệu định rõ vị trí của đối tượng âm thanh trong không gian 3D. Một tín hiệu thứ nhất trong số các tín hiệu siêu dữ liệu có thể, ví dụ, định rõ góc phương vị của vị trí của đối tượng âm thanh. Một tín hiệu thứ hai trong số các tín hiệu siêu dữ liệu có thể, ví dụ, định rõ góc độ cao của vị trí của đối tượng âm thanh. Một tín hiệu thứ ba trong số các tín hiệu siêu dữ liệu có thể, ví dụ, định rõ bán kính liên quan đến khoảng cách của đối tượng âm thanh.

Góc phương vị, góc độ cao và bán kính định rõ một cách rõ ràng vị trí của đối tượng âm thanh trong không gian 3D từ gốc. Điều này được minh họa với việc tham chiếu đến Fig.4.

Fig.4 minh họa vị trí 410 của đối tượng âm thanh trong không gian ba chiều (3D) từ gốc 400 được biểu thị bởi phương vị, độ cao và bán kính.

Góc độ cao định rõ, ví dụ, góc giữa đường thẳng từ gốc đến vị trí đối tượng và hình chiếu vuông góc của đường thẳng này lên trên mặt phẳng xy (mặt phẳng được xác định bởi trục x và trục y). Góc phương vị xác định, ví dụ, góc giữa trục x và hình chiếu vuông góc đã nêu. Bằng cách định rõ góc phương vị và góc độ cao, đường thẳng 415 qua gốc 400 và vị trí 410 của đối tượng âm thanh có thể được xác định. Bằng cách định rõ thêm nữa bán kính, vị trí chính xác 410 của đối tượng âm thanh có thể được xác định.

Trong phương án, góc phương vị được xác định trong khoảng: $-180^\circ < \text{góc phương vị} \leq 180^\circ$, góc độ cao được xác định trong khoảng: $-90^\circ \leq \text{góc độ cao} \leq 90^\circ$ và bán kính có thể, ví dụ, được xác định theo mét [m] (lớn hơn hoặc bằng 0m).

Trong phương án khác, mà có thể, ví dụ, được giả định rằng tất cả các giá trị x của các vị trí đối tượng âm thanh trong hệ tọa độ xyz là lớn hơn hoặc bằng không, góc phương vị có thể được xác định trong khoảng: $-90^\circ \leq \text{góc phương vị} \leq 90^\circ$, góc độ cao có thể được xác định trong khoảng: $-90^\circ \leq \text{góc độ cao} \leq 90^\circ$, và bán kính có thể, ví dụ, được xác định theo mét [m].

Trong phương án nữa, các tín hiệu siêu dữ liệu có thể được định tỉ lệ sao cho góc phương vị được xác định trong khoảng: $-128^\circ < \text{góc phương vị} \leq 128^\circ$, góc độ cao được xác định trong khoảng: $-32^\circ \leq \text{góc độ cao} \leq 32^\circ$ và bán kính có thể, ví dụ, được xác định trên tỉ lệ lôgarit. Trong một số phương án, lần lượt các tín hiệu siêu dữ liệu ban đầu, các tín hiệu siêu dữ liệu được xử lý và các tín hiệu siêu dữ liệu được khôi phục, có thể bao gồm biểu diễn được định tỉ lệ của thông tin vị trí và/hoặc biểu diễn được định tỉ lệ của âm lượng của một trong số một hoặc nhiều tín hiệu đối tượng âm thanh.

Bộ tạo kênh âm thanh 120 có thể, ví dụ, được tạo cấu hình để tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào các tín hiệu siêu dữ liệu được khôi phục, trong đó các tín hiệu siêu dữ liệu được khôi phục, ví dụ, biểu thị vị trí của các đối tượng âm thanh.

Fig.5 minh họa các vị trí của các đối tượng âm thanh và thiết lập loa phát thanh được giả định bởi bộ tạo kênh âm thanh. Gốc 500 của hệ tọa độ xyz được minh họa. Ngoài ra, vị trí 510 của đối tượng âm thanh thứ nhất và vị trí 520 của đối tượng âm thanh thứ hai được minh họa. Hơn nữa, Fig.5 minh họa trường hợp, trong đó bộ tạo kênh âm thanh 120 tạo ra bốn kênh âm thanh cho bốn loa phát thanh. Bộ tạo kênh âm thanh 120 giả định rằng bốn loa phát thanh 511, 512, 513 và 514 được đặt tại các vị trí được thể hiện trên Fig.5.

Trên Fig.5, đối tượng âm thanh thứ nhất được đặt tại vị trí 510 gần các vị trí được giả định của các loa phát thanh 511 và 512, và được đặt cách xa các loa phát thanh 513 và 514. Do đó, bộ tạo kênh âm thanh 120 có thể tạo ra bốn kênh âm thanh sao cho đối tượng âm thanh thứ nhất 510 được tái tạo bởi loa phát thanh 511 và 512 nhưng không được tái tạo bởi các loa phát thanh 513 và 514.

Trong các phương án khác, bộ tạo kênh âm thanh 120 có thể tạo ra bốn kênh âm thanh sao cho đối tượng âm thanh thứ nhất 510 được tái tạo với âm lượng cao bởi loa phát thanh 511 và 512 và được tái tạo với âm lượng thấp bởi các loa phát thanh 513 và 514.

Ngoài ra, đối tượng âm thanh thứ hai được đặt tại vị trí 520 gần các vị trí được giả định của các loa phát thanh 513 và 514, và được đặt cách xa các loa phát thanh 511 và 512. Do đó, bộ tạo kênh âm thanh 120 có thể tạo ra bốn kênh âm thanh sao cho đối tượng âm thanh thứ hai 520 được tái tạo bởi các loa phát thanh 513 và 514 nhưng không được tái tạo bởi các loa phát thanh 511 và 512.

Trong các phương án khác, bộ tạo kênh âm thanh 120 có thể tạo ra bốn kênh âm thanh sao cho đối tượng âm thanh thứ hai 520 được tái tạo với âm lượng cao bởi các loa phát thanh 513 và 514 và được tái tạo với âm lượng thấp bởi các loa phát thanh 511 và 512.

Trong các phương án thay thế, chỉ hai tín hiệu siêu dữ liệu được sử dụng để định rõ vị trí của đối tượng âm thanh. Ví dụ, chỉ phương vị và bán kính có thể được định rõ, ví dụ, khi được giả định rằng tất cả các đối tượng âm thanh được đặt trong mặt phẳng đơn.

Trong các phương án khác nữa, với mỗi đối tượng âm thanh, chỉ tín hiệu siêu dữ liệu đơn được mã hóa và được truyền dẫn như thông tin vị trí. Ví dụ, chỉ góc phương vị có thể được định rõ như thông tin vị trí cho đối tượng âm thanh (ví dụ, có thể được giả định rằng tất cả các đối tượng âm thanh được đặt trong cùng mặt phẳng có cùng khoảng cách từ điểm trung tâm, và do đó được giả định có cùng bán kính). Thông tin phương vị có thể, ví dụ, là đủ để xác định rằng đối tượng âm thanh được đặt gần với loa phát thanh bên trái và xa loa phát thanh bên phải. Trong tình huống như vậy, bộ tạo kênh âm thanh 120 có thể, ví dụ, tạo ra một hoặc nhiều kênh âm thanh sao cho đối tượng âm thanh được tái tạo bởi loa phát thanh bên trái, nhưng không được tái tạo bởi loa phát thanh bên phải.

Ví dụ, Quét biên độ gốc vectơ (Vector Base Amplitude Panning-VBAP) có thể được sử dụng (xem, ví dụ, tài liệu [11]) để xác định trọng số của tín hiệu đối tượng âm thanh trong mỗi kênh âm thanh của các loa phát thanh. Ví dụ, đối với VBAP, được giả định rằng đối tượng âm thanh liên quan đến nguồn ảo.

Trong các phương án, tín hiệu siêu dữ liệu thêm nữa có thể định rõ âm lượng, ví dụ, độ khuếch đại (ví dụ, được biểu thị dạng dexiben [dB]) cho mỗi đối tượng âm thanh.

Ví dụ, trên Fig.5, giá trị độ khuếch đại thứ nhất có thể được định rõ bởi tín hiệu siêu dữ liệu thêm nữa cho đối tượng âm thanh thứ nhất được đặt tại vị trí 510 mà cao hơn giá trị độ khuếch đại thứ hai được định rõ bởi tín hiệu siêu dữ liệu thêm nữa khác cho đối tượng âm thanh thứ hai được đặt tại vị trí 520. Trong tình huống này, các loa phát thanh 511 và 512 có thể tái tạo đối tượng âm thanh thứ nhất với âm lượng cao hơn âm lượng mà với các loa phát thanh 513 và 514 tái tạo đối tượng âm thanh thứ hai.

Các phương án cũng giả định rằng các giá trị độ khuếch đại này của các đối tượng âm thanh thường thay đổi chậm. Do đó, không cần thiết để truyền dẫn thông tin

siêu dữ liệu như vậy tại mọi thời điểm. Thay vào đó, thông tin siêu dữ liệu chỉ được truyền dẫn tại thời điểm nhất định. Tại các thời điểm trung gian, thông tin siêu dữ liệu có thể, ví dụ, được lấy gần đúng sử dụng mẫu siêu dữ liệu trước và mẫu siêu dữ liệu sau, mà được truyền dẫn. Ví dụ, phép nội suy tuyến tính có thể được sử dụng cho phép gần đúng của các giá trị trung gian. Ví dụ, độ khuếch đại, phương vị, độ cao và/hoặc bán kính của mỗi đối tượng trong số các đối tượng âm thanh có thể được tính gần đúng cho các thời điểm, mà tại đó siêu dữ liệu này được truyền dẫn.

Bằng cách tiếp cận này, tiết kiệm đáng kể tốc độ truyền dẫn của siêu dữ liệu có thể đạt được.

Fig.3 minh họa hệ thống theo phương án.

Hệ thống bao gồm thiết bị 250 để tạo ra thông tin âm thanh được mã hóa bao gồm một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý như mô tả ở trên.

Ngoài ra, hệ thống bao gồm thiết bị 100 để nhận một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, và để tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu âm thanh được mã hóa và phụ thuộc vào một hoặc nhiều tín hiệu siêu dữ liệu được xử lý như được mô tả ở trên.

Ví dụ, một hoặc nhiều tín hiệu âm thanh được mã hóa có thể được giải mã bởi thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh bằng cách sử dụng bộ giải mã SAOC theo tình trạng kỹ thuật để thu được một hoặc nhiều tín hiệu đối tượng âm thanh, khi thiết bị 250 để mã hóa đã sử dụng bộ mã hóa SAOC để mã hóa một hoặc nhiều đối tượng âm thanh.

Các phương án dựa trên việc phát hiện ra rằng, các khái niệm Điều biến Mã Xung Vi phân có thể được mở rộng, và các khái niệm được mở rộng này sau đó để mã hóa phù hợp các tín hiệu siêu dữ liệu cho các đối tượng âm thanh.

Phương pháp Điều biến Mã Xung Vi phân (Differential Pulse Code Modulation-DPCM) là phương pháp được thiết lập cho các tín hiệu thời gian biến đổi chậm mà

giảm tính không thích hợp thông qua sự lượng tử hóa và phần dư thông qua sự truyền dẫn vi phân [10]. Bộ mã hóa DPCM được thể hiện trên Fig.6.

Trong bộ mã hóa DPCM của Fig.6, mẫu đầu vào thực $x(n)$ của tín hiệu đầu vào x được nạp vào bộ trừ 610. Tại đầu vào khác của bộ trừ, giá trị khác được nạp vào bộ trừ. Có thể được giả định rằng giá trị khác này là mẫu được nhận trước $x(n-1)$, mặc dù các sai số lượng tử hóa hoặc các sai số khác có thể có kết quả mà giá trị tại đầu vào khác là không giống một cách chính xác với mẫu có trước $x(n-1)$. Vì các độ lệch có thể này từ $x(n-1)$, đầu vào khác của bộ trừ có thể đề cập đến như $x^*(n-1)$. Bộ trừ trừ $x^*(n-1)$ khỏi $x(n)$ để thu được giá trị chênh lệch $d(n)$.

$d(n)$ sau đó được lượng tử hóa trong bộ lượng tử hóa 620 để thu được mẫu đầu ra khác $y(n)$ của tín hiệu đầu ra y . Thông thường, $y(n)$ hoặc là bằng với $d(n)$ hoặc là giá trị gần bằng $d(n)$.

Ngoài ra, $y(n)$ được nạp vào bộ cộng 630. Hơn nữa, $x^*(n-1)$ được nạp vào bộ cộng 630. Vì các kết quả $d(n)$ từ phép trừ $d(n) = x(n) - x^*(n-1)$, và vì $y(n)$ là giá trị bằng hoặc ít nhất gần bằng $d(n)$, đầu ra $x^*(n)$ của bộ cộng 630 là bằng với $x(n)$ hoặc ít nhất gần bằng $x(n)$.

$x^*(n)$ được giữ cho chu kỳ lấy mẫu trong bộ phận 640, và sau đó, việc xử lý được tiếp tục với mẫu tiếp theo $x(n+1)$.

Fig.7 thể hiện bộ giải mã DPCM tương ứng.

Trên Fig.7, mẫu $y(n)$ của tín hiệu đầu ra y từ bộ mã hóa DPCM được nạp vào bộ cộng 710. $y(n)$ biểu diễn giá trị chênh lệch của tín hiệu $x(n)$ mà sẽ được khôi phục. Tại đầu vào khác của bộ cộng 710, mẫu được khôi phục có trước $x'(n-1)$ được nạp vào bộ cộng 710. Đầu ra $x'(n)$ của bộ cộng thu được từ phép cộng $x'(n) = x'(n-1) + y(n)$. Vì $x'(n-1)$, thông thường bằng hoặc ít nhất gần bằng $x(n-1)$, và vì $y(n)$, thông thường bằng hoặc gần bằng $x(n) - x(n-1)$, đầu ra $x'(n)$ của bộ cộng 710, thông thường bằng hoặc gần bằng $x(n)$.

$x'(n)$ được giữ cho chu kỳ lấy mẫu trong bộ phận 740, và sau đó, việc xử lý được tiếp tục với mẫu tiếp theo $y(n+1)$.

Trong khi phương pháp nén DPCM đáp ứng hầu hết các tính năng được yêu cầu đã nêu trước đây, nó không cho phép truy nhập ngẫu nhiên.

Fig.8a minh họa bộ mã hóa siêu dữ liệu 801 theo phương án.

Phương pháp mã hóa được sử dụng bởi bộ mã hóa siêu dữ liệu 801 của Fig.8a là sự mở rộng của phương pháp mã hóa DPCM cổ điển.

Bộ mã hóa siêu dữ liệu 801 của Fig.8a bao gồm một hoặc nhiều bộ mã hóa DPCM 811, ..., 81N. Ví dụ, khi bộ mã hóa siêu dữ liệu 801 được tạo cấu hình để nhận N tín hiệu siêu dữ liệu ban đầu, bộ mã hóa siêu dữ liệu 801 có thể, ví dụ, bao gồm chính xác N bộ mã hóa DPCM. Trong phương án, mỗi bộ mã hóa trong số N bộ mã hóa DPCM được thực hiện như được mô tả với Fig.6.

Trong phương án, mỗi bộ mã hóa trong số N bộ mã hóa DPCM được tạo cấu hình để nhận các mẫu siêu dữ liệu $x_i(n)$ của một trong số N tín hiệu siêu dữ liệu ban đầu $x_1, \dots, x_N$, và tạo ra giá trị chênh lệch như mẫu chênh lệch $y_i(n)$ của tín hiệu chênh lệch siêu dữ liệu y_i cho mỗi mẫu siêu dữ liệu $x_i(n)$ của tín hiệu siêu dữ liệu ban đầu đã nêu x_i , mà được nạp vào bộ mã hóa DPCM đã nêu. Trong phương án, việc tạo ra mẫu chênh lệch $y_i(n)$ có thể, ví dụ, được điều khiển như được mô tả với Fig.6.

Bộ mã hóa siêu dữ liệu 801 của Fig.8a còn bao gồm bộ chọn 830 ("A"), mà được tạo cấu hình để nhận tín hiệu điều khiển $b(n)$.

Ngoài ra, bộ chọn 830 được tạo cấu hình để nhận N tín hiệu chênh lệch siêu dữ liệu $y_1 \dots y_N$.

Hơn nữa, trong phương án của Fig.8a, bộ mã hóa siêu dữ liệu 801 bao gồm bộ lượng tử hóa 820 mà lượng tử hóa N tín hiệu siêu dữ liệu ban đầu $x_1, \dots, x_N$ để thu được N tín hiệu siêu dữ liệu được lượng tử hóa $q_1, \dots, q_N$. Trong phương án này, bộ lượng tử hóa có thể được tạo cấu hình để nạp N tín hiệu siêu dữ liệu được lượng tử hóa vào bộ chọn 830.

Bộ chọn 830 có thể được tạo cấu hình để tạo ra các tín hiệu siêu dữ liệu được xử lý z_i từ các tín hiệu siêu dữ liệu được lượng tử hóa q_i và từ các tín hiệu siêu dữ liệu chênh lệch được mã hóa 830 y_i phụ thuộc vào tín hiệu điều khiển $b(n)$.

Ví dụ, khi tín hiệu điều khiển b là ở trạng thái thứ nhất (ví dụ, $b(n)=0$), bộ chọn 830 có thể được tạo cấu hình để xuất ra các mẫu chênh lệch $y_i(n)$ của các tín hiệu chênh lệch siêu dữ liệu y_i như các mẫu siêu dữ liệu $z_i(n)$ của các tín hiệu siêu dữ liệu được xử lý z_i .

Khi tín hiệu điều khiển b ở trạng thái thứ hai, khác với trạng thái thứ nhất (ví dụ, $b(n)=1$), bộ chọn 830 có thể được tạo cấu hình để xuất ra các mẫu siêu dữ liệu $q_i(n)$ của các tín hiệu siêu dữ liệu được lượng tử hóa q_i như các mẫu siêu dữ liệu $z_i(n)$ của các tín hiệu siêu dữ liệu được xử lý z_i .

Fig.8b minh họa bộ mã hóa siêu dữ liệu 802 theo phương án khác.

Trong phương án của Fig.8b, bộ mã hóa siêu dữ liệu 802 không bao gồm bộ lượng tử hóa 820, và thay vào đó N tín hiệu siêu dữ liệu được lượng tử hóa $q_1, \dots, q_N$, N tín hiệu siêu dữ liệu ban đầu $x_1, \dots, x_N$ được nạp trực tiếp vào bộ chọn 830.

Trong phương án này, ví dụ, khi tín hiệu điều khiển b ở trạng thái thứ nhất (ví dụ, $b(n)=0$), bộ chọn 830 có thể được tạo cấu hình để xuất ra các mẫu chênh lệch $y_i(n)$ của các tín hiệu chênh lệch siêu dữ liệu y_i như các mẫu siêu dữ liệu $z_i(n)$ của các tín hiệu siêu dữ liệu được xử lý z_i .

Khi tín hiệu điều khiển b ở trạng thái thứ hai, khác với trạng thái thứ nhất (ví dụ, $b(n)=1$), bộ chọn 830 có thể được tạo cấu hình để xuất ra các mẫu siêu dữ liệu $x_i(n)$ của các tín hiệu siêu dữ liệu ban đầu x_i như các mẫu siêu dữ liệu $z_i(n)$ của các tín hiệu siêu dữ liệu được xử lý z_i .

Fig.9a minh họa bộ giải mã siêu dữ liệu 901 theo phương án. Bộ mã hóa siêu dữ liệu theo Fig.9a tương ứng với các bộ mã hóa siêu dữ liệu của Fig.8a và Fig.8b.

Bộ giải mã siêu dữ liệu 901 của Fig.9a bao gồm một hoặc nhiều khối con của bộ giải mã siêu dữ liệu 911, ..., 91N. Bộ giải mã siêu dữ liệu 901 được tạo cấu hình để nhận một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $z_1, \dots, z_N$. Ngoài ra, bộ giải mã siêu dữ liệu 901 được tạo cấu hình để nhận tín hiệu điều khiển b . Bộ giải mã siêu dữ liệu được tạo cấu hình để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $x_1', \dots, x_N'$ từ một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $z_1, \dots, z_N$ phụ thuộc vào tín hiệu điều khiển b .

Trong phương án, mỗi tín hiệu trong số N tín hiệu siêu dữ liệu được xử lý $z_1, \dots, z_N$ được nạp vào một khối con khác trong số các khối con của bộ giải mã siêu dữ liệu 911, ..., 91N. Ngoài ra, theo phương án, tín hiệu điều khiển b được nạp vào mỗi khối con trong số các khối con của bộ giải mã siêu dữ liệu 911, ..., 91N. Theo phương án, số khối con của bộ giải mã siêu dữ liệu 911, ..., 91N là đồng nhất với số tín hiệu siêu dữ liệu được xử lý $z_1, \dots, z_N$ mà được nhận là bộ giải mã siêu dữ liệu 901.

Fig.9b minh họa khối con của bộ giải mã siêu dữ liệu (91i) trong số các khối con của bộ giải mã siêu dữ liệu 911, ..., 91N của Fig.9a theo phương án. Khối con của bộ giải mã siêu dữ liệu 91i được tạo cấu hình để điều khiển việc giải mã cho tín hiệu siêu dữ liệu được xử lý đơn lẻ z_i . Khối con của bộ giải mã siêu dữ liệu 91i bao gồm bộ chọn 930 ("B") và bộ cộng 910.

Khối con của bộ giải mã siêu dữ liệu 91i được tạo cấu hình để tạo ra tín hiệu siêu dữ liệu được khôi phục x_i' từ tín hiệu siêu dữ liệu được xử lý được nhận z_i phụ thuộc vào tín hiệu điều khiển $b(n)$.

Điều này, có thể, được thực hiện như sau:

Mẫu siêu dữ liệu được khôi phục cuối cùng $x_i'(n-1)$ của tín hiệu siêu dữ liệu được khôi phục x_i' được nạp vào bộ cộng 910. Ngoài ra, mẫu siêu dữ liệu thực $z_i(n)$ của tín hiệu siêu dữ liệu được xử lý z_i cũng được nạp vào bộ cộng 910. Bộ cộng được tạo cấu hình để cộng mẫu siêu dữ liệu được khôi phục cuối cùng $x_i'(n-1)$ và mẫu siêu dữ liệu thực $z_i(n)$ để thu được giá trị tổng $s_i(n)$ mà được nạp vào bộ chọn 930.

Ngoài ra, mẫu siêu dữ liệu thực $z_i(n)$ cũng được nạp vào bộ cộng 930.

Bộ chọn được tạo cấu hình để chọn hoặc giá trị tổng $s_i(n)$ từ bộ cộng 910 hoặc mẫu siêu dữ liệu thực $z_i(n)$ như mẫu siêu dữ liệu thực $x_i'(n)$ của tín hiệu siêu dữ liệu được khôi phục $x_i'(n)$ phụ thuộc vào tín hiệu điều khiển b .

Khi, ví dụ, tín hiệu điều khiển b ở trạng thái thứ nhất (ví dụ, $b(n) = 0$), tín hiệu điều khiển b biểu thị rằng mẫu siêu dữ liệu thực $z_i(n)$ là giá trị chênh lệch, và do đó, giá trị tổng $s_i(n)$ là mẫu siêu dữ liệu thực đúng $x_i'(n)$ của tín hiệu siêu dữ liệu được khôi phục x_i' . Bộ chọn 830 được tạo cấu hình để chọn giá trị tổng $s_i(n)$ như mẫu siêu

dữ liệu thực $x_i'(n)$ của tín hiệu siêu dữ liệu được khôi phục x_i' , khi tín hiệu điều khiển ở trạng thái thứ nhất (khi $b(n) = 0$).

Khi tín hiệu điều khiển b ở trạng thái thứ hai, khác trạng thái thứ nhất (ví dụ, $b(n)=1$), tín hiệu điều khiển b biểu thị rằng mẫu siêu dữ liệu thực $z_i(n)$ không phải là giá trị chênh lệch, và do đó, mẫu siêu dữ liệu thực $z_i(n)$ là mẫu siêu dữ liệu thực đúng $x_i'(n)$ của tín hiệu siêu dữ liệu được khôi phục x_i' . Bộ chọn 830 được tạo cấu hình để chọn mẫu siêu dữ liệu thực $z_i(n)$ như mẫu siêu dữ liệu thực $x_i'(n)$ của tín hiệu siêu dữ liệu được khôi phục x_i' , khi tín hiệu điều khiển ở trạng thái thứ hai (khi $b(n) = 1$).

Theo các phương án, khối con của bộ giải mã siêu dữ liệu 91i' còn bao gồm bộ 920. Bộ 920 được tạo cấu hình để giữ mẫu siêu dữ liệu thực $x_i'(n)$ của tín hiệu siêu dữ liệu được khôi phục cho khoảng thời gian của chu kỳ lấy mẫu. Trong phương án, điều này đảm bảo, rằng khi $x_i'(n)$ được tạo ra, $x'(n)$ được tạo ra không được nạp trở lại quá dễ dàng, sao cho khi $z_i(n)$ là giá trị chênh lệch, $x_i'(n)$ thực sự được tạo ra dựa trên $x_i'(n-1)$.

Trong phương án của Fig.9b, bộ chọn 930 có thể tạo ra các mẫu siêu dữ liệu $x_i'(n)$ từ thành phần tín hiệu được nhận $z_i(n)$ và tổ hợp tuyến tính của thành phần đầu ra bị trễ (mẫu siêu dữ liệu đã được tạo ra của tín hiệu siêu dữ liệu được khôi phục) và thành phần tín hiệu được nhận $z_i(n)$ phụ thuộc vào tín hiệu điều khiển $b(n)$.

Sau đây, các tín hiệu được mã hóa DPCM được biểu thị như $y_i(n)$ và tín hiệu đầu vào thứ hai (tín hiệu tổng) của B như $s_i(n)$. Với các thành phần đầu ra mà chỉ phụ thuộc vào các thành phần đầu vào tương ứng, đầu ra bộ mã hóa và bộ giải mã được đưa ra như sau:

$$z_i(n) = A(x_i(n), v_i(n), b(n))$$

$$x_i'(n) = B(z_i(n), s_i(n), b(n))$$

Giải pháp theo phương án cho cách tiếp cận chung phác thảo trên là sử dụng $b(n)$ để chuyển đổi giữa các tín hiệu được mã hóa DPCM và tín hiệu đầu vào được lượng tử hóa. Bỏ qua chỉ số thời gian n để đơn giản hóa, các khối chức năng A và B sau đó được đưa ra như sau:

Trong các bộ mã hóa siêu dữ liệu 801, 802, bộ chọn 830 (A) chọn:

$$A: z_i(x_i, y_i, b) = y_i, \quad \text{nếu } b = 0 \quad (z_i \text{ biểu thị giá trị chênh lệch})$$

$$A: z_i(x_i, y_i, b) = y_i, \quad \text{nếu } b = 1 \quad (z_i \text{ không biểu thị giá trị chênh lệch})$$

Trong các bộ con của bộ giải mã siêu dữ liệu 91i, 91i', bộ chọn 930 (B) chọn:

$$B: z_i(x_i, y_i, b) = y_i, \quad \text{nếu } b = 0 \quad (z_i \text{ biểu thị giá trị chênh lệch})$$

$$B: z_i(x_i, y_i, b) = y_i, \quad \text{nếu } b = 1 \quad (z_i \text{ không biểu thị giá trị chênh lệch})$$

Điều này cho phép truyền dẫn tín hiệu đầu vào được lượng tử hóa mỗi khi $b(n)$ bằng 1 và truyền dẫn tín hiệu DPCM mỗi khi $b(n)$ bằng 0. Trong trường hợp sau, bộ giải mã trở thành bộ giải mã DPCM.

Khi được áp dụng cho sự truyền dẫn siêu dữ liệu đối tượng, có cấu trúc này được sử dụng để truyền dẫn một cách đều đặn các vị trí đối tượng không được nén mà có thể được sử dụng bởi bộ giải mã cho truy cập ngẫu nhiên.

Trong các phương án ưu tiên, số bit ít hơn được sử dụng để mã hóa các giá trị chênh lệch so với số bit được sử dụng để mã hóa các mẫu siêu dữ liệu. Các phương án này dựa trên việc phát hiện ra rằng (ví dụ, N) mẫu siêu dữ liệu sau trong hầu hết thời gian chỉ thay đổi nhỏ. Ví dụ, nếu một loại mẫu siêu dữ liệu được mã hóa, ví dụ, bằng 8 bit, mẫu siêu dữ liệu này có thể nhận một giá trị ngoài 256 giá trị khác nhau. Thông thường, vì những thay đổi nhỏ này của các giá trị siêu dữ liệu sau, có thể được coi là đủ, để mã hóa các giá trị chênh lệch, ví dụ, chỉ bằng 5 bit. Do đó, thậm chí nếu các giá trị chênh lệch được truyền dẫn, số bit được truyền dẫn có thể được giảm.

Trong phương án, bộ mã hóa siêu dữ liệu 210 được tạo cấu trúc để mã hóa mỗi mẫu tín hiệu siêu dữ liệu được xử lý $(z_i(1), \dots, z_i(n))$ của một tín hiệu $z_i()$ trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $(z_1, \dots, z_N)$ với số bit thứ nhất khi tín hiệu điều khiển biểu thị trạng thái thứ nhất ($b(n)=0$), và với số bit thứ hai khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$), trong đó số bit thứ nhất nhỏ hơn số bit thứ hai.

Trong phương án được ưu tiên, một hoặc nhiều giá trị chênh lệch được truyền dẫn, mỗi giá trị trong số một hoặc nhiều giá trị chênh lệch được mã hóa với ít bit hơn mỗi mẫu siêu dữ liệu và mỗi giá trị chênh lệch là giá trị số nguyên.

Theo phương án, bộ mã hóa siêu dữ liệu 110 được tạo cấu hình để mã hóa một hoặc nhiều mẫu siêu dữ liệu của một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý với số bit thứ nhất, trong đó mỗi mẫu trong số một hoặc nhiều mẫu siêu dữ liệu của một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý biểu thị số nguyên. Hơn nữa bộ mã hóa siêu dữ liệu (110) được tạo cấu hình để mã hóa một hoặc nhiều giá trị chênh lệch với số bit thứ hai, trong đó mỗi giá trị trong số một hoặc nhiều giá trị chênh lệch biểu thị số nguyên, trong đó số bit thứ hai nhỏ hơn số bit thứ nhất.

Xem xét, ví dụ, trong phương án, các mẫu siêu dữ liệu có thể biểu diễn phương vị được mã hóa bởi 8 bit. Ví dụ, phương vị có thể là số nguyên giữa $-90 \leq$ phương vị ≤ 90 . Do đó, phương vị có thể nhận 181 giá trị khác nhau. Tuy nhiên nếu, có thể giả định rằng (ví dụ N) mẫu phương vị sau chỉ khác bởi không lớn hơn, ví dụ, ± 15 , sau đó, 5 bit ($2^5=32$) có thể đủ để mã hóa các giá trị chênh lệch. Nếu các giá trị chênh lệch được biểu diễn dưới dạng các số nguyên, sau đó xác định những các giá trị chênh lệch biến đổi tự động các giá trị bổ sung, được truyền dẫn, sang khoảng giá trị thích hợp.

Ví dụ, xem xét trường hợp mà giá trị phương vị thứ nhất của đối tượng âm thanh thứ nhất là 60° và các giá trị sau của nó thay đổi từ 45° đến 75° . Ngoài ra, xem xét rằng giá trị phương vị thứ hai của đối tượng âm thanh thứ hai là -30° và các giá trị sau của nó thay đổi từ -45° đến -15° . Bằng cách xác định các giá trị chênh lệch cho cả hai giá trị sau của đối tượng âm thanh thứ nhất và cho cả hai giá trị sau của đối tượng âm thanh thứ hai, các giá trị chênh lệch của giá trị phương vị thứ nhất và giá trị phương vị thứ hai cả hai trong khoảng giá trị từ -15° đến $+15^\circ$, sao cho 5 bit là đủ để mã hóa mỗi giá trị chênh lệch và sao cho chuỗi bit, mà mã hóa các giá trị chênh lệch, có nghĩa tương tự cho các giá trị chênh lệch của góc phương vị thứ nhất và các giá trị chênh lệch của giá trị phương vị thứ hai.

Sau đây, các khung siêu dữ liệu đối tượng theo các phương án và biểu diễn biểu tượng theo các phương án được mô tả.

Siêu dữ liệu đối tượng được mã hóa được truyền dẫn trong các khung. Các khung siêu dữ liệu đối tượng này có thể chứa hoặc dữ liệu đối tượng được mã hóa nội bộ

hoặc dữ liệu đối tượng động mà sau đó chứa những thay đổi từ khung được truyền dẫn cuối cùng.

Một số hoặc tất cả các vị trí của cú pháp sau cho các khung siêu dữ liệu đối tượng có thể, ví dụ, được sử dụng:

	Số bit	dễ nhớ
object_metadata() { has_intracoded_object_metadata; (has_intracoded_object_metadata) { intracoded_object_metadata (); } else { dynamic_object_metadata(); } }	1	bslbf

Sau đây, dữ liệu đối tượng được mã hóa nội bộ theo phương án được mô tả.

Truy nhập ngẫu nhiên của siêu dữ liệu đối tượng được mã hóa được thực hiện thông qua dữ liệu đối tượng được mã hóa nội bộ (“khung-I”) mà chứa các giá trị được lượng tử hóa được lấy mẫu trên khung lưới đều đặn (ví dụ mỗi 32 khung có chiều dài 1024). Các Khung-I này có thể, ví dụ, có cú pháp sau, *position_azimuth*, *position_elevation*, *position_radius*, and *gain_factor* định rõ các giá trị được lượng tử hóa hiện thời:

	Số bit	Dễ nhớ
intracoded_object_metadata() { if (num_objects>1) { fixed_azimuth; if (fixed_azimuth) { default_azimuth; } else { common_azimuth; if (common_azimuth) { default_azimuth; } else { for (o=1:num_objects) { position_azimuth[o]; } } } } }	1 8 1 8 8	bslbf tcimsbf bslbf tcimsbf tcimsbf

}		
}		
fixed_elevation;	1	bslbf
if (fixed_azimuth) {		
default_elevation;	6	tcimsbf
}		
else {		
common_elevation;	1	bslbf
if (common_azimuth) {		
default_elevation;	6	tcimsbf
}		
else {		
for (o=1:num_objects) {		
position_azimuth[o];	6	tcimsbf
}		
}		
}		
fixed_radius;	1	bslbf
if (fixed_azimuth) {		
default_radius;	4	tcimsbf
}		
else {		
common_radius;	1	bslbf
if (common_azimuth) {		
default_radius;	4	tcimsbf
}		
else {		
for (o=1:num_objects) {		
position_radius [o];	4	tcimsbf
}		
}		
}		
fixed_gain;	1	bslbf
if (fixed_azimuth) {		
default_gain;	7	tcimsbf
}		
else {		
common_gain;	1	bslbf
if (common_azimuth) {		
default_gain;	7	tcimsbf
}		
else {		
for (o=1:num_objects) {		
gain_factor [o];	7	tcimsbf
}		
}		
}		
}		
else {		
position_azimuth;	8	tcimsbf
position_elevation;	6	tcimsbf

position_radius;	4	tcimsbf
gain_factor;	7	tcimsbf
}		
}		

Sau đây, dữ liệu đối tượng động theo theo phương án được mô tả.

Dữ liệu DPCM được truyền dẫn trong các khung đối tượng động mà có thể, ví dụ, có cú pháp sau:

	Số bit	Để nhớ
dynamic_object_metadata() { flag_absolute; for (o=1:num_objects) { has_object_metadata; if (has_object_metadata) { single_dynamic_object_metadata(flag_absolute); } } }	1 1	bslbf bslbf

	Số bit	Để nhớ
single_dynamic_object_metadata (flag_absolute) { if (flag_absolute) { if (!fixed_azimuth*) { position_azimuth; } if (!fixed_elevation*) { position_elevation; } if (!fixed_radius*) { position_radius; } if (!fixed_gain*) { gain_factor; } } else { nbits; if (!fixed_azimuth*) { flag_azimuth; if (flag_azimuth) { position_azimuth_difference ; } } if (!fixed_elevation*) { flag_elevation; }	8 6 4 7 3 1 num_bits 1	tcimsbf tcimsbf tcimsbf tcimsbf uimsbf bslbf tcimsbf bslbf

if (flag_elevation) { position_elevation_difference ;	min(num_bits,7)	tcimsbf
}		
if (!fixed_radius*) { flag_radius ;	1	bslbf
if (flag_radius) { position_radius_difference ;	min(num_bits,5)	tcimsbf
}		
}		
if (!fixed_gain*) { flag_gain ;	1	bslbf
if (flag_gain) { gain_factor_difference ;	min(num_bits,8)	tcimsbf
}		
}		

Lưu ý: num_bits = nbits + 2;
 Chú thích *: đã cho trước
 intracoded_object_data()-frame

Cụ thể, trong phương án, ví lệnh nêu trên có thể, ví dụ, có nghĩa như sau:

Definition of object_data() payloads theo phương án:

has_intracoded_object_metadata biểu thị hoặc khung được mã hóa nội bộ hoặc được mã hóa vi phân.

Definition of intracoded_object_metadata() payloads theo phương án:

fixed_azimuth biểu thị có hay không giá trị phương vị được cố định cho tất cả các đối tượng và hoặc không được truyền trong trường hợp của siêu dữ liệu _đối tượng_động ()

default_azimuth xác định giá trị của góc phương vị chung hoặc được cố định

common_azimuth biểu thị có hay không góc phương vị chung được sử dụng cho tất cả các đối tượng

position_azimuth nếu không có giá trị phương vị chung, giá trị cho mỗi đối tượng được truyền dẫn

fixed_elevation biểu thị có hay không giá trị độ cao được cố định cho tất cả các đối tượng và hoặc không được truyền dẫn trong trường hợp của siêu dữ liệu _đối tượng_động ()

default_elevation xác định giá trị của góc độ cao chung hoặc được cố định

common_elevation biểu thị có hay không góc độ cao chung được sử dụng cho tất cả các đối tượng.

position_elevation nếu không có giá trị độ cao chung, giá trị cho mỗi đối tượng được truyền dẫn

fixed_radius cờ hiệu biểu thị có hay không bán kính được cố định cho tất cả đối tượng và hoặc không được truyền dẫn trong trường hợp của siêu dữ liệu `_đối tượng_động ()`

default_radius định nghĩa giá trị của các bán kính chung

common_radius biểu thị có hay không giá trị bán kính chung được sử dụng cho tất cả các đối tượng.

position_radius nếu không có giá trị bán kính chung, giá trị cho mỗi đối tượng được truyền dẫn

fixed_gain cờ hiệu biểu thị có hay không hệ số khuếch đại được cố định cho tất cả các đối tượng và hoặc không được truyền dẫn trong trường hợp của siêu dữ liệu `_đối tượng_động ()`

default_gain xác định giá trị của hệ số khuếch đại chung hoặc được cố định

common_gain biểu thị có hay không giá trị độ khuếch đại chung được sử dụng cho tất cả các đối tượng.

gain_factor nếu không có giá trị độ khuếch đại chung, giá trị cho mỗi đối tượng được truyền dẫn

position_azimuth nếu chỉ có một đối tượng, đó là góc phương vị của nó.

position_elevation nếu chỉ có một đối tượng, đó là góc độ cao của nó.

position_radius nếu chỉ có một đối tượng, đó là bán kính của nó.

gain_factor nếu chỉ có một đối tượng, đó là hệ số khuếch đại của nó.

Definition of dynamic_object_metadata() payloads theo phương án:

flag_absolute biểu thị hoặc các giá trị của các thành phần được truyền dẫn vi phân hoặc các giá trị tuyệt đối

has_object_metadata biểu thị hoặc có dữ liệu đối tượng có mặt trong dòng bit hoặc không

Definition of single_dynamic_object_metadata() payloads theo phương án:

position_azimuth giá trị tuyệt đối của góc phương vị nếu giá trị không được cố định

position_elevation giá trị tuyệt đối của góc phương vị nếu giá trị không được cố định

position_radius giá trị tuyệt đối của bán kính nếu giá trị không được cố định

gain_factor giá trị tuyệt đối của góc phương vị nếu giá trị không được cố định

nbits bao nhiêu bit được yêu cầu để biểu diễn các giá trị vi phân

flag_azimuth cờ hiệu mỗi đối tượng biểu thị các hay không giá trị phương vị thay đổi

position_azimuth_difference chênh lệch giữa giá trị có trước và giá trị hoạt động

flag_elevation cờ hiệu mỗi đối tượng biểu thị các hay không giá trị phương vị thay đổi

position_elevation_difference chênh lệch giữa giá trị có trước và giá trị hoạt động

flag_radius cờ hiệu mỗi đối tượng biểu thị các hay không giá trị bán kính thay đổi

position_radius_difference chênh lệch giữa giá trị có trước và giá trị hoạt động

flag_gain cờ hiệu mỗi đối tượng biểu thị các hay không bán kính độ khuếch đại thay đổi

gain_factor_difference chênh lệch giữa giá trị có trước và giá trị hoạt động

Trong tình trạng kỹ thuật, kỹ thuật linh hoạt một mặt tồn tại tổ hợp mã hóa kênh và một mặt mã hóa đối tượng sao cho chất lượng âm thanh có thể chấp nhận tại tốc độ bit thấp thu được.

Giới hạn này được khắc phục bởi Hệ thống bộ mã hóa-giải mã âm thanh 3D. Bây giờ, Hệ thống bộ mã hóa-giải mã âm thanh 3D được mô tả.

Fig.10 minh họa bộ mã hóa âm thanh 3D phù hợp với phương án của sáng chế. Bộ mã hóa âm thanh 3D được tạo cấu hình để mã hóa dữ liệu đầu vào âm thanh 101 để thu được dữ liệu đầu ra âm thanh 501. Bộ mã hóa âm thanh 3D bao gồm giao diện đầu vào để nhận nhiều kênh âm thanh được biểu thị bằng CH và nhiều đối tượng âm thanh được biểu thị bằng OBJ. Hơn nữa, như được minh họa trên Fig.10, giao diện đầu vào 1100 nhận bổ sung siêu dữ liệu liên quan đến một hoặc nhiều trong số nhiều đối tượng âm thanh OBJ. Hơn nữa, bộ mã hóa âm thanh 3D bao gồm bộ trộn 200 để trộn nhiều đối tượng và nhiều kênh để thu được nhiều kênh được trộn-trước, trong đó mỗi kênh được trộn-trước bao gồm dữ liệu âm thanh của kênh và dữ liệu âm thanh của ít nhất một đối tượng.

Hơn nữa, bộ mã hóa âm thanh 3D bao gồm bộ mã hóa lõi 300 để mã hóa lõi dữ liệu đầu vào bộ mã hóa lõi, bộ nén siêu dữ liệu 400 để nén siêu dữ liệu liên quan đến một hoặc nhiều trong số nhiều đối tượng âm thanh.

Hơn nữa, bộ mã hóa âm thanh 3D có thể bao gồm bộ điều khiển chế độ 600 để điều khiển bộ trộn, bộ mã hóa lõi và/hoặc giao diện đầu ra 500 theo một trong số nhiều chế độ vận hành, trong đó trong chế độ thứ nhất, bộ mã hóa lõi được tạo cấu hình để mã hóa nhiều kênh âm thanh và nhiều đối tượng âm thanh được nhận bởi giao diện đầu vào 1100 mà không có bất kỳ tương tác bởi bộ trộn, tức là, không có bất kỳ sự trộn nào bởi bộ trộn 200. Tuy nhiên, trong chế độ thứ hai, trong đó bộ trộn 200 hoạt động, bộ mã hóa lõi mã hóa nhiều kênh được trộn, tức là, đầu ra được tạo ra bởi khối 200. Trong trường hợp thứ hai này, ưu tiên không mã hóa bất kỳ dữ liệu đối tượng nào nữa. Thay vào đó, siêu dữ liệu biểu thị các vị trí của các đối tượng âm thanh đã được sử dụng bởi bộ trộn 200 để kết xuất các đối tượng trên các kênh như được biểu thị bởi siêu dữ liệu. Nói cách khác, bộ trộn 200 sử dụng siêu dữ liệu liên quan đến nhiều đối tượng âm thanh để kết xuất-trước các đối tượng âm thanh và sau đó các đối tượng âm

thanh được kết xuất-trước được trộn với các kênh để thu được các kênh được trộn tại đầu ra của bộ trộn. Trong phương án này, bất kỳ đối tượng có thể không cần thiết để được truyền dẫn và điều này cũng áp dụng cho siêu dữ liệu được nén như đầu ra bởi khối 400. Tuy nhiên, nếu không phải tất cả các đối tượng đầu vào vào giao diện 1100 được trộn mà chỉ lượng nhất định các đối tượng được trộn, sau đó chỉ các đối tượng không được trộn còn lại và siêu dữ liệu được kết hợp tuy nhiên lần lượt được truyền dẫn đến bộ mã hóa lõi 300 hoặc bộ nén siêu dữ liệu 400.

Trên Fig.10, bộ nén siêu dữ liệu 400 là bộ mã hóa siêu dữ liệu 210 của thiết bị 250 để tạo ra thông tin âm thanh được mã hóa theo một trong số các phương án được mô tả ở trên. Ngoài ra, trên Fig.10, bộ trộn 200 và bộ mã hóa lõi 300 cùng với nhau tạo thành bộ mã hóa âm thanh 220 của thiết bị 250 để tạo ra thông tin âm thanh được mã hóa theo một trong số các phương án được mô tả ở trên.

Fig.12 minh họa phương án thêm nữa của bộ mã hóa âm thanh 3D mà, theo cách bổ sung, bao gồm bộ mã hóa SAOC 800. Bộ mã hóa SAOC 800 được tạo cấu hình để tạo ra một hoặc nhiều kênh vận chuyển và dữ liệu theo tham số từ dữ liệu đầu ra bộ mã hóa đối tượng âm thanh không gian. Như được minh họa trên Fig.12, dữ liệu đầu vào bộ mã hóa đối tượng âm thanh không gian là các đối tượng mà không được xử lý bởi bộ kết xuất-trước/bộ trộn. Ngoài ra, được cung cấp mà bộ kết xuất-trước/bộ trộn được đi vòng như chế độ một mà việc mã hóa kênh/đối tượng riêng lẻ được hoạt động, tất cả các đối tượng vào giao diện đầu vào 1100 được mã hóa bởi bộ mã hóa SAOC 800.

Hơn nữa, như được minh họa trên Fig.12, bộ mã hóa lõi 300 được thực hiện theo cách ưu tiên như bộ mã hóa USAC, tức là, như bộ mã hóa như được định nghĩa và được tiêu chuẩn hóa trong tiêu chuẩn MPEG-USAC (USAC= unified speech and audio coding - mã hóa âm thanh và tiếng nói hợp nhất). Đầu ra của bộ mã hóa âm thanh 3D hoàn toàn được minh họa trên Fig.12 là dòng dữ liệu MPEG 4 có các kết cấu như vật chứa cho các dạng dữ liệu riêng lẻ. Hơn nữa, siêu dữ liệu được biểu thị như dữ liệu "OAM" và bộ nén siêu dữ liệu 400 trên Fig.10 tương ứng như bộ mã hóa OAM 400 để thu được dữ liệu OAM được nén mà là đầu vào vào bộ mã hóa USAC 300 mà, như có thể thấy trên Fig.12, bao gồm theo cách bổ sung giao diện đầu ra để thu được

dòng dữ liệu đầu ra MP4 không chỉ có dữ liệu kênh/đối tượng được mã hóa mà còn có dữ liệu OAM được nén.

Trên Fig.12, bộ mã hóa OAM 400 là bộ mã hóa siêu dữ liệu 210 của thiết bị 250 để tạo ra thông tin âm thanh được mã hóa theo một trong số các phương án được mô tả ở trên. Ngoài ra, trên Fig.12, bộ mã hóa SAOC 800 và bộ mã hóa USAC 300 cùng với nhau tạo thành bộ mã hóa âm thanh 220 của thiết bị 250 để tạo ra thông tin âm thanh được mã hóa theo một trong số các phương án được mô tả ở trên.

Fig.14 minh họa phương án nữa của bộ mã hóa âm thanh 3D, trong đó trái với Fig.12, bộ mã hóa SAOC có thể được tạo cấu hình để hoặc mã hóa, với thuật toán mã hóa SAOC, các kênh được cung cấp tại bộ kết xuất-trước/bộ trộn 200 không được hoạt động trong chế độ này, hoặc theo cách thay thế, để mã hóa SAOC các kênh kết xuất-trước cộng các đối tượng. Do đó, trên Fig.14, bộ mã hóa SAOC 800 có thể vận hành trên ba loại dữ liệu đầu vào khác nhau, tức là, các kênh không có bất kỳ đối tượng được kết xuất-trước nào, các kênh và các đối tượng được kết xuất-trước hoặc chỉ một mình các đối tượng. Hơn nữa, ưu tiên để cung cấp bộ giải mã OAM bổ sung 420 trên Fig.14 để bộ mã hóa SAOC 800 sử dụng, cho việc xử lý của nó, dữ liệu tương tự như trên phía bộ giải mã, tức là dữ liệu thu được bởi việc nén tổn hao hơn là dữ liệu OAM ban đầu.

Bộ mã hóa âm thanh 3D Fig.14 có thể vận hành trong nhiều chế độ riêng lẻ.

Ngoài ra các chế độ thứ nhất và thứ hai như được thảo luận trong phạm vi của Fig.10, bộ mã hóa âm thanh 3D của Fig.14 có thể vận hành theo cách bổ sung trong chế độ thứ ba trong đó bộ mã hóa lõi tạo ra một hoặc nhiều kênh vận chuyển từ các đối tượng riêng lẻ khi bộ kết xuất-trước/bộ trộn 200 không hoạt động. Theo cách bổ sung hoặc thay thế, trong chế độ thứ ba bộ mã hóa SAOC 800 có thể tạo ra một hoặc nhiều kênh vận chuyển bổ sung hoặc thay thế từ các kênh ban đầu, tức là, một lần nữa khi bộ kết xuất-trước/bộ trộn 200 tương ứng với bộ trộn 200 của Fig.10 không hoạt động.

Cuối cùng, bộ mã hóa SAOC 800 có thể mã hóa, khi bộ mã hóa âm thanh 3D được tạo cấu hình trong chế độ thứ tư, các kênh cộng với các đối tượng được kết xuất-trước như được tạo ra bởi bộ kết xuất-trước/bộ trộn. Do đó, trong chế độ thứ tư các ứng dụng tốc độ bit thấp nhất sẽ được cung cấp chất lượng tốt do thực tế rằng các kênh

và các đối tượng được biến đổi hoàn toàn sang các kênh vận chuyển SAOC riêng lẻ và thông tin phụ được kết hợp như được biểu thị trên các Fig. 3 và 5 như “SAOC-SI” và, ngoài ra, bất kỳ siêu dữ liệu được nén không cần phải được truyền dẫn trong chế độ thứ tư.

Trên Fig.14, bộ mã hóa OAM 400 là bộ mã hóa siêu dữ liệu 210 của thiết bị 250 để tạo ra thông tin âm thanh được mã hóa theo một trong số các phương án được mô tả ở trên. Ngoài ra, trên Fig.14, bộ mã hóa SAOC 800 và bộ mã hóa USAC 300 cùng với nhau tạo thành bộ mã hóa âm thanh 220 của thiết bị 250 để tạo ra thông tin âm thanh được mã hóa theo một trong số các phương án được mô tả ở trên.

Theo phương án, thiết bị để mã hóa dữ liệu đầu vào âm thanh 101 để thu được dữ liệu đầu ra âm thanh 501 được đề xuất. Thiết bị mã hóa dữ liệu đầu vào âm thanh 101 bao gồm:

- giao diện đầu vào 1100 để nhận nhiều kênh âm thanh, nhiều đối tượng âm thanh và siêu dữ liệu liên quan đến một hoặc nhiều trong số nhiều đối tượng âm thanh,
- bộ trộn 200 để trộn nhiều đối tượng và nhiều kênh để thu được nhiều kênh được trộn-trước, mỗi kênh được trộn-trước bao gồm dữ liệu âm thanh của kênh và dữ liệu âm thanh của ít nhất một đối tượng, và
- thiết bị 250 để tạo ra thông tin âm thanh được mã hóa mà bao gồm bộ mã hóa siêu dữ liệu và bộ mã hóa âm thanh như được mô tả ở trên.

Bộ mã hóa âm thanh 220 của thiết bị 250 để tạo ra thông tin âm thanh được mã hóa là bộ mã hóa lõi (300) để mã hóa lõi dữ liệu đầu vào bộ mã hóa lõi.

Bộ mã hóa siêu dữ liệu 210 của thiết bị 250 để tạo ra thông tin âm thanh được mã hóa là bộ nén siêu dữ liệu 400 để nén siêu dữ liệu liên quan đến một hoặc nhiều trong số nhiều đối tượng âm thanh.

Fig.11 minh họa bộ giải mã âm thanh 3D phù hợp với phương án của sáng chế. Bộ giải mã âm thanh 3D nhận, như đầu vào, dữ liệu âm thanh được mã hóa, tức là dữ liệu 501 của Fig.10.

Bộ giải mã âm thanh 3D bao gồm bộ giải nén siêu dữ liệu 1400, bộ giải mã lõi 1300, bộ xử lý đối tượng 1200, bộ điều khiển chế độ 1600 và bộ xử lý sau 1700.

Cụ thể, bộ giải mã âm thanh 3D được tạo cấu hình để giải mã dữ liệu được mã hóa và giao diện đầu vào được tạo cấu hình để nhận dữ liệu âm thanh được mã hóa, dữ liệu âm thanh được mã hóa bao gồm nhiều kênh được mã hóa và nhiều đối tượng được mã hóa và siêu dữ liệu được nén liên quan đến nhiều đối tượng trong chế độ nhất định.

Hơn nữa, bộ giải mã lõi 1300 được tạo cấu hình để giải mã nhiều kênh được mã hóa và nhiều đối tượng được mã hóa và, ngoài ra, bộ giải nén siêu dữ liệu được tạo cấu hình để giải nén siêu dữ liệu được nén.

Hơn nữa, bộ xử lý đối tượng 1200 được tạo cấu hình để xử lý nhiều đối tượng được giải mã như được tạo ra bởi bộ giải mã lõi 1300 sử dụng siêu dữ liệu được giải nén để thu được số được xác định trước của các kênh đầu ra bao gồm dữ liệu đối tượng và kênh được giải mã. Các kênh đầu ra này như được biểu thị tại 1205 sau đó là đầu vào vào bộ xử lý sau 1700. Bộ xử lý sau 1700 được tạo cấu hình để chuyển đổi số kênh đầu ra 1205 thành định dạng đầu ra nhất định mà có thể là định dạng đầu ra lập thể hoặc định dạng đầu ra loa phát thanh như định dạng đầu ra 5,1, 7,1, v.v.

Ưu tiên, bộ giải mã âm thanh 3D bao gồm bộ điều khiển chế độ 1600 mà được tạo cấu hình để phân tích dữ liệu được mã hóa để phát hiện biểu thị chế độ. Do đó, bộ điều khiển chế độ 1600 được nối với giao diện đầu vào 1100 trên Fig.11. Tuy nhiên, theo cách thay thế, bộ điều khiển chế độ không nhất thiết phải có ở đây. Thay vào đó, bộ giải mã âm thanh linh hoạt có thể là thiết lập-trước bằng loại khác bất kỳ của dữ liệu điều khiển như đầu vào người dùng hoặc điều khiển khác bất kỳ. Bộ giải mã âm thanh 3D trên Fig.11 và, được điều khiển ưu tiên bởi bộ điều khiển chế độ 1600, được tạo cấu hình để hoặc đi vòng bộ xử lý đối tượng và để nạp nhiều kênh được giải mã vào bộ xử lý sau 1700. Điều đó là vận hành trong chế độ 2, tức là, trong đó chỉ các kênh được kết xuất-trước được nhận, tức là khi chế độ 2 được áp dụng trên bộ mã hóa âm thanh 3D của Fig.10. Theo cách thay thế, khi chế độ 1 được áp dụng trong bộ mã hóa âm thanh 3D, tức là, khi bộ mã hóa âm thanh 3D được thực hiện mã hóa kênh/đối tượng riêng lẻ, sau đó bộ xử lý đối tượng 1200 không được đi vòng, nhưng nhiều kênh được giải mã của các đối tượng được giải mã được nạp vào bộ xử lý đối tượng 1200 cùng với siêu dữ liệu được giải nén được tạo ra bởi bộ giải nén siêu dữ liệu 1400.

Ưu tiên, việc biểu thị hoặc chế độ 1 hoặc chế độ 2 được áp dụng để được chứa trong dữ liệu âm thanh được mã hóa và sau đó bộ điều khiển chế độ 1600 phân tích dữ liệu được mã hóa để phát hiện biểu thị chế độ. Chế độ 1 được sử dụng khi sự biểu thị chế độ biểu thị rằng dữ liệu âm thanh được mã hóa bao gồm các kênh được mã hóa và các đối tượng được mã hóa và chế độ 2 được áp dụng khi sự biểu thị chế độ biểu thị rằng dữ liệu âm thanh được mã hóa không chứa bất kỳ đối tượng âm thanh nào, tức là, chỉ chứa các kênh được kết xuất-trước thu được bởi chế độ 2 của bộ mã hóa âm thanh 3D của Fig.10.

Trên Fig.11, bộ giải nén siêu dữ liệu 1400 là bộ giải mã siêu dữ liệu 110 của thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh theo một trong số các phương án được mô tả ở trên. Ngoài ra, trên Fig.11, bộ giải mã lời 1300, bộ xử lý đối tượng 1200 và bộ xử lý sau 1700 cùng với nhau tạo thành bộ giải mã âm thanh 120 của thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh theo một trong số các phương án được mô tả ở trên.

Fig.13 minh họa phương án ưu tiên được so sánh với bộ giải mã âm thanh 3D của Fig.11 và phương án của Fig.13 tương ứng với bộ mã hóa âm thanh 3D của Fig.12. Ngoài sự thực hiện bộ giải mã âm thanh 3D của Fig.11, bộ giải mã âm thanh 3D trên Fig.13 bao gồm bộ giải mã SAOC 1800. Hơn nữa, bộ xử lý đối tượng 1200 của Fig.11 được thực hiện như bộ kết xuất đối tượng riêng biệt 1210 và bộ trộn 1220 trong khi đó, phụ thuộc vào chế độ, chức năng của bộ kết xuất đối tượng 1210 có thể cũng được thực hiện bởi bộ giải mã SAOC 1800.

Hơn nữa, bộ xử lý sau 1700 có thể được thực hiện như bộ kết xuất lập thể 1710 hoặc bộ chuyển đổi định dạng 1720. Theo cách thay thế, đầu ra trực tiếp của dữ liệu 1205 của Fig.11 có thể cũng được thực hiện như được minh họa bởi 1730. Do đó, ưu tiên để thực hiện xử lý trong bộ giải mã trên số lượng cao nhất của các kênh chẳng hạn như 22,2 hoặc 32 để có sự linh hoạt và sau đó xử lý-sau nếu định dạng nhỏ hơn được yêu cầu. Tuy nhiên, khi nó trở nên rõ ràng ngay từ đầu rằng chỉ có định dạng nhỏ như định dạng 5.1 được yêu cầu, sau đó ưu tiên, như được biểu thị bởi Fig.11 hoặc 6 biểu tượng tắt 1727, mà điều khiển nhất định trên bộ giải mã SAOC và/hoặc bộ giải mã

USAC có thể được áp dụng để tránh những vận hành trộn tăng không cần thiết và những vận hành trộn giảm sau đó.

Trong phương án được ưu tiên của sáng chế, bộ xử lý đối tượng 1200 bao gồm bộ giải mã SAOC 1800 và bộ giải mã SAOC được tạo cấu hình để giải mã một hoặc nhiều đầu ra kênh vận chuyển bằng bộ giải mã lõi và dữ liệu tham số được kết hợp và sử dụng siêu dữ liệu được giải nén để thu được nhiều đối tượng âm thanh được kết xuất. Để kết thúc, đầu ra OAM được nối với hộp 1800.

Hơn nữa, bộ xử lý đối tượng 1200 được tạo cấu hình để kết xuất đầu ra các đối tượng được giải mã bằng bộ giải mã lõi mà không được mã hóa trong các kênh vận chuyển SAOC nhưng được mã hóa riêng lẻ trong các phần tử được tạo kênh đơn đặc trưng như được biểu thị bởi bộ kết xuất đối tượng 1210. Hơn nữa, bộ giải mã bao gồm giao diện đầu ra tương ứng với đầu ra 1730 để xuất ra đầu ra của bộ trộn tới các loa phát thanh.

Trong phương án nữa, bộ xử lý đối tượng 1200 bao gồm bộ giải mã mã hóa đối tượng âm thanh không gian 1800 để giải mã một hoặc nhiều kênh vận chuyển và thông tin phụ theo tham số biểu diễn các tín hiệu âm thanh được mã hóa hoặc các kênh âm thanh được mã hóa, trong đó bộ giải mã mã hóa đối tượng âm thanh không gian được tạo cấu hình để chuyển mã thông tin theo tham số được kết hợp và siêu dữ liệu được giải nén thành thông tin phụ theo tham số được chuyển mã có thể sử dụng cho việc kết xuất trực tiếp định dạng đầu ra, như ví dụ được định nghĩa trong phiên bản sớm hơn của SAOC. Bộ xử lý sau 1700 được tạo cấu hình để tính toán các kênh âm thanh của định dạng đầu ra sử dụng các kênh vận chuyển được giải mã và thông tin phụ theo tham số được chuyển mã. Việc xử lý được thực hiện bởi bộ xử lý sau có thể tương tự với xử lý âm thanh nổi MPEG hoặc có thể là xử lý khác bất kỳ chẳng hạn như xử lý BCC hoặc như vậy.

Trong phương án nữa, bộ xử lý đối tượng 1200 bao gồm bộ giải mã mã hóa đối tượng âm thanh không gian 1800 được tạo cấu hình để trộn tăng trực tiếp và kết xuất các tín hiệu kênh cho định dạng đầu ra sử dụng các kênh vận chuyển được giải mã (bởi bộ giải mã lõi) và thông tin phụ theo tham số.

Hơn nữa, và quan trọng, bộ xử lý đối tượng 1200 của Fig.11 bao gồm một cách bổ sung bộ trộn 1220 mà nhận, như đầu vào, đầu ra dữ liệu bởi bộ giải mã USAC 1300 trực tiếp khi các đối tượng được kết xuất-trước được trộn với kênh tồn tại, tức là, khi bộ trộn 200 của Fig.10 được hoạt động. Ngoài ra, bộ trộn 1220 nhận dữ liệu từ bộ kết xuất đối tượng thực hiện kết xuất đối tượng không gồm giải mã SAOC. Hơn nữa, bộ trộn nhận dữ liệu đầu ra bộ giải mã SAOC, tức là, các đối tượng được kết xuất SAOC.

Bộ trộn 1220 được nối với giao diện đầu ra 1730, bộ kết xuất lập thể 1710 và bộ chuyển đổi định dạng 1720. Bộ kết xuất lập thể 1710 được tạo cấu hình để kết xuất các kênh đầu ra thành hai kênh lập thể sử dụng các chức năng chuyển đổi liên quan đứng đầu hoặc các đáp ứng xung trong phòng lập thể (binaural room impulse responses-BRIR). Bộ chuyển đổi định dạng 1720 được tạo cấu hình để chuyển đổi các kênh đầu ra thành định dạng đầu ra có số lượng kênh thấp hơn các kênh đầu ra 1205 của bộ trộn và bộ chuyển đổi định dạng 1720 yêu cầu thông tin trên sự bố trí tái tạo như các loa 5.1 hoặc như vậy.

Trên Fig.13, bộ giải mã-OAM 1400 là bộ giải mã siêu dữ liệu 110 của thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh theo một trong số các phương án được mô tả ở trên. Ngoài ra, trên Fig.13, bộ kết xuất đối tượng 1210, bộ giải mã USAC 1300 và bộ trộn 1220 cùng với nhau tạo thành bộ giải mã âm thanh 120 của thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh theo một trong số các phương án được mô tả ở trên.

Bộ giải mã âm thanh 3D của Fig.15 khác với bộ giải mã âm thanh 3D của Fig.13 mà trong đó bộ giải mã SAOC không thể chỉ tạo ra các đối tượng được kết xuất và còn tạo ra các kênh được kết xuất và đó là trường hợp khi bộ mã hóa âm thanh 3D của Fig.14 được sử dụng và sự nối 900 giữa các kênh/các đối tượng được kết xuất-trước và giao diện đầu vào bộ mã hóa SAOC 800 được hoạt động.

Hơn nữa, giai đoạn Quét biên độ gốc vectơ (Vector Base Amplitude Panning-VBAP) 1810 được tạo cấu hình mà nhận, từ bộ giải mã SAOC, thông tin về bố trí tái tạo và mà xuất ra ma trận kết xuất tới bộ giải mã SAOC sao cho bộ giải mã SAOC có thể, cuối cùng, cung cấp các kênh được kết xuất mà không có bất kỳ vận hành nào thêm nữa của bộ trộn trong định dạng kênh cao 1205, tức là, các loa phát thanh 32.

khối VBAP ưu tiên nhận dữ liệu OAM được giải mã để rút ra các ma trận kết xuất. Tổng quát hơn, ưu tiên thông tin hình học yêu cầu không chỉ của bố trí tái tạo mà còn của các vị trí mà các tín hiệu đầu vào cần được kết xuất lên trên bố trí tái tạo. Dữ liệu đầu vào hình học này có thể là dữ liệu OAM cho các đối tượng hoặc thông tin vị trí kênh cho các kênh mà được truyền dẫn sử dụng SAOC.

Tuy nhiên, nếu chỉ giao diện đầu ra cụ thể được yêu cầu sau giai đoạn VBAP 1810 có thể đã cung cấp ma trận kết xuất được yêu cầu các ví dụ đầu ra 5.1. Bộ giải mã SAOC 1800 sau đó thực hiện kết xuất trực tiếp từ các kênh vận chuyển SAOC, dữ liệu tham số được kết hợp và siêu dữ liệu được giải nén, kết xuất trực tiếp thành định dạng đầu ra được yêu cầu mà không có bất kỳ tương tác của bộ trộn 1220. Tuy nhiên, khi trộn nhất định giữa các chế độ được áp dụng, tức là, tại đó nhiều kênh được mã hóa SAOC nhưng không phải tất cả các kênh được mã hóa SAOC hoặc tại đó nhiều đối tượng được mã hóa SAOC nhưng không phải tất cả các đối tượng được mã hóa SAOC hoặc khi chỉ lượng nhất định các đối tượng được kết xuất-trước với các kênh được giải mã SAOC và các kênh còn lại không được xử lý SAOC sau đó bộ trộn sẽ đặt cùng với dữ liệu từ các phần đầu vào riêng lẻ, tức trực tiếp từ bộ giải mã lõi 1300, từ bộ kết xuất đối tượng 1210 và từ bộ giải mã SAOC 1800.

Trên Fig.15, bộ giải mã-OAM 1400 là bộ giải mã siêu dữ liệu 110 của thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh theo một trong số các phương án được mô tả ở trên. Ngoài ra, trên Fig.15, bộ kết xuất đối tượng 1210, bộ giải mã USAC 1300 và bộ trộn 1220 cùng với nhau tạo thành bộ giải mã âm thanh 120 của thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh theo một trong số các phương án được mô tả ở trên.

Thiết bị giải mã dữ liệu âm thanh được mã hóa được đề xuất. Thiết bị giải mã dữ liệu âm thanh được mã hóa bao gồm:

- giao diện đầu vào 1100 để nhận dữ liệu âm thanh được mã hóa, dữ liệu âm thanh được mã hóa bao gồm nhiều kênh được mã hóa hoặc nhiều đối tượng được mã hóa hoặc nén siêu dữ liệu liên quan đến nhiều đối tượng, và
- thiết bị 100 bao gồm bộ giải mã siêu dữ liệu 110 và bộ tạo kênh âm thanh 120 để tạo ra một hoặc nhiều kênh âm thanh như được mô tả ở trên.

Bộ giải mã siêu dữ liệu 110 của thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh là bộ giải nén siêu dữ liệu 400 để giải nén siêu dữ liệu được nén.

Bộ tạo kênh âm thanh 120 của thiết bị 100 để tạo ra một hoặc nhiều kênh âm thanh bao gồm bộ giải mã lõi 1300 để giải mã nhiều kênh được mã hóa và nhiều đối tượng được mã hóa.

Ngoài ra, bộ tạo kênh âm thanh 120 còn bao gồm bộ xử lý đối tượng 1200 để xử lý nhiều đối tượng được giải mã sử dụng siêu dữ liệu được giải nén để thu được số lượng kênh đầu ra 1205 bao gồm dữ liệu âm thanh từ các đối tượng và các kênh được giải mã.

Hơn nữa, bộ tạo kênh âm thanh 120 còn bao gồm bộ xử lý sau 1700 để chuyển đổi số kênh đầu ra 1205 thành định dạng đầu ra.

Mặc dù một số khía cạnh đã được mô tả trong phạm vi của thiết bị, rõ ràng rằng các khía cạnh này cũng thể hiện sự mô tả của phương pháp tương ứng, trong đó khối hoặc thiết bị tương ứng với bước phương pháp hoặc đặc tính của bước phương pháp. Tương tự, các khía cạnh được mô tả trong phạm vi của bước phương pháp cũng biểu diễn sự mô tả của khối hoặc mục hoặc dấu hiệu tương ứng của thiết bị tương ứng.

Tín hiệu đã được tách ra có thể được lưu trữ trên vật ghi lưu trữ số hoặc có thể được truyền dẫn trên vật ghi truyền dẫn như vật ghi truyền dẫn không dây hoặc vật ghi truyền dẫn có dây như Liên mạng.

Phụ thuộc vào các yêu cầu thực hiện nhất định, các phương án của sáng chế có thể được thực hiện trong phần cứng hoặc trong phần mềm. Phương án có thể được thực hiện sử dụng vật ghi lưu trữ số, ví dụ, đĩa mềm, DVD, CD, ROM, PROM, EPROM, 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 thể kết hợp) với hệ thống máy tính lập trình được sao cho phương pháp tương ứng được thực hiện.

Một số phương án theo sáng chế gồm có vật mang dữ liệu không tạm thời có các tín hiệu điều khiển đọc được nhờ điện tử, mà có khả năng kết hợp với hệ thống máy tính lập trình được, do đó một trong các phương pháp được mô tả ở đây được thực hiện.

hông thường, các phương án theo sáng chế có thể được thực hiện như sản phẩm chương trình máy tính với mã chương trình, mã chương trình hoạt động để thực hiện một trong các phương pháp khi sản phẩm chương trình máy tính chạy trên máy tính. Mã chương trình có thể ví dụ được lưu trữ trên vật mang có thể đọc được bằng máy.

Các phương án 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.

Nói cách khác, một phương án của phương pháp theo sáng chế là, chương trình máy tính có mã chương trình để thực hiện một trong các phương pháp được mô tả ở đây, khi chương trình máy tính chạy trên máy tính.

Phương án khác của các phương pháp theo sáng chế là, vật mang dữ liệu (hoặc môi trường lưu trữ số, hoặc vật ghi có thể đọc được bằng máy tính) gồm có, đã được ghi lại trên đó, chương trình máy tính để thực hiện một trong các phương pháp được mô tả ở đây.

Phương án khác của các phương pháp theo sáng chế là, dòng dữ liệu hoặc chuỗi tín hiệu thể hiện chương trình máy tính để thực hiện một trong các phương pháp đã được mô tả ở đây. Ví dụ, có thể cấu hình dòng dữ liệu hoặc chuỗi tín hiệu để được truyền thông qua sự kết nối truyền thông dữ liệu, ví dụ thông qua Liên mạng.

Phương án khác gồm có phương tiện xử lý, ví dụ, máy tính, hoặc thiết bị logic lập trình được tạo cấu hình để hoặc được làm thích ứng để thực hiện một trong các phương pháp được mô tả ở đây.

Phương án nữa bao gồm máy tính có chương trình máy tính được cài đặt trên máy tính để thực hiện một trong các phương pháp được mô tả ở đây.

Theo một số phương án, 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. Theo một số phương án, 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 bất kỳ.

Các phương án được mô tả ở trên chỉ mang tính minh họa cho các nguyên lý của sáng chế. Cần hiểu rằng các sự biến thể và biến đổi của các phương án và các chi tiết được mô tả ở đây sẽ là rõ ràng đối với người có trình độ trong lĩnh vực kỹ thuật tương ứng. Do đó, 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 phương án ở đây.

YÊU CẦU BẢO HỘ

1. Thiết bị tạo ra một hoặc nhiều kênh âm thanh (100), trong đó thiết bị bao gồm:

bộ giải mã siêu dữ liệu (110; 901) để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ từ một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $(z_1, \dots, z_N)$ phụ thuộc vào tín hiệu điều khiển (b), trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh, trong đó bộ giải mã siêu dữ liệu (110; 901) được tạo cấu hình để tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ bằng cách xác định nhiều mẫu siêu dữ liệu được khôi phục $(x_1'(n), \dots, x_N'(n))$ cho mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$, và

bộ tạo kênh âm thanh (120) để tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$,

trong đó bộ giải mã siêu dữ liệu (110; 901) được tạo cấu hình để nhận nhiều mẫu siêu dữ liệu được xử lý $(z_1(n), \dots, z_N(n))$ của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $(z_1, \dots, z_N)$,

trong đó bộ giải mã siêu dữ liệu (110; 901) được tạo cấu hình để nhận tín hiệu điều khiển (b),

trong đó bộ giải mã siêu dữ liệu (110; 901) được tạo cấu hình để xác định mỗi mẫu siêu dữ liệu được khôi phục $(x_i'(n))$ trong số nhiều mẫu siêu dữ liệu được khôi phục $(x_i'(1), \dots, x_i'(n-1), x_i'(n))$ của mỗi tín hiệu siêu dữ liệu được khôi phục (x_i') trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$, sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục $(x_i'(n))$ đã nêu là tổng của một trong số các mẫu siêu dữ liệu được xử lý $(z_i(n))$ của một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (z_i) và mẫu siêu dữ liệu được khôi phục đã được tạo ra khác $(x_i'(n-1))$ của tín hiệu siêu dữ liệu được khôi phục đã nêu (x_i') , và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được khôi phục $(x_i'(n))$ đã nêu

là một mẫu ($z_i(n)$) đã nêu trong số các mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n)$) của một tín hiệu (z_i) đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$).

2. Thiết bị (100) theo điểm 1,

trong đó bộ giải mã siêu dữ liệu (110; 901) được tạo cấu hình để nhận hai hoặc nhiều hơn các tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), và được tạo cấu hình để tạo ra hai hoặc nhiều hơn các tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$),

trong đó bộ giải mã siêu dữ liệu (110; 901) bao gồm hai hoặc nhiều hơn các khối con của bộ giải mã siêu dữ liệu (911, ..., 91N),

trong đó mỗi khối con (91i; 91i') trong số hai hoặc nhiều hơn các khối con của bộ giải mã siêu dữ liệu (911, ..., 91N) được tạo cấu hình bao gồm bộ cộng (910) và bộ chọn (930),

trong đó mỗi khối con (91i; 91i') trong số hai hoặc nhiều hơn các khối con của bộ giải mã siêu dữ liệu (911, ..., 91N) được tạo cấu hình để nhận nhiều mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n-1), z_i(n)$) của một tín hiệu (z_i) trong số hai hoặc nhiều hơn các tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), và được tạo cấu hình để tạo ra một tín hiệu (z_i) trong số hai hoặc nhiều hơn các tín hiệu siêu dữ liệu được khôi phục ($z_1, \dots, z_N$),

trong đó bộ cộng (910) của khối con của bộ giải mã siêu dữ liệu đã nêu (91i; 91i') được tạo cấu hình để cộng một mẫu ($z_i(n)$) trong số các mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n)$) của một tín hiệu (z_i) trong số hai hoặc nhiều hơn các tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$) và mẫu siêu dữ liệu được khôi phục đã được tạo ra khác ($x_i'(n-1)$) của một tín hiệu (z_i) đã nêu trong số hai hoặc nhiều hơn các tín hiệu siêu dữ liệu được khôi phục ($z_1, \dots, z_N$), để thu được giá trị tổng ($s_i(n)$), và

trong đó bộ chọn (930) của khối con của bộ giải mã siêu dữ liệu đã nêu (91i; 91i') được tạo cấu hình để nhận một mẫu đã nêu trong số các mẫu siêu dữ liệu được xử lý ($z_i(n)$), giá trị tổng đã nêu ($s_i(n)$) và tín hiệu điều khiển, và trong đó bộ chọn (930) đã nêu được tạo cấu hình để xác định một trong số nhiều mẫu siêu dữ liệu ($x_i'(1), \dots, x_i'(n-1), x_i'(n)$) của tín hiệu siêu dữ liệu được khôi phục đã nêu (x_i') sao cho,

khi tín hiệu điều khiển b biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục đã nêu ($x_i'(n)$) là giá trị tổng ($s_i(n)$), và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$), mẫu siêu dữ liệu được khôi phục đã nêu ($x_i'(n)$) là một mẫu ($z_i(n)$) đã nêu trong số các mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n)$).

3. Thiết bị (100) theo điểm 1 hoặc điểm 2,

trong đó ít nhất một tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) biểu thị thông tin vị trí về một trong số một hoặc nhiều tín hiệu đối tượng âm thanh, và

trong đó bộ tạo kênh âm thanh (120) được tạo cấu hình để tạo ra ít nhất một trong số một hoặc nhiều kênh âm thanh phụ thuộc vào một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào thông tin vị trí đã nêu.

4. Thiết bị (100) theo một điểm trong số các điểm nêu trên,

trong đó ít nhất một tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) biểu thị âm lượng của một trong số một hoặc nhiều tín hiệu đối tượng âm thanh, và

trong đó bộ tạo kênh âm thanh (120) được tạo cấu hình để tạo ra ít nhất một trong số một hoặc nhiều kênh âm thanh phụ thuộc vào một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào âm lượng đã nêu.

5. Thiết bị giải mã dữ liệu âm thanh được mã hóa, bao gồm:

giao diện đầu vào (1100) để nhận dữ liệu âm thanh được mã hóa, dữ liệu âm thanh được mã hóa bao gồm nhiều kênh được mã hóa hoặc nhiều đối tượng được mã hóa hoặc siêu dữ liệu nén liên quan đến nhiều đối tượng, và

thiết bị (100) theo một điểm trong số các điểm từ 1 đến 4,

trong đó bộ giải mã siêu dữ liệu (110;901) của thiết bị (100) theo một điểm trong số các điểm từ 1 đến 4 là bộ giải nén siêu dữ liệu (400) để giải nén siêu dữ liệu được nén,

trong đó bộ tạo kênh âm thanh (120) của thiết bị (100) theo một điểm trong số các điểm từ 1 đến 4 bao gồm bộ giải mã lỗi (1300) để giải mã nhiều kênh được mã hóa và nhiều đối tượng được mã hóa,

trong đó bộ tạo kênh âm thanh (120) còn bao gồm bộ xử lý đối tượng (1200) để xử lý nhiều đối tượng được giải mã sử dụng siêu dữ liệu được giải nén để thu được số kênh đầu ra (1205) bao gồm dữ liệu âm thanh từ các đối tượng và các kênh được giải mã, và

trong đó bộ tạo kênh âm thanh (120) còn bao gồm bộ xử lý sau (1700) để chuyển đổi số kênh đầu ra (1205) thành định dạng đầu ra.

6. Thiết bị tạo ra thông tin âm thanh được mã hóa (250) bao gồm một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, trong đó thiết bị bao gồm:

bộ mã hóa siêu dữ liệu (210; 801; 802) để nhận một hoặc nhiều tín hiệu siêu dữ liệu ban đầu và để xác định một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu bao gồm nhiều mẫu siêu dữ liệu ban đầu, trong đó các mẫu siêu dữ liệu ban đầu của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh, và

bộ mã hóa âm thanh (220) để mã hóa một hoặc nhiều tín hiệu đối tượng âm thanh để thu được một hoặc nhiều tín hiệu âm thanh được mã hóa,

trong đó bộ mã hóa siêu dữ liệu (210; 801; 802) được tạo cấu hình để xác định từng mẫu siêu dữ liệu được xử lý ($z_i(n)$) trong số nhiều mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n-1), z_i(n)$) của mỗi tín hiệu siêu dữ liệu được xử lý (z_i) trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($z_i(n)$) đã nêu biểu thị chênh lệch hoặc chênh lệch được lượng tử hóa giữa một trong số nhiều mẫu siêu dữ liệu ban đầu ($x_i(n)$) của một trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu (x_i) và mẫu siêu dữ liệu được xử lý đã được tạo ra khác của tín hiệu siêu dữ liệu được xử lý (z_i) đã nêu, và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai

($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được xử lý ($z_i(n)$) đã nêu là một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$) của một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (x_i), hoặc là sự biểu diễn được lượng tử hóa ($q_i(n)$) của một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$).

7. Thiết bị (250) theo điểm 6,

trong đó bộ mã hóa siêu dữ liệu (210; 801; 802) được tạo cấu hình để nhận hai hoặc nhiều hơn các tín hiệu siêu dữ liệu ban đầu ($x_1, \dots, x_N$), và được tạo cấu hình để tạo ra hai hoặc nhiều hơn các tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$),

trong đó bộ mã hóa siêu dữ liệu (210; 801; 802) bao gồm hai hoặc nhiều hơn các bộ mã hóa DCPM (811, ..., 81N),

trong đó mỗi bộ mã hóa trong số hai hoặc nhiều hơn các bộ mã hóa DCPM (811, ..., 81N) được tạo cấu hình để xác định chênh lệch hoặc chênh lệch được lượng tử hóa giữa một mẫu ($x_i(n)$) trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$) của một tín hiệu (x_i) trong số hai hoặc nhiều hơn các tín hiệu siêu dữ liệu ban đầu ($x_1, \dots, x_N$) và mẫu siêu dữ liệu được xử lý đã được tạo ra khác của một tín hiệu (z_i) trong số hai hoặc nhiều hơn các tín hiệu siêu dữ liệu được khôi phục ($z_1, \dots, z_N$), để thu được mẫu chênh lệch ($y_i(n)$), và

trong đó bộ mã hóa siêu dữ liệu (210; 801; 802) còn bao gồm bộ chọn (830) được tạo cấu hình để xác định một trong số nhiều mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n-1), z_i(n)$) của tín hiệu siêu dữ liệu được xử lý đã nêu (z_i) sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được xử lý đã nêu ($y_i(n)$) là mẫu chênh lệch ($y_i(n)$), và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$), mẫu siêu dữ liệu được xử lý đã nêu ($z_i(n)$) là một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$) hoặc là biểu diễn được lượng tử hóa ($q_i(n)$) của một mẫu ($x_i(n)$) trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$).

8. Thiết bị (250) theo điểm 6 hoặc điểm 7,

trong đó ít nhất một tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu biểu thị thông tin vị trí về một trong số một hoặc nhiều tín hiệu đối tượng âm thanh, và

trong đó bộ mã hóa siêu dữ liệu (210; 801; 802) được tạo cấu hình để tạo ra ít nhất một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý phụ thuộc vào ít nhất một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu mà biểu thị thông tin vị trí đã nêu.

9. Thiết bị (250) theo một điểm trong số các điểm từ 6 đến 8,

trong đó ít nhất một tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu biểu thị âm lượng của một trong số một hoặc nhiều tín hiệu đối tượng âm thanh, và

trong đó bộ mã hóa siêu dữ liệu (210; 801; 802) được tạo cấu hình để tạo ra ít nhất một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý phụ thuộc vào ít nhất một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu mà biểu thị thông tin vị trí đã nêu.

10. Thiết bị (250) theo một điểm trong số các điểm từ 6 đến 9, trong đó bộ mã hóa siêu dữ liệu (210; 801; 802) được tạo cấu hình để mã hóa mỗi mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n)$) của một tín hiệu $z_i()$ trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$) với số bit thứ nhất khi tín hiệu điều khiển biểu thị trạng thái thứ nhất ($b(n)=0$), và với số bit thứ hai khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$), trong đó số bit thứ nhất nhỏ hơn số bit thứ hai.

11. Thiết bị mã hóa dữ liệu đầu vào âm thanh (101) để thu được dữ liệu đầu ra âm thanh (501), bao gồm:

giao diện đầu vào (1100) để nhận nhiều kênh âm thanh, nhiều đối tượng âm thanh và siêu dữ liệu liên quan đến một hoặc nhiều trong số nhiều đối tượng âm thanh,

bộ trộn (200) để trộn nhiều đối tượng và nhiều kênh để thu được nhiều kênh được trộn-trước, mỗi kênh được trộn-trước bao gồm dữ liệu âm thanh của kênh và dữ liệu âm thanh của ít nhất một đối tượng, và

thiết bị (250) theo một điểm trong số các điểm từ 6 đến 10,

trong đó bộ mã hóa âm thanh (220) của thiết bị (250) theo một điểm trong số các điểm từ 6 đến 10 là bộ mã hóa lỗi (300) để mã hóa lỗi dữ liệu đầu vào bộ mã hóa lỗi, và

trong đó bộ mã hóa siêu dữ liệu (210; 801; 802) của thiết bị (250) theo một điểm trong số các điểm từ 6 đến 10 là bộ nén siêu dữ liệu (400) để nén siêu dữ liệu liên quan đến một hoặc nhiều trong số nhiều đối tượng âm thanh.

12. Hệ thống tạo ra thông tin âm thanh được mã hóa và tạo ra một hoặc nhiều kênh âm thanh, bao gồm:

thiết bị (250) theo một điểm trong số các điểm từ 6 đến 10 để tạo ra thông tin âm thanh được mã hóa bao gồm một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, và

thiết bị (100) theo một điểm trong số các điểm từ 1 đến 4 để nhận một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, và để tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu âm thanh được mã hóa và phụ thuộc vào một hoặc nhiều tín hiệu siêu dữ liệu được xử lý.

13. Phương pháp tạo ra một hoặc nhiều kênh âm thanh, trong đó phương pháp bao gồm:

tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ từ một hoặc nhiều tín hiệu siêu dữ liệu được xử lý $(z_1, \dots, z_N)$ phụ thuộc vào tín hiệu điều khiển (b), trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh, trong đó tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$ được điều khiển bằng cách xác định nhiều mẫu siêu dữ liệu được khôi phục $(x_1'(n), \dots, x_N'(n))$ cho mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$, và

tạo ra một hoặc nhiều kênh âm thanh phụ thuộc vào một hoặc nhiều tín hiệu đối tượng âm thanh và phụ thuộc vào một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục $(x_1', \dots, x_N')$,

trong đó tạo ra một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$) được điều khiển bằng cách nhận nhiều mẫu siêu dữ liệu được xử lý ($z_1(n), \dots, z_N(n)$) của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), bằng cách nhận tín hiệu điều khiển (b), và bằng cách xác định từng mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) trong số nhiều mẫu siêu dữ liệu được khôi phục ($x_i'(1), \dots, x_i'(n-1), x_i'(n)$) của mỗi tín hiệu siêu dữ liệu được khôi phục (x_i') trong số một hoặc nhiều tín hiệu siêu dữ liệu được khôi phục ($x_1', \dots, x_N'$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) đã nêu là tổng của một trong số các mẫu siêu dữ liệu được xử lý ($z_i(n)$) của một trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (z_i) và mẫu siêu dữ liệu được khôi phục đã được tạo ra khác ($x_i'(n-1)$) của tín hiệu siêu dữ liệu được khôi phục đã nêu (x_i'), và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được khôi phục ($x_i'(n)$) đã nêu là một mẫu ($z_i(n)$) đã nêu trong số các mẫu siêu dữ liệu được xử lý ($z_i(1), \dots, z_i(n)$) của một tín hiệu (z_i) đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$).

14. Phương pháp tạo ra thông tin âm thanh được mã hóa bao gồm một hoặc nhiều tín hiệu âm thanh được mã hóa và một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, trong đó phương pháp bao gồm:

nhận một hoặc nhiều tín hiệu siêu dữ liệu ban đầu,
xác định một hoặc nhiều tín hiệu siêu dữ liệu được xử lý, và
mã hóa một hoặc nhiều tín hiệu đối tượng âm thanh để thu được một hoặc nhiều tín hiệu âm thanh được mã hóa,

trong đó mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu bao gồm nhiều mẫu siêu dữ liệu ban đầu, trong đó các mẫu siêu dữ liệu ban đầu của mỗi tín hiệu trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu biểu thị thông tin được kết hợp với tín hiệu đối tượng âm thanh của một hoặc nhiều tín hiệu đối tượng âm thanh, và

trong đó xác định một hoặc nhiều tín hiệu siêu dữ liệu được xử lý bao gồm xác định mỗi mẫu siêu dữ liệu được xử lý ($z_i(n)$) trong số nhiều mẫu siêu dữ liệu được xử

lý ($z_i(1), \dots, z_i(n-1), z_i(n)$) của mỗi tín hiệu siêu dữ liệu được xử lý (z_i) trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý ($z_1, \dots, z_N$), sao cho, khi tín hiệu điều khiển (b) biểu thị trạng thái thứ nhất ($b(n)=0$), mẫu siêu dữ liệu được khôi phục ($z_i(n)$) đã nêu biểu thị chênh lệch hoặc chênh lệch được lượng tử hóa giữa một trong số nhiều mẫu siêu dữ liệu ban đầu ($x_i(n)$) của một trong số một hoặc nhiều tín hiệu siêu dữ liệu ban đầu (x_i) và mẫu siêu dữ liệu được xử lý đã được tạo ra khác của tín hiệu siêu dữ liệu được xử lý (z_i) đã nêu, và sao cho, khi tín hiệu điều khiển biểu thị trạng thái thứ hai ($b(n)=1$) khác với trạng thái thứ nhất, mẫu siêu dữ liệu được xử lý ($z_i(n)$) đã nêu là một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$) của một tín hiệu đã nêu trong số một hoặc nhiều tín hiệu siêu dữ liệu được xử lý (x_i), hoặc là sự biểu diễn được lượng tử hóa ($q_i(n)$) của một mẫu ($x_i(n)$) đã nêu trong số các mẫu siêu dữ liệu ban đầu ($x_i(1), \dots, x_i(n)$).

15. Vật ghi có thể đọc được bằng máy tính bao gồm chương trình máy tính để thực hiện phương pháp theo điểm 13, khi được chạy trên máy tính hoặc bộ xử lý tín hiệu.

16. Vật ghi có thể đọc được bằng máy tính bao gồm chương trình máy tính để thực hiện phương pháp theo điểm 14, khi được chạy trên máy tính hoặc bộ xử lý tín hiệu.

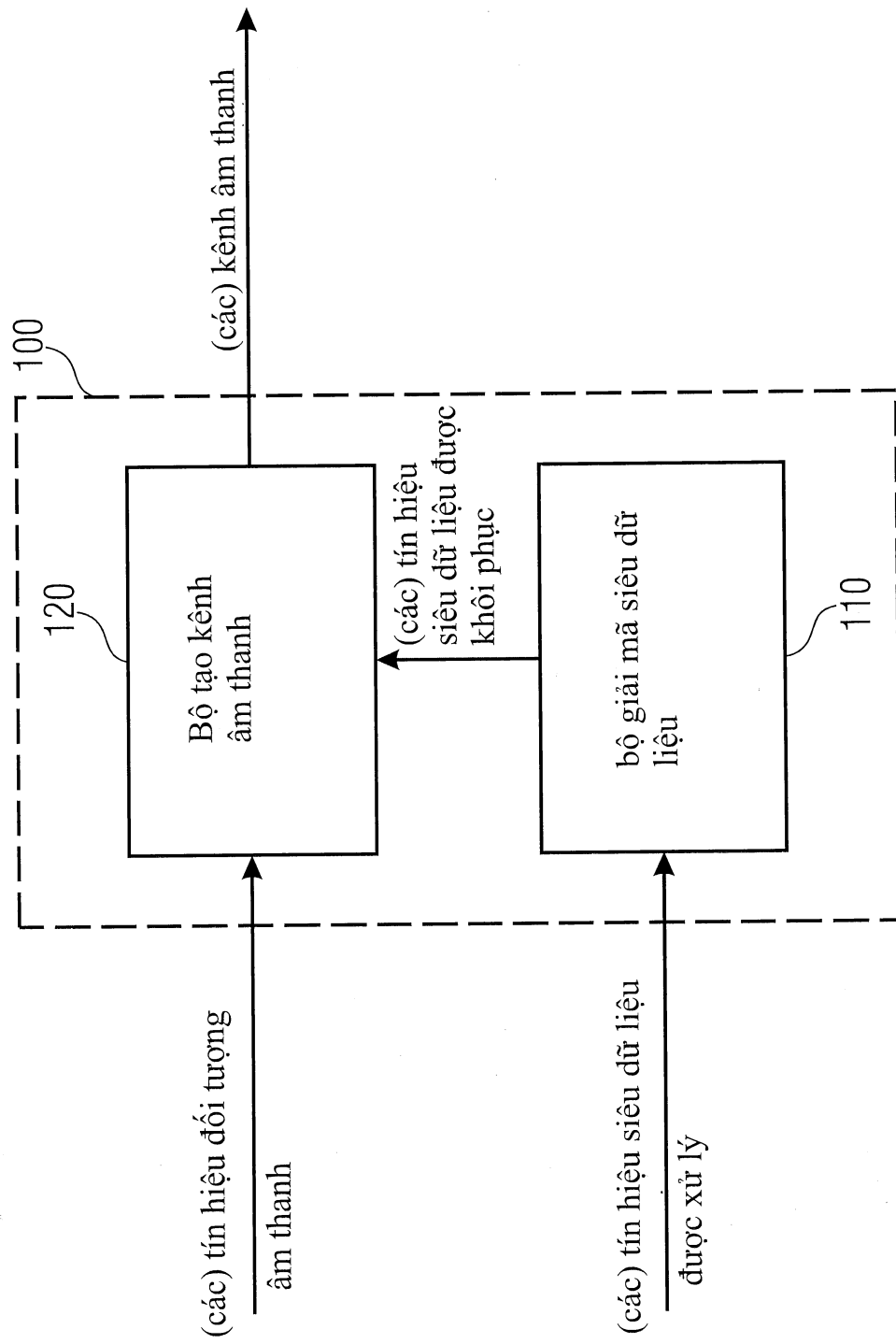


FIGURE 1

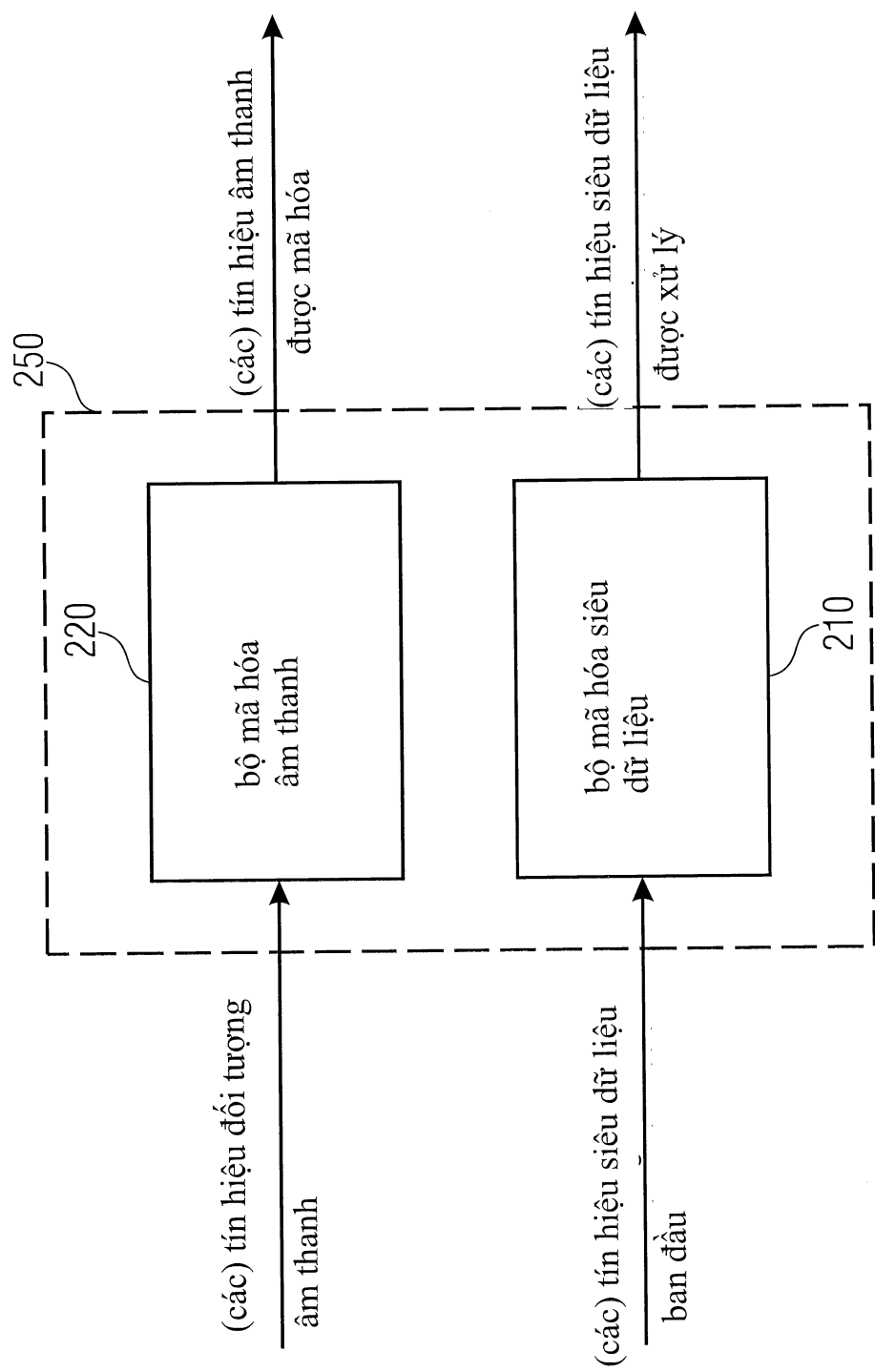


FIGURE 2

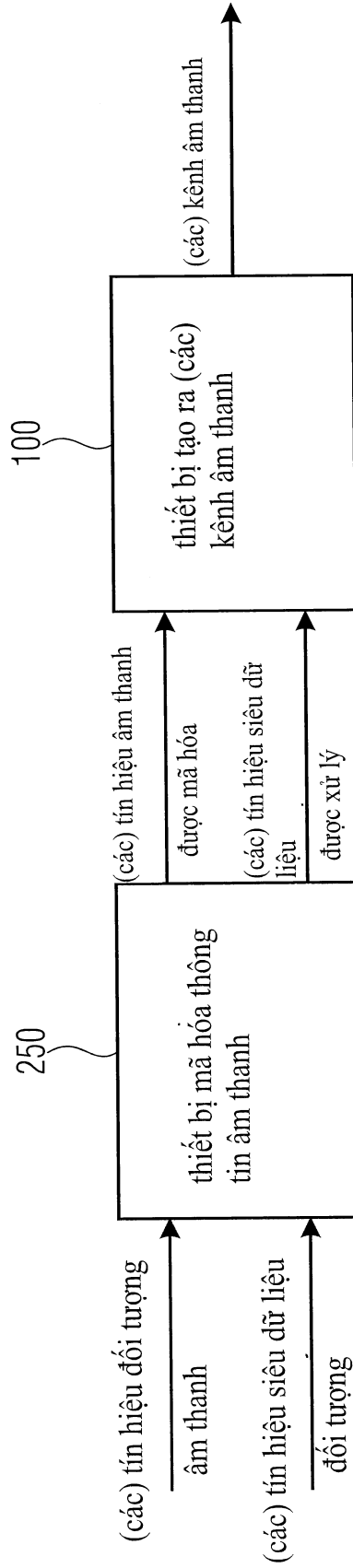


FIGURE 3

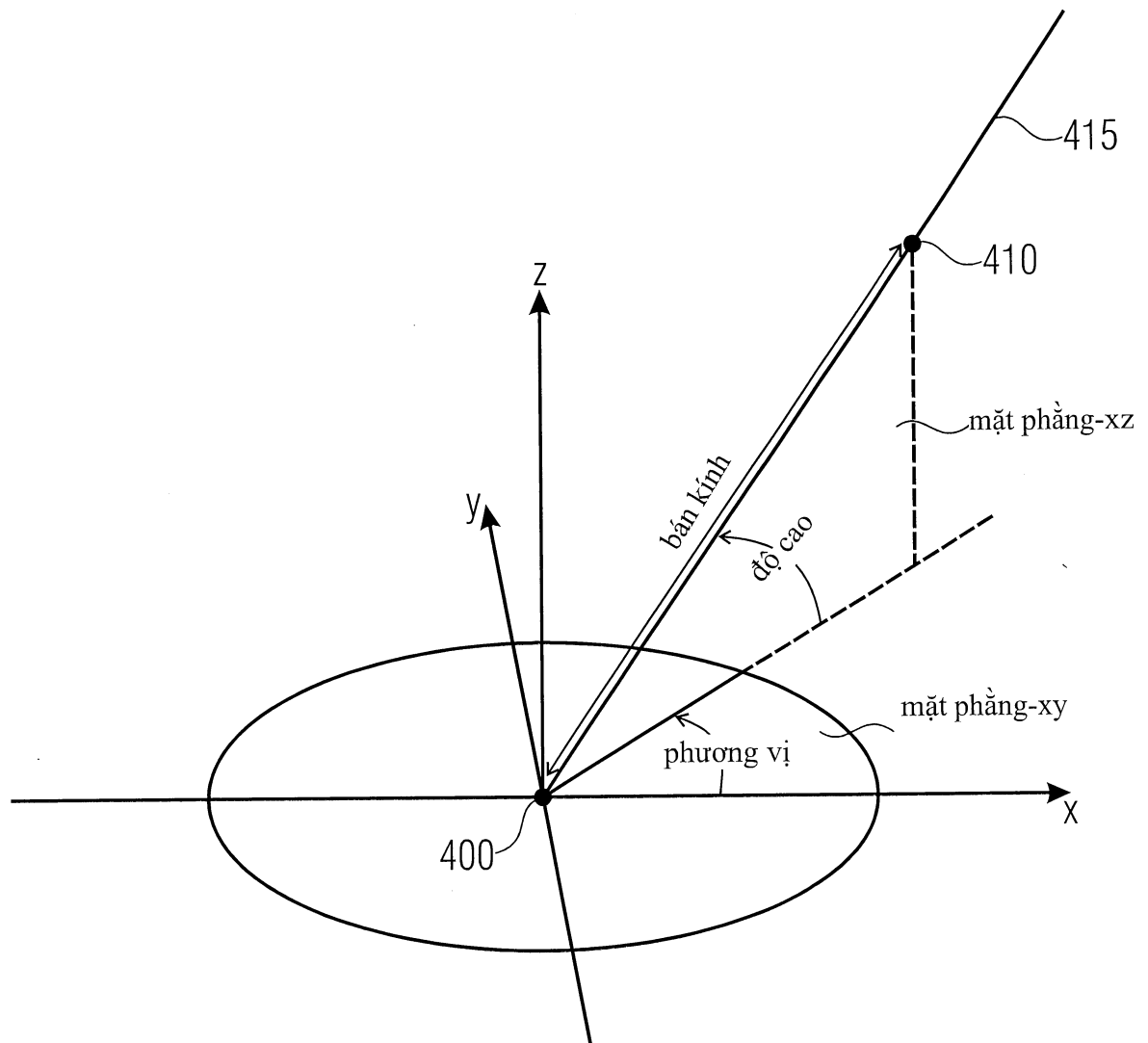


FIGURE 4

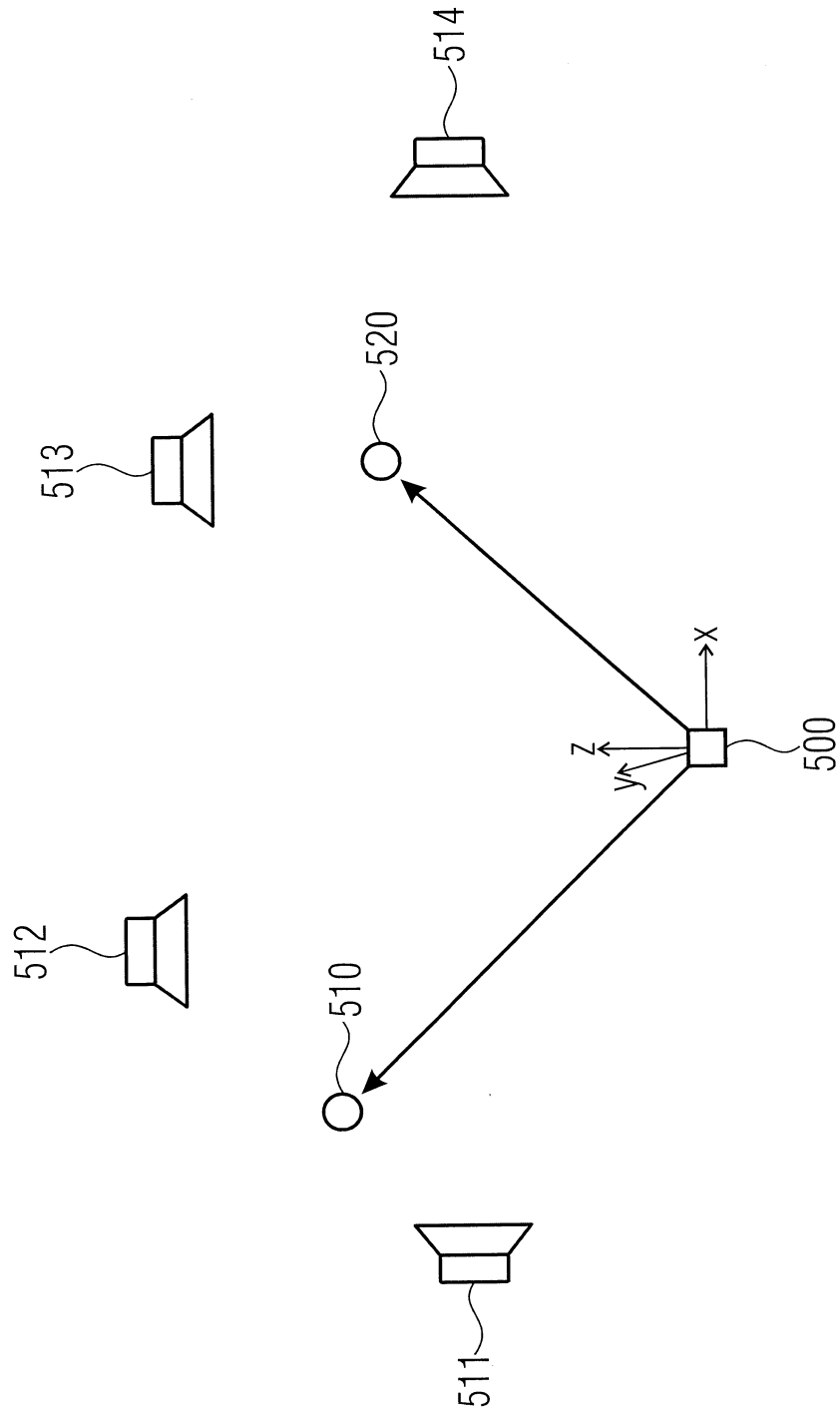


FIGURE 5

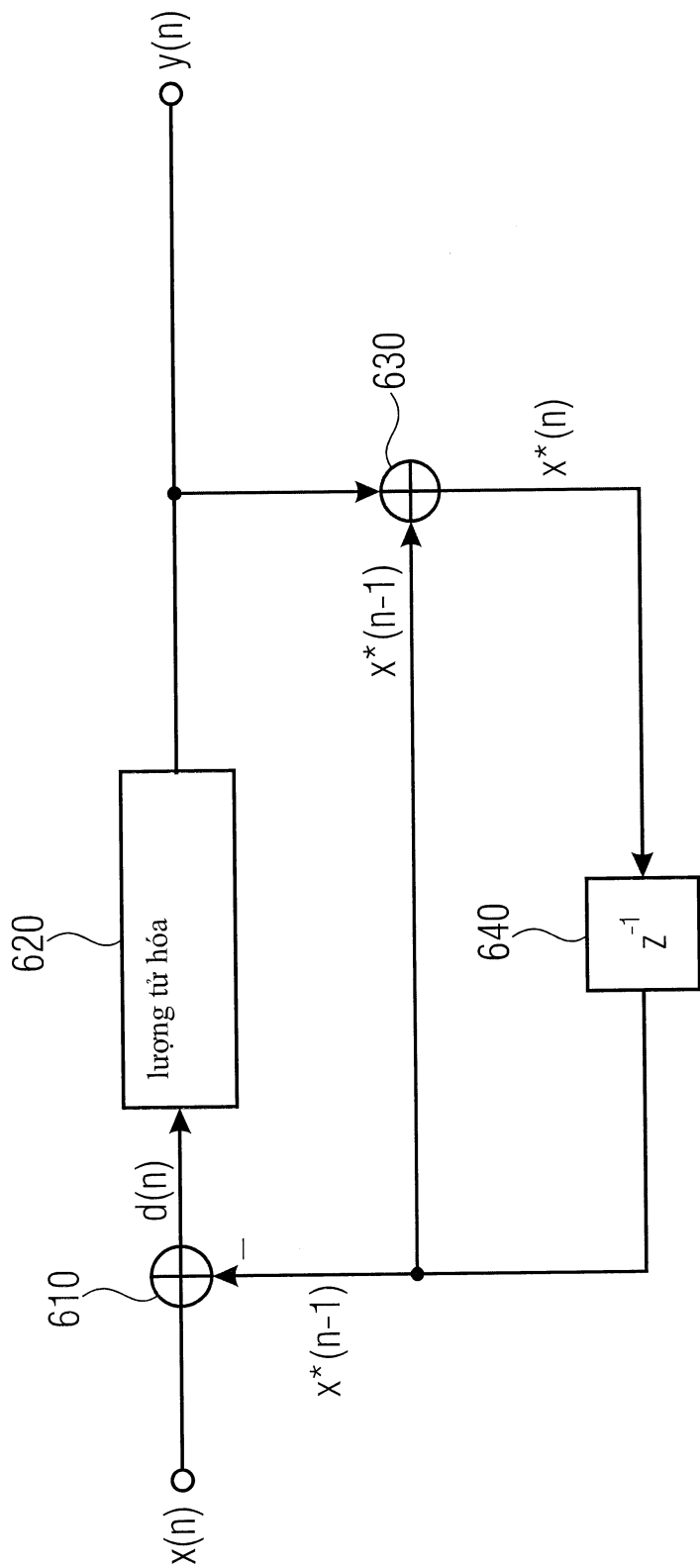


FIGURE 6

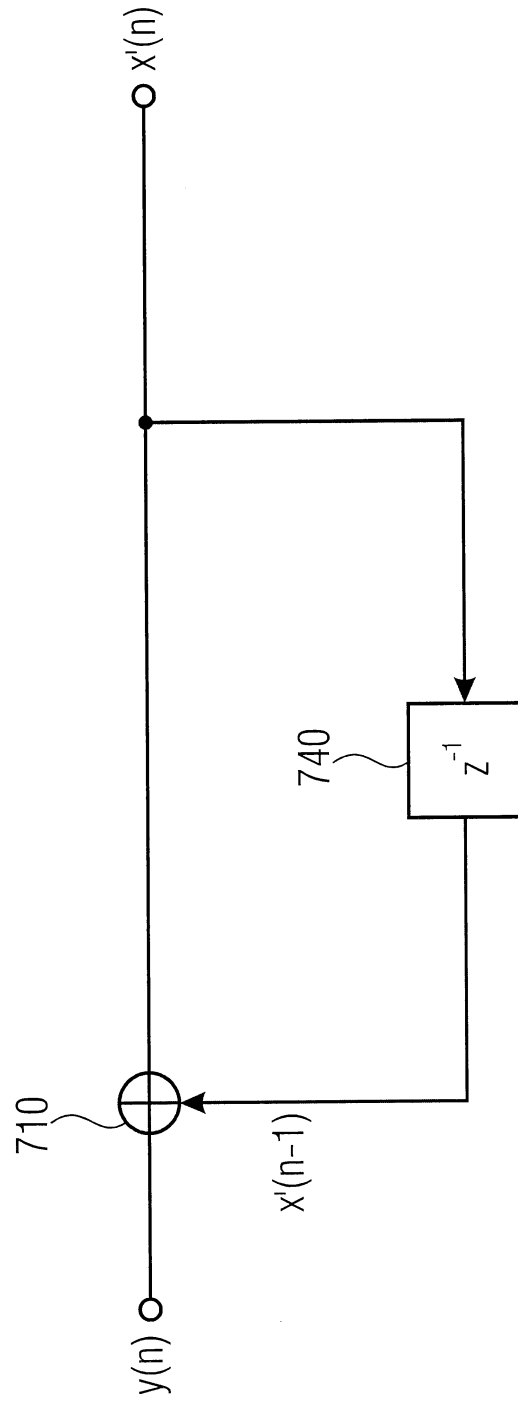


FIGURE 7

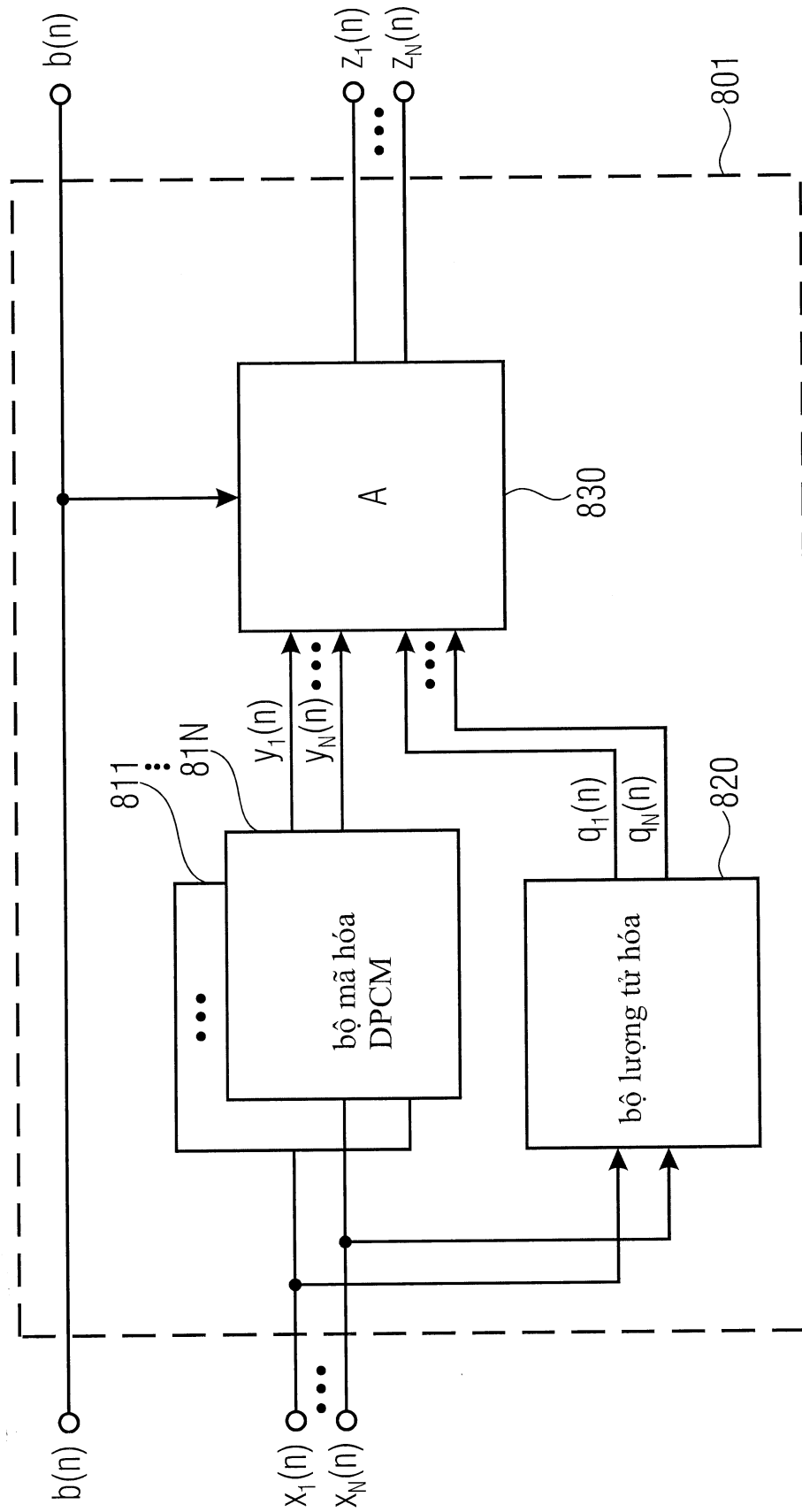


FIGURE 8A

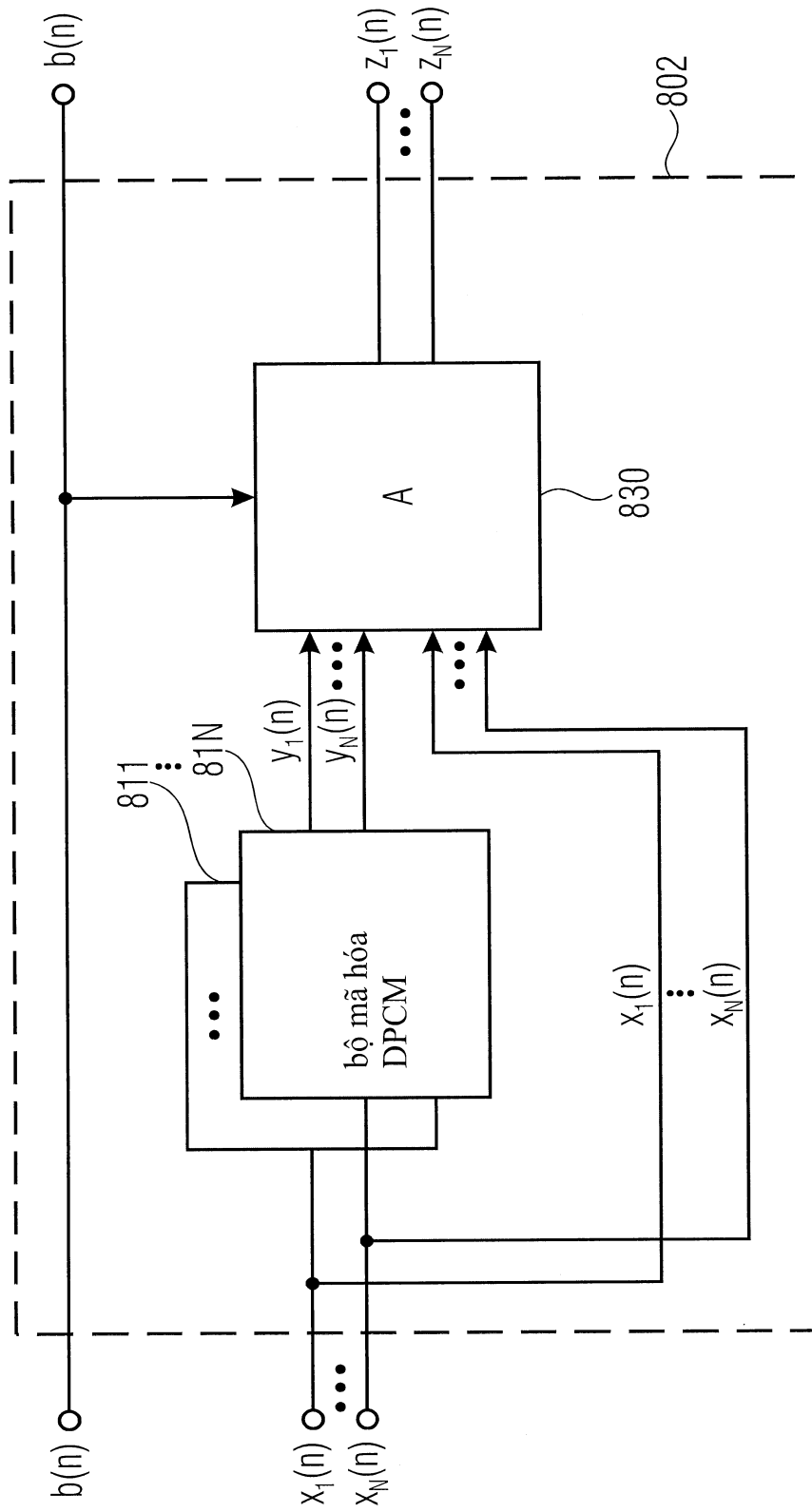


FIGURE 8B

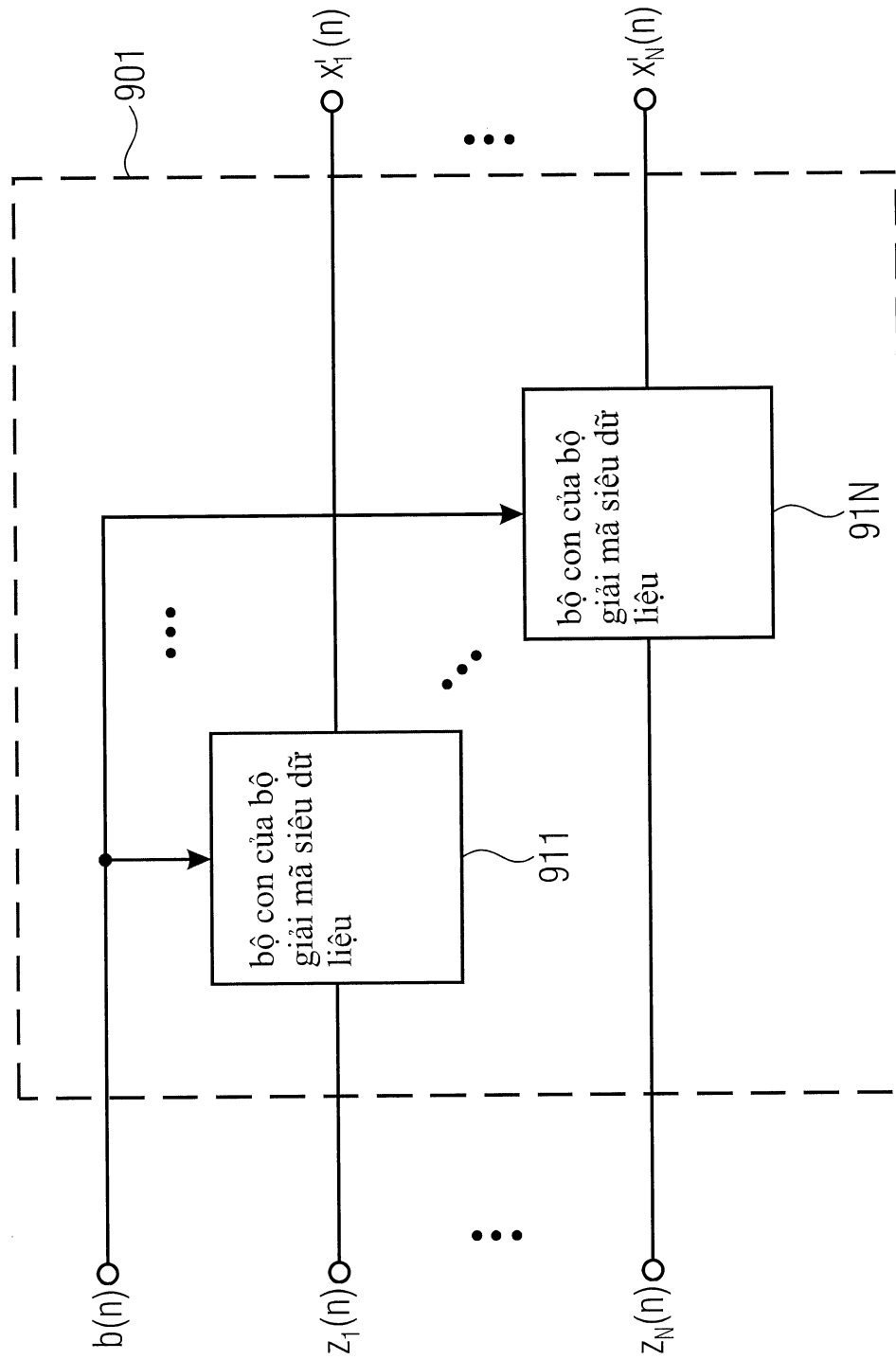


FIGURE 9A

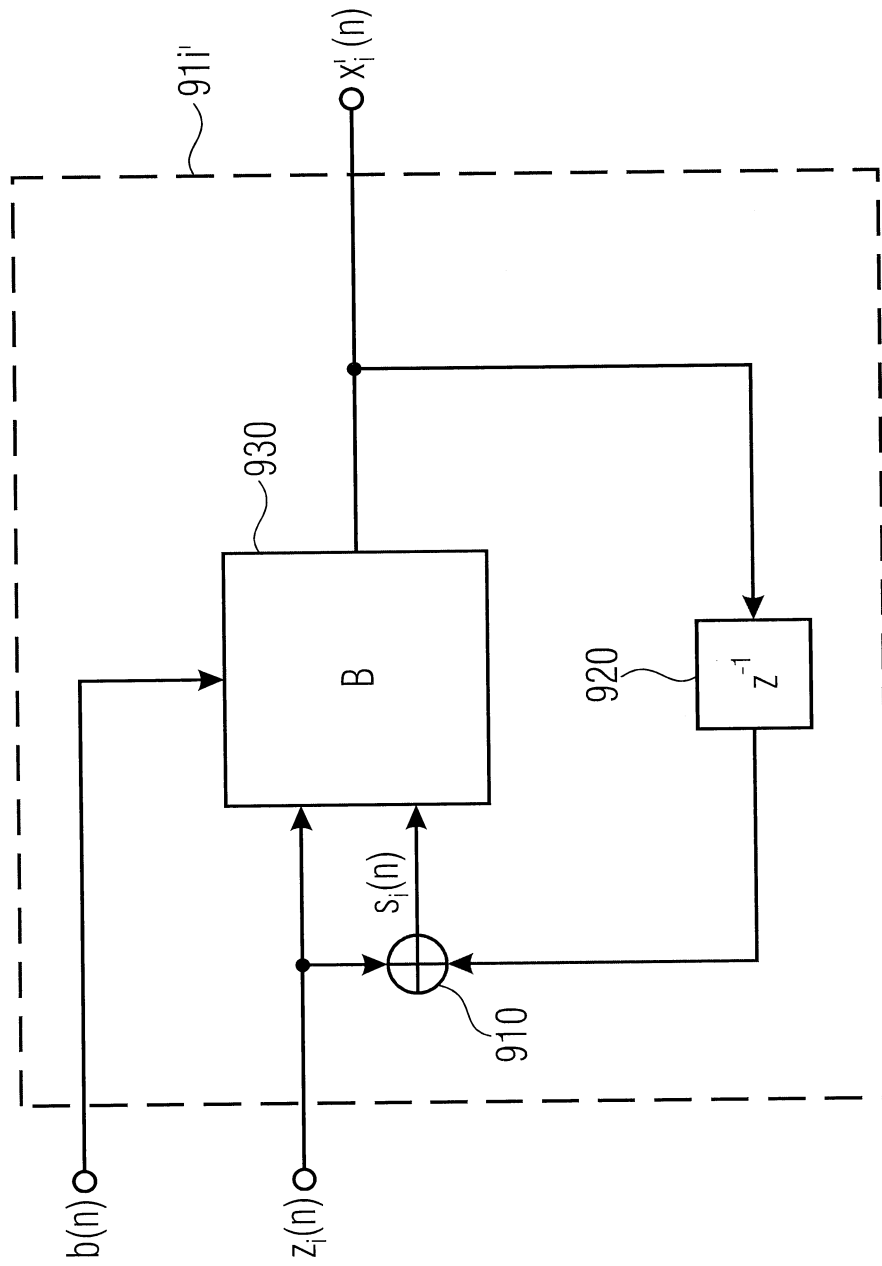
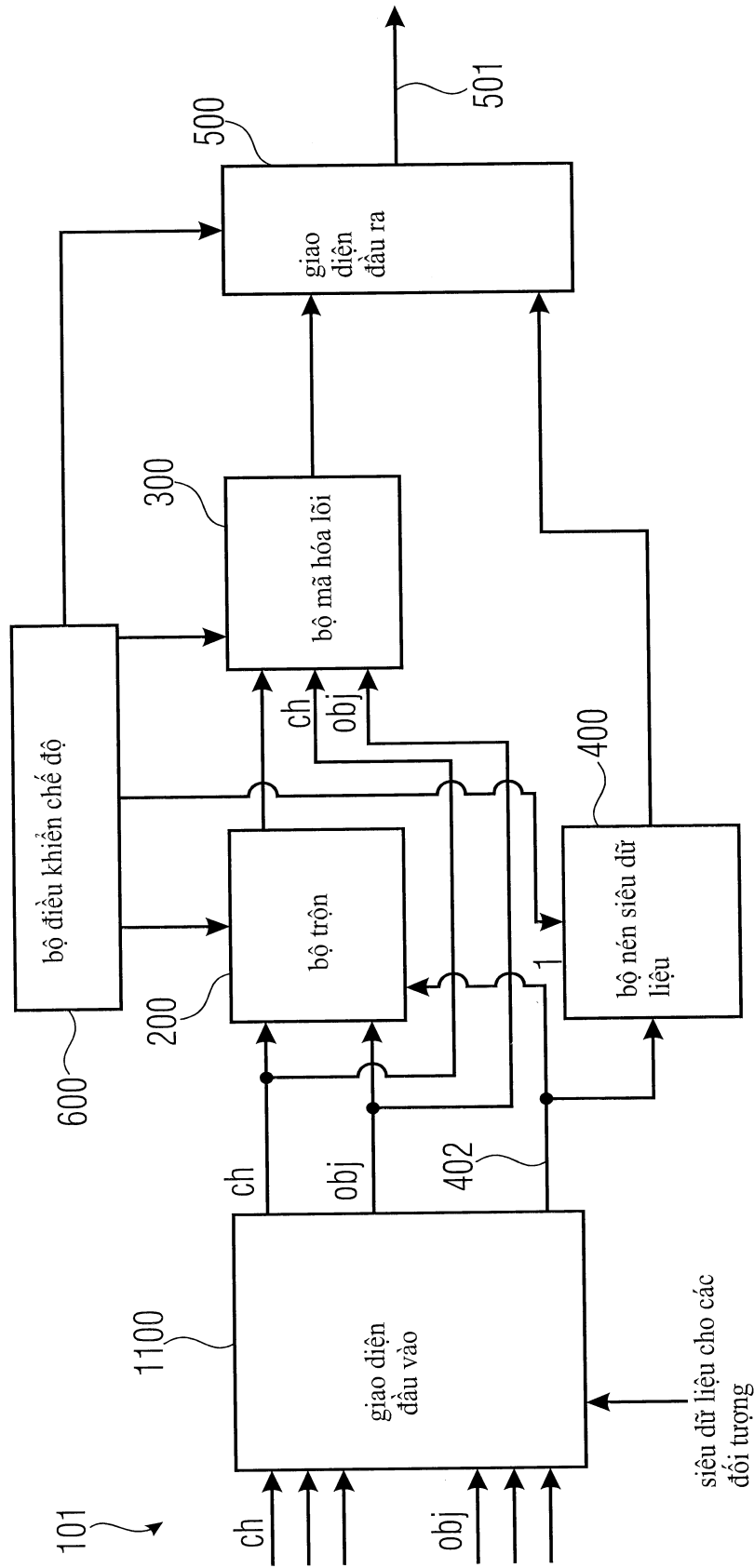


FIGURE 9B



Chế độ 1: mã hóa kênh riêng lẻ/đối tượng
 Chế độ 2: trộn các kênh và cả đối tượng được kết xuất

FIGURE 10
 (BỘ MÃ HÓA)

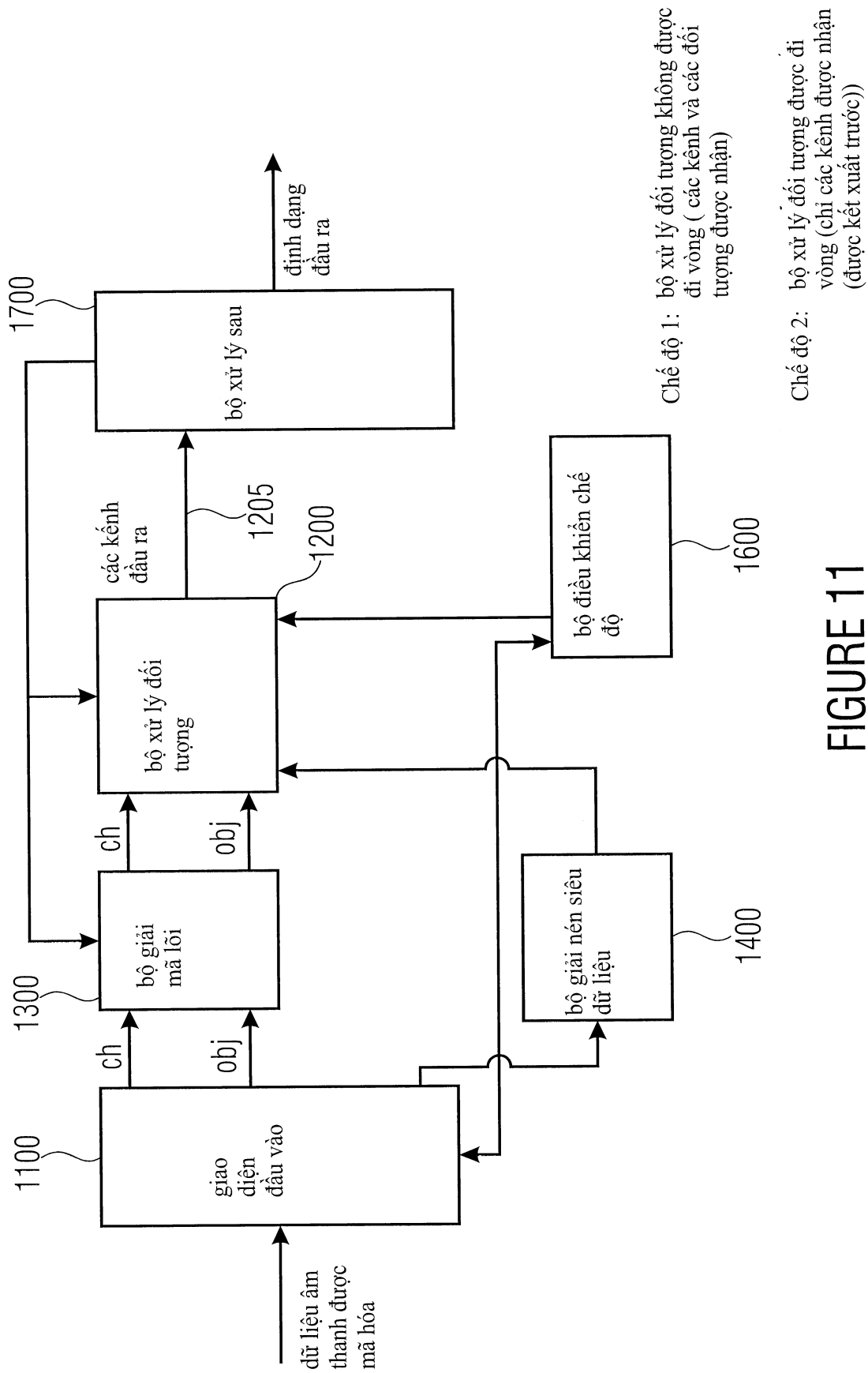


FIGURE 11
(BỘ GIẢI MÃ)

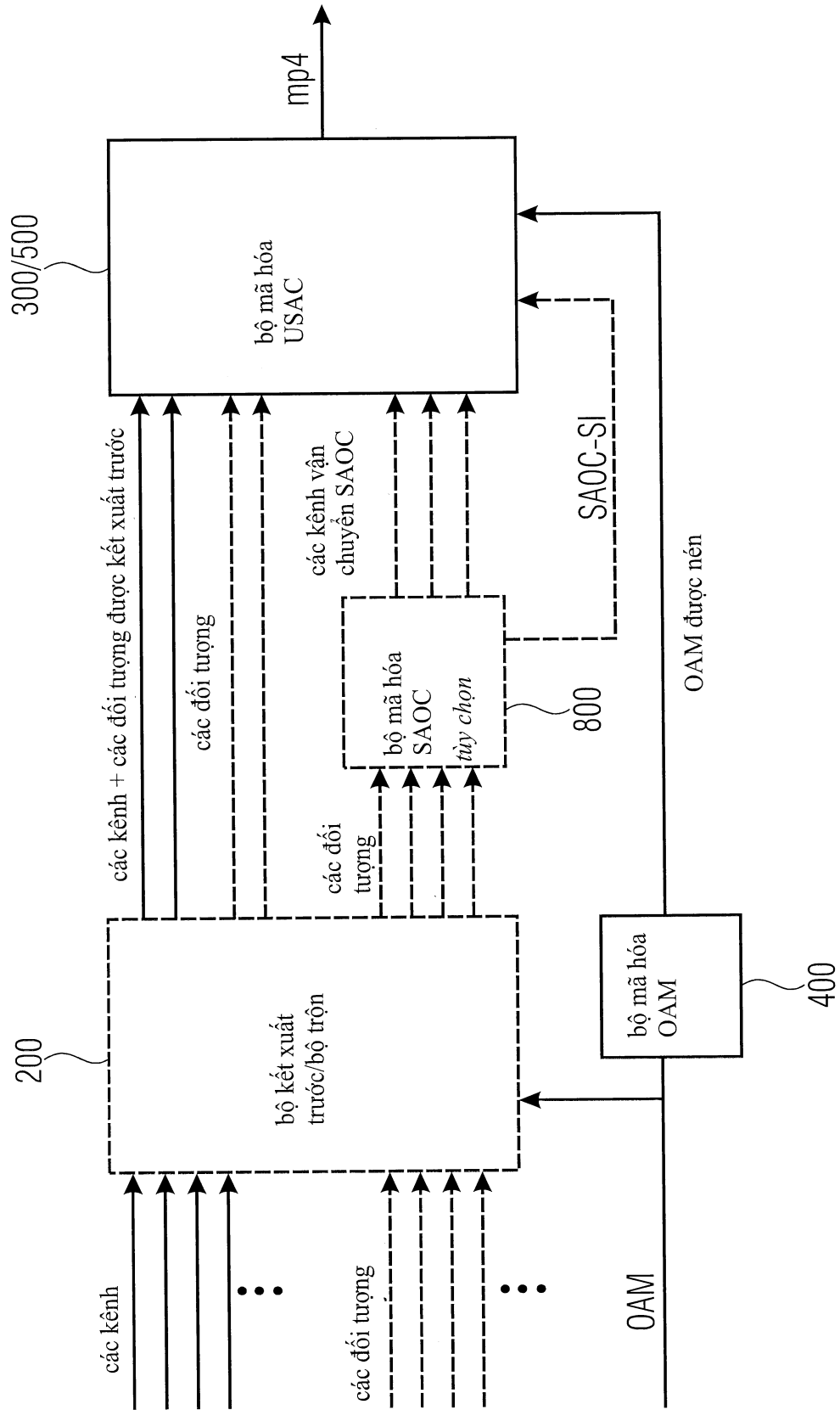


FIGURE 12
(BỘ MÃ HÓA)

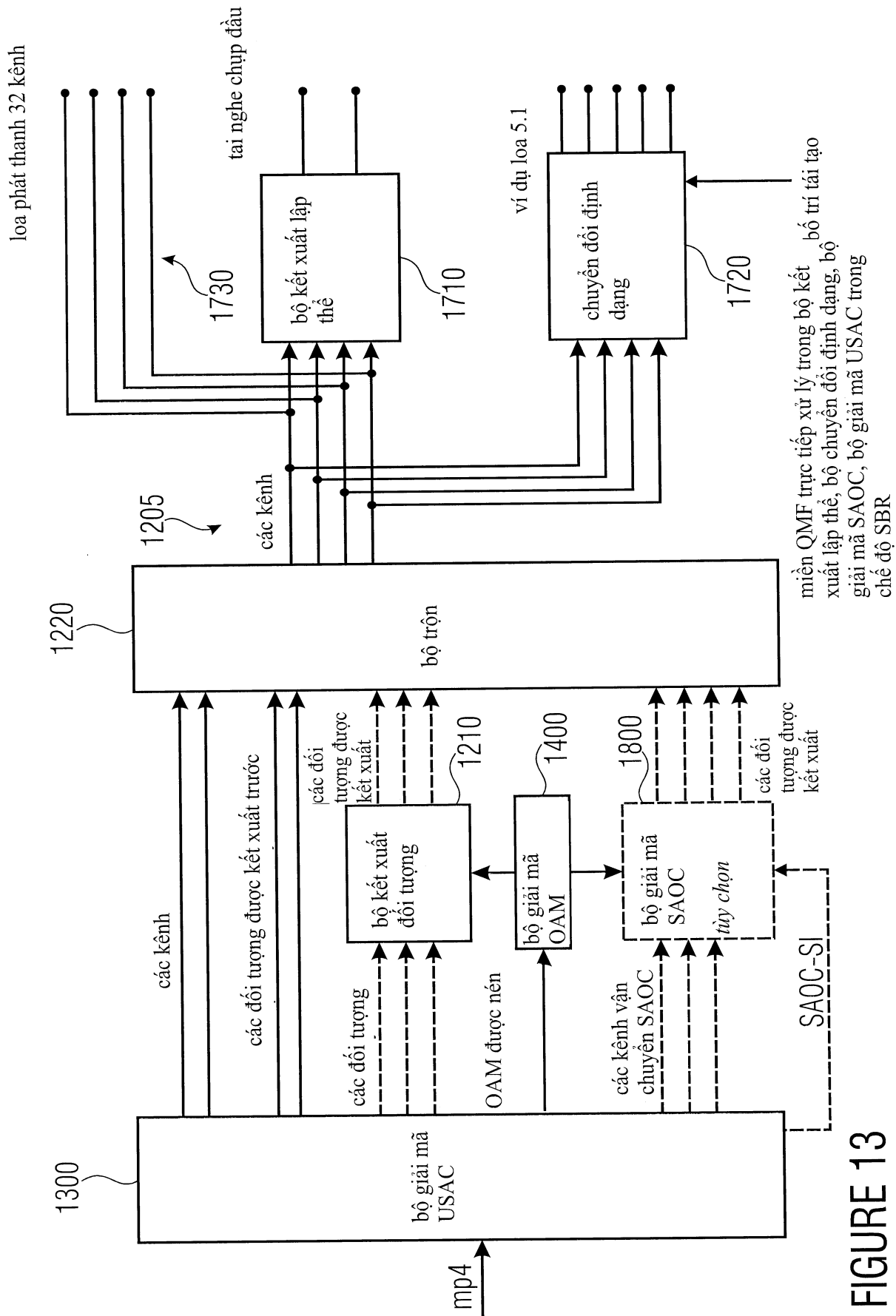


FIGURE 13

(BỘ GIẢI MÃ)

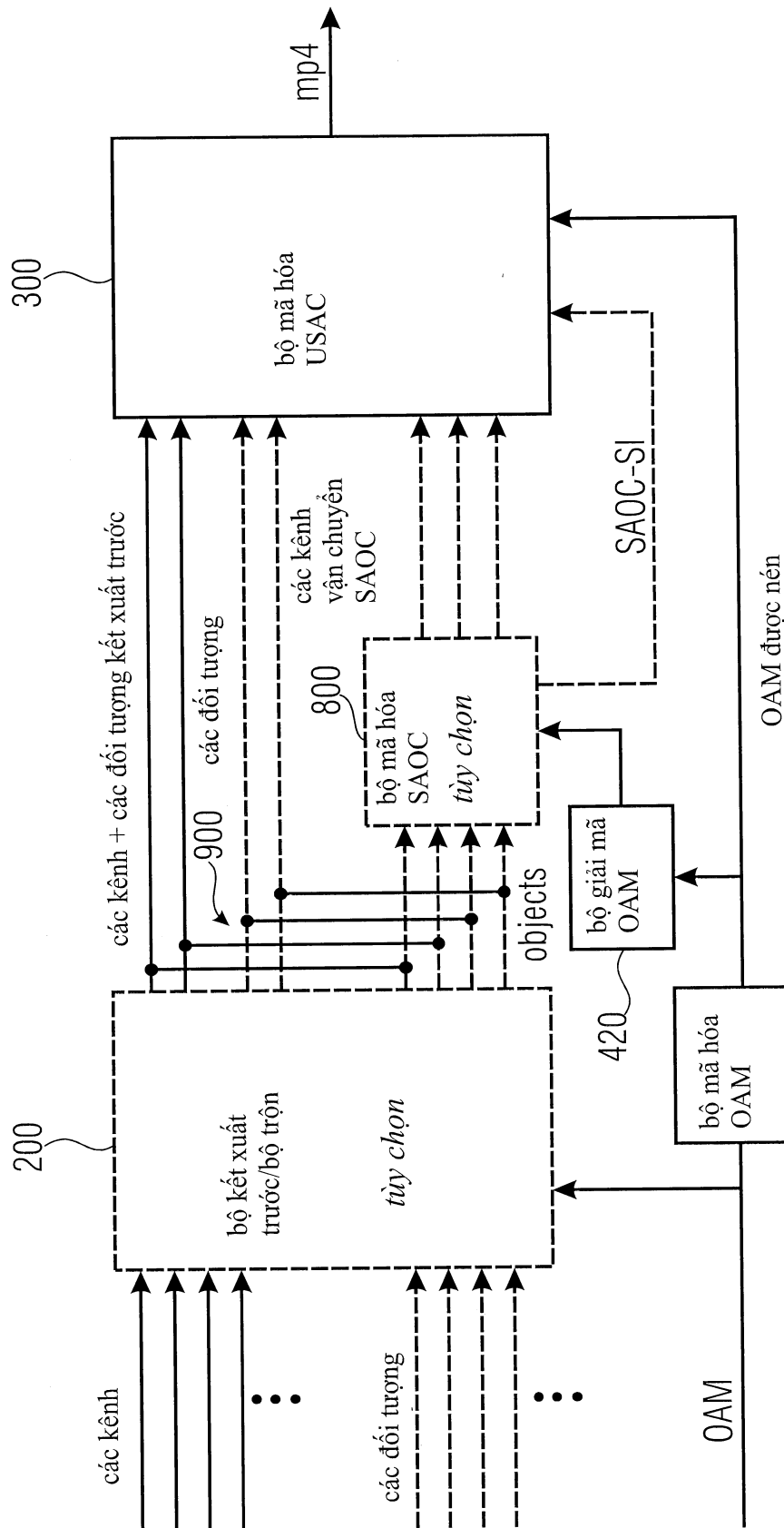


FIGURE 14
(BỘ MÃ HÓA)

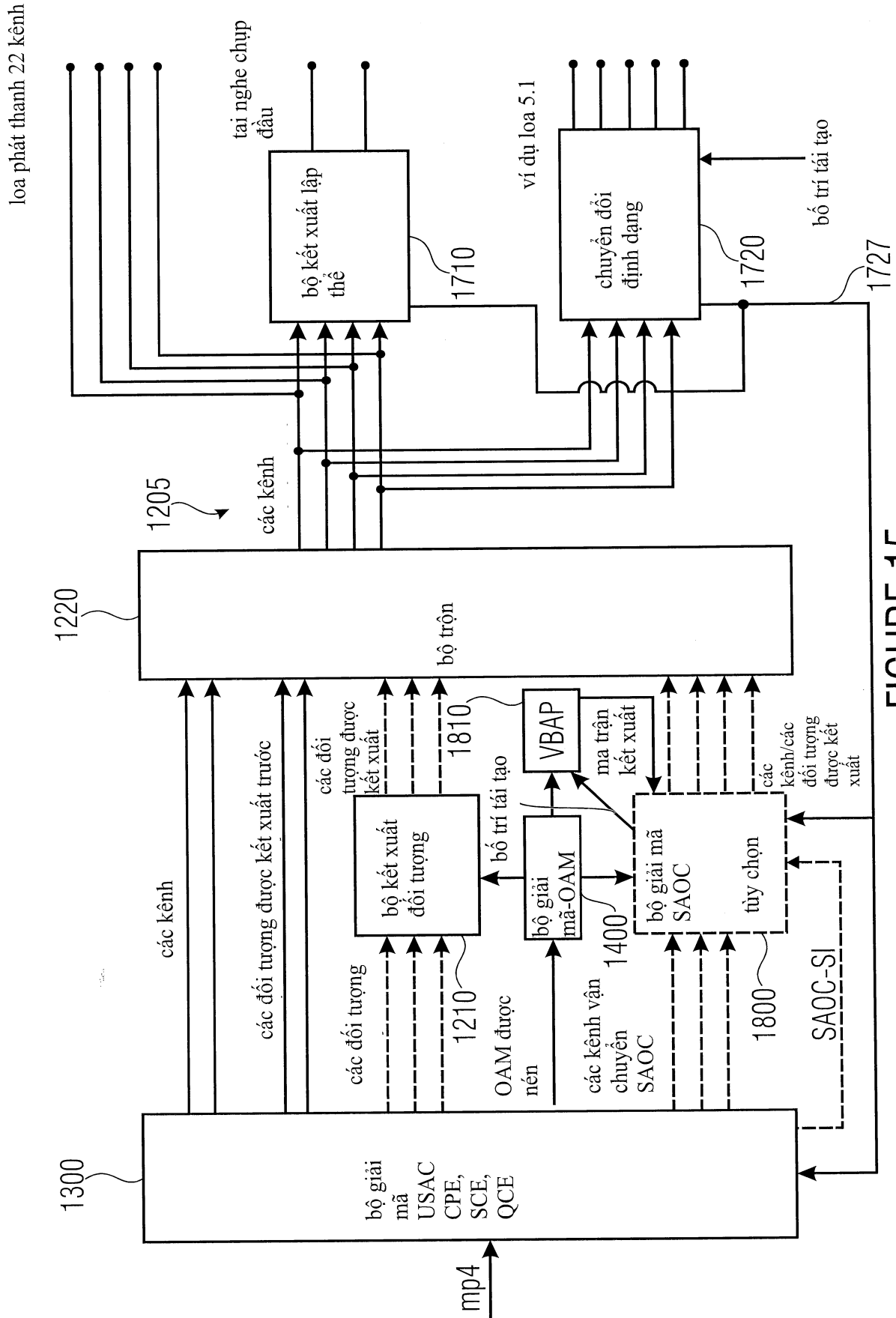


FIGURE 15
(BỘ GIẢI MÃ)